universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

„Democratizing Measurement of Critical Mobile Infrastructure: Security and Privacy in an Increasingly Centralized Communication Ecosystem“

verfasst von / submitted by

Dipl.-Ing. Gabriel Karl Gegenhuber, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der technischen Wissenschaften (Dr. techn.)

Wien, 2026 / Vienna, 2026

Studienkennzahl lt. Studienblatt / degree programme code:	UA A 786 880
Studienrichtung lt. Studienblatt / degree programme:	Informatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Mag. Dr.techn. Edgar Weippl Univ.-Prof. DI Dr. techn. Johanna Ullrich, BSc

Schlage die Trommel und fürchte dich nicht,
Und küsse die Marketenderin!
Das ist die ganze Wissenschaft,
Das ist der Bücher tiefster Sinn.

Trommle die Leute aus dem Schlaf,
Trommle Reveille mit Jugendkraft,
Marschiere trommelnd immer voran,
Das ist die ganze Wissenschaft.

Heinrich Heine, *Doktrin*

Acknowledgements

This thesis was made possible by the support of many people.

I would like to express my sincere gratitude to my advisors, Edgar Weippl and Johanna Ullrich, for their continuous support throughout this endeavor and for granting me the freedom to develop and pursue my own research topics and ideas.

I am deeply grateful to Wilfried Mayer, who introduced me to the world of academia, both the good and the bad, and who was an exceptional advisor and mentor during the early stages of my PhD. I would also like to express my sincere thanks to Adrian Dabrowski, who repeatedly enlightened me with his deep and specialized knowledge of cellular networks and taught me countless invaluable tricks in LaTeX.

Furthermore, I would like to thank my colleagues and co-authors, above all Florian, but also Markus and David, who accompanied me throughout my PhD journey. Working with you was not only productive but always great fun. I would also like to thank Michael Pucher, who supported me on numerous teaching assignments and consistently placed the needs of his students before his own.

Philipp Frenzel was a fantastic collaborator and student assistant. Over the years, he contributed significantly to advancing MobileAtlas and to many of the publications included in this dissertation. I am excited to see all the things he will achieve in the coming years. Together with Maximilian Günther, we spent many late but very enjoyable nights in the lab researching vulnerabilities, creating measurements, and eating pickles (or pizza at Pagliacci's).

I also greatly enjoyed working with Aljosha Judmayer, who always knew how to keep our research within ethical boundaries, thereby helping us avoid legal trouble, and who was, in addition, the ideal person for discussions on cryptography.

I would also like to thank Barbara Limbeck-Lilienau, for juggling an impressive workload while always taking the time to provide clear and thorough answers to administrative questions.

I am deeply grateful to my partner Nadine Caroline for reminding me that there is a life beyond a computer screen and for ensuring that I take time to rest and spend time in nature. Finally, I would like to thank my parents, Eva and Karl, for awakening and encouraging my curiosity at a young age and my sisters, Julia and Hannah, for supporting me throughout my life.

Abstract

Cellular networks serve as the backbone of global communication, providing critical access to telephony and the Internet, often in regions lacking alternatives. However, the growing complexity of these networks, driven by architectural innovations (e.g., Voice over IP, eSIMs) and commercial dynamics (e.g., roaming, virtual operators, zero-rating), remains poorly understood due to the lack of open, scalable, and geographically diverse measurement tools and independent measurement studies.

Moreover, access to mobile networks today is no longer limited to the traditional radio interface. Technologies like Voice-over-WiFi (VoWiFi) offer alternative connectivity paths via third-party Internet infrastructure, extending operator reach into environments with limited cellular coverage. At the same time, over-the-top (OTT) messaging services such as WhatsApp and Signal have become central to modern communication, accounting for a substantial share of global messaging and voice traffic while bypassing traditional operator-controlled channels entirely.

This dissertation addresses these challenges by introducing new approaches for independent, scalable, and reproducible measurements of mobile communication systems without requiring cooperation from network or platform operators. We design, implement, and open-source measurement platforms that enable controlled experiments across cellular radio networks, operator-provided services, and OTT messaging applications. Using these tools, we conduct multi-layer empirical studies and uncover security- and privacy-relevant weaknesses, including inconsistencies in roaming billing and traffic classification, insecure VoWiFi configurations, and metadata leaks in widely used messaging platforms that enable silent user monitoring and denial-of-service attacks. Overall, this dissertation demonstrates that independent, active measurements are essential for understanding the evolving cellular communication system. It provides practical tools and empirical evidence that increase transparency and support future research into the security and privacy of modern mobile communication systems.

Kurzfassung

Mobilfunknetze bilden das Rückgrat der globalen Kommunikation und ermöglichen den Zugang zu Telefondiensten und dem Internet, häufig auch in Regionen, in denen keine alternativen Zugangstechnologien verfügbar sind. Die zunehmende Komplexität dieser Netze, bedingt durch architektonische Innovationen (z. B. Voice over IP, eSIMs) sowie durch kommerzielle Dynamiken (z. B. Roaming, virtuelle Netzbetreiber, Zero-Rating), ist jedoch bislang nur unzureichend erforscht. Ein wesentlicher Grund hierfür ist das Fehlen offener, skalierbarer und geografisch diverser Messwerkzeuge sowie unabhängiger empirischer Messstudien.

Darüber hinaus ist der Zugang zu Mobilfunkdiensten heute nicht mehr ausschließlich auf die klassische Funkschnittstelle beschränkt. Technologien wie Voice-over-WiFi (VoWiFi) ermöglichen alternative Zugangspfade über das Internet und erweitern damit die Reichweite von Netzbetreibern auf Umgebungen mit eingeschränkter Netzabdeckung. Gleichzeitig haben sich sogenannte Over-the-Top-(OTT-)Messaging-Dienste wie WhatsApp und Signal zu zentralen Bestandteilen moderner Kommunikation entwickelt. Sie tragen einen erheblichen Anteil des weltweiten Nachrichten- und Sprachverkehrs, ohne dabei auf die traditionellen, von Mobilfunkbetreibern kontrollierten Signalisierungs- und Abrechnungsstrukturen zurückzugreifen.

Diese Dissertation adressiert die genannten Herausforderungen durch die Entwicklung neuer Ansätze für unabhängige, skalierbare und reproduzierbare Messungen mobiler Kommunikationssysteme, ohne eine Kooperation mit Netz- oder Plattformbetreibern vorauszusetzen. Hierzu werden Messplattformen entworfen, implementiert und als Open-Source-Software veröffentlicht, die kontrollierte Experimente über Mobilfunkzugangsnetze, betreiberseitige Dienste sowie OTT-Messaging-Anwendungen hinweg ermöglichen. Mithilfe dieser Werkzeuge werden umfassende empirische Studien durchgeführt, die sicherheits- und privatsphärerelevante Schwachstellen aufdecken, darunter Inkonsistenzen bei Roaming-Abrechnung und Traffic-Klassifikation, unsichere VoWiFi-Konfigurationen, sowie Metadatenleaks in weit verbreiteten Messaging-Plattformen, die eine unbemerkte Überwachung von Nutzer:innen und Denial-of-Service-Angriffe ermöglichen.

Insgesamt zeigt diese Arbeit, dass unabhängige, aktive Messungen unerlässlich sind, um das sich wandelnde Mobilfunkkommunikationssystem zu verstehen. Sie stellt praxisnahe Werkzeuge und empirische Erkenntnisse bereit, die die Transparenz erhöhen und zukünftige Forschung zur Sicherheit und zum Schutz der Privatsphäre moderner mobiler Kommunikationssysteme unterstützen.

Contents

1 Introduction

Cellular networks are a primary access technology for both the public telephone system and the Internet — often the only available option in many parts of the world [SB24]. Beyond their role in personal and business communication, they are vital for crisis response and emergency scenarios. Technologies such as roaming, virtual network operators (MVNOs), and travel (e)SIMs interconnect previously isolated infrastructures, forming a compound system embedded in a complex global web. Free-roaming agreements — most notably within the European Union — further blur national boundaries, offering users a seamless and unified mobile experience across borders.

Despite their critical role in today's society, cellular networks remain among the least transparent and least independently measurable components of the global Internet. This limited measurability is not a consequence of limited adoption, but of the absence of controlled and scalable measurement tools that reflect the unique architecture and global scope of cellular access networks. As a result, independent insight into provider practices — particularly under roaming conditions — remains severely limited.

In addition to traditional 3GPP-specified telephony and messaging, third-party instant messaging platforms such as WhatsApp play a central role in today's mobile communication ecosystem. Although they carry a substantial share of message traffic, these services are developed and controlled by private companies, introducing similar transparency and auditability challenges at the application layer, and ultimately requiring a significant degree of blind trust.

To improve transparency, security, and resilience in this sector, independent and accessible measurement tools are urgently needed — tools that can serve researchers, regulators, and industry stakeholders alike for auditing and controlled testing. As complexity increases, so do potential vulnerabilities and corner cases. Without proper tools to introspect and analyze the inner workings of these networks and applications, critical issues may remain undetected, compromising both security and infrastructure resilience.

To address this gap, this dissertation proposes new measurement tools and presents independent empirical studies of mobile communication systems, examining multiple layers of the ecosystem, from the radio access layer to VoWiFi and third-party instant messaging services, with the common goal of enabling independent and reproducible measurements and strengthening security and privacy.

1.1 Thesis Statement

This dissertation argues that democratized, independent measurement (i.e., controlled and scalable experiments without operator cooperation) enables the systematic discovery and quantification of security, privacy, and policy failures across mobile communication

layers—failures that have direct consequences for end users and would otherwise remain invisible.

Across the presented case studies, the key methodological contribution is not merely measurement at scale, but *independent* measurement: experiments that remain feasible even when operators or platforms do not collaborate, and that can be replicated across countries, providers, and time. Concretely, this dissertation argues that democratized measurement enables:

- **coverage** across operators, countries, and deployment settings,
- **control** over experimental conditions without privileged operator access or platform cooperation, and
- **comparability** of results across technology-defined layers through standardized, repeatable experiments and shared artifacts.

These properties enable systematic discovery and quantification of vulnerabilities and risks that are unlikely to surface through documentation or the operators and platforms themselves.

1.2 Cellular Network Research Gap

To evaluate the thesis statement across layers, this dissertation combines reusable measurement artifacts with empirical studies. Each layer, defined by its underlying technology, implies different trade-offs in independence, control, and scale and exposes distinct security and privacy risks.

For measurements in the cellular network domain, we distinguish between passive and active measurements.

Measurements with **passive data** are often conducted with the help of industry partners, e.g., Internet Service Providers (ISPs) or Mobile Network Operators (MNOs). These measurements rely on data collected from operational networks, typically at a large scale. However, such data is rarely publicly accessible and generally requires privileged access to core infrastructure. As a result, passive measurements are largely inaccessible to independent researchers or early-stage academic efforts.

On the other hand, **active measurements** have proven to be a vital tool for conducting independent and open research. The two most common practices for active measurements include (i) in-situ- or in-vivo measurements, and (ii) exclusive measurement setups.

App-based approach. *In-situ* ("at location") and *in-vivo* ("within the system") measurements often run on a (volunteer's) phone that is also used for other tasks. This method increases coverage by reducing the financial burden of participating in a given study, thereby adding more measurement units. While appropriate for, e.g., user studies, this method might impact the accuracy of technical measurements since the non-exclusivity blurs the distinction between measurement (*signal*) and background- or user activity (*noise*). The user's mobility also interferes with the common goal of steady (consistent and repeatable) measurement environments. Volunteers also bear

responsibility for any billing charges incurred during the experiment, rendering this approach impractical for certain scenarios, particularly those involving roaming or phone calls. Moreover, the measurement operator often lacks full control over key parameters, such as the access technology in use (e.g., 3G, 4G, or 5G), and has limited visibility into critical data. For instance, billing statements are typically accessible only to the volunteer, restricting opportunities for external validation and comprehensive analysis.

Deploying dedicated hardware. Exclusive measurement setups require a separate test unit in a controlled environment. While some external factors (such as the network's utilization or radio noise) are commonly outside of the researchers' sphere of influence, other factors can be controlled; this includes the distance to a cell tower, the used access technology, and background data usage.

While deploying, operating, and maintaining a large fleet of distributed measurement probes remains a logistical challenge, such platforms can, in principle, be democratized and scaled through community-driven efforts. This model has proven successful in the context of independent fixed-line Internet measurements, most notably through community-driven efforts like RIPE Atlas [RIP15, AWR14].

Replicating this model in the cellular domain, however, presents fundamental challenges. Requiring the probe host to supply and manage their own SIM card introduces billing liability, which not only deters participation but also severely limits the number of network operators measurable from a single probe. Supporting a diverse set of SIM cards across different locations or within roaming scenarios does not scale, thereby impeding broader deployment and coverage.

Taken together, these constraints create a structural barrier for independent research, particularly for academic groups, regulators, and researchers without privileged operator access. Meaningful security and privacy analysis of cellular networks requires controlled, repeatable, and globally diverse measurements, precisely the properties that existing approaches fail to provide. Without these properties, security flaws, billing inconsistencies, and privacy leaks may remain systematically invisible.

In summary, existing measurement approaches force researchers to trade off scalability, control, and independence. Consequently, none of these approaches is sufficient for conducting controlled, repeatable, and globally diverse measurements of cellular networks without operator cooperation.

1.3 Research Questions

Motivated by the limitations of existing measurement approaches and guided by the thesis statement above, this dissertation investigates the following research questions:

RQ1 How can cellular networks be measured independently, in a controlled manner, and at scale, without requiring cooperation from network operators?

RQ2 How do roaming scenarios, alternative access technologies, and operator billing mechanisms interact in real-world cellular network deployments?

RQ3 How do security and privacy properties, as well as measurement capabilities, differ across layers of the mobile communication ecosystem, ranging from cellular access networks to operator-provided services and OTT messaging platforms?

RQ1 and RQ2 focus on how independent measurement can characterize operator behavior and operator-provided services without requiring cooperation. RQ3 applies the same measurement perspective to increasingly centralized OTT messaging platforms, where Internet-scale observation can reveal ecosystem-wide attack surfaces and privacy risks.

RQ1 is addressed by Chapters 2–3 (measurement platforms and scalability).
RQ2 is addressed by Chapters 3–6 (roaming, billing, alternative access).
RQ3 is addressed by Chapters 5–8 (cross-layer security/privacy properties and observability). Chapter 4 departs from cellular measurements and demonstrates how a distributed community-driven measurement platform (RIPE Atlas) can reveal Internet centralization that affects anonymity in Tor, reinforcing the dissertation's broader theme of democratized, independent measurements and the risks introduced by centralization in global communication systems.

1.4 New Measurement Approaches and Results

To answer these research questions, this dissertation introduces new approaches to large-scale (i.e., international) security and privacy measurements in cellular networks and mobile communication systems. It follows a multi-layer measurement perspective, where each layer targets a different part of the mobile communication ecosystem and exposes distinct measurement challenges and security properties. This progression from radio access networks to VoWiFi and ultimately to OTT messaging platforms mirrors the ongoing shift of control and observability from decentralized, operator-bound infrastructure toward globally centralized, Internet-based platforms.

Figure 1.1 illustrates the three complementary research perspectives considered in this work: the cellular radio access layer, the Voice over Wi-Fi (VoWiFi) layer, and third-party instant messaging services. Each angle exposes a different form of centralization, opacity, or control.

1.4.1 Democratizing Radio Layer Measurements

Large-scale and controlled experiments at the cellular radio access layer require a local point of presence to interact with base stations, making such measurements difficult to scale across operators, countries, and roaming scenarios. Existing approaches typically rely on physically co-locating subscriber identities (i.e., SIM cards) with measurement devices, which introduces significant logistical and operational constraints. As a result, repeatable radio-layer measurements remain largely inaccessible to independent researchers.

To address this gap in dedicated measurement platforms for cellular networks, this dissertation introduces the MobileAtlas framework [GMWD23], based on low-cost hardware (e.g., a Raspberry Pi 4 and a COTS cellular modem). Conceptually, MobileAtlas is inspired by RIPE Atlas: both aim to democratize measurement

Figure 1.1: Overview of the three research angles addressed in this work. Large-scale measurements at the radio access layer require a local point of presence for connectivity to cellular base stations, which imposes significant logistical and deployment challenges, while Wi-Fi calling and instant messaging rely on standard Internet connections. Although the first two technologies are standardized by 3GPP, all three lack adequate tooling for comprehensive auditing and independent security testing. The progression from radio access networks to VoWiFi and finally to OTT messaging platforms reflects an increasing degree of centralization, in which control and observability shift from decentralized, operator-bound infrastructure toward globally centralized, Internet-based services.

by enabling distributed experiments from many independent vantage points. As shown in Figure 1.2, MobileAtlas geographically decouples the SIM card from the modem by tunneling the SIM card's protocol over the Internet and emulating its signal at the modem. With this approach, measurement probes and SIM cards can be at different locations and thus flexibly shared with other participants, without the need for permanently installed SIM cards or any physical movement of components. A SIM card hosted at a fixed location can be virtually connected to measurement probes around the world within seconds, each offering a local breakout to its respective cellular infrastructure. This enables the SIM to be quickly evaluated within diverse network environments. Additionally, the platform utilizes Linux namespaces to isolate the modem connection from any background activity and provide sterile measurement access for fine-grained measurements.

While SIM tunneling inherently introduces additional round-trip latency, authentication and session key generation remain robust to such delays, as they are designed to tolerate network congestion and retransmissions caused by weak signal conditions. We validate this robustness experimentally across a wide range of operators, including intercontinental tunneling scenarios.

Furthermore, since all ISO 7816 [ISO06] APDU commands between the modem and SIM card are relayed through the tunnel, the corresponding traffic can be inspected to gain further insights into protocol behavior and SIM-card interactions (e.g., proactive SIM commands). The resulting measurement platform provides a controlled environment, scalability, and cost-effectiveness. Moreover, it is extensible and fully open-source[1], enabling other researchers to contribute locations, SIM cards,

[1] `https://github.com/sbaresearch/mobile-atlas`

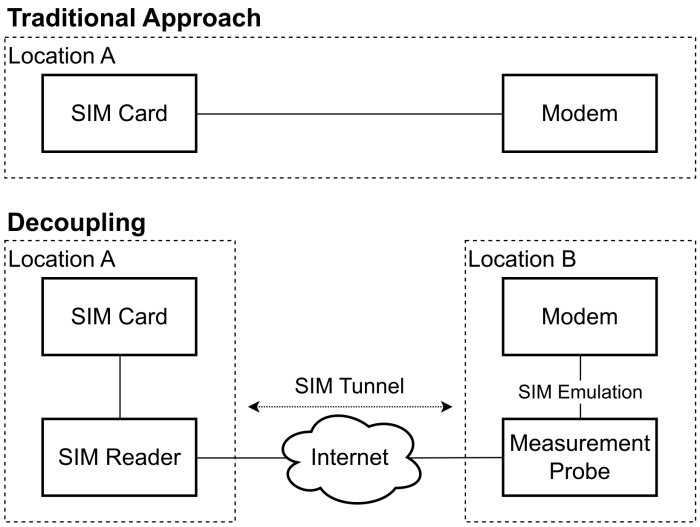


Figure 1.2: SIM card and modem are geographically decoupled via a SIM tunnel relaying the protocol over the Internet.

and measurement scripts.

In an initial study [GMWD23], we use the platform to inspect and compare network configurations in both domestic and (home-routed) roaming scenarios. This analysis revealed weak firewall configurations during operator migrations from Carrier-Grade NAT (CGNAT) to IPv6-based architectures. Additionally, we uncovered instances of hidden SIM card activity, such as silent SMS messages to the operator sent via proactive SIM commands. To further investigate signaling behavior, we generated test calls and analyzed subtle variations in ringback tones. These differences enable detailed fingerprinting of mobile operators and can inadvertently reveal a user's operator and, by extension, their country-level location.

Moreover, we use the platform for fine-grained traffic experiments, investigating operators' billing mechanisms under differential pricing schemes (i.e., zero-rating offers) [GMW22]. Our analysis reveals potentially problematic behavior (i.e., inadvertent billing of zero-rated traffic) at nearly all operators examined and identifies possible vectors for free-riding attacks (e.g., spoofed Host- or SNI- headers).

Low-cost SIM tracing. Since the SIM tunnel relies on low-cost, readily available peripherals (i.e., a UART interface and GPIO ports), we further reduce the economic barrier by implementing and publishing the design for the Raspberry Pi Pico [GFD25]. This enables SIM-tracing capabilities (i.e., inspecting, rewriting, and relaying traffic) at a hardware cost of as little as 5 USD. In addition to supporting physical SIM cards, our system also supports *in vitro* analysis of eSIMs by leveraging the SIM Access Profile (SAP) [blu03] available on Android devices.

1.4.2 Leveraging Alternative Access Technologies

In current cellular network generations (4G, 5G) the IMS (IP Multimedia Subsystem) plays an integral role in terminating voice calls and short messages. Many operators

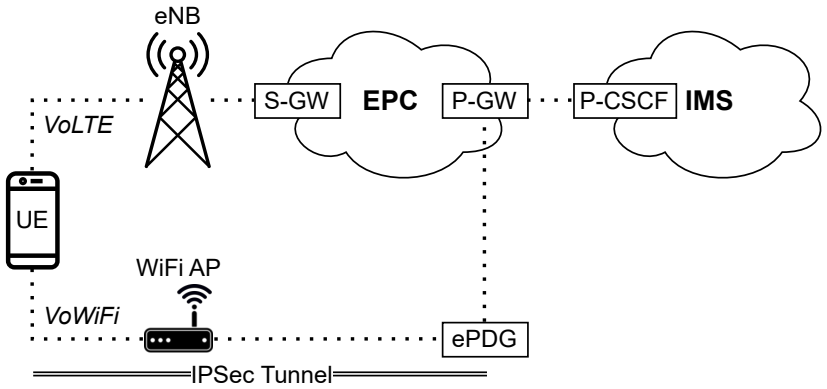


Figure 1.3: (Simplified) LTE network architecture for VoLTE and VoWiFi. The ePDG server is exposed to the Internet.

use VoWiFi (Voice over Wi-Fi, also Wi-Fi calling) as an alternative network access technology to complement their cellular coverage in areas where no radio signal is available (e.g., rural territories or shielded buildings). In practice, VoWiFi is often prioritized over VoLTE, making it a primary calling path in many deployments and increasing its relevance for communication security and privacy. Since this technology requires operators to publicly expose parts of their infrastructure to the Internet, it also creates a new measurement vector for large-scale cellular measurements (cf. ePDG in Figure 1.3). We leverage these alternative access technologies by conducting Internet-based scanning and reconnaissance measurements, thereby offering a comprehensive global perspective on current deployment states and operational practices.

Our work evaluates the current deployment status of VoWiFi among worldwide operators and uses commercial VPN subscriptions to analyze existing geoblocking measures on the IP layer [GFW24b]. More specifically, we show that a substantial share of operators implement geoblocking at the DNS- or VoWiFi protocol level, and highlight severe drawbacks in terms of emergency calling service availability.

Moreover, we found critical vulnerabilities in commercial VoWiFi implementations (e.g., security downgrade attacks [GHF+24]) and configurations (e.g., outdated ciphers [GFW24a], private-key reusage [GHF+24]), both affecting the communication security and privacy of hundreds of millions of users.

We open-source both our VPN-driven scanning solution for rapid distributed Internet measurements[2] and our scripts to evaluate global VoWiFi deployments and configurations[3].

1.4.3 Evaluating Over-the-Top Messaging Applications

Beyond the radio layer and operator-controlled services such as VoWiFi or RCS, a significant share of voice and messaging traffic is now carried by OTT messaging platforms. Popular applications like WhatsApp, iMessage, and Signal account for a large portion of global communication volume, bypassing traditional signaling infrastructures and operator billing mechanisms entirely. Evaluating the security, privacy,

[2] https://github.com/sbaresearch/scanywhere/
[3] https://github.com/sbaresearch/vowifi-epdg-scanning

and network behavior of these platforms is essential, as they shape user experience and often serve as the primary channel in regions with censorship, surveillance, or restricted network access. However, since many of these applications are closed-source and developed by commercial entities, comprehensive auditing and in-depth evaluation of their security and privacy properties remain challenging. Although OTT platforms are not cellular technologies, they now dominate user-facing mobile communication and must therefore be included in any realistic and comprehensive security analysis.

Despite offering end-to-end encryption (E2EE), our measurements show that many widely used messaging applications remain susceptible to privacy leaks. For example, refilling mechanisms for ephemeral encryption keys can unintentionally expose a user's online status on individual devices [GFGJ25] and reveal information about their used operating system. In addition, global synchronization issues in backend infrastructures can lead to denial-of-service (DoS) conditions for targeted users. To support independent verification and raise user awareness, we open-source a research prototype that enables querying key bundles for arbitrary numbers registered on WhatsApp[4].

Moreover, acknowledgment messages for successful message transmission (i.e., delivery receipts) can be used to silently, precisely, and consistently monitor a victim's connection round-trip time (RTT), thereby leaking activity states or behavioral patterns that may disclose a person's location or routines [GGM+25]. Such *silent pings* can also be abused for resource exhaustion attacks, allowing adversaries to drain a user's battery or consume mobile data quota without their awareness.

Recent work shows that dominant OTT messaging platforms (such as WhatsApp) may increase centralization and give rise to enumeration vulnerabilities, exposing roughly 3.5 billion phone numbers and corresponding to a significant fraction of the global mobile user base [GFG+26].

These findings highlight the close interplay between OTT applications and cellular networks and underline the need to examine both layers as part of a comprehensive security and privacy analysis.

1.5 Contributions

This dissertation advances the state of independent cellular network measurement by developing new measurement approaches and providing empirical insights into modern mobile communication systems across multiple layers. In line with the thesis statement, it shows that democratized, controlled, and scalable measurements can reveal and quantify cross-layer security, privacy, and policy failures without operator cooperation. Collectively, these contributions establish a general methodology for independent, scalable security measurements in mobile communication systems. The main contributions of this work are as follows:

- **A scalable and independent platform for radio-layer cellular measurements.** This dissertation introduces MobileAtlas, a low-cost, open-source measurement framework that enables controlled and repeatable cellular measurements without operator cooperation by geographically decoupling SIM cards

[4] `https://github.com/sbaresearch/prekey-pogo`

from measurement probes.

- **Empirical analysis of roaming behavior and billing mechanisms.** Using MobileAtlas, this work provides fine-grained measurements of domestic and roaming scenarios, uncovering inconsistencies in IP addressing, firewall configurations, proactive SIM behavior, and traffic classification under zero-rating schemes.
- **Large-scale evaluation of VoWiFi availability, geoblocking, and security.** This work presents a global-scale analysis of commercial VoWiFi deployments, revealing deployment gaps, geoblocking practices, and security-relevant misconfigurations affecting service availability, emergency calling, and user privacy for subscribers worldwide.
- **Systematic exposure of security and privacy weaknesses in prevalent OTT messaging applications.** This dissertation investigates widely used instant messaging applications, showing how protocol design and backend behavior can leak metadata, enable silent monitoring, and expose users to denial-of-service and enumeration risks.
- **Open-source tools and datasets for reproducible mobile network research.** Measurement platforms and analysis tools developed in this dissertation are released as open source, enabling independent validation, reuse, and extension by the research community.

1.6 Publications (Conference and Journal Papers)

[GMWD23]
Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, Adrian Dabrowski. *MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research.* In 32nd USENIX Security Symposium (USENIX Security), 2023.

[GMW22]
Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl. *Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs.* In IEEE Global Communications Conference (GLOBECOM), 2022.

[GMH+23]
Gabriel K. Gegenhuber, Markus Maier, Florian Holzbauer, Wilfried Mayer, Georg Merzdovnik, Edgar Weippl, Johanna Ullrich. *An Extended View on Measuring Tor AS-level Adversaries.* In Computers & Security (COSE) 132, 2023.

[GFW24b]
Gabriel K. Gegenhuber, Philipp É. Frenzel, Edgar Weippl. *Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi.* In 22nd Annual International Conference on Mobile Systems, Applications and Services (MobiSys), 2024.

[GHF+24]
Gabriel K. Gegenhuber, Florian Holzbauer, Philipp Frenzel, Edgar Weippl, Adrian Dabrowski. *Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments.* In 33rd USENIX Security Symposium (USENIX Security), 2024.

[GGM+25]
Gabriel K. Gegenhuber, Maximilian Günther, Markus Maier, Aljosha Judmayer, Florian Holzbauer, Philipp É. Frenzel, Johanna Ullrich. *Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers.* In 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2025. **Distinguished with the Best Paper Award.**

[GFGJ25]
Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Aljosha Judmayer. *Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp's Handshake Mechanism.* In 19th USENIX WOOT Conference on Offensive Technologies (WOOT), 2025.

[GFG+26]
Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Johanna Ullrich, Aljosha Judmayer. *Hey there! You are using WhatsApp: Enumerating Three Billion Accounts for Security and Privacy.* In 33rd Annual Network and Distributed System Security Symposium (NDSS), 2026. **Complementary work, not part of the thesis.**

1.7 Workshops, Extended Abstracts, and Posters

[GFW24a]
Gabriel K. Gegenhuber, Philipp É. Frenzel, Edgar Weippl. *Never Gonna Give You Up: Exploring Deprecated NULL Ciphers in Commercial VoWiFi Deployments.* Poster & Extended Abstract. In 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), 2024.

[GF25]
Gabriel K. Gegenhuber, Philipp É. Frenzel. *Scanywhere: Distributed Internet Scanning Leveraging Commercial VPN Subscriptions.* Poster & Extended Abstract. In 9th Network Traffic Measurement and Analysis Conference (TMA), 2025. **Distinguished with the Best Poster Award.**

[GFD25]
Gabriel K. Gegenhuber, Philipp É. Frenzel, Adrian Dabrowski. *SIMulator: SIM Tracing on a (Pico-)Budget.* Poster & Extended Abstract. In 18th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), 2025.

[Geg25]
Gabriel K. Gegenhuber. *Security and Privacy Measurements in Cellular Net-*

works: Novel Approaches in a Global Roaming Context. Extended Abstract (Doctoral Symposium). In 32nd ACM Conference on Computer and Communications Security (CCS), 2025.

[GLHS25]
Gabriel K. Gegenhuber, Leonid Liadveikin, Florian Holzbauer, Sebastian Strobl. *A Relay a Day Keeps the AirTag Away: Practical Relay Attacks on Apple's AirTags.* Poster & Extended Abstract. In 41st Annual Computer Security Applications Conference (ACSAC), 2025.

1.8 Artifacts and Impact

This dissertation follows an artifact-driven research approach, where measurement studies are accompanied by open-source tools and responsible disclosure procedures.

1.8.1 Open Source Projects and Artifacts

mobile-atlas: Cellular measurement platform for scalable roaming measurements

scanywhere: Global Internet scanning solution driven by commercial VPN solutions

vowifi-epdg-scanning: Evaluating VoWiFi cipher and key exchange methods in commercial networks

mbn-mcfg-tools: Parsing and packing proprietary Qualcomm MBN files (used for modem configurations)

prekey-pogo: Requesting device directory and prekey information of arbitrary WhatsApp users

device-activity-tracker: Monitoring WhatsApp devices via silent pings. *This project is based on our publication, but was developed by an external party.*

1.8.2 Impact and Responsible Disclosure

Prior to the public release of the publications and artifacts included in this dissertation, all identified security and privacy issues were reported to the affected parties in accordance with responsible disclosure practices.

The findings of this work were subsequently shared with the broader security research and practitioner community through presentations at two DEF CON conferences, one Black Hat conference, and several additional venues (e.g., in front of the GSMA Fraud and Security Group).

In addition, the disclosed vulnerabilities and their implications were reported by high-profile international media outlets, contributing to broader public awareness of security and privacy risks in mobile communication systems.

The following vulnerability identifiers and disclosures are directly associated with the results presented in this dissertation:

CVE-2025-20647 (medium severity): MediaTek, VoLTE/VoWiFi Denial of Service due to NULL Pointer Dereference at SIP

CVE-2024-22064 (high severity): ZTE, Private-Key Sharing among Global Operators

CVE-2024-20069 (high severity): MediaTek, Downgrade Vulnerability at VoWiFi IKE Protocol

CVD-2024-0089: GSMA, Deprecated VoWiFi Configurations and Coordinated Disclosure of CVE-2024-22064, CVE-2024-20069

Signal: The official Signal GitHub repository contains an ongoing discussion regarding potential countermeasures against silent ping attacks.

WhatsApp: Security reports submitted as part of this work were acknowledged by WhatsApp and are currently in the process of being addressed.

1.9 How to Read This Dissertation

This dissertation is a cumulative thesis. Chapters 2–8 correspond to peer-reviewed publications that are included verbatim to preserve their original technical context and reproducibility. Readers primarily interested in the overarching narrative can focus on Chapter 1 (motivation, thesis statement, research questions, and contributions) and Chapter 9 (synthesis and outlook), and consult Chapters 2–8 selectively for technical details and empirical evidence.

A multi-layer measurement perspective. This dissertation adopts a layered view of the mobile communication ecosystem. Across these layers, the central methodological goal is to enable *independent, controlled, and scalable* measurements without requiring cooperation from operators or platform providers. This progression reflects a shift in where control and observability reside: from decentralized, operator-bound infrastructure at the radio access layer, to operator-managed services exposed via the public Internet (e.g., VoWiFi), and finally to globally centralized OTT messaging platforms.

Chapter dependencies and reading order. Chapters 2 and 3 introduce and operationalize the dissertation's core measurement capability at the radio layer, enabling repeatable experiments across countries, providers, and roaming conditions. Chapters 5 and 6 study operator-controlled VoWiFi services using Internet-based measurement vectors that complement radio-layer experiments. Chapters 7 and 8 extend the same measurement mindset to OTT messaging applications, exposing ecosystem-wide attack surfaces and privacy risks. Chapter 4 deliberately departs from cellular measurements. Using a community-driven measurement platform, it shows how Internet centralization can be exposed and how such centralization affects anonymity in Tor. This chapter provides an external validation of the dissertation's main theme: that accessible, large-scale measurements can uncover otherwise hidden weaknesses and risks.

Research questions and evidence. The research questions stated in Section 1.3 are addressed as follows. RQ1 is addressed by Chapters 2–3 (measurement platforms and scalability). RQ2 is addressed by Chapters 3–6 (roaming, billing, alternative

access technologies). RQ3 is addressed by Chapters 5–8 (cross-layer security/privacy properties and observability), with Chapter 4 providing complementary evidence of how democratized measurements can reveal centralization effects in a different ecosystem.

Artifacts and reproducibility. This dissertation follows an artifact-driven research approach: measurement studies are paired with released tooling and responsible disclosure where applicable. Section 1.8 lists the open-source projects and artifacts that support the presented experiments and facilitate independent validation, reuse, and extension by other researchers.

Scope. Although the chapters cover multiple layers and technologies, this dissertation does not aim to provide an exhaustive survey of mobile standards or deployments. Instead, it focuses on concrete measurement capabilities and empirical case studies that collectively support the thesis statement: that independent measurements enable systematic discovery and quantification of security, privacy, and policy failures across layers.

2 Relaying SIM Communication for Cellular Network Measurements

Publication Info

Title	MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research
Authors	Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, Adrian Dabrowski
Publication Status	This paper is included in the Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), pp. 3493–3510, ISBN: 978-1-939133-37-3, 2023. CORE2023 Ranking: A*.
Publication Page	`https://www.usenix.org/conference/usenixsecurity23/presentation/gegenhuber`
Code Artifacts	`https://github.com/sbaresearch/mobile-atlas`
arXiv	`https://arxiv.org/abs/2403.08507`
Reference	[GMWD23]

MOBILEATLAS: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research

Gabriel K. Gegenhuber
University of Vienna[*]

Wilfried Mayer
SBA Research

Edgar Weippl
University of Vienna

Adrian Dabrowski
CISPA Helmholtz Center for Information Security[†]

Abstract

Cellular networks are not merely data access networks to the Internet. Their distinct services and ability to form large complex compounds for roaming purposes make them an attractive research target in their own right. Their promise of providing a consistent service with comparable privacy and security across roaming partners falls apart at close inspection.

Thus, there is a need for controlled testbeds and measurement tools for cellular access networks doing justice to the technology's unique structure and global scope. Particularly, such measurements suffer from a combinatorial explosion of operators, mobile plans, and services. To cope with these challenges, we built a framework that geographically decouples the SIM from the cellular modem by selectively connecting both remotely. This allows testing any subscriber with any operator at any modem location within minutes without moving parts. The resulting GSM/UMTS/LTE measurement and testbed platform offers a controlled experimentation environment, which is scalable and cost-effective. The platform is extensible and fully open-sourced, allowing other researchers to contribute locations, SIM cards, and measurement scripts.

Using the above framework, our international experiments in commercial networks revealed exploitable inconsistencies in traffic metering, leading to multiple *phreaking* opportunities, i.e., fare-dodging. We also expose problematic IPv6 firewall configurations, hidden SIM card communication to the home network, and fingerprint dial progress tones to track victims across different roaming networks and countries with voice calls.

1 Introduction

Large-scale measurement platforms and distributed testbeds such as RIPE Atlas [36] and PlanetLab [34] contributed to the security and privacy research community in at least three ways: (i) they allow the measurement of network properties once or longitudinal from different vantage points, (ii) they allow to quickly measure the scale of a found or known problem, i.e., gauge the real-world impact, and (iii) they function as a testbed to rapidly develop and test potential security vulnerabilities on a large scale. Additionally, tools such as ZMAP [17] provide the ability to routinely make Internet-wide scans, which became a staple for papers on measurement and security alike.

These platforms and tools share that –in accordance with the layered network model– they are access technology agnostic. However, mobile networks, unlike any other access network, combine multiple access technologies and generations on top of each other. Furthermore, since Mobile Network Operators (MNOs) are only given a small geographical area (usually a country) to operate in, they form vast roaming alliances to allow devices (and their traffic) to traverse through multiple networks. This creates complex compound systems where entities of different operators handle different aspects of the user traffic.

To explore such systems, physically moving devices (or SIM cards) between countries for each case adds a staggering, prohibitive overhead. MONROE [45] approached this problem by duplicating each SIM card set at each location – effectively realizing the combinatorial explosion of *countries* $\times$ *mobile plans for each operator* – with tremendous costs hindering growth.

In this paper, we present a different approach. The key insight is that by geographically decoupling the SIM from the device, we can work with just one set of devices in the field and virtually connect them to one set of SIM cards – without multiplying our SIM arsenal for each location. To this end, we tunnel the SIM card protocol from one of the SIM cards to any of our remote-controlled devices in different countries and territories and have them connect selectively to specific operators as if the subscriber SIMs were actually there.

This enables us to test security and privacy-relevant aspects of different operators and combinations of operators as well as perform large-scale international studies and investigations.

We implemented the method in MOBILEATLAS, based on

[*]Supported by the UniVie Doctoral School Computer Science DoCS.
[†]Partly as postdoc at University of California, Irvine.

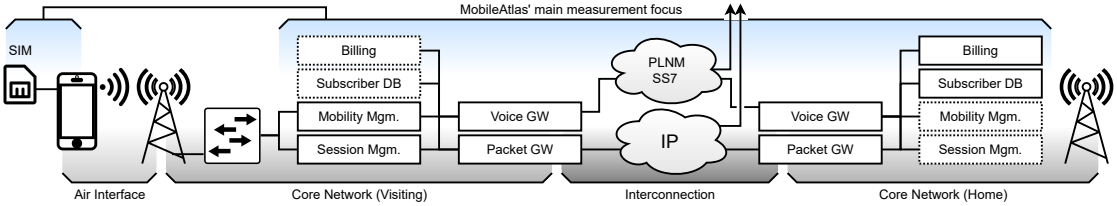


Figure 1: Simplified technology neutral structure of a cellular 3GPP network with roaming, and MOBILEATLAS's primary focus

low-cost off-the-shelf hardware to facilitate easy deployment, deployed it in ten countries, and added an orchestration & management interface to encourage other researchers to join. As an open and scalable tool, we hope it will help researchers, but it could also be used by operators or regulatory bodies to validate local or roaming network properties. MOBILEATLAS is versatile enough to handle data, voice, SMS, and USSD[1] services as well as capture hidden SIM communication.

We demonstrate the capabilities and flexibility of this framework with five security- and privacy-related use cases in three different domains: (i) high-scalable, isolated network measurements in domestic & roaming scenarios, (ii) voice connections with and between roaming partners, and (iii) low-level SIM communication analysis. The results demonstrate how to leak a traveling callee's country and operator as well as transmit free data by bypassing traffic accounting.

The remainder of this paper is structured as follows: Section 2 and 3 introduces the background and related work. In Section 4, we analyze the problems and requirements of measurement systems. In Section 5, we describe the general system design, followed by the concrete implementation in Section 6. We present current use cases in Section 7, discuss our work in Section 8, and conclude in Section 9.

2 Background

2.1 Cellular Networks and Roaming

For the sake of space, we describe a 3GPP-style cellular network (i.e., GSM, UMTS, LTE, 5G) in a simplified structure and with technology neutral terminology (Figure 1).

The *SIM card* (physical or as embedded SIM) is issued by the home operator and holds secret cryptographic keys. Any device (i.e., UE, User Equipment) successfully passing authentication based on those keys is recognized as a specific subscriber. The SIM itself does not encrypt the data over the air interface (RAN, radio access network), it only provides the session keys.

An MNO operates a large number of base stations[2] that are connected via the core network to centralized or decentralized services. A subscriber database and a billing system manage the customers at the home operator. A session management system allocates network services of a given quality for a given subscriber, e.g., Internet access at a set speed over a high-latency tunnel, a voice channel on a low-latency low-bandwith tunnel/channel, a phone number, etc. A mobility management unit keeps track of the UE's position within the network and can page (call out) the UE if necessary. It makes sure the session tunnels are rerouted accordingly when the UE changes location.

2.1.1 Services

Today, cellular networks offer three main services to customers: data, voice, and SMS (i.e., text).

Data services are provided via a data tunnel (*bearer* since LTE) that follows the subscriber and (typically) terminates the traffic via a public-Internet-facing gateway.

In GSM and UMTS, voice is a different network service than data. It uses a different type of tunnels and channels (*circuit-switched*), priorities, and endpoints. On LTE and 5G, voice (Voice over LTE, VoLTE) is IP traffic and transported via the same type of tunnels as Internet data, although with a different priority. Those tunnels terminate at an IMS-Server (IP Multimedia Subsystem) that can route calls to the public phone network. Networks often operate two separate voice infrastructures, one for VoLTE and one for legacy circuit-switched calls. The network can use Circuit-Switched Fall Back (CSFB) to move 5G/LTE voice calls to UMTS or GSM.

2.1.2 Roaming

Frequencies for cellular networks are typically given exclusively to a selected number of operators in each country or territory. To provide services outside of the physical coverage area, an MNO arranges a roaming agreement with one or more MNOs in another area.

In the roaming case, a UE uses a mixture of services from the two involved operators: the *visiting operator* (where the subscriber is physically located) and the *home operator* (where the subscriber got their SIM from). From the subscriber's view, both operators fuse to a single system that strives to provide the same services as the home operator in their home territory.

[1] Unstructured Supplementary Service Data codes are user dialable messages to control provider services, e.g., ##21# to disable all call forwarding.

[2] *Base Transceiver Station BTS* in GSM, *NodeB* in UMTS, *E-UTRAN Node B* or short *eNodeB* in LTE, *gNodeB* in 5G: The reader may excuse that we will use a high-level technology neutral terminology to focus on the structure and not on a particular implementation.

Data Internet data can be roamed in two ways: (i) *Local Breakout* (LBO) terminates the traffic at the visiting operator's Internet gateway. A customer will therefore receive an IP address from their visiting country which may effect website localization and geoblocked services. Billing records are collected by the visiting operator. (ii) With *Home Routing* (HR), the visiting operator hands over all subscriber data traffic to the home operator (usually via an Internet interconnection), where it is billed and routed to the Internet with a home operator's IP address.

Legacy Voice and LTE's CSFB The visiting operator assigns a temporary local phone number, i.e., from the visited country. Voice calls take the same infrastructure as the visiting operator's own subscribers, with the exception that the outgoing caller ID is matched to the original visiting customer's number. In this regard, it is similar to the local breakout data roaming case. Incoming calls are handled by the home operator, which redirects the call to the temporary foreign phone number by opening another phone call.

VoLTE roaming VoLTE roaming can be handled as LBO, HR, or a combination of those that involves both IMS. However, because the complexity of the underlying SIP protocol and the large number of settings, e.g., for bitrate and codec, the standardization and interoperability is still in its infancy. Many operators use *Circuit-Switched Fall Back* (CSFB) for roaming, and some (such as AT&T in the U.S.) stopped providing voice roaming altogether.

2.2 Types of Measurements

For measurements in the mobile network sector, we differentiate between (i) passive and (ii) active measurements. Passive measurements often require elevated network access and are therefore collected with the help of network operators, e.g., Mobile Network Operators (MNOs) or Internet Service Providers (ISPs). This data is seldom publicly available.

In contrast, many active measurements work with a customer-grade access and are a vital path for independent and open research. The two most common practices for active measurements include (i) in-situ or in-vivo measurements and (ii) exclusive measurement setups.

In-situ- ("at location") and *in-vivo* ("within the system") measurements often run on a (volunteer's) phone that is also used for other tasks. This method increases coverage by lowering the financial burden of participating in a given study and thus adds more measurement units. While appropriate for, e.g., user studies, this method might impact the accuracy of technical measurements since the non-exclusivity blurs the distinction between the signal (that is to be measured) and the noise (created by user activity or background processes). The user's mobility also interferes with the common goal of steady (consistent and repeatable) measurement environments.

Exclusive measurement setups require a separate test unit in a controlled environment. While some external factors (such as the network's utilization or radio noise) are commonly outside the researchers' sphere of influence, other factors can be controlled; this includes the distance to a cell tower, the direction of the antenna, and background data usage.

3 Related Work

Measurements for Security Measurement of the impact of a found vulnerability is part of many network-related security papers. ZMAP [17] and other similar tools became a staple for evaluation of Internet phenomena and especially security vulnerabilities.

Measurement of and in cellular networks An example for a passive measurement is Lutu et al.'s [27] insights into a large IP exchange provider (IPX-P). They analyzed passively collected data and gathered operational insights regarding, among others, global data roaming.

In 2019, Li et al. [26] developed an active *in-vivo* mobile application called *Wehe* to gather more than one million crowdsourced measurements. They identified traffic differentiation at 30 ISPs. MOBILEATLAS can, in addition, use further mobile network capabilities to conduct measurements, e.g., SMS, calls, and USSD. It provides a controlled measurement environment with no background noise from other apps. Thus, our technique is well suited for longitudinal observations under stable conditions and can measure fine-grained differentiations.

In 2015, Alay et al. [2] presented the MONROE measurement platform and later refined the architecture and node design [1]. MONROE is an open, European-scale, and flexible measurement platform for mobile broadband networks. However, each node is physically connected to the measured SIM card, leading to maintenance and hardware overhead. Similar to MOBILEATLAS, MONROE is an exclusive measurement platform that also supports roaming. However, MOBILEATLAS improves scalability by reusing one single SIM card at different nodes. We contrast those architectures in Figures 2 and 3.

Distributed Measurement Networks Landline and fixed Internet measurement platforms matured over the decades and are a staple of empirical Internet research. For instance, the RIPE Network Coordination Centre introduced the RIPE Atlas measurement platform [36,37] in 2010, where a large number of simple measurement probes allows a distributed execution of a small set of network commands, e.g., traceroute. Users can create self-defined measurements that get accounted for through platform-specific credits. RIPE Atlas has continuously improved [6] over the last decade and consists of approximately 12,000 measurement *probes* and 800 *anchors* (February 2023). It is utilized by numerous research projects, covering issues such as network routing information [5], various protocol measurements [22] and censorship detection [4]. However, RIPE Atlas is focused on fixed Internet connections and only allows a small set of possible measurement commands – MOBILEATLAS replicates

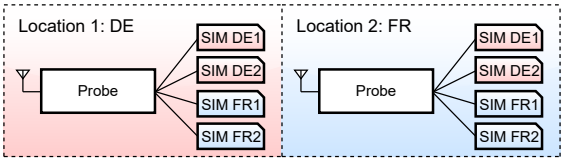


Figure 2: Traditional approach with poor scalability: Every new location needs a new set of all SIMs and mobile plans.

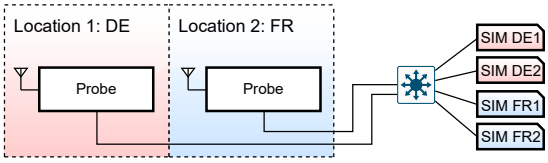


Figure 3: Decoupling the station from the SIM via tunneling requires only one set of SIMs.

these (and other) capabilities in the cellular world.

Kakhki et al. [23, 24] studied traffic differentiation in mobile networks, using an app-based record-replay approach to identify traffic differentiation based on mobile application traffic. MOBILEATLAS could improve these results because our probes are fully controllable and the provisioning of different SIM cards to test geographical differentiation is possible.

Voice Interconnection Security and Fraud In 2016, Sahin and Francillon [39, 41] measured so-called *over-the-top* (OTT) bypass fraud, i.e., telephone calls that get fraudulently redirected to OTT services, bypassing terminating operators. The authors detected OTT fraud using Test Call Generation (TCG) networks to initiate tens of thousands of phone calls. This approach is helpful for incoming bypasses, whereas outgoing routes would require an international network. In addition to their TCGs, Sahin et al. had to geographically move phones with their SIM cards between different countries and operators. MOBILEATLAS supports voice calls and can therefore automate OTT and other fraud [40] detection.

Decoupling and Redirecting cellular subscribers Samsung Call & Message Continuity (CMC) [42] is a way to seemingly transfer calls and chats between devices on the fly. However, CMC does not change the transfer point with the cellular network, it just acts as a relay to forward data. SIM Banks or SIM Pools are mainly used in OTT fraud [39] to distribute and shuffle subscriber SIMs between different exit points domestically via SIM tunneling to circumvent roaming. In contrast, MOBILEATLAS moves SIMs internationally, to actively trigger roaming. SIM Banks are occasionally available from AliExpress and other sites but use an undocumented transmission protocol.

Air Interface Attacks Cellular network research is often narrowly considered as research of the air interface or radio access technology. Certainly, *Fake Base Stations* (FBS) or *IMSI Catchers* [3, 13, 29, 32, 49] are one of the most prominent ones. FBS are a major building block for a large number of further cellular network exploits [38]. MOBILEATLAS can partly work as Fake Base Station detection, but would need a much denser distribution pattern to cover single base stations (Section 6.3) and is therefore not a goal at this stage (Figure 1).

4 Analysis

As briefly touched above, current measurement platforms suffer from one or more of the following problems or challenges:

(P1) *Access technology obliviousness.* Some measurement systems are ignorant regarding the underlying properties and features of the access network. In many cases, the transparent layering of a network stack from the physical layer up to the application layer is a desirable property, e.g., receiving e-mail should work regardless of Wi-Fi, 3G, LTE, or roaming. However, for measurements, the layering may hide crucial information and defining characteristics.

Without full access to domain-specific parameters (e.g., Cell- & Location IDs) and features (e.g., voice calls), a measurement system may lack sufficient insights to successfully audit complex mechanisms (e.g., roamed billing) within cellular networks.

(P2) *Resources scale quadratically by coverage.* If a cellular device and a mobile plan (mediated through the SIM card) are treated as a single unit, they are location-bound. Devices and mobile plans (i.e., SIM cards) need to be duplicated at every location to allow for location-independent measurements (Figure 2), e.g., for international roaming measurements. Alternatively, SIM cards can be shipped around and switched manually.

The first approach scales in $O(|P| \times |S|)$, where $|P|$ is the number of probes, and $|S|$ is the number of SIMs. The second approach binds human resources at every location and limits the number of possible time-shared measurements. Additionally, SIM orchestration (i.e., physically provisioning SIMs to other locations) and provider mobile plans (e.g., one contract per location) are the highest cost- and maintenance drivers in a measurement system [2].

Thus, a measurement system without flexible switching between SIM cards at target locations does not meet the requirements for large-scale cellular measurements.

(P3) *Background noise.* Some measurement frameworks lack exclusivity regarding the network interface. This is not a problem in many qualitative measurement settings (e.g., IPv6 capabilities); however, exclusivity is required for most quantitative measurements (e.g., how IPv6 is metered). Especially metering is often unavailable in real-time and requires careful coordination with other tests.

Depending on the target, even a tiny amount of background traffic (e.g., DNS lookups) can influence the results and call for sterile access to the network interface

and other cellular services.

4.1 Goals and Requirements

Based on these problems, we identified the following requirements for a new system:

R1 *Scalability.* A proper, scalable system must deal with a multi-dimensional combination space of different mobile providers, access technologies, carrier products, tariffs – all of these in many different countries and with every roaming partner. Thus, manually managing these components is tedious and costly.

The system should achieve complete coverage of all combinations without manual work, duplication of hardware setups, or multiple identical mobile plans. Security measurements need a flexible and easy-to-deploy base that enables security investigations of edge-cases, where unhabitual processes or actors are involved (e.g., roaming). Given the profound impact on the completeness and validity of results, we identify scalability as the essential requirement.

R2 *Control of SIM communication.* SIM cards are a significant component in the cellular network stack. These often underestimated microcontrollers can receive over-the-air (OTA) updates from the operator, load custom SIM applets, and proactively send commands to the modem they are connected with. Moreover, they contain the key material that is the baseline to generate session keys for radio and also VoLTE connections. Smartphone operating systems and their network stacks hide low-level SIM communication from the application layer. Without the ability to observe and possibly change communication, any measurement system lacks fundamental insights into the access network.

To gain these insights and to enable security- and privacy-related auditing of the opaque communication stream between SIM card and modem, we identify control of SIM communication as one of our design goals.

R3 *Utilization of the full feature spectrum.* A cell phone connection is much more than just an Internet outlet. The selection of specific operators, cells, and access technology shapes the properties of the data channel. Additionally, voice, SMS, and USSD codes each form a complete ecosystem of their own. Ignoring these features deprives the researcher of substantial tools to gain in-depth knowledge on the measured networks and may hide important aspects that may be valuable from a security and privacy perspective.

Therefore, a new measurement system should provide full access to cellular-specific parameters and features.

R4 *Isolated & Controlled Environment.* Many operator-related systems (e.g., billing, provisioning, QoE, and QoS) are opaque and thus difficult to analyze. A clean, noise-free (or -reduced) environment is paramount for accurate measurements and conclusions. Only application- and traffic-exclusivity with tight control over the network stack and possible background data will provide the required data quality. Additionally, only a geographically stationary system can provide the environment to gather stable and comparable longitudinal data.

R5 *Breadth.* Networks and all their components are very complex. This also holds true for the interplay between different access technologies as well as networks, both of which can have unintended or unanticipated side effects. The handling of such corner cases is often not covered by standard procedures (e.g., billing), and, in some cases, not even the network operator might be aware of these effects. Root causes might be ad-hoc decisions during network design and setup or the fact that the operator lacks means to test traffic flow through all combinations of foreign networks.

To enable security inspection of complex corner cases, a newly developed system should provide a broader overview than a single network operator has.

R6 *Low-cost, self-managing, open-design.* The geographical and functional breadth can only be achieved through a simple deployment by an (easily recruitable) community of station caretakers. Neither they nor the overall organizers should need to invest a lot of resources to enhance, grow, or maintain the system. Furthermore, low-cost design and off-the-shelf hardware reduce the entry barrier for further research and may lower the barrier for testing known security risks in advance.

5 System Design

In this section, we first describe the key design elements and then provide a system overview.

5.1 Geographical Decoupling of SIM Cards and Modems

A mobile device and its SIM are often viewed as an indivisible tandem, assuming that one cannot work without the other. However, testing mobile connections through SIMs issued by different operators in different networks (e.g., roaming) includes (a) physically moving SIMs or (b) physically moving SIMs and their mobile devices, or (c) replicating each SIM and the measurement setup in every territory (cf. Figure 2) — thus, scaling poorly based on manual or financial effort.

There are numerous ways to geographically decouple the identity (i.e., SIM) from its actual usage in a particular network. At higher layers, the mobile device's radio stack communicates via a low-bandwidth protocol with the SIM. Moving down through the stack, it produces radio messages in the lower layers until it modulates high-bandwidth, low-latency digital radio samples. At the lower layers, it transforms these

samples into an analog radio frequency (RF) signal and amplifies it before radiating it via an antenna (and vice versa for receiving). Thus, a geographical displacement of the lower layers entails a much higher engineering overhead than at higher layers. While software-defined radio (SDR) would also provide more freedom on the radio layer (i.e., custom or non-standard messages) and open the system to air interface testing, their use is generally not permitted on real-world networks. Even though strong GSM, LTE, and 5G software implementations for SDRs exist, they lack the radio-regulatory permissions to operate. Therefore, using a globally-licensed off-the-shelf modem and decoupling on the low-bandwidth communication between the SIM and the modem provides the best trade-off for rapid global deployment.

Based on these considerations, we opted for tunneling the SIM card's ISO7816 T=0 protocol [21] through the Internet by electrically connecting the modem's SIM socket with GPIO pins of our measurement platform, and recreated the protocol via software. Modem and SIM exchange Application Protocol Data Units (APDUs) via the T=0 connection.

Connecting any SIM with any radio module (i.e., modem) – regardless of geography – facilitates measurements across a large number of SIMs and radio networks. SIMs can be either pooled centrally or hosted de-centrally, using an infrastructure to add and remove SIMs dynamically.

This method fulfills the scalability (R1), low-level control (R2), and breadth (R5) requirements. Using GPIO pins directly and implementing T=0 in software reduced costs and complexity (R6). Furthermore, in our design, we do not have to support all protocol speeds and electrical variants on both ends of the tunnel as they are electrically independent.

eSIM eSIM is a complementary technology that overlaps slightly with to our proposed technique but can not replace it for two reasons. First, eSIMs are not available from all operators, and where available, they are not always obtainable for all (subscription) plans. Second, eSIM deployment is not always device-independent and may require different distribution methods. For example, some eSIMs are only available via specific apps – some of which are not (virtual) MNOs but merely resellers. Others require network support. Some operators only make them available for specific device types (e.g., Apple smartwatches). Furthermore, the eSIM ecosystem is heavily fenced off with certificates. Thus, our solution is the most universal, as we can tunnel physical SIMs as well as eSIMs from supported devices via Bluetooth (remote) SIM access profile (rSAP), as presented in Section B.1.

5.2 Process and Network Isolation

As we intend to reduce network noise (R4), we can not use ordinary smartphones - as the operating system would induce too much background traffic noise. However, even in our dedicated system, we needed to split the network stacks. The SIM tunnel and control connections need to be separated from the measurement system. Docker and other container systems use Linux network namespaces [16] to create shielded networking enclaves. Such a network namespace has a separate device configuration and separate routing tables. In these setups, only selected processes are admitted to a namespace that uses the cellular connection as their default gateway, thus, shielding the measurement script from any other system activity.

5.3 Metering Measurements

With most network operators, billing is not a real-time service. Charge Data Records (CDRs) are collected decentrally and accumulated asynchronously in the customer records. Delays span from minutes to hours, depending on the operator. While some operators account for every single byte, others collate (and usually round up) the charges. This may also vary in roaming situations since additional parties are involved in the accounting process.

For customers (and researchers), there is no standardized way to query data plan usage. Most operators offer a custom app or website. In addition, some providers still operate USSD and SMS methods as used in legacy SIM Toolkit applications. This is captured by the (R3) feature utilization requirement.

In order to speed up the measurement of different traffic classes, we can utilize a binary power encoding scheme by varying the data amount, i.e., $amount = 2^{class\,nr}$. For example, one MB of class 1 is produced, two MB of class 2, four MB of class 3, and so on. Hours later, when the CDRs are available, it is possible to separate these traffic classes.

As metering measurements are one of the possible use cases of a new cellular measurement system, built-in credit-checking support is advisable.

5.4 Overview

Our measurement framework is designed as follows.

Our fundamental architecture consists of five components, as illustrated in Figure 4:

(i) Probes that execute experiments and measurements on the cellular network via a modem. Furthermore, the probe electrically emulates the SIM card for the modem.

(ii) SIM providers that connect to multiple genuine operator SIM cards (i.e., concentrate them or allow users to *temporarily donate* a SIM with special properties), allowing to connect them to any modem via a virtual circuit.

(iii) The actual SIM cards, which contain the secret key from the operators.

(iv) A separate VPN-protected Internet connection for management purposes, allowing deployments behind NATs and firewalls.

(v) A management interface to monitor and administrate the components and orchestrate tests. It remotely updates and introduces new measurement scripts. Additionally, it collects the status (dashboard) and allows interaction with probes (e.g., schedule tasks, connect to selected SIMs, reboot probes).

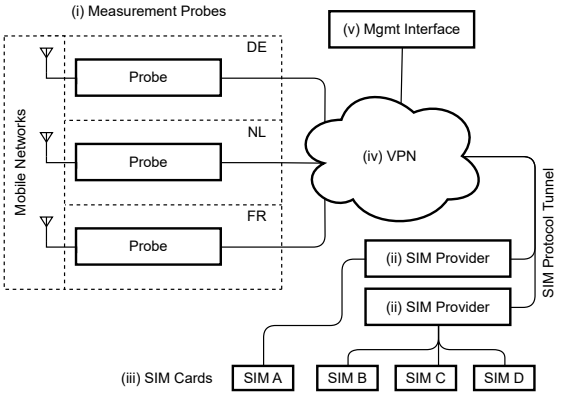


Figure 4: High-level overview of MOBILEATLAS components: SIMs connected to our SIM Providers can be virtually connected to any measurement probe at the target location.

5.5 Ethical Considerations

We consider ethical aspects for three roles: The designer of the method, the operator of the testbed and measurement network, and the researcher that executes tests and measurements. Not all of the following apply to all of the roles.

Technical and radio regulatory. Radio regulatory compliance and certification is paramount if stations are to be operated by third parties or lay people. Therefore we can not use SDRs, and we only employ globally-certified modems and refrain from making any (hardware) modifications to them. We only capture and transmit signals (e.g., APDUs) via the designated external interfaces.

Live network influence. We conducted our measurements in real-world, live mobile networks; our metrics reflect normal user behavior (data access, SMS functions, and telephone calls). Ideally, our activities should not affect the network. However, due to SIM tunneling, our SIM cards change the "country" in an irregular and fast fashion. This might spark confusion among the operator systems or trigger fraud control alerts. In order to exercise caution, we manually impose a waiting time between country switches.

Guest network access. Our probes are connected to our host's LANs. We hardened the operating system configuration and executed measurements exclusively in their own namespaces. To further facilitate secure deployment, we ensured that the Internet-facing control traffic passes to the control server only via our VPN.

SIM registration. Regarding the use of SIM cards, we must consider that SIM registration laws in certain countries may require personalization or registration of SIM cards [46, 47]. Where required, we registered the SIM cards to ourselves and exclusively used them for measurements.

Provider's terms of service. Probing, testing, and speed measurements could violate an operator's terms and conditions. We took precautions not to harm the networks and did not use our technology to exploit networks economically. Furthermore, we share the implementation and results, as this is the only way – in our opinion – to ascertain robust public infrastructure knowledge. While part of the project shares similarities with SIM banks, we do not conduct any of the other activities of the SIM bank fraud scheme, such as terminating calls on local SIM cards, faking International Mobile Equipment Identifiers (IMEI), or enriching ourselves.

Vulnerability exposure. Measurements aim to improve understanding of cellular networks and their interplay. Whenever we identify serious problems leading to harmful behavior, we contact the network operators directly and report our findings responsibly prior to publication.

Economic losses. After careful evaluation we are convinced that there is no other way of testing real-world billing systems. Thus, we always make sure not to enrich ourselves by not oversizing the generated traffic and by letting an equal or greater amount of our monthly traffic allowance expire at the end of the month (as if we were billed for the traffic).

6 Implementation

This section is a brief summary of the implementation details and choices as well es some performance measurements described in the Appendix.

6.1 Probe

Probe Hardware The hardware went through two major iterations for cost, stability, and supported features – as it was revised after the deployment of the first batch. Since a large-scale deployment requires a grassroots-like approach and costs multiply by the number of probes, we eventually settled for a low-cost *Raspberry Pi 4* and a *Quectel EG25-G* modem via a mini-PCIe adapter. The SIM is controlled via an UART on GPIO pins, and the Internet connection is realized via the Ethernet port.

Probe Software Architecture The architecture consists of a Python-written dynamically deployed test-suite that runs in a Linux (network namespace) container for isolation, similar to Docker. Only the containers traffic is routed through the modem and recorded to a PCAP file.

6.2 SIM Interface and tunneling

SIM Interface Multiple generations of the synchronous SIM card interfaces in various voltages (1.8 V, 3 V, and 5V) with multiple speed (clock speeds and clock divider) settings exist. Our approach is significantly simplified in contrast to other SIM-interfacing tools (e.g., SIM trace [30]) in that both ends of the tunnel are electrically separated. We can thus implement and negotiate speed and voltage independently

through the *Answer on Reset* (ATR) handshake and the *Protocol and Parameter Selection* (PPS) command. Since the on-board UART only supports asynchronous transfers, we approximate the speed characteristics. Generous divider settings, tolerances, and the ability to start communication at any clock cycle work in favor of an extreme minimalistic hardware interface. For the open-collector data bus, the modem already provides a pull-up, and the UART driver pin is shielded via a Shottky diode (Figure 9).

Additionally, the (simulated) SIM has the ability to signal *Waiting Time eXtensions* (WTX) towards the modem to mask the effects of network latency.

SIM Provider The dynamic SIM tunnel eventually has to terminate at a real (e)SIM. For a medium-sized operation, cheap USB card readers (either via a serial or PC/SC interface) are the most economic option. Dedicated hardware (SIM Banks), sometimes used for OTT-Fraud [39], serves more SIM cards but lacks standardized interfaces. While USB theoretically supports up to 127 devices, in our experience we seldom observed a reliable operation beyond 32 devices even on quality hardware.

eSIM and Bluetooth SIM Access Profile We have also implemented a Bluetooth-based SIM provider via the *remote SIM Access Profile* (rSAP), such as used by many cars. This offers a route to attach *embedded SIM* (eSIM) into the MOBILEATLAS system: Off-the-shelf non-rooted Android phones (in our example a Google Pixel 3a) are happy to share an eSIM via rSAP when it is configured as primary SIM.

6.3 Deployment

We currently have probes in eight European countries (Austria, Belgium, Croatia, Finland, Germany, Romania, Slovenia, and Slowakia), and two North American countries (Canada and the United States). These ten countries account for more than 510 million people.

We argue that for core network, subscriber billing, and roaming tests, one probe with good radio connectivity per country is sufficient. These are operator-wide properties (Figure 1), not dependent on individual base stations. Some low-level or fine-grained RAN testing might need multiple stations, such as testing handovers, radio propagation under different geographical conditions, domestic roaming (as used in China until 2018), or special radio installations (e.g., cruise ships).

We included SIM cards from bare-metal (MNO) and virtual network operators (MVNO) in our Proof-of-concept setup. eSIMs are supported by via Bluetooth's (remote) SIM Access Profile (SAP).

6.4 Benchmarks

A greater variety of benchmarks can be found in the Appendix C. The tunnel latency stands out in its importance for deployment. In a local setting, APDU round-trips are mostly negligible. However, it was never designed for a remote-tunnel setting where latency and round-trip numbers multiply.

6.4.1 SIM tunneling latency analysis

Tunneling inevitably increases latency at the SIM interface and subsequently in the network (e.g., for authentication and session key generation round-trips). As described above, we used WTX to avoid timeouts from the modem. For this section, we tested the tunnel's robustness on the radio interface to the network operators by artificially inserting long delays.

We tested three different operators with the Quectel EG25-G and added an artificial delay of 1,000 ms before each APDU response. This matches high-latency network conditions.

In all cases, the modem successfully attached to the cellular network. We verified the functionality by sending and receiving SMS and initiating a voice call.

Once the modem is attached, APDU traffic is minimal. We could not detect any network performance loss (e.g., data, SMS, call) due to the SIM tunnel. In general, handovers between cell towers do not cause re-authentications. However, simultaneous radio access technology changes (e.g,. LTE to UMTS) can trigger re-authentications and cause a slight delay.

Operators are aware of vastly diverging authentication round-trip times [12] between devices. However, in theory, our extreme round-trip times could stand out.

7 Showcases

This section presents five security and privacy focused use cases demonstrating diverse capabilities of MOBILEATLAS. The first group of use cases demonstrates a few of the vast possibilities for roaming-related measurements and studies. The second group showcases how to test a hypothesis, measure a network property, or collect provider configurations on a large scale domestically and internationally.

7.1 Roaming and Price Differentiation

Previous work on differentiation [8, 15, 23, 26, 51] analyzed traffic as well as the implications of international data roaming [28]. However, the question arises if zero-rated traffic is handled differently in roaming. Understanding roaming billing is essential as it is either much more expensive than domestic traffic or subject to stricter limits (as in the EU's free-roaming agreement [18]). We hypothesized that the roamed traffic classification is identical due to home routing; however, unexpected differences in metering exist. In this use case, we focus on the measurement of fine-grained metering differentiation of roamed DNS traffic, which would inform an attacker how and when to use DNS tunnels to hide traffic from billing.

Methodology. We tested the provider's default DNS configured via DHCP (internal), as well as an external self-hosted DNS resolver. First, the current data quota is checked automatically (see Section 5.3). Second, we repeatedly sent DNS queries (for top domains according to Tranco [25] list V78N) to the target DNS server until the necessary traffic size (e.g., one MB) was reached. Third, we waited for the billing systems to update and recheck the available quota. This process

was repeated for each SIM card, each tested country, and each tested DNS server. Finally, we analyzed the metering differences between domestic and roaming data usage.

Results. Table 1 presents the results as of March 2022. External DNS traffic was always billed by all providers in both scenarios (i.e., domestic and roaming). Nine providers did not bill internal DNS traffic regardless of roaming, whereas three others did. Interestingly, two providers (P-RO-1, P-SI-1) behaved differently under roaming conditions: Internal DNS traffic is not billed in the home network but is subtracted from the data quota when abroad. Twelve providers host their customer-facing DNS servers (that are served via DHCP) in public IP ranges; the other two have private IPs (P-AT-1, P-SK-3). Furthermore, we noticed that the DNS queries took noticeably longer during roaming due to increased round-trip times within home-routed connections (cf. Section 7.3).

These results demonstrate that roaming differentiation in billing exists and that MOBILEATLAS can be utilized for large-scale and fine-grained analysis.

7.2 Zero-Rating and Free-Riding

Zero-Rating specific applications (e.g., Facebook, Instagram, Snapchat) within dedicated data packages or tariffs is a common practice for many operators to reach specific market demographics. To correctly exempt traffic of those applications from a customer's data quota, all incurring data packets need to be classified by the billing system. For maintenance reasons, many providers do not use IP-based whitelisting because those can change on short notice in cloud-hosted environments. In early experiments, we discovered that many operators revert to client-supplied HTTP's hostname and TLS's SNI headers instead. This opens the door for *phreaking* or *free riding*, where a user can disguise traffic so it is misclassified by the billing system, allowing them to dodge fees.

Table 1: DNS billing for 14 providers from five countries. P-RO-1 and P-SI-1 differ in roaming billing behavior.

		Billing	
Provider	DNS IP	Domestic	Roaming[†]
P-AT-1	private	◐	◐
P-AT-2	public	◐	◐
P-AT-3	public	◐	◐
P-HR-1	public	◐	◐
P-HR-2	public	◐	◐
P-HR-3	public	◐	◐
P-RO-1	public	◐	●
P-RO-2	public	●	●
P-RO-3	public	◐	◐
P-SK-1	public	●	●
P-SK-2	public	◐	◐
P-SK-3	private	●	●
P-SI-1	public	◐	●
P-SI-2	public	◐	◐

● Internal & external DNS traffic billed
◐ Only external DNS traffic billed
○ DNS traffic not billed
discrepancy
[†] Measured with automatic network selection in two different EU countries

Methodology. We chose Snapchat as one of the most popular zero-rated services for our experiment. We tested all operators for which we could obtain a matching subscription plan or data package (see Table 1). First, we validated that the zero-rating package is active by sending web requests to a web endpoint that is used within our target application (i.e., app.snapchat.com).

Second, we evaluated whether the host header is used for classification and can be abused for free-riding. Instead of connecting to the legitimate web endpoint, we make the same request to an AWS server under our control that accepts web connections for arbitrary host and SNI names. We run each test for HTTP and HTTPS in a domestic and roamed usage scenario. After analyzing the billed data traffic, we know whether the legitimate application is zero-rated and which operators are vulnerable to free-riding by a spoofed hostname.

Results. The results for May 2022 are presented in Table 2. While we verified Snapchat zero-rating during domestic usage for all tested providers, one provider (P-HR-1) is still billing their roaming customers. Three providers turned out to be vulnerable to spoofed host headers. In practice, an attacker could abuse this type of classification by sending arbitrary data via a proxy server masquerading as the same host. Although the legitimate application communicates exclusively via HTTPS, two providers also zero-rate legacy HTTP traffic with a matching hostname header. The one provider in our sample that was not vulnerable to our free-riding attack via spoofed host and SNI names presumably uses a different metric (e.g., IP-ranges) for traffic classification.

7.3 Network- and Firewall-Configuration

MOBILEATLAS is a great tool for quickly determining IP level configurations for different operators or usage scenarios. This includes observation of used IP address ranges, as well as experiments that test for deployed firewall rules. Previous work has shown that security configurations can be different for IPv4 and IPv6 on dual-stack enabled servers and routers [11]. MOBILEATLAS supports IPv4, IPv6, and dual-stack connections, which makes it an excellent tool to investigate, whether this assumption holds for the cellular field.

For roaming, there are two common scenarios: local breakout (LBO) and home routing (HR) (see Section 2.1.2). When LBO is used, the customer gets an IP address from the roaming partner. With HR, all data is forwarded to the home network. While HR usually provides better security, it often adds additional latency because all packets are routed via the home network.

Methodology. Since MOBILEATLAS provides verbose insights into the parameters used for cellular network connections, we leveraged the data from our previous use cases to analyze these configurations. For all operators that provide us with a public IP address, we ran additional tests (e.g., ping and a low-volume nmap scan) against our own device to check whether incoming connections are filtered. (Note: No other

Table 2: With many providers, zero-rating can be abused for free-riding.

	Zero-Rating		Free-Riding	
Provider	Domestic	Roaming	HTTP(Host)	HTTPS(SNI)
P-AT-1	✓	✓	✓	✓
P-AT-2	✓	✓	×	×
P-HR-1	✓	×	✓	✓
P-HR-2	✓	✓	×	✓

customers were scanned.)

Results. All 14 tested providers use carrier-grade NAT for IPv4, and only two (P-AT-1, P-AT-2) offered IPv6 addresses (single- or dual-stack). The provided IP range for the carrier-grade NAT was 100.64.0.0/10 for three providers (P-SK-3, P-SI-1, P-SI-2), 100.64.0.0/10 + 10.0.0.0/8 for two providers (P-AT-2, P-HR-2), and just 10.0.0.0/8 for all other providers[3]. During our intra-EU roaming measurements, all 14 tested providers used home-routed data connections. Therefore the deployed IP configuration was identical in both domestic and roaming usage scenarios.

All used IPv4 addresses lie within non-public ranges; thus, the carrier-grade NAT acted like an implicit firewall against incoming connections.

For IPv6, however, both providers (P-AT-1, P-AT-2) use public IP ranges. P-AT-1 is blocking incoming connections, allowing the feature to be switched off in the customer profile. However, P-AT-2 is fully accessible from the outside and provides no such option. This might reveal application service ports to the public Internet and surprise many users and developers who rely on the implicit firewall [11, 48]. Several applications knowingly or accidentally expose TCP ports to the cell-network facing side, e.g., Netflix with port 9080, and the Android Debug Bridge (adb) on port 5555 (when *tcpip* mode is enabled). Apart from that, low-level attacks with traffic-usage inflation are possible due to the public IP address.

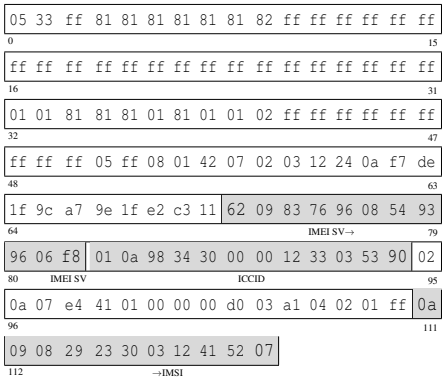
7.4 Low-level APDU analytics

In 2019, the Simjacker [43] report gained broad media attention as it found that many SIM cards were not protected against malicious over-the-air updates, given that many operators did not implement authentication. In 2021, McDaid [14] showed that this is still an issue in many countries.

Hidden SIM communication between the operator and the SIM can happen in two ways: It can be initiated by (i) the operator through a binary SMS message (e.g., over-the-air update) or (ii) proactively by an application on the SIM at any time. The latter can send binary SMS as well as instruct the phone to open TCP connections [9, 10]. Although playing a crucial role in the security of cellular networks, the operator's SIM card is a black box, demonstrating why more research is needed in this area.

[3]According to RFC 6598 [50], only 100.64.0.0/10 should be used for Carrier-Grade NAT to prevent IP address conflicts with private ranges.

Table 3: Example of a proactive binary SMS by P-AT-2. The opaque message contains encoded IMSI, ICCID and IMEI of the used devices.



MOBILEATLAS is an excellent research platform for this purpose because it provides complete insight into the hidden communication channel, especially if the SIM breaks the usual pattern of just answering the operator and starts proactively sending out requests.

Methodology. By default, MOBILEATLAS stores all APDU traffic in a database. Therefore we were able to investigate recordings from our previous use cases for this analysis.

Results. We observed two interesting cases of hidden proactive SIM communication. Two of the 14 SIM cards (P-AT-2, P-RO-1) sent SMS to the home operator without indicating this on the screen. Even though both were in an unknown binary format, we could clearly identify IMSI, ICCID, and IMEI in those messages. An example from P-AT-2 is shown in Table 3. The remaining unidentified data fields was static over all observed messages.

Charges. Even though those SMS were created outside the user's control by a device provided by the operator (i.e., the SIM), we were charged for them outside the EU's free-roaming area.

7.5 Country-grained Location Leakage via Call Progress Tones

In modern telephony networks, signaling and audio are two separate channels (i.e., out-of-band). However, it became common ("early media") to establish the audio channels already during dialing (as opposed to establishing them when the call is accepted) and let the closest switching center to the target create the ringback, busy tone, or voice announcements. This concept allows for additional audio messages or ringback music to be played to the caller. While recommendations for standard call progress signals exist by ETSI and other regulatory bodies, it is ultimately the operator to decide and the switching center on the callee side to create those signals with all the subtle differences that this might entail. Historically,

the US uses dual-frequency ringback tones, while European networks use a single-frequency tone. However, even within Europe, subtle differences in on-off times, the primary frequency and side lobes, as well as the amplitude exist.

On a technical level, circuit-switched voice roaming is realized as a local breakout (see Section 2.1.2) with a call forward to a temporary phone number in the visiting network, i.e., the visiting network creates the ringback or busy tones. In contrast, VoLTE can be broken out locally or home routed.

Previous work has already shown that sophisticated audio fingerprinting can be used to detect fraud (i.e., hijacked calls) [33, 35] and to determine call provenance [7]. We suspect it is possible in many cases to locate a cell phone subscriber's current network or country, based on simple metrics in the ringback tone of a single test call. Since MNOs are bound to a specific territory by the frequency regulatory bodies, one can infer the coarse location of a phone if the network operator is to be identified, especially in an area of the world with many small countries and permeable borders. This has various privacy implications.

For example, a thief can test if a possible victim is abroad. A competitor could test if a high-level official of another company or country is in a specific country (e.g., for a specific summit). Alternatively, a nosy supervisor could test if an employee misuses home office or sick leave. This technique also allows the identification of the actual operator of a domestically ported cellphone number (e.g., in a SIM-swap attack).

Methodology. Since the audio signal is a digitally compressed audio stream, we assume that (under normal conditions) it is passed unaltered and the subtle differences originate in the target network.

In the first step, we record three voice call setups from one base point to a roaming target. Then the target SIM was set into a different roaming network, and the process repeated. If inconclusive, we added additional test calls. We used SIMs from three different operators from one country to virtually *travel* to Austria, Canada, Croatia, Finland, Germany, Romania, Slovakia, Slovenia, and the United States via 19 distinct roaming partners.

For tone analysis, we cut off recordings after 20 seconds to ignore the following voicemail greeting. Then we identified metrics and features to distinguish the ringback tones. Where available, we repeated the setup for VoLTE connections.

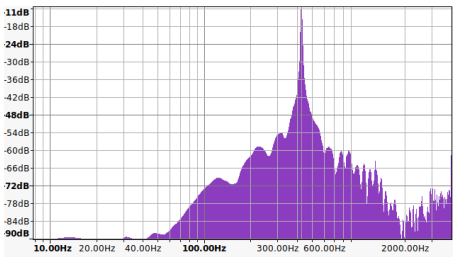
We analyzed the data for call setup times, ringback duty cycle (on/off times), ringback signal amplitude(s), and ringback frequency composition, but also manually heard them to find any other anomalies.

Results. *Early media* call progress tones heavily depend on the combination of source and (foreign) target network and the used access technology. Except for a few combinations, most circuit-switched voice calls (CSFB on LTE, 3G, and GSM) used target-network-generated tones with stable results. However, some networks seem to have two distinct settings, e.g., from a misconfigured load-balancing setup. VoLTE more often used origin-generated ringback tones. On average, VoLTE also had the shortest call setup times, and LTE's CSFB was the slowest (as it has to switch to 3G/2G first).

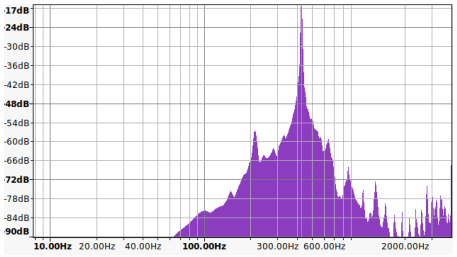
As for the ringback tone itself, we found several distinguishing features: (i) the base frequency, (ii) over-tone composition, (iii) on-off duty times, and (iv) the signal amplitude. The repeated recordings enforce our confidence in the stability of these features.

Call setup times turn out to be less stable and are likely influenced heavily by outside factors. Furthermore, we noticed transient audio channel artifacts with some operators when the call is transferred to voice mail.

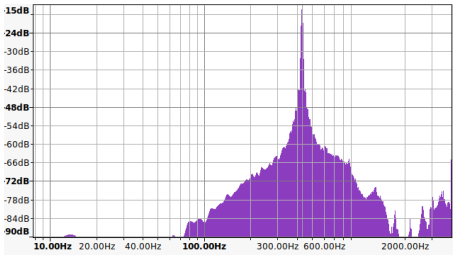
(i) **Base frequency.** In our European sample, the most prominent base frequencies are 420, 425, 426(.5), and 430 kHz. In contrast, all tested North American networks use a dual-frequency signal comprising 440 and 480 kHz, leading to a noticeable beat.



(a) DE-o2's ringback tone is a loud 426 kHz sine wave with -9.2 dB.



(b) DE-Telekom's ringback tone at 426 kHz is quieter with a -14.7 dB signal with a side peak at 212 kHz.



(c) A clean RO-Vodafone ringback tone at 430 kHz with -15.6 dB.

Figure 5: Comparison of ringback spectra.

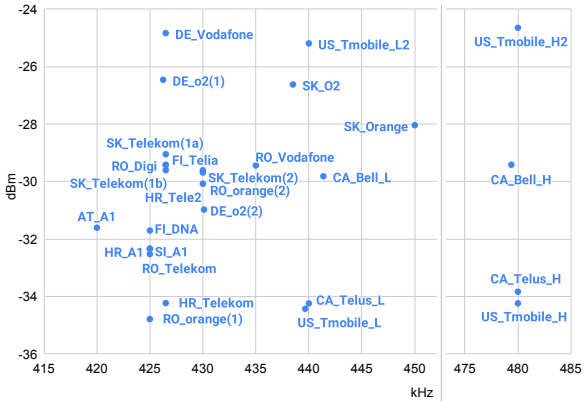


Figure 6: Fingerprinting ringback tones (without VoLTE).

(ii) **Overtones.** Some frequency diagrams show distinct overtones and side lobes. Figures 5a, 5b, and 5c present a few examples.

(iii) **Duty cycle.** Except for one network, all European providers use four seconds off-time and one second on-time. Only AT-A1 has an off-time of five seconds in their circuit-switch service but four seconds off-time with VoLTE. In contrast, all tested North American networks used a two seconds on-time and four seconds off-time.

(iv) **Amplitude.** The amplitude can differ between networks by 10 dB (Figure 6).

The identified features show high stability between experiments, with two exceptions. Some providers had two different amplitude levels (or frequency settings) and required us to make more test calls and recordings. This might be an artifact of multiple load-balanced switches being configured slightly differently. Besides our North American samples, three out of 92 European test calls experienced a non-repeatable U.S.-style 440/480 Mhz ringback tone. We might have observed an attempted interconnect bypass fraud attack, as described by Sahin et al. [39], or a case where the early media stream prematurely terminated at an earlier voice switch (e.g., transient connection errors).

For Figure 6, we followed an AT_A1 SIM card across 19 providers in nine countries – plotting only frequencies and signal strengths. Dual frequencies are suffixed with an _L and _H for low and high tones, respectively. In the case of AT_A1, all four markers change as soon as the SIM card leaves its home operator. A few cases of (near) collision exist between HR_A1 & SL_A1 as well as FI_Telia & RO_Digi. Clashes between operators with multiple frequencies (e.g., RO_Orange & SK_Telekom) are easily distinguished with repeated calls: The alternative frequencies do not clash. In contrast, the North American operators only differ in amplitudes.

VoLTE roaming is currently in its infancy, i.e., non-functional with many operators. They either use CSFB to switch to legacy voice or inform their roaming customers that voice is not available (e.g., AT&T in the U.S.). Domestic VoLTE calls showed slightly different signal compositions.

Charges. Between operators and other wholesale, phone calls are only charged after they were successfully established (e.g., the callee answered) and then typically on a per-second basis. Most operators pass the former modality to their customers but round up call durations to whole minutes. For calls long enough to be redirected to voice mail, there is a passive charge (for the incoming call) and an active charge for redirecting the call to the voice mail at the home operator. However, we also found that at least one operator charged for roaming voice calls to and from AT&T, even though AT&T does not support voice roaming.

8 Discussion and Future Work

Net neutrality, traffic discrimination, and Internet censorship are ongoing research topics. For example, some providers use questionable techniques to throttle video streaming [26]. Although the scientific community tries to address these issues, roaming scenarios are notoriously hard to measure.

In our first use case we discovered two traffic discrimination cases in roaming scenarios. Under the EU "roam like at home" doctrine [19], this traffic should be treated equally. In our second use case we extend the idea to hide traffic from the provider's accounting in *included services*. This opens possibilities to tunnel arbitrary traffic without metering.

Further research opportunities can be found in (i) measuring more roaming user discrimination, (ii) fingerprinting roaming artifacts to identify a roaming user's operator and country, and (iii) reverse-engineering zero-rating identification techniques employed by operators [20] (we only tested SNI and host headers).

Inspecting proactive APDU traffic is highly relevant for the privacy and security: The SIM microcontroller is much more than just a configuration storage or a cryptographic co-processor. Proactive or otherwise hidden SMS and Internet connections are opaque to the user and run in the background. Their actions can have negative effects on the monthly bill.

We can expect eSIMs to behave even more intransparently since they are provisioned and updated over the air. Their SIM card communication is not inspectible by current physical MITM tools [30]. MOBILEATLAS's rSAP and remote SIM bridge closes this gap. Apart from proactive communication, we also experienced SIMs changing their IMSI on the fly for roaming purposes. We expect even more IMSI provisioning from *travel SIMs* with flexible country/region data plans.

MOBILEATLAS provides a robust and versatile testbed and measurement platform for such investigations. We tested our framework with round-trip times of up to 1,000 ms. In comparison, geostationary Internet connections typically have round-trip times between 300 and 600 ms. Therefore we are confident the technique is robust enough to tunnel any SIM to any point on earth.

Expansion Our framework currently consists of twelve probes that cover eight European and two North American countries. We open sourced the design and specifications of our probe to enable other researchers to participate. While budgetary constraints originally motivated the low-cost design, it now allows the project to scale quickly. We continuously seek organizations to host probes and provide on-the-ground support with SIM acquisitions.

User groups We identified multiple user groups – in addition to researchers – who might be interested but are currently not considered: (i) Internet activists, (ii) regulatory authorities, and (iii) ISPs. Due to its open and low-cost design, Internet activists and watch groups could easily use MOBILEATLAS to test networks themselves. By hosting their own testing infrastructure based on our technique, network operators could also benefit, e.g., by testing characteristics and interplay with foreign partner networks. Still, MOBILEATLAS is currently adapted to research purposes and, in the first step, we plan to collaborate with other researchers.

Research directions Future additional measurement and testbed uses can be, but are not limited to (i) cellular infrastructure scans, (ii) SIM vulnerability analyses, (iii) IPv4/IPv6 deployment surveys, (iv) Quality of Experience measurements, (v) geoblocking configurations, (vi) roaming traffic speed discrimination, and (vi) OTT and PBX fraud detection.

Improve SIM tunneling Certain known APDU requests, e.g., static files, should be emulated locally. This includes many local settings, such as the phone book or the list of last calls. This will speed up initialization and connection times as it eliminates round-trips over the network.

Probe hardware iteration with 5G A 5G prototype is in development and will be deployed when funding is secured.

Mobile measurement probes Currently, the probes are used in a stationary manner. With a proper control channel (e.g., Wi-Fi) and sufficient power supply (e.g., in trains or buses), the probes could also be used in a mobile scenario to better reflect user behavior or map network properties geographically.

8.1 Limitations

Low-level radio control MOBILEATLAS lacks the radio versatility of an SDR. However, such a setup would incur radio-regulatory challenges and was therefore out of scope for this paper.

Detectability Certain behaviors (e.g., fast geographical movements of SIMs) could trigger fraud detection systems. So far, we did not experience that.

Test automation The current implementation lacks the anticipated versatile multi-tenant scheduling system as test scripts have to be run semi-automated.

SIM card management The proposed technique vastly reduces hardware and operational complexity and costs. However, managing and maintaining the SIM cards and phone plans remains a time consumer.

The cards have to be purchased, registered, and regularly topped up, in connection with multiple different systems and in different languages.

Many operators do not ship their SIMs internationally, so we need to build a network of partners to either send SIM cards to us or host them locally. So far, eSIMs are not universally offered and may only alleviate some of the efforts.

9 Conclusion

Many current measurement frameworks and tools insufficiently capture (a) the unique properties of cellular networks and (b) their emergent complexities arising from interconnecting them internationally (with all the entailing security and privacy risks). MOBILEATLAS overcomes access technology obliviousness, the quadratic scaling of resources, and background noise challenges. The improved scalability and automation allow for measurements and investigations deemed unpractical before.

To this end, we geographically decoupled radio devices from the subscriber module by taking advantage of the low-level control in SIM communication. Our TCP tunneling technique proved to be robust under high-latency conditions, allowing for worldwide use. Furthermore, MOBILEATLAS provides a high level of data hygiene by decoupling the measurement from any other IP traffic. For billing-related research, MOBILEATLAS offers integrated support for various data accounting interfaces. The low-cost and open-source design makes it ideal for decentralized deployment. All source material and schematics will be available upon publication.

We presented the capabilities in security and privacy research through five exemplary use cases: (i) roaming and price differentiation, (ii) systematic traffic accounting experimentation with the ability to free-ride traffic, (iii) large-scale network property analysis, (iv) low-level APDU analytics, and (v) fingerprinting call characteristics to pinpoint subscribers.

In all cases, we found surprising results. Despite the fact that all operators in our sample use home routing, differences in the metering of roaming connections occur. With those use cases, we demonstrated how to send unaccounted traffic, i.e., *free-riding* or *phreaking*.

Furthermore, in many cases, minuscule differences in call progress tones leak the callee's visiting network operator – enabling a country-level localization. Additionally, some SIM cards proactively and covertly send binary SMS to their home operators; however, their analysis is inconclusive so far.

To lower the barrier for large-scale security and privacy measurements in cellular networks, we plan to expand the network to more researchers and countries. Additionally, we

open-sourced all software and schematics[4].

10 Acknowledgments

We would like to thank the reviewers and shepherd of this conference for their constructive feedback, but also those of NDSS, MobiCom and IMC.

This work were not possible without the hosters of our probes. We heartily thank Giovanni Camurati (EURECOM, now ETH Zurich), Aurélien Francillon (EURECOM), Michael Franz (UCI), Katharina Krombholz (CISPA), nakano, Sebastian Sieh, Tanja Šarčević (SBA Research), Daniel Sokolov (Heise Online), and six more unnamed probe hosters.

We also like to thank Harald Welte for an experience exchange early in the project.

Financing for cellphone research in Austria proven very challenging over the years. Therefore we are very pleased that the this project was funded through the NGI0 PET Fund, a fund established by NLnet with financial support from the European Commission's Next Generation Internet programme, under the aegis of DG Communications Networks, Content and Technology under grant agreement No. 825310. Furthermore the awards from Office of Naval Research, N00014-22-1-2232 and N00014-21-1-2409, DARPA I2O Small Business Technology Transfer (STTR) Program, W31P4Q-20-C-0052, Austrian Science Fund (FWF) P30637-N31 partly supported this work. The competence center SBA Research (SBA-K1) is funded within the framework of COMET–Competence Centers for Excellent Technologies by BMVIT, BMDW, and the federal state of Vienna, managed by the FFG.

References

[1] Özgü Alay, Andra Lutu, Rafael García, Miguel Peón-Quirós, Vincenzo Mancuso, Thomas Hirsch, Tobias Dely, Jonas Werme, Kristian Evensen, Audun Hansen, et al. Measuring and Assessing Mobile Broadband Networks with MONROE. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016.

[2] Özgü Alay, Andra Lutu, David Ros, Rafael Garcia, Vincenzo Mancuso, Audun Fosselie Hansen, Anna Brunstrom, Marco Ajmone Marsan, and Hakon Lonsethagen. MONROE: Measuring Mobile Broadband Networks in Europe. In *Workshop on Research and Applications of Internet Measurements (RAIM)*. IRTF & ISOC, 2015.

[3] Hamad Alrashede and Riaz Ahmed Shaikh. Imsi catcher detection method for cellular networks. In *2nd International Conference on Computer Applications & Information Security (ICCAIS)*, 2019.

[4] https://github.com/sbaresearch/mobile-atlas

[4] Collin Anderson, Philipp Winter, and Roya. Global Network Interference Detection Over the RIPE Atlas Network. In *Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[5] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. Investigating Interdomain Routing Policies in the Wild. In *2015 Internet Measurement Conference (IMC)*, 2015.

[6] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. Lessons Learned from using the RIPE Atlas Platform for Measurement Research. *ACM SIGCOMM Computer Communication Review*, 45(3), 2015.

[7] Vijay A Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael T Hunter, and Patrick Traynor. Pindr0p: Using single-ended audio features to determine call provenance. In *17th ACM conference on Computer and communications security*, 2010.

[8] Vitali Bashko, Nikolay Melnikov, Anuj Sehgal, and Jürgen Schönwälder. BonaFide: A Traffic Shaping Detection Tool for Mobile Networks. In *International Symposium on Integrated Network Management (IM)*, 2013.

[9] David Allen Burgess. More Proactive SIMs., 2021. https://medium.com/telecom-expert/more-proactive-sims-f8da2ef8b189, accessed 2022-10-11.

[10] David Allen Burgess. What is AT&T doing at 1111340002? Welcome to the magical world of proactive SIMs., 2021. https://medium.com/telecom-expert/what-is-at-t-doing-at-1111340002-c418876c212c, accessed 2022-10-11.

[11] Jakub Czyz, Matthew Luckie, Mark Allman, Michael Bailey, et al. Don't forget to lock the back door! a characterization of ipv6 network security policy. In *Network and Distributed Systems Security (NDSS)*, 2016.

[12] Adrian Dabrowski, Georg Petzl, and Edgar R. Weippl. The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection. In *Research in Attacks, Intrusions, and Defenses: 19th International Symposium (RAID)*, pages 279–302. Springer, 2016.

[13] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. IMSI-Catch Me If You Can: IMSI-Catcher-Catchers. In *ACM Annual Computer Security Applications Conference (ACSAC)*, 2014.

[14] Cathal Mc Daid. STK, a-ok? mobile messaging attacks on vulnerable sims. Technical report, 2021. https://vblocalhost.com/uploads/VB2021-Mc-Daid.pdf.

[15] Marcel Dischinger, Massimiliano Marcon, Saikat Guha, P Krishna Gummadi, Ratul Mahajan, and Stefan Saroiu. Glasnost: Enabling End Users to Detect Traffic Differentiation. In *Networked Systems Design and Implementation (NSDI)*. USENIX, 2010.

[16] Docker Documentation. Isolate containers with a user namespace. https://docs.docker.com/engine/security/userns-remap/, accessed: 2022-03-25.

[17] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium*, 2013.

[18] European Parliament and Council of the European Union. Regulation (ec) no 717/2007 on roaming on public mobile telephone networks within the community and amending directive 2002/21/ec. *OJ*, L 171:32–40, 2007. http://data.europa.eu/eli/reg/2007/717/oj.

[19] European Parliament and Council of the European Union. Regulation (eu) 2017/920 amending regulation (eu) no 531/2012 as regards rules for wholesale roaming markets. *OJ*, L 147:30–47, 2017. https://eur-lex.europa.eu/eli/reg/2017/920/oj.

[20] Gabriel K. Gegenhuber, Wilfried Mayer, and Edgar Weippl. Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs. In *IEEE Global Communications Conference (GLOBECOM)*, 2022.

[21] ISO/IEC 7816-3:2006 - Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols. Standard, International Organization for Standardization, Geneva, CH, November 2006.

[22] Ben Jones, Nick Feamster, Vern Paxson, Nicholas Weaver, and Mark Allman. Detecting DNS Root Manipulation. In *International Conference on Passive and Active Network Measurement (PAM)*, 2016.

[23] Arash Molavi Kakhki, Fangfan Li, David Choffnes, Ethan Katz-Bassett, and Alan Mislove. BingeOn Under the Microscope: Understanding T-Mobile's Zero-Rating Implementation. In *Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE)*, 2016.

[24] Arash Molavi Kakhki, Abbas Razaghpanah, Anke Li, Hyungjoon Koo, Rajesh Golani, David Choffnes, Phillipa Gill, and Alan Mislove. Identifying Traffic Differentiation in Mobile Networks. In *Internet Measurement Conference (IMC)*, 2015.

[25] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *26th Annual Network and Distributed System Security Symposium*, NDSS 2019, 2019.

[26] Fangfan Li, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, and Alan Mislove. A Large-Scale Analysis of Deployed Traffic Differentiation Practices. In *Proceedings of the ACM Special Interest Group on Data Communication*. 2019.

[27] Andra Lutu, Diego Perino, Marcelo Bagnulo, and Fabián E Bustamante. Insights from Operating an IP Exchange Provider. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021.

[28] Anna Maria Mandalari, Andra Lutu, Ana Custura, Ali Safari Khatouni, Özgü Alay, Marcelo Bagnulo, Vaibhav Bajpai, Anna Brunstrom, Jörg Ott, Marco Mellia, et al. Experience: Implications of Roaming in Europe. In *Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[29] Peter Ney, Ian Smith, Gabriel Cadamuro, and Kohno Tadayoshi. Sea-Glass: Enabling City-wide IMSI-Catcher Detection. In *Privacy Enhancing Technologies Symposium (PETS)*. Springer, 2017.

[30] OSMOCOM. Simtrace 2. https://osmocom.org/projects/simtrace2/wiki, accessed 2022-10-12.

[31] osmocom.org. pySim-prog - Utility for programmable SIM/USIM-Cards. https://osmocom.org/projects/pysim/wiki, accessed 2022-03-25.

[32] Shinjo Park, Altaf Shaik, Ravishankar Borgaonkar, Andrew Martin, and Jean-Pierre Seifert. White-Stingray: Evaluating IMSI Catchers Detection Applications. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2017.

[33] Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting ss7 redirection attacks with audio-based distance bounding. In *2018 IEEE Symposium on Security and Privacy (SP)*, 2018.

[34] PlanetLab. An open platform for developing, deploying, and accessing planetary-scale serivces, 2002–2020. https://planetlab.cs.princeton.edu/, accessed 2023-02-15.

[35] Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor. Boxed out: Blocking cellular interconnect bypass fraud at the network edge. In *24th USENIX Security Symposium*, 2015.

[36] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal*, 18(3), 2015.

[37] RIPE Network Coordination Centre. RIPE Atlas. https://atlas.ripe.net, accessed 2022-03-25.

[38] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. On Security Research Towards Future Mobile Network Generations. *IEEE Communications Surveys and Tutorials*, 2018.

[39] Merve Sahin and Aurélien Francillon. Over-The-Top Bypass: Study of a Recent Telephony Fraud. In *Conference on Computer and Communications Security (CCS)*, 2016.

[40] Merve Sahin and Aurélien Francillon. Understanding and detecting international revenue share fraud. In *NDSS*, 2021.

[41] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. SoK: Fraud in Telephony Networks.

In *IEEE European Symposium on Security and Privacy (EuroSP)*. IEEE, 2017.

[42] Samsung. What is call & message continuity and how do i use it?, 2023. https://www.samsung.com/uk/support/mobile-devices/call-and-message-continuity/, accessed 2023-02-15.

[43] Security AdaptiveMobile. Simjacker Technical Paper, v1.01. Technical report, 2019. https://simjacker.com/.

[44] The GNOME Project. GNOME Projects NetworkManager. https://wiki.gnome.org/Projects/NetworkManager, accessed 2022-03-25.

[45] The MONROE Alliance. Measuring Mobile Broadband Networks in Europe. https://www.monroe-project.eu, accessed 2022-03-25.

[46] Yiannis Theodorou. The Mandatory Registration of Prepaid SIM Card Users. Technical report, GSM Association, 2013. https://www.gsma.com/publicpolicy/wp-content/uploads/2013/11/GSMA_White-Paper_Mandatory-Registration-of-Prepaid-SIM-Users_32pgWEBv3.pdf.

[47] Yiannis Theodorou, Ken Okong'o, and Erdoo Yongo. Digital Identity: Access to Mobile Services and Proof of Identity 2019: Assessing the impact on digital and financial inclusion. Technical report, GSM Association, 2019. https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2019/02/ProofofIdentity2019_WebSpreads.pdf.

[48] Johanna Ullrich, Katharina Krombholz, Heidelinde Hobel, Adrian Dabrowski, and Edgar Weippl. IPv6 Security: Attacks and Countermeasures in a Nutshell. In *8th USENIX Workshop on Offensive Technologies (WOOT'14)*, 2014.

[49] Thanh van Do, Hai Thanh Nguyen, and Nikolov Momchil. Detecting IMSI-Catcher Using Soft Computing. In *Springer International Conference on Soft Computing in Data Science (SCDS)*, pages 129–140. Springer, 2015.

[50] J. Weil. RFC 6598: IANA-Reserved IPv4 Prefix for Shared Address Space, 2012. accessed 2023-02-16.

[51] Ying Zhang, Zhuoqing Morley Mao, and Ming Zhang. Detecting Traffic Differentiation in Backbone ISPs with NetPolice. In *Internet Measurement Conference (IMC)*. ACM, 2009.

A Probe

A.1 Hardware

MOBILEATLAS probes execute the measurements. Each probe is deployed on a single-board computer that provides an Internet uplink, a USB host support, and a general-purpose in- and output (GPIO) interface that is used for SIM card emulation. The single-board computer is connected to a modem that provides access to the cellular network.

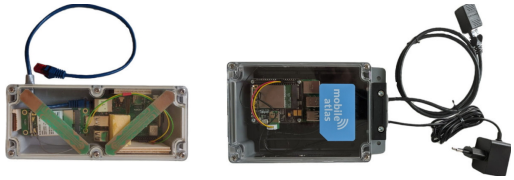


Figure 7: Left: first prototype; right: current version.

We chose the *Raspberry Pi 4* as a cost-friendly platform with more than sufficient computing power for our purpose. After comparing various modems, we selected the *Quectel EG25-G* modem; it has good ModemManager support and tolerates the imprecise timing of our universal asynchronous receiver-transmitter (UART). Since modern modems are designed for notebooks and embedded systems, we used an additional *USB – Mini PCIe adapter*. In order to facilitate user-friendly deployment, we packaged the probe with all modules and antennas into a single case with just two connectors, namely for RJ45 Ethernet and power. The selection for this study was driven by cost-efficiency, and our software is optimized for – but not restricted to – the components described above.

We developed an initial prototype and refined our design in a second version, as presented in Figure 7. We found and solved several issues by testing our prototype in multiple field tests. First, we improved the modem adapter and modem mounting and changed the casing, as there were problems due to rough handling during shipping. Second, we introduced a SIM adapter printed-circuit board (see Figure 10), instead of soldering directly on the SIM slot. Further, we updated the *Raspberry Pi 3* (used for the prototype) to the current version 4. Finally, we exchanged the Huawei ME909 modem with a Quectel EG25-G for worldwide radio band coverage and better ModemManager support thanks to its popularity in the open-source community[5].

A.2 Software

An overview of the probe's software architecture is presented in Figure 8. All components were implemented in Python 3 for rapid prototyping. The low-level protocol communication with the SIM card and modem was built on pySim [31]. For a specific measurement or experiment, the *probe* requests a particular SIM card from the management system (using its IMSI) and sets up a TCP connection to the appropriate *SIM provider*. The SIM provider accepts it and bi-directionally forwards *application protocol data unit* (APDU) traffic.

On the probe's side, a Linux network namespace binds only selected processes to the cellular network connection. Within the namespace, ModemManager and NetworkManager [44] handle the initialization of the modem connection and the networking environment. Next, the measurement or experiment functionality is executed within the namespace

[5]The EG25-G modem is also used in the open-sourced PinePhone.

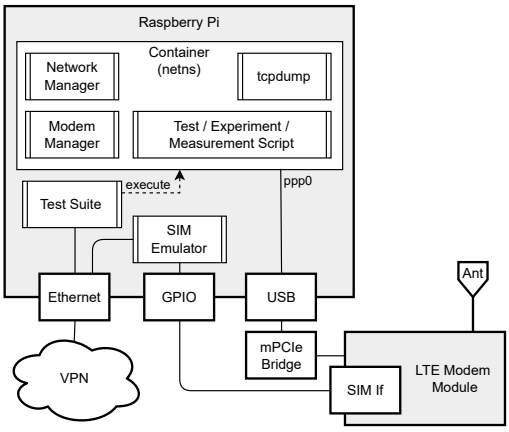


Figure 8: Probe architecture: GPIO ports physically emulate the SIM card to the modem. Separate network namespaces eliminate unwanted background traffic (e.g., from the OS or other system services) in experiments.

and its separated network stack. Any direct test case results, as well as the logs of ModemManager and NetworkManager, are stored in a JSON file, and a tcpdump instance records the occurring traffic for (optional) later analysis. Additionally, tunneled APDU traffic is stored in a separate database and can be exported as a *GSM SIM* PCAP file for later analysis.

In order to automate the credit checking for tests that involve billing, a feasible way to query the current quota needs to be manually implemented for each provider or tariff. The MOBILEATLAS software provides predefined checking procedures and a rich set of methods (e.g., network, SMS, USSD) that can be utilized to access this information in an automated and iterative manner. This reduces manual intervention during test execution.

A.3 SIM Interface

After power-on, the SIM waits for a rising edge on the reset (RST) pin. Since the specification allows different SIM operating voltages (1.8V, 3V, and 5V), the modem increases the voltage and resends the RST signal if no response was received. When an appropriate RST signal is detected, the SIM reacts with an *Answer To Reset* (ATR) message containing various card characteristics (e.g., maximum supported clock frequency). After that, the modem can (but does not have to) perform a *Protocol and Parameter Selection* (PPS) command to change the current connection properties. Our modems use the fastest offered rate; without a PPS they stay at the initial speed.

Data (i.e., the APDUs) are sent via a bi-directional open-collector I/O line with the modem-provided clock (CLK) signal divided by the negotiated divider (ATR or PPS).

Because the UART of our single board computer does not offer a synchronous mode (i.e., a USART), we approximate it through an oscilloscope measurement and program the

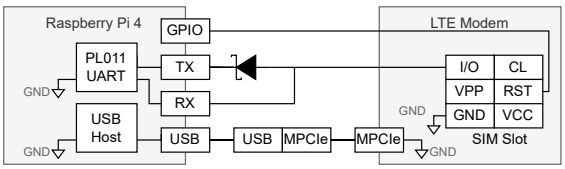


Figure 9: Wiring diagram between the SoC and the modem. The UART emulates SIM cards placed at remote locations.

UART's baud rate accordingly. We emulate an open-collector output with the help of a Schottky diode (Figure 9). The modem already provides a necessary pull-up resistor.

Generous divider settings, tolerances, and the ability to start communication at any clock cycle work in our favor and reduce cost. Additionally, our SIM emulator can repeatedly signal *Waiting Time eXtensions* (WTX) to cover for longer network round-trip times.

An additional GPIO port serves the SIM's RST pin. For the sake of simplicity, we left out the ground wire and reused the USB's ground path, leading to slightly increased noise levels.

As our SIM tunnel implementation is aware of the ISO7816 protocol, we can independently choose connection parameters via ATR and PPS on both sides of the tunnel. We can trigger a PPS procedure from the modem when we present an ATR with relatively fast settings. Alternatively, we can signal the same settings that were used for the initial ATR transmission so the modem skips PPS.

B SIM Provider

The TCP tunnel between a SIM emulator process on the probe and the SIM provider forms a virtual circuit in which one SIM can be connected to precisely one modem at a time.

SIM providers are responsible for connecting the actual SIM cards to the measurement framework. There exists dedicated hardware to connect a multitude of different SIM cards. The so-called SIM banks are sometimes used for OTT VoIP gateways. These devices are often expensive, proprietary, and do not perfectly reflect our use case. Instead, we chose the cost-efficient approach of connecting SIM cards with individual SIM readers. These devices are cheap, readily available, and connect via USB. Our system supports cheaper USB-serial-type devices as well as full-fledged PC/SC readers.

Although the USB standard theoretically allows for up to 127 connected devices, we experienced non-deterministic failures on multiple hosts (enumeration failure, USB driver errors) when chaining multiple USB hubs and connecting an *unusually* high amount of devices to the USB bus. We tested multiple hardware configurations, and since some devices (e.g., Raspberry Pi, Intel NUC, cheap USB hubs) did not work

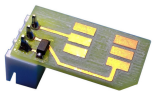


Figure 10: SIM adapter is plugged into the modem's socket.

reliably, we ended up using a commodity laptop computer (in combination with multiple FE 2.1. hubs). We were able to successfully test our SIM provider for up to 32 concurrently attached SIM cards via one USB host. The framework allows us to add any additional SIM cards from other distributed sources, although we currently obtain all SIM cards from one SIM provider. With this distributed approach, we facilitate the testing of a multitude of mobile contracts, as one could temporarily add a SIM card and test it on a large scale.

B.1 Bluetooth SIM Access Profile

As an alternative to USB-based readers, we also support sharing a SIM card via Bluetooth's (remote) SIM Access Profile (rSAP). This legacy protocol was used for automotive integration, allowing external devices without a SIM card (such as cars) to access a phone's SIM card.

Our SIM provider can act as an SAP client to a smartphone's rSAP server. On Android, new rSAP connections require user approval and only serve the primary SIM card, i.e., a possible eSIM needs to be configured as the primary SIM card. Our tests run successfully on an off-the-shelf, non-rooted Google Pixel 3a.

C Benchmarking

In this Section, we evaluate the compatibility of different modem- and SIM reader models.

C.1 Different modems

We tested five different modem types (four state-of-the-art LTE modems, one legacy 3G device) and compared their characteristics, as shown in Table 4. Since we want our framework to utilize the full spectrum of mobile phone functions, we tested key features of every modem via the ModemManager interface. MOBILEATLAS supports network connections, calling, SMS, and sending AT commands with all of the tested modems. USSD codes work with four tested modems, whereby two modems required a small patch to set the correct character encoding during the setup routine. The remaining modem accepts USSD directly via AT commands, but not via the ModemManager interface. Furthermore, we benchmarked the time needed for setting up a successful mobile network connection with MOBILEATLAS in a domestic setting with good reception and a low-latency connection. Although the average time differs, it stays below one minute for all models.

Table 5 summarizes the SIM tunneling characteristics for these five modems. All modems but one support tunneling with a faster PPS-negotiated baud rate. The initialization procedure and the number of requested APDUs differ vastly between modems. Telit's modem (M4) is a clear outlier because it requests large parts of the SIM's file system at startup.

To determine the right baud rate for our SIM tunnel interface, we measured each modem's clock with an oscilloscope.

Table 4: Supported functionality for different modems (via `option` driver).

	Quectel EG25-G	Huawei ME909s-120	SIMCom SIM7600E-H	Telit LE910	Sierra MC8355
	M1	M2	M3	M4	M5
Network	✓	✓	✓	✓	✓
Connection time (sec)	28	58	29	51	51
Start/Stop Call	✓	✓	✓	✓	✓
Send/Receive SMS	✓	✓	✓	✓	✓
Send/Receive USSD	∼	✓	∼	×	✓
Send AT Cmd	✓	✓	✓	✓	✓

∼ works with adjusted encoding

Table 5: Our SIM tunneling works well across all tested COTS modem models.

	M1	M2	M3	M4	M5
Tunnel without PPS	✓	✓	✓	✓	✓
Tunnel with PPS	✓	✓	✓	×	✓
Initialization (sec)	24	49	25	131	51
Initialization (#APDUs)	261	638	271	1913	501
SIM clock (MHz)	3.842	3.759	3.842	3.250	3.842
Clock tolerance (neg %)	4.1	3.8	4.1	4.2†	4.1
Clock tolerance (pos %)	4.5	4.9	4.5	5.2†	4.4

† without PPS

Furthermore, we checked how tolerant the modems are to APDUs that were based on deviant clock speeds. Our measurements show that the tested modems are fairly tolerant and the SIM tunnel's base clock can diverge by around 4% without having any negative impact on the APDU communication.

C.2 SIM readers

SIM providers rely on pySim [31] in combination with hardware SIM readers, with either PC/SC (CCID) interfaces or a simple USB serial interface. The former are more expensive (USD 10–30), but are typically more versatile. Several low-cost readers (USD 2–5) provide a simplistic serial interface via USB (*USB Serial Adapter*). We tested seven SIM readers with 14 different SIM cards from five countries, all purchased in 2021. The same SIM cards are deployed in our use cases in Section 7.1. As Table 6 shows, three of the SIM readers worked with every SIM card. Two SIM readers had problems with one specific SIM card. One particular low-cost SIM reader had problems with several SIM cards in PPS mode, but accepted 12 out of 14 SIMs without PPS.

In our sample, PC/SC devices processed APDUs faster than the low-cost serial-based SIM readers.

Table 6: SIM reader compatibility

Modem	Supported SIMs
Gemalto HWP108841HY	14 / 14
Manhattan 102049	14 / 14
Woxter CN43296012	14 / 14
Bit4id MiniLector EVO	13 / 14
Low-cost SIM reader, White	13 / 14
Low-cost SIM reader, Blue	7 / 14
Low-cost SIM reader, Blue (w/o PPS)	12 / 14

3 Measuring Traffic Classification and Zero-Rating Tariffs during International Roaming Scenarios

Publication Info

Title	Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs
Authors	Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl
Publication Status	This paper is included in the Proceedings of the IEEE Global Communications Conference (GLOBECOM 2022), pp. 203–208, 2022. CORE2023 Ranking: B.
DOI	`https://doi.org/10.1109/GLOBECOM48099.2022.10001701`
Code Artifacts	`https://github.com/sbaresearch/mobile-atlas`
arXiv	`https://arxiv.org/abs/2403.08066`
Reference	[GMW22]

Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs

Gabriel K. Gegenhuber*†, Wilfried Mayer‡§ and Edgar Weippl¶
University of Vienna†, SBA Research§
Email: *gabriel.karl.gegenhuber@univie.ac.at, ‡wmayer@sba-research.org, ¶edgar.weippl@univie.ac.at

***Abstract*—Zero-rating, the practice of not billing data traffic that belongs to certain applications, has become popular within the mobile ecosystem around the globe. There is an ongoing debate whether mobile operators should be allowed to differentiate traffic or whether net neutrality regulations should prevent this. Despite the importance of this issue, we know little about the technical aspects of zero-rating offers since the implementation is kept secret by mobile operators and therefore is opaque to end-users and regulatory agencies.**

This work aims to independently audit classification practices used for zero-rating of four popular applications at seven different mobile operators in the EU. We execute and evaluate more than 300 controlled experiments within domestic and internationally roamed environments and identify potentially problematic behavior at almost all investigated operators. With this study, we hope to increase transparency around the current practices and inform future decisions and policies.

***Index Terms*—zero-rating, net neutrality, mobile broadband, roaming, traffic differentiation, network management**

I. Introduction

Cellular networks have become a major access technology to the public Internet that can also be used across national borders. In June 2017, the European Union abolished data roaming fees for the intra-EU/EEA area under the "roam like at home" doctrine. This regulation made roaming in foreign cellular networks feel and behave like at the home operator and led to a drastic increase in roaming traffic [1]. Mobile broadband connections do not only differ from landline data connections from a usage perspective but also diverge in terms of tariff models. According to BEREC [2], a growing number of mobile network operators (MNOs) in the EU have introduced differential pricing offers (e.g., zero-rating), and some of them have already been taken to court [3] by regulators for disrespecting net neutrality principles (e.g., not applying zero-rating during intra-EU data roaming). Obviously, correct zero-rating is crucial to consumer protection since misclassification can lead to illegitimately billed units for customers. Possibly, many net neutrality violations remain undiscovered and proving an operator's misbehavior is not easily possible for end-users or regulatory agencies.

To address these issues, this work independently audits zero-rating practices of selected operators and gives valuable insights into the classification metrics that are currently used within the industry.

Our main contributions are:

- We propose a methodological approach to probe web endpoints for zero-rating.
- We use this approach to evaluate zero-rating of four popular applications at seven operators from three countries.
- We test the effect that intra-EU roaming has on zero-rating by executing our experiments during domestic and roaming usage scenarios in eight different countries.
- We evaluate our results and give an overview of the used classification metrics and encountered classification errors.

The remainder of this paper is organized as follows. Section II gives an overview of related studies that are relevant to this work. In Section III, we describe our methodological approach and quickly introduce the testbed that was used to execute our experiments. Section IV summarizes the results that were collected throughout this study. Finally we discuss our results in Section V and conclude with Section VI.

II. Related Work

Aside from hot debates about political and regulatory decisions, net neutrality has also been an interesting research target from a technical perspective. Net neutrality measurements usually aim to detect traffic differentiation in terms of

- rate limiting (traffic shaping, traffic policing),
- traffic blocking (censorship),
- traffic manipulation,
- economic differentiation (differential pricing, zero-rating).

Identifying traffic differentiation. *Glasnost* [4] and *NetPolice* [5] were among the first tools that detect bandwidth throttling of specific protocols (e.g., BitTorrent). While those tools were built for landline and desktop environments, *BonaFide* [6] was released as a smartphone application. It replicates the capabilities of *Glasnost* to the mobile world, minimizes data consumption, and supports a broader set of protocols (e.g., SIP and video streaming). In contrast to prior work, *Differentiation Detector* [7] and *Wehe*[1] [8] do not target specific protocols, but moved to a application-centric design that mimics arbitrary protocols. By replaying pre-recorded application-generated traffic, they can treat it as a "black box"

This work was funded through the NGI0 PET Fund, a fund established by NLnet with financial support from the European Commission's Next Generation Internet program, under the aegis of DG Communications Networks, Content and Technology under grant agreement No 825310.

[1] https://wehe.meddle.mobi

and do not need to provide specific implementation details of occurring protocols.

Although the mentioned tools were created to detect traffic differentiation in terms of rate-limiting, economic differentiation is usually built upon the same classifiers and therefore relies on similar metrics. A study that investigates traffic classification in middleboxes [9] has shown that the used policies are often relatively simple and only match certain keywords within HTTP/HTTPS fields. In contrast, our work does not only focus on keyword-based classifiers but also shows that IP-based classification is in widespread use by many operators.

Economic differentiation. Differential pricing practices like zero-rating or application-specific data quotas were rarely seen in the fixed-line landscape but have become common over the past years in the cellular field. Although there is one case study investigating a zero-rating offer by T-Mobile (US) that targets video streaming [10], there is no work that compares current zero-rating practices across different operators or countries.

Our work aims to cover this research gap by comparing the used classification rules of popular applications at different operators in various European countries. Besides investigating domestic usage we also take the roaming context into account.

Emerging technologies. The Internet and the used communication protocols are under constant change and evolution. Previous work [11], [12] has shown that emerging technologies, such as IPv6 and QUIC (which is used for UDP-based communication at HTTP3), can cause problems with existing security policies or firewall configurations that are rather static.

Our work investigates whether current classification mechanisms also work with cutting-edge technology (i.e., IPv6 and HTTP3) that is already used by popular (zero-rated) applications in the wild.

III. Methodology

There are several parts to our methodology: analyzing the European cellular market and available zero-rating offers, characterizing web endpoints that are used within zero-rated applications, and using our testbed to determine which metrics are applied to classify the data traffic corresponding to a certain application.

A. Market, Tariff, and Application Analysis

To find out which countries and providers offer zero-rating tariffs, we conducted an EU-wide market analysis. It has become increasingly popular that MNOs lease their wireless network infrastructure to Mobile Virtual Network Operators (MVNOs) that offer services to their customers but do not own any infrastructure. Compared to an MNO, becoming an MVNO is relatively easy and requires less financial effort. Thereby, many countries have got a vast amount of operators (e.g., Austria currently has about 40 MVNOs, despite being a relatively small country). However, due to well-established MNOs and the high fluctuation of MVNOs, the latter usually play a minor role in terms of actual market penetration. To limit the effort but accordingly respect the market situation, we limited our market analysis to bare-metal consumer-grade MNOs in every country. After identifying the relevant players, we looked at the available tariffs to find out whether they offer differential pricing or zero-rating programs. For this step, our primary source of information was a provider's website. The language barrier at foreign countries and complex tariff structures (e.g., prepaid vs. postpaid, minimum contract duration, packages that are only available in certain tariffs) often made it cumbersome to get the required information.

According to our market analysis that was conducted in May 2021, operators in 24 EU countries (ca. 89%) have implemented differential pricing or zero-rating offers. SIM card registration is currently required in 14 of 27 EU countries (ca. 52%) [13].

We selected three countries (Austria, Croatia, and Romania) that offered a good coverage of zero-rating tariffs and were available in our testbed environment and then acquired SIM cards of the relevant MNOs. In Austria, all three MNOs implement zero-rating, while in Croatia and Romania, two out of three providers offer relevant tariffs. Table I shows an overview of the applications that are included in the analyzed tariffs at each operator.

For our measurements, we selected the four topmost included applications: WhatsApp (seven operators), Snapchat (four operators), Messenger (four operators), and Facebook (four operators).

Although our target applications use a variety of communication protocols (e.g., XMPP, RTP, or MQTT) they usually rely on Content Delivery Networks (CDNs) that distribute the actual content via web (i.e., HTTPS). Legacy protocols often use TLS connections to encrypt the data (e.g., SRTP, or MQTTS). Since those web endpoints are responsible for a substantial part of the data traffic, this needs to be zero-rated by the operator.

To determine relevant web endpoints that are used within the applications of interest, we obtained exemplary traffic dumps for each application. We ensured to use the most recent Android applications from the AppStore and recorded the data traffic that occurred within five minutes of application usage via *PCAPdroid*[2]. Since most operators explicitly exclude external and advertisement content from zero-rating in their

[2]https://github.com/emanuele-f/PCAPdroid

TABLE I
Overview of the selected tariffs and zero-rated applications

Operator	Included Applications (Zero-Rating)
AT-1	WhatsApp, Snapchat, Messenger, Viber
AT-2	WhatsApp, Snapchat, Facebook, Instagram, TikTok, Twitter
AT-3	WhatsApp, Messenger, FM4, Ö3, Ö1, Energy, Superfly, Antenne, 886, Kronehit, Radio Arabella
HR-1	WhatsApp, Snapchat, Messenger, Facebook
HR-2	WhatsApp, Snapchat, Messenger, Facebook, Instagram, TikTok
RO-1	WhatsApp, Facebook, TikTok
RO-2	WhatsApp, Message+

Terms of Service (ToS), we ensure to stay within the original application during our recording sessions. The same applies to voice- and video calls (e.g., via WhatsApp), which often are exempted from the zero-rating offer. To check whether an application or web endpoint supports different technology stacks (e.g., IPv4/IPv6, or HTTPS/HTTP3) we record multiple traffic dumps in varying testing environments (e.g., IPv4 only, IPv6 only, or Dual Stack).

By analyzing our traffic dumps, we chose one web endpoint for each application that causes a substantial part of data traffic and, furthermore, supports a broad set of communication protocols. Table II shows which endpoints were selected for the probed applications. All selected web endpoints support both IPv4 and IPv6, as well as HTTP, HTTPS, and HTTP3. Notably, the endpoints which are used by Messenger and Facebook overlap to a high degree. Presumably, one application cannot be separately zero-rated without also triggering classification of traffic that belongs to the other application.

TABLE II
WEB ENDPOINTS AND RESOURCES THAT WERE USED TO TEST FOR CLASSIFICATION

Application	Endpoint	Used Resource
WhatsApp	static.whatsapp.net	Logo[a]
Snapchat	app.snapchat.com	User Avatar[b]
Messenger	scontent.xx.fbcdn.net	Favicon[c]
Facebook	scontent.xx.fbcdn.net	Favicon[c]

[a] static.whatsapp.net/rsrc.php/v3/yP/r/rYZqPCBaG70.png
[b] app.snapchat.com/web/deeplink/snapcode
[c] scontent.xx.fbcdn.net/favicon.ico

B. Testbed

As described in Section II, traffic differentiation measurements that aim to detect rate limiting of certain applications are often crowd-sourced and executed on volunteers' smartphones. This does not work for detecting economic differentiation since we do not have insights into the data units that are billed in tariffs of foreign entities. Furthermore, unwanted background activities (e.g., traffic that is caused by the smartphone user) might distort measurement results. Therefore, we need controlled experiments on dedicated SIM cards where only explicit data traffic is transmitted.

To execute measurements within a domestic and various roaming environments we used the MOBILEATLAS [3] measurement platform. MOBILEATLAS geographically decouples SIM card and modem by tunneling the SIM card's protocol over the Internet and emulating its signal on the LTE modem. This boosts the scalability and flexibility of international measurements in the cellular field because it allows testing roaming effects on a large number of operators without physically moving any hardware between different countries. The platform is currently deployed in eight European countries: Austria, Belgium, Croatia, Finland, Germany, Romania, Slovakia, and Slovenia. As Fig. 1 shows, the framework can be structured

[3] https://www.mobileatlas.eu

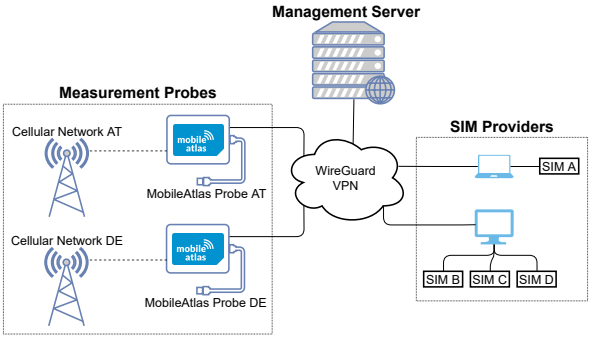


Fig. 1. Architecture and components of the MOBILEATLAS measurement platform

into three main components: SIM providers that allow sharing SIM card access, measurement probes that act as a local breakout to the cellular network, and a management server that connects the prior two components and acts as command and control unit for the measurement probes.

To execute a measurement, any SIM card that is attached to our system via a reader device (e.g., a PC/SC reader) can be virtually connected with a measurement probe in the desired target country. MOBILEATLAS provides various interfaces to interact with the measurement probe's modem and isolates the cellular network connection from any unwanted noise sources. For billing-related measurements, it offers a template for credit checking that can leverage different modem capabilities (e.g., sending SMS messages or USSD-codes and a separate network gateway to access the customer zone) to retrieve the current data quota of a SIM card.

Since there is no standardized interface that allows retrieving the available credit information across different operators, we needed to provide a specific implementation for each tariff tested in this study. Most operators implement credit retrieval through one or more of the following ways: SMS message, USSD code, voice call, website (customer area), or mobile app. We analyzed the available methods for all operators and usually chose the approach that provided the most verbose billing information. The granularity in which the current data quota is reported differs heavily from fine-grained reporting in byte precision to coarse gigabyte estimates between each

TABLE III
OPERATOR-WISE OVERVIEW OF USED CREDIT CHECKING METHOD

Operator	Credit Checking Method
AT-1	SMS (domestic), APP (roaming)
AT-2	APP
AT-3	APP
HR-1	APP
HR-2	APP
RO-1	APP (+ SMS for OTP Login)
RO-2	USSD

operator and retrieval method. Table III shows that we use the mobile app approach in most cases but opted for the SMS and USSD approach for one operator each. In one case (RO-1), the password-based login form required to solve a captcha, which forced us to use an alternative login page that uses a one-time password (OTP) that is sent via SMS. To find the appropriate API endpoints that are used within the apps to retrieve the credit from the network, we had to reverse engineer each provider's mobile app. Depending on the application, we used various approaches and tools (e.g., static/dynamic analysis, JADX, Frida, and Burp Suite) to determine how a provider's app retrieves a customer's credit.

C. Measurement Implementation

For tests on differential pricing, we need to know whether specific traffic is deducted from the available credit units or funds. To cope with different update latency of consumed units and to enable running multiple payloads without in-between waiting for the billing records to update, we use binary exponents, i.e., every payload uses traffic amounts selected from $baseunit \times 2^{index}$. For example, when the first payload causes one megabyte of traffic, the second has two megabytes, the third four megabytes, and so on. When the final traffic billing arrives (which in our case is a control payload that is always billed), we can unambiguously deduct which payloads were counted towards our tariff's quota.

To reveal potential metrics that are used for classification, we designed three different experiments: at first, we verify that an endpoint is actually zero-rated by an operator, secondly, we check for IP-based classification, and lastly, we check for hostname-based classification.

1) Verify Zero-Rating: In the first experiment, we cause regular data traffic to our examined web endpoint. Fig. 2 describes the involved actors and the traffic flow that occurs during verification for an endpoint with the hostname *application.com*. We always query a static resource (e.g., the favicon) that actually is present at the target web server to increase the response size and speed up our experiments. Alternatively, we can request the web server's root or index file, which does not require any additional knowledge about the probed web application but is usually slower because of possible `HTTP 404`-responses. We repeatedly send out web queries until the caused data traffic reaches our target size (e.g., one megabyte). To minimize unwanted background noise, we ensure that DNS queries are only issued once and cached locally for subsequent requests. Finally, the test generates control traffic to a third party that is not part of any zero-rating program and, therefore, normally billed. As previously described, the test terminates as soon as the control traffic is recognized. To execute this experiment for multiple protocols (i.e., HTTP, HTTPS, and HTTP3), we use the payload multiplexing technique that is described at the beginning of this section.

2) Detect IP-based Classification: This experiment aims to expose IP-based classification rules. As Fig. 3 shows, the involved actors have not changed from the previous step. We connect to the application's web server but replace the

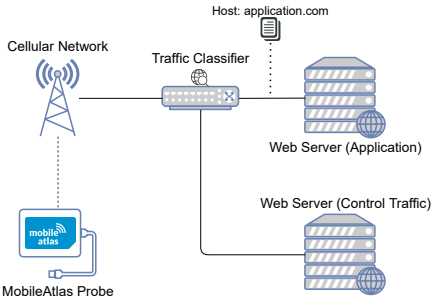


Fig. 2. Involved actors and traffic flow when verifying zero-rated data traffic

hostname that is sent in an HTTP request or within the initial TLS handshake (i.e., the SNI header for HTTPS or HTTP3). Although the packets are sent to the real application's web server (i.e., its actual IP address), they do not contain the real endpoint's hostname because it was exchanged with a dummy value (i.e., *example.com*).

When our test traffic is nevertheless zero-rated by an operator, the used classification is presumably based on IP addresses. In case of billed test traffic, we suppose that hostname-based classification rules were used.

3) Detect Hostname-based Classification: The first step at this experiment is to retrieve the IP address of the server that holds the examined web resource. Secondly, an Amazon EC2-instance is launched automatically and forwards the corresponding ports for the protocols that should be tested (e.g., `TCP80` and `TCP443` for HTTP and HTTPS and `UDP443` for HTTP3). Thus, when a TCP connection to the freshly spawned EC2 server is initiated on port 80, the connection is forwarded to the original web server. Thereby, the same content is served, although the data packet that is processed by the provider is headed to a different IP address. When executing the payload for a certain protocol, the measurement environment pins the hostname of the original web resource to the IP address of the EC2 instance. Therefore, the measurement is conducted against a third-party IP address (i.e., against the EC2 server). Fig. 4 gives an overview of the involved actors and the traffic flow during this experiment. When the data packets are passing

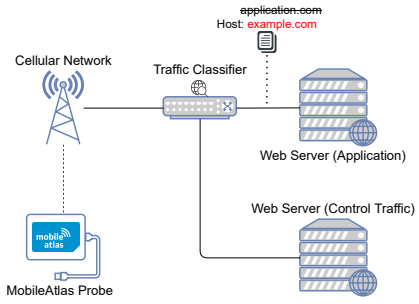


Fig. 3. Actors and traffic flow when checking for IP-based classification

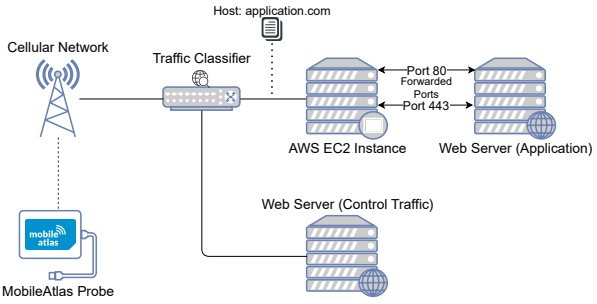


Fig. 4. Involved actors and traffic flow when checking for hostname-based classification

TABLE IV
USED CLASSIFICATION METRICS AT THE TESTED OPERATORS AND APPLICATIONS

Operator	Roaming	WhatsApp	Snapchat	Messenger/Facebook
AT-1	Yes	IP	IP, Host	$
AT-2	Yes	IP	IP[a]	IP
AT-3	Yes	IP	×	$
HR-1	No	IP	Host	IP
HR-2	Yes	IP	IP, Host[b]	IP
RO-1	No	IP, Host[b]	×	IP, Host[b]
RO-2	×	IP[c]	×	×

$ traffic fully billed. × not part of zero-rating tariff.
[a] IPv4 only. [b] HTTPS only. [c] TCP only.

the classifier, the hostname within the packets matches the one from the application. However, the IP address of the packets does not match the address of the application's web server because the packets are headed to the EC2 instance. Yet, the content of the data packets is equal to the previous test because the EC2 instance simply acts as a proxy to the actual application's web server.

When our test traffic is zero-rated by an operator, we imply that hostname-based classification was used.

D. Ethical Considerations

Ethical considerations are vital to the field of measurements, especially with active measurements conducted in live systems. We tried to reflect normal user behavior whenever possible, e.g., by introducing a minimum waiting time between switching to another country with a SIM card. Furthermore, we took care not to overstress any mobile operator's infrastructure we were interacting with (e.g., by rate-limiting credit retrieval). Because our measurements might still stay in conflict with an operator's ToS and possibly lead to blocked SIM cards, we ensured only to use SIM cards that were exclusively purchased for measurements in this paper. We conduct our measurements to get a better understanding of current traffic differentiation mechanisms. This can be of advantage for a variety of stakeholders, including but not limited to: customers, content providers, regulatory authorities, policy makers, and MNOs.

IV. Experiment Results

We executed the experiments that are described in Section III for all selected operators and applications and tested classification metrics for the chosen endpoints using HTTP, HTTPS, and HTTP3.

Table IV summarizes the results of our experiments that were executed within two measurement periods (September 2021 and May 2022). Some operators combine IP- and hostname-based classification for a single endpoint. In such cases, the traffic is zero-rated when at least one of the two rules applies.

General misclassification. Our measurements indicate that several operators use wrongfully configured classification and therefore bill traffic to their customers that should actually be zero-rated. For example, two Austrian operators (AT-1 and AT-3) do not zero-rate any traffic that goes to our selected web endpoint of the Messenger app. In both cases, we verified that there is indeed a classification problem in practice by plugging the SIM card into a smartphone, using the Messenger app, and capturing the caused data traffic. We took snapshots of the available data quota before and after the test and could verify that only minor parts of the occurred data traffic were zero-rated (e.g., only 0.47 out of 17.39 megabytes in a randomly selected test sample).

IPv6 and HTTP3. At the two operators (AT-1 and AT-2) that already deployed IPv6 (DualStack) to their customers, we also tested whether accessing an endpoint via IPv6 instead of IPv4 influences traffic classification. The operator AT-2 does only zero-rate traffic that goes to the IPv4 address of the Snapchat endpoint but fully bills data packets that go to the corresponding IPv6 address.

For HTTP3 we observed similar behaviour at multiple providers. The provider HR-1 relies on hostname-based classification for the Snapchat endpoint. During our first measurement period in September 2021, HTTP3 traffic to Snapchat was wrongfully billed. However, in May 2022, the same traffic was correctly zero-rated (i.e., the operator updated the responsible classification rules).

Similarly, AT-1 upgraded its hostname-based classification metrics to work with HTTP3 between our two measurement periods. In contrast to HR-1, the traffic at AT-1 was nevertheless classified correctly with the old metrics because a combined classification approach was used, and the additional IP-address rule ensured the correct classification of HTTP3.

Lastly, the IP-based classification for the selected WhatsApp endpoint does not work with HTTP3 at the operator RO-2.

Roaming. For all tested tariffs that include EU-wide roaming in their tariffs (i.e., all, except RO-2), we checked whether zero-rating is active when connecting from a foreign country. To examine whether there is any difference in classification between local and roamed connections, we executed our classification experiments in all eight countries that are available in our testbed at each provider. To choose the appropriate roaming partner, we used automatic network selection. We

did not observe any cases of local-breakout, since all operators used home-routing to terminate their roamed data connections.

We observed that four operators (AT-1, AT-2, AT-3, and HR-2) also applied zero-rating during roaming, while two operators (HR-1 and RO-1) had zero-rating disabled and fully billed the occurring traffic. We did not notice any differences between different visited countries (e.g., our results in Germany were identical to the results measured in Finland).

V. Discussion and Limitations

The results of our experiments show that several operators:

- wrongfully bill a substantial amount of traffic from zero-rated applications.
- do not correctly apply zero-rating when newer technology stacks (i.e., IPv6 or HTTP3) are used.
- turn off zero-rating during intra-EU roaming, although this has already been impeached by regulatory agencies in several countries.

Overall, we were surprised by the huge amount of wrongfully billed traffic. Although we noticed that several operators adapted their rules during our two measurement periods to support HTTP3 connections at Snapchat, it is already used by the application for at least one year [14]. From a customer's perspective, paying for traffic that is already part of the purchased tariff is not acceptable.

We believe that operators should be more transparent and make the technical implementation of zero-rating offers publicly available. This would enable Internet activists and regulatory authorities to find classification errors more quickly and prevent wrongfully billed traffic.

Our methodology is built to detect simple classification metrics (i.e., IP- or hostname-based classification). Theoretically, an operator could also deploy more complex algorithms (e.g., traffic fingerprinting, machine learning [15]) to detect data packets that belong to a certain application. Furthermore, our approach could trigger anomaly detection when repeatedly requesting one single web resource. However, the amount of traffic that was caused by our experiments was considerably small (e.g., one megabyte), and we did not observe anything that would hint at being flagged by an operator. To verify the lack of zero-rating for a certain application, we usually ran additional tests by manually plugging the SIM card into a smartphone and observing the billed units when the probed application is in actual use.

Although our used testbed allows rather flexible and mostly automated measurements, it still requires effort to run and evaluate these experiments, limiting our tests' coverage. Since one application usually communicates with a plethora of web endpoints, it remains to future work to improve test automatability and run more verbose experiments.

VI. Conclusion

This paper provides a technical analysis of differential pricing practices at European MNOs. Our analysis shows that operators currently use both IP- and hostname-based classification. Moreover, we show that several operators do not correctly classify the traffic of certain applications or when particular protocols are used. Furthermore, we demonstrate that some providers do not apply zero-rating in roaming usage scenarios.

To encourage other researchers to look into the topic and improve the available tools for controlled and international cellular measurements, we've open-sourced the implementation of our experiments alongside our MobileAtlas testbed and will publish all collected measurement artifacts that were used within this study upon publication. We hope that this study will help to inform future decisions and policies around net neutrality.

References

[1] "International Roaming BEREC Benchmark Data Report April 2019 – September 2019," BEREC, Tech. Rep., 2020. [Online]. Available: https://berec.europa.eu/eng/document_register/subject_matter/berec/download/0/9031-international-roaming-berec-benchmark-da_0.pdf

[2] "BEREC Report on the implementation of Regulation (EU) 2015/2120 and BEREC Open Internet Guidelines 2021," BEREC, Tech. Rep., 2021. [Online]. Available: https://berec.europa.eu/eng/document_register/subject_matter/berec/download/0/8840-report-on-the-implementation-of-regulati_0.pdf

[3] E. C. of Justice, "Judgments of the EU Court of Justice in Case C-854/19," https://curia.europa.eu/juris/document/document.jsf?docid=245531&doclang=EN, 2021, accessed: 2022-05-21.

[4] M. Dischinger, M. Marcon, S. Guha, P. K. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation," in *Networked Systems Design and Implementation (NSDI)*. USENIX, 2010, pp. 405–418.

[5] Y. Zhang, Z. M. Mao, and M. Zhang, "Detecting Traffic Differentiation in Backbone ISPs with NetPolice," in *Internet Measurement Conference (IMC)*. ACM, 2009, pp. 103–115.

[6] V. Bashko, N. Melnikov, A. Sehgal, and J. Schönwälder, "BonaFide: A Traffic Shaping Detection Tool for Mobile Networks," in *International Symposium on Integrated Network Management (IM)*. IFIP/IEEE, 2013, pp. 328–335.

[7] A. Molavi Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove, "Identifying Traffic Differentiation in Mobile Networks," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 239–251.

[8] F. Li, A. A. Niaki, D. Choffnes, P. Gill, and A. Mislove, "A Large-Scale Analysis of Deployed Traffic Differentiation Practices," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 130–144.

[9] F. Li, A. M. Kakhki, D. Choffnes, P. Gill, and A. Mislove, "Classifiers unclassified: An efficient approach to revealing IP traffic classification rules," in *Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 239–245.

[10] A. M. Kakhki, F. Li, D. Choffnes, E. Katz-Bassett, and A. Mislove, "BingeOn Under the Microscope: Understanding T-Mobiles Zero-Rating Implementation," in *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, 2016, pp. 43–48.

[11] J. Czyz, M. Luckie, M. Allman, M. Bailey *et al.*, "Don't forget to lock the back door! A characterization of IPv6 network security policy," in *Network and Distributed Systems Security (NDSS)*, 2016.

[12] K. Y. Gbur and F. Tschorsch, "A QUIC (K) Way Through Your Firewall?" *arXiv preprint arXiv:2107.05939*, 2021.

[13] Y. Theodorou, C. Lowe, and E. Yongo, "Access to Mobile Services and Proof of Identity 2021," GSM Association, Tech. Rep., 2021. [Online]. Available: https://www.gsma.com/mobilefordevelopment/resources/access-to-mobile-services-and-proof-of-identity-2021/

[14] Snapchat Client Network Team, "QUIC at Snapchat," https://eng.snap.com/quic-at-snap, June 2021, accessed: 2021-06-24.

[15] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

4 Detecting AS-level Centralization in Tor Using Democratized Traceroute Measurements

Publication Info

Title	An Extended View on Measuring Tor AS-level Adversaries
Authors	Gabriel K. Gegenhuber, Markus Maier, Florian Holzbauer, Wilfried Mayer, Georg Merzdovnik, Edgar Weippl, Johanna Ullrich.
Publication Status	This paper is included in Computers & Security, Volume 132, Article 103302, 2023. SCImago Ranking: Q1.
DOI	`https://doi.org/10.1016/j.cose.2023.103302`
Code Artifacts	`https://github.com/sbaresearch/ripe-tor`
arXiv	`https://arxiv.org/abs/2403.08517`
Reference	[GMH+23]

Computers & Security 132 (2023) 103302

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

An extended view on measuring tor AS-level adversaries

Gabriel K. Gegenhuber[a,*], Markus Maier[a], Florian Holzbauer[a], Wilfried Mayer[b], Georg Merzdovnik[b], Edgar Weippl[a], Johanna Ullrich[c]

[a] University of Vienna, Research Group Security and Privacy, Kolingasse 14-16, Vienna 1090, Austria
[b] SBA Research, Floragasse 7, Vienna 1040, Austria
[c] Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle, University of Vienna, Kolingasse 14-16, Vienna 1090, Austria

ARTICLE INFO

Article history:
Received 9 January 2023
Revised 12 May 2023
Accepted 22 May 2023
Available online 25 May 2023

Keywords:
Tor
RIPE atlas
Traceroute measurements
Censorship
Privacy
Anonymity
Routing

ABSTRACT

Tor provides anonymity to millions of users around the globe which has made it a valuable target for malicious actors. As a low-latency anonymity system, it is vulnerable to traffic correlation attacks from strong passive adversaries such as large autonomous systems (ASes). In preliminary work Mayer et al.(2020), we have developed a measurement approach utilizing the RIPE Atlas framework – a network of more than 11,000 probes worldwide – to infer the risk of deanonymization for IPv4 clients in Germany and the US.

In this paper, we apply our methodology to additional scenarios providing a broader picture of the potential for deanonymization in the Tor network. In particular, we (a) repeat our earlier (2020) measurements in 2022 to observe changes over time, (b) adopt our approach for IPv6 to analyze the risk of deanonymization when using this next-generation Internet protocol, and (c) investigate the current situation in Russia, where censorship has been intensified after the beginning of Russia's full-scale invasion of Ukraine. According to our results, Tor provides user anonymity at consistent quality: While individual numbers vary in dependence of client and destination, we were able to identify ASes with the potential to conduct deanonymization attacks. For clients in Germany and the US, the overall picture, however, has not changed since 2020. In addition, the protocols (IPv4 vs. IPv6) do not significantly impact the risk of deanonymization. Russian users are able to securely evade censorship using Tor. Their general risk of deanonymization is, in fact, lower than in the other investigated countries. Beyond, the few ASes with the potential to successfully perform deanonymization are operated by Western companies, further reducing the risk for Russian users.

1. Introduction

Tor is the most notable anonymity network, used by two to three million people every day. A total of 6,500 voluntarily operated Tor relays advertise up to 700 Gbit/s of bandwidth, and provide anonymity by rerouting traffic via three Tor nodes. As a low-latency network, Tor is prone to traffic correlation attacks; thereby, a malicious actor must be able to observe the traffic between the client originating the connection and the first Tor node as well as the traffic between Tor's exit node and the destination. A global passive observer is capable to do so, but this form of an attacker is explicitly excluded from Tor's threat model. Yet, powerful observers exist, potentially threatening the anonymity of Tor users. Their capabilities are, however, not exactly clear. One reason for this is the theoretical assumption that the underlying Internet hierarchy is flat and evenly distributed. This is not the case, as the Internet is shaped in different tiers as well as various entities with different levels of control, e.g., Internet Exchange Points (IXP) with a high level of control and smaller Internet Service Providers (ISPs) with a lower level of control. Also, the Tor network does not utilize the Internet in an evenly distributed manner as the location of Tor relays is depending on various external parameters, e.g., economical (the price of bandwidth) or political (censorship, prosecution) reasons.

Prior work (Edman and Syverson, 2009; Feamster and Dingledine, 2004; Nithyanand et al., 2016) has shown that Tor traffic takes only a limited set of routes on the Internet. These studies,

* Corresponding author.
E-mail addresses: gabriel.gegenhuber@univie.ac.at (G.K. Gegenhuber), markus.maier@univie.ac.at (M. Maier), florian.holzbauer@univie.ac.at (F. Holzbauer), wmayer@sba-research.org (W. Mayer), gmerzdovnik@sba-research.org (G. Merzdovnik), edgar.weippl@univie.ac.at (E. Weippl), johanna.ullrich@univie.ac.at (J. Ullrich).

https://doi.org/10.1016/j.cose.2023.103302

however, rely on BGP updates and route prediction, and claim that measurements – despite being more reliable – would be infeasible due to lacking measurement nodes in the autonomous systems (ASes) that host Tor users, nodes, and destinations. With the introduction of the RIPE Atlas framework (Staff, 2015) – a global measurement network with more than 11,000 probes – this assumption no longer holds. In our preliminary work (Mayer et al., 2020), we developed a measurement methodology utilizing this network to actively probe the Tor network. In more detail, we used the probes to traceroute the Internet paths that are taken by Tor traffic and, based on the collected data, estimated the correlation potential of AS-level adversaries. In comparison to BGP-based approaches of path prediction, active measurements based on traceroute reveal how the packets are actually routed over the Internet. This provides a more realistic risk estimation for Tor users as BGP-based approaches are known to overestimate their risk (Juen et al., 2015).

The paper at hand is an extended version of our preliminary work (Mayer et al., 2020): We apply our methodology to three additional use cases creating an extended view on AS-level adversaries. In particular, we (a) repeat our measurements from 2020 to observe changes in Tor's service quality over time, (b) adopt our approach for IPv6 to analyze the threat of deanonymization when using this next-generation Internet protocol, and (c) investigate the current situation in Russia as censorship has been intensified since the beginning of its full-scale invasion of Ukraine, starting on February 24th, 2022. More specifically, the contributions of this paper are as follows:

- **Updated View on AS Interconnections.** By repeating our measurements from 2020, we investigate whether economical or political factors impacted Tor's service quality. Like in our previous measurements, we identified a few ASes with the potential to successfully deanonymize Tor users; although individual numbers vary over time, the overall picture has remained unchanged. According to our results, Tor provides anonymity at a constant quality to its users in Germany and the US.
- **AS-level Adversaries in IPv6.** We are the very first to conduct active measurements investigating the status quo of IPv6 in the Tor network. Despite the fact that the number of IPv6 Tor relays is smaller than their IPv4 counterparts, we could not identify an increased threat of deanonymization for clients using Tor over IPv6, neither in Germany or the US, nor in Russia.
- **Censorship in Russia.** With Russia's full-scale invasion in Ukraine, Russian state authorities also intensified Internet censorship, i.e., blocking media outlets reporting on the ongoing war. We investigated whether Russian clients evading censorship with Tor are prone to deanonymization, particularly when accessing blocked destinations in their geographic proximity.

The remainder of the paper is organized as follows: Section 2 provides background on Tor, and Section 3 discusses related work. Section 4 explains our measurement methodology. Section 5 provides our measurements' results which are then discussed in Section 6. We outline the limitations of our work in Section 7 and draw our final conclusions in Section 8.

2. Background

Tor was designed by Dingledine et al. (2004) in 2004 and soon became the most popular anonymity system. Tor's protocol specifications are open source and updated on a regular basis (Torproject, 2022b).

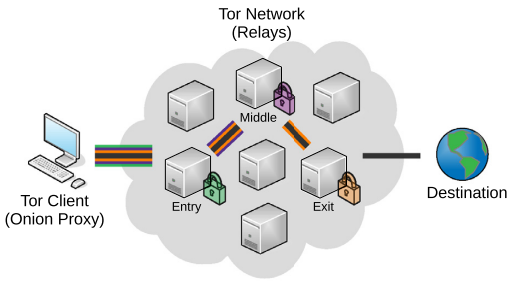


Fig. 1. Tor network. Traffic is relayed via three Tor nodes to hinder correlation of the client and the destination.

Functionality Tor is a low-latency anonymity network based on onion routing. It forms an overlay network of at least three relay nodes that are used to detour user traffic. The entrance to the Tor network is established by the onion proxy, also referred to as the *Tor client*. The proxy handles connections from user applications and is responsible for fetching the initial network information about the Tor network from a set of trusted directory servers. This information is then used to select Tor nodes for relaying. The first relay along a Tor path is called the *guard relay* – it is the only one that knows the client's IP address. The last one on the path is the *exit relay* which is the only one that knows the target IP address. The design of the Tor network is shown in Fig. 1.

Path selection For path selection, the onion proxy relies on information retrieved from the directory servers. The information includes relay flags and bandwidth information about Tor nodes. The exit node is selected first, then the guard relay, and finally the middle relay. The guard- and exit relays are selected randomly; however, the relays are weighted by their bandwidth. The middle relay is selected from the remaining set of nodes. To protect the users and maximize their anonymity, guard- and exit relays are reused according to a strict ruleset (e.g., guard pinning). Additionally, directory servers ensure that only nodes fulfilling certain uptime- and bandwidth requirements are selected as guard nodes. Another requirement for path selection is that the nodes have to belong to different /16 IPv4 prefixes. In reaction to new threat models, these rules are updated frequently, see also Section 3.

Deanonymization of users Tor's design makes it vulnerable to a global passive observer, which monitors all traffic going to and coming from the anonymity network. Such a global observer is explicitly excluded from Tor's threat model; however, powerful observers exist and threaten user anonymity. If an entity is able to monitor both the incoming and outgoing packets of a communication channel, it is able to correlate traffic entering Tor with traffic exiting the network based on timing. Our work precisely focuses on this threat and estimates probabilities of individual ASes appearing in a client's entry- and exit path.

IPv6 support Currently, Tor relays are either operated IPv4 only or Dualstack (i.e., providing an IPv4 and an IPv6 address). This way, Tor allows IPv6 traffic to enter and exit the network. Thereby, Tor relays can also act as bridges between IPv4 and IPv6. It should be noted that connecting from or to an IPv6 address reduces the set of possible relay candidates on the respective connection endpoint.

3. Related work

Feamster and Dingledine (2004) provided the first analysis of location diversity in the Tor network for independently operated ASes based on BGP routing tables. They analyzed the probability of an entry path to the network and an exit path from the net-

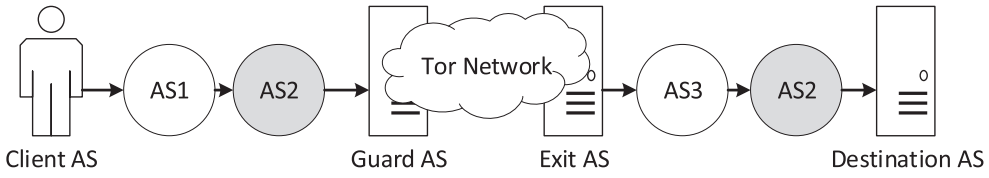


Fig. 2. Threat model. AS2 appears on the Tor entry path, between the client and the guard relay, and on the exit path, between the exit relay and the destination, and is thus in a position to perform traffic correlation deanonymizing the client.

work crossing through the same AS. Their analysis showed that previous methods of choosing paths/nodes based on IP prefixes are not sufficient to guarantee a diverse set of ASes, since there was a 10% to 30% chance, that both the entry and exit path to the mix network crossed the same AS. A refinement of this approach by Edman and Syverson (2009) showed that the previous study had even underestimated the potential threat. A study of Tor security properties against traffic correlation attacks was presented by Johnson et al. (2013). Their results showed that, depending on location, a user's chance of compromise can be at 95% within three months of monitoring against a single AS. One mitigation they proposed is to carefully select which entry and exit nodes to use. Wacek et al. (2013) built a graph of the Tor network to capture the networks' AS boundaries. Using this graph they provided an evaluation of a set of proposed relay selection methods and quantified their respective anonymity properties. Their results showed that bandwidth is an important property for the performance of such algorithms, and should not be neglected.

The importance of location diversity in the Tor network has been shown by several attacks proposed in recent years. Vanbever et al. (2014) provided a study of the capabilities of AS-level adversaries. Sun et al. (2015) described a set of advanced routing attacks on Tor, named *Raptor*. They also described the feasibility of asymmetric AS-level attacks by observing not only data traffic from the exit relay to the server but also TCP acknowledgment traffic on other routes which increases the capabilities of AS-level adversaries. Including the reverse path, they found 31.7% of the Tor circuits to be vulnerable in their measurements. However, paths had different probabilities to be selected by a client, and the actual number was likely to be lower. In 2016, Nithyanand et al. (2016) also used data on the Internet's topology (Giotsas et al., 2014) in a combination with AS-topology simulations (Gill et al., 2012) to estimate the threat posed by adversaries to Tor users. While previous attempts at the correlation of traffic (Hopper et al., 2010; Mittal et al., 2011) had very limited performance or required a large amount of captured traffic or time, *DeepCorr* (Nasr et al., 2018), developed by Nasr et al. greatly improved the feasibility of such attacks. By leveraging emerging learning mechanisms they managed to achieve drastically higher performance compared to existing state-of-the-art systems.

To mitigate the threat of AS-level adversaries that are able to correlate traffic and thereby monitor Tor users, various kinds of protection mechanisms have been proposed (Alsabah and Goldberg, 2016). Nithyanand et al. proposed *Astoria* (Nithyanand et al., 2016), an AS-aware Tor client. While similar in functionality to *LASTor* (Akhoondi et al., 2012), it provided improved protection with concern to threat models and attacker capabilities. Sun et al. (2017) presented a measurement study on the security of Tor against BGP hijacking attacks and presented a new relay selection mechanism to mitigate such attacks on Tor. In contrast to previous approaches, *DeNASA* from Barton and Wright (2016) provided a mechanism for AS-aware path selection independently of the destination. Additionally, they proposed another system for the creation of efficient and anonymous Tor circuits (Barton et al., 2018). Hanley et al. (2019) proposed an extension to the work presented by Sun et al. (2017) to increase the provided privacy and anonymity guarantees. Wan et al. (2019) showed that several attacks against a set of the proposed protections (Counter-RAPTOR, DeNASA, and LASTor) were still possible, but they also proposed simple solutions, which allowed to mitigate the threat posed by their developed methods. Rochet et al. (2020) introduced client-location-aware path selection (CLAPS) to overcome the pitfalls detected in previous path selection solutions (Counter-Raptor, DeNASA). They proved that based on the path selection of the earlier approaches the client's location can be revealed only after a few connections. Eaton et al. (2022) further enhanced the receiver side anonymity of Tor by introducing Private Information Retrieval (PIR) to hide which information is retrieved from the Hidden Service directory servers (HSDirs). Next to security, recent related work also focused on improving the Tor core. Jansen and Johnson (2021) estimated that the actual bandwidth of the Tor network could be much higher. They suggested a new measurement system for bandwidth calculation of Tor nodes. The authors found that with the current system the bandwidth self-measurements resulting in the *observed bandwidth* are rather imprecise.

4. Methodology

In the following section, we describe our method to measure strong AS-level observers, which are in a good position to conduct correlation attacks. As an overlay network, Tor depends on the underlying structure of the Internet. While often a flat hierarchy is assumed, it is clear that this is not the case. We can model the structure of the Internet by looking at autonomous systems identified by a unique AS number (ASN). One AS can be seen as an administrative entity that is responsible for a defined routing policy. Some AS are large and include a lot of Tor users, destinations, or relays, others do not contain users and destinations but are used for routing Tor traffic through the Internet and others are not important for Tor routing at all. Thus, some entities can observe more traffic than others.

With our measurement approach, we find a way to quantify which entities are in a stronger position. Figure 2 illustrates the basic idea of a standard traffic correlation attack, where one adversary (AS2) is placed on the incoming route to Tor as well as on the outgoing route to the destination. Sun et al. (2015) showed that it is also possible to correlate reverse-path traffic that may be routed differently. Other work already quantified strong adversaries with the help of BGP route updates. In contrast, we develop a method that utilizes the RIPE Atlas framework to actively acquire routing

Table 1
Tor relay statistics. While the number of relays increased, they are now spread among fewer ASes. The total Tor bandwidth increased by 66% over the past two years.

(a) Relays				(b) Diff. ASes			(c) Bandwidth (GBit/s)		
	2020	2022	Diff	2020	2022	Diff	2020	2022	Diff
All	6509	6559	*+1%*	1104	981	**-11%**	418	694	*+66%*
Exit	1000	1597	*+60%*	275	222	**-19%**	113	181	*+60%*
Guard	2415	2272	**-6%**	470	469	**-0%**	255	368	*+44%*

Table 2
IPv6 support statistics. As of Sept. 2022, 45% of the Tor relays support IPv6, while the exit bandwidth is 71% of the IPv4's one.

(a) Relays				(b) Diff. ASes			(c) Bandwidth (GBit/s)		
	All	IPv6	Share	All	IPv6	Share	All	IPv6	Share
All	6559	2924	45%	981	375	38%	694	342	49%
Exit	1597	1083	68%	222	94	42%	181	128	71%
Guard	2272	951	42%	469	175	37%	368	152	41%

information as this allows to study how packets actually travel the Internet.

The paper at hand extends our preliminary work; therefore, we apply our measurement approach to three additional use cases to gain a broader view of the potential of deanonymization in the Tor network. (a) We repeat the measurements and compare the state of 2020 with the current state (September 2022). (b) As IPv6 support at Tor relays has improved over the recent years (Torproject, 2021), we adapt our methodology to additionally acquire routing information for IPv6. (c) Lastly, we investigate a practical case study, Russia's full-scale invasion of Ukraine, and analyze AS-level packet routing by simulating access to websites that are blocked by Russian state authorities.

4.1. Relay AS diversity

As shown in Table 1, the Tor network currently (September 17th, 2022) consists of 6,559 relays. Only relays with the *Guard* flag (stable and reliable relays after a ramp-up phase (Dingledine, 2013)), are used as entry relays. Only relays configured to allow exiting traffic are potential exit relays in a Tor circuit. Because of the more stringent requirements, the number of guard and exit relays (with guard/exit probability > 0) is smaller than 6559. This also affects the AS diversity, which is the number of different ASes these relays are placed in.

Table 1 compares the metrics of Tor relay nodes for our two measurement snapshots in 2020 and 2022. Overall, the network has grown in terms of size and offered bandwidth. However, it has become more centralized, as for example, the AS diversity at exit relays has decreased by nearly 20%. Although the number of guard relays dropped by 6%, the number of bridges – providing an alternative and more anonymous entry to the network – nearly doubled (1,350 vs. 2,450) in the respective time period. Since a Tor relay node can – additionally to its IPv4 address – also offer an IPv6 address, we give an overview of the current IPv6 support in Table 2. With IPv6, the AS diversity drops by more than 60% making it an interesting target for our study.

Tor relays are chosen based on their flags and consensus weight. In Fig. 3, we show the AS diversity relation to guard and exit probability. We see that a small number of ASes have a large share of (a) guard and (b) exit probability. For IPv4, only five ASes control more than 50% of exit probability and 43 ASes have more than 90%. We also see that six ASes have a summarized guard probability of more than 50% and 131 have more than 90%. During our measurements in 2020, half of the exit probability was controlled by eight ASes and only four ASes dominated half of the guard probability. Therefore, the accumulated exit probabilities among top ASes has become even more centralized, while the guard probabilities are now slightly more diversified. For IPv6 the centralization is even worse, as only three ASes control more than 50% of guard resp. exit probability. Summarizing, Tor relays are distributed in almost 1000 ASes, the majority of entry and exit routing endpoints are however placed in a few ASes only.

Location diversity provides a similar picture: Two countries (Germany and the US) account for more than 47% of the relays. While the top five countries are still the same as in 2020, we noticed that Russia has lost a majority of its relays and has dropped from the sixth to the 18th rank (from 297 down to just 65 relays).

4.2. The RIPE Atlas framework

The RIPE Atlas framework is a highly distributed measurement network consisting of more than 11,000 available probes, deployed in over 3,600 different ASes. Regarding IPv6, it offers more than 5,000 vantage points (i.e., probes) in over 1,600 different ASes.

The measurement platform allows us to execute various low-level commands, e.g., ping or traceroute, on these probes and further processes the results. We will utilize this to execute traceroute commands from RIPE Atlas probes that are deployed in the same ASes as Tor guard- or exit relays, as well as clients and popular destinations.

Figure 3 also shows the cumulated guard- and exit probability for ASes that contain RIPE Atlas probes. From 222 ASes that contain exit relays, only 98 also contain a probe (837 relays out of 1597). Still, that makes approx. 43% of the total exit probability (35% with only 12 ASes). This differs from the cumulated guard probability. From 469 ASes that contain 2,272 relays, 249 ASes (with 1,723 relays) also include a RIPE Atlas probe, which represents guard relays with a sum of 80% guard probability (60% with 15 ASes). Especially for exit relays, these numbers could be drastically increased if only a few, exit-focused ASes would also host RIPE Atlas probes. Table 3 identifies ASes, that are currently not hosting any RIPE probes. By adding only five probes we could measure ASes with 81% exit probability in total and ten probes would reach 88% probability in total.

Figure 4 shows a bubble graph of current exit relays sorted by AS. The top five ASes that are not covered by RIPE Atlas (cf. Table 3) are marked in red (numbers 1–5). `AS208323 APPLIEDPRIVACY` which is represented by a green bubble (6) was missing RIPE Atlas coverage in 2020, but now hosts a RIPE

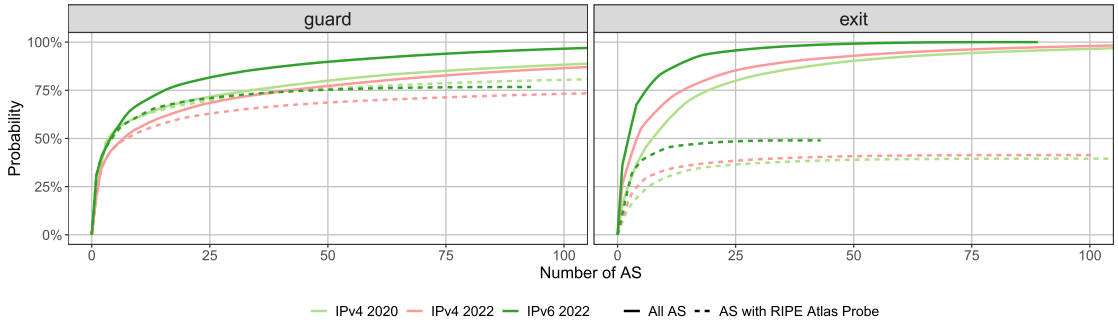


Fig. 3. Accumulated percentage of (a) guard, and (b) exit probability with the number of ASes. While RIPE probes cover 75% of guard probability, they cover less than 50% of exit probability.

Table 3
AS hosting Tor relays but no RIPE Atlas probe. Adding a single probe to AS60729 would increase the accumulated exit probability by 22.1%.

	AS	Name	Relays	Gbit/s	P_{exit}	P_{guard}
Exit	60729	ZWIEBELFR.	225	39.2	0.221	0.002
	205100	F3NETZE	32	11.9	0.084	0.000
	200651	FlokiNET	48	5.95	0.030	0.000
	62744	QUINTEX	100	6.77	0.026	0.000
	4224	CALYX	29	5.36	0.023	0.001
Guard	201814	SKYTECH	81	13.62	0.023	0.018
	46844	SHARKTECH	36	7.17	0.000	0.014
	19437	SS-ASH	10	2.51	0.000	0.006
	200303	LUMASERV	20	2.54	0.000	0.006
	264617	PANAGLOBAL	5	1.83	0.000	0.005

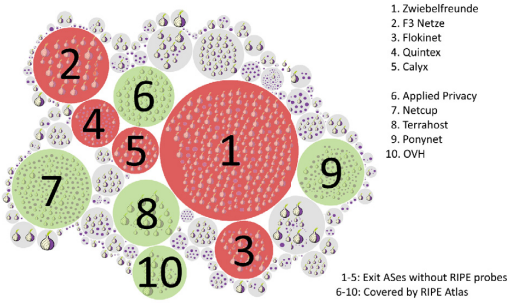


Fig. 4. RIPE Atlas probe coverage of ten large exit relay ASes. Adding a few probes to the exit relay ASes in dark red color (numbers 1–5) could significantly increase the coverage (Metrics, 2022).

probe. Other exemplary exit ASes that are covered by RIPE Atlas are also marked in green (numbers 7–10). Compared to 2020, the overall RIPE Atlas coverage of ASes that contain guard- and exit relays has not changed much (exit: 41 to 43%; guard: 83 to 80%).

4.3. *Active* traceroute *probing with RIPE Atlas*

As illustrated in Fig. 5 we perform *traceroute* measurements to identify routes taken for four different directions: (1) all client ASes to all guard ASes, (2) exit ASes with probes installed to the destination ASes, (3) destination ASes to all exit ASes, and (4) guard ASes with probes installed to the client ASes. With these measurements, we do not cover all possible routes since not all ASes have probes installed. For IPv4, depending on the direction, we measure at step (1) 100%, (2) ~43%, (3) 100%, and (4) ~80% in terms of route probability. For IPv6, we cover at step (1) 100%, (2) ~52%, (3) 100%, and (4) ~85% in terms of route probability.

In detail, this process works as follows:

1. Create the following sets:
   i. AS_{client} ... ASes of the clients (as chosen for the individual scenario, see Section 4.4)
   ii. AS_{guard} ... all ASes with guard relays
   iii. $AS_{guard \cap probe}$ ... all ASes with guard relays and RIPE Atlas probes
   iv. AS_{exit} ... all ASes with exit relays
   v. $AS_{exit \cap probe}$ ... all ASes with exit relays and RIPE Atlas probes
   vi. $AS_{destination}$ ... ASes of the destinations (as chosen for the individual scenario, see Section 4.4)
2. Generate ICMP traceroute measurement definitions for the following directions:
   (1) $AS_{client} \xrightarrow{traceroute} AS_{guard}$
   (2) $AS_{exit \cap probe} \xrightarrow{traceroute} AS_{destination}$
   (3) $AS_{destination} \xrightarrow{traceroute} AS_{exit}$
   (4) $AS_{guard \cap probe} \xrightarrow{traceroute} AS_{client}$
3. Execute the *traceroute* with the RIPE Atlas measurement API (''protocol'': ''ICMP'', ''response_timeout'': 20000, ''packets'': 1).
4. Process all results and look up the corresponding AS from the *ip2asn* database.
5. For every *traceroute*, mark all included ASes with the probability of that path being chosen, i.e., the corresponding guard/ exit probability.
6. Combine the values for directions 1 and 4 for the entry side, and 2 and 3 for the exit side, s.t., if an AS appears on either the forward or the reverse path it is assigned with the probability of that path being chosen. For multiple destinations, all traceroutes are combined.
7. Point out the top ASes, that appear on entry and exit side by looking at $P_{guard} \cap P_{exit}$.

4.4. *Origin and destination AS*

The goal of this work is to investigate multiple Tor scenarios for their proneness to deanonymization attacks. Therefore, we (a) repeat the measurements of our preliminary work in 2020 to observe changes over time, (b) adopt our approach for IPv6 to analyze the threat when using Tor over the next-generation Internet protocol, and (c) extend our measurements to investigate the current situation in Russia as censorship has intensified after its full-scale in-

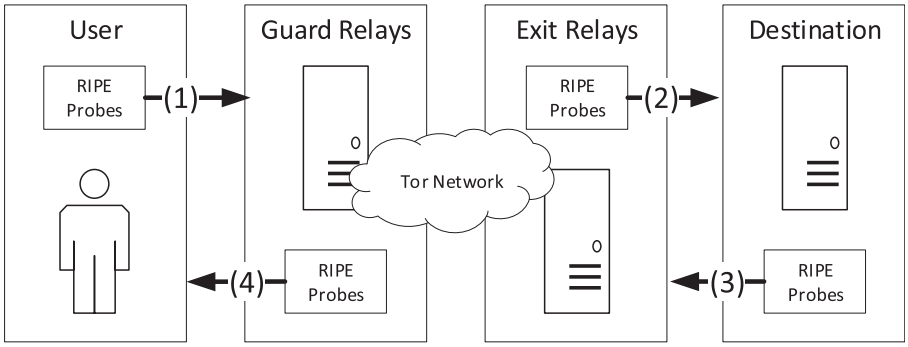


Fig. 5. RIPE Atlas *traceroute* scans. The forward path is covered by D_1, from client to guard relay ASes, and D_2, from exit relay ASes with probes to the destination AS. The reverse path is covered by D_3, from destinations to exit relay ASes with ripe probes, and D_4, from guard relay ASes with probes back to the client.

vasion of Ukraine, starting on February 24th, 2022. The following paragraphs describe how we derived the ASes for our client and destination data sets.

IPv4 measurements As mentioned in Section 4.1 Germany and the US account for nearly half of all Tor nodes. In our 2020 measurements, we have therefore chosen the ten ASes in Germany and the US containing most RIPE Atlas probes – an indicator of the AS's popularity in the respective country – for the client set C. For destinations, we derive the ASes from the Tranco (Pochat et al., 2019) top sites list. In particular, we take the 100 top-ranked domains, resolve the domain, and retrieve the corresponding ASes. We include only those ASes with deployed RIPE Atlas probe(s) in our destination set D. For our *traceroute* measurements, we select one RIPE Atlas probe for each AS in the client and destination set.

For the repetition of our measurements in 2022, we slightly adapted the approach in the following manner: In addition to the ASes inferred according to the just described procedure, we also included ASes that have been investigated in the first iteration (i.e., $C_{ipv4} = C_{2022} \cup C_{2020}$ and $D_{ipv4} = D_{2022} \cup D_{2020}$). For some cases, we were not able to gather updated results for ASes that have been measured in 2020. For example, `AS3356 LEVEL 3` was included in our destination set in 2020, but was not measured in 2022 as it does not host a RIPE Atlas probe anymore. Similarly, the 2020 client set contains historic ASes that do not exist nowadays (e.g., two consumer-grade ASes `AS6830` and `AS31334` were merged to `AS3209` – which is already present in our client set).

IPv6 measurements For clients, we again select ten ASes in Germany and the US with the most RIPE Atlas probes offering IPv6 support. For destinations, we increased the number of included domains from the Tranco list from 100 to 250 due to overall low IPv6 support. For comparison, we also included all ASes supporting IPv6 in the IPv4 datasets and vice versa.

Websites blocked by Russia Russian ISPs had started to block Tor in December 2021 (Xynou and Filastò, 2022b), i.e., three months before the beginning of Russia's full-scale invasion of Ukraine on February 24th 2022. Afterwards, Russia introduced even more rigorous censoring, blocking access to social media and independent news outlets (Migliano and Woodhams, 2022). Many of the blocked destinations are hosted either in Russia or Ukraine and report on the ongoing war. Circumvention of censorship is among the main goals of Tor, making Russia's full-scale invasion of Ukraine an interesting case study. Thus, we investigate whether users from Russian client ASes could be deanonymized when accessing these censored destinations in their geographical proximity.

For the client set, we again determine the ASes with the most RIPE Atlas probes in the respective country, i.e., Russia. For the destination set, we use a public list of websites blocked by Russian state authorities (Xynou and Filastò, 2022a) and rank the domains by popularity using the Tranco list. Then, we resolve these domain names and filter for ASes in Russia or Ukraine. Finally, we match our results with the RIPE Atlas deployment which determines the destination set for this measurement. As there were only two AS candidates supporting IPv6 within this data set, we refrained from a distinct IPv6 measurement in this particular case.

Summary of data set In total, we have eight data sets representing client ASes: 2020 IPv4 Germany, 2020 IPv4 US, 2022 IPv4 Germany, 2022 IPv4 US, 2022 IPv4 Russia, 2022 IPv6 Germany, 2022 IPv6 US, 2022 IPv6 Russia. Please note that there is no 2020 data set for Russia as the country was not included in our previous measurements. Beyond, we have four data sets representing destination ASes: 2020 IPv4 Tranco, 2022 IPv4 Tranco, 2022 IPv6 Tranco, 2022 IPv4 Blocked Websites. A detailed list of the ASes that are included in the data set is found in the Appendix A.

Our approach allows to measure Tor's entry and exit paths independently of each other, and to combine the results only in a successive processing step. Thus, it is sufficient to measure each of the data sets only once.

4.5. Data sources

To facilitate reproducibility and encourage openness, all used data files are publicly available at the project website.[1] In particular, our work relies on following data sources:

1. The Tor consensus, that contains all Tor relays with their IP addresses (IPv4, IPv6), associated flags (particularly "Guard" and "Exit"), advertised bandwidth, and guard and exit probability. We collect this information via the Tor network status protocol *onionoo*.[2]
2. Statistical data about the *RIPE Atlas probes*.[3] We use different data (e.g., id, number and AS of the probes) to find all probes connected to the same ASes as guard and exit relays.
3. Freely accessible *ip2asn*[4] databases to match IP addresses with the corresponding AS number.
4. Active RIPE Atlas *traceroute* results.[5]

[1] Project website: https://www.github.com/sbaresearch/ripe-tor
[2] onionoo: https://www.metrics.torproject.org/onionoo.html
[3] probes: https://www.atlas.ripe.net/probes/
[4] ip2asn: https://www.iptoasn.com/
[5] measurements: https://www.atlas.ripe.net/measurements

Table 4
Entry path and exit path probabilities for a single client and a single destination. HETZNER appears on the entry path due to its high number of entry relays as well as on the exit path due to hosting the destination.

	AS	Name	C	P	P_{relays}	P_{routes}	R
entry	1764	NEXTLAYER	○	1.00	–	1.00	468
	3356	LEVEL3	●	0.330	0.002	0.328	59
	24940	HETZNER	○	0.224	0.224	0.000	2
	16276	OVH	◐	0.131	0.130	0.000	2
	174	COGENT	●	0.123	0.002	0.121	110
exit	24940	HETZNER	○	1.00	-	1.00	220
	60729	ZWIEBELFR.	◐	0.221	0.221	–	1
	25291	INTERDOTL.	●	0.221	–	0.221	1
	47147	AS-ANX	●	0.162	–	0.162	2
	6939	HURRICANE	●	0.130	0.002	0.128	28

○ Client or Destination AS. ◐ Guard or Exit AS. ● Transit AS.

Table 5
AS with the potential for traffic correlation. For all measurements, HETZNER has the potential to deanonymize the client due to appearing on the entry and exit path.

		AS	Name	C	P_{guard}	P_{exit}	$P_{\&}$
2020	v4	24940	HETZNER	○	0.202	0.988	0.199
		1200	AMS-IX1	●	0.180	0.068	0.012
		16276	OVH	●	0.152	0.065	0.010
2022	v4	24940	HETZNER	○	0.224	1.00	0.224
	v6	24940	HETZNER	○	0.350	0.998	0.350
		6939	HURRICANE	●	0.087	0.393	0.034
		47147	AS-ANX ANE	●	0.107	0.223	0.024
		197540	NETCUP-AS	●	0.107	0.139	0.015

○ Client or Destination AS. ● Transit AS.

5. Evaluation

In this section, we discuss the results of our measurements: For readability, we first illustrate our approach in an exemplary measurement including a single client and destination AS only (see Section 5.1). Then, we focus on the full measurement discussing the ASes residing on Tor's entry paths (see Section 5.2), and those on the exit paths (see Section 5.3). Finally, we combine these results to infer ASes appearing on Tor's entry and exit path with high probability as they have the potential to perform traffic correlation deanonymizing Tor users (see Section 5.4).

5.1. Exemplary measurement: single client and destination

As an illustration of the capabilities of our methodology, we evaluate the results of measurements with a single fixed client AS and a fixed destination AS. herefore, we choose the AS of our research center $C = \{AS1764\}$ as client, and the AS of one mirror of the torproject.org website $D = \{AS24940\}$ as destination. In these ASes, we selected RIPE Atlas probes and scheduled 1,194 traceroutes, as defined in Section 4.3; out of which 1,177 (98.6%) were executed successfully (D1: 563/563, D2: 104/109, D3: 240/240, D4: 270/282).

Table 4 shows the results for IPv4; the ASes are grouped depending on whether they reside on a path towards a guard relay, or on a path from an exit relay. As expected, the client resp. destination AS (HETZNER, NEXTLAYER) is found in all traceroutes. Beyond, the ASes HETZNER, OVH and ZWIEBELFREUNDE appear in the tables; serving a high share of guard resp. exit bandwidth in the Tor network, the respective route is likely to be chosen as a Tor path (P_{relays}). However, we want to focus on intermediate ASes, that are different from those hosting relays as well as client/destination and appear on many routes. We identified LEVEL3, COGENT, and HURRICANE to be in a powerful position.

Eventually, we filter for intermediate ASes that have a probability of 1% or higher to appear on both sides, and only a single AS remains, namely HETZNER. With $P_{guard} = 0.224$ and $P_{exit} = 1.000$, it has a chance of $P = .224$ to deanonymize Tor traffic from our research center to torproject.org.

Table 5 provides an overview of the ASes with a probability of 1% or higher to appear on both sides for the three measurements 2020 IPv4, 2022 IPv4, and 2022 IPv6. A comparison of the IPv4 measurements reveals that the number of such ASes decreased; however, the probability of HETZNER increased by 2.5 percentage points. This means that the AS has now an even higher chance of deanonymization due to its increased guard probability. For IPv6-based traffic, this number is even higher. Because the set of possible guard relays decreases in IPv6, the guard probability of HETZNER increases once again by more than 10 percentage points. Beyond, there are three transit ASes in IPv6 with a $P_{\&}$ of up to 3.4%.

The case of HETZNER is particularly interesting as its chance of deanonymization arises from a distinct combination: On the one hand, it is the destination of our measurements; on the other hand, it hosts a high share of guard bandwidth and is thus more likely to be pinned as a guard node. This raises the question of whether path selection should include the destination AS to prevent such scenarios.

5.2. Tor entry: ASes between clients and guard relays

In the following paragraphs, we investigate the chance of ASes to be on a route to/from a guard relay and the chosen client ASes in Germany, the US, and Russia. For a total of 20 intermediate ASes, Fig. 6 shows their entry path probabilities as inferred from our measurements. The 20 AS were chosen according to the following rules: For every country, we select the 15 most likely intermediary ASes. We then show all intermediary ASes that occur for more than five clients in every measured country in our graph. For graphs that correspond to measurements in 2022, we also include ASes that were selected at the previous measurement period. Each data point represents one specific client AS. For a (transit) AS that is present in our measured routes to Tor guard relays, a data point in the graph denotes the summarized probabilities of all routes, i.e., the probability that this AS can trace packets from the client to the Tor network. On the right side of each row, we show the total number of data points. To visualize the range of the single data points, we draw a line between the minimum and maximum values. The figure allows a comparison among our three measurements (2020 IPv4, 2022 IPv4, and 2022 IPv6) as well as among the chosen countries.

In essence, the overall picture for IPv4 was confirmed, and the ASes with high entry path probability in 2022 remain the same as in 2020. For example, any client AS uses – though with varying probabilities – paths including AS174 COGENT, AS1299 TWELVE99, AS3356 LEVEL3. Yet, certain changes were observed: First, AS1200 AMS-IX1, AS12876 ONLINE S.A.S., and AS35807 SKYNET-SPB-AS appeared in the 2020 measurements, but not in the latest of 2022. In return, previously unknown ASes were seen (AS44530 HOPUS HOPUS, AS47147 AS-ANX ANEXIA). Second, AS2914 TT-COMMUNICATIONS-2 was frequently observed for US-based client ASes in 2020; nowadays, its probability is roughly comparable to the Germany-based client ASes.

Comparing IPv4 and IPv6, AS6939 HURRICANE is found more often on paths from US-based client ASes; in return, AS174 COGENT, AS1299 TWELVE99 and AS3257 GTT-BACKBONE are traversed less often. Beyond this fact, Tor paths of IPv4 and IPv6 appear to be highly similar, particularly for client ASes in Germany and Russia.

Local differences Most clients taking routes through high probability transit ASes are from the US: For example, AS6939

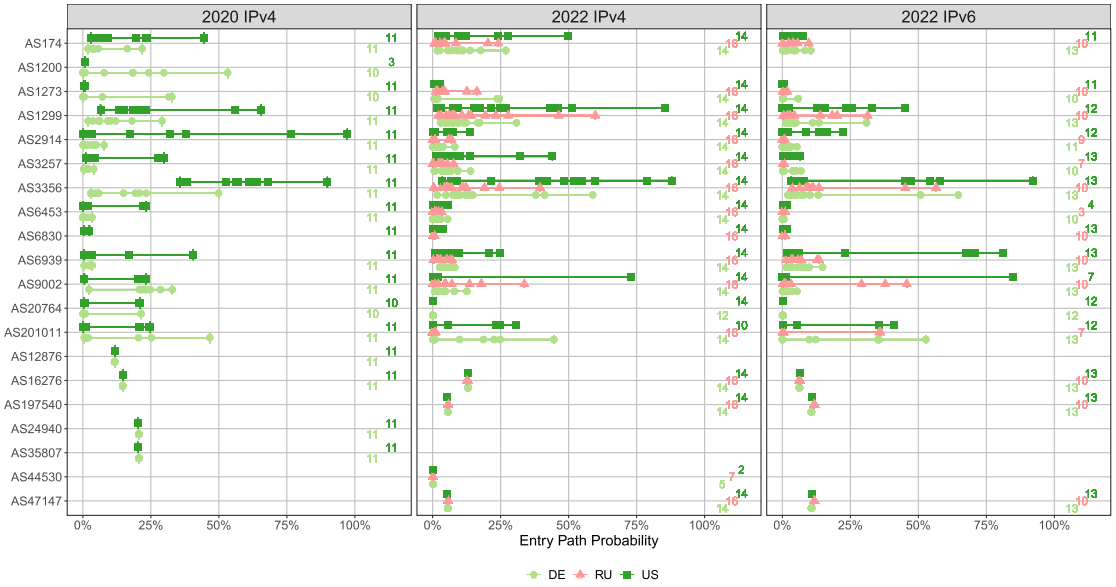


Fig. 6. Entry path probability describing the chance of an AS to appear between the client AS of three different countries and the guard relay. Each data point represents a client AS. The number on the right is the total number of data points.

HURRICANE is particularly dominant for IPv6 in the US, but plays only a minor role for German and Russian ASes. AS9002 RETN-NET plays a strong role in the US but has lower probability in Russia and Germany. This might indicate that routing in the US is more centralized than in the other countries. Beyond, it appears that ASes that are frequently found on paths from German client ASes, are also often seen on paths from Russian ASes; this might be a consequence of their geographic proximity.

While the sole presence of an AS on a path to/from a guard relay is not sufficient to conduct traffic correlation, it might however be sufficient to identify clients – and successively their users – connecting to the Tor network. Thus, the discussed results, covering Tor's entry side, also provide insights on which ASes are capable to detect clients using Tor.

5.3. Tor exit: ASes between exit relays and destinations

In the following paragraphs, we investigate the chance of ASes to be on a route between an exit relay and the chosen destinations for two distinct destinations sets: the Tranco List representing the most popular domains and those officially blocked by the Russian state-authority Roskomnadzor.

Tranco list For a total of 14 intermediary ASes, Fig. 7 shows the probability for the destination ASes that have been inferred from the Tranco list. For every destination we select all ASes that have a maximum probability of more than 20%. To filter for significant ASes we remove rows with a median value of less than 1% or less than five data points. For graphs that correspond to measurements in 2022, we also include ASes that were selected during the previous measurement period. Each data point represents a specific destination AS, and the figure allows a comparison among our three measurements (2020 IPv4, 2022 IPv4, 2022 IPv6). ASes that were selected because they are hosting a substantial amount of exit relays (e.g., AS60729 ZWIEBELFREUNDE) are marked with an asterisk.

For the AS that have already been seen in the 2020 measurements, we see a similar picture in 2022, and only minor changes are apparent: AS6461 ZAYO is barely seen anymore, and AS1200 AMS-IX1 is gone. The latter has also been identified for the entry side. Beyond, we found five new ASes with a considerable chance of being along the path. Comparing IPv4 and IPv6, we see that the maximum probabilities are typically lower for IPv6 for most ASes. Conversely, AS6939 HURRICANE has better chances to be on the path towards the destination, i.e., this AS appears to be a dominant player in the IPv6 Internet.

Destinations blocked by Russia Figure 8 shows the respective probability for the destination ASes that are blocked by the Russian state. As these websites have been predominantly blocked since the start of Russia's full-scale invasion of Ukraine, we do not have any data from 2020. We refrained from measuring IPv6 as only two of the candidate ASes were IPv6-ready. The ASes that are found towards these destinations are also found towards the Tranco List, with a single exception: AS3223 VOXILITY, an Internet infrastructure provider based in UK.

5.4. Potential ASes for traffic correlation

As a final step, we combine the results from Tor's entry paths, between client and guard relays, and exit paths, between exit relays and destinations. We calculate the probability that an intermediate AS is residing on both paths as the latter is the prerequisite to conduct a successful correlation attack deanonymizing the client.

Tranco list Our results are depicted in Fig. 9, providing the respective probability for the three measurements 2020 IPv4, 2022 IPv4, and 2022 IPv6, as well as the three investigated countries Germany, the US, and Russia. Each data point in a graph represents a transit AS that has both entry and exit path probability higher than 0%. For the entry path, we show data points for all relevant clients. For the exit path, we use the maximum probability

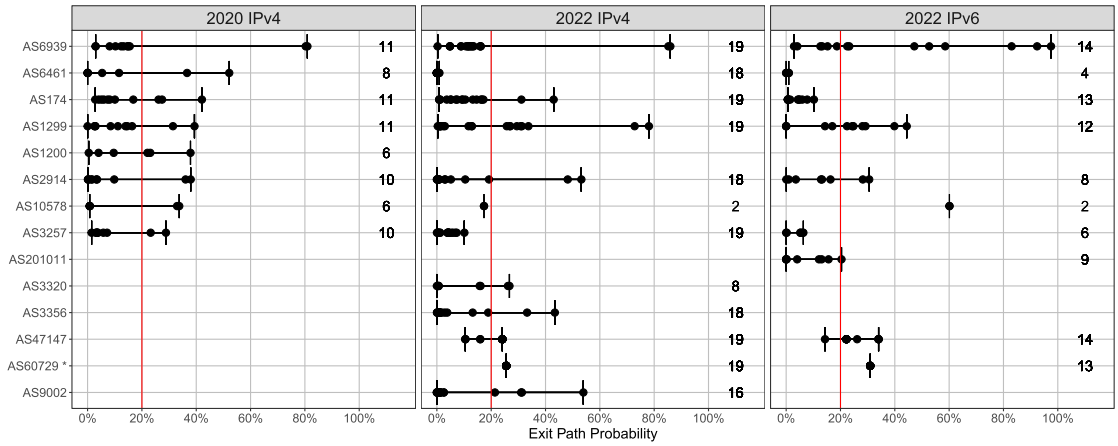


Fig. 7. Exit path probability describing the chance of an AS to appear between the exit relay and the Tranco list destinations. Each data point represents a destination AS. The number on the right is the total number of data points.

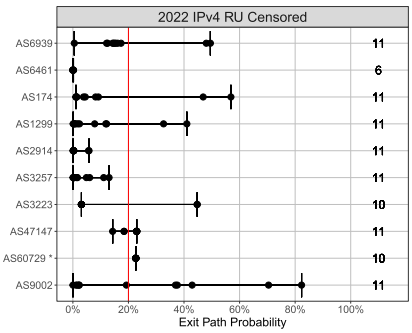


Fig. 8. Exit path probability for the ASes hosting websites that are blocked by Russia. The depicted ASes are all operated by Western companies (i.e., US, SE, UK, AT, DE).

of all measured destination ASes, which represents the worst-case scenario – i.e., an attacker has the best chance to correlate traffic when this target is visited by the Tor user.

In summary, AS24940 HETZNER is strong in all scenarios: First, it serves destinations and is thus likely to be on the exit side. Second, it hosts a high share of guard bandwidth, and is thus likely to serve as a guard relay, eventually appearing on the entry side. In combination, this leads to a high chance of being capable to correlate Tor traffic. As an exception, the measurement on the bottom left (2022 IPv6 DE) shows a reduced exit probability for AS24940 HETZNER. In this case, the selected measurement probe for scheduling traceroutes from AS24940 HETZNER to the Tor network – corresponding to (3) in Fig. 5 – ran into a timeout and did not yield any results. Although we also measure the same routes in the opposite direction – i.e., from the Tor network to AS24940 HETZNER, corresponding to (2) in Fig. 5 – this only covered about 50% of AS probability, due to the lack of RIPE Atlas availability in exit relay ASes (cf. Fig. 3).

Since 2020, the exit probability of AS3356 LEVEL3 has decreased substantially. With this, its overall chance for a successful attack decreased, both for German-based and US-based clients. Yet, this AS has still been considered relevant due to having a good probability to be found on the entry side, particularly in the US. In return, AS1299 TWELVE99 has increased its exit probability in this time span.

For IPv6, we see high chances for US-based traffic to be correlated: AS6939 HURRICANE stands out. It has high exit probability and also respectable entry probability at several client ASes. Beyond, AS3356 LEVEL3 is noteworthy because it has an excellent entry probability for specific client ASes. For both protocols, it appears that there are less chances to correlate traffic originating from Russian-based client ASes.

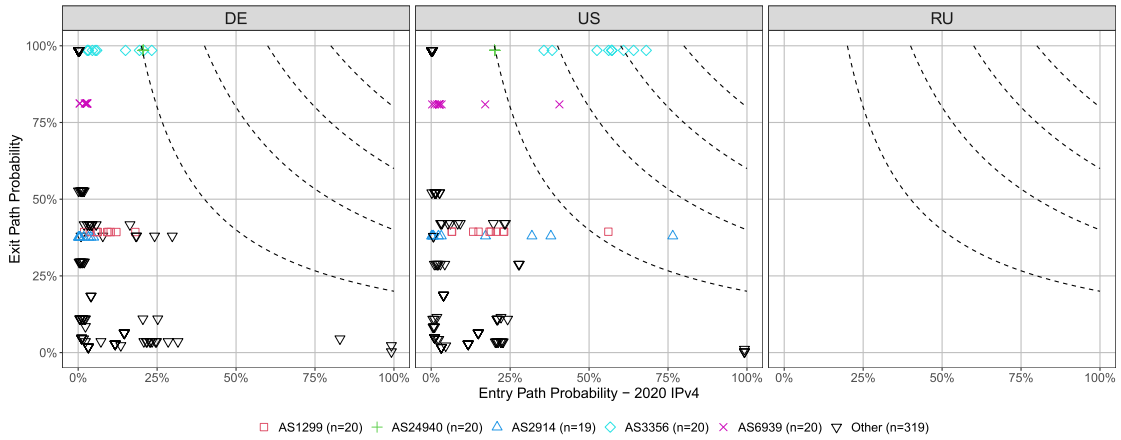
Destinations blocked by Russia The combined probabilities for the ASes hosting websites that are blocked by the Russian state are presented in Fig. 10. Again, the contour lines highlight data points at 20%, 40%, 60% and 80% combined probability. In this case, we see again that there are lower chances to correlate Russian-originating traffic than those from other countries. Although the client ASes (Russia) are within regional proximity to our destination ASes (Russia, Ukraine), the relevant transit ASes do not change much from our previous results. As an outlier, AS20764 RASCOM appears with an exit path probability of 52.3%. It is a consequence of the client AS simultaneously being the transit of the destination and appearing for a single client (AS20764 itself) only.

Consequently, we assume that a regional attacker (e.g., a nation-state) is not able to match entry- and exit packets of local Tor clients.

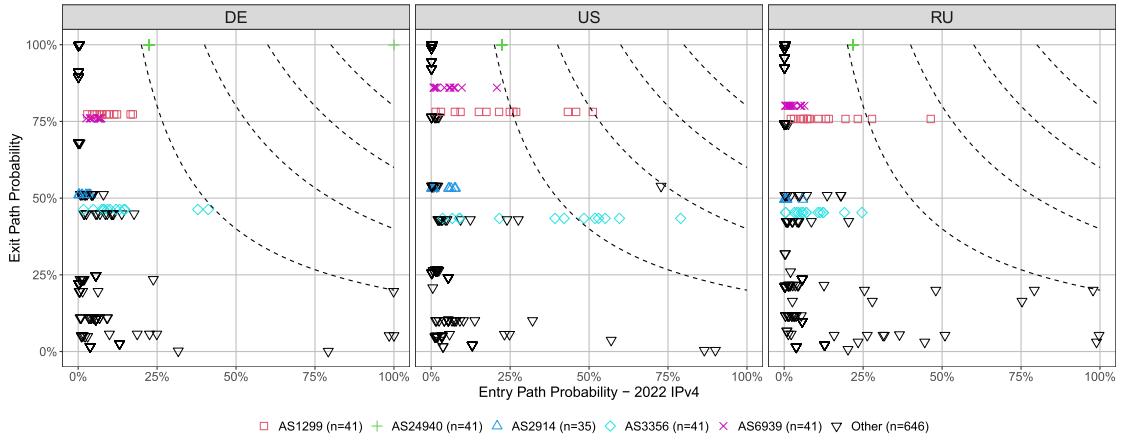
6. Discussion

Adversaries residing along the path to/from a guard relay and from/to an exit relay bear the potential to correlate traffic, thus defeating the very goal of the anonymization network Tor. In this paper, we applied our previously developed measurement methodology (Mayer et al., 2020) – capable to detect such potentially malicious players – to additional scenarios. In particular, we (a) repeated our measurements from 2020 to observe changes over time, (b) adopted our approach for IPv6 to analyze the threat when using this next-generation Internet protocol, and (c) extended our client- and destination sets to investigate the current situation in Russia where censorship intensified after its full-scale invasion of Ukraine, starting on February 24th, 2022.

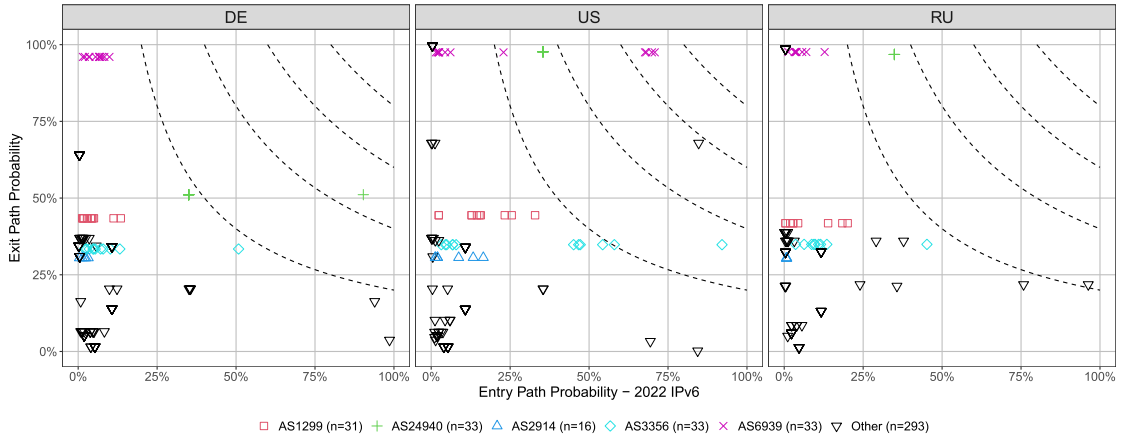
Development over time and protocols Our work does not provide any new impending AS-level adversaries. The probability of an AS to be on the entry side and/or on the exit side is – apart from a handful of changes – stable over time (2020 and 2022) and proto-



(a) 2020 IPv4



(b) 2022 IPv4



(c) 2022 IPv6

Fig. 9. ASes and their potential for traffic correlation for different years, protocols and countries for Tranco List destinations. Each data point represents an AS that appears on the entry and the exit path, and thus has the potential to perform traffic correlation. Contour lines at 20%, 40%, 60% and 80% highlight data points with highest combined probability.

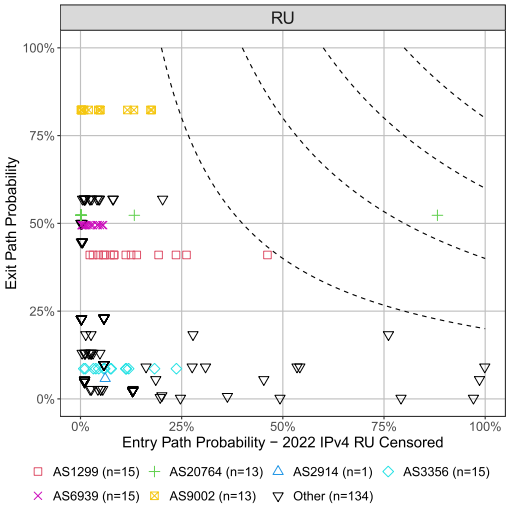


Fig. 10. ASes and their potential for traffic correlation for ASes hosting websites that are blocked by Russia. Each data point represents an AS that appears on the entry and the exit path, and thus has the potential to perform traffic correlation.

col versions (IPv4 and IPv6). This is good news: The Tor network and also the underlying routing structure of the Internet remain to a large extent stable. Tor is able to provide anonymity to users at a constantly high quality; however, targeted attacks against hand-picked combinations of clients and destinations in close proximity cannot be fully prevented (e.g., RASCOM). Beyond, it means that active measurements like ours do not necessarily have to be performed on a daily basis – longer intervals are fine, reducing the effort for measurements.

Division of roles Major transit ASes like HURRICANE or LEVEL3 are the prime suspects and are indeed capable of performing traffic correlation for many combinations of client and destination. In addition, we identified networks simultaneously serving multiple roles, which puts them in a good position for correlation attacks. For example, the data center operator HETZNER serves as a hosting provider for many destinations (e.g., major websites); at the same time, it hosts a high amount of guard relays. In total, they account for 22.4% of the guard bandwidth. This puts the AS in a favorable position for correlation attacks: The AS is likely to be part of a Tor path's entry side due to the many guard relays, and there is a high chance of it being included in the exit path due to the many hosted destinations. An operator of a HETZNER-based guard relay even found that 15% of the relay's traffic is forwarded to a relay within the same AS (Torproject, 2022a).

Ideally, guard relays should be – in network terms – close to the clients (e.g., in an ISP), and the exit guards close to the destination (e.g., in a data center), meaning that HETZNER would be a good candidate to operate exit nodes. We suggest to take this into account when deploying new guard- and/or exit relays, either as a private individual or an organization. An AS-aware circuit selection algorithm of Tor might also be beneficial but bears the risk that the chosen ASes allow to trace it back to the origin, see Section 3 on Related Work. Finally, we argue for increased AS diversity in the Tor network. Even with simple measurements, we see that the distribution of Tor relays is skewed. We hope that our measurements can improve an informed decision of how this diversity should be achieved.

Russia Since its full-scale invasion of Ukraine, Russian state authorities are blocking access to online information that is not in line with the official reports. This includes, among others, social networks, as well as local and independent media outlets. Censorship might be overcome using Tor, and our measurements show that the chance of deanonymization due to traffic correlation is low for Russian users. In fact, it is even lower than for users in Western democracies like Germany or the US (in which information censored in Russia is accessible anyway). Beyond, ASes that have the potential to perform successful correlation attacks are operated by companies in Western countries, further reducing the risk for Russian users. At the moment, however, the main challenge is to access Tor: Russian authorities aim to block guard relays, thus hindering the technology's use. The Tor community puts in a lot of effort to stay ahead of governmental blocking strategies (Dingledine, 2022).

Open source We publish our source code openly available. This enables other entities such as large relay operators to also perform measurements. All measurement results gathered through RIPE Atlas are openly available as well and could include valuable results for the Tor network. We argue that large relay operators should deploy RIPE Atlas probes in their networks, not only to further improve our (future) results but also to enable other measurements. Just a few more probes would increase the coverage significantly.

7. Limitations and future work

AS coverage Our traceroute measurements are limited by the current AS-level coverage[6] of the RIPE Atlas platform. While RIPE Atlas provides considerable coverage of a country's Internet users for Western countries (e.g., 92% in Germany and 86% in the US), its scope in illiberal or censoring states is often constrained. For example, the coverage, at the time of our measurements, was 26% in Russia, declining from 60% in 2020. Due to the current geo-politic situations and lacking alternatives, we nevertheless opted to for the inclusion of Russia as our case study. In comparison to Russia, China's Internet population is covered well by RIPE Atlas (83%), and renders it a candidate for further studies. Additionally, revisiting our measurements with increased IPv6 coverage and support among Tor relays could yield interesting results in the future.

Selection of client and destination ASes Since tracerouting all possible client and destination ASes was not feasible, we had to limit our measurements to a subset of ASes. The chosen AS sets are intended to reflect the reality best possible, i.e., the client sets should match ASes that contain actual Tor users and the destination sets destinations that are actually requested via Tor. A simple way to determine these ASes would be to capture traffic from (self-hosted) Tor relays; this, however, raises ethic concers due to snooping on Tor users and we used popular client and destination ASes instead. For our case study of Russia's full scale invasion of Ukraine, we used destinations that are blocked by the Russian regulator Roskomnadzor. We expect these destinations to be accessed via Tor as Russian Internet users cannot access them in a regular way; thus, we believe this destination set to be closer to reality than the others. Yet, there are no figures supporting this belief available.

Adversary granularity While this study specifically looks for adversaries at the granularity of ASes, there are other ways to group entities that could perform traffic correlation attacks. In some cases, organizations act as multiple ASes which means that the results (i.e., probabilities) of these ASes from our measurements have to be cumulated. Additionally, powerful nation states or intelligence agencies could force compliance of ASes within their jurisdiction to form an even more potent adversary. Finally, we executed a single traceroute for each AS pair to determine traffic

[6] https://www.sg-pub.ripe.net/petros/population_coverage/table.html

routes. Future research could provide more precise results by doing this in a more fine-grained manner, e.g., by measuring routes from different network prefixes or regions for every selected AS.

Simplified Tor model Our study is based on the traditional model of Tor covering only publicly known guard- and exit relays. In practice, Tor's architecture is constantly updated to cope with the ongoing censorship efforts of nation states like China or Russia. Therefore, Tor has introduced modular "plugable transports" (e.g., obfs4 bridges, Snowflake proxies) serving as relays which are not publicly known. This makes it harder to block these relays. We speculate that these add-ons could have positive effects on the AS distribution of the entry nodes (cf. *Division of Roles* in Section 6) due to being more lightweight, ephemeral, and easy to set up by inexperienced users (e.g., via a browser plugin). We consider this aspects to be part of future research.

8. Conclusion

We applied our measurement technology, which was developed in preliminary work (Mayer et al., 2020), to additional three use cases. This line of action allowed us to get a broader picture of current deanonymization attacks in the Tor network, and to infer actors with the potential to do so. In particular, we (a) repeated our measurements from 2020 to observe changes over time for users in Germany and the US, (b) adopted our approach for IPv6 to analyze the threat when using this next-generation Internet protocol, and (c) investigated the current situation in Russia where censorship has been intensified with the beginning of its full-scale invasion of Ukraine on February 24th, 2022.

We indeed identified a small set of ASes with the potential to perform deanonymization attacks. Most of them are large transit providers, but we also found an AS which simultaneously hosts high numbers of destinations and Tor guard relays. Hence, this AS has a high chance to appear on a Tor circuit's entry- and exit path, and consequently, successfully conducting traffic correlation to deanonymize individual Tor users. Once again, this exposes the problems of centralization and shows that there is room for improvement regarding the placement of guard-, and exit relays on the Internet. The former should be close to the clients, the latter close to the destinations.

While the numbers of individual ASes have changed since 2020, the overall picture does not reveal a significant change for Tor users in Germany and the US. Just as little does the protocol choice, i.e., IPv4 or IPv6, have a significant impact. We conclude that the Tor network provides anonymization to its users at a consistent quality. According to our results, Russian users are even less prone than Western ones to become deanonymized. Tor allows the former to securely access popular international websites as well as websites that have been censored. Beyond, the few ASes with the potential to perform successful deanonymization attacks are operated by Western companies, further reducing the risk for users in Russia.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Johanna Ullrich reports financial support was provided by Christian Doppler Research Association. Markus Maier, Florian Holzbauer, Johanna Ullrich, Georg Merzdovnik reports financial support was provided by Austrian Research Promotion Agency. Wilfried Mayer reports financial support was provided by Austrian Science Fund. Edgar Weippl is a member of the editorial board of this journal

CRediT authorship contribution statement

Gabriel K. Gegenhuber: Conceptualization, Methodology, Software, Writing – review & editing. **Markus Maier:** Data curation, Software, Validation, Visualization. **Florian Holzbauer:** Data curation, Validation, Writing – original draft. **Wilfried Mayer:** Conceptualization, Methodology, Software. **Georg Merzdovnik:** Supervision. **Edgar Weippl:** Supervision. **Johanna Ullrich:** Project administration, Funding acquisition, Writing – review & editing, Supervision.

Data availability

we have published the used source code and artifacts on github (referenced in the paper)

Acknowledgments

We want to thank David Schmidt for his preliminary work on this topic. This material is based upon work partially supported by (1) the Christian-Doppler-Laboratory for Security and Quality Improvement in the Production System Lifecycle; the financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association are gratefully acknowledged; (2) SBA Research (SBA-K1), a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the province of Vienna. The COMET Programme is managed by FFG; (3) Project 877110 2big2fail funded by the Program "BRIDGE1" (FFG); (4) Project DynAISEC FO999887504 funded by the Program "ICT of the Future" – an initiative of the Austrian Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology.

Appendix A. Client and destination AS sets

A1. Client sets

$C_{2020-DE-v4}$ = {AS3320, AS6830, AS31334, AS8881, AS3209, AS6805, AS553, AS680, AS8422, AS9145}

$C_{2020-US-v4}$ = {AS7922, AS701, AS7018, AS209, AS20115, AS22773, AS5650, AS20001, AS10796, AS11427}

$C_{2022-DE-v4}$ = {AS3320, AS3209, AS8881, AS6805, AS553, AS680, AS60294, AS24940, AS8422, AS9145}

$C_{2022-US-v4}$ = {AS7922, AS7018, AS701, AS209, AS20115, AS22773, AS5650, AS20001, AS47583, AS20473}

$C_{2022-RU-v4}$ = {AS12389, AS8402, AS25513, AS42610, AS35807, AS12714, AS3216, AS8359, AS12668, AS31200}

$C_{2022-DE-v6}$ = {AS3320, AS3209, AS8881, AS6805, AS8422, AS199284, AS60294, AS24940, AS8767, AS680}

$C_{2022-US-v6}$ = {AS7922, AS7018, AS701, AS47583, AS20473, AS62538, AS20001, AS209, AS22773, AS20115}

$C_{2022-RU-v6}$ = {AS42610, AS25513, AS202422, AS8331, AS12668, AS20764, AS50716, AS35807, AS12714, AS15974}

A2. Destination sets

$D_{2020-v4}$ = {AS3, AS15169, AS4837, AS24940, AS36351, AS14618, AS16509, AS14907, AS3356, AS7941}

$D_{2022-TRANCO-v4}$ = {AS15169, AS16509, AS8075, AS4837, AS14907, AS55990, AS37963, AS132203, AS4134, AS4812, AS47764, AS29169, AS14618, AS396982}

$D_{2022-TRANCO-v6}$ = {AS15169, AS16509, AS14907, AS47764, AS63949, AS3, AS37963, AS197695, AS32, AS14618}

$D_{2022-RU-CENSORED-v4}$ = {AS200350, AS15497, AS25532, AS207651, AS9123, AS28907, AS3326, AS197695, AS25521, AS12722}

References

Akhoondi, M., Yu, C., Madhyastha, H.V., 2012. LASTor: a low-latency AS-aware Tor client. Symposium on Security and Privacy. IEEE.

Alsabah, M., Goldberg, I., 2016. Performance and security improvements for Tor: A survey. ACM Comput. Surv. (CSUR) 49 (2), 1–36.

Barton, A., Wright, M., 2016. DeNASA: destination-naive AS-awareness in anonymous communications. Proc. Privacy Enhanc. Technol. 2016 (4).

Barton, A., Wright, M., Ming, J., Imani, M., 2018. Towards predicting efficient and anonymous Tor circuits. USENIX Security Symposium.

Dingledine, R., 2013. The lifecycle of a new relay. https://www.blog.torproject.org/lifecycle-new-relay.

Dingledine, R., 2022. How Russia is trying to block Tor. https://www.media.defcon.org/DEFCON30/DEFCON30presentations/RogerDingledine-HowRussiaistryingtoblockTor.pdf.

Dingledine, R., Mathewson, N., Syverson, P., 2004. Tor: The Second-Generation Onion Router:. Technical Report. Defense Technical Information Center, Fort Belvoir, VA doi:10.21236/ADA465464. http://www.dtic.mil/docs/citations/ADA465464

Eaton, E., Sasy, S., Goldberg, I., 2022. Improving the privacy of Tor onion services. In: Ateniese, G., Venturi, D. (Eds.), Applied Cryptography and Network Security. In: Lecture Notes in Computer Science, vol. 13269. Springer International Publishing, Cham, pp. 273–292. doi:10.1007/978-3-031-09234-3_14.

Edman, M., Syverson, P., 2009. AS-awareness in Tor path selection. In: Conference on Computer and Communications Security. ACM.

Feamster, N., Dingledine, R., 2004. Location diversity in anonymity networks. Workshop on Privacy in the Electronic Society. ACM.

Gill, P., Schapira, M., Goldberg, S., 2012. Modeling on quicksand: dealing with the scarcity of ground truth in interdomain routing data. ACM SIGCOMM Comput. Commun. Rev. 42 (1), 40–46.

Giotsas, V., Luckie, M., Huffaker, B., kc claffy, 2014. Inferring complex AS relationships. In: Internet Measurement Conference. ACM.

Hopper, N., Vasserman, E.Y., Chan-Tin, E., 2010. How much anonymity does network latency leak? ACM Trans. Inf. Syst. Secur. (TISSEC) 13 (2), 1–28.

Jansen, R., Johnson, A., 2021. On the accuracy of Tor bandwidth estimation. In: Hohlfeld, O., Lutu, A., Levin, D. (Eds.), Passive and Active Measurement. In: Lecture Notes in Computer Science, vol. 12671. Springer International Publishing, Cham, pp. 481–498. doi:10.1007/978-3-030-72582-2_28.

Johnson, A., Wacek, C., Jansen, R., Sherr, M., Syverson, P., 2013. Users get routed: traffic correlation on Tor by realistic adversaries. In: Conference on Computer and Communications Security. ACM.

Juen, J., Johnson, A., Das, A., Borisov, N., Caesar, M., 2015. Defending Tor from network adversaries: a case study of network path prediction. Proc. Privacy Enhanc. Technol. 2015 (2), 171–187.

Mayer, W., Merzdovnik, G., Weippl, E., 2020. Actively probing routes for Tor as-level adversaries with ripe atlas. In: IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, pp. 234–247.

Hanley, H., Sun, Y., Wagh, S., Mittal, P., 2019. DPSelect: a differential privacy based guard relay selection algorithm for Tor. Proc. Privacy Enhanc. Technol., 2019 (2).

metrics, T., 2022. Servers. Retrieved Sept. 19, 2022 from https://www.metrics.torproject.org/bubbles.html/as-exits-only.

Migliano, S., Woodhams, S., 2022. Websites blocked in russia since ukraine invasion. Retrieved Sept. 30, 2022 from https://www.top10vpn.com/research/websites-blocked-in-russia/.

Mittal, P., Khurshid, A., Juen, J., Caesar, M., Borisov, N., 2011. Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting. In: Conference on Computer and Communications Security. ACM.

Nasr, M., Bahramali, A., Houmansadr, A., 2018. DeepCorr: strong flow correlation attacks on Tor using deep learning. In: Conference on Computer and Communications Security. ACM.

Nithyanand, R., Starov, O., Zair, A., Gill, P., Schapira, M., 2016. Measuring and mitigating AS-level adversaries against Tor. Network and Distributed System Security Symposium (NDSS).

Pochat, V.L., Goethem, T.V., Tajalizadehkhoob, S., Korczynski, M., Joosen, W., 2019. Tranco: a research-oriented top sites ranking hardened against manipulation. Network and Distributed System Security Symposium.

Rochet, F., Wails, R., Johnson, A., Mittal, P., Pereira, O., 2020. CLAPS: client-location-aware path selection in Tor. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. ACM, Virtual Event USA, pp. 17–34. doi:10.1145/3372297.3417279.

Staff, R., 2015. RIPE atlas: a global internet measurement network. Internet Protoc. J. 18 (3), 2–26.

Sun, Y., Edmundson, A., Feamster, N., Chiang, M., Mittal, P., 2017. Counter-RAPTOR: safeguarding Tor against active routing attacks. Symposium on Security and Privacy. IEEE.

Sun, Y., Edmundson, A., Vanbever, L., Li, O., Rexford, J., Chiang, M., Mittal, P., 2015. RAPTOR: routing attacks on privacy in Tor. USENIX Security Symposium.

Torproject, 2021. The state of IPv6 support on the Tor network. Retrieved Sept. 30, 2022 from https://www.blog.torproject.org/state-of-ipv6-support-tor-network.

Torproject, 2022a. Are Hetzner servers in both the guard and middle position for a lot of Tor circuits? Observations from Hetzner traffic numbers vs. own monitoring. Retrieved Sept. 30, 2022 from https://www.forum.torproject.net/t/are-hetzner-servers-in-both-the-guard-and-middle-position-for-a-lot-of-tor-circuits-observations-from-hetzner-traffic-numbers-vs-own-monitoring/1851.

Torproject, 2022b. Tor protocol specifications. Retrieved Sept. 14, 2022 from https://www.gitweb.torproject.org/torspec.git/tree/.

Vanbever, L., Li, O., Rexford, J., Mittal, P., 2014. Anonymity on QuickSand: using BGP to compromise Tor. In: Proceedings of the 13th ACM Workshop on Hot Topics in Networks. ACM.

Wacek, C., Tan, H., Bauer, K.S., Sherr, M., 2013. An empirical evaluation of relay selection in Tor. Network and Distributed System Security Symposium.

Wan, G., Johnson, A., Wails, R., Wagh, S., Mittal, P., 2019. Guard placement attacks on path selection algorithms for Tor. Proc. Privacy Enhanc. Technol., 2019(4).

Xynou, M., Filastò, A., 2022a. New blocks emerge in russia amid war in ukraine: an OONI network measurement analysis. Retrieved Sept. 30, 2022 from https://www.ooni.org/post/2022-russia-blocks-amid-ru-ua-conflict/.

Xynou, M., Filastò, A., 2022b. Russia started blocking Tor. Retrieved Sept. 30, 2022 from https://www.ooni.org/post/2021-russia-blocks-tor/.

Gabriel K. Gegenhuber is a Ph.D. student at University of Vienna, Austria, Research Group Security and Privacy. Gabriel received a B.Sc. in Software & Information Engineering and an M.Sc. in Software Engineering & Internet Computing at the Technical University of Vienna. His research interests include mobile networks, network measurements, network security, and privacy-enhancing technologies.

Markus Maier is a Ph.D. student at University of Vienna, Austria, Research Group Security and Privacy. He received his B.Sc. and M.Sc. at Technical University of Vienna in Software Engineering & Internet Computing. His research interests include routing, network measurement and network security.

Florian Holzbauer is a Ph.D. student at University of Vienna, Austria, Research Group Security and Privacy. He received his B.Sc. and M.Sc. at University of Applied Sciences St.Pölten. His research focuses on Internet measurements. He detected flaws in email-related protocol adoption and is currently looking for flaws in IPv6 deployments.

Wilfried Mayer received a master's degree in Software Engineering and Internet Computing, and a doctoral degree in computer science at TU Wien. His research interests are focused on measuring privacy-enhancing technologies.

Georg Merzdovnik received a B.Sc. in computer engineering, an M.Sc. in software and information engineering, and a Ph.D. in computer science with distinction at TU Wien. Currently, he leads the research group on Systems and (I)IoT Security at SBA Research. Georg's research interests include applied systems and software security, IoT security (ranging from device to network level) as well as online privacy in general.

Edgar Weippl Edgar graduated with a Ph.D. from TU Wien. Afterwards, he was an assistant professor at Beloit College, WI, and a consultant for the software vendor ISIS Papyrus in New York, NY, Albany, NY, and Frankfurt, Germany. Returning to Vienna, he co-founded the research center SBA Research in 2004. In 2020, Edgar accepted a position as full professor at the University of Vienna.

Johanna Ullrich received a Ph.D. sub auspiciis praesidentis from TU Wien. Currently, she is a key researcher at SBA Research, Austria, leading the Networks and Critical Infrastructures Security Research Group, and a researcher of the Christian Doppler laboratory SQI. She was awarded the Research Prize of the Dr. Maria Schaumayer Foundation and nominated for the Hedy Lamarr Prize twice. Her research focuses on network security, particularly measuring experiments and IPv6.

5 Measuring Geoblocking in Commercial WiFi Calling Deployments

Publication Info

Title	Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi
Authors	Gabriel K. Gegenhuber, Philipp É. Frenzel, Edgar Weippl
Publication Status	This paper is included in the Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services (MobiSys), pp. 183–195, ISBN 979-8-4007-0581-6, 2024 CORE2023 Ranking: A.
DOI	`https://doi.org/10.1145/3643832.3661883`
Code Artifacts	`https://github.com/sbaresearch/scanywhere`
arXiv	`https://arxiv.org/abs/2403.11759`
Reference	[GFW24b]

Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi

Gabriel K. Gegenhuber
gabriel.gegenhuber@univie.ac.at
University of Vienna
Faculty of Computer Science
Doctoral School Computer Science
Vienna, Austria

Philipp É. Frenzel
pfrenzel@sba-research.org
SBA Research
Vienna, Austria

Edgar Weippl
edgar.weippl@univie.ac.at
University of Vienna
Faculty of Computer Science
Vienna, Austria

ABSTRACT

In current cellular network generations (4G, 5G) the IMS (IP Multimedia Subsystem) plays an integral role in terminating voice calls and short messages. Many operators use VoWiFi (Voice over Wi-Fi, also Wi-Fi calling) as an alternative network access technology to complement their cellular coverage in areas where no radio signal is available (e.g., rural territories or shielded buildings). In a mobile world where customers regularly traverse national borders, this can be used to avoid expensive international roaming fees while journeying overseas, since VoWiFi calls are usually invoiced at domestic rates. To not lose this revenue stream, some operators block access to the IMS for customers staying abroad.

This work evaluates the current deployment status of VoWiFi among worldwide operators and analyzes existing geoblocking measures on the IP layer by measuring connectivity from over 200 countries. We show that a substantial share (IPv4: 14.6%, IPv6: 65.2%) of operators implement geoblocking at the DNS- or VoWiFi protocol level, and highlight severe drawbacks in terms of emergency calling service availability.

CCS CONCEPTS

• **Networks** → **Mobile networks**; *Network management*; • **Security and privacy** → *Mobile and wireless security*.

KEYWORDS

geoblocking, telecommunication, roaming, cellular networks, mobile networks, VoWiFi, Wi-Fi calling, IMS, net neutrality, censorship, network measurements

ACM Reference Format:
Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. 2024. Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi . In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3643832.3661883

MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan

ACM ISBN 979-8-4007-0581-6/24/06.
https://doi.org/10.1145/3643832.3661883

1 INTRODUCTION

Mobile network services are a crucial lifeline in today's society, given that in 2023 over 5.4 billion people relied on cellular networks for connectivity and communication [44]. With 4G currently being the most used wireless standard and 5G rapidly gaining penetration, numerous operators are actively decommissioning older legacy networks (2G and 3G), marking the completion of the shift from circuit-switched to a comprehensive packet-switched network paradigm.

In the packet-switched domain, operators use VoIP (Voice over IP) based technology to terminate voice calls and messages. Additionally to the VoLTE (Voice over LTE) standard, VoWiFi (Voice over Wi-Fi, also known as Wi-Fi calling) was introduced. While VoLTE uses the traditional radio infrastructure that is provided by the operator as its access medium, VoWiFi is a complementary solution that allows the use of third-party wireless networks as an alternative uplink to the operator. Consequently, customers can leverage existing Wi-Fi access points (APs) and continue utilizing their mobile phones for voice calls in areas with poor or no cellular reception.

To support this functionality, operators need to expose parts of their infrastructure to the public Internet. This opens new possibilities for active measurement studies since it allows the investigation of exposed parts of a mobile network without requiring any radio equipment. Moreover, it allows measuring a huge number of international operators, without the need for sophisticated measurement hardware at the target locations.

Presumably, the general idea behind VoWiFi is to expand the cellular coverage to allow uninterrupted service e.g., in rural areas with weak reception. Thereby, a voice call can be handed over from VoLTE to VoWiFi, and vice versa, on the fly. However, VoWiFi can also be used completely independent from VoLTE, i.e., it requires no radio signal at all and also works e.g., when the mobile phone is in airplane mode but has Wi-Fi connectivity. In a mobile world that facilitates seamless transitions across national borders, it thereby can also be used overseas, possibly allowing customers to escape from

Figure 1: An Indian operator states in the FAQs [2] that VoWiFi cannot be used during international roaming.

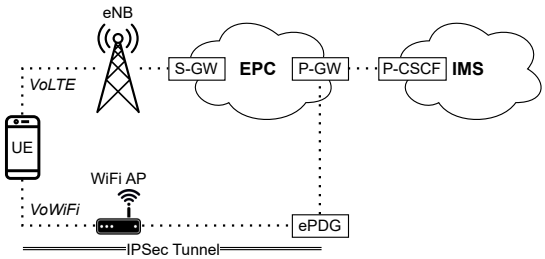


Figure 2: (Simplified) LTE network architecture for VoLTE and VoWiFi.

expensive roaming fees. In practice, some operators are actively denying their customers access to VoWiFi from foreign countries, as the screenshot in Figure 1 shows.

This paper aims to offer a comprehensive overview of the global deployment of VoWiFi and analyzes geoblocking measures of worldwide operators. More specifically, we use commercial VPN- and cloud services to simulate customers connecting from a diverse set of distinct foreign locations and to thereby determine geoblocking measures.

In summary, the main contributions of this paper are as follows:

- We propose a methodological approach to discover existing VoWiFi deployments and to probe them for IP-based geoblocking measures.
- We map the current VoWiFi support at worldwide operators, analyze the used infrastructure and give an overview of the latest global market penetration.
- We probe worldwide operators for IP-based geoblocking both at the DNS- and VoWiFi-protocol levels and provide an overview of current practices.

The remainder of the paper is organized as follows. Section 2 introduces the topic by providing background knowledge on the architecture and implementation of VoWiFi. In Section 3, we describe our methodological approach to discover and probe the VoWiFi infrastructure of global operators. Section 4 presents the results that were collected throughout this study and Section 5 briefly outlines related studies. Finally, we discuss our results and limitations in Section 6 and draw final conclusions in Section 7.

2 BACKGROUND

Figure 2 compares VoLTE and VoWiFi within a simplified cellular network architecture. For the sake of clarity, we've excluded several nonsubstantial components. Furthermore, we stick to the terminology that was specified in LTE, while later generations often introduced new names for similar components or services (e.g., VoLTE is called Voice over New Radio (VoNR) or Voice over 5G (Vo5G) in the fifth network generation).

***3GPP Access*, Voice over LTE (VoLTE).** As shown in the upper path of Figure 2, the User Equipment (UE) attaches to a base station (Evolved Node B (eNB) in LTE) via the Radio Acces Network (RAN). The Serving Gateway (S-GW) and the Packet Data Network Gateway (P-GW) within the Evolved Packet Core (EPC) system are responsible for assigning IP addresses to Access Point Names (APNs) and for routing and forwarding the traffic to external Packet Data Networks (PDNs). Finally, the Proxy Call Session Control Function (P-CSCF) acts as a Session Initiation Protocol (SIP) proxy and is the ingress point to the IP Multimedia Subsystem (IMS). All data traffic between the UE and the P-CSCF is encapsulated in an IPsec tunnel. The UE can then directly send SIP messages, e.g., after successful establishment of the IPSec tunnel it can send a *SIP REGISTER* request to the IMS core network. While SIP is used for signaling, the actual audio stream of a call is transferred via the Real-Time Transport Protocol (RTP).

***Non 3GPP Access*, Voice over WiFi (VoWiFi, Wi-Fi Calling).** In this scenario (lower path of Figure 2), the UE does not use the operator's RAN, but connects via an *untrusted* Wi-Fi Access Point (AP). More specifically, it establishes another IPsec tunnel to an Evolved Packet Data Gateway (ePDG) that is accessible via the public Internet. After successful authentication of the UE (via its IMSI and the cryptographic keys that are saved on the SIM card) and establishment of the IPSec tunnel between UE and ePDG, all traffic is forwarded to the IMS via the P-GW. Note that for VoWiFi, the SIP traffic is actually wrapped within two different IPSec tunnels (i.e., the first between the UE and ePDG, and the second between the UE and P-CSCF).

Internet Protocol Security (IPsec). As described above, VoLTE and VoWiFi heavily rely on IPSec [16] for authentication and traffic encapsulation. To set up a Security Association (SA) it uses the Internet Key Exchange (IKE) protocol. More specifically it uses IKEv2 [26, 27] with EAP-AKA [4] (Authentication and Key Agreement) for key derivation and thereby leverages the SIM card's secret keys to obtain a new session key. The negotiation can be divided into two phases: `IKE_SA_INIT` that negotiates the ciphering suite and other security parameters, and `IKE_AUTH` where the SIM card authenticates by solving a random challenge using its secret keys.

3 METHODOLOGY

VoWiFi calls are usually issued via domestic Wi-Fis, i.e., the customer's location can be inferred by looking at the client's IP address. For VoWiFi, the UE needs to communicate with at least two servers, as shown in Figure 3. After discovering the responsible ePDG IPs via DNS (1, 2), the UE establishes a secure connection to the ePDG (3, 4, 5) that will be used to terminate calls and messages.

There are multiple ways for an operator to block VoWiFi based on a subscriber's IP address:

DNS. Global companies often use GeoDNS (also GeoIP) to minimize network latency by pointing their clients to a geographically close server. Similarly, this can be used for geoblocking, by configuring the responsible DNS server to ignore queries that stem from unwanted client countries. Due to caching and the recursive manner of DNS, this method is relatively inaccurate and might nevertheless leak IP addresses. Finally, there are relatively easy anti-blocking techniques, e.g., skillful customers can use custom DNS servers or manually add host entries to circumvent geoblocking.

ePDG. To achieve more effective blocking, operators can also implement measures at the ePDG. As an example, an operator could

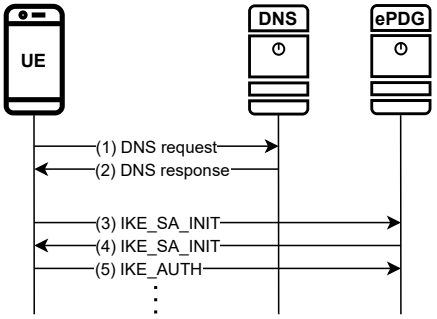


Figure 3: To connect via VoWiFi, the client fetches the ePDGs IP address via the DNS server and starts the IKE negotiation with the ePDG.

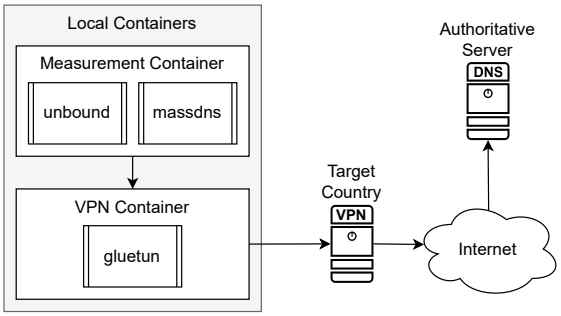


Figure 4: A containerized architecture isolates independent measurements and enables parallel execution for simple upscaling.

simply deploy firewall rules on their ePDGs to drop packets from IPs that do not belong to the domestic country. Additionally, operators could develop more complex rules that only permit specific *premium users* (identified by their IMSI), or also check the latest roaming status (via radio) of a subscriber before deciding whether to block the connection. To achieve worldwide coverage in our study, we decided to focus on straightforward IP-based rules because it does not require the acquisition of SIM cards of the tested operators.

To allow structured testing, our methodological approach is divided into two steps: (i) mapping DNS records and (ii) probing the actual servers.

3.1 Mapping DNS Records

The Fully Qualified Domain Name (FQDN) of an ePDG is specified in 3GPP TS 23.003 [12] and can be built from an operator's Mobile Country Code (MCC) and Mobile Network Code (MNC):

```
epdg.epc.mnc{y}.mcc{x}.pub.3gppnetwork.org
```

According to the specification, the MCC (x) always consists of three decimal digits, while the MNC (y) can be either two or three decimal digits. The first digit of the MCC is allocated according to the geographic region of the operator and thereby easily allows to differentiate operators based within different continents e.g., Europe or North America.

If we want to get an exhaustive list of ePDGs from all operators around the globe, requesting the IP addresses via normal (recursive) DNS requests (i.e., offloading them to popular DNS servers like Cloudflare or Google) from one central location does not work, or would at least yield imprecise or non-deterministic results (e.g., due to caching, Anycast routing, etc.). To make recursive DNS servers query for geographically close IP addresses, the eDNS Client Subnet (ECS) [7] mechanism allows propagating the client's IP address range (usually a `/24` subnet) to the authoritative DNS server. However, some DNS servers (e.g., Cloudflare [6]) do not enable ECS due to privacy concerns. Also, we found several operators' authoritative DNS servers do not support ECS and therefore solely use the request's IP address as baseline to build their responses. Addressing this requires a more complex approach, that uses different locations worldwide to issue DNS requests in an authoritative manner.

To cope with these needs we use a containerized infrastructure that leverages Virtual Private Networks (VPNs) for global distribution of DNS requests. Figure 4 describes the approach in detail: the VPN container connects to an existing VPN, providing Internet connectivity to the measurement container. The measurement container runs a local *unbound*[1] resolver that is configured to resolve DNS requests in an iterative manner (i.e., to get the IP addresses from the authoritative server). All DNS requests to the local server are issued with *massdns*[2].

The VPN container is built upon *gluetun*[3]. Thereby our setup works with any *OpenVPN* or *WireGuard* server and already implements native support for many consumer-grade VPN services (e.g., ProtonVPN, NordVPN, CyberGhost). To quickly achieve broad coverage, we purchase several VPN services and additionally use boto3[4] to implement automatic generation of ephemeral Amazon EC2 instances that are spawned in all available AWS Regions and act as WireGuard-based relays. Table 1 provides a summary of the utilized services along with the corresponding number of countries advertised by each service.

3.2 Discovering DNS Records

To discover existing ePDGs, we simply construct a list of all possible MCC and MNC combinations, resulting in 1.1 million domain names. We use the presented infrastructure to resolve these domain names (`A` and `AAAA` entries) from globally distributed vantage points. For all `CNAME` responses, we iteratively resolve the referenced domain until a final response is returned.

After retrieving the corresponding `A` and `AAAA` entries for a domain we can generate an exhaustive list of all ePDGs. For entries that only occur within specific client countries, we derive that the operator is possibly using DNS for load balancing, to reduce latency, or as a geoblocking measure.

[1] https://github.com/NLnetLabs/unbound
[2] https://github.com/blechschmidt/massdns
[3] https://github.com/qdm12/gluetun
[4] https://github.com/boto/boto3

Service	Countries[a]	IPv6 Support
Amazon EC2 (Cloud)	23	✓
Cloudflare WARP	120	✓
CyberGhost	91	×
hide.me	50	✓
HideMyAss	210	×
IVPN	36	✓
Mullvad	43	✓
NordVPN	60	×
Private Internet Access	84	×
ProtonVPN	68	×
Surfshark	100	×

[a] As advertised by the VPN/cloud service.

Table 1: Overview of VPN and cloud services that were used to distribute our measurements across the globe.

3.3 Probing Servers via IKE Initialization

To scan for IP-based geoblocking at the ePDG server, we can simply leverage the containerized infrastructure that was used to distribute our DNS requests in the previous section. Within the measurement container, we run a Python script that iterates over all IP addresses that were discovered in the previous step, trying to do an `IKE_SA_INIT` exchange (step 3, 4 in Figure 3). The script logs its current public IP address and whether the ePDG servers respond to the sent initialization packet. Any server that does not answer the first packet is probed repeatedly (i.e., five times) with an added back-off period. After querying a server's status from different source IP addresses we are able to determine whether the operator drops requests that are issued from unwanted countries.

The second phase (step 5 onwards in Figure 3) of the IKE protocol requires additional parameters like a subscriber's IMSI and cryptographically signed challenges that prove the identity of the customer. Since we want to get a big picture of global geoblocking practices and because it is not feasible to get access to SIM cards of a considerable amount of all worldwide operators we focus on detecting geoblocking relying on simple rules, such as dropping packets from unwanted IP addresses.

Therefore, the results of our methodology provide a lower bound on the number of operators that deploy geoblocking for VoWiFi. While our results in Section 4 show that we're able to detect a great share of blocking operators, we outline potential factors that restrict the detection capabilities of our method in Section 6.1.

4 RESULTS AND EVALUATION

We started with some preliminary exploratory measurements in May 2023 and subsequently improved our measurement methodology. The majority of our measurement results were obtained within a condensed measurement campaign during July and August 2023 (DNS discovery: Jul 13th to Aug 15th, IKE probing: Jul 13th to Aug 22th).

When citing a particular operator, we employ the notation $\texttt{CarrierName}_{[MCCMNC]}$.

As explained in Section 3.1, we issue DNS queries for all possible domain names from multiple clients distributed worldwide. From an abstract perspective, only a single DNS request is required to find the IP addresses that are currently associated with a domain name (from a particular location). In practice, however, this theoretical assumption does not hold. In fact, first of all, a client needs to find the responsible authoritative nameserver by querying the root and Top Level Domain (TLD) servers before the actual query is sent. Additionally, when asked for an `A` or `AAAA` record, the authoritative nameserver can refer to another domain by returning a `CNAME` entry. To cover this, we subsequently resolve `CNAME` chains until a final response (i.e., either `A`/`AAAA` or `NXDOMAIN`) is reached. Lastly, we enable rigorous caching at our local *unbound* instance, to prevent repeated queries and lower the amount of actual requests issued to external servers.

We experienced that most authoritative nameservers return a complete list of all IP addresses that are assigned to the requested domain name. In contrast, some nameservers only answer with a single IP and withhold the rest of the addresses that are also assigned to the requested domain name. While it is relatively easy to request all IP addresses for the first case, the latter complicates the matter and can only be tackled by querying the desired resource over and over again.

Lastly, some nameservers are configured to return IP addresses deterministically, depending on the source of the request.

In our approach, we split the requests into two phases (executed consecutively at every location), retrieving all available `A` and `AAAA` records respectively.

In addition to the original strategy where we query for all possible hostnames (*domain discovery*), we also ran some instances only querying domains already discovered in previous rounds. This was done to reduce the overall number of queries necessary to find all IP addresses (*IP discovery*) that exist for a single domain.

Overall, we ran 8,555 domain discovery and 47,902 IP discovery scans that were distributed across 219 countries.

Collected IP Addresses. Table 2 shows the amount of collected domains and (distinct) IP addresses. Overall, we collected 1,026 (`A`) and 66 (`AAAA`) domain-to-IP mappings. However, many IP addresses occur multiple times within one country, e.g., when an operator occupies multiple MNCs or when a Mobile Virtual Network Operator (MVNO) uses the ePDG of its parent provider, which reduces the set to 725 (`A`) and 40 (`AAAA`) unique ePDG IPs. About 7.3% of all domains support dual-stack (i.e., they have both an `A` and `AAAA` entry). For all found `AAAA` records, there is also a corresponding `A` entry, i.e., there is no operator that runs the ePDG via IPv6 only.

The vast majority of providers use three digits for the MNC in their ePDG domain, while $\texttt{Vodacom}_{[64004]}$ (Tanzania) is the only one that reserves an additional two-digit domain (i.e., `epdg.epc.mnc04.mcc640.pub.3gppnetwork.org`), serving the same IPs as its three digit counterpart.

Geographic Location of ePDGs. To determine the country of origin for an MNO (by its MCC), we rely on the most recent version of Android's MCC table [1], which is based on T-SP-E.212A [25] standardized by the International Telecommunication Union (ITU). To geolocate IP addresses we use the free MaxMind GeoLite2 database[5]. The MCC country of an ePDG and the location of an ePDG

[5] https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

	IPv4			IPv6		
Region (via MCC) [a]	Countries [b]	Domains	IPs	Countries [b]	Domains	IPs
0 Test networks	0	0	0	0	0	0
2 Europe	41	148	311	8	9	16
3 North America & Caribbean	20	69	127	2	2	7
4 Asia, Middle East	21	133	164	3	17	11
5 Australia, Oceania	9	32	60	1	1	3
6 Africa	8	14	22	1	1	2
7 South- & Central America	9	26	40	1	1	1
9 Worldwide [c]	1	1	1	0	0	0
Total	109	423	725	16	31	40

[a] digits 1 and 8 are not specified. [b] according to the MCC. [c] Satellite, Air, Maritime, Antarctica.

Table 2: Encountered ePDGs via DNS discovery, grouped by MCC region. For *Test networks* we found no public DNS entries.

according to its IP addresses do not necessarily need to be identical. However, the majority of the operators use ePDGs that are hosted within their own network range and country. In most cases where the ePDG is not located within the MCC country, it is hosted within close proximity (i.e., in a neighbouring country). We noticed that this practice is especially popular with relatively small countries (e.g., `Tele2`$_{[24603,24702]}$ in Latvia and Lithuania via Sweden, `Orange`$_{[27099]}$ in Luxembourg via Belgium or `Claro`$_{[74810]}$ in Uruguay via Argentina) and in Caribbean island countries (e.g., `Flow`$_{[365840,364390]}$ in Anguilla and the Bahamas via Jamaica). An outlier with no geographical proximity of MCC and ePDG country is `Tata Communications`$_{[23427]}$ which is based in the United Kingdom but uses an ePDG located in the United States. `MTX Connect`$_{[90139]}$, the only operator we found within the "Worldwide" MCC range, is pointing to an ePDG IP hosted in Luxembourg. For IPv6 we do not see any divergences between MCC and ePDG country.

Non-Routable IP Addresses. While the majority of returned IP addresses lie within public address ranges, some results are not publicly routable. For example, German `Voiceworks`$_{[26220]}$ and Italian `Wind Tre`$_{[22288,22299]}$ return loopback IP addresses (`127.0.0.1` and `127.0.0.9`). Obviously, these addresses will not work in practice and were presumably just deployed as a placeholder or for internal testing purposes.

For IPv6, we see several providers referring to their IPv4 siblings. More specifically, `Three Mobile`$_{[23594]}$ (United Kingdom) and `Méditélécom`$_{[60400]}$ (Morocco) use the NAT64 IPv6 transition mechanism [38] via the `64:ff9b::/96` prefix and `Maxis`$_{[50212]}$ (Malaysia) refers to its IPv4 sibling via an IPv4-Compatible IPv6 address (deprecated in RFC4291 [9]).

4.1 Analyzing Differences in DNS Responses

We experience several cases where the returned DNS results deviate for repeated queries from different locations. For every IP address that is returned for a specific domain, we inspect the set of countries from which we were able to discover this IP address. Additionally, we also count the number of occurrences per country.

By inspecting these results, we can group the DNS servers according to their characteristic behavior:

(G1) *Returning Multiple IPs for Redundancy.* This group contains all domains where the DNS server directly returns all IP addresses that are associated with the queried hostname (without differentiating by the client's location). The vast majority of all domains (at least 78%[6] of all IPv4 domains) show this behavior. Responding with a greater set of IP addresses makes sense from an availability perspective: a client that wants to connect to a service can always switch to another IP address if something goes wrong when connecting to the first endpoint. Additionally, due to round-robin DNS [5], this will also enforce automatic load-balancing across all associated IP addresses.

While the median number of IP addresses that are returned is only two, some operators return a greater number of IPs. For instance, the maximum number that we discovered was `Telekom`$_{[26201]}$ (Germany) where all 12 IPs that exist for their ePDG domain are instantly returned by the responsible DNS server.

(G2) *Using DNS for targeted Load-balancing.* Some operators do not disclose all IP addresses in every DNS answer, but just return a subset (often just a single entry) of their responsible IP address pool. In this case, every request receives a partition of the address pool without any specific bias or determination (regarding the request's source location). Therefore, we assume that this is just used as a load-balancing measure, e.g., to balance clients among existing servers and that there is no intention to use this to block any resources. Additionally, the operator could use this for A/B testing or more fine-grained balancing based on the current network status, e.g., by purposefully forwarding clients to servers that currently have free resources.

`T-Mobile`$_{[310240,310260]}$ (United States) stands out as the primary example within this category, as they provide a total of 39 IP addresses for each of their domains. All ePDG servers were in deployment concurrently (i.e., discovered during various scans in the same time period) but their DNS server only answers with a single IP per request, as described above. Supposedly, the returned IP is selected randomly, as we do not see any location bias.

[6] According to our simple heuristic algorithm.

	IPv4		IPv6	
Service	Countries	Measurements	Countries	Measurements
Amazon EC2 (Cloud)	21	2,456	22	2,212
Cloudflare Warp	208	8,934	208	7,417
CyberGhost	90	4,025	0	0
hide.me	49	1,994	46	1,641
HideMyAss	207	2,969	0	0
IVPN	36	3,975	34	791
Mullvad	34	1,930	33	1,538
NordVPN	59	2,166	0	0
Private Internet Access	83	5,337	0	0
ProtonVPN	68	3,801	0	0
Surfshark	100	4,562	0	0
Total	219	42,149	208	13,599

Table 3: To achieve global coverage and to improve diversity of our client-side vantage points, we evenly distributed the IKE probing measurements to numerous VPN- and cloud services. Surprisingly, Cloudflare Warp provided far more countries than advertised in the service description (120 advertised vs. 208 actual countries). While they only list locations of their data centers the service presumably also uses smaller edge locations as exit points.

(G3) *Using DNS for Geolocational Grouping.* Again, only a single entry of a bigger address pool is returned. Additionally, the returned address is determined by the source IP address. Analyzing the IPs returned for specific countries, we see that the operator uses predetermined IP addresses to serve customers in different locations. For example, `Reliance Jio`$_{[405874]}$[7] (India) clearly separates their domestic and foreign users. All queries issued within India received IP addresses that never occurred within any other country. For their external customers, we could not see any further differentiation (i.e., customers in Europe usually get the same responses as customers in America or Africa). Additionally, we noticed that their DNS server does not support the eDNS Client Subnet (ECS) [7] mechanism. Therefore, the only way to discover their local IP addresses is to issue DNS queries from a location within India.

Regular use cases for this behaviour include its utilization for structural grouping of customers based on their location, or reducing latency by pointing to geolocationally close servers that are close to the customers (GeoDNS). Additionally, this behavior could be used to block the resource by returning an IP that does not accept any connections from the requester's location. To check whether this is the case, we need further analysis, i.e., we need to check whether the ePDG response differs between all returned IP addresses (cf. Section 4.2).

(G4) *Using DNS for Geoblocking.* Finally, operators could only answer local DNS queries and simply block or drop any external requests, to prohibit their customers from accessing the IMS via Wi-Fi when being abroad. In our results, we found that `Vodafone`$_{[26202]}$ (Germany) is only serving its ePDG IP addresses for DNS requests from appropriate client IPs. This was occasionally noticed by actual customers, as we also found anecdotal evidence [3, 53, 55, 45, 23, 52] within several posts on blogs and online forums. In contrast to `Jio`$_{[405874]}$ (see above), `Vodafone`$_{[26202]}$ actually respects ECS queries, making it possible to easily demonstrate the filtering[8]. While most of the queries that were able to discover `Vodafone`$_{[26202]}$'s ePDG were issued within Germany, the DNS server occasionally answered requests from external countries (e.g., several close countries like Austria, and Ireland, but also more distant countries like Kazakhstan and Japan). We tried to investigate what caused these false positives and checked the IP information of several results according to the MaxMind database. We found, that many misclassified IPs had Germany set as the `registered_country` (an additional meta-information in the database), although delivering an external country as the primary hosting country.

IPv6. Again, the vast majority (over 90 %) of all domains that return `AAAA` records can be found within the first group (G1). Additionally, we found members of (G2) and (G3), namely Israeli `Cellcom`$_{[42502]}$, replying with a single IP chosen without any visible location bias ((G2)) during a transition period switching their ePDG server, and `Canadian Bell`$_{[302610]}$, that serve their customers with dedicated IPs that are (vaguely) grouped by geographic region. In contrast to IPv4, we did not see any operator that used DNS for geoblocking purposes.

4.2 Probing VoWiFi Servers

After identifying the ePDG target IP addresses, we need to investigate whether the ePDG accepts connections from different geographic regions. To distribute our measurements across the globe, we again use our containerized architecture that leverages various VPN- and cloud services.

Data Sources and Covered Countries. Both MaxMind [39] and Android's MCC table [1] link entities (i.e., IP addresses, operators) to

[7] Jio also uses 21 other MNCs that were shortened for the sake of visibility.

[8] Example queries are shown in the Appendix A.1

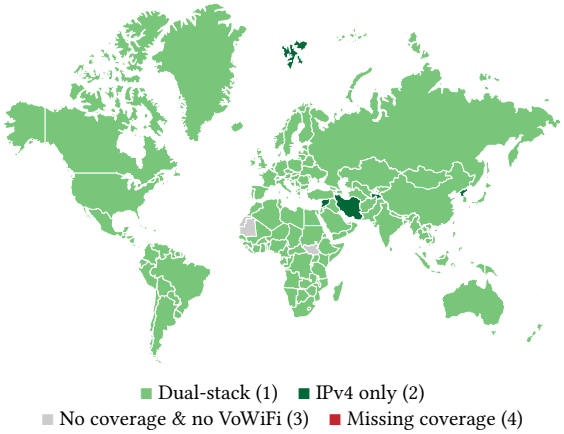


Figure 5: Leveraging VPN and cloud services, we reach de facto worldwide measurement coverage.

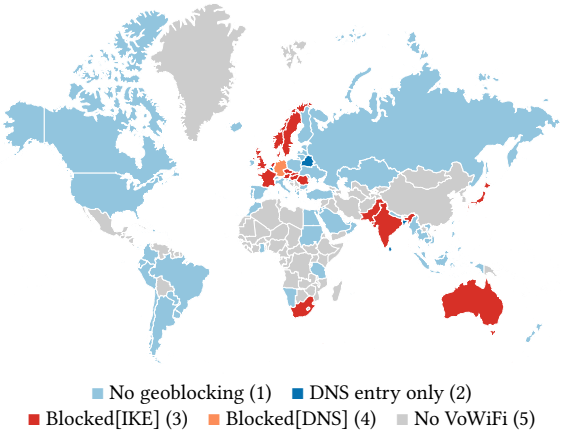


Figure 6: Summarized results or our DNS discovery and IKE probing measurements.

a geographical region (i.e., an ISO 3166-1 [24] country). ISO 3166-1 currently comprises 249 countries, and each can be linked to its corresponding continent.

In total, we issued more than 55,700 IKE scan rounds, that were executed from 219 different countries for IPv4 and 208 for IPv6 respectively. Table 3 gives an overview on how the measurements are distributed among countries and services.

The number of countries we used for scanning greatly exceeds the number of countries where operators actually support and use VoWiFi in practice. To maximize the scope of our study and gain a defacto worldwide view, we scanned from all 219 available countries. More specifically, this was done i) to maximize the discovered DNS entries in case of DNS-based blocking, and ii) to also detect blocking of countries that do not have VoWiFi yet, but are blocked by foreign operators (e.g., for political reasons).

4.2.1 Measurement Coverage and Domestic Results. Our measurements are limited by the countries that are available via our VPN- and cloud services. We cover 107/109 (IPv4) and 16/16 (IPv6) of our target countries (target territories defined by the DNS discovery). The two countries we are missing for IPv4 are both overseas departments of France, which are relatively small countries: French Polynesia (one operator) and La Réunion (two operators). Within French Polynesia, the single operator that was discovered via DNS ($\texttt{Ora}_{[54705]}$) responds to IKE requests from all over the globe and thereby is not geoblocked. The same holds true for one Reunionese carrier($\texttt{Zeop}_{[64704]}$), while the second one ($\texttt{Orange}_{[64700]}$) was not responsive from any country we scanned from and thereby is out of scope of our measurement coverage[9]. Figure 5 gives an overview of the global scope of our measurements: We've accomplished dual-stack connectivity in nearly all covered countries (1) and have some countries with IPv4 only coverage (2). Several countries that were not available via our scan infrastructure do not have VoWiFi yet (3), i.e., there are no DNS entries for any ePDG within that area. The two countries with missing coverage (i.e., French Polynesia and La Réunion) are very small and thereby not visible on the global map (4).

We scan every IP address that was discovered via DNS in the previous step (Table 2) from all available countries. However, not all endpoints are reachable and actually respond to requests. Presumably, some of the DNS entries are only set for testing purposes and are in fact not in active use. Table 4 reduces the original DNS set by removing all IPs that do not answer requests from any possible location (i.e., not even when connecting to the ePDG from a domestic IP address).

Dealing with Inaccuracies. Besides blocking according to a user's location, operators might also block the usage of popular VPN services or cloud infrastructure as a connection relay. More specifically, they might block some of our VPN services, or Autonomous Systems (ASes) that are commonly shared among popular VPN services. Furthermore, not all operators use MaxMind as their geolocation broker, introducing potential inconsistencies between our classification and the actual geolocation at the operator's sides. Additionally, we might occasionally experience connection hiccups within our scan infrastructure (e.g., when the used VPN unexpectedly closes the current connection). Lastly, some ePDGs seem to be very sensitive and do not accept repeated connections from the same IP address, simply ignore IKE handshakes that contain inappropriate cipher sets or go on/offline (i.e., experience downtimes) throughout our measurement study. To cope with these instabilities, we randomly switch VPN servers between subsequent measurements and aim for a big and diverse set of VPN services and realized measurements.

Also, we use the *responsiveness* (i.e., the percentage an ePDG was reported as responsive) of the domestic case as a *baseline*, to decide whether the server was blocked in a roaming case. As a matter of precaution, we consider some space for generous error bounds (e.g., when we or the operator misclassify the VPN IP and assume the wrong country) by only flagging an area as *geoblocked*, when it has

[9]For the remaining paper, we've treated it as a non-responsive ePDG, although technically it could be geoblocked and merely reachable via the home country (La Reunion).

	IPv4						IPv6					
Region (via MCC)	Countries [b]		Domains		IPs		Countries [b]		Domains		IPs	
2 Europe	37	90%	128	86%	271	87%	5	62%	5	56%	9	56%
3 North America & Caribbean	19	95%	61	88%	113	89%	1	50%	1	50%	5	71%
4 Asia, Middle East	19	90%	122	92%	141	86%	3	100%	16	94%	8	73%
5 Australia, Oceania	9	100%	28	88%	49	82%	0	0%	0	0%	0	0%
6 Africa	8	100%	12	86%	19	86%	0	0%	0	0%	0	0%
7 South- & Central America	9	100%	24	92%	35	88%	1	100%	1	100%	1	100%
9 Worldwide [c]	0	0%	0	0%	0	0%	0	-	0	-	0	-
Total	101	93%	375	89%	628	87%	10	62%	23	74%	23	57%

[b] according to the MCC. [c] Satellite, Air, Maritime, Antarctica.

Table 4: Overview of ePDGs responding to our IKE scan from at least one vantage point. The percentage columns compare the values to the results of the previous step (i.e., the values in Table 2).

less than 10% of the assumed *baseline responsiveness*. Thereby, the responsiveness remains comparable and we can correctly classify an ePDG, even when unexepcted events occur, e.g., when the ePDG server experiences a downtime during our measurement campaign. Also, this approach allows us to more accurately flag a country as *geoblocked*, even when we get responses in a minority of cases due to geolocation errors from the operator.

We do not differentiate between a successful `IKE_INIT` phase and a received error message. For example, when the ePDG responds with a `NO_PROPOSAL_CHOSEN` error to indicate that the offered ciphers are not accepted by the gateway we assume that there is no geoblocking for this client IP since a response packet was received (i.e., there is no IP-based blocking for this location).

4.2.2 Roaming Results. We found geoblocking at the IKE layer in various forms and granularities. For example, the target area of the blocking can be rather big (e.g., blocking all foreign connections) or small (e.g., only blocking a specific set of target countries). The confidence and robustness of our classification results (whether a server blocks by location or not) corresponds to the amount and diversity (e.g., IP- and AS diversity) of our measurements within that area. Therefore, we automatically classify global and continental geoblocking, but rely on manual inspection to identify country-level geoblocking, due to reduced diversity in small target countries. A domain is only flagged as geoblocked, when we observe geoblocking for all corresponding IP addresses.

Large-scale Geoblocking. According to our results, we experience global geoblocking for 12.5% (IPv4) and 65.2% (IPv6) of all tested domains. If we extend this to domains that are blocked from at least one continent (while being reachable from other areas) the percentage increases to 14.6% for IPv4 and remains unchanged for IPv6. Table 5 shows the detailed roaming results of our IKE probing. Overall, our measurements show that geoblocking is especially common within Europe and Asia but there are also continents without any large-scale blocking at all (e.g., North and South America).

Interestingly, we found numerous operators in European countries (Austria, Czech Republic, Denmark, France, Hungary, Luxembourg, the Netherlands, Norway, Romania, Sweden, and the United Kingdom) with large-scale blocking measures. Moreover, many operators within the EU also block connections from their neighboring EU countries. In contrast, intra-EU roaming via regular radio access is possible without additional costs in these countries, due to the Roam Like At Home (RLAH) doctrine [15]. As an exception, we've found a single Slovakian operator with worldwide geoblocking that specifically exempts EU/EAA countries from the blocking.

While most operators employing large-scale blocking within Asia are based in India (15/37), we've also discovered geoblocking at operators from Hong Kong, Israel, Japan, Pakistan, and Singapore. Lastly, the remaining operators were found in Australia and South Africa.

The overall results are visualized in Figure 6. For the majority of countries there was no large-scale geoblocking discovered (1). Some territories had DNS entries for an ePDG, but did not respond to any of our IKE scans (2). Moreover, we found geoblocking measures via IKE (3) and DNS (4) scans. Finally, some countries do not have DNS entries for ePDGs at all and therefore do not support VoWiFi yet (5).

Country-targeted Geoblocking. In some cases we see more fine-grained blocking or exemptions from large-scale blocks. For example, one Australian operator that generally blocks foreign IPs specifically allows connections from other Oceanian (e.g., New Zealand) and Asian (e.g., the Philippines, Malaysia , China) countries.

However, we also see country-targeted blocking within continents and countries that otherwise do not engage geoblocking. For example, an Israeli operator specifically blocks several African (e.g., Lybia, Algeria) and Asian (e.g., Iran, Iraq) countries, despite their geographical proximity. Moreover, several operators, e.g., in the United States or Ecuador, block connections coming from Russia or Ukraine, potentially for political or security reasons.

IPv6. For IPv6, we've detected geoblocking for 65.2% of all tested domains. However, these numbers are only caused by operators from two distinct countries: India and Japan. Compared to IPv4, the overall percentage is higher because India is among the leading nations when it comes to IPv6 adoption and simultaneously a country where geoblocking at VoWiFi is fairly popular.

Interestingly, we've found one Hungarian operator that supports both IPv4 and IPv6 and blocks external connections only on the IPv4

Region (via MCC)	IPv4 Countries [b]		IPv4 Domains		IPv6 Countries [b]		IPv6 Domains	
2 Europe	12	32%	19	15%	0	0%	0	0%
3 North America & Caribbean	0	0%	0	0%	0	0%	0	0%
4 Asia, Middle East	5	26%	32	26%	2	67%	15	94%
5 Australia, Oceania	2	22%	3	11%	0	-	0	-
6 Africa	1	12%	1	8%	0	-	0	-
7 South- & Central America	0	0%	0	0%	0	0%	0	0%
Total	20	20%	55	15%	2	20%	15	65%

[b] according to the MCC.

Table 5: Overview of ePDGs where we encountered large-scale (global or continental) geoblocking. The percentage columns relate the values to the parent population of responsive ePDGs in the corresponding area (cf. Table 4).

stack. Thereby, customers could circumvent the blocking by simply connecting via IPv6 (a similar phenomenon has been discovered by previous research [8]).

Overall, the available geolocation data seems to be more accurate for IPv6 because — compared to IPv4 — we see less blurring (i.e., for each distinct country the responsiveness is either 0 or 100%).

Revisited: `Reliance Jio`$_{[405874]}$ India. As stated in Section 4.1, we found that this operator clearly separates the IP addresses that are returned via DNS for local and external customers. Analyzing the two subsets (i.e., local, and external) at the ePDG layer, we see that this behavior also reoccurs when connecting to the gateway: The set of local IP addresses does not accept any connections from abroad. Interestingly, this behavior is also mutual, i.e., the external ePDG does not accept any local connections from India either. Since a customer can thereby establish a connection to an ePDG from any location we did not account this behaviour as geoblocking. However, the existing separation mechanism could possibly be used to differentiate or block at a later stage.

Revisited: `Vodafone`$_{[26202]}$ Germany. In Section 4.1 we showed that this operator uses DNS-based blocking to prevent customers from connecting to the ePDG from external locations. Taking a closer look at the probing results of the corresponding endpoints, we see that the blocking is solely done by the DNS server and does not occur at the IKE layer (i.e., it accepts connections from all tested countries). Thereby, sneaky customers may simply evade the blocking by manually adding the correct DNS entries to their system, or by using specific DNS servers that always return the corresponding IPs (e.g., a local resolver in Germany that does not forward the client's subnet).

5 RELATED WORK

This section presents an overview of existing research and studies that contribute to the understanding and context of the subject matter at hand.

Geoblocking and Internet Censorship. In 2018, the European Union (EU) banned unjustified geoblocking within the European single market [14]. Nevertheless, the regulation includes a number of exceptions (e.g., for copyrighted audiovisual content like Netflix) and significant portions of the world remain without regulatory measures.

McDonald et al. [40] proved the prevalence of geoblocking practices in the Internet by finding geoblocking at large CDNs for nearly all of the 177 examined countries.

Additionally, Kumar et al. [31] analyzed the mobile app ecosystem from vantage points in 26 countries. Aside from geoblocking being a common practice in the mobile app field they also found geodifferences occurring between differing countries (i.e., developers shipping different versions of an app to specific regions).

Lastly, geoblocking is often introduced as a governmental censorship measure. Ramesh et al. [47] showed, that — ever since Russia's invasion of Ukraine in February 2022 — there are geofences between Russia and the rest of the world. Thereby, Russian users are not able to consume Western news or social media and Russian government domains remain inaccessible from regions like the EU and US.

VoLTE and VoWiFi Security. Prior research discovered numerous security- and privacy-related vulnerabilities in VoLTE and VoWiFi. For example, Kim [29] showed that early VoLTE deployments were prone to data traffic free-riding attacks since the packet-switched voice channel provided an unmetered breakout to the public Internet. Furthermore, both VoLTE and VoWiFi were found to occasionally leak precise subscriber info (e.g., Cell IDs) via the underlying SIP traffic [30]. Additionally, VoWiFi is vulnerable to IMSI catching attacks [42, 43].

More recently, Lu et al. [33], Xie et al. [56], and Lee et al. [32] presented practical Denial of Service (DoS) attacks for VoLTE and/or VoWiFi. Moreover, Hu et al. showed that VoLTE's emergency services are also vulnerable to DoS and free-riding attacks [22].

In 2023, the Google Project Zero team discovered four severe Exynos vulnerabilities that allowed an attacker to execute arbitrary commands on the baseband processor of the most recent Pixel and many Samsung phones by injecting malicious SIP messages [41] into the VoLTE/VoWiFi VoIP traffic.

Lastly, Gegenhuber et al. [17, 18] uncovered insecure VoWiFi configurations and shortcomings at the corresponding key exchange.

Active Roaming Experiments. Large-scale studies that involve many operators and countries are usually limited by the complex

ecosystem and the required coordination effort. However, there are several approaches [37, 51] where specifically built measurement devices were placed into target locations to measure the implications (e.g., QoE) of roaming. Additionally, Sahin and Francillon [50] observed hijacking of traditional voice calls that were redirected to over-the-top (OTT) services (e.g., WhatsApp, Viber) to bypass/monetize termination fees.

Recently, Gegenhuber et al. [20, 19] introduced the MobileAtlas measurement platform that tries to overcome the mentioned scalability issues by tunneling the communication between SIM card and modem over the Internet. Their platform provides flexible roaming measurements and capabilities for a rich set of cellular features, including Internet and voice-based measurements.

Evaluating and Fingerprinting VPNs. The commercial VPN ecosystem is a multi-billion dollar industry [21] and thereby has been an interesting research target. For example, previous work [28, 46] analyzed and compared existing VPN services and found that many solutions leak user traffic or advertise wrong server locations.

In contrast, Maghsoudlou et al. [36] did not purchase any VPN subscriptions but executed Internet-wide scans to discover and fingerprint VPNs, finding over 7 million IPsec servers.

6 DISCUSSION

Our findings indicate that a notable proportion of operators are implementing IP-based restrictions to prevent customers from using VoWiFi in specific locations. While we also discovered DNS-based approaches, most operators implement it directly at the IKE layer. Furthermore, the found geoblocking measures exhibit a degree of regionality, meaning they are more prevalent in certain areas (e.g., Eurasia) compared to others (e.g., North- and South America).

To the best of our knowledge, this is the first study providing a comprehensive overview of the existing VoWiFi infrastructure on a global scale. Understanding the current deployment of any widely used telecommunication system is vital from a security standpoint. Our findings go beyond this by revealing that the observed blocking has significant repercussions for emergency calling.

Implications to Emergency Calling. Recent reports show, that there currently is no adequate support for VoLTE roaming in substantial parts of the world [48, 54]. Since many operators are actively shutting down their 2G/3G legacy networks, this scenario has the potential to result in significant repercussions for the functionality of emergency calling [49, 10, 11].

Although we believe that affected operators will address and fix these issues in the long term, VoWiFi could help to mitigate these shortcomings in the present day. Tourists or travellers frequently have access to WiFi, such as in their accommodations or through free WiFi hotspots in public spaces. According to the specification [13], VoWiFi should be used by the UE for emergency calling when traditional radio services (VoLTE roaming, CSFB) are unavailable. The phone ultimately tries to reach both the home and the visited ePDG. However, there are circumstances where the visited ePDG might not be available, e.g., when there is no VoWiFi support in the visited country or when the customer leaves the phone in flight mode to not cause any unintentional roaming fees and thereby the current location is not known to the phone).

While operators can override the default ePDG for emergency services by setting corresponding DNS records (`sos.epdg.epc.mnc{y}.mcc{x}.pub.3gppnetwork.org`) [12], only four operators had appropriate DNS entries. In all four cases, the emergency ePDG referenced the original ePDG's IP address, which is also the default behaviour when no `sos` entry is found. Controversially, two of the operators specifically using `sos`-domains nevertheless block IKE-inquiries coming from foreign IP addresses.

If an operator deploys DNS- or IKE-based geoblocking, these measures will also impact emergency services, actively denying persons in need from making an emergency call. Similarly, the emergency service via Wi-Fi is also unavailable in the home country, when customers use a VPN connection or an international SIM card (e.g., utilizing a travel router) that provides the Internet uplink via a foreign country.

Economic and Net Neutrality Perspective. In contrast to regular roaming over the radio interface – where the foreign roaming partner charges the home operator for the terminated calls and services – there is no additional economic overhead for VoWiFi calls that are initiated from overseas customers. In fact, calls terminated via VoWiFi are notably cost-effective for operators, as they eventually reduce expenses associated with the required radio transmission infrastructure (i.e., base stations) and spectrum licensing fees. Instead, the traffic is routed via external infrastructure (i.e., a WiFi AP) that was provided and paid for by the customer.

Moreover, adhering to the principle of net neutrality and the Open Internet, it is not allowed to discriminate (i.e., block) a customer's data packets by their source or destination IP address, particularly when motivated solely by economic interests (cf. differential pricing and zero-rating). Assuming the discovered blocking practices were employed mainly for economic reasons, they could potentially be seen as a net neutrality violation.

Ways to Evade Geoblocking. Common mobile operating systems (i.e., Android and iOS) support changing the used DNS server for WiFi interfaces out of the box, which should allow easy bypassing of DNS-based blocking.

Additionally, both Android and iOS have built-in support for applying a system-wide VPN connection. However, even with an active VPN, VoWiFi uses a direct route over the WiFi interface. While this hinders standalone solutions on non-rooted phones, a viable alternative could be to use so-called travel routers. These devices act as an intermediary between WiFi AP and smartphone and typically offer the ability to redirect all traffic through a VPN connection, and thus via a customer's home country.

Inaccurate Geolocation and Blocked VPN Services. Each VPN service uses one or more IP addresses for each country within its coverage. In the course of this study, we occasionally experienced outliers where certain IP addresses, address ranges, or Autonomous Systems (ASes) from specific VPNs experienced blocking even though belonging to the same country as the probed operator. We suppose that those connections were either blocked because their geolocation was wrongfully classified from the operator, or because the operator specifically blocked connections from IPs that are associated with certain VPN services.

6.1 Limitations

Although we are able to detect a considerable amount of blocking operators, several factors limit our approach. Due to the fact that these constraints limit the geoblocking variants that we're able to detect, our results can be seen as a lower bound. Thereby, the actual occurrence of geoblocking in practice is most likely even higher.

More Advanced Blocking. Our current approach is designed to detect simple IP-based blocking rules. However, some operators use more sophisticated (e.g., IMEI-based) ways to decide whether a customer should be able to use VoWiFi under current circumstances. For example, an operator could use the last known roaming status via the radio network to decide whether a customer is currently in their home country or abroad. Additionally, VoWiFi roaming can only be offered to a limited set of customers (i.e., *premium users*). For example, we found an Austrian operator that requires an additional subscription package to get VoWiFi enabled during international roaming [35]. In this case, the operator initially accepts IKE packets from all locations and decides whether to grant (or drop) access to VoWiFi at a later stage, when the subscriber's identity is known. We focus on straightforward IP-based blocking because measuring these more advanced blocking techniques on a global scale would necessitate the acquisition of SIM cards for an unfeasible number of operators. Moreover, it is crucial to highlight that relying solely on IP address-based blocking for VoWiFi access also results in the restriction of access to emergency calling services. Implementing blocking techniques that compromise the unconditional availability of emergency services that people depend on in life-threatening situations is highly ill-advised.

Implementation Incompatibility and Blocked VPN Services. Our IKE probing is based on a cellular-specific open-source implementation of the IKEv2 protocol[10]. Possibly, some ePDGs are not compatible with this implementation or do not answer requests that offer inappropriate cipher suites. Similarly, operators could block access from well-known VPN services or ASes and IP ranges that are used by them. As a countermeasure, we use multiple services to increase the diversity within our clients. Since we got responses from a high share of ePDGs that were discovered via DNS, these two factors do not considerably influence our results.

Non-Native and Non-3GPP-Compliant VoWiFi. Whereas our approach focuses on the native 3GPP version of VoWiFi there are also other ways to support voice calling functionalities over Wi-Fi networks. For example, an operator might require their customers to download and install a dedicated VoWiFi app that gives direct access to the IMS network in a non-compliant way [34]. Additionally, some operators might use non-default ways to communicate their ePDGs, e.g., only resolve the hostname via an internal DNS server that is shipped to their customers via DHCP. Lastly, we do not cover VoIP calling via OTT services (e.g., WhatsApp, Viber, Skype).

6.2 Ethical Considerations

Ethical considerations are vital to the field of measurements, especially with active measurements conducted in live systems. From an operator's perspective, we only interact with two endpoints i.e., retrieving IP addresses via the authoritative DNS server and performing the initial IKE handshake with the ePDG. In both cases, we tried to mimic normal user behavior by sending well-formed requests, i.e., we did not deviate from the protocol specification or do any fuzzing. From a bandwidth perspective, our measurements should not overstress any operator, since for each measurement target we were only sending several requests per hour. Finally, our measurements did not involve any actual user data (e.g., IMSIs or IMEIs), since we were only performing the initial handshake where the cryptographic parameters for the subsequent connection are exchanged.

6.3 Dissemination and Responsible Disclosure

Beyond disseminating our findings to the scientific community, we want to enable an informed debate by raising awareness within the industry and highlight potential issues regarding IP-based geoblocking among regulators and emergency associations. Therefore, we reported our results to the GSM Association (GSMA), the Body of European Regulators for Electronic Communications (BEREC), and the European Emergency Number Association (EENA). GSMA and EENA invited us to expand upon and discuss our findings within dedicated meetings and BEREC responded to our inquiry via Email.

More specifically, GSMA discussed the topic within their panel of experts and invited us to *"present this case to the GSMA's Fraud and Security Architecture Group, to make the issue know to their members"*. According to BEREC *"there are no legal obligations for Wi-Fi calls in roaming stemming from the Roaming Regulation, and they do not see this as a breach of the Open Internet Regulation"*. Lastly, *"EENA will study the topic and discuss with its community to understand the extent of any potential impact the practice of geoblocking could have on access to emergency services through emergency communications"*.

7 CONCLUSION

Many operators worldwide have implemented support for VoLTE and VoWiFi and rely on the IP Multimedia Subsystem (IMS) as a centerpiece for their communication services. Additionally, the IMS and VoWiFi seamlessly integrate with the upcoming 5G network generation and thereby will also play a relevant role in the future.

Just like any crucial system that impacts our daily lives, it's important to be aware of its current state and to comprehend its internal workings. We therefore give a comprehensive overview of the global deployment of VoWiFi and investigate existing geoblocking measures, discovering IP-based blocking mechanisms both at the DNS and IKE layer. We emphasize that, unlike geoblocking measures commonly employed in web or streaming applications, telecommunications is a more sensitive domain, where such measures could potentially have adverse effects on the functionality of emergency calls. Thus, we hope that the insights of our study will raise the awareness among customers and security researchers, while also contributing to the decision-making processes of policy makers and operators in the future.

To encourage other researchers to further profit from our work and engineering effort, we've publicly released the source code of our measurement infrastructure [11].

[10] https://github.com/fasferraz/SWu-IKEv2

[11] https://github.com/sbaresearch/scanywhere

ACKNOWLEDGMENTS

We want to thank Quentin McGaw and all other Gluetun contributors – their previous work made it considerably easier to build *scanywhere* and thus to globally distribute our measurements.

This project was funded through the NGI0 Entrust Fund, a fund established by NLnet with financial support from the European Commission's Next Generation Internet, under the aegis of DG Communications Networks, Content and Technology under grant agreement No 101069594.

SBA Research (SBA-K1) is a COMET Centre within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMAW, and the federal state of Vienna. COMET is managed by FFG.

A APPENDIX

A.1 DNS-based Blocking at Vodafone

Resolving standardized ePDG domain to `CNAME` reference:

```
$ dig epdg.epc.mnc002.mcc262.pub.3gppnetwork.org
=> returns CNAME epdg.epc.drz1.vodafone-ip.de
```

Actual resolution (Google vs. Vodafone IP range):

```
# requesting via Google IP (United States)
$ dig +trace epdg.epc.drz1.vodafone-ip.de +subnet=104.154.0.0/24

# requesting via Vodafone IP (Germany)
$ dig +trace epdg.epc.drz1.vodafone-ip.de +subnet=109.192.0.0/24
```

A.2 Artifact Appendix

The research artifacts accompanying this paper are available via 10.5281/zenodo.11089362.

REFERENCES

[1] Google (AOSP). 2006. Android MCC Table. WebPage. Accessed: 2023-07-23. (2006). https://android.googlesource.com/platform/frameworks/opt/telephony/+/refs/heads/main/src/java/com/android/internal/telephony/MccTable.java.

[2] 2023. Airtel India: WiFi calling FAQs. WebPage. Accessed: 2023-08-21. (2023). https://www.airtel.in/wifi-calling/faqs.

[3] 2017. Android WiFi Calling VoWifi not working, wont resolve domain. WebPage. Accessed: 2023-07-27. (2017). https://groups.google.com/g/public-dns-discuss/c/AExOcbpl09w.

[4] Jari Arkko and Henry Haverinen. 2006. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). RFC 4187. (Jan. 2006). DOI: 10.17487/RFC4187.

[5] Thomas P. Brisco. 1995. DNS Support for Load Balancing. RFC 1794. (Apr. 1995). DOI: 10.17487/RFC1794.

[6] Cloudflare. 2021. Cloudflare FAQ: ECS Policy. WebPage. Accessed: 2023-07-04. (2021). https://developers.cloudflare.com/1.1.1.1/faq/#does-1.1.1.1-send-edns-client-subnet-header.

[7] Carlo Contavalli, Wilmer van der Gaast, David C Lawrence, and Warren "Ace" Kumari. 2016. Client Subnet in DNS Queries. RFC 7871. (May 2016). DOI: 10.17487/RFC7871.

[8] Jakub Czyz, Matthew Luckie, Mark Allman, Michael Bailey, et al. 2016. Don't forget to lock the back door! a characterization of ipv6 network security policy. In *Network and Distributed Systems Security (NDSS)*.

[9] Dr. Steve E. Deering and Bob Hinden. 2006. IP Version 6 Addressing Architecture. RFC 4291. (Feb. 2006). DOI: 10.17487/RFC4291.

[10] 2022. EENA 2022: Access to emergency services is being impacted by the lack of VoLTE interoperability. WebPage. Accessed: 2023-08-21. (2022). https://eena.org/knowledge-hub/press-releases/many-europeans-cannot-call-911-when-traveling-to-the-us/.

[11] 2022. Ensuring continuity of access to emergency services in the transition to IMS/VoLTE services. WebPage. Accessed: 2023-08-21. (2022). https://eena.org/blog/webinars/volte-standardisation-problem/.

[12] ETSI. 2023. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Numbering, addressing and identification. ETSI, (2023). https://www.etsi.org/deliver/etsi_ts/123000_123099/123003/17.10.00_60/ts_123003v171000p.pdf.

[13] ETSI. 2022. Universal Mobile Telecommunications System (UMTS); LTE; Architecture enhancements for non-3GPP accesses. ETSI, (2022). https://www.etsi.org/deliver/etsi_ts/123400_123499/123402/17.00.00_60/ts_123402v170000p.pdf.

[14] European Parliament, Council of the European Union. 2018. Regulation (EU) 2018/302. (2018). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32018R0302.

[15] European Parliament, Council of the European Union. 2022. Regulation (EU) 2022/612. (2022). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=LEGISSUM:4593182.

[16] Sheila Frankel and Suresh Krishnan. 2011. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. RFC 6071. (Feb. 2011). DOI: 10.17487/RFC6071.

[17] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. 2024. POSTER: Never Gonna Give You Up: Exploring Deprecated NULL Ciphers in Commercial VoWiFi Deployments. In *17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[18] Gabriel K. Gegenhuber, Florian Holzbauer, Philipp É. Frenzel, Edgar Weippl, and Adrian Dabrowski. 2024. Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments.

[19] Gabriel K. Gegenhuber, Wilfried Mayer, and Edgar Weippl. 2022. Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs. In *IEEE Global Communications Conference (GLOBECOM)*.

[20] Gabriel Karl Gegenhuber, Wilfried Mayer, Edgar Weippl, and Adrian Dabrowski. 2023. MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research. In *Usenix Security Symposium 2023*.

[21] 2020. Google Trends Reveals Surge in Demand for VPN. WebPage. Accessed: 2023-08-21. (2020). https://www.namecheap.com/blog/vpn-surge-in-demand/.

[22] Yiwen Hu et al. 2022. Uncovering Insecure Designs of Cellular Emergency Services (911). In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 703–715.

[23] 2018. Ich hab Probleme wifi calling zu aktivieren. WebPage. Accessed: 2023-07-27. (2018). https://forum.vodafone.de/t5/Archiv-Apple/Ich-hab-Probleme-wifi-calling-zu-aktivieren/td-p/1718155.

[24] 2000. ISO 3166-1:1997: Codes for the representation of names of countries and their subdivisions — Part 1: Country codes. Standard. International Organization for Standardization, (Nov. 2000). https://www.iso.org/iso-3166-country-codes.html.

[25] ITU. 2017. T-SP-E.212A: List of Mobile Country or Geographical Area Codes. WebPage. Accessed: 2023-07-23. (2017). http://handle.itu.int/11.1002/pub/80f1788f-en.

[26] Charlie Kaufman. 2005. Internet Key Exchange (IKEv2) Protocol. RFC 4306. (Dec. 2005). DOI: 10.17487/RFC4306.

[27] Charlie Kaufman, Paul E. Hoffman, Yoav Nir, Pasi Eronen, and Tero Kivinen. 2014. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296. (Oct. 2014). DOI: 10.17487/RFC7296.

[28] Mohammad Taha Khan, Joe DeBlasio, Geoffrey M Voelker, Alex C Snoeren, Chris Kanich, and Narseo Vallina-Rodriguez. 2018. An Empirical Analysis of the Commercial VPN Ecosystem. In *Proceedings of the Internet Measurement Conference 2018*, 443–456.

[29] Hongil Kim, Dongkwan Kim, Minhee Kwon, Hyungseok Han, Yeongjin Jang, Dongsu Han, Taesoo Kim, and Yongdae Kim. 2015. Breaking and Fixing VoLTE: Exploiting Hidden Data Channels and Mis-implementations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 328–339.

[30] Sekwon Kim, Bonmin Koo, and Hwankuk Kim. 2015. Tracking Location Information of VoLTE Phones. In *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 703–708.

[31] Renuka Kumar, Apurva Virkud, Ram Sundara Raman, Atul Prakash, and Roya Ensafi. 2022. A Large-scale Investigation into Geodifferences in Mobile Apps. In *31st USENIX Security Symposium (USENIX Security 22)*, 1203–1220.

[32] Hyunwoo Lee, Imtiaz Karim, Ninghui Li, and Elisa Bertino. 2022. Vwanalyzer: a systematic security analysis framework for the voice over wifi protocol. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 182–195.

[33] Yu-Han Lu, Chi-Yu Li, Yao-Yu Li, Sandy Hsin-Yu Hsiao, Tian Xie, Guan-Hua Tu, and Wei-Xun Chen. 2020. Ghost calls from operational 4G call systems: IMS vulnerability, call DoS attack, and countermeasure. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–14.

[34] 2017. Magenta Austria: VoWiFi Roaming. WebPage. Accessed: 2023-08-21. (2017). https://www.gsma.com/futurenetworks/digest/vowifi-implementation-insight-china-unicom/.

[35] 2023. Magenta Austria: VoWiFi Roaming. WebPage. Accessed: 2023-08-21. (2023). https://www.magenta.at/handytarife/zusatzpakete/vowifi-roaming.

[36] Aniss Maghsoudlou, Lukas Vermeulen, Ingmar Poese, and Oliver Gasser. 2023. Characterizing the VPN Ecosystem in the Wild. In *International Conference on Passive and Active Network Measurement*. Springer, 18–45.

[37] Anna Maria Mandalari et al. 2018. Experience: Implications of Roaming in Europe. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 179–189.

[38] Philip Matthews, Iljitsch van Beijnum, and Marcelo Bagnulo. 2011. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146. (Apr. 2011). DOI: 10.17487/RFC6146.

[39] MaxMind, Inc. 2023. GeoLite2 Free Geolocation Data. WebPage. Accessed: 2023-08-01. (2023). https://dev.maxmind.com/geoip/geolite2-free-geolocation-data.

[40] Allison McDonald, Matthew Bernhard, Luke Valenta, Benjamin VanderSloot, Will Scott, Nick Sullivan, J. Alex Halderman, and Roya Ensafi. 2018. 403 Forbidden: A Global View of CDN Geoblocking. In *Proceedings of the Internet Measurement Conference 2018*, 218–230.

[41] 2023. Multiple Internet to Baseband Remote Code Execution Vulnerabilities in Exynos Modems. WebPage. Accessed: 2023-08-21. (2023). https://googleprojectzero.blogspot.com/2023/03/multiple-internet-to-baseband-remote-rce.html.

[42] Piers O'Hanlon and Ravishankar Borgaonkar. 2016. WiFi-Based IMSI Catcher. Blackhat Europe 2016. (2016).

[43] Piers O'Hanlon, Ravishankar Borgaonkar, and Lucca Hirschi. 2017. Mobile Subscriber WiFi Privacy. In *2017 IEEE Security and Privacy Workshops (SPW)*. IEEE, 169–178.

[44] Kenechi Okeleke, Harry Fernando Aquije Ballon, and James Joiner. 2023. GSMA: The Mobile Economy 2023. https://www.gsma.com/mobileeconomy/wp-content/uploads/2023/03/270223-The-Mobile-Economy-2023.pdf.

[45] 2017. Probleme und Workarround für WiFi Calling mit Vodafone. WebPage. Accessed: 2023-07-27. (2017). https://ntankl3.de/probleme-und-workaround-fuer-wifi-calling-mit-vodafone/.

[46] Reethika Ramesh, Leonid Evdokimov, Diwen Xue, and Roya Ensafi. 2022. VPNalyzer: Systematic Investigation of the VPN Ecosystem. In *Network and Distributed System Security*, 24–28.

[47] Reethika Ramesh et al. 2023. Network Responses to Russia's Invasion of Ukraine in 2022: A Cautionary Tale for Internet Freedom. In *32nd USENIX Security Symposium (USENIX Security 23)*, 2581–2598.

[48] 2022. Roaming goes down the drain as AT&T disconnects European travellers. WebPage. Accessed: 2023-08-21. (2022). https://www.capacitymedia.com/article/2a5x2tyoz25np4z7yz1fk/news/roaming-goes-down-the-drain-as-at-t-disconnects-european-travellers.

[49] Hendrik Rood and Rudolf Berg. 2022. RE-VoLTE: Should we stop the shutdown of 2G/3G to save lives?? A lack of VoLTE standardisation breaks voice calling globally. (July 2022). DOI: 10.13140/RG.2.2.12321.28007.

[50] Merve Sahin and Aurélien Francillon. 2016. Over-The-Top Bypass: Study of a Recent Telephony Fraud. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1106–1117.

[51] Matteo Varvello and Yasir Zaki. 2023. A Worldwide Look Into Mobile Access Networks Through the Eyes of AmiGos. In *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–10.

[52] 2018. Vodafone Germany VoWiFi in Roaming. WebPage. Accessed: 2023-07-27. (2018). https://volteromania.blogspot.com/p/blog-page_25.html.

[53] 2016. Vodafone WiFi Calling nur ohne Google DNS möglich. WebPage. Accessed: 2023-07-27. (2016). https://www.mielke.de/blog/Vodafone-WiFi-Calling-Google-DNS--471/.

[54] 2022. VoLTEgate: Tomia says 50m visitors to US this year will be cut off from voice roaming. WebPage. Accessed: 2023-08-21. (2022). https://www.capacitymedia.com/article/2a9s377bx0f611rx9lfr5/news/voltegate-tomia-says-50m-visitors-to-us-this-year-will-be-cut-off-from-voice-roaming.

[55] 2019. WiFi-Calling, Evolved Packet Data Gateway und Google DNS. WebPage. Accessed: 2023-07-27. (2019). https://blog.rolandmoriz.de/2019/01/13/wifi-calling-evolved-packet-data-gateway-und-google-dns/.

[56] Tian Xie, Guan-Hua Tu, Bangjie Yin, Chi-Yu Li, Chunyi Peng, Mi Zhang, Hui Liu, and Xiaoming Liu. 2020. The Untold Secrets of WiFi-Calling Services: Vulnerabilities, Attacks, and Countermeasures. *IEEE Transactions on Mobile Computing*, 20, 11, 3131–3147.

6 Measuring Insecure Configurations in Commercial WiFi Calling Deployments

Publication Info

Title	Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments
Authors	Gabriel K. Gegenhuber, Florian Holzbauer, Philipp Frenzel, Edgar Weippl, Adrian Dabrowski
Publication Status	This paper is included in the Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), pp. 451–468, ISBN 978-1-939133-44-1, 2024. CORE2023 Ranking: A*.
Publication Page	`https://www.usenix.org/conference/usenixsecurity24/presentation/gegenhuber`
Code Artifacts	`https://github.com/sbaresearch/vowifi-epdg-scanning` `https://github.com/sbaresearch/mbn-mcfg-tools`
arXiv	`https://arxiv.org/abs/2407.19556`
Reference	[GHF+24]

Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments

Gabriel K. Gegenhuber[1,2], Florian Holzbauer[1,2], Philipp É. Frenzel[3], Edgar Weippl[1,4], and Adrian Dabrowski[5]

[1]University of Vienna, Faculty of Computer Science, [2]UniVie Doctoral School Computer Science, [3]SBA Research, [4]Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), [5]CISPA Helmholtz Center for Information Security

Abstract

Voice over Wi-Fi (VoWiFi) uses a series of IPsec tunnels to deliver IP-based telephony from the subscriber's phone (User Equipment, UE) into the Mobile Network Operator's (MNO) core network via an Internet-facing endpoint, the Evolved Packet Data Gateway (ePDG). IPsec tunnels are set up in phases. The first phase negotiates the cryptographic algorithm and parameters and performs a key exchange via the Internet Key Exchange protocol, while the second phase (protected by the above-established encryption) performs the authentication. An insecure key exchange would jeopardize the later stages and the data's security and confidentiality.

In this paper, we analyze the *phase 1* settings and implementations as they are found in phones as well as in commercially deployed networks worldwide. On the UE side, we identified a recent 5G baseband chipset from a major manufacturer that allows for fallback to weak, unannounced modes and verified it experimentally. On the MNO side –among others– we identified 13 operators (totaling an estimated 140 million subscribers) on three continents that all use the same globally static set of ten private keys, serving them at random. Those *not-so*-private keys allow the decryption of the shared keys of every VoWiFi user of all those operators. All these operators deployed their core network from one common manufacturer.

1 Introduction

The term *non-3GPP Access Networks* refers to the method of accessing cellular network core services without the use of a GSM/GPRS/UMTS/LTE/NR radio access network. This technique has been around since the times of GSM and has been updated multiple times since then. Some operators in the U.S. and Japan have used it to offload traffic via unlicensed Wi-Fi bands.

There are two types of non-3GPP access networks: *trusted networks* (e.g., provider-operated Wi-Fi access points) and *untrusted networks* (third-party Wi-Fi and Internet connections). In recent years, the latter variant started enjoying massive adoption as *Voice over Wi-Fi* (VoWiFi), also called *Wi-Fi Calling* or *Voice over WLAN* (VoWLAN). For the end user, it often provides better coverage, and for the operator, it provides a way to externalize the last mile's costs while keeping the full revenue.

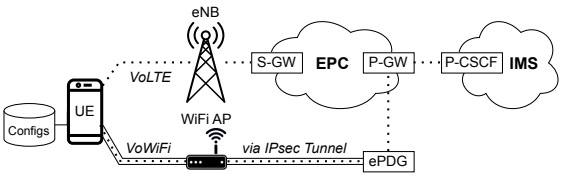


Figure 1: VoLTE compared to VoWiFi over an untrusted Internet connection – as relevant for this paper

On iPhone and Android, by default, VoWiFi is the preferred call termination channel when available.

At its core, *untrusted non-3GPP access* works by setting up at least one IPsec tunnel to the operator's Evolved Packet Data Gateway (ePDG). It uses the Internet Key Exchange (IKE) protocol [34] and relies heavily on predefined Diffie-Hellman (DH) groups, some of which are known to be weak. For example, since 2015 [15], $DH1^{768\ bits}$ is assumed to be breakable by motivated academic actors, while $DH2^{1024}$ is within reach of nation-states.

Within the IPsec tunnel, all core network access is handled like regular Voice over LTE (VoLTE). Because of its recently massively increased popularity and its security perimeter function to many Evolved Packet Core (EPC) services, we investigated its specified as well as practical security from different vantage points. We facilitate static configuration analysis, active measurements of real-world implementations in network operators, as well as active measurements of handset implementations to answer the following research questions.

RQ1 What VoWiFi key exchange methods and security parameters are preset in phones for their Mobile Network Operator (MNO)?

RQ2 What key exchange methods do operators actually sup-

port on their ePDG, and will they always prefer the strongest one?

RQ3 How strong are VoWiFi connections in the real world, and how realistic is it to downgrade to weaker, breakable key exchange methods?

We found that most operators are non-compliant with 3GPP's specifications, by still announcing and supporting deprecated DH groups weaker than 2048 bits. Furthermore, only 42% will take the extra step to request an upgrade if the client chooses a weaker group, but both parties actually support stronger groups. We also found one handset manufacturer that will silently support the much weaker DH1^{768} group, albeit not proposing it in the handshake. DH1^{768} was never part of a 3GPP specification, making those handsets susceptible to man-in-the-middle attacks. We simulate such an attack by intercepting and rewriting actual VoWiFi traffic.

More abstractly, our findings illustrate that functional overprovisioning and missing predefined procedures for deprecating cryptographic algorithms create a massive technical debt. Last but not least, we uncovered at least 13 operators[1] that used the same private keys on three continents.

The paper is structured as follows. In Section 2, we give the necessary background of how IPsec with IKE is used and embedded within the 3GPP structure. The threat model and the methodology are outlined in Sections 3 and 4, the latter of which also includes ethical considerations. Sections 5 to 7 describe our implementation and report the findings, followed by an outline on how to put those findings to work for a full stack VoWiFi attack. A related work section, a discussion, and recommendations round up the picture in Sections 8 through 10. The paper ends with a conclusion in Section 11 and an Appendix for supplementary material.

2 Background

VoWiFi is a technology that transfers voice traffic over non-3GPP access networks, typically unsecured Wi-Fi networks. It effectively routes VoLTE traffic to the EPC (and ultimately to the IMS) by encapsulating it in an IPsec tunnel over the public Internet, as shown in Figure 1. This basic technique has been around since the GSM era for network traffic off-loading and is now experiencing a resurgence due to the popularity of VoWiFi.

2.1 The IKE/IPsec/SIP Stack

The complete stack consists of a nested stack of tunnels (Figure 2). The outer (or *Phase 1*) IKEv2 layer (L1 in Figure 2) is responsible for securing the inner layers (e.g., negotiating security parameters and creating key material for the nested tunnels via IKEv2 [21]). Within this layer, the client (UE) authenticates the user via the (U)SIM card and creates a CHILD_SA (*Phase 2*), that allocates an IPsec tunnel into the EPC via the Packet Gateway (P-GW). Via this tunnel (L2 in Figure 2), the UE is assigned a dedicated IP address and can reach internal endpoints within the EPC. This level of access (and also the assigned IP address) is functionally identical to connecting to the `IMS` APN over the regular radio access network (VoLTE). Lastly, to be able to terminate voice calls, the UE uses the created CHILD_SA (IPSec tunnel) to talk to the P-CSCF (Proxy Call Session Control Function) and establish a SIP (Session Initiation Protocol) and an RTP (Real-time Transport Protocol) connection over *ipsec-3gpp* [17], secured via IPsec in transport mode. The encryption on this final layer (L3 in Figure 2) is, however, often optional and not enforced by many clients or servers.

2.2 Creating the ePDG Connection L1

In the first step, the phone connects to the Internet-facing side of the ePDG server of the appropriate MNO using its Fully Qualified Domain Name (FQDN), standardized in ETSI/3GPP TS 23.003 [24]:

`epdg.epc.mnc`⟨*id*⟩`.mcc`⟨*id*⟩`.pub.3gppnetwork.org`,

where the Mobile Country Code (MCC) and the Mobile Network Code (MNC) are globally unique for each operator. The IKE protocol (nowadays IKEv2 [21, 33, 34] or, more precisely, its slightly modified 3GPP variant [22]) is used to negotiate a session key using the Diffie-Hellman key exchange mechanism. Hereby, the client proposes its supported

[1] 12 during our initial scan and one more during responsible disclosure.

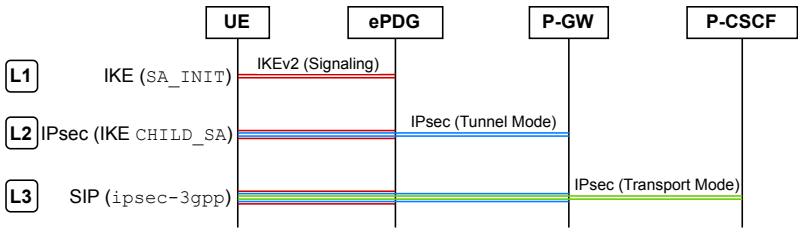


Figure 2: VoWiFi uses multiple tunnels to ensure security: L1 provides a trusted channel and manages the subsequent connections, L2 acts as a gateway to the internal infrastructure and L3 is used for the actual voice and messaging functionalities.

Security Associations (SAs), i.e., the available encryption, integrity, and key exchange algorithms (DH groups) and its preferred DH group. The ePDG chooses a subset of the proposed SAs and either accepts the favored DH group or indicates its preference toward a different DH group from the proposal. After this initial `SA_INIT` phase, all subsequent messages are encrypted and integrity-protected.

2.3 IPSec Tunnel Mode (CHILD_SA) L2

After establishing the encryption on the outer IKE layer, both endpoints (i.e., the UE and the ePDG) authenticate using EAP-AKA using credentials from the (U)SIM. Furthermore, the AKA procedure provides both parties with secret keys for the first CHILD_SA (i.e., the IPSec tunnel into the EPC). Note that the secret keys (and other SA parameters) used by the parent (i.e., the outer IKE) and child SAs are regularly renewed via repeated DH key exchanges and thus only valid for a certain period. However, the authentication of both endpoints is not renewed. Thus, cracking the outer key exchange is enough to gain stealth rewriting capabilities within the first two layers.

2.4 Session Initiation Protocol (SIP) Layer

The *ipsec-3gpp* protocol that secures this layer can ensure the confidentiality and integrity of the SIP and RTP traffic. The first two packets before establishing the encrypted channel are transmitted in plaintext (a *SIP REGISTER* that is usually answered by a *SIP Unauthorized* packet with the AKA challenge). In the past, some implementations were vulnerable on the SIP layer, e.g., Exynos [13].

However, in practice, not many operators enforce encryption and integrity on this layer. In Appendix A , we verify this experimentally. In such cases, an attacker who cracked the outer IKEv2 key exchange and is thus able to take over the first two layers could subsequently also hijack the third layer after the SIP authenticated between UE and P-CSCF is finished, effectively seizing control of all three communication layers.

Table 1: Relevant DH groups for this work, as named/numbered by IANA [32]

Name	Bits	Type
DH1[1]	768	MODP
DH2	1024	MODP
DH5[1]	1536	MODP
DH14	2048	MODP
DH15	3072	MODP
DH16	4096	MODP
DH17	6144	MODP
DH18	8192	MODP

Name	Bits	Type
DH25	192	ECP
DH26	224	ECP
DH19	256	ECP
DH20	384	ECP
DH21	512	ECP
DH31	Curve25519	

[1] never specified for 3GPP usage deprecated [23]

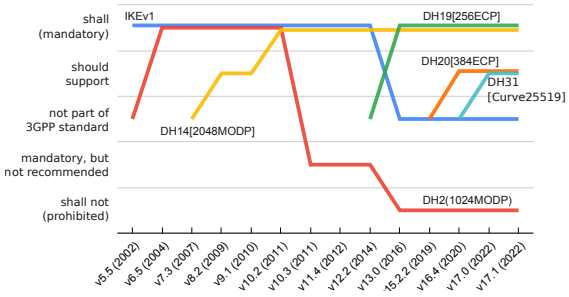


Figure 3: Development of the IPsec IKE Profile as defined in IETF TS 133.210 (IKE_SA_INIT) [23] Note: LTE started with v8, and IKEv1 has been phased out since v12.

2.5 IPsec/IKE Diffie-Hellman Key Exchange in the 3GPP's VoWiFi Ecosystem

For most of the paper, we only look at the first-stage IKE handshake, i.e., the outermost layer L1. All subsequent operations rely on the confidentiality and integrity of the negotiated encryption with the negotiated shared session key.

In contrast to much of the VoLTE/VoWiFi world (see Section 2.7), IKE itself offers an automatic negotiation for key exchange mechanisms via capabilities announcements and the selection of different Diffie-Hellman (DH) groups. A DH group defines an algorithm, a key length, and a set of public parameters. The relevant DH groups for VoWiFi (and this paper) are listed in Table 1. In 2015, researchers estimated that cracking $DH1^{768}$ is within the capabilities of a determined academic group, while $DH2^{1024}$ is within reach for nation-states [15].

Accordingly, ETSI/3GPP changed its recommendations and requirements over the years. The results of our requirement analysis for the IKE profile in TS 133 210 [23] are depicted in Figure 3. While $DH1^{768}$ was never part of the standard, $DH2^{1024}$ was recommended until 2011, then demoted to *required but not recommended* and finally prohibited (*shall not use*) in 2016. At the same time, new ECP^2-based DH groups were added as recommendations (our results show that they are rarely used).

2.6 Modular Exponential (MODP) Diffie-Hellman Key Exchange in IKE

After the client and the server agree on the key parameters, including the DH group, the server initiates a DH exchange.

Let a and A be the private and the public key of the server, and likewise b and B the private and public key of the client. Further, let p be a publicly known prime and g an integer smaller than p. p and g are predefined by the chosen DH

[2] Elliptic Curve Groups modulo a Prime

group.

The server provides the client with its public key $A = g^a \bmod p$ along with the chosen DH group. With that information, the client can compute its public key $B = g^b \bmod p$ and transmit it to the server. Both parties can now compute a shared session key using $K = B^a \bmod p$ (on the server side) and $K = A^b \bmod p$ (on the client side).

Only A, B, and the DH group (defining p, g) are transmitted in clear over the wire. Ultimately, both parties know the secret symmetric session key K. If at least one of the private keys (or the nonce) is a fresh random integer, the generated session key K will not repeat.

2.6.1 Optimization through Precomputation

A server can precompute A from a (temporarily) fixed private key a for each DH group, as it is independent of the client. This is a valid approach if the rekeying period is considerably less time than it takes a potent attacker to crack those keys, and if the client doesn't follow a similar strategy.

However, as pointed out by Flesh et al. [25], those a keys should not be shared between different DH groups. Otherwise, an attacker could crack a on a weak DH group and use it for stronger ones.

2.7 VoWiFi Provisioning Ecosystem

The 3GPP VoLTE/VoWiFi ecosystem lacks a comprehensive autoconfiguration or provisioning protocol (similar to USIM files or MIB/SIB announcements used for other cellular parameters). This has caused (and still causes) massive compatibility problems for operating VoLTE on handsets [40]. The modem and mobile OS vendors help themselves by preloading configuration databases for known MNOs in their firmware and OS images. Those configurations define a multitude of properties, from the bearer and tunnel settings down to IMS/SIP codec parameters. Some operators use an app with operator privileges to push a configuration onto the device.

The GSM Association (GSMA) approaches the problem in three ways. First, it created a database[3] as a paid service for use by manufacturers. Second, it created a small set of standard configurations (e.g., to ease VoLTE roaming). Third, they recently started a new Internet-based configuration service under `aes.mnc`⟨*id*⟩`.mcc`⟨*id*⟩`.pub.3gppnetwork.org`. At the time of writing, only 66 operators registered that domain.

2.7.1 Apple iOS

Independent of the used modem chipset (i.e., Qualcomm or Intel), Apple organizes country-specific and operator-specific configurations into `.ipcc` files (called Country Bundles and Carrier Bundles, respectively). They can be distributed via the iOS system image and system updates as well as via itunes.com. *ipcc-downloader*[4] extracts them from latter source.

2.7.2 Qualcomm: Xiaomi, Oppo

Qualcomm uses proprietary encoded binary `.mbn` files to load carrier-specific modem configurations (also called `MCFGs`) into their modems. These configuration files can be extracted from the modem image (often named `NON-HLOS.bin`) that is part of the smartphone ROM.

There have been efforts from the open source community towards providing tools to inspect the loaded configuration settings of a smartphone (e.g., *EfsTools*[5]) and to sideload configurations from other smartphones with similar chipsets (e.g., to enable VoLTE support on non-carrier-branded devices). The VoWiFi-related settings are located within the `/data/iwlan_s2b_config.xml` file of the unpacked configuration tree.

2.7.3 Samsung

The VoWiFi configuration on Samsung devices can be found within the `/system/etc/epdg_apns_conf.xml` file on the smartphone. We believe these settings are also used with other modem chipsets on Samsung devices since the file also exists in ROMs for MediaTek- and Qualcomm-equipped models. In contrast to the OEMs mentioned above (e.g., Xiaomi), the Qualcomm-based Samsung devices do not contain additional `.mbn` modem configurations in their modem image.

2.7.4 Google Pixel

Google Pixel generations up to the Pixel 5 used a Qualcomm chipset, utilizing the Qualcomm configuration approach described above. Starting with the Pixel 6, Google introduced its own Tensor-based SoCs, where VoWiFi-specific configuration parameters are consolidated into Android-generic *Carrier Configuration* settings[6]. However, inspecting the publicly accessible operator-specific configuration files[7] shows that the responsible `iwlan` settings are not used in practice. Besides shipping *Carrier Configurations* via the Android-wide presetting, operators can change these settings via their own carrier app (to gain *Carrier Privileges*, an app needs to be signed with a specific certificate that is saved on the SIM card). In practice, many modern Pixel phones fall back to the default values (defined in the Android source code[8]).

[3] https://imeidb.gsma.com/nsx/index

[4] https://github.com/mrlnc/ipcc-downloader

[5] https://github.com/JohnBel/EfsTools

[6] https://source.android.com/docs/core/connect/carrier

[7] https://android.googlesource.com/platform/packages/apps/CarrierConfig/+/main/assets

[8] https://android.googlesource.com/platform/frameworks/base/+/refs/heads/android14-release/telephony/java/android/telephony/CarrierConfigManager.java#9099

3 Threat and Attacker Model

Goals From an adversary's perspective, three main goals motivate an attack:

- G1 Eavesdropping on private communications (e.g., extracting the signaling or voice channel of realized calls or spying on sent SMS messages).
- G2 Using the trusted communication channel as an attack vector towards the phone (e.g., by injecting maliciously formed SIP messages as seen in the recent Exynos vulnerabilities [13]).
- G3 Injecting actions towards the provider (e.g., spoofing SMS messages to impersonate the user or monetizing the exploit by calling value-added numbers) or EPC access in general.

Capabilities Traffic interception and modification can happen at any point over the Wi-Fi (e.g., via ARP/RA spoofing, the Wi-Fi access point (e.g., from a hotspot operator), or while on Internet transit.

For some of the presented attacks, we further assume a determined attacker with the capability to break $DH1^{768}$ or even $DH2^{1024}$, according to Adrian et al. [15].

Criteria If both the server and client support those weak DH groups and actually use them (either by tricking them or by default config), those VoWiFi tunnels into the EPC would be vulnerable.

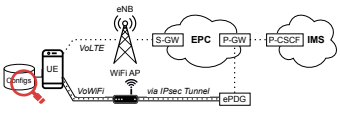
Further, any divergence from the privateness (secrecy) of a *private key* constitutes a broken encryption.

4 Methodology

To explore the VoWiFi landscape and answer RQ1-3, we had to approach it from multiple vantage points: a) We examine the different client-stored operator configurations. b) We test the configuration of commercial network operators worldwide and their corner cases. c) We examine real UE-operator interaction by observing and modifying traffic. In the Discussion in Secion 9, we then contrast those results.

4.1 Static Client-Side Configuration Analysis

To approach RQ1, we chose a static configuration analysis since otherwise, we would need a valid SIM card for each operator.

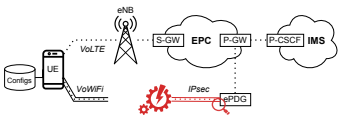


As stated in Section 2, critical information about the VoLTE and VoWiFi data bearer (or tunnels), as well as the IMS settings, need to be known to the UE in advance for each MNO. In lieu of a 3GPP autoconfigure protocol, a database of known settings for each operator is preloaded to the device. Due to different VoLTE and VoWiFi implementations – depending on the OS, OEM, and modem chipset manufacturer – there is no standardized way to access these settings. Thus, we extract those settings from Apple, Qualcomm-based, Tensor-based (Pixel), and Exyonos-based (Samsung) phones separately, roughly following the market share [14].

We identified the following interesting parameters: 1) the key exchange methods (e.g., Diffie–Hellman groups), 2) rekeying timers, and 3) encryption, integrity algorithms & pseudo-random function (PRF).

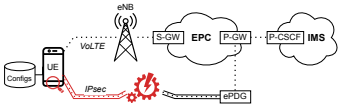
4.2 Active MNO-Side ePDG Scanning

To answer RQ2, we have to analyze operators' IKE handshakes and probe different key exchange methods. First, we query all possible DNS names (described in Section 2), resolving DNS requests in an iterative manner (i.e., getting all the IP addresses from authoritative servers). For each operator, we try to negotiate an IKE phase 1 key with every key exchange method from Table 1 separately – i.e., we pretend to support only one method at a time. For all operators, we record how their servers react and whether they propose a different DH group (if so, which one) or accept the client's choice for a stronger one. Additionally, we test if the server tolerates the client's choice for a weaker DH group, even if both parties announce support for a stronger one. This would ease downgrade attacks to an attackable bit length.



4.3 Testing Implementations and Composition

To answer RQ3, we need to combine multiple results from the client and the server side to form a view of the complete system and its security properties.



4.3.1 Downgrading Possibilities

We examine if phones from different manufacturers accept weaker DH groups even if higher ones are available, with an active test against the phone. We also want to know if phones support any undocumented DH groups. To this end, we redirect real traffic between our phone and the ePDG to a script that allows us to intercept and rewrite data.

Further, we test if networks would accept a weaker DH group despite both parties indicating support for higher groups. To this end, we probe each operator's ePDG but announce multiple methods at once.

4.3.2 Interception Opportunity

When attacking the key, the attacker has to outrace the rekeying period of connections, i.e., crack the key before it loses

validity. We extract those settings from the phones. Putting both sides together, we also consider key value re-usage and undocumented DH groups.

4.4 Limitations

4.4.1 Limited MCC-MNC Mapping

One operator can have multiple MCC-MNC designations, for example, because of past mergers. Likewise, one MCC-MNC tuple can (but does not have to) be shared between multiple virtual network operators. This depends on whether the MVNO operates its own Home Subscriber Server (HSS). MVNOs can also contract services, i.e., share the same physical servers with another operator.

We refrained from error-prone manual disambiguation. Thus, unless otherwise stated (for a very particular vulnerability), results operate on a one-⟨MCC-MNC⟩-tuple-per-operator approximation.

4.4.2 Consistent Configuration

For resilience and load balancing, operators could either explicitly or invisibly operate multiple servers/gateways, some of which could have diverging configurations. Explicit load balancing is externally visible, e.g., via DNS round robin, while a dedicated load balancer would conceal load balancing from the outside world (it has only one IP address). Unless otherwise stated, we assume consistent configurations within an operator for our measurements and results.

4.5 Ethical Considerations

Since some parts of this paper include measurements on real-world production provider infrastructure, we assessed its necessity and the least invasive method to perform the investigation.

Invasiveness. We always measured properties with our devices or in sandboxes unless the research subject required real-world data.

Traffic and Server Load. Connections to production systems were of low volume and with the minimum number of connections necessary for the task. Since those systems (e.g., ePDG) are made to handle traffic for (millions) of customers, we are confident that our attempts did no harm.

Traffic Abnormality. Our handshake attempts for the ePDG always confirmed the appropriate RFC format and never contained illegal data or malformed structures.

Confidentiality and Integrity. Our handshake attempts never contained real credentials and, therefore, should never have access to any privileged functions of confidential data.

5 Static Client-Side Configuration

To cover a considerable share of real-world client configurations, we analyzed different implementations and extracted the corresponding settings for the available operators out of smartphones and smartphone firmware images that reflect the current market situation [14]. While not part of our threat model, we additionally extract encryption, integrity and pseudo-random function (PRF) algorithms alongside DH groups and rekey timers for the IKEv2 security association parameters and evaluate their prevalence.

5.1 Implementation

After downloading and extracting the available **Apple iOS** carrier bundles (Section 2.7.1), we filter for iPhones (discarding other device types such as Apple watches) and group them by operator. For our statistical analysis, we select the latest VoWiFi configuration for each operator.

Configurations for **Qualcomm**-based phones, such as Xiaomi and Oppo, use the Qualcomm `.mbn` mechanism as described in Section 2.7.2. Leveraging the information from other open-source projects, we implemented a parsing tool[9] to unpack and parse the modem configuration files. We analyzed configuration files from the Xiaomi 13 Pro (2023-08-22) and the Oppo X6 Pro (2023-12-06).

In the category of **Samsung**'s VoWiFi configurations (see Section 2.7.3), we analyzed the most recent (2023-12-29) configuration file from the Exynos-based Galaxy S24+.

As **Google Pixel** phones have multiple ways to receive carrier configurations, we used them primarily to extract Android 14 default values.

IKEv2 Default Values We focused our client-side analysis on operator-specific settings, overriding the default state. However, operators may also refrain from providing specific values, leading to a fallback to a predefined default. Additionally, these default settings are also used for operators that are not part of the preloaded configuration files at all (e.g., smaller mobile virtual network operators, MVNOs).

In our static analysis, we were able to recover the default settings for Samsung devices (cf. Section 2.7.3) and for newer Pixel phones (cf. Section 2.7.4).

5.2 Results

Table 3 compares the presence of operator-specific VoWiFi settings in our analyzed client configuration files. The percentage column shows the share of operators that actually provide dedicated VoWiFi settings. Figure 4 shows the prevalence of the different DH groups in the analyzed client configurations. We see that on the client side, DH groups with larger key sizes have not reached widespread support yet. Within all analyzed

[9] https://github.com/sbaresearch/mbn-mcfg-tools

Table 3: IKEv2 security association parameters inside static UE configurations

Vendor		Apple	Xiaomi	Oppo	Samsung
Configs	DH Group	219 (29%)	150 (56%)	221 (59%)	156 (49%)
	Rekey Timer	219 (29%)	231 (86%)	340 (90%)	95 (30%)
	Encryption	219 (29%)	126 (47%)	211 (56%)	141 (44%)
	Integrity	219 (29%)	130 (48%)	212 (56%)	141 (44%)
	PRF	219 (29%)	120 (44%)	203 (54%)	0 (0%)
Total MNO Configs		745	270	377	319

device groups, only a single operator (T-Mobile Germany) on Samsung devices signals support for an elliptic curve group (i.e., $DH19^{256}_{ECP}$).

5.2.1 Apple iOS

Of a total of 745 operator-specific `.ipcc`-configurations for iPhone devices, 219 specify VoWiFi-related settings. The remaining 526 operators either do not support VoWiFi yet or rely on the device's default configuration.

Analyzing the operator-specific VoWiFi settings for iPhones, we discover two properties:

1. While other vendors (e.g., Qualcomm, Samsung) usually define a broad set of supported security parameters, Apple, with the exception of three MNO configs, only defines a single algorithm setting for each VoWiFi-related attribute. Thus, on the network, it just signals support for one single DH group. The same holds true for the other configuration settings (e.g., rekeying or encryption and integrity algorithms).
2. Whenever one IKEv2-related parameter is set for an operator, the configuration also contains all the other parameters. (i.e., the `.ipcc`-configurations always contain complete settings). This can be seen in Table 3, as all columns contain the same percentage values (i.e., 29%).

Due to these properties, Figure 4 shows that iPhones exclusively support (never 3GPP-standardized) $DH1^{768}$ to connect to 9% of the analyzed operators.

5.2.2 Android

Qualcomm `.mbn` files can be deployed and adjusted both by the chipset vendor and OEMs, leading to a different number of `.mbn` files for Xiaomi and Oppo. Our analyzed Xiaomi device includes 270 `.mbn` files, compared to 377 for the selected

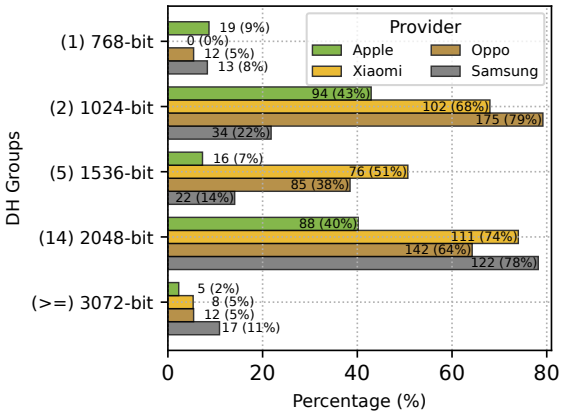


Figure 4: Number of MNOs per supported DH group (client side, grouped by device type).

Oppo smartphone. While MBN files are more likely to include VoWiFi-specific settings, they do not always specify all parameters (as opposed to Apple, Section 5.2.1). Nearly every `.mbn` file includes a rekey timer and differentiates between soft and hard timers. The soft timer specifies the number of seconds until the client tries to renew the corresponding SA. The hard timer states the maximum lifetime of an IKEv2 SA. If only one timer is specified, as is the case for Samsung, it represents the hard timer, thus the total lifetime of the SA. In contrast to the rekey timer only half of the `.mbn` files include SA parameters such as the DH groups, encryption, integrity, or PRF algorithms.

5.2.3 Default (Fallback) Values

As described in Section 5.1, we extracted the default values for Samsung devices and recent Google Pixel phones. Since there were no IKEv2-specific SA parameters available within our default Qualcomm `.mbn` profiles, the used settings are taken from the modem's default (defined in even deeper layers of the modem firmware). To gain comparable settings for Qualcomm, we thus extracted the proposed values from an active capturing of our lab's Qualcomm-based Xiaomi device (the Xiaomi Poco X3 NFC, using the Snapdragon 732G) when no specific carrier `.mbn` file was loaded. We list the default values in Table 2. As the table shows, Samsung only sets

Table 2: Default parameters for IKEv2 if no MNO-specific configuration is present.

Vendor		Qualcomm (Xiaomi†)	Samsung	Google Pixel
Defaults	DH Group	$DH2^{1024}$, $DH5^{1536}$, $DH14^{2048}$	$DH2^{1024}$	$DH2^{1024}$, $DH5^{1536}$, $DH14^{2048}$
	Rekey Timer	64,800s (soft), 64,900s (hard)	86,400s	7,200s (soft); 14,400s (hard)
	Encryption	$AES\_CBC^{128,256}$, 3DES	AES_CBC^{128}	$AES\_CBC^{128,192,256}$
	Integrity	$SHA1^{96}$, AES_XCBC^{96}, $MD5^{96}$	$SHA1^{96}$	$XCBC^{96}$, $SHA1^{96}$, $SHA2^{256,384,512}$
	PRF	$SHA2^{256}$, SHA1, AES^{128}	*	SHA1, AES_XCBC^{128}, $SHA2^{256,384,512}$

* If no PRF is set, the PRF can be derived from the integrity algorithms. † Xiaomi Poco X3 NFC deprecated DH [23]

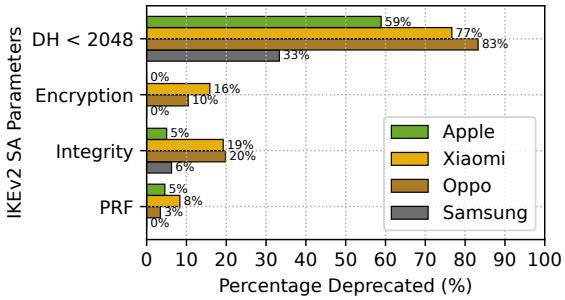


Figure 5: Share of deprecated IKEv2 parameters within all operator-specific VoWiFi settings, i.e., 83% of Oppo's configured DH settings include a deprecated DH group.

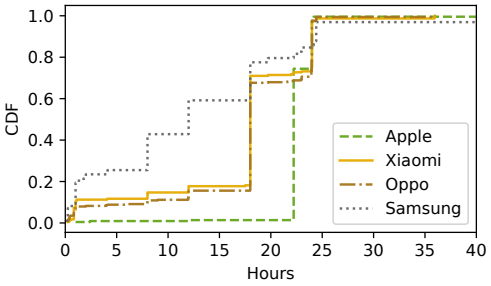


Figure 6: Operator-specific configuration of rekey timings. The majority of Apple devices is configured to renew the keys after 22 hours. For Xiaomi and Oppo, the graph represents the configured soft timers (peaking at 18 hours).

one specific value for each IKEv2 parameter category (similar to the operator-specific behavior observed for iPhones). In contrast, our Xiaomi and Google Pixel devices propose various settings to the server endpoint.

Both Xiaomi and the Pixel phone default to the first (and weakest) setting and propose a $DH2^{1024}$ key exchange within the first `SA_INIT` handshake packet. Note that the initial DH client preference is not relevant for active attackers, because it can be arbitrarily switched by sending an IKEv2 protocol extension packet to the client (as described in Section 7.3 towards the end of the paper).

In the first packet (`SA_INIT`), the UE has to choose a DH group from the list. During our test, the first and weakest DH group $DH2^{1024}$ was chosen.

5.2.4 Deprecated IKE Parameters

Our static analysis of IKEv2 security parameters on the client side shows an alarming share of deprecated algorithms. Figure 5 shows the deprecation share by each IKEv2 security algorithm group and device type.

$DH2^{1024}$ is the most dominant group among the deprecated groups, but also among all measured groups in total for many devices (e.g., Apple, Xiaomi, and Oppo). It is also the go-to fallback value for many configurations (cf. Table 2).

Regarding encryption and integrity, many clients still support the deprecated DES and MD5 algorithms. Table 4 in the Appendix lists decrepated IKEv2 SA algorithms.

Note that Figure 5 only shows the results of the operator-specific settings, not considering default values. For example, Samsung only shows a DH group deprecation of 33%. However, only 49% of the operators override the default value (cf. Table 3); thus, in practice, the deprecated $DH2^{1024}$ group is used as a fallback in many real life scenarios.

5.2.5 Key Lifetimes

The key lifetime on the IKEv2 layer essentially defines the available timeframe for cracking the key. From a security perspective, shorter lifetimes (and, obviously, strong DH groups) are recommended to extend the time and resources needed to crack the key.

Figure 6 shows the rekey intervals set by each vendor. In almost all cases, re-keying takes place within a 24-hour time frame. 40% of Samsung devices tried to rekey in the first 10 hours, while most iPhones rekey after 22 hours, which should leave enough time to be in reach for nation-state attackers. We observed three outliers inside Samsung's MNO configurations that specify a key lifetime of a year. The 3GPP specification [23] does not give recommendations for rekey timers, which ultimately delegates the decision to the operators.

5.2.6 Client Side Validation (Sanity Check)

We used a random sample (n=12) of available smartphone devices (i.e., all testing devices from our lab and some additional models from volunteers that allowed us to record the IKEv2 handshake from their regular smartphone) to do a sanity check and verify whether the obtained results from our static analysis are feasible. Our selection covers every device group from our static analysis with at least one model (i.e., using iPhones, Qualcomm-based devices, Samsung models, Google Pixel, and additionally, several MediaTek-based devices). Although the extracted IKEv2 proposals from our captures are biased towards operators from our home country Austria, we used them as a sanity check to verify the results from our static analysis. For all devices that matched the exact models from the configuration file analysis (i.e., iPhone and Google Pixel), we were able to verify the obtained results, i.e., the proposals were identical to the settings in the configuration files. Moreover, the residual devices from our sample also showed a similar distribution (e.g., $DH2^{1024}$ being the most popular DH group).

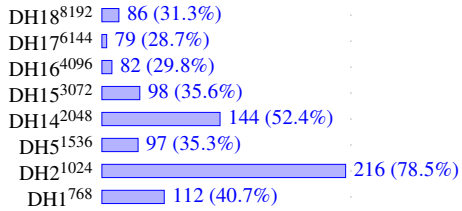


Figure 7: Number of MNOs per supported DH group

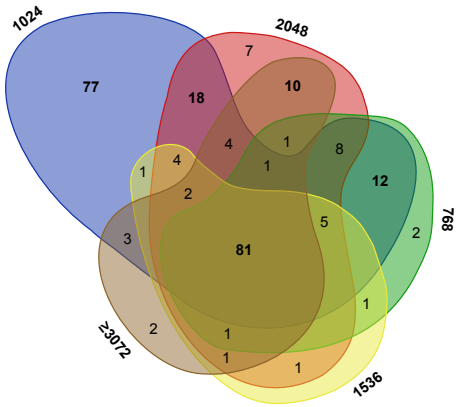


Figure 8: Number of MNOs that support a specific combination of DH key exchange groups. 3072-8192 bit groups are combined because of their low diversity.

6 Active MNO-side ePDG Scanning

6.1 Implementation

To analyze operators' IKE handshakes and probe different key exchange methods we operated as follows. First, we queried all possible ePDG DNS names (Section 2.2) with *massdns*[10], delegating all queries to a local *unbound*[11] instance, iteratively resolving DNS requests (i.e., getting the IP addresses from the authoritative server). Afterward, our Python-based IKEv2 implementation tried to negotiate a key with each of the methods from Table 1 with every operator. Our implementation[12] is based on predefined packet structures from *scapy*[13] and was verified against a self-hosted *strongSwan* server. For each tested operator, we recorded the server's answer, including any optional DH group suggestions, the public key value, and additionally the whole interaction as a PCAP file. Additionally, we tested if the server tolerates a client's choice of a weaker DH group, even if both parties announce support for a stronger one. This would ease downgrade attacks to a feasibly attackable bit length.

6.2 ePDG Supported Key Exchange Methods

As of Q4 2023, operators maintained 423 ePDG domain names (minimum one A record, of which 16 additionally provided `AAAA` records). Of these 423 operators, 275 responded to our handshake, of which 33 rejected all of our proposed key exchange methods. We suspect that some might have geoblocked their VoWiFi services to prevent roaming evasion or for increased security.

6.2.1 MODP Groups

275 ePDG servers responded to our handshake attempts. By offering only one DH group, we tested the servers' capabilities. Some servers tend to ignore requests with unsupported groups, while most reported a handshake error; none proposed a downgrade. As depicted in Figure 7, 79% support DH2^{1024}, followed by DH14^{2048} with 52%, and DH1^{768} with 41%.

Figure 8 shows the combinations of supported methods per operator. Only two operators solely support DH1^{768}, and 77 only support DH2^{1024}. The former was never proposed by the 3GPP for usage [23]. 12 and 18 operators support combinations of DH1^{768}+DH2^{1024} and DH2^{1024}+DH14^{2048}, respectively. Once an operator chooses to support DH15^{3072}, it usually supports most of the groups up to DH18^{8192}. 65 operators supported all groups from DH1^{768} up to DH18^{8192}.

6.2.2 ECP Groups

Except for one private operator, there is no support for elliptic curve groups. 13 operators proposed a downgrade to DH1^{768} in their response, even though all of them support up to DH18^{8192}. Two operators proposed a downgrade to DH14^{2048}. All others (including T-Mobile Germany, which signals ECP support in the client-side config on Samsung devices) either ignored the handshake, returned a negative answer, or reported an error. Thus, in practice, ECP appears to be rarely used for real-world VoWiFi connections.

6.3 Tolerating Weak DH Preferences

We want to know if ePDGs tolerate weaker DH groups than their common set of supported methods allows for. In this test, our client connected, indicating support for all DH groups, but chose DH2^{1024} as the preferred one.

41% of the operators accepted the proposed less secure method, 12% returned an error without indicating which group to choose instead, but 42% desired an upgrade by the client. Roughly half of them chose DH18^{8192}, most of the others DH14^{2048}, with a few single-digit outliers requesting DH15^{3072}, DH16^{4096}, and DH5^{1536} (in descending order).

Curiously, 4% seemed to indicate a desired downgrade to DH1^{768}. However, as all those networks actually support our

[10] https://github.com/blechschmidt/massdns
[11] https://github.com/NLnetLabs/unbound
[12] https://github.com/sbaresearch/vowifi-epdg-scanning
[13] https://github.com/secdev/scapy/blob/master/scapy/contrib/ikev2.py

proposed DH2[1024], this probably represents a generic *make it work at all costs* error message (e.g., if some of the higher groups are not recognized, similar to what we have seen with ECP groups).

6.4 Inter-MNO Static Key Sharing

In our scans, initially[14], 14 ePDG servers (12 operators, based on IP addresses and background story, see Appendix C) showed a very peculiar behavior: They repeatedly served the same keys. A repeated scan with apx. 500 DH2[1024] handshakes on those MNOs revealed a globally shared set of exactly ten static public keys randomly used on each connection attempt by every one of those operators.

However, this, in return, means that those 12(+1) operators all use the same ten private keys. Violating the secrecy requirement of the *private key* allows any of those operators (or anyone else who seizes the keys from them legally or through other means), as well as the originator of those keys, to decrypt any other operators' shared session secrets instantly.

Using the notation from Section 2.6, if an attacker can read the plaintext DH group and B from the wire and knows one of the private keys (in our case a), the secret session key can be reconstructed using $K = B^a \bmod p$.

In a smaller sample, we also confirmed that similar sets exist for other DH groups on the same operators. As the key a is independent of the DH group, an attacker who does not know the private key A can crack the weakest group DH1[768] and then use it to reconstruct K generated for the stronger groups.

Using passive banner analysis with Shodan[15], we confirmed that at least three of those ePDGs are from ZTE (the others were firewalled). For all MNOs (except for one), press releases show contracts with, winning bids by, or strategic cooperation with ZTE to build an LTE or 5G network. Eight of those networks are located in Asia, three in central Europe, and two in South America.

Without knowledge about how those operators arrived at using the same static set of then non-randomized keys, we initiated a responsible disclosure with the GSMA. The process, the manufacturer's response, and a list of key hashes are to be found in Appendix D.2. We later found that the same ten keys are also used for the phase 2 (L2) key exchange.

6.5 Intra-MNO Key Reusage

We also encountered MNOs that reused keys between handshakes. If handled carefully, this can be a valid optimization on the server side as described in Section 2.6.1.

We have also encountered rare instances of nonce reuse, which violates the IKEv2 specification and also defies the common definition of *number used once*.

[14] After the manufacturer provided a fix, a 15th ePDG/13th MNO appeared.
[15] https://www.shodan.io/

7 Downgrading Vulnerabilities

Based on the results from the above sections, we devise experiments to assess and test the resilience against downgrade attacks. As per our threat model from Section 3, a downgrade to a sufficiently weak key exchange method is considered a successful attack.

7.1 Implementation

As described in the threat model in Section 3, a user's traffic can be intercepted locally (e.g., by a malicious WiFi operator), anywhere on the path, or on a large scale (e.g., by a nation-state monitoring an IXPs traffic). To simulate these threats, we set up a Wi-Fi AP (monitoring the occurring traffic with `tcpdump`) and use it as an Internet uplink for off-the-shelf smartphones equipped with SIM cards of commercial operators within our home country.

For invasive traffic-altering attacks, we devised `iptable` rules that forward the corresponding packets to our MitM (Monster in the Middle) script. For the traffic rewriting we reused the Scapy-based implementation of our server-side scanning solution.

7.2 Outdated Software

While preparing the exploit chain and testing the setup described above, we identified that Samsung and (some) MediaTek-based devices use strongSwan[16] as a foundation for their VoWiFi support.

While Samsung uses a recent version of strongSwan (i.e., version 5.9.8 for the Galaxy S24+), the `charon` binary of our MediaTek device (i.e., the Xiaomi Redmi A1) identifies itself to be part of strongSwan 5.1.2, released in March 2014.

7.3 Pivoting DH Groups via `INVALID_KE`

Whenever a client connects to an IKEv2 server, it has to communicate its supported SA (Security Association) parameters within the `SA_INIT` packet. While it has to decide on a specific key exchange method (i.e., DH group), it can also signal support for other groups within its proposal. The server can then either accept the proposed key exchange method or switch to another offered group by sending an `INVALID_KE` (invalid key exchange) message. This message can carry the server's proposed method. The client then retries and sends a fresh `SA_INIT` packet with the chosen key exchange, as shown in Figure 9. While the proposed SAs within the `SA_INIT` packet are normally protected against rewriting attacks by subsequent integrity checks, downgrading by the `INVALID_KE` message is possible because the client discards its current state and starts from scratch with the indicated key exchange.

[16] https://github.com/strongswan/strongswan

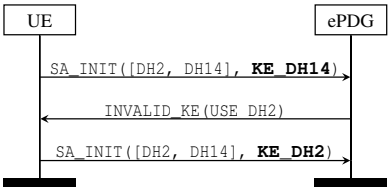


Figure 9: An ePDG server can switch from the initially selected DH group ($DH14^{2048}$) to a different group that is offered by the client within the proposal ($DH2^{1024}$).

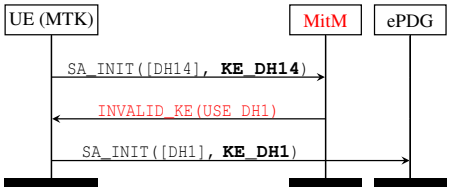


Figure 10: Some of MediaTek's Dimensity basebands are vulnerable to severe downgrade attacks, allowing the selection of DH groups that are known to be weak and were never part of the initial IKE_INIT-proposal nor 3GPP specification.

Thus, an active attacker can suppress the first SA_INIT message and send the client a spoofed INVALID_KE packet proposing a lower DH group and effectively, downgrading the key length. Clients supporting weak DH groups and servers tolerating insecure proposals (without demanding the client to switch to a stronger group if available) facilitate this kind of downgrade attack.

7.3.1 Results

Properly implemented clients that propose multiple SAs are prone to this attack. For example, since Apple devices only announce a single DH group within their client-side proposal, they are not vulnerable to this kind of downgrade attack. The same holds true for other scenarios where the client explicitly uses a single DH group (e.g., devices using Samsung's default configuration that is shown in Table 2). However, as our client-side analysis showed, most devices are overprovisioned with multiple DH groups, and their settings include deprecated groups. For all those devices in our sample, we successfully switched the used key exchange to the weakest offered group using the attack described above – if they were not already making the weakest selection their default anyway.

In context with our results probing the ePDGs, where we have seen that at least 41% of the operators tolerate weak client preferences over stronger available DH groups, we can conclude that it is feasible to execute this attack under real-world circumstances.

7.4 MediaTek Implementation Bug

Besides testing our available UEs for (maliciously played) *protocol-conform* downgrade attacks (as described above), we also tested whether the INVALID_KE message is properly implemented. Specifically, we test how the client will react to the server's (or spoofed) request to switch to a DH group not part of the preloaded configuration.

7.4.1 Results

While all devices behaved as expected (rejection of the offer), some MediaTek-based devices stood out.

We found that there are at least two different IPsec implementations for VoWiFi support on MediaTek devices. The first one (presumably older devices with a Helio chipset) uses strongSwan (based on strongSwan-related configurations in the firmware images). In contrast, newer Dimensity-based devices lack the necessary strongSwan files and thus presumably use a different IPsec/VoWiFi stack.

Our active measurements on multiple models reveal that the latter MediaTek devices are not only *vulnerable* to the attack described above, but also accept INVALID_KE fixations to any DH group (i.e., also to a group that was not part of the UE's proposal). In practice, an attacker can thus downgrade to the weakest $DH1^{768}$, as shown in Figure 10. According to our threat model and Adrian et al. [15], we consider this breakable by well-funded academic researchers and certainly within reach of resourceful (not necessarily nation-state) actors. We want to emphasize that the two downgrade attacks described above work on unmodified, unrooted smartphones within commercial networks.

7.5 Responsible Disclosure

We disclosed those vulnerabilities to MediaTek and a fix is available (Appendix D.1).

7.6 Escalating (L1) Attacks

This Section gives some context, on how (L1) attacks can be facilitated to gain control over the full IKE/IPSec/SIP stack. The key observations are:

(L1) downgrading drastically eases key recovery Even a downgrade to $DH1^{768}$ (Section 7.4) still needs massive computing power to be cracked, but is considered in reach for several years now [15].

Regular Rekeying without Reauthentication IKE's keys are regularly regenerated using a DH exchange based on the selected lifetime (see key lifetime analysis in Section 5.2.5). An attacker can hijack the rekeying as shown in our experiment in Appendix B. However, no authentication is performed

on (L2) making it roll over into the next session key.

(L2) Child SA has no integrity protection It relies on (L1) for providing all of the integrity protection. Since there is no subsequent reauthentication, cracking the outer key exchange is enough to gain stealth rewriting capabilities within the first two layers.

(L3) SIP encryption is optional and not enforced As shown in Experiment Appendix A, SIP encryption is considered optional by most providers.

7.6.1 Full Attack Outline

1. An active attacker inhibits the first `INIT_SA` message from the client to the server and proposes a weak DH group (as described in Section 7).
2. From now on, the attacker lets the client and server handshake the weaker DH group, authenticate the connection (via EAP-AKA), and create a session key for (L2).
3. The attacker can now race to crack the outer key exchange and thus gain rewriting capabilities on (L1) – before the key lifetime expires and a rekeying is triggered. Note: the attacker does not have (L2)'s session key yet.
4. If or when the time comes[17] (Section 5.2.5) for a rekeying of (L1), the attacker can handshake both sides independently and inject themselves in between.
5. Similarly, when (L2) is rekeyed, the attacker can handshake both sides independently and inject themselves in between without needing authenticating. Note: The attacker now has also control over (L2).
6. As encryption of the SIP connection (L3) is optional (Appendix A), an attacker also likely gains control over the client' authenticated IMS session.

8 Related Work

Encryption in Cellular Networks Cryptographic problems have plagued cellular networks from the start. More recently, Yomna et al. [37] presented Android's approach to combat so-called *null ciphers*. Null ciphers are mock ciphers that can be inserted into the encryption stack in case no actual encryption is desired. Cholesta et al. [18] put European networks to the test - many of them still accepted null ciphers on the radio layer. Tsay and Mjølsnes [44] found impersonation vulnerabilities in the Authentication and Key Agreement Protocols (AKA) in UMTS and LTE. Rupprecht et al. [41] categorized past cellular network vulnerabilities to identify classes of errors and how to combat them.

[17]The standard allows both peers to trigger a rekeying prematurely, but we have not tested that.

Diffie-Hellman Groups and IKE In *Imperfect Forward Secrecy*, Adrian et al. [15] show all the little ways in which DH implementations fail in practice. Bhargavan et al. [16] try to answer the question of how to support reconfigurability while at the same time guaranteeing the preferred mode is negotiated. Felsch et al. [25] reports on Bleichenbacher attacks on IKEv1 and IKEv2.

Evaluating Real-World Security Configurations Hue et al. [31] evaluated both client- and server-side WPA2-enterprise configurations for education institutes (e.g., eduroam), uncovering deprecated settings and suspected private key sharing across different institutes. Valenta et al. [46, 47] performed Internet-wide scans for TLS, SSH, and IPsec, surveying their elliptic curve usage and improper curve validation. Heninger et al. [30] analyzed the occurrence of weak (factorable) keys in the wild.

Evaluation of VPN Servers Maghsoudlou et al. [36] executed Internet-wide scans to discover and fingerprint VPNs, finding over 7 million IPsec servers. Kahn et al. [35] and Ramesh et al. [39] made large-scale measurements in the commercial VPN ecosystem exposing leaked user traffic. Wu et al. [49] investigated academic VPNs, which have become an integral part of the home office life.

Roaming Experiments and Large-Scale Cellular Measurements Sahin and Francillon [42] observed hijacked and thus monetized voice calls being redirected to over-the-top (OTT) services (e.g., WhatsApp, Viber). Gegenhuber et al. [28, 29] introduced a measurement platform enabling scalable cellular measurements by tunneling the communication between SIM card and modem over the Internet. Besides measurements on the radio layer, Gegenhuber et al. [26, 27] also evaluated global VoWiFi deployments, exposing geoblocking practices at VoWiFi by simulating clients from different countries.

SIP in VoLTE and RCS Tu et al. [45] uncovered spoofing and injection vulnerabilities at VoLTE's SIP layer. Similarly, Yang et al. [50] exposed weaknesses in real-world RCS deployments. In 2023, the Google Project Zero team discovered four severe Exynos vulnerabilities, including a remote code execution on the most recent Pixel and many Samsung baseband processors by injecting malicious SIP messages [13] into the VoLTE/VoWiFi traffic.

9 Discussion

This paper set out to cartograph the state of VoWiFi on the UE and MNO side. Little did we know what awaited us. The ecosystem is haunted by multiple structural and standardization problems:

a) an inadequate and slow process of provisioning provider settings to the UEs, with too many middlemen,

b) structural disincentives for phasing out deprecated cryptography and a naïve standardization approach,

c) optional encryption in certain parts of the ecosystem and the prevalence of the dumb client paradigm,

d) critical bugs on the UE and MNO side.

9.1 Provisioning and Configuration

The missing VoLTE/VoWiFi autoconfiguration feature inspired handset manufacturers to find (non-interoperable) ways to preload known settings and curate their own databases (RQ1). It is a painful, tedious task for the operators and the handset manufacturers alike, with multiple middlemen that do not inspire quick, painless updates to new settings, disincentivizing updates.

This is represented in the very inconsistent settings among the different vendors (Section 5) and the large adoption of deprecated DH groups in provider-specific settings (Section 6) as well as default settings, as seen in Figures 4 and 5.

Ultimately, MNOs should have the power to make configuration changes, including removing deprecated cryptographic algorithms without impairing service, if they wish to.

9.2 Structural Hurdles of Deprecation

The MNOs have little to no incentive to phase out older insecure key exchange methods (RQ2). On the one hand, (anticipated) compatibility issues with legacy devices and the slow update process might stoke sentiments against changes. In our sample, only 7% of operators ditched all the insecure DH groups below 2048 bits.

On the other hand, 3GPP/ETSI lacks a defined depreciation path. Just removing it from the standard does not actually remove the method from the world nor the affected devices.

In standards, there is no room to be stingy on the number of key bits. If anything, it is the place to be bold and visionary. If shorter lengths are required at the start, a stringent phase-out plan/process should be defined with it. The development of computing power turned out to be somewhat predictable, and the same can be expected for the deprecation of key lengths.

9.3 Optionality and Strict Configurations

The *dumb client* paradigm, often found in large infrastructure, envisions the majority of decisions to be made by the network and not the client.

Using SIP encryption? If the network does not mandate it, the client will definitely not object.

However, the VoWiFi ecosystem, which is built upon many Internet technologies, has the infrastructure and protocolary means for clients to request better settings at their discretion. UE chipset and operating system manufacturers should take this chance.

Furthermore, the data suggest the importance of those preloaded configurations might also be simply overestimated - as seemingly conflicting carrier configurations from different handset vendors still work, and 42% of the operators request an upgrade of the key exchange method if a common higher group is available. 3GPP, the operators, and handset/baseband manufacturers should trust more in autoconfiguration measures (or enforce their own minimum standards), even at the expense of slightly longer connection times.

9.4 Downgrades, Bugs, and Vulnerabilities

Attacks (Section 7.6.1) against $DH1^{768}$ still require heavy lifting for cracking the key exchange within the key lifetime, but it is assumed to be within reach for resourceful attackers. This is not for everyone, and nation-state actors would, therefore, likely choose a legal approach for domestic key seizure.

However, downgrading to a weaker DH group alone should already be considered a serious vulnerability. Otherwise, selecting different key lengths would be pointless.

Recovered (downgrade attack) and leaked static keys do not always have to be used to attack higher layers up to the SIP/IMS connection (snooping conversations or spoofing commands). (L1) decryption alone can be used as a type of IMSI Catcher [19] on VoWiFi by sniffing EAP-AKA identifiers.

9.4.1 IPsec Rekeying Problems

We see a number of downgrade attacks against the IKE phase 1 manifesting in the 3GPP VoWiFi ecosystem (RQ3). An attacker should not be able to force an exchange method and bit length upon the parties.

Attacks on (L1) and the rekeying system gain impact because they inherit a previous EAP-AKA authentication on (L2). And since (L3) SIP Encryption is optional in many cases (Experiment see Appendix A), this gives an attacker control over all three signal and user data panes.

9.4.2 Accepting Undocumented and Unoffered Algorithms or Key Lengths

The MediaTek vulnerability of accepting unproposed and non-complaint key exchange methods is an *insecure implementation* by over-fulfilling the specification, as described by Rupprecht et al. [41]. In this type of implementation error, the attack surface is unnecessarily enlarged (the client accepts a larger input language than required or even advertised) by including extra functionality outside of the specification. The weak key exchange method was likely inherited from a general-purpose IKE implementation or library. The developer removed it from the advertised methods but never checked the received selection. Ideally, unsupported methods should also be removed from the code.

9.4.3 Not-so-private Private Keys

As the manufacturer was identified as the source of the global static set of ten round-robin keys, they can not be considered *private* for a number of reasons:

a) All of the affected operators are in possession of the same keys and can decode each other's traffic.

b) The manufacturer (and any demo or test customer) are also in possession of those keys.

c) Security or private actors could seize the opportunity to get those keys from an institution under their jurisdiction or other control.

d) Used telco equipment finds its way to second-hand hardware marketplaces [43] and might leak those keys into the public.

10 Evasive Recommendations

10.1 Default to Strongest DH Group

Operator configurations that are preloaded to clients are updated only irregularly and thus often outdated. In practice, supported DH groups within those configs are often comparatively weaker than on the server side. To counter this, clients should treat the preloaded options as a lower bound, and always signal (and prefer) stronger DH groups in their proposal. In the worst case, this adds another roundtrip where the server indicates that it does not support that mode. To save that on subsequent connections, server capabilities could be temporarily cached.

Failure Mode A rejected VoWiFi handshake on security grounds does usually not lead to loss of service for the customer, as the phone falls back to cellular service.

10.2 Not-so-private Private Keys

Leakage or re-usage of private keys can happen for a number of reasons - but from the perspective of a phone that most of the time connects to a single operator's ePDG, only the intra-operator reusage is detectable, not the inter-operator reusage.

UE-local Freshness Tests In lieu of a cryptographically ensured freshness, the client can detect key intra-operator re-usage with a history mechanism. However, based on the observation time frame and the network volatility, this history might grow large. A constant size and complexity key history could employ a temporal ring of Bloom Filters [20].

Distributed methods Inter-operator key re-usage detection would require cooperation between a (vast) number of phones either with a common infrastructure (e.g. like DNS blocklists) or in peer-to-peer mode.

Failure Mode A security-rejected handshake is tolerable, as the user would not experience a loss of service due to fallback to cellular service.

10.3 Fallback to an Unannounced Mode

Do not roll your own crypto! is valuable advise. However, if a standard library is used, unsupported methods and ciphers should be removed not only from the negotiations but also from the code base. A missing test coverage for a particular piece of code could either hint at a missing test or a removable over-implementation.

10.4 Defined Upgrade Path in Standardization

As discussed in Section 9.2, standards of cryptographic applications should define an upgrade timeline for minimum supported security features, such as key length.

11 Conclusion

The VoWifi ecosystem relies on IKE and IPsec to set up secure tunnels into the operator's EPC. However, multiple factors lead to a delayed adoption of up-to-date key exchange mechanisms. Deprecated DH groups (by 2015 standard) and other dated cryptographic primitives are the norm on the client and the operator sides in 2024 – and computing power only got cheaper in that time frame.

Furthermore, we encountered client implementation issues with a major smartphone SoC vendor, facilitating downgrade attacks to weak, non-compliant key exchange methods.

The biggest surprise was the operator side, as at least 13 operators serving 140 million customers apparently used the same global set of static private keys. In both cases, we helped to remove those vulnerabilities through responsible disclosure programs and tracked their progress.

Acknowledgments

This paper has been in part funded by the *UniVie Doctoral School Computer Science* and *Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle* with financial support by the Austrian Federal Ministry of Labor and Economy, the National Foundation for Research, Technology and Development and the *Christian Doppler Research Association*. Further support by the *Usable Security Group* led by Katharina Krombholz at the *CISPA Helmholtz Center for Information Security*. Further funding originates from the *NGI0 Entrust Fund*, established by *NLnet* with financial support from the European Commission's Next Generation Internet, under grant agreement No 101069594 and Project FO999887504 *DynAISEC* funded by the Program *ICT of the Future*. The competence center *SBA Research* (SBA-K1) is funded within the framework of *COMET – Competence Centers for Excellent Technologies* by BMVIT, BMDW, and the federal state of Vienna, managed by the FFG.

References

[1] Austria Drei: Number of Mobile Subscribers. https://www.drei.at/de/ueber-uns/unternehmen/, Accessed: 2024-05-21.

[2] Brazil: Number of Mobile Subscribers per Operator. https://informacoes.anatel.gov.br/paineis/acessos/telefonia-movel, Accessed: 2024-05-21.

[3] Hungary Yettel: Number of Mobile Subscribers. https://www.ppftelecom.eu/our-companies/yettel-hungary, Accessed: 2024-05-21.

[4] Indonesia Smartfren: Number of Mobile Subscribers. https://www.prnewswire.com/news-releases/smartfren-telecom-and-aviat-establish-strategic-collaboration-in-indonesia-302028401.html, Accessed: 2024-05-21.

[5] Malaysia DiGi: Number of Mobile Subscribers. https://celcomdigi.listedcompany.com/misc/PressRelease/PressRelease__4Q23.pdf, Accessed: 2024-05-21.

[6] Malaysia Telekom: Number of Mobile Subscribers. https://www.mobileworldlive.com/operators/telekom-malaysia-profit-climbs-despite-revenue-dip/, Accessed: 2024-05-21.

[7] Malaysia U Mobile: Number of Mobile Subscribers. https://www.u.com.my/en/about-us/our-company/awards-and-milestones, Accessed: 2024-05-21.

[8] Malaysia unifi: Number of Mobile Subscribers. https://www.malaysianwireless.com/2023/05/telekom-malaysia-unifi-consumer-spending-1q23/, Accessed: 2024-05-21.

[9] Nepal Telecom: Number of Mobile Subscribers. https://www.nta.gov.np/misreport, Accessed: 2024-05-21.

[10] Pakistan Telenor: Number of Mobile Subscribers. https://www.telenor.com/about/our-companies/asia/telenor-pakistan/, Accessed: 2024-05-21.

[11] Russia Beeline: Number of Mobile Subscribers. https://www.statista.com/statistics/1361212/beeline-subscribers-russia/, Accessed: 2024-05-21.

[12] Slovakia 4ka: Number of Mobile Subscribers. https://www.telecompaper.com/news/4ka-grows-to-over-623000-subscribers-in-september--1481574, Accessed: 2024-05-21.

[13] Multiple Internet to Baseband Remote Code Execution Vulnerabilities in Exynos Modems. WebPage, 2023. https://googleprojectzero.blogspot.com/2023/03/multiple-internet-to-baseband-remote-rce.html, Accessed: 2023-08-21.

[14] Apple Grabs the Top Spot in the Smartphone Market in 2023, 2024. https://www.idc.com/getdoc.jsp?containerId=prUS51776424, accessed: 2024-02-05.

[15] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *22nd ACM Conference on Computer and Communications Security*, 2015.

[16] Karthikeyan Bhargavan, Christina Brzuska, Cédric Fournet, Matthew Green, Markulf Kohlweiss, and Santiago Zanella-Béguelin. Downgrade Resilience in Key-Exchange Protocols. In *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

[17] Gonzalo Camarillo, Vesa Torvinen, Jari Arkko, Aki Niemi, and Tao Haukka. Security Mechanism Agreement for the Session Initiation Protocol (SIP). RFC 3329, January 2003.

[18] Merlin Chlosta, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lte security disabled: Misconfiguration in commercial networks. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019.

[19] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. Imsi-catch me if you can: Imsi-catcher-catchers. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC 2014)*. ACM, 12 2014.

[20] Adrian Dabrowski and Edgar R Weippl. Mobile phone's wifi presence for continuous implicit secondary deauthentication. In *11th International Conference on Passwords*, volume 12, 2016.

[21] Pasi Eronen, Yoav Nir, Paul E. Hoffman, and Charlie Kaufman. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996, September 2010.

[22] ETSI. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 3G security; 3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses. ETSI TS 133 402; collection of multiple releases: https://www.etsi.org/deliver/etsi_ts/133400_133499/133402/.

[23] ETSI. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 3G security; Network Domain Security (NDS); IP network layer security. ETSI TS 133 210; collection of multiple releases: https://www.etsi.org/deliver/etsi_ts/133200_133299/133210/.

[24] ETSI. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Numbering, addressing and identification, 2023. ETSI TS 123 003 V17.10.0, https://www.etsi.org/deliver/etsi_ts/123000_123099/123003/17.10.00_60/ts_123003v171000p.pdf.

[25] Dennis Felsch, Martin Grothe, Jörg Schwenk, Adam Czubak, and Marcin Szymanek. The dangers of key reuse: Practical attacks on IPsec IKE. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 567–583, 2018.

[26] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. POSTER: Never Gonna Give You Up: Exploring Deprecated NULL Ciphers in Commercial VoWiFi Deployments. In *17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2024.

[27] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2024)*, 2024.

[28] Gabriel K. Gegenhuber, Wilfried Mayer, and Edgar Weippl. Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs. In *IEEE Global Communications Conference (GLOBECOM)*, 2022.

[29] Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, and Adrian Dabrowski. MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research. In *Usenix Security Symposium 2023*, 2023.

[30] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In *21st USENIX Security Symposium (USENIX Security 12)*, 2012.

[31] Man Hong Hue, Joyanta Debnath, Kin Man Leung, Li Li, Mohsen Minaei, M. Hammad Mazhar, Kailiang Xian, Endadul Hoque, Omar Chowdhury, and Sze Yiu Chau. All your credentials are belong to us: On insecure wpa2-enterprise configurations. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, New York, NY, USA, 2021. Association for Computing Machinery.

[32] Internet Assigned Numbers Authority. Internet key exchange version 2 (ikev2) parameters. https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml#ikev2-parameters-8.

[33] Charlie Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306, December 2005.

[34] Charlie Kaufman, Paul E. Hoffman, Yoav Nir, Pasi Eronen, and Tero Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296, October 2014.

[35] Mohammad Taha Khan, Joe DeBlasio, Geoffrey M. Voelker, Alex C. Snoeren, Chris Kanich, and Narseo Vallina-Rodriguez. An empirical analysis of the commercial vpn ecosystem. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, New York, NY, USA, 2018. Association for Computing Machinery.

[36] Aniss Maghsoudlou, Lukas Vermeulen, Ingmar Poese, and Oliver Gasser. Characterizing the vpn ecosystem in the wild. In Anna Brunstrom, Marcel Flores, and Marco Fiore, editors, *Passive and Active Measurement*, 2023.

[37] Yomna Nasser. Cellular radio "null ciphers" and android. Real World Crypto Symposium 2023, slides available: https://iacr.org/submit/files/slides/2023/rwc/rwc2023/3/slides.pdf.

[38] Yoav Nir, Tero Kivinen, Paul Wouters, and Daniel Migault. Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2). RFC 8247, September 2017.

[39] Reethika Ramesh, Leonid Evdokimov, Diwen Xue, and Roya Ensafi. Vpnalyzer: Systematic investigation of the vpn ecosystem. In *Network and Distributed Systems Security (NDSS) Symposium 2022*, 2022.

[40] Hendrik Rood and Rudolf van der Berg. Re-volte: Should we stop the shutdown of 2g/3g to save lives?, 2022. (Presentation), May contain Hackers MCH2022, Netherlands, https://program.mch2022.org/mch2022/talk/7TVHSD/, slides available: https://www.slideshare.net/3G4GLtd/should-we-stop-the-shutdown-of-2g3g-to-save-lives.

[41] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. On security research towards future mobile network generations. *IEEE Communications Surveys Tutorials*, 20(3):2518–2542, thirdquarter 2018.

[42] Merve Sahin and Aurélien Francillon. Over-The-Top Bypass: Study of a Recent Telephony Fraud. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[43] Hendrik Schmidt and Brian Butterly. Attacking basestations, 2015. DEF CON 24 Presentation, slides online: https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEF%20CON%2024%20-%20Hendrik-Schmidt-Brian-Butter-Attacking-BaseStations-UPDATED.pdf.

[44] Joe-Kai Tsay and Stig F. Mjølsnes. A vulnerability in the UMTS and LTE authentication and key agreement protocols. In Igor Kotenko and Victor Skormin, editors, *Computer Network Security*, 2012.

[45] Guan-Hua Tu, Chi-Yu Li, Chunyi Peng, Yuanjie Li, and Songwu Lu. New security threats caused by ims-based sms service in 4g lte networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[46] Luke Valenta, Nick Sullivan, Antonio Sanso, and Nadia Heninger. In search of curveswap: Measuring elliptic curve implementations in the wild. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[47] Luke Valenta, Nick Sullivan, Antonio Sanso, and Nadia Heninger. In search of curveswap: Measuring elliptic curve implementations in the wild. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[48] Paul Wouters. Deprecation of the Internet Key Exchange Version 1 (IKEv1) Protocol and Obsoleted Algorithms. RFC 9395, April 2023.

[49] Ka Lok Wu, Man Hong Hue, Ngai Man Poon, Kin Man Leung, Wai Yin Po, Kin Ting Wong, Sze Ho Hui, and Sze Yiu Chau. Back to school: On the (In)Security of academic VPNs. In *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

[50] Yaru Yang, Yiming Zhang, Tao Wan, Chuhan Wang, Haixin Duan, Jianjun Chen, and Yishen Li. Uncovering security vulnerabilities in real-world implementation and deployment of 5g messaging services. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2024.

Appendix

A Experiment: SIP Encryption Optionality

Not all operators enforce encryption and integrity on L3 in practice, which leaves room for even more severe attacks G1-3. This is partly visible in our static configuration file analysis but also in-vivo verifiable:

For example, when we removed client-side encryption and authentication preferences on our testing device (using the

Table 4: Deprecated IKEv2 SA proposal parameters [23, 38, 48]

Category	ID	Name
Encryption Algorithms	1	DES IV64
	2	DES
	4	RC5
	5	IDEA
	6	CAST
	7	BLOWFISH
	8	3IDEA
	9	DES IV32
Pseudo-Random-Functions	1	HMAC MD5
	3	HMAC Tiger
Integrity Algorithms	1	HMAC MD5_96
	3	DES MAC
	4	KPDK MD5
	6	HMAC MD5_128
	7	HMAC SHA1_160
Key Exchange Methods	1	768 Bit MODP
	2	1024 Bit MODP
	5	1536 Bit MODP
	22	1024 Bit MODP 160 Prime

stock MTK EngineerMode app), we were still able to connect to the home operator successfully.

In such cases, an attacker that cracked the outer IKEv2 key exchange and is thus able to take over the first two layers can subsequently also hijack the third layer after the SIP authenticated between UE and P-CSCF is finished, effectively dominating all three communication layers and reaching all available goals (G1-3).

B Experiment: No Integrity Protection in Regular Rekeying

As described in Section 7.3, downgrade attacks via the `INVALID_KE` message are not integrity protected and work as a stepping stone to taking over the full (L1-L3) stack.

In this experiment, we verify that the same is true for the regular rekeying of (L1).

In this instance only, to simulate the capability of breaking the key exchange, we used a rooted phone. We inject Frida[18] into the process responsible for the IKEv2-related communication and extract the used encryption and authentication keys by intercepting the corresponding library functions.

Thus, we were able to manipulate the rekeying interval and observe it in vivo.

The results confirm that regular rekeying lacks integrity protection similar to `INVALID_KE`-triggered rekeying.

C Static Key Re-usage: Mapping MCC-MNC to Operators

As mentioned in Section 4.4.1, an MCC-MNC tuple does not necessarily constitute an operator. Old MCC-MNCs are often kept alive for historical reasons.

In our set of 15 ePDGs using static keys (Table 5), three pairs had identical IP addresses. *Hutchison Drei* actively maintains 232-05 and 232-10 after merging *Orange* and *One*. The case is very similar for *Smartfren* and their 510-09 and 510-28 designations. In contrast, Malaysia's *U Mobile* and *DiGi* cooperate by maintaining a common 5G infrastructure but are otherwise (mostly) independent operators. Thus, the latter ones are counted as two operators.

Pakistan's Telenor newly showed up in our scans on April 2nd, 2024, over a month after we started the responsible disclosure, and several operators had already rolled out the patch.

D Responsible Disclosure and Remediation

D.1 MediaTek Unannounced DH Group and Downgrade

MediaTek confirmed our findings and issued CVE-2024-20069[19] (severity: high) for the described downgrade attack. The affected basebands[20] with the NR15 modem are from the Dimensity product line.

They released patches to all affected customers. All Android devices with a Security Patch Level (SPL) of 2024-06-05[21] or later are protected from the downgrade attack.

D.2 Globally Static Set of DH Exchange Keys

After the experience with the few and slow responses from the operators themselves [29], this time we reached out to the GSMA's Coordinated Vulnerability Disclosure (CVD) program to timely contact the affected operators and the manufacturer on 2024-02-13. We further reached out to Apple and Google to consider countermeasures for their mobile operating systems.

The GSMA has issued CVD-2024-0089 to track our findings and further helped to communicate them with the affected manufacturers and operators.

ZTE confirmed our findings and issued CVE-2024-22064[22] (severity: high). The software component responsible is ZXUN-ePDG from their CCN (Computing and Core Network) product line. According to ZTE, the bug has been present in all versions before V5.20.20. The issue was caused by incorrectly shipping integration test keys in the production release, they explained. Besides a fixed version, ZTE also offers a volatile runtime-only fix for MNOs that can not currently be updated; it must be reapplied after each restart.

[18] https://github.com/frida/frida/

[19] https://corp.mediatek.com/product-security-bulletin/June-2024#CVE_2024_20069

[20] MT6833, MT6853, MT6855, MT6873, MT6875, MT6875T, MT6877, MT6883, MT6885, MT6889, MT6891, MT6893, MT8675, MT8771, MT8791T, MT8797

[21] https://source.android.com/docs/security/bulletin/2024-06-01

[22] https://support.zte.com.cn/support/news/LoopholeInfoDetail.aspx?newsId=1035524

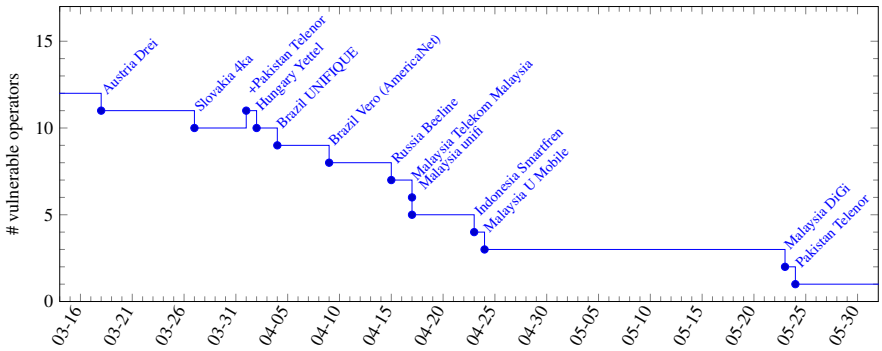


Figure 11: Globally Static Set of DH Keys: Remediation over Time

Table 5: Static IPSec keys: Vulnerable Operators.

MCC-MNC	Country	Operator	Subscribers(M)		Remediation[b]
232-05, 232-10	Austria	Drei	4.1	[1]	2024-03-18
231-03	Slovakia	4ka	0.6	[12]	2024-03-27
216-01	Hungary	Yettel	3.7	[3]	2024-04-02
724-29	Brazil	UNIFIQUE	< 0.5	[2]	2024-04-04
724-26	Brazil	Vero (AmericaNet)	< 0.5	[2]	2024-04-09
250-99	Russia	Beeline	44	[11]	2024-04-15
502-11	Malaysia	Telekom Malaysia	2	[6]	2024-04-17
502-153	Malaysia	unifi	0.8	[8]	2024-04-17
510-09, 510-28	Indonesia	Smartfren	36	[4]	2024-04-23
502-18	Malaysia	U Mobile	8.5	[7]	2024-04-24
502-16	Malaysia	DiGi	20.6	[5]	2024-05-23
410-06[a]	Pakistan	Telenor	(44)	[10]	2024-05-24
429-01	Nepal	Nepal Telecom	20	[9]	-
Total			> 140.3 Mio		

[a] Vulnerability introduced April 2nd 2024. [b] Cut-off date: May 31th 2024

D.2.1 Remediation Timeline

To track the progress of the update, we ran scans hourly. In mid-March 2023, one operator confirmed to us that they received a patch and were testing it. Shortly thereafter, most operators started rolling out the patch into production. Figure 11 and Table 5 note the last time for each operator, we recorded an ePDG server using one of those static keys.

Interestingly, Pakistan's Telenor first started showing up as vulnerable on April 2nd in the scans well after a fix was available. We were later informed, that this is a test site not ready for commercial use and will be upgraded before it is open to customers.

D.2.2 Key Hashes

We include the SHA256 hashes of the found globally-used non-random public keys (bytes in network order) to facilitate blacklisting of those keys. Because of space constraints, we only include DH1^{768} through DH15^{3072}.

DH1^{768}:
c91fbb17c38e95c3590c54838bab62808df808cce198c3ba24e830c8f3cc2fc7
564a3bad4f7504c4c1c515b8cad7687cfdf19af0cce3b527cfc56093fab266c8
7050ed52e9222665ad11f20cc51218c253a1a54695ab246bf0e408d9ac041176
498f3bbfa8f8f2b5b1b3ec7cd6790d960f0c760ecbaf5eca5d31752aa2fcc1fd
0caf2731a9dfd392821134adff2f0ffa8097e220dcd0955c3370571da0ee9586
2ade9b103ce1cb75a7894905b55c51ec5cf6bc78513cdd9373c3266f0d1c2ed2
4612bc13632fb814fac5ce1b24aba1a79ef8284ea737b241e5311423c0510782
43090ffd7d3285678d3b1d98e4bfbe5775911a5595258639287a641b48eb32a3
b2b8e7eb53256495519209eebd98a2f9d7974241c848c7ad37ec586676ae4116
a005b8aac47ce34c6293f1f37da868107f5e05db5aebb4618d899a94927af47e

DH2^{1024}:
e99889815a602ab1863e4d900650233bfc00aa64ce29fde18c36219f6aa361e7
a9f58f41c2e3b1d36f74d14e0a60e7e833e1faf438b71e42a0ee76fd208420d3
d24c415adb25bb2a8adbde6ee9da4f5c65c39b746a87e9ac71b613b664720c41
9c3fee28e4a984db043924ccd42a8121bf1fd696428a82fac624197df10a4d35
3289fe1551fc0fbf372f293ffa9867c52e30d2357d3ab2ae5d4b8c896f4b57b5
9dd792e808954c00fc1565a447324788a34913a7df977a836bd523edca1a89d5
e8bc246f549cab69d4e6789cfa611d24828d532f839597c02d7b193ae0dc7cab
423516e9e2e67f0018abf66e20f2ded682da51d12752fe010103698782575f6f
936aeb8c8d413d03682a0ab68ed7ae0f98c0be630575704eb3f395d70bea83ab
01870a3a8e23257f81ea50e84a7cac4d7be949c18681725576c66e32b811c6b9

DH5^{1536}:
b22cdb284a4db637fe76f1f7ab8a1f8c2530976805822686b008176e60e5cf29
398c99f3e84bef849fc62a6216c5b66f95445a92ce6b8d49b56f8c73c994e774
578a16523d85ac4ab566b7cefa2545a64a4a7175b4432eab9a2a5e4200a3e391
74e0aa079e655f8caeefa0dc35c8c81d5c81e4bc4b5d3e4f975a1f1ed198f68f
d083fb11ed987a56f28e5887980ead2a173396f0676e5b322aa58e81dd85d4c5
2ba8036874c1cc870d280d1a388a2746ac6ed62f26e427362c8aadc7e2ea13bd
337095ee22be46044dd0e4626b0b0a19728273001bda8fbf7e77afc514fc8e5f
4a1d84aab699e209fd96dd611f3c25128c314d37b43fa4d325057b7a1943f09a
9bd20704d7ab4c879514f2d69f2cbcd8787cbed3c44db0843fe6c540de143799
662cf2600a8432a09a96f7e2ca480df04712c5ef58d51c95c4341bb085a80491

DH14^{2048}:
9247ccc73f6fcc832fd43458f96c8517d45df5548d9841650aec23dae765d918
c75706b089bfc671a7678661786f6aaad53afa5f57d415d305e0d7e4ee996c0b
8f9dd2cd05e0b1884e9dde3dcc74b01d782014df68b86bc074782a7a2d78b467
5aaa0f715a5affea5b40e1e8dad902d8f90adb64b02a99f36b01340878f39a89
55895ae426a8487ccde1a38cc6631e5728a5eb8605269d4c907f2a93c09e1e03
e0b4747bff0898e52ef435e930842dea8f3a48ca7d0e3e37b32a0e390a81adaf
b156c9089f74e32a2e88e110b24010aa1824c5ec625d591081f8d46c49f4b7d2
08e81e262766a1baadde5ad543a83477be7153394d7dba6d61682eb3f8e24284
58bfb525033d093dfcce2c483c25495dad2591965253cfbddfc98579c6e8bdf4
59b6b9094b1b6dfd637d80cc59c8a44160e053c0bb1c66e1f58e1a871258a53a

DH15^{3072}:
13ca255b94a7284399177e828f1f39c4a66d618cd735455e5391b4445c603c8d
5a4f387f10bee59a6209244e43d0eaa67c1e6255c5b2376ffce51f7448d72870
9ce716182a2790cebe900630bdef64def59ed90e45e7d87029b60d145c20a22b
1f57f25e95eaeba86e5004b030584333783675e5db9126483b10b9a9262cd25b1
855aa6a8bb2b2327f52ed5d791f7ef211cc3177e50fa47c907f9a9004e91d002
b0b84567c90008babed048914325b15ff016d72b5aa46283447a6ea0b16a8fe5
8fdc2945a14dd9e7f6646107b9d324ab16a5d378e138c282c679f1de343fe447
516fb8ca4462ff067cd0611f391a289d1aaae6e73b2d96a5d7ad8444ce714a6f
a9fc5fabd3b4d94c2d11eb4ece812548a93ecb87a4ce82f883b55e6f683a5bc8

7 Individual User Monitoring via Silent Pings on Instant Messengers

Publication Info

Title	Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers
Authors	Gabriel K. Gegenhuber, Maximilian Günther, Markus Maier, Aljosha Judmayer, Florian Holzbauer, Philipp É. Frenzel, Johanna Ullrich
Publication Status	This paper is included in the Proceedings of the 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2025. To appear in the proceedings. *Distinguished with the Best Paper Award.* CORE2023 Ranking: A.
DOI	To be assigned by the publisher.
arXiv	`https://arxiv.org/abs/2411.11194`
Reference	[GGM+25]

Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers

Gabriel K. Gegenhuber[1,2], Maximilian Günther[1], Markus Maier[1],
Aljosha Judmayer[1], Florian Holzbauer[1], Philipp É. Frenzel[3], and Johanna Ullrich[1]

[1]University of Vienna, Faculty of Computer Science, [2]UniVie Doctoral School Computer Science, [3]SBA Research

***Abstract*— With over 3 billion users globally, mobile instant messaging apps have become indispensable for both personal and professional communication. Besides plain messaging, many services implement additional features such as delivery and read receipts informing a user when a message has successfully reached its target. This paper highlights that delivery receipts can pose significant privacy risks to users. We use specifically crafted messages that trigger silent delivery receipts allowing any user to be pinged without their knowledge or consent. By using this technique at high frequency, we demonstrate how an attacker could extract private information such as following a user across different companion devices, inferring their daily schedule, or deducing current activities. Moreover, we can infer the number of currently active user sessions (i.e., main and companion devices) and their operating system, as well as launch resource exhaustion attacks, such as draining a user's battery or data allowance, all without generating any notification on the target side. Due to the widespread adoption of vulnerable messengers (*WhatsApp* and *Signal*) and the fact that any user can be targeted simply by knowing their phone number, we argue for a design change to address this issue.**

I. Introduction

Instant messengers serve a vast global audience with WhatsApp alone reaching over 3 billion users [9], [28] and handling billions of messages daily. Besides being very common in general, instant messaging services are also used by high-profile government officials for sensitive conversations [1], [13], which adds an entirely different dimension to privacy issues within these services. In this paper, we present a novel privacy and availability attack vector on instant messaging systems, leveraging the (ab)use of *delivery receipts*.

There are two basic ways used by instant messengers to inform senders about message delivery, namely *delivery receipts* (consisting of *server ack & device ack)* and *read receipts*. The first acknowledges a message's receipt at the server or the destination device, the latter their view by the destination device's user. Read receipts have been misused to spy on conversation partners [7] and nowadays messenger applications allow to disable them in their privacy settings. Delivery receipts, however, cannot be deactivated due to design choices of the underlying protocol. Previous work has triggered delivery receipts through sending regular text messages in ongoing conversations and thereby showed that, based on the measured round-trip times (RTTs), country-level geolocation of a user's device is feasible [21]. Regular messages, however, trigger notifications for the target user, potentially alerting them to the ongoing attack, particularly when the probing messages are sent frequently.

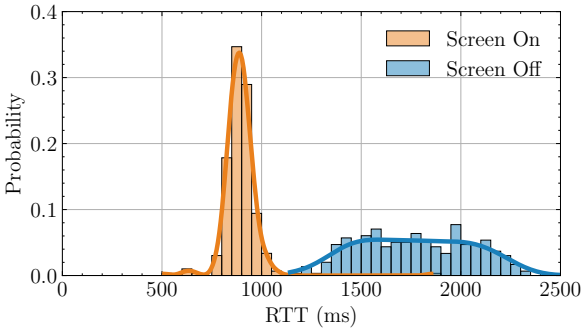


Fig. 1: Round-trip times (RTT) of delivery receipts, which are ≤ 1 second for *Screen On* states and > 1 second and above for *Screen Off* states measured on an iPhone using WhatsApp with a sampling rate of 1 Hz.

Using techniques described in this paper, an adversary can craft stealthy messages that enable probing a target at high frequency (up to sub-second granularity) while not causing any notification at the target side and also in the absence of an ongoing conversation. With such an increased sampling rate, we systematically show that delivery receipts can leak user information beyond the country level.

For example, we show that the on/off state of a mobile phone's screen manifests in the delivery receipts' timing, see Figure 1, and, among others, allows to track the victim's screen time.

Moreover, we demonstrate that a user's activity can be followed across multiple devices (i.e., smartphone and companion sessions), creating further (and more severe) monitoring and tracking possibilities. In addition to utilizing delivery receipt RTT as a timing side channel, we demonstrate that implemen-

tation inconsistencies across different target architectures also leak information about the operating systems and application clients in use.

Using the same techniques, also resource exhaustion attacks, such as draining the battery or data quota, can be performed. Similar to [21] we focus our analysis on three instant messenger platforms, i.e., WhatsApp and the more security-orientied alternatives Signal and Threema.

In summary, we make the following contributions:

Stealthy Delivery Receipts. While previous work required an ongoing conversation and consequently notified the victim about every probing message, we show that delivery receipts are also issued for other message types (e.g., reactions) and furthermore can be triggered in a silent way preventing a notification of the victim. This allows **constant** and **high-frequency probing** (i.e., sub-second interval) of target devices without the risk of getting noticed and blocked by the victim.

Multi-Device Amplification. We demonstrate that in multi-device setups – in which victims use web or desktop clients for messaging in addition to their mobile phones – each device can be independently probed enabling comprehensive and precise observation of user behavior across devices throughout the day.

Arbitrary Targets. For WhatsApp and Signal, we show that it is not even required to have any kind of association with the victim, e.g., being in their contact list or having an existing conversation. Thereby, anyone having these messenger applications installed on their mobile phone can be selected as a victim just by knowing their phone number and monitored using the techniques described in this paper. With billions of users, the number of potential victims is not only huge but also includes potential high profile targets such as government officials [1], [6], [13], [29].

Activity Leakage. We show that delivery receipt timings are influenced by the phone's activity and sleep states. This enables us to determine whether the remote device is actively used or in standby by distinguishing between screen on/off states and in certain scenarios even if the messaging app is currently in foreground. Furthermore, we demonstrate that complex routines, schedules, and activities can influence delivery receipt timings across a user's different devices, as shown by an open-world measurement case conducted under real-world conditions.

Resource Exhaustion Attacks. We show that our findings can not only be used for the disclosure of private information, but also in an offensive way exhausting a victim's resources like battery or data quota – similarly, this type of attack does not alert the victim through notifications.

Countermeasures. We discuss how these attacks can be mitigated, including both client and service-side mitigation strategies.

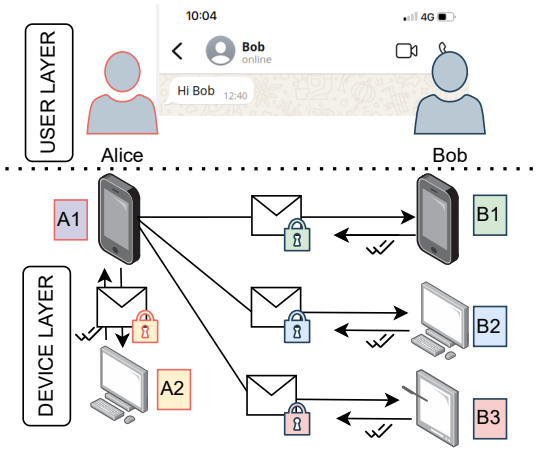


Fig. 2: Simplified depiction of client-fanout for Multi-Device-Support: Alice's message is sent to all of Bob's devices as well as her desktop device. Each message copy is individually encrypted. The recipient devices inform Alice's device of the successful decryption via delivery receipts.

II. Background

In prevalent end-to-end encrypted (E2EE) messaging protocols used in WhatsApp and Signal, the role of the server is essentially reduced to forwarding encrypted messages to their recipients, and large parts of the protocol logic are shifted to the client side. This also includes the handling of re-encryptions and re-submission in case of decryption failures at the receiver, e.g., if the session state and associated keys on a recipient device have been deleted, or rolled back. Thus, in addition to serving as a convenience feature for users, clients also depend on information about the successful delivery and decryption of messages from a technical standpoint. These acknowledgments are commonly referred to as *delivery receipts*. Delivery receipts indicate the successful decryption of a message and thus allow the sender to mark the transmitted message and the associated ephemeral keying material for deletion. This is necessary to uphold the security property of forward secrecy as promised by these messaging services. As we show in this paper, this design decision with the according shift of responsibilities to clients in combination with the desired responsiveness and convenience of low latency interactions can have a significant impact on the privacy as well as the security of users.

Message Delivery. The *server ack* delivery receipt, usually represented by a single checkmark (✓) on client devices, indicates that the message was queued for further transmission at the message server. Due to E2EE, the message server does not see the message content and thus cannot perform message validation checks in this step. The *device ack* delivery receipt, often represented by two checkmarks (✓✓) on client devices, highlights that the recipient has received and successfully decrypted the message. Therefore, the delay between sending

the message and receiving the delivery receipts indicates the time it took for sending the message first to the message server, then from the message server to the target device (if it is currently online) and back again. For details regarding the message server infrastructure, the resulting RTTs and the inferred limits for geolocation see Appendix A1. The *read receipt* (✓✓) is a message type returned by the target device in case the target user accesses the message. However, WhatsApp, Signal and Threema allow to disable this message type in the device settings.

Multi Device Setups. There exist *leader-* and *client-fanout* based approaches to support multi device (i.e., companion) setups. In contrast to leader-based approaches, where one device acts as a primary device accountable for redistributing messages to other devices, current client-fanout implementations trigger delivery receipts by *all* user devices connected to the account. This client-fanout is now implemented in WhatsApp and Signal. Threema is currently in the process to enable multi-device setups [25]. In client-fanout setups all devices of the user maintain their own key pair, cf. Figure 2. For messaging, the sender creates an individual E2EE channel with *each* device of the receiver as if messaging multiple recipients. For remaining consistency among all devices of the sender, the sending device also forwards the message (and other information) to other devices of the same user [17]. This approach avoids the single point of failure inherent in the leader-based method. However, assigning multiple keys to an account reveals the number of devices under the control of a user, as other accounts necessarily have to retrieve these keys from a central inventory. Since each device has its own unique key, the recipient can also infer from which of the sender's devices the message originated [2].

Mitigation Status. Although previous work [21] has shown that delivery receipts introduce a timing-based side channel that coarsely leaks a user's location or the used access technology (i.e., cellular, Wi-Fi), there is still no way of turning them off at most popular messengers. Moreover, other proposed mitigation techniques like delaying the delivery receipt by a random delay of a few seconds were not adopted.

III. Threat Model & Measurement Setup

We assume an adversary that aims to impinge a victim's privacy by exploiting instant messenger services' delivery receipts. More specifically, we differentiate between i) three distinct attacker goals, ii) two attacker types differing in their relation to the victim and iii) limit our analysis to three popular messaging services – WhatsApp and its more security-oriented alternatives Signal and Threema.

A. Attack Goals

From an adversary's perspective, the three goals are:

G1 Fingerprinting the number and types of a victim's devices by observing received messages and delivery receipts, thereby inferring each device's online status and enabling tracking across multiple devices, locations, and behavioral routines.

G2 Monitoring a user's behavior (e.g., screen on/off, or messenger app currently in foreground) by covertly probing the device for longer periods and analyzing relative RTT differences of delivery receipts.

G3 Launching offensive resource exhaustion attacks that increase a user's traffic, draining their battery or data allowance, or performing denial-of-service attacks to degrade the usability of bandwidth-intensive applications such as video calls on secure messaging platforms.

The adversary aims to carry out the attack as stealthily as possible, ensuring that the victim remains unaware not only of the source and cause, but also of the fact that an attack is occurring at all.

Application	Installations	E2EE Messaging	Multi Device E2EE	Delivery Receipts	Open Source
WhatsApp	9.64B	●	●	●	○
Facebook Messenger	5.89B	●	●	●	○
Instagram	5.48B	●	●	○	○
TikTok	3.94B	○	○	○	○
Snapchat[a]	1.94B	○	○	●	○
Telegram[b]	1.64B	◐	○	○	●
Viber	1.18B	●	●	●	○
Line	1B	●	○	○	○
Signal	136M	●	●	●	●
Threema[c]	5M	●	◐	●	●

[a] E2EE for snaps, not for private messages.
[b] Non-default E2EE.
[c] Multi-device support for iOS only.

TABLE I: Overview over popular (>1B installations) and security-oriented instant messaging services (Sources: androidrank.org, manufacturer information).

B. Attacker Types

Depending on the relationship between the adversary and the victim, we differentiate between two attacker types, namely *creepy companion* and *spooky stranger*.

- ***Creepy companion***: The attacker and the victim have an active chat session containing one or more messages in the messaging app. Such adversaries, e. g., a jealous (ex-)partner or a nosy employer, typically also have real-life relationship with the victim.
- ***Spooky stranger***: There is no prior relationship – like a previous preexisting contact or conversation – between the attacker and the victim in the instant messaging app. The adversary only knows the latter's phone number. This way, any customer of a messaging service could be attacked just by knowing their phone number. This could be used to spy on public figures, e. g., celebrities or politicians, or to gather intelligence about a company's

App	Used Open Source Projects[2]
WhatsApp	whatsmeow$^{\Psi}$(web), Cobalt$^{\Psi}$(mobile)
Signal	signal-cli$^{\Psi}$(uses libsignal$^{\clubsuit}$)
Threema	threema-android$^{\clubsuit}$

$^{\clubsuit}$ Offical project. $^{\Psi}$ Community project (reverse-engineered).

TABLE II: Both official- and community-driven open source projects were leveraged to get API level application access.

CEO for industrial espionage. *Spooky strangers* can simply use a so-called burner phone with a prepaid SIM card to entirely hide their identity against the messaging service.

C. Messenger Selection

Table I shows the most popular instant messenger applications including also the social networks Instagram, TikTok, and Snapchat as they also support direct messaging. At the table's bottom, we listed the less prevalent but more security-oriented messengers Signal and Threema. For our analysis, we require messengers to support multi-device E2EE and delivery receipts limiting the potential target applications to five, namely WhatsApp, Facebook Messenger, Viber, Signal, and Threema. Beyond, we need API-level access. While this is clearly available for open-source messengers, the communication protocol and API endpoints need to be reverse engineered for proprietary messengers. In our research, we found web gateways enabling messaging as a non regular user, e.g., the Viber REST API[1], and projects emulating a webbrowser providing automation capabilities but no in-depth access to message delivery states. Finally, we only considered projects that implement the full feature set of a regular client, support E2EE key management and expose low-level API access, see Table II for an overview. This left us with three messengers for further analysis, *WhatsApp*, *Signal* and *Threema*. This choice of messengers is consistent with [21] and allows comparison with previous work.

D. Measurement Setup

We conducted our measurements in two steps: First, we looked at message types triggering delivery receipts (on the iOS and Android apps) without looking at specific device models and evaluate them based on our threat model, see Section IV for details. Based on this assessment, we craft attacks on the individual messengers and test the privacy leaks for specific manufacturers and models in a second step, see Section V details. The full list of our testing devices including their chipsets and software versions is shown in Table VIII in the Appendix.

[1] `developers.viber.com/docs/api/rest-bot-api`
[2] whatsmeow: `github.com/tulir/whatsmeow`
Cobalt: `github.com/Auties00/Cobalt`
signal-cli: `github.com/AsamK/signal-cli`
libsignal: `github.com/signalapp/libsignal-service-java`
threema-android: `github.com/threema-ch/threema-android`

Action	Delivery Receipt			Push Notification		
Message	●	●	●	●	●	●
Reaction	●	●	○	◐	◐	○
Edit	●	●	○		○	○
Delete	●	●	○	○	○	○

 Edits cause (silent) notifications for iOS users only (no notifications are shown on Android).

TABLE III: Different actions notify the sender via delivery receipt and the receiver via push notification. On WhatsApp and Signal, reactions only cause push notifications for messages originated by the receiver but not those by other users (hence marked with ◐). For edits and deletions, WhatsApp and Signal employ restrictions (i.e., time window, recurrence).

IV. Side Channel Vectors

To explore the probing capabilities and corresponding limitations for exploitation, we tested for side channels based on delivery receipts. More specifically, we analyzed i) which actions cause delivery receipts, ii) whether delivery receipts are also issued for messages coming from a *spooky stranger*, iii) how delivery receipts are issued within multi-device setups.

A. Delivery Receipt Sources

We systematically tested which actions trigger delivery receipts by using our custom clients to send messages to both Android and iOS phones. Moreover, we examined which actions notify the user (e.g., trigger a push notification or mark a conversation as unread) and which actions remain covert. Due to their stealth, the latter bear the potential of continuous monitoring of target users without them being notified by the messaging app.

More specifically, we systematically test and explore delivery receipts caused by the following actions:

- **Send message**: sending a normal (text) message to the target.
- **Edit message**: changing the content of a previously sent message.
- **React to message**: sending a message reaction (e.g., a 👍 or ❤ emoji) to an existing message.
- **Delete message**: revoking a previously sent message for all chat participants (*"delete for everyone"*).

Table III shows the results. While WhatsApp and Signal also send delivery receipts for reactions, edits, and message deletions, Threema restricts the delivery receipts to regular messages. Editing or deleting a message usually does not trigger a notification on the target's phone and could thus be used for tracking purposes. However, both WhatsApp and Signal impose restrictions on these actions. WhatsApp permits message deletion for up to two days and allows unlimited edits within 15 minutes. In Signal, the time frame for deleting and editing messages is 24 hours with an upper limit of 10 edits per message.

B. Stealthy Probing (Creepy Companions)

Threema only allows reacting to someone else's message but users on WhatsApp and Signal can also send reactions to their own messages. A user is only notified when somebody reacts to a message originally sent by them. Self-reactions do not provoke a notification for other chat participants but nevertheless trigger a delivery receipt. Therefore, self-reactions provide an inconspicuous way of probing a target to receive delivery receipts. To make things worse, there are no time or quantity restrictions on message reactions and users can change or remove their reaction at a later point in time. Thus, an attacker could simply react to an old message they sent themselves to stay under the radar. Finally, removing a reaction (i.e., sending an empty string as a message reaction) is entirely invisible to the targeted user providing an ideal vector for consistent monitoring.

The just described side channels do not necessarily require full API access. An attacker could use an official client, or manipulate the client's state (e.g., by using developer tools via the web app) to trigger inconspicuous reactions and observe the (encrypted) traffic to derive delivery report timings[3]. However, having API-level access facilitates the probing and shows that invalid messages are generously confirmed via delivery receipts. For example, a message deletion packet can be sent multiple times to harvest continuous delivery receipts (with only the first message deletion actually being effective). Although message deletion or edits that were sent after the official time window were not considered (i.e., executed) by the receiving client[4] these messages also generated delivery receipts, providing another stealthy side channel that could be used for tracking purposes.

Summing up, we show that an attacker (more specifically a *creepy companion*) can use the *remove reaction* action, or reactions to their own messages to stealthily monitor any target that has an existing conversation with them. The only requirement is a conversation with at least one message for the target that is then used by the attacker for reactions.

C. Stealthy Probing (Spooky Strangers)

Our tests showed that there is little validation done by the receiving client and that message reactions referring to non-existing messages also trigger delivery receipts. This removes the prerequisite of having an existing conversation containing a message that a reaction refers to. Therefore, this could also be exploited by *spooky strangers*.

To explore this scenario we purchased a new prepaid SIM, plugged it into a burner phone, and again used our custom clients to probe various target phones that do not have any previous relation (e.g., contact, conversation, group chat) with the attacker's phone number.

[3]This technique is used in [21]. In contrast to this work, the authors exploit delivery receipts of regular messages triggering notifications at the victim device. This implies that their probing is not stealthy and cannot be applied in a continous and high-frequent way as we do.

[4]Officially announced times differ from the actually enforced ones, e.g., deleting on WhatsApp is possible for 60 hours instead of 48 hours and editing for 20 minutes instead of 15 minutes. For Signal, the observed time window is 48 hours instead of 24 hours.

Messenger used for covert probing	Spooky stranger	Creepy companion	Each Device
WhatsApp	yes	yes	yes
Signal	yes	yes	yes
Threema	no	no	no

TABLE IV: Ability to covertly probe a target, i.e, without triggering a notification, using delivery receipts. The last column specifies if this is possible for each of a user's devices individually.

Our results showed that both WhatsApp and Signal allow arbitrary targets to be stealthily monitored by a *spooky stranger* via reactions referencing non-existing messages. Due to missing self-reactions, we did not identify a covert way of probing arbitrary targets on Threema as a *spooky stranger*.

D. Multi-Device Probing

Besides observing which actions can be used to probe a target user via delivery receipts, we also analyzed how delivery receipts are handled when the target uses multiple devices. For both, WhatsApp and Signal, all devices (Android, iOS, Mac, Windows, Linux, and Web) issue independent (i.e., duplicated) delivery receipts for all tested message types. If the device is online, delivery receipts are issued right away; if not, they are sent as soon as the connectivity of the device is regained. This amplification of delivery receipts further increases an attacker's tracking and fingerprinting possibilities (as shown in Section V-B) as they are capable of inferring when a device comes online from the receipt of pending delivery receipts. On Threema, we checked for multi-device receipts by sending normal text messages. Threema appears to synchronize issued delivery receipts among all devices of a user causing only a single delivery receipt per message.

E. Summary of Probing Capabilities

We summarize the identified side channels that can be used for covert probing of user devices in Table IV. Threema only responds with a single delivery receipt even in settings with multiple devices per user. Moreover, Threema does not allow a *spooky stranger* or *creepy companion* to covertly probe a user's device without triggering notifications for the victim. The only way that remains to trigger delivery receipts by a previously unknown user is by sending a normal text message and starting a new conversation, but this is obviously not stealthy. Summarizing, Threema handles delivery receipts in a restrictive way impeding stealth probing. Consequently, we focus on the the messaging services WhatsApp and Signal in the remainder of this paper.

V. Attacks & Exploitation

In this section, we show the manifold potential for abuse and exploitation of the discovered delivery receipt-based side channels for the two vulnerable applications WhatsApp and

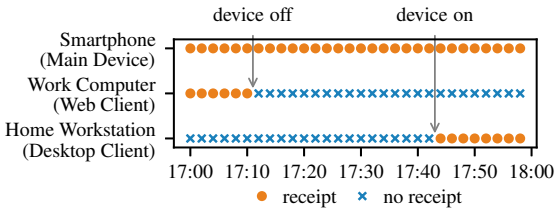


Fig. 3: A device's online status can be consistently and stealthily monitored with second-based granularity, possibly leaking the user's location and daily routines.

Signal. More specifically, we use stealthy message reactions to extract privacy-sensitive information from and for offensive resource exhaustion attacks against the victim. Most of the presented exploits require the attacker to continuously send stealthy message reactions towards the victim.

A. Tracking Users Across Devices

Concurrent work [2] shows that WhatsApp's key directory leaks the device setup information of users and that the used sender device can be extracted from received messages. We validated these findings and confirmed that this is also the case for Signal. For WhatsApp and Signal, the main device has the lowest device index (0 and 1 respectively) enabling differentiation between main- and companion devices. Monitoring a user's device directory and consequently observing a user's number of devices can be executed by *spooky strangers*.

B. Monitoring Device Online Status

Besides monitoring a target's key directory on the server, we are able to actively send packets that remain hidden to the victim. As all devices answer individually with a delivery receipt, continuous probing enables independent monitoring of each device's online status. In this use case, the attacker does not evaluate the RTT of the delivery receipts but simply their time of receipt to trace the online state of a device.

Main devices, i.e., mobile phones, are expected to be online most of the time, either via Wi-Fi or a cellular connection. If the main device ceases to respond, an attacker might deduce a brief connection disruption or that the phone has been switched to airplane mode. By monitoring this status over a longer period of time, an attacker might then be able to extrapolate the victim's behavioral patterns. For example, absent delivery receipts could indicate the victim being on a flight, or at a location with no coverage, e.g., in the metro, elevator, or basement, or simply using the phone's airplane mode to mute all messages during the night revealing their sleep schedule.

Companion devices (desktop or web clients) return delivery receipts upon missed messages as soon as they come online and the adversary will automatically be notified. This behavior might be abused to track the victim across devices and potentially expose their location, e.g., in case the companion device is a desktop computer in the office or at home (cf. Figure 3).

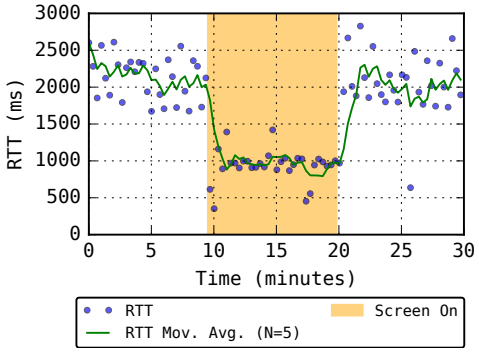


Fig. 4: WhatsApp Screen On/Off: Measured with low frequency (1 ping per 20 s), RTTs enable to differentiate between inactive and active screen states.

While web sessions need to be initiated by opening the corresponding website in the browser, desktop clients are often automatically started as a system service allowing an attacker to precisely monitor the online status of a companion device. As presented in the previous section, WhatsApp and Signal allow stealthy and independent pinging of all existing devices allowing both *creepy companions* and *spooky strangers* to exploit this attack.

C. Fingerprinting User Behavior

Besides using delivery receipts to classify a device's (binary) online status, relative differences in the observed RTTs can be used to derive the activity of the target device. We found that the operating system, the smartphone's model, its underlying chipset, and the current environment (e.g., screen- and target application status or Wi-Fi vs. LTE) heavily influence the occurring RTTs for a device.

Ping Frequency. We use different probing frequencies to measure characteristic RTTs on testing devices. On WhatsApp, we did not experience any rate limiting or server-side queuing and could also send high-frequency ping messages, e.g., one reaction every 50 ms, without any restrictions. On Signal, short bursts were also permitted but sending multiple messages per second continuously over an extended period caused them to queue. Thus, we refrained from sending more than one ping per second to circumvent rate limiting.

1) Case Study: iPhone: To investigate the feasibility of extracting detailed statistics about the ongoing user activity (e.g., screen time and app activity state), we systematically measured Android and iOS phones in different environments. Due to its dominant market share (currently, more than 50% in the US [24]), we select the iPhone to showcase our results. In our tests, we compared two different iPhone models (iPhone 13 Pro, iPhone 11), both showing the same characteristic patterns for specific activities (screen on/off, application in foreground). For our systematic measurements, we fixed the ping rate to one packet every 2, or 20 seconds.

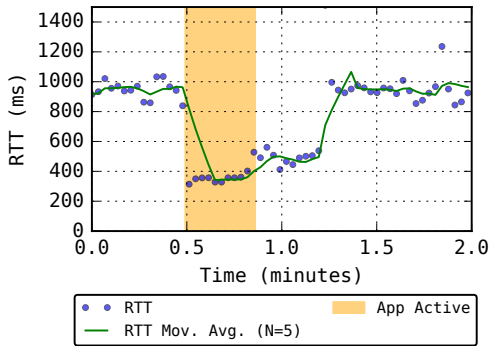


Fig. 5: WhatsApp Use: RTTs are 350 ms if the application is active (foreground). If minimized, RTTs become 500 ms for 30 s before eventually returning to 1 s *screen on* as typical for long-term app standby.

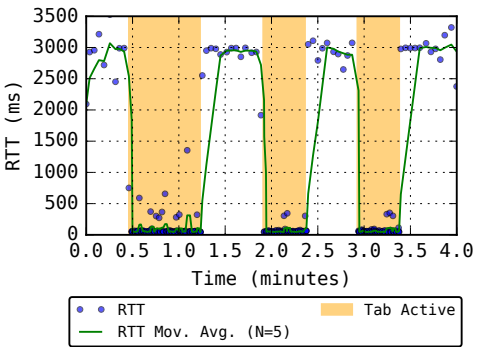


Fig. 6: WhatsApp Web on Firefox (Windows): RTTs are 50 ms for an open browser tab and 3 s if the user switches to another tab.

Complete graphs of our systematic measurements comparing different attacker types (*spooky stranger* vs. *creepy companion*), access technologies (Wi-Fi vs. LTE), Applications (WhatsApp vs. Signal), and iPhone models (iPhone 13 Pro, iPhone 11) can be found in Appendix A5 and proof the feasibility of our monitoring across all tested environments. We also uploaded a video[5] demonstrating a single measurement case (*creepy companion*, 20 s interval, Wi-Fi on the iPhone 11, video speedup 30x). Initially, we conducted our measurements manually. However, we later automated the process using an ESP32, which emulates a Bluetooth keyboard and executes the keyboard inputs to switch between the required activity states.

For clarity and ease of demonstration, we isolate particular patterns and present graphs focusing on these findings under fixed conditions (i.e., a *creepy companion* using WhatsApp to track a target connected via Wi-Fi). However, complementary measurements (Appendix A5) showed that these patterns generalize and can similarly be observed under varying conditions.

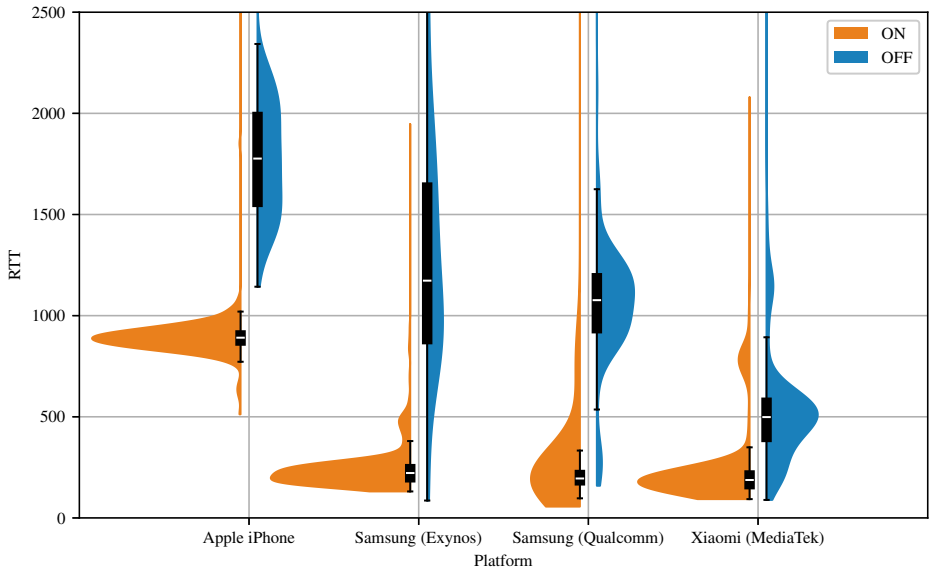
Showcase I: Deriving Smartphone Screen Time: While knowing (one own's) screen time is convenient for digital wellbeing and parental control, it is also interesting for external entities. For example, a nosy employer might want to know whether their employees use their (private) phone when at work or a marketing company might be specifically interested in targeting users with excessive screen time. Figure 4 compares the observed RTT for WhatsApp with an active and inactive screen state on the iPhone. An inactive screen leads to RTTs of about two seconds, an active of about one second. Extracting the screen time only worked with less frequent probing, e.g., one ping per 20 seconds as more frequent probes would have prevented the phone from pivoting into a deep sleep state.

Showcase II: Deriving IM Application Activity: Further, we show that even the use of an application could be extracted from the measured RTTs. In particular, we checked whether RTTs change if the target application is open, i.e., in foreground. An envious ex might be curious about how much time their former partner spends on the messaging app potentially chatting with new acquaintances. Figure 5 shows that the RTT drops to about 300 ms as soon as the application is opened on the phone. If the application is closed, i.e., moved to the background, the timings dwell in an intermediary state (RTT: 500 ms) for about 30 seconds before normalizing at their initial level (RTT: 1 s) as the screen is still active. To further support our findings, we looked for evidence of this behavior in the iPhone's system log (via *idevicesyslog*). We are able to confirm that the application is first put on hold leading to the observed intermediary state for 30 seconds before being moved into standby.

2) Behavior Fingerprinting on Android Devices: For Android, we likewise found characteristic patterns allowing to differentiate between screen and application activity states on a variety of phones. Due to the more diverse landscape of manufacturers, chipsets, OS flavors, and software versions, the patterns for certain activity states differ among models and need to be individually adjusted for each target device. Yet, some general rules hold across all tested device models, e.g., deeper standby states causing increased jitter and higher RTTs.

Besides identifying characteristic patterns, selecting an appropriate ping frequency greatly influences a measurement's outcome and thus needs to be fine-tuned for specific phone models. For example, on Samsung models, lower ping frequencies (e.g., 1 ping per minute) allow the phone to enter a deep sleep state, resulting in more distinct RTT differences between activity states. Conversely, on Qualcomm- and MediaTek-based Xiaomi phones, higher probing frequencies (e.g., 1 ping per second) do not disrupt the phone's standby behavior and still allow for a clear separation between active and inactive states. To compare characteristic screen on/off timings for various manufacturers and models, we measured the RTTs for a range of smartphones. Using a probing interval of one ping per second (to also cover standby states on Samsung models), we recorded delivery receipt RTTs for different screen states on each device. For each individual phone and state, we made

[5] https://drive.proton.me/urls/DHACRYX250#CLYkdE3Rb7Ho



Device models: Apple iPhone 11, Samsung Galaxy A54 5G, Samsung Galaxy S23, Xiaomi Poco M5s

Fig. 7: Characteristic screen on/off timings for different manufacturers and chipsets (all measured as *creepy companion* over Wi-Fi with 1 ping per minute). The box and violin plots show, that we observed differences between timings for the screen on vs. screen off state across all tested devices. While this only shows the overall RTT distribution, local patterns (e.g., jitter) could be used to further refine the distinction between states. The leftmost graph (iPhone) corresponds to the data that is shown in Figure 1.

sure to gather at least 300 data points (i.e., 5 hours of capture time). Figure 7 compares the RTT distribution for each screen state on our testing devices. Despite distinct characteristics across models, differentiating between screen-on and screen-off timings appears feasible for all devices. Two detailed plots that show characteristic patterns on Android phones can be found in Appendix A4.

3) Monitoring Behavior on Companion Devices: Besides examining characteristic RTT patterns on main devices, we analyzed the RTTs on desktop- and web-companion devices. We tried to differentiate between the active (i.e., application or corresponding browser tab in foreground) and inactive (i.e., application minimized or tab in background) state.

Figure 6 shows that it is trivial to differentiate by RTTs between an active and inactive (i.e., browser tab46904690 in the background on Firefox) WhatsApp Web session. While we got immediate responses (roughly within 50 ms) in the active state, responses took about 3s when another tab was focused or when the browser was minimized. Moreover, the high response times occurred as soon as another window was fully covering the canvas of the Firefox window (i.e., the supposed standby mechanism kicks in as soon as the WhatsApp window is not actively painted on the screen). Clearly, this behavior allows sophisticated tracking of the victim's Whatsapp usage within their browser. For Firefox, all tested Operating Systems (Windows, Linux, Mac) showed the same behavior. For the residual companion devices (other browsers and Desktop Apps), we did not see any obvious distinctions.

	OS	Delivery Receipts	Read Receipts
WhatsApp	Android	Separate	Stacked
	iOS	Separate	Stacked (Reversed)
	Web	Stacked	Stacked
	Windows	Stacked	Stacked
	macOS	Stacked (Reversed)	Stacked (Reversed)
Signal	Android	Separate	Stacked
	iOS	Separate	Stacked (Random)
	Desktop	Stacked	Stacked (Reversed)

TABLE V: Deivce/OS Fingerprinting: WhatsApp and Signal show different receipt handling for different platforms.

D. Device OS Fingerprinting

Section V-C showed differences in RTTs that might be abused to observe a victim's behavior, but we also noticed differences in the applications' implementations for different target architectures. This includes handling delivery- and read receipts, e.g., receipt ordering and receipt stacking, for missed messages due to the device being offline. Therefore, we sent multiple messages to an offline target. As soon as the target device went online again, the pending messages were fetched from the server and acknowledged via delivery receipts. As

Fig. 8: Real-World Tracking Scenario of two companion devices (web-client and native client) and the main device (smartphone) across different access technologies (LTE, Wi-Fi and Wired LAN) and usage scenarios (active usage, voice call, screen off phases), measured by a *spooky stranger* with one ping every 2 s.

soon as the target opened the conversation, it sent out the corresponding read receipts. Although the protocol supports stacking, i.e., combining the receipt of multiple messages into a list within one single receipt, some implementations always issue separate receipts. Whenever summarized receipts were used, the messages' order was found to differ among different implementations (natural order vs. reversed vs. randomized order). Table V summarizes the found discrepancies. While read receipts are only sent to *creepy companions* during an active conversation, the behavior regarding delivery receipts can also be measured by *spooky strangers*. These differences can be viewed as information disclosure that can be used to fingerprint a victim's system and to refine further exploits against the target.

E. Real World Tracking Example

While the previous examples clearly demonstrate the feasibility of extracting privacy-sensitive information, they were measured within a relatively stable and controlled environment (i.e., within testing devices in our lab). We further show that attacking a user's privacy is also feasible in an open-world scenario even with limited prior information and thus created a measurement under real-world settings, monitoring the phone (a Xiaomi Poco X3 NFC device) of a volunteering colleague

on their way to our lab. The volunteer afterward provided us with information about their devices and their actions during the capture, which was used to annotate the corresponding Figure 8. The sending device did not have any prior relation to the observed device, i.e., this measurement reflects the worst attack scenario of a *spooky stranger*.

Before we started the measurement, we inspected our victim's device list (cf. Section V-A): `[0, 1, 9]`. We can see that they currently have three existing sessions, one main device (index 0), and two companion devices (index 1, 9). Due to the (auto-incrementing) index, we know in which order the sessions were created, i.e., the first companion device corresponds to a relatively old and stable device, while device 9 is a newer (or potentially just temporary) session.

After starting our probing, we receive receipts from two – currently online– devices (cf. Section V-B): the main device and companion device 1 (our volunteer's desktop computer, running WhatsApp Web in the browser). According to our volunteer, device 1 is connected via LAN, which is reflected in the graph by very stable RTT timings with low jitter. At 19:28, our volunteer turned off their desktop computer (thus, no more receipts are received from this device), shortly before leaving their flat and heading to the office. The phone switched from Wi-Fi to LTE, which is reflected by a changed RTT pattern (slightly higher, but more dense RTT timings). On their walk towards the office, the victim issues a phone call, again resulting in a more dense receipt distribution (since the phone is in a high activity state). Finally, at 19:45, our victim arrives at the office, as their phone switches back from LTE to Wi-Fi. Shortly after, our volunteer turned on their work laptop (device 9), which synchronizes all missed messages, causing it to send a lot of receipts for the previous probing requests. Due to the reversed ordering of the stacked delivery receipts (cf. Section V-D), we know that this device is a macOS computer running the WhatsApp desktop client. Comparing the jitter and density of the RTT patterns of the two companion devices (devices 1 and 9) we can easily spot the difference between LAN and Wi-Fi.

F. Resource Exhaustion Attacks

Although covert messages are not displayed on the target's phone, they still use resources (e.g., traffic, battery, phone storage). To amplify the resource exhaustion that can be achieved with a single message, we attempted sending different message actions with different payload sizes, aiming to detect server- and client-side limits. Table VI shows the discovered size limits. Interestingly, WhatsApp uses different boundaries, depending on the message size, and allows message reactions to carry up to 1 MB of payload data. While the client-side limit of actually handling the message seems to be much lower (i.e., no delivery receipts are issued for reactions containing more than 30 bytes of data), the message is still received and processed before it is discarded.

Traffic Inflation. Our traffic inflation measurements for WhatsApp showed that an attacker can cause 3.7 MB per second (i.e., 13.3 GB per hour)[6] of data traffic for the victim without the latter receiving any notification in the application. This value was reached by a single client session continuously sending message reactions with 1 MB of payload and might be further amplified using multiple clients or sessions. The attack covertly inflates a victim's data bill and might use the bandwidth planned for other applications, potentially leading to their denial of service.

	Send	Edit	React	Delete	Consumable Data
	65	65	1,000	-	13,320 MB/h
	194	194	194	-	360 MB/h

TABLE VI: Server-side payload limits (in KB) for different message types. (Invalid) reactions can contain arbitrary data and are not displayed at the target. Thus, besides abusing them for resource exhaustion, they could also be utilized as a covert channel or for data exfiltration.

Battery Drainage. Besides using up a user's data allowance, receiving many and large messages additionally drains the smartphone's battery. We measured the battery exhaustion on three phones by blasting large reaction messages via WhatsApp for a period of one hour. While regular (idle) battery drainage for all phones was less than 1 % per hour, we were able to drain a considerable share of the battery (iPhone 13 Pro: 14 % per hour, iPhone 11: 18 % per hour, Samsung Galaxy S23: 15 % per hour)[7]. During our tests the phones were on normal standby (screen off), connected to Wi-Fi and all attacks were executed by *spooky strangers*. For Signal, we were not able to considerably drain the battery of our testing phone (iPhone 13 Pro). Due to considerably stricter rate limits, it only decreased by 1 % after an hour of attack.

VI. Related Work

Mobile Instant Messaging Security and Privacy. Instant messenger security has been investigated since their early days. Schrittwieser et al. [22] analyzed attack vectors exploiting insufficient authentication in nine messenger applications. Back then, WhatsApp was found to be vulnerable to account hijacking, the unauthorized modification of the users' status pages, the unlimited delivery of unrequested SMS, and user account enumeration. A subset of attacks continued to exist over multiple years [9], [14], [15], [19] and user account enumeration even become feasible for the more security-oriented messenger Signal [14], [15]. With the advent of E2EE, Be'ery [2] and Gegenhuber et al. [8] discussed undesirable leaks of multi-device architectures, relevant for WhatsApp. For E2EE, one public key has to be maintained per device in the application's inventory and user behavior (i.e., addition or change of device) might be observed by changes

[6]Capture period: 2 hours. In addition to our captured traffic dump, both the phone's system-level data usage statistics and WhatsApp's internal data consumption view confirmed the volume of traffic generated.

[7]Again, the system's battery usage overview confirmed that WhatsApp was responsible for the observed battery drain.

in an account's public keys. Based on the delivering session, a receiver is also able to infer the sender's device issuing a message. Our research emphasizes that this also holds for Signal. Beyond, we discovered further privacy implications by exploiting delivery receipts as a side channel for WhatsApp and Signal.

Delivery and Read Receipts. Instant messengers typically acknowledge receipt and reading of a message. Reading receipts facilitate stalking, e.g., in the context of intimate partner abuse [7], even by weak adversaries that are bound to the messenger's regular user interface. It is now possible to disable these reading receipts, but delivery receipts are continuously returned by WhatsApp, Signal and Threema. Simulating a regular WhatsApp conversation with the victim, delivery receipts were used to narrow down user location (e.g., UAE vs Germany) and to distinguish between cellular- and Wi-Fi-based connections [21]. Thereby, the adversary requires an on-going conversation with the victim and each probe triggers another message, i.e., the attack remains overt to the victim. The principal idea has been transferred to cellular networks [3], and later extended by multi-location measurements to improve accuracy [4]. The latter approaches rely on delivery reports of silent SMSes. Thereby, the victim remains unaware of the attack, enabling geolocation also at unusual time of the day or at regular intervals. In our work, we show how to trigger delivery receipts of instant messengers without any notification of the victim and in the absence of an on-going conversation, rendering WhatsApp- and Signal-based attacks as stealthy as silent SMS-based ones. Overall, the impact on privacy is more substantial as we are not only able to infer a user's geolocation but also more detailed information on device activity and user behavior (e.g., screen on/off, browser active/inactive).

Battery Drain of Instant Messengers. Battery draining attacks had already been known in the era of feature phones [20] exploiting MMS services to consume battery up to 22 times faster, and only later transferred to smart phones [18]. Regular battery drain of instant messengers like WhatsApp has been investigated as early as 2014, proposing message bundling to save energy [27]. More recent work on mobile Tor use, points into a similar direction [16]. Consumption is primarily caused by radio transmission and might be reduced by adequate message scheduling, whereas consumption due to cryptography is negligible. The significance of our battery drain attacks lies in its versatility and stealth. Two billion WhatsApp users, i.e., a fourth of the world population, might become a victim to our attack, again an on-going conversation with the adversary is not a prerequisite.

Covert Channels. Camoufler uses the Signal infrastructure as a tunnel to evade Internet censorship [23] as censors fear collateral damage caused by the prohibition of popular messaging applications. Our covert channels follow a different rationale. Instead of disguising traffic from a censor, they evade visible representation in the messenger's user interface.

Security and Privacy Issues in Cellular Services. Beyond Over-the-Top (OTT) applications like WhatsApp and Signal, prior work has demonstrated that tracking a mobile user's geolocation (e.g., under roaming conditions) is possible via the traditional cellular network [3], [4], [12]. 3GPP-standardized messaging services that are terminated over third-party Internet connections, such as VoWiFi and RCS, have also been shown to be vulnerable to various security and privacy issues [10], [11], [26], [30]. In contrast, the present work does not rely on native 3GPP services but instead requires an additional OTT application to be installed on the phone.

VII. Mitigations

Restricting Delivery Receipts. Our measurements show that all three analyzed messengers also send delivery receipts for unknown users that are not in the victim's contact list. Restricting this feature to real conversations and automatically dropping messages or preventing receipts for unknown numbers would hinder *spooky strangers* from tracking arbitrary victims. Additionally, privacy-conscious users should be able to disable the instant transmission of delivery receipts.

Coarser Receipt Timings. Letting the sender know that a message was successfully received can be a convenient feature in an asynchronous conversation. However, there are no strict real-time requirements, i.e. the perceived experience does not change when this information is only updated after a few seconds. Adding noise to these acknowledgment timings would easily prevent tracking (i.e., geolocation- and activity monitoring) based on the receipt's RTTs.

Improve Client-side Validation. When messages are not E2EE, they can be validated by the server and only forwarded to the receiver when passing the validation. However, this server-side validation is not possible with E2EE, requiring more rigorous validation by the receiving client. For example, many of the presented attacks are not possible when clients properly validate the referenced message IDs and thus discard invalid messages (instead of acknowledging them via a delivery receipt). While our primary focus is on privacy-related issues, the shift from server-validated input to E2EE content is particularly important from a security standpoint. Parsing unvalidated data can quickly introduce severe security vulnerabilities.

Rate Limiting. In our measurements, we were able to drain a user's data quota and battery by sending large messages over a prolonged time. In contrast, regular (text) messaging only needs very limited bandwidth (note: media messages are usually transmitted over separate media servers). Thus, employing restrictive messaging rate limits on the server side could mitigate these attacks. Moreover, receiving an excessive amount of messages could also be automatically detected by the receiver and then trigger a UI notification and (temporarily) block the corresponding phone number.

Syncronized Multi-Clients. To cope with multi-device leakage, devices could prioritize synchronizing their state before

issuing receipts for recent messages. While only introducing a minor timing overhead, this would ensure that a delivery receipt is just sent once. Alternatively, other proposals for multi-device protocols consider hiding the amount of companion devices [5].

Harmonizing Client Behavior. Supporting different operating systems often requires having multiple codebases that are written in different programming languages. In many cases, these different implementations behaved inconsistently in how they responded to specific messages, introducing fingerprinting possibilities for the attacker. Harmonizing client behavior or moving towards a single code base that can be used across different platforms could solve this problem.

VIII. Discussion

Delivery receipts have already been known as veritable timing side channels compromising user privacy. In this work, we discovered the existence of stealth delivery receipts in two major messaging services – market leader WhatsApp and its security-oriented alternative Signal. This way, an adversary is able to trigger delivery receipts at another user's client without leaving a trace for the latter – indeed, stealth probing makes long-term and high-frequency probing only possible in the first place and surpasses previous possibilities of remote observation by far. State-of-the-art E2EEE encryption, requiring an individual encryption key per registered device, only exacerbates the situation as it enables the attribution of messages to a user's different devices (mobile phone, desktop, web). This way, we are able to remotely create comprehensive observation profiles of victims solely by observing delivery receipts. Beyond that, we are able to stealthily launch resource exhaustion attacks (data quota, battery) against mobile phones.

Our attacks' impact is significant: First, the requirements for the attacker are low. They only need a phone number for registration at the messaging service and a mobile phone – a prepaid card and an older phone model are perfectly adequate. Second, with more than two billion users on WhatsApp, the number of potential victims is vast. According to our results, Signal – dedicatedly developed with security and privacy in mind – does not appear to protect its users better and might even put high-profile users like US Senate [29] and European Commission [6] staff at risk. Moreover, recent media revelations have shown that high-ranking US officials, including the Secretary of Defense, use both Signal and WhatsApp for personal and professional communication [13], and in some cases even have their phone numbers publicly accessible online [1], making them easy targets for such attacks. The quality of the results varies somewhat between different phones, but our results pinpoint that practically all of them are affected. But even when only looking at the iPhone with its particular clear measurement results, its worldwide market share of 20 to 30 % renders several hundred million people vulnerable.

From a user perspective, the situation is particularly dire as an individual cannot take any protective measures except the complete deinstallation of the service. Due to the attacks' stealth, they do not realize an ongoing attack either (with the obvious exceptions of drained battery and overdrawn quota as a consequence of resource exhaustion attacks). Moreover, users are unable to locate the source of the attack – it is important to note that the adversary does not need to be in the victim's contact list. It is therefore essential that the operators of the messaging server take action and implement security measures like those suggested in Section VII.

A. Ethical Considerations

In the course of our research, we only probed WhatsApp, Signal, and Threema accounts belonging to and used by the authors of this paper, all of whom provided their explicit consent. Most of the accounts used in our experiments were test accounts created specifically for the purpose of these measurements. The devices used to explore and demonstrate the impact of various usage and environmental scenarios were either i) dedicated lab devices intended solely for research purposes, or ii) personal devices owned by the test operator (i.e., one of the authors). The volunteer involved in the real-world experiment described in Section V-E was also one of the authors, and was therefore fully informed about the associated security and privacy implications. Tracking was conducted strictly within the agreed-upon duration of the experiment.

Server Infrastructure. As the traffic is E2EE, the messenger infrastructure remains unaware of and thus unaffected by non-compliant messages, e.g., the deletion/modification or reaction to nonexistent messages. The application servers only see the overall traffic volume and pattern that is forwarded from our sending to the receiving accounts. We assume that the messengers' infrastructure is laid out for massive data forwarding, and consider our maximum rate of 3.7 MB/s, as caused by the resource exhaustion attacks, a moderate load for an infrastructure serving more than two billion users.

Responsible Disclosure. Our attacks heavily inflict user privacy and additionally enable resource exhaustion (data quota, battery). Beyond, they affect all (> 3 billion) users of the messenger platforms WhatsApp and Signal. For both messenger operators, we consequently submitted our findings to their security contacts on September 5th, 2024. On September 24th, 2024, we received a confirmation receipt from Meta, responsible for WhatsApp, indicating that our results had been passed on to the relevant development team but have not received a substantive response ever since. On August 8th, 2025, more than 11 months after the initial report, we once again received the information that the report was reviewed by the security team and has been forwarded to the relevant engineering team.

As of November 14th, 2024, it appears that the Firefox activity leakage as described in Section V-C3 has been fixed but we did not receive any more detailed information.

From the Signal Technology Foundation, we did not get any answer at all.

Open Science. We believe in open science and therefore plan to release our modified clients in the future. However, we will not release them at the current publication date (2025-08-14), as the issues we identified have not yet been addressed by the platform operators. Through the responsible disclosure process, we expect these vulnerabilities to be resolved eventually, ensuring that our modified clients will no longer pose any risk of misuse.

IX. Conclusion

In this work, we demonstrated that modern E2EE messaging architectures like WhatsApp and Signal unintentionally expose privacy-sensitive information about their users. Specifically, an adversary armed with only a target's phone number can determine the exact amount, type, and online status of the target's devices. Furthermore, detailed behavioral patterns, such as screen time or messaging app usage duration, can be inferred with a resolution down to the second.

This vulnerability is exploited through covert probing messages that trigger delivery receipts without generating any notification within the targeted application, akin to a stealth SMS. Additionally, the structure of E2EE messaging, combined with the absence of server-side message quotas, enables attackers to misuse these capabilities for resource exhaustion attacks draining a target's battery or data allowance. Notably, there is currently hardly anything a targeted user can do about this for multiple reasons. These attacks neither cause any notification on the targeted device, nor require an active conversation between the attacker and the target, nor can the attacking account be blocked or reported, nor is the deactivation of delivery receipts entirely possible at the moment.

Our findings reveal that mechanisms embedded in modern E2EE messaging architectures – such as delivery receipts and multi-device support – can have significant implications on user privacy. Consequently, it is essential to balance functional requirements, usability and convenience with privacy and security, particularly in E2EE applications that are inherently privacy-sensitive per design.

Acknowledgment

This material is based upon work partially supported by (1) the University of Vienna, Faculty of Computer Science, Security & Privacy Group, (2) the University of Vienna, Faculty of Computer Science, Communication Technologies Group, (3) the FFG Bridge project 46322124 SecKey, (4) the FFG KIRAS/K-PASS project 59103683 TelCrit, (5) the Austrian Science Fund (FWF) (SFB SPyCoDe F85), (6) SBA Research (SBA-K1 NGC) is a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMIMI, BMWET, and the federal state of Vienna. The COMET Programme is managed by FFG.

References

[1] Patrick Beuth, Jörg Diehl, Roman Höfner, Roman Lehberger, Friederike Röhreke, and Fidelius Schmid. Private Data and Passwords of Senior U.S. Security Officials Found Online. Der Spiegel, 2025. Accessed: 2025-04-24.

[2] Tal A. Beery. WhatsApp with privacy? Privacy issues with IM E2EE in the Multi-device setting. In *18th USENIX WOOT Conference on Offensive Technologies (WOOT 24)*, pages 11–16. USENIX Association.

[3] Evangelos Bitsikas, Theodor Schnitzler, Christina Pöpper, and Aanjhan Ranganathan. Freaky Leaky SMS: Extracting User Locations by Analyzing SMS Timings. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2151–2168, Anaheim, CA, August 2023. USENIX Association.

[4] Evangelos Bitsikas, Theodor Schnitzler, Christina Pöpper, and Aanjhan Ranganathan. Amplifying Threats: The Role of Multi-Sender Coordination in SMS-Timing-Based Location Inference Attacks. In *18th USENIX WOOT Conference on Offensive Technologies (WOOT 24)*, pages 59–73, Philadelphia, PA, August 2024. USENIX Association.

[5] Sébastien Campion, Julien Devigne, Céline Duguey, and Pierre-Alain Fouque. Multi-Device for Signal. page 1363.

[6] Laurens Cerulus. EU Commission to staff: Switch to Signal messaging app. *Politico*, 2020. Retrieved Sep 4th, 2024 from https://www.politico.eu/article/eu-commission-to-staff-switch-to-signal-messaging-app/.

[7] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. "A Stalker's Paradise" How Intimate Partner Abusers Exploit Technology. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018.

[8] Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, and Aljosha Judmayer. Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp's Handshake Mechanism, 2025.

[9] Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Johanna Ullrich, and Aljosha Judmayer. Hey there! You are using WhatsApp: Enumerating Three Billion Accounts for Security and Privacy, 2025.

[10] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2024)*, 2024.

[11] Gabriel K. Gegenhuber, Florian Holzbauer, Philipp É. Frenzel, Edgar Weippl, and Adrian Dabrowski. Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 451–468, Philadelphia, PA, August 2024. USENIX Association.

[12] Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, and Adrian Dabrowski. MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3493–3510, Anaheim, CA, August 2023. USENIX Association.

[13] Jeffrey Goldberg. The Trump Administration Accidentally Texted Me Its War Plans. The Atlantic, 2025. Accessed: 2025-04-24.

[14] Christoph Hagen, Christian Weinert, Christoph Sendner, Alexandra Dmitrienko, and Thomas Schneider. All the numbers are US: Large-scale abuse of contact discovery in mobile messenger. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, San Diego, California, USA, February 21 - February 25, 2021*. The Internet Society.

[15] Christoph Hagen, Christian Weinert, Christoph Sendner, Alexandra Dmitrienko, and Thomas Schneider. Contact Discovery in Mobile Messengers: Low-cost Attacks, Quantitative Analyses, and Efficient Mitigations. *ACM Trans. Priv. Secur.*, 26(1), nov 2022.

[16] Daniel Hugenroth and Alastair R. Beresford. Powering Privacy: On the Energy Demand and Feasibility of Anonymity Networks on Smartphones. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5431–5448, Anaheim, CA, August 2023. USENIX Association.

[17] Moxie Marlinspike and Trevor Perrin. The Sesame Algorithm: Session Management for Asynchronous Message Encryption. Retrieved Aug 26th, 2024 from https://signal.org/docs/specifications/sesame/sesame.pdf.

[18] Benjamin R. Moyers, John P. Dunning, Randolph C. Marchany, and Joseph G. Tront. Effects of Wi-Fi and Bluetooth Battery Exhaustion Attacks on Mobile Devices. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–9, 2010.

[19] Robin Mueller, Sebastian Schrittwieser, Peter Fruehwirt, Peter Kieseberg, and Edgar Weippl. What's new with WhatsApp & Co.? Revisiting the Security of Smartphone Messaging Applications. iiWAS '14, page 142–151, New York, NY, USA, 2014. Association for Computing Machinery.

[20] Radmilo Racic, Denys Ma, and Hao Chen. Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. In *2006 Securecomm and Workshops*, pages 1–10, 2006.
[21] Theodor Schnitzler, Katharina Kohls, Evangelos Bitsikas, and Christina Pöpper. Hope of delivery: Extracting user locations from mobile instant messengers. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society.
[22] Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar Weipp. Guess who is texting you? Evaluating the security of smartphone messaging applications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5 - February 8, 2012*. The Internet Society.
[23] Piyush Kumar Sharma, Devashish Gosain, and Sambuddho Chakravarty. Camoufler: Accessing The Censored Web By Utilizing Instant Messaging Channels. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, ASIA CCS '21, page 147–161, New York, NY, USA, 2021. Association for Computing Machinery.
[24] Statcounter. Mobile Vendor Market Share United States Of America. Retrieved Sept 05th, 2024 from https://gs.statcounter.com/vendor-market-share/mobile/united-states-of-america.
[25] Threema. Threema for iOS: Preview of Upcoming Multi-Device Functionality. https://threema.ch/en/blog/posts/ios-test-multi-device, dec 2022. Accessed: 2024-11-14.
[26] Guan-Hua Tu, Chi-Yu Li, Chunyi Peng, Yuanjie Li, and Songwu Lu. New security threats caused by IMS-based SMS service in 4G LTE networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.
[27] Ekhiotz Jon Vergara, Simon Andersson, and Simin Nadjm-Tehrani. When mice consume like elephants: instant messaging applications. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, page 97–107, New York, NY, USA, 2014. Association for Computing Machinery.
[28] WhatsApp. About WhatsApp, 2024. Retrieved Aug 26th, 2024 from https://www.whatsapp.com/about/.
[29] Zach Whittaker. In encryption push, Senate staff can now use Signal for secure messaging . *ZDNet*, 2017. Retrieved Sep 4th, 2024 from https://www.zdnet.com/article/in-encryption-push-senate-approves-signal-for-encrypted-messaging/.
[30] Yaru Yang, Yiming Zhang, Tao Wan, Chuhan Wang, Haixin Duan, Jianjun Chen, and Yishen Li. Uncovering Security Vulnerabilities in Real-world Implementation and Deployment of 5G Messaging Services. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2024.

APPENDIX

A. Messenger Infrastructure Analysis

As a basis for our research, we investigated the messaging services' infrastructure for submitting and receiving messages and provide novel insights. Applying source code analysis and inspecting real-world network traffic of our mobile devices, we discovered the relevant web endpoints and domains, see Table VII, in a first step.

App	Domain
	web.whatsapp.com (web), g.whatsapp.net (mobile)
	chat.signal.org
	ds.g-xx.0.threema.ch

TABLE VII: Domain names that are used to connect to the messaging services (all available via dual-stack). Usually a websocket connection (port 443) is used. The only exception are mobile WhatsApp clients connecting directly via port 5222 (xmpp).

In a second step, we investigated these endpoints from different vantage points in the AWS cloud in a threefold manner:

- (L1) DNS resolution of the endpoint domains,
- (L2) measurement of the application-agnostic latency by probing with ICMP and TCP,
- (L3) measurement of application-level latency between a messaging client and the server with the applications' keepalive/heartbeat functionality.

While (L1) and (L2) is feasible for any domain name, (L3) requires client emulation and thus use and adaptation of the software projects presented in Table II.

WhatsApp: WhatsApp uses GeoDNS to route traffic from clients, based on their source IP address, to different target addresses, see also previous work on these aspects [21]. A reverse DNS lookup of the latter addresses reveal their domain names, containing the three letter airport code of the edge locations (e.g., `whatsapp-cdn-shv-01-vie1.fbcdn.net`for Vienna, Austria). Ensured by GeoDNS, client and edge locations are close, leading to (L2) latencies of 1 - 10 ms when measured from our AWS instances.

Our measurements for (L3) latencies measured with WhatsApp keepalives are however significantly higher. Further analysis showed, that the edge location revealed by DNS only plays a minor role for the overall messaging RTTs, since they only serve as an entry node to Meta's internal network.

In fact, message delivery is handled via a lower number of centralized messaging servers. We discovered eight such servers (represented by a three letter location attribute), three within Europe and six in the US:

- **`odn`**: Odense, Denmark
- **`cln`**: Clonee, Ireland
- **`lla`**: Luleå, Sweden
- **`frc`**: Forest City, North Carolina, US
- **`atn`**: Altoona, Iowa, US
- **`nao`**: New Albany, Ohio, US
- **`rva`**: Sandston, Virginia, US
- **`vll`**: Huntsville, Alabama, US
- **`cco`**: Prineville, Oregon, US (same as `prn`)

The selection of the central messaging server is influenced by the client's `routing_info` cookie. If it is empty, as for example in the very first connection attempt, a random messaging server is assigned. Upon re-connection, the client communicates its cookie in the connection handshake, indicating the previously used (and usually closest) location, effectively pinning the proposed server.

In conclusion, a message between two messaging clients that are connected to the same edge location cannot be directly forwarded, but rather takes a detour via the clients' messaging servers.

Signal: Signal uses a set of static IP addresses for all messenger clients ((L1)). Pinging these IP addresses ((L2)) from our AWS instances results in RTT of less than 1 ms

Device	Modem Chipset	OS	WhatsApp	Signal
iPhone 13 Pro	Qualcomm	iOS 17.6.1	2.24.17.78	7.26
iPhone 11	Intel	iOS 17.6.1	2.24.17.78	7.26
Samsung Galaxy S23	Qualcomm	Android 14	2.24.17.79	7.15.4
Samsung Galaxy A54 5G	Exynos	Android 14	2.24.17.79	7.15.4
Xiaomi Poco M5s	MediaTek	Android 13 (MIUI 14.0.4)	2.24.16.10	7.15.4
Xiaomi Poco M3 Pro 5G	MediaTek	Android 13 (MIUI 14.0.4)	2.24.16.10	7.15.4
Xiaomi POCO X3 NFC	Qualcomm	Android 12 (MIUI 14.0.5)	2.24.16.10	7.15.4

TABLE VIII: Overview of the devices including software versions that were used throughout our tests.

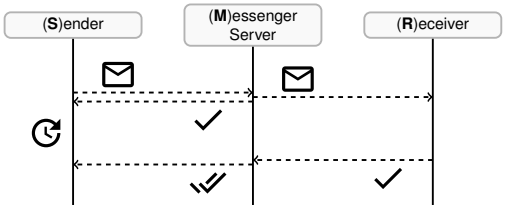


Fig. 9: Whats App Message Flow as discovered by Schnitzler et al. [21]. Our measurements show that these servers only serve as entry points to the Meta network.

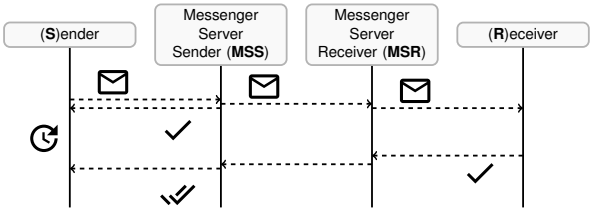


Fig. 10: Updated WhatsApp Message Flow: Sender and receiver each connect to one of eight messenger servers. Messages are forwarded by both servers to reach the intended destination.

from all locations suggesting the use of Anycast routing. A reverse DNS lookup reveals that these IP addresses belong to AWS Global Accelerator[8], a service providing a static entry to applications hosted within the Amazon cloud. Measuring the (L3) RTT from all 29 AWS EC2 regions suggests that the service is hosted within *us-east-1*. Measuring from there, we see an application-layer keepalive RTT of only 3 ms. In comparison, the corresponding RTT from *us-west-1* is 62 ms and goes up to 252 ms for *ap-southeast-3*.

Threema: Threema uses static unicast IP addresses for all messenger clients ((L1)). According to their website[9], they host their infrastructure in Zurich, Switzerland. Our plausibility checks confirm this as we see low RTTs for measurements from vantage points in Central Europe ((L2): 5 ms, (L3): 50 ms) and increased RTT from the US (e.g., (L2): 90 ms, (L3): 140 ms).

B. Testing Devices

We conducted our tests on seven devices from three different vendors, of which two run iOS and five Android. To maximize the validity of our results, we included devices from the top three global manufacturers for both smartphone brands (Apple, Samsung, Xiaomi) and modem chipsets (MediaTek, Qualcomm, Exynos). All devices operated on up-to-date software, including both the underlying OS and target applications.

C. Geolocation with Delivery Receipts

Sending a message to a receiver triggers two acknowledgments, see Figure 9. First, the messenger server acknowledges the receipt of the message and forwards it to the receiver. The receiver returns a delivery receipt to the server that is eventually forwarded to the sender. Previous work [21] subtracted the RTT between message sending and receipt of the first acknowledgment from the total RTT to estimate the RTT between the server and the receiver. The latter is then used for coarse geolocation of smartphones (e.g., UAE vs. Germany). Based on our insights on messaging infrastructure, all three messengers facilitate such coarse geolocation. For Signal and Threema, an adversary is able to measure the RTT between the victim and the central server in Amazon's *us-east-1* region and Zurich, respectively.

Our infrastructure analysis however refines previous results [21] for WhatsApp, see our updated message and delivery receipt flow in Figure 10. First, the message is forwarded to the sender's messenger server triggering an acknowledgment; then, forwarded to the receiver's messenger server before eventually reaching the receiver. The latter then issues a delivery receipt that is forwarded by both servers before reaching the sender. At first sight, WhatsApp, providing a total of eight such messaging servers appears to allow multilateration i.e., the measurement from multiple vantage points for more precise geolocation. This is however not true as the victim individually chooses its server via the `routing_info` cookie. The adversary is only able to discover this server by choosing the one with the lowest latency after iterating through all of them. Once connected to the same server, the adversary is again able to conduct the same coarse-grained geolocation as with Signal or Threema.

D. Characteristic Patterns for Android Phones

On Android, different probing frequencies worked differently well, depending on the target phone model. Two char-

[8] `aws.amazon.com/global-accelerator/features`
[9] `threema.ch/en/faq/server_location`

acteristic examples are shown in Figure 11 and Figure 12.

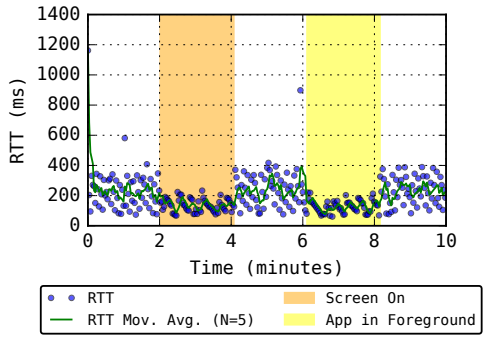


Fig. 11: For the MediaTek-based Xiaomi Poco M3 Pro 5G, we used higher probing frequencies (one ping every 2 s) to differentiate between an active and inactive screen with second-level granularity (WhatsApp, *creepy companion*, Wi-Fi).

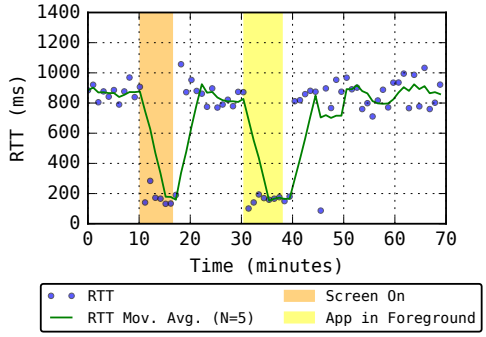
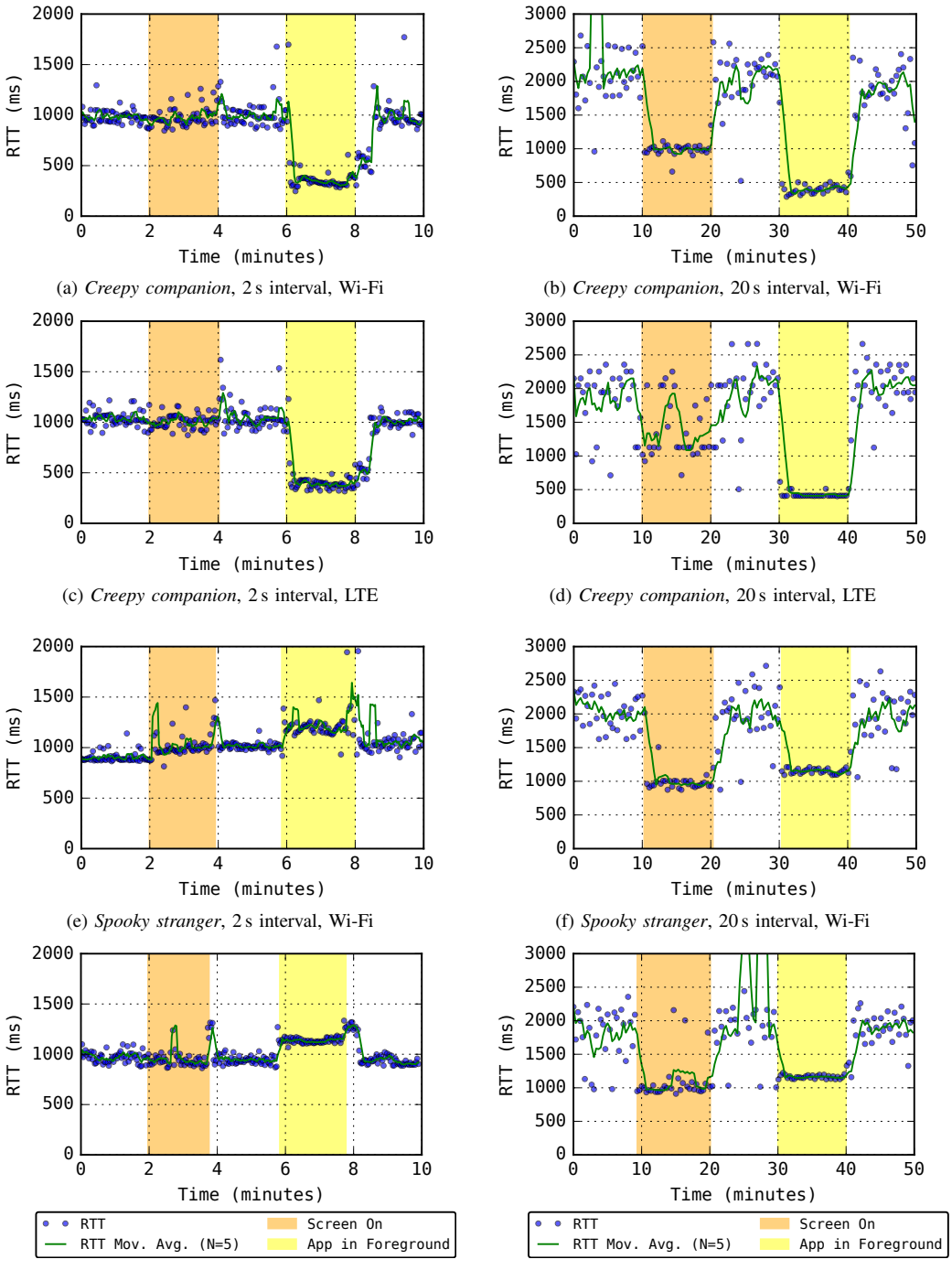


Fig. 12: For the Samsung Galaxy S23, we needed to lower the probing frequency to one ping per minute, to be able to differentiate between an active and inactive screen (WhatsApp, *creepy companion*, Wi-Fi).

E. iPhone

We systematically measured the RTTs within different environments (e.g., on an iPhone 13 Pro and iPhone 11, and via a cellular vs. LTE data connection) and applications (i.e., WhatsApp and Signal). The methodology was switching between active and inactive phases. The first active phase corresponds to a normal phone unlock (e.g., with an active homescreen), while the second active phase corresponds to the IM application being in the foreground. Each phase corresponds to 2 minutes for the measurements with one ping every 2 seconds and 10 minutes for one ping every 20 seconds respectively.

Figure 13 shows that for WhatsApp, the observed timings for *creepy companions* and *spooky strangers* differ. For Signal (cf. Figure 14) there is no such differentiation (i.e., probing as a *spooky stranger* leads to the same RTTs as probing as a *creepy companion*). Comparing the plots for WhatsApp and Signal, we see that some OS-specific patterns (e.g., application switch from foreground to background, as presented in Figure 5) occur across both applications. Note that, WhatsApp shows a counter-intuitive pattern for all four *spooky strangers* cases (Figure 13e to Figure 13h), since the RTTs within the *WhatsApp in Foreground* phase are consistently higher than in the *Screen On* phase. We verified that this is NOT a measurement error, by repeating the corresponding measurements multiple times. Further analysis on systems level as well as access to WhatApp source code, could improve the accuracy of predicated usage patterns.



(a) *Creepy companion*, 2 s interval, Wi-Fi

(b) *Creepy companion*, 20 s interval, Wi-Fi

(c) *Creepy companion*, 2 s interval, LTE

(d) *Creepy companion*, 20 s interval, LTE

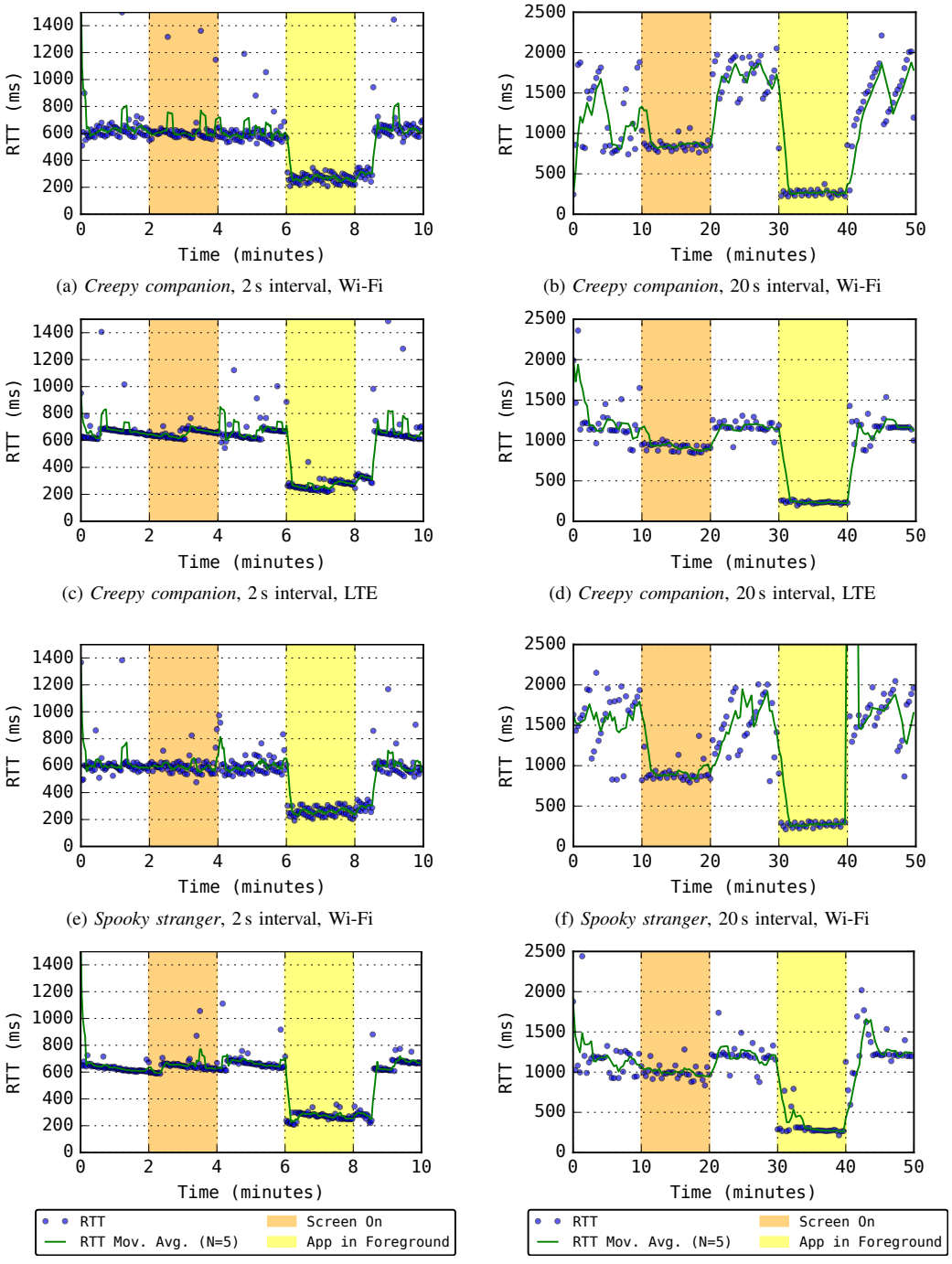
(e) *Spooky stranger*, 2 s interval, Wi-Fi

(f) *Spooky stranger*, 20 s interval, Wi-Fi

(g) *Spooky stranger*, 2 s interval, LTE

(h) *Spooky stranger*, 20 s interval, LTE

Fig. 13: Comparison of different probing intervals (2 s, 20 s), scenarios (*creepy companion*, *spooky stranger*), and access technologies (Wi-Fi, LTE) for WhatsApp (measured on an iPhone 11)



(a) *Creepy companion*, 2 s interval, Wi-Fi

(b) *Creepy companion*, 20 s interval, Wi-Fi

(c) *Creepy companion*, 2 s interval, LTE

(d) *Creepy companion*, 20 s interval, LTE

(e) *Spooky stranger*, 2 s interval, Wi-Fi

(f) *Spooky stranger*, 20 s interval, Wi-Fi

(g) *Spooky stranger*, 2 s interval, LTE

(h) *Spooky stranger*, 20 s interval, LTE

Fig. 14: Comparison of different probing intervals (2 s, 20 s), scenarios (*creepy companion*, *spooky stranger*), and access technologies (Wi-Fi, LTE) for Signal (measured on an iPhone 13 Pro)

8 Exploits via E2EE Prekeying Mechanism on Instant Messengers

Publication Info

Title	Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp's Handshake Mechanism
Authors	Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Aljosha Judmayer
Publication Status	This paper is included in the Proceedings of the Proceedings of the 19th USENIX Conference on Offensive Technologies (WOOT 25), pp. 209–227, 2025.
Publication Page	`https://www.usenix.org/conference/woot25/presentation/gegenhuber`
Code Artifacts	`https://github.com/sbaresearch/prekey-pogo`
arXiv	`https://arxiv.org/abs/2504.07323`
Reference	[GFGJ25]

Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp's Handshake Mechanism

Gabriel K. Gegenhuber[1,2], Philipp É. Frenzel[3], Maximilian Günther[1], and Aljosha Judmayer[1]

[1]University of Vienna, Faculty of Computer Science
[2]UniVie Doctoral School Computer Science
[3]SBA Research

Abstract

WhatsApp, the world's largest messaging application, uses a version of the Signal protocol to provide end-to-end encryption (E2EE) with strong security guarantees, including Perfect Forward Secrecy (PFS). To ensure PFS right from the start of a new conversation –even when the recipient is offline– a stash of ephemeral (one-time) prekeys must be stored on a server. While the critical role of these one-time prekeys in achieving PFS has been outlined in the Signal specification, we are the first to demonstrate a targeted depletion attack against them on individual WhatsApp user devices. Our findings not only reveal an attack that can degrade PFS for certain messages, but also expose inherent privacy risks and serious availability implications arising from the refilling and distribution procedure essential for this security mechanism.

1 Introduction

WhatsApp is the world's largest messaging application, with more than 3 billion users worldwide [31]. Under the hood, WhatsApp uses its own version of the Signal protocol for end-to-end encryption (E2EE) of messages [2].

The Signal protocol suite consists of several different protocols [19–22, 27] which together form one of the best end-to-end encrypted communication options available to end users today. Parts of the protocol suite have also been formally analyzed and proven secure in their respective security models [3, 6, 9, 10, 12]. Nonetheless, it remains crucial to continuously analyze protocols in their entirety—including their real-world composition and implementations—to uncover and test new attack strategies, identify real-world limitations, and enhance them accordingly. For our research, we depleted the *ephemeral prekeys* (also *one-time prekeys*) of our test accounts to analyze attacks on *perfect forward secrecy* (PFS) and to highlight novel privacy and availability implications arising from the current replenishing and distribution mechanisms for such prekey bundles.

The importance of one-time prekeys for the PFS of initial messages has already been noted in the specification of Signal's X3DH protocol [22]:

"This reduction in initial forward secrecy could also happen if one party maliciously drains another party's one-time prekeys, so the server should attempt to prevent this, e.g. with rate limits on fetching prekey bundles."

To the best of our knowledge, we are not only the first to test this concrete attack against forward secrecy, but also the first to analyze its feasibility and the general implications of this feature regarding the privacy of users. Hereby, we not only show that WhatsApp currently does not employ any rate limiting on fetching prekey bundles of participants, but also highlight that the lack of a detailed specification on how to handle and replenish ephemeral one-time prekeys, allows for device fingerprinting and gives away the online status of the targeted device. Moreover, extensively querying prekey bundles for a targeted account may cause errors, potentially preventing the retrieval of *any* prekey bundle for that account (even without one-time prekeys). As a result, no one would be able to establish new chat sessions with the victim, leading to an availability issue. While PFS is undoubtedly affected as well, we consider the real world confidentiality impact of this attack to be modest. This is due to the careful design and a clever defense-in-depth strategy of the Signal protocol suite. After being able to circumvent the noise protocol [26], an attacker would still need to get his hands on the long- and medium-term keys of a victim to exploit the lack of forward secrecy and decrypt the previously recorded messages. Even without one-time prekeys, the "self-healing" properties of the double ratchet [27] restore forward secrecy after the first round trip. Nevertheless, the attack on PFS shows that the strong claim from the WhatsApp whitepaper, would at least require a footnote that this currently might not hold for all messages:

"Due to the ephemeral nature of the cryptographic keys, even in a situation where the current encryption keys from a user's device are physically compromised, they cannot be used to decrypt previously transmitted messages." [2]

1.1 Threat Model

We consider two different attack models: A *PFS attack model* where the attacker is assumed to have far-reaching capabilities, as well as a *privacy and availability attack model* where the sole requirement for the attacker is having a WhatsApp account and the phone number of the target.

1.1.1 PFS Attack Model

The goal of the attacker in this case is as follows:

(G1) **PFS Degradation and Exploitation:** Force Bob's communication partners to send initial messages without forward secrecy and exploit this by recording the respective messages for possible decryption later on after Bob's long- and medium- term secret keys have been compromised.

Attacker Capabilities. In this attack, we assume a passive attacker that has access to WhatsApp's data center internal network communication, s.t., they are able to gain access to end-to-end encrypted messages of WhatsApp users. In other words, the attacker is able to strip the first layer of transport encryption – usually provided by TLS or the Noise protocol framework between the client and the server [26] – on top of the E2EE communication used in the Signal protocol suite. This is not an unrealistic scenario if a nation state actor, or compromised WhatsApp inter-server communication (e.g., by an internal employee), is considered. Moreover, minimizing the trust in the operator of the servers (i.e., WhatsApp in this case), is an explicit design goal of the Signal protocol family [8]. It is therefore reasonable to assume that an attacker could, under certain circumstances, gain access to end-to-end encrypted (E2EE) messages.

Moreover, we require the attacker to know the phone number of the target user, as well as a WhatsApp account.

Prekey Depletion. Under these assumptions, we describe the prekey depletion attack on a user Bob, in which the PFS of initial messages sent to him is violated by constantly depleting the ephemeral (one-time) prekeys, usually automatically deposited by Bob on the WhatsApp servers. We assume Bob to be a security-cautious user with a very strict deletion policy regarding messages (i.e., disappearing messages set to the lowest value; currently 24h). Therefore, a compromise of his long- and medium-term security keys usually would not lead to a compromise of message content, if PFS guarantees hold, as the plaintext of messages would no longer be available on his device.

In our attack, the attacker (Eve) with passive access to the end-to-end encrypted messages, tries to constantly deplete the ephemeral prekeys of a user Bob, s.t. a new session with Bob initiated by another user Alice will have no PFS for the initial messages sent from Alice to Bob. Note that it is not uncommon, even for long-term communication partners, to create new sessions with each other. This is mainly due to the increasing popularity of WhatsApp Web, which usually results in more short-lived browser sessions. An active already established, smartphone app session between both communication partners will not be directly affected by this attack though.

Note that this paper focuses exclusively on point-to-point communication and does not consider group messaging scenarios.

1.1.2 Privacy and Availability Attack Model

In this case the attacker Eve is not interested in uncovering the content of Bobs conversations, but in gathering information about's Bob behavior and his devices and in denying that *any* account can communicate with Bob via WhatsApp at all. Therefore, the goals in this case are as follows:

Although the objectives of the two attack models differ, both exploit the same underlying prekey mechanisms and functionalities. As a result, the attacks are closely related and interconnected, underscoring the inherent privacy and security challenges associated with this mechanism.

(G2) **Device Status Tracking:** Monitoring the online/offline status and activity phases (active use vs. standby) of Bob's device(s).

(G3) **Fingerprinting:** Gather information about Bob's operating system (Android, iOS, macOS, Windows), their device's age and the number of (new) interactions within a given time span.

(G4) **Denial of Service:** Deny any other account to establish a new communication session via WhatApp with Bob.

Attacker Capabilities. For this attack model, the only requirements for the attacker are knowing the telephone number of the target and having a WhatsApp account.

Prekey Side-channel and Refills. To achieve their goals, the attacker inspects subtle differences in the way the victim pushes fresh prekeys to the server.

2 Background

This section should provide a high level overview of the necessary protocol aspects of the Signal protocol. For more details we refer to Appendix B.

In this paper we target to the current WhatsApp adaption of the Signal protocol(s) described in their Whitepaper [2], but since the official WhatsApp client software is not open source the exact implementation of the protocols is not easily obtainable. The origins of the Signal protocol, date back

Keys		Description
ipk^A	ik^A	Long-term identity key pair of Alice
ipk^B	ik^B	Long-term identity key pair of Bob
$prepk^A$	$prek^A$	Medium-term prekey pair of Alice, aka. *signed prekey*
$prepk^B$	$prek^B$	Medium-term prekey pair of Bob, aka. *signed prekey*
$eprepk_n^A$	$eprek_n^A$	Short-term prekey pair number n of Alice, aka. *ephemeral prekey* or *one-time prekey*
$eprepk_n^B$	$eprek_n^B$	Short-term prekey pair number n of Bob, aka. *ephemeral prekey* or *one-time prekey*
$\langle ipk^A, prepk^A, Sig(ik^A, prepk^A), [eprek_n^A] \rangle$		A prekey bundle deposited by Alice on the server
$\langle ipk^B, prepk^B, Sig(ik^B, prepk^B), [eprek_n^B] \rangle$		A prekey bundle deposited by Bob on the server

Table 1: Main cryptographic keys of the signal protocol relevant for our attacks. Public keys in asymmetric schemes always end in pk. The naming convention of the keying material is according to Cohn-Gordon et al. [9]. The ephemeral prekeys (one-time prekeys), which are considered optional in the prekey bundle, are depicted in red. This is the key type drained in our attack. The secret keys of Bob which an attacker has to compromise to benefit from the violation of forward secrecy, i.e., if no ephemeral prekeys can be used, are depicted in orange.

to the messaging App *TextSecure*, started in 2010, which introduced a *double ratchet* construction[1] where communicating parties derive new keys for sending and receiving messages. The Signal protocol actually consists of an entire family of protocols [19–22, 27] which been studied in a variety of works [3, 6, 9–12, 32].

In this paper we focus on the desired *(perfect) forward secrecy* (PFS) guarantees of the handshake protocol and the practical implications regarding privacy to ensure it. NIST defines perfect forward secrecy, or just *forward secrecy* for short, as follows:

"Forward Secrecy (FS): Assurance obtained by one party in a key-agreement transaction that the keying material derived during that transaction is secure against the future compromise of the static private key-agreement keys (if any) of the participants." [25]

The Signal protocol uses three different types of Diffi-Hellman public keys to ensure forward secrecy right from the start: Long-term identity keys, medium-term (signed) prekeys and short-term (one-time) ephemeral prekeys (see Table 1 for an overview). In our scenario Alice is the *initiator* and wants to establish a secure connection with Bob (the *responder*). The Signal protocol, as also implemented by WhatsApp, uses prekey bundles deposited by every user at a central server to allow any *initiator* to negotiate a shared secret (via Diffi-Hellman Key exchange) even if the *responder* is currently not online/available using the X3DH protocol [22] [2].

The initial handshake works as depicted in Figure 1 and starting with Formula 1. For more details we refer the reader to the Appendix B. First, the prekey bundle of the responder (in our case Bob) is fetched from the server by the initiator (in

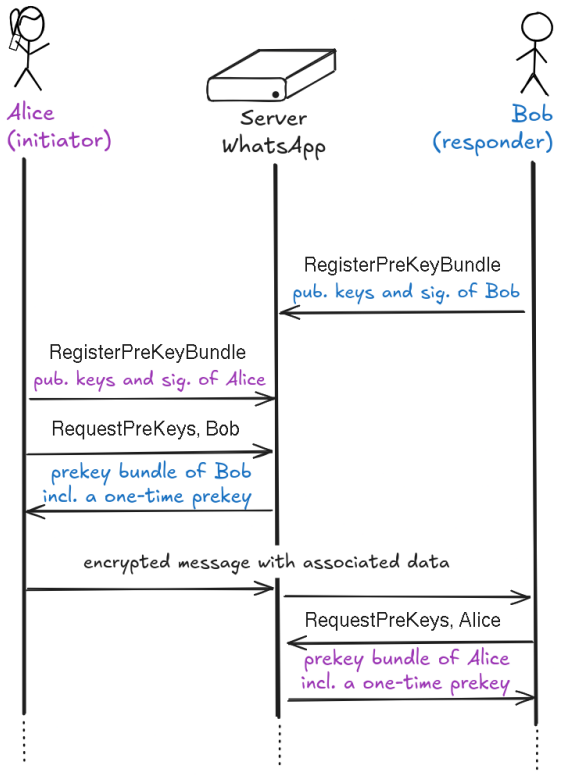


Figure 1: High-level overview of the intended prekey bundle data deposit and retrieval process, illustrating the ideal case in which everything functions as expected (i.e., prekey bundle of Bob contains also a one-time prekey).

[1] Initially referred to as *Axolotl Ratchet* to emphasize the self-healing properties of the protocol.

[2] Since late 2023 Signal replaced X3DH by PQXDH [19]. On a high level, PQXDH is comparable to X3DH with the difference that an additional key from a post quantum key encapsulation mechanism (KEM) is used.

our case Alice). The information from the prekey bundle is verified by Alice through checking the signature on the *signed prekey* using the (long-term) identity pubic key of Bob. As within other works [9], it is assumed that Alice has already verified out-of-band that the long-term identity public key indeed belongs to Bob.

Then the public keys from Bob's prekey bundle are used to compute shared keys for the ratcheting and message encryption and authentication. Here now we have to distinguish between the case where an ephemeral (one-time) prekey $eprepk^B$ of Bob is available or not. If no ephemeral prekey is available, the DH invocation dh_4 in Formula 5 is omitted. Since this is the key type drained in our attack it is colored in red.

In any case, before initiating the session and sending the first message to Bob, Alice generates ephemeral keys including a ephemeral key pair (ek^A, epk^A). Those are also used in the initial handshake and to initialize the DH ratchet construction, also referred to as the *asymmetric ratchet*.

$$dh_1 \leftarrow DH(ik^A, prepk^B) \tag{1}$$
$$dh_2 \leftarrow DH(ek^A, ipk^B) \tag{2}$$
$$dh_3 \leftarrow DH(ek^A, prepk^B) \tag{3}$$
$$[dh_4 \leftarrow DH(ek^A, eprepk^B)] \tag{4}$$
$$rk_0 \leftarrow KDF_r(dh_1 \,||\, dh_2 \,||\, dh_3[||\, DH4]) \tag{5}$$
$$\vdots$$

At the end, a new message key mk is derived using a key derivation function (KDF). This message key is then used to encrypt and authenticate (AE) the first chat *message* from Alice to Bob, as well as to authenticate some associated data AD consisting of the ephemeral public keys generated previously by Alice. This is shown in Formula 6 and 7.

$$AD \leftarrow \langle \text{ephemeral public keys of Alice, id of } prepk^B, [eprepk^B] \rangle \tag{6}$$
$$AE_{mk}, AD \leftarrow E(mk, message, AD) \tag{7}$$

Once Bob receives this initial message, he can compute the same shared keys using his identity key ik^B and his prekey $prek^B$, as well as the public keys of Alice consisting of her identity key ipk^A (Formular 8) and her ephemeral handshake public keys (Formular 9 and 10). Since the later are transmitted in the associated data AD, they are authenticated, but not encrypted. Tho benefit from the PFS violation (due to missing one-time prekys) and successfully decrypt recorded E2EE messages, the orange keys have to compromised by the attacker later on.

$$dh_1 = DH(ipk^A, prek^B) \tag{8}$$
$$dh_2 = DH(epk^A, ik^B) \tag{9}$$
$$dh_3 = DH(epk^A, prek^B) \tag{10}$$
$$rk_0 \leftarrow KDF_r(dh_1 \,||\, dh_2 \,||\, dh_3) \tag{11}$$
$$\vdots$$

The received message is then decrypted using the previously computed shared message key mk:

$$message, AD \leftarrow D(mk, AE_{mk}, AD) \tag{12}$$

If no ephemeral prekeys have been fetched by Alice, this initial message sent by Alice has no forward secrecy if observed by an attacker. Therefore, an attacker who is able to compromise Bobs medium-term and long-term secret keys $prek^B$ and ik^B later on, can recompute the same message key mk, which highlights that there is no forward secrecy for this message. If Alice sends multiple messages before receiving any response from Bob, all these messages are affected as well, as the keys for these messages come from the symmetric ratchet. Forward secrecy is restored through the asymmetric ratchet, when Bob responds to a message. As soon as these ephemeral ratchet keys are deleted, forward secrecy is regained for this as well as subsequent messages. Note, that even if forward secrecy is regained in a chat session, the initial messages sent from Alice to Bob have been encrypted using the symmetric ratchet only. Therefore, they have no forward secrecy for the entire lifetime of the signed prekey $prek^B$ of Bob. According to the specifications, the signed prekey should be periodically rotated [2, 22, 23], where suggested intervals reach from once a week to once a month [3]. WhatsApp refreshes signed prekeys usually once every month. To the best of our knowledge, the open source implementations whatsmeow, baileys and cobalt never replace the initially uploaded signed prekey, which of course would increase the impact of a loss in forward secrecy.

3 Testing Environment

To effectively test WhatsApp's session and encryption procedures, we set up a test and experimentation environment which we describe in this section. To capture and understand low-level protocol messages, we base our work on existing community projects (i.e., unofficial WhatsApp clients that are based on reverse-engineering of the official implementation).

We've used the community projects that are shown in Table 2 to dynamically send and inspect requests from our own

[3] In practice Signal rotates the signed prekey every two days https://github.com/signalapp/Signal-Android/blob/481dc162d80292a046b4229cceba2ac2f2a73f36/app/src/main/java/org/thoughtcrime/securesms/jobs/PreKeysSyncJob.kt#L57-L66

Project	GitHub Stars	Lines of Code	Project Scope
Baileys	4,856	134,052	Emulating WhatsApp Web (companion) devices
whatsmeow	2,549	67,088	Emulating WhatsApp Web (companion) devices
Cobalt	708	41,115	Emulating main (Android/iOS) and WhatsApp Web (companion) devices
CobaltAnalyzer	37	331	Capturing decrypted traffic of legitimate WhatsApp Web browser sessions

Table 2: Relevant WhatsApp community projects and their offered features that we used throughout our analysis.

WhatsApp devices (Android, iOS, Web, Desktop). Furthermore, we wrote a custom client that allows querying the existing WhatsApp devices and their cryptographic keys (*ipk*, *prepk*, *eprepk*) for an arbitrary telephone number.

3.1 Relevant Endpoints and Message Structs

To uniquely identify and address users within the messaging service, WhatsApp uses so-called JIDs (*Jabber ID*), following a specific addressing scheme: `<phoneNumber>@<serverName>`.

Every device that is registered for a specific phone number, gets their own *device ID*. The device ID is an auto-incrementing index for each user, that is reset whenever the user sets up WhatsApp on their phone. Device 0 always represents the main device (i.e., the smartphone), while non-zero device IDs are used for companion devices (i.e., web- or desktop clients). To address specific devices within a JID, the device ID is encoded between the phone number and the server name: `<phoneNumber>:<deviceId>@<serverName>`. For example, `123456789:1@s.whatsapp.net` represents the first companion device that is registered for the US-based phone number `+123456789`.

Using our custom client, we can fetch the available device IDs for an arbitrary phone number:

```
pogo@prekey: $ ./query-devices -t 123456789
Querying registered devices for target number.

Found 3 existing devices: [0, 1, 3]
```

In this case, the target has one main device (index 0) and two companion devices (index 1 and 3). Since no device with index 2 is available in this list, we can deduce that there has been a linked device (e.g., a desktop computer or WhatsApp web session) with index 2, which has been logged out (unlinked). At some later point a new device has been linked, which due to the auto-incrementing nature of the device IDs, now has index 3.

WhatsApp's encryption scheme requires the message sender to individually encrypt and send messages for every device of the recipient. Thus, the sender can query a users' current device list from the server via so-called `usync infoqueries`. WhatsApp also allows sending messages to new contacts (or unknown phone numbers), thus this endpoint is not only limited to known contacts, but can also be queried for external numbers. Using our testing client, we can retrieve (and consistently monitor) the registered devices and their corresponding device IDs for arbitrary phone numbers as already demonstrated in [15].

Besides knowing the recipient's device list, the sender also needs to retrieve each device's DH keys, to individually encrypt the message for every target device. Again, the corresponding `inforquery` can be issued to retrieve a *prekey bundle* for an arbitrary phone number. Listing 1 shows an example for the information that is returned by this query. In summary, the endpoint returns,

- the three DH public keys (*ipk*, *prepk*, *eprepk*).
- their corresponding *key IDs* (registration ID, signed prekey ID and one-time prekey ID).
- the signature of the signed prekey *prepk*.
- an epoch timestamp indicating when the device last updated the relevant object (i.e., pushed a new *prepk* or *eprepk* to the server).

The *key ID* of the signed prekeys and the *key ID* of the one-time prekeys are also incremented with every new key (of the respective type) uploaded to the server, but these IDs do not always start with zero (for more information we refer to Section 4.3).

While the device's long-term (static) identity key and the medium-term signed prekey typically remain unchanged across subsequent queries, the included one-time prekey changes with each query. As the name suggests, this prekey is intended for single use and is therefore discarded from the server after being disclosed to a third party. In practice, the endpoint is not called very often from a single account, as a device's prekey bundle only needs to be retrieved from the server when initiating a new session. For active sessions, the key material is continuously renewed and embedded within ongoing direct messages between the communicating parties. However, as we show, a malicious actor can deliberately request multiple prekey bundles from the server, effectively draining a device's one-time prekey reserve that is saved at the server.

[3] Baileys: github.com/WhiskeySockets/Baileys
whatsmeow: github.com/tulir/whatsmeow
Cobalt: github.com/Auties00/Cobalt
CobaltAnalyzer: github.com/Auties00/CobaltAnalyzer

Listing 1: WhatsApp prekey bundle containing identity key, signed prekey and a single one-time prekey, queriably for arbitrary phone numbers. The values for the byte arrays are shortened due to the limited space.

```
{
  "jid": "123456789:1@s.whatsapp.net",
  "t": "1740182155"        // epoch timestamp
  "registration": "000005DB",
  "type": "05",            // key type (djb)
  "identity": "76..77",    // 32 bytes pubkey
  "skey": {
    "id": "000001",        // signed prekey
    "value": "44...6a",    // 32 bytes pubkey
    "signature": "0d..02" // 64 bytes
  },
  "key": {
    "id": "0001a",         // one-time prekey
    "value": "0e..0b"      // 32 bytes pubkey
  }
}
```

3.2 Testing Methodology

To assess whether an attacker can deplete the saved prekeys of a specific target device and to compare official implementations and the impact across different device categories, we systematically retrieve and analyze the returned key values in various settings and scenarios.

3.2.1 Server-side Prekey Output and Rate Limits

In our custom client, we've implemented a function that queries the prekeys for a specific target device repeatedly. By targeting our own devices, we want to investigate whether there are server-side protections or rate-limiting mechanisms, stopping an attacker from doing so. Furthermore, we want to test how fast an attacker is able to consume prekeys and whether consistent prekey depletion is potentially feasible. Finally, we check whether the server properly removes the prekeys, or whether certain keys are accidentally returned more than once.

3.2.2 Client-side Prekey Input and Push Behavior

According to the protocol design, client devices push new prekeys to the server whenever necessary. We want to investigate how many prekeys are typically saved on the server and under which circumstances they're refilled.

Prekey Reserve Batches. Our analysis covers different device types (Android, iOS, Windows, macOS, Web) and different prekey states (initial reserve vs. prekey refills). To measure the amount of prekeys that were pushed in different client states, we consume (and count) all available prekeys for our target device using our custom client. To remove additional noise during the measurement and to prevent the target device from immediately refilling the prekey buffer, we put it into flight mode.

Device	Standby WiFi	Standby 4G	Screen On WiFi	Screen On 4G
iPhone SE	85%	94%	90%	93%
iPhone 8	90%	88%	89%	88%
iPhone 11	80%	96%	74%	80%
Poco X3	76%	55%	18%	17%
Galaxy A54	10%	9%	18%	4%
Redmi 10	15%	72%	13%	19%

Table 3: Success rate among different devices, i.e., how likely it is that a prekey bundle fetched for a new session will *not* contain a one-time prekey even if the target is currenlty online and powered on and an attacker is actively depleting its one-time prekeys.

Prekey Refill Trigger. After measuring typical prekey batch sizes, we repeat the above experiment without putting the target device into flight mode. A prekey refill also updates the epoch timestamp that is sent alongside the prekey bundle `infoquery` response (cf. Listing 1). Thereby, we can also observe when the last prekey refill mechanism was triggered for the target device.

Fingerprinting Device Types and Activity States. Besides identifying general conditions that trigger a prekey refill, we want to investigate whether different device types or activity states (e.g., standby) influence how fast the prekey reserve is refilled.

3.2.3 Exploration and Exploitation

After understanding the design decisions of the official WhatsApp clients and potential rate-limits on the server, we outline various exploitation scenarios. To test and verify them in practice, we use our custom client to target our own testing devices. A detailed list of the used devices can be found in Section 5 in the Appendix.

4 Results and Exploitation

Our tests showed that depleting a target's prekeys is possible and that WhatsApp currently employs little to no countermeasures against it. We furthermore show our detailed results and outline the abuse potential and specific security- and privacy implications.

4.1 Perfect Forward Secrecy Degradation

For this section, we consider the *PFS attack model* described in Section 1.1.1, so the goal (G1) of the attack is it to de-

grade PFS for new sessions initiated with Bob by other users (such as Alice). Therefore, Eve wants to deplete all one-time prekeys of Bob.

For our first attack we consider the simplest case, where the targeted device is currently offline and thus cannot refill depleted one-time prekeys at the moment. This attack is depicted in Figure 2. Our tests showed, since WhatsApp does not enforce any rate limiting, constantly querying prekey bundles for any user is possible, thereby eventually depleting all available one-time prekeys. Although one-time prekeys are crucial for ensuring PFS, messages can also be transmitted when only identity and signed prekey are available to generate a shared secret (i.e., when no one-time prekeys are available on the server). In our tests, this PFS degradation is *not* indicated to the users UI (e.g. by a security notification in the WhatsApp client initiating the session). Therefore, the PFS degradation attack can be executed without the affected users noticing in their UI. Note that even if Bob's device comes online after Alice has initiated a new communication session without a one-time prekey, messages sent by Alice will not achieve PFS until Bob's device responds for the first time within that session.

Now let's consider the case where the targeted device of Bob is constantly online. In this case, it receives a notification from the server as soon as there are less than 11 one-time prekeys left. Depending on the device type and its current power saving state, it reacts upon this notification and uploads 812 new one-time prekeys to the server. For measurements on how fast depletion of one-time prekeys is possible in case the targeted device is online, as well as the detailed behavior and availability implications, see Section 4.4 and 4.5. Table 3 provides an overview of the measured success probabilities of the one-time prekey depletion attack in case the targeted device is online. In other words, it indicates the likelihood that another party (e.g., Alice) will initiate a new communication session with Bob without using a one-time prekey while our depletion attack is in progress.

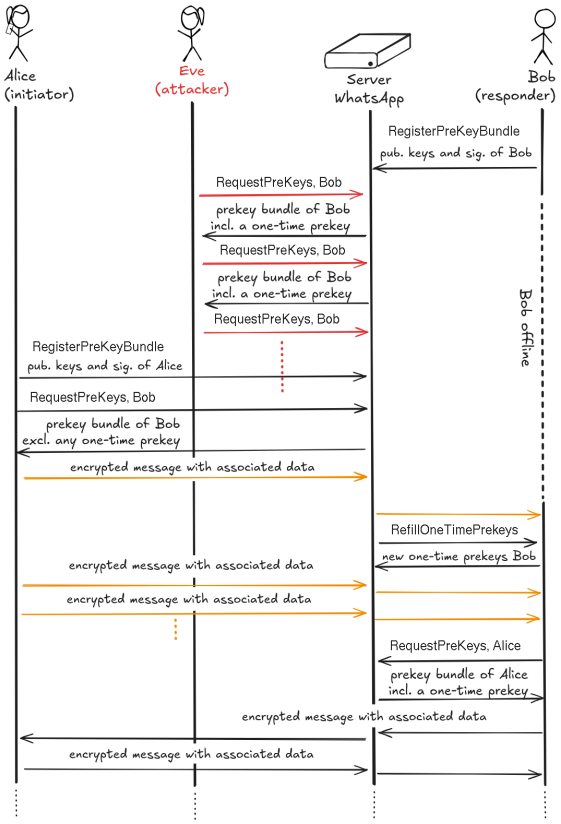


Figure 2: High-level overview of the attack. Eve is flooding the server with requests for prekey bundles of Bob. This prevents any other user, such as Alice, from obtaining a one-time prekey of Bob. Therefore, in this case all messages from Alice to Bob (orange) do not have forward secrecy, as long as the associated secret key of the signed prekey belonging to Bob is not deleted. New messages in this session after a response of Bob are not affected since new ephemeral keys are used and therefore forward secrecy is regained as soon as these new ephemeral keys are deleted.

4.2 Device Online Status Leak

For this and the remaining section we consider the *privacy and availability attack model* described in Section 1.1.2. In this section we focus on the goal (G2) *device status tracking*.

A device can only push fresh prekeys to the server, if it is turned on and connected to the Internet. When a device's prekeys are drained by an attacker and the reserve on the server drops to less than 11 prekeys, the targeted device will receive a notification from the server to refill its one-time prekeys. Consequently, depending on whether or not the one-time prekeys are refilled timely, this could also leaks the current online state of this particular device: If the one-time prekeys are not refilled timely, the targeted device is probably offline. Furthermore, any refill will update the corresponding epoch timestamp of the prekey bundle (cf. Listing 1), which

Client Implementation	Initialization Values for *key IDs*			Prekey Batch Size		Refill Trigger
	Registration	Signed PK	One-Time PK	Initial	Refill	
Android	*R*	0	*R*	812	812	10
iPhone	*R*	*R*	1	812	812	10
WhatsApp Web[a]	*R* & 0x3FFF	1	1	200	812	10
Desktop App macOS	*R*	*R*	1	200	812	10
Desktop App Windows	*R*	1	1	50	812	10

R Random number. [a] Verified on Firefox, Chrome, Safari.

Table 4: Different initialization values, prekey batch sizes, and incrementing ID patterns used across various implementations enable device fingerprinting (e.g., OS, device age). Beyond being a privacy risk, this could be exploited by an attacker during the reconnaissance phase to tailor further attacks.

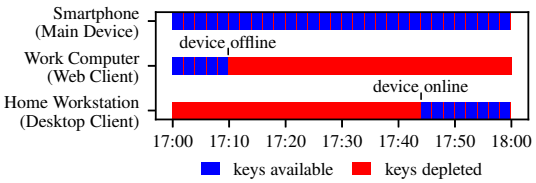


Figure 3: Each device's online status can be independently, consistently and stealthily monitored, possibly leaking the victim's location and daily routines.

thereby can be seen as a lower bound for the last online time of the respective device. Since this attack can be executed independently for every device of the victim, it can be used for stealthy and consistent tracking of connection states throughout the day. This is especially critical for companion devices that are usually not always online, such as desktop computers. For example, Eve could use this to track the victim's daily routines (and thus, corresponding locations) when switching between devices, as shown in Table 3. To match devices to a specific context or location (e.g., work vs. home), the attacker could monitor the devices over a period of multiple weeks.

Main devices are usually always online and are thus less prone to this attack. Nevertheless, omission of prekey refills for extended periods could still leak information about their current activity. For example, the attack could be used to determine whether a person is currently on a flight, in a shielded building, or other locations without any Internet connection. Delayed refills could hint to blind spots in cellular reception, e.g., when driving through a tunnel. Finally, some people use the phone's flight mode to mute all notifications during the night, disclosing a person's sleep schedule. The next Section addresses the question, how one-time prekeys are actually refilled by different devices and what information can be gathered through this feature.

4.3 Device Fingerprinting

Table 4 shows characteristic *key ID* initialization values and particular prekey batch sizes used when uploading fresh keys to the server. Different client implementations use different initialization values for assigning initial key IDs. While this also holds true regarding the initial prekey batch size, all implementations use a fixed refill batch of 812 elements. Furthermore, we did not see any difference regarding the refill trigger, as all clients push a new prekey batch (812 elements) to the server as soon as it has only 10 remaining prekeys left. In detail, this is triggered by a server-side notification that is received by the client. Pushing new one-time prekeys to the server always invalidates any previous one-time prekeys After the initial assignment, signed prekeys and one-time prekeys are issued by incrementing their IDs.

OS Disclosure. Finding out a target's operating system might be a valuable information for an attacker during the reconnaissance phase to efficiently prepare for further attacks. Due to different strategies for the used initialization values and the fact that we can naturally differentiate main and companion devices by their *device IDs* (0 for main devices, > 0 for companion devices), we can distinguish the used client implementation and runtime environment for an arbitrary device with high confidence. For users utilizing WhatsApp web, all operating systems and browsers showed the same behavior (because all browsers execute the same javascript code).

Fingerprinting a target primary device by characteristic *signed prekey IDs* (e.g., Android vs. iOS) is possible with high confidence and just requires the attacker to query the prekey bundle once for the victims phone number and device ID 0. If the signed prekey ID is high ($>> 0$), than it has probably be chosen at random and thus it must be an iPhone.

Distinguishing companion device types by their initial prekey batch sizes requires repeated querying, but is still feasible. Due to homogeneous refill batch sizes across all device types, the initial prekey batch size can be deduced at any later point in time, by consuming the entire (812 element-sized) prekey batch from the server and looking at the returned min-

imum and maximum prekey IDs.

To facilitate this kind of OS fingerprinting, our client collects all fetched prekey bundles and summarizes the stats of the returned values:

```
pogo@prekey:~$ ./deplete-keys -t 123456789
Fetching all available prekeys of target device...

All prekeys depleted, consumed 812 bundles in 45s.
[Prekey Stats] Cnt: 812, MinID: 1674, MaxID: 2486
```

In the above example, we know that the victim uses a Windows desktop application, where a batch of 50 prekeys were uploaded in the initial batch (corresponding calculation: $1,674 - 812 - 812 = 50$). If the initial batch size would be 200, we could still distinguish between macOS and WhatsApp Web by the value of the signed prekey ID as described before.

Device Age and Activity Score. The registration ID and identity key are static for the lifetime of the device installation. In contrast, signed- and one-time prekeys (and their IDs) change over time. When the device updates the signed prekey (usually done approx. once a month in WhatsApp) it also increments the corresponding signed prekey ID. Thus, for all devices except iOS and macOS, the current signed prekey ID roughly corresponds to the device's age in months.

Similarly, one-time prekey IDs are incremented by all devices. When asked by a client, the server always randomly selects one of the available one-time prekeys, making it more cumbersome to monitor for the attacker. However, when consuming all available prekeys, the attacker can still deduce how many prekeys have been used since the last refill, by calculating the difference between 812 and the number of returned prekeys. For all devices that initialize the one-time prekey with 1 (all except Android), the attacker can also derive the total amount of used one-time prekeys since the initial setup of the device. Due to the natural depletion of the one-time prekeys that is caused by new people (or devices) contacting the victim, this can be used to estimate an activity- or chattiness score of the target.

4.4 Observing Characteristic Refill Behavior

While measuring the approximate success rates for our PFS downgrade attack (Table 3 in Section 4.1), we noticed that the refill behavior can vary widely, depending on the victim's phone and its current state (e.g., current access technology and screen on/off state).

For example, across all captured experiments, iPhones were more vulnerable to prekey depletion –thus, took longer to react on a drained prekey bundle– than Android devices. Among the various Android models, the Samsung Galaxy A54 consistently showed the fastest refill times.

In addition to identifying the victim's operating system through specific key ID values, as presented in Section 4.3, continuous monitoring of refill behavior could help determine the victim's operating system or device model.

Interestingly, the Xiaomi Poco X3 also showed significantly faster reactions when the screen was active compared to when it was in standby mode. This aligns with previous work [15], exposing activity fingerprinting by measuring message RTT times via delivery receipts.

Characteristic examples for the monitored refill behavior of different phones can be found in Figure 4 in the Appendix.

4.5 Rapid Retrieval and Denial of Service

For the previously presented attacks, our client sends synchronous queries to the server. Thus, before sending another query requesting the victim's prekey bundle, we always waited for the response of the previous request, leading to a depletion rate of up to 20 prekeys per second. In practice, the time for every request is limited by the connection round trip time (RTT) between client and server. Furthermore, the process seems to be significantly influenced by the current server load, with the depletion of all 812 prekeys taking anywhere from 40 seconds to 2 minutes within our different experiments.

One-time prekeys are supposed to be returned just once, which requires synchronization across concurrent requests. To test for the maximum retrieval rate for a single session, we increased the request rate, by sending queries in an asynchronous manner. Using parallel requests, we were able to consistently deplete 812 prekeys within just 10 seconds. After a certain retrieval rate (roughly more than 50 requests per second), the server occasionally returns a `503 Service Unavailable` instead of the actual prekey bundle.

We observed that the `503` server error is not merely a rate limit affecting the current client session. Instead, generating a high volume of requests in one session also causes unsuccessful queries for other unrelated clients requesting prekeys for the same device for all querying devices.

By further increasing the request rate to $> 2{,}000$ requests per second, we can entirely clog the prekey retrieval for the corresponding victim device. We showed the feasibility of this attack by clogging prekey retrieval with one session and concurrently trying to retrieve a valid prekey bundle by an unrelated client.

Denial of Service. In Section 4.1, the attacker only tries to eliminate the one-time prekey layer from the key exchange, thus, starting a new conversation is still possible. In contrast, the failure to retrieve the entire prekey bundle will generally hinder any new conversation attempts with the victim.

We verified this in practice by trying to contact our victim from different phones while simultaneously clogging prekey bundle retrieval using our custom client implementation. In all cases, the phones were not able to effectively start a new session and send the corresponding message, but kept stuck at the sending symbol 🕓). Also, trying to audio/video call the

target via WhatsApp resulted in the call being immediately dropped[4]. To demonstrate the attack in practice, we prove a short demonstration video[5].

We tried to infer the retransmission strategy empirically. Thereby, we did not see any timing-based back-off strategies for message retransmission (i.e., even when we stopped our DoS attack, the corresponding clients were not automatically trying to resend the message). Re-entering the chat or minimizing/activating the application to/from standby does not automatically trigger a retry-procedure. However, when entirely closing and reopening the WhatsApp application, the client makes another attempt to request the prekey bundle and eventually transmits the message to the target. Our testing attempts to execute this attack over extended time periods (e.g., multiple hours) were not blocked by WhatsApp, thus performing this attack consistently seems currently feasible. Thereby, an attacker could completely prevent any new conversation attempt to the target and thus force a downgrade to use less secure messaging solutions (e.g., SMS, Telegram).

4.6 Battery Drainage

Consistently generating new prekeys and pushing them to the server results in additional battery drain and likely prevents the phone from entering deep sleep states. We measured this extra drain under conditions of rapid (i.e., asynchronous) prekey depletion, which forces the device to frequently regenerate and upload fresh prekeys. For this case, we use our Samsung Galaxy A54 5G as target, since it refilled prekey almost immediately, even when being put into standby mode (cf. Section 4.4), presumably increasing abuse potential and battery drain. Our measurements showed an additional battery drain of approximately 2% per hour (measured during standby in LTE). During this time, prekeys were depleted roughly every 15 seconds, and the process of uploading new prekeys resulted in about 8 MB of additional data usage per hour. While this attack could definitely be annoying for the victim (i.e., forcing them to recharge their phone throughout the day), we consider this only a minor availability issue. Nevertheless, similar to the previously discussed vulnerabilities, any WhatsApp user can be targeted covertly, with minimal evidence left behind on the victim's device.

4.7 Peripheral Observations

Besides the presented exploits, we made additional observations that could be relevant for WhatsApps security or privacy.

Desynchronization of Prekey Depletion State. During rapid prekey depletion, we observed instances where the prekey state between the server and client became desynchronized. As a result, the client was not aware that the prekeys had been drained and thus did not push fresh prekeys to the server. In practice, this increases the success rate for our PFS depletion attack.

Additionally, when inspecting the decrypted traffic of legitimate WhatsApp web clients (using *CobaltAnalyzer*), we observed that prekey refills of the client were occasionally rejected with a `503 Service Unavailable` error. While the client was eventually able to upload a fresh prekey bundle, this of course also enlarges the available time window for a PFS degradation attack.

Repeated Prekey Distribution. In the course of depleting prekeys from our test devices to evaluate the effectiveness of the PFS downgrade attack, we collected and analyzed the returned prekey bundles to verify whether one-time prekeys were correctly discarded after use.

While the server generally behaved as expected – successfully synchronizing concurrent queries from two independent sessions targeting the same device– we did observe rare instances in which one-time prekeys were handed out more than once. In total, we documented four such occurrences of prekey reuse, potentially indicating isolated failures in the server's prekey distribution.

Omitted Prekey IDs (Android). While adhering to the uniform batch-size of 812 elements, we noticed that for every additional prekey batch that is uploaded to the server, 2 prekey IDs are omitted by the Android client. This behavior suggests the presence of a potential off-by-two error in the Android client's prekey generation implementation. While simply skipping prekey IDs is not a security issue by itself, it could again by abused to determine the victim's operating systems as presented in Section 4.3.

Blocked Contacts. Because the attacker interacts only with the server and never sends direct packets to the victim, the victim has no means of identifying the source of the attack. Nevertheless, we evaluated whether blocking an account in WhatsApp has any impact by testing a scenario in which the victim had already blocked the attacker prior to the attack. The results show that blocking has no effect, thus the attack remains fully feasible.

5 Related Work

Security and Privacy at Instant Messaging. Schrittwieser et al. were the first to investigate security and privacy issues in mobile instant messaging and VoIP applications, uncovering vulnerabilities such as account hijacking and number spoofing [24, 29]. Beyond Over-the-Top (OTT) applications, similar security vulnerabilities have been identified in VoIP-based messaging solutions like VoLTE, VoWiFi and RCS [16, 17, 30, 33]. In many cases, these vulnerabilities re-

[4] According to a technical report of Meta, the call initiation should have been unaffected, as the required prekey bundle is sent directly via webRTC [23], as the communication partner needs to be online anyway to establish a call.

[5] https://drive.proton.me/urls/H2P7VCW9R4#6ZGjnwFjf4TT

mained undiscovered for years due to proprietary clients and the lack of tooling for security experiments. We adopt a similar security testing approach to find vulnerabilities within WhatsApp, leveraging open-source tools to emulate a real client sending protocol queries to the server. In contrast, Hagen et al. [18] demonstrated that large-scale account enumeration is possible in major messaging applications (e.g., WhatsApp, Signal) simply by automating interactions with the regular user interface. Beyond that, efficient account enumeration can also be achieved by using open-source clients to directly extract data from internal user APIs [14].

Fingerprinting and Side Channels. Previous work has shown that convenience features, such as read receipts in WhatsApp and other instant messaging apps, are frequently misused for stalking, even by non-technical users [13]. Beyond read receipts, delivery receipts expose message round-trip times (RTTs), which can be exploited to infer a user's coarse geolocation [28]. Recent work [4] has further revealed that WhatsApp's multi-device feature inadvertently leaks users' device lists and the specific device used to send a message, due to the design approach used for end-to-end encryption (E2EE). Moreover, Gegenhuber et al. have demonstrated that read receipts can be leveraged for malformed and thus invisible messages in multi-device settings, enabling independent tracking and fingerprinting of all a user's devices, as well as their online status and operating system [15]. As a potential mitigation, WhatsApp now allows users to block messages from unknown accounts[6]. However, while our prekey depletion exploit can be used for similar fingerprinting techniques –such as tracking a user's device online status– we do not send any direct messages to the victim's phone. Instead, our approach interacts solely with WhatsApp's central prekey server.

Signal Protocol. The Signal Protocol and its variants has been analyzed in a series of works [6, 7, 9–12, 32]. Cohn-Gorden et. al [9] provides a nice overview of the protocol as well as a formal security analysis of the triple Diffie-Hellman (X3DH) key agreement and the Double Ratchet (DR). The DR was initially analyzed in [3]. A new variant of the key agreement, called PQXDH, which includes PQ-KEM as been described and formally analyzed in [19]. Post-Compromise Security (PCS) was initially defined and analyzed in [10]. Attacks on PCS in a multi device setting have been described in [11, 32]. The entire conversation layer, potentially consisting of multiple sessions/devices of a user, has been analyzed in [12] also with a focus on PCS and cloned devices.

[6] https://faq.whatsapp.com/3379690015658337/

6 Discussion

6.1 Ethical Considerations

For our measurements and during experimentation we only targeted WhatsApp accounts under our direct control. Additionally, we tried to adhere to WhatsApp's protocol through the use of community-proven open source projects (some of them being used in widely deployed production systems [7]). In our depletion experiments, we issued a substantially higher volume of prekey queries compared to typical client implementations. However, this increased traffic is unlikely to pose a significant risk to WhatsApp's infrastructure, which is built to serve more than three billion users. Moreover, we limited our offensive depletion to at most two concurrent client sessions. Lastly, none of our testing accounts were blocked throughout the study, hinting that we did not cause any significant harm and most likely were not even noticed by the platform operator. Lastly, to accurately assess the feasibility of the proposed attack, it was essential to conduct experiments against the actual WhatsApp infrastructure. Given the minimal risk of adverse effects on other users or the service itself, as argued above, we considered this a reasonable approach. Finally, all our findings have been responsibly disclosed to Meta.

6.2 Limitations

Although WhatsApp is the most popular instant messenger using the Signal protocol, many other messaging applications rely on the same protocol suite. While our analysis specifically focuses on WhatsApp, some of our findings may generalize to other Signal-based messengers. Due to WhatsApp's closed-source nature, we were unable to directly inspect the source code of the official clients or the server backend. However, given its immense popularity, it is crucial to scrutinize its overall security. We hope this work serves as a first step toward shedding light on WhatsApp's real-world implementation and the design decisions underlying its deployment of the Signal protocol.

6.3 Countermeasures

Many of the exploits and side channels presented in this work are inherent to Signal's session handshake protocol, which relies on the availability of fresh one-time prekeys. As a result, completely eliminating these issues is challenging; for example, the device's online state will inevitably be exposed when new prekeys are uploaded. Nevertheless, we propose several mitigations that would substantially reduce the practical exploitability of the identified vulnerabilities.

Rate Limiting. A single account should not be able to constantly query prekey bundles for the same device in rapid

[7] https://github.com/element-hq/mautrix-whatsapp

succession. Given the fast refill rate of most Android devices, even a modest artificial slow down (i.e., rate limiting), would reduce the likelihood of a successful one-time prekey depletion attack against these devices significantly.

Reduce Signed Prekey Renewal Interval. The lifetime of a signed prekey in WhatsApp is higher ($\approx$ month), than the lifetime of a signed prekey in Signal (two days). Reducing the lifetime of signed prekeys would also reduce the impact regarding PFS through missing one-time prekeys.

Visual Indication of Missing PFS in the UI. Currently neither sender, nor receiver are notified if a new session is established without a one-time prekey, therefore there is no obvious way to detect such an attack as a user. To not flood all users with complicated warnings and to prevent misunderstandings, the settings could offer a verbose option for security-cautious or high-profile users, which would show such security related UI notifications when messages lack PFS.

Signed Prekey Update on Demand. If a prekey bundle without a one-time prekey is used to initiate a new session, the responder device could update his signed prekey together with the next batch of one-time prekeys it pushes to the server. This would minimize the damage a lack of PFS could cause, in case the responder device is online. Due to asynchronous communication, there may still exist prekey bundles in circulation that contain outdated signed prekeys, but this should not be a large problem since there is no valid use case for keeping them around and not immediately initiate a new session. To prevent an attacker from turning this countermeasure into a DoS attack, there should be a minimum validity period (in the order of minutes) for signed prekeys. Otherwise an attacker could trigger signed prekey updates all the time by initiating new sessions without one-time prekeys, which would prevent everybody else from establishing a new session as signed prekeys are constantly outdated.

Redesign Key IDs. Due to their initial values and the fact that they are incremented by one, key IDs leak information and make device fingerprinting possible. The question is, if key IDs could not be enlarged and entirely be replaced with hashes of the respective public keys they refer to, which would completely mitigate this information disclosure vulnerability.

7 Conclusion

In this work we have demonstrated that WhatsApp does not enforce any rate limiting regarding the querying of prekey bundles, thereby violating the Signal X3DH specification. This enables an attacker to deplete all one-time prekeys of a targeted device, subsequently degrading the perfect forward secrecy (PFS) of new sessions initiated with the victim. Although, PFS is undoubtedly effected by such an attack, the successful exploitation of this degraded forward secrecy would still require a compromise of the involved long- and medium-term secret keys, as well as passive eavesdropping capabilities to record the respective encrypted messages.

In contrast to this rather strong attacker model, we also describe attacks on privacy and availability, with the sole requirement of having a WhatsApp account. Hereby, we were able to show that the refilling of one-time prekeys necessarily leaks the current online status of the respective device, as well as in certain cases: the device age, operating system and the approximate total number of new sessions initiated with the targeted device. Moreover, we where able to highlight a DoS issue by rapidly querying prekey bundles of a device such that the retrieval of any prekey bundle (even without one-time prekeys) was no longer possible. As a consequence, for the duration of the attack no new session can be established with the victim. All attacks described in this paper can be executed covertly, and targeted at any of WhatsApp's more than 3 billion users.

To mitigate the discovered issues, we suggest a range of countermeasures. Most notably the notification of users regarding the degraded PFS in the UI, as well as the introduction of rate limits regarding the repeated fetching of prekey bundles for the same device from a single account.

Artifact Evaluation

The artifact accompanying this paper is publicly available at https://github.com/sbaresearch/prekey-pogo/tree/woot25ae. It includes a modified WhatsApp client based on *whatsmeow*, which enables the retrieval of a user's WhatsApp devices and their associated prekey material. As part of the WOOT 2025 artifact evaluation process, we applied for and were awarded both the *Artifacts Available* and *Artifacts Functional* badges.

Acknowledgments

This material is based upon work partially supported by (1) the Christian-Doppler-Laboratory for Security and Quality Improvement in the Production System Lifecycle; The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the Nation Foundation for Research, Technology and Development and University of Vienna, Faculty of Computer Science, Security & Privacy Group is gratefully acknowledged; (2) the FFG Bridge project 46322124 SecKey; (3) SBA Research (SBA-K1 NGC), a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme, funded by BMIMI, BMWET, and the federal state of Vienna. The COMET Programme is managed by FFG.

We further would like to thank Markus Maier for his outstanding support in providing and maintaining the IT infrastructure, which was essential for the success of this work. Moreover, we would also like to thank our anonymous reviewers for their valuable feedback and suggestions.

References

[1] CRYSTALS cryptographic suite for algebraic lattices. Retrieved Aug 26th, 2024 from https://pq-crystals.org/index.shtml.

[2] WhatsApp encryption overview: Technical white paper. Retrieved Aug 26th, 2024 from https://faq.whatsapp.com/82012443585354.

[3] Joël Alwen, Sandro Coretti, and Yevgeniy Dodis. The double ratchet: Security notions, proofs, and modularization for the signal protocol. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 129–158. Springer.

[4] Tal A. Be'ery. WhatsApp with privacy? privacy issues with IM e2ee in the multi-device setting. In *18th USENIX WOOT Conference on Offensive Technologies (WOOT 24)*, pages 11–16. USENIX Association.

[5] Daniel J. Bernstein. Curve25519: New diffie-hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006*, pages 207–228, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[6] Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila. Post-quantum asynchronous deniable key exchange and the signal handshake. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part II*, volume 13178 of *Lecture Notes in Computer Science*, pages 3–34. Springer.

[7] Melissa Chase, Trevor Perrin, and Greg Zaverucha. The signal private group system and anonymous credentials supporting efficient verifiable encryption. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1445–1459. ACM.

[8] Melissa Chase, Trevor Perrin, and Greg Zaverucha. The Signal Private Group System and Anonymous Credentials Supporting Efficient Verifiable Encryption. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1445–1459. ACM, 2020.

[9] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A formal security analysis of the signal messaging protocol. 33(4):1914–1983.

[10] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. On post-compromise security. In *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*, pages 164–178. IEEE Computer Society.

[11] Cas Cremers, Jaiden Fairoze, Benjamin Kiesl, and Aurora Naska. Clone Detection in Secure Messaging: Improving Post-Compromise Security in Practice. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1481–1495. ACM, 2020.

[12] Cas Cremers, Charlie Jacomme, and Aurora Naska. Formal Analysis of Session-Handling in Secure Messaging: Lifting Security from Sessions to Conversations. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 1235–1252. USENIX Association.

[13] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. "A Stalker's Paradise" How Intimate Partner Abusers Exploit Technology. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018.

[14] Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Johanna Ullrich, and Aljosha Judmayer. Hey there! You are using WhatsApp: Enumerating Three Billion Accounts for Security and Privacy, 2025.

[15] Gabriel K. Gegenhuber, Maximilian Günther, Markus Maier, Aljosha Judmayer, Florian Holzbauer, Philipp É. Frenzel, and Johanna Ullrich. Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers, 2024.

[16] Gabriel K. Gegenhuber, Florian Holzbauer, Philipp É. Frenzel, Edgar Weippl, and Adrian Dabrowski. Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 451–468, Philadelphia, PA, August 2024. USENIX Association.

[17] Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, and Adrian Dabrowski. MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research. In *Usenix Security Symposium 2023*, 2023.

[18] Christoph Hagen, Christian Weinert, Christoph Sendner, Alexandra Dmitrienko, and Thomas Schneider. All the numbers are US: Large-scale abuse of contact discovery in mobile messenger. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, San Diego, California, USA, February 21 - February 25, 2021*. The Internet Society.

[19] Ehren Kret and Rolfe Schmidt. The PQXDH key agreement protocol. Retrieved Aug 26th, 2024 from https://signal.org/docs/specifications/pqxdh/pqxdh.pdf.

[20] Moxie Marlinspike. Private group messaging. Retrieved Aug 26th, 2024 from https://signal.org/blog/private-groups/.

[21] Moxie Marlinspike and Trevor Perrin. The sesame algorithm: Session management for asynchronous message encryption. Retrieved Aug 26th, 2024 from https://signal.org/docs/specifications/sesame/sesame.pdf.

[22] Moxie Marlinspike and Trevor Perrin. The x3dh key agreement protocol. Retrieved Aug 26th, 2024 from https://signal.org/docs/specifications/x3dh/x3dh.pdf.

[23] Meta. Messenger end-to-end encryption overview, December 2023. Accessed: 2025-03-11.

[24] Robin Mueller, Sebastian Schrittwieser, Peter Fruehwirt, Peter Kieseberg, and Edgar Weippl. What's new with WhatsApp & Co.? Revisiting the Security of Smartphone Messaging Applications. iiWAS '14, page 142–151, New York, NY, USA, 2014. Association for Computing Machinery.

[25] National Institute of Standards and Technology. Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography (Revision 3). Technical Report NIST SP 800-56A Rev. 3, National Institute of Standards and Technology, April 2018.

[26] Trevor Perrin. The Noise Protocol Framework. Online, 2018. Version 34, Retrieved March 5, 2025.

[27] Trevor Perrin and Moxie Marlinspike. The double ratchet algorithm. Retrieved Aug 26th, 2024 from https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf.

[28] Theodor Schnitzler, Katharina Kohls, Evangelos Bitsikas, and Christina Pöpper. Hope of delivery: Extracting user locations from mobile instant messengers. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society.

[29] Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar Weipp. Guess who is texting you? Evaluating the security of smartphone messaging applications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5 - February 8, 2012*. The Internet Society.

[30] Guan-Hua Tu, Chi-Yu Li, Chunyi Peng, Yuanjie Li, and Songwu Lu. New security threats caused by IMS-based SMS service in 4G LTE networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[31] WhatsApp. About WhatsApp, 2024. Retrieved Aug 26th, 2024 from https://www.whatsapp.com/about/.

[32] Jan Wichelmann, Sebastian Berndt, Claudius Pott, and Thomas Eisenbarth. Help, My Signal has Bad Device! - Breaking the Signal Messenger's Post-Compromise Security Through a Malicious Device. In Leyla Bilge, Lorenzo Cavallaro, Giancarlo Pellegrino, and Nuno Neves, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment - 18th International Conference, DIMVA 2021, Virtual Event, July 14-16, 2021, Proceedings*, volume 12756 of *Lecture Notes in Computer Science*, pages 88–105. Springer.

[33] Yaru Yang, Yiming Zhang, Tao Wan, Chuhan Wang, Haixin Duan, Jianjun Chen, and Yishen Li. Uncovering Security Vulnerabilities in Real-world Implementation and Deployment of 5G Messaging Services. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2024.

Appendix

A Tested Phone Models and OS Versions

Device	Modem Chipset	OS	WhatsApp
iPhone SE 2020	Intel	iOS 18.3.1	2.25.4.77
iPhone 8	Intel	iOS 16.7.10	2.25.5.74
iPhone 11	Qualcomm	iOS 18.3.1	2.25.5.74
Xiaomi POCO X3 NFC	Qualcomm	Android 12 (MIUI 14.0.5)	2.25.2.78
Samsung Galaxy A54 5G	Exynos	Android 14	2.25.2.78
Xiaomi Redmi 10 5G	MediaTek	Android 14	2.25.2.78

Table 5: Overview of the devices including software versions that were used throughout our tests. For our WhatsApp Web tests (Chrome, Firefox, Safari) we've used the most recent browser and WhatsApp web versions available (testing date 2025-02-21).

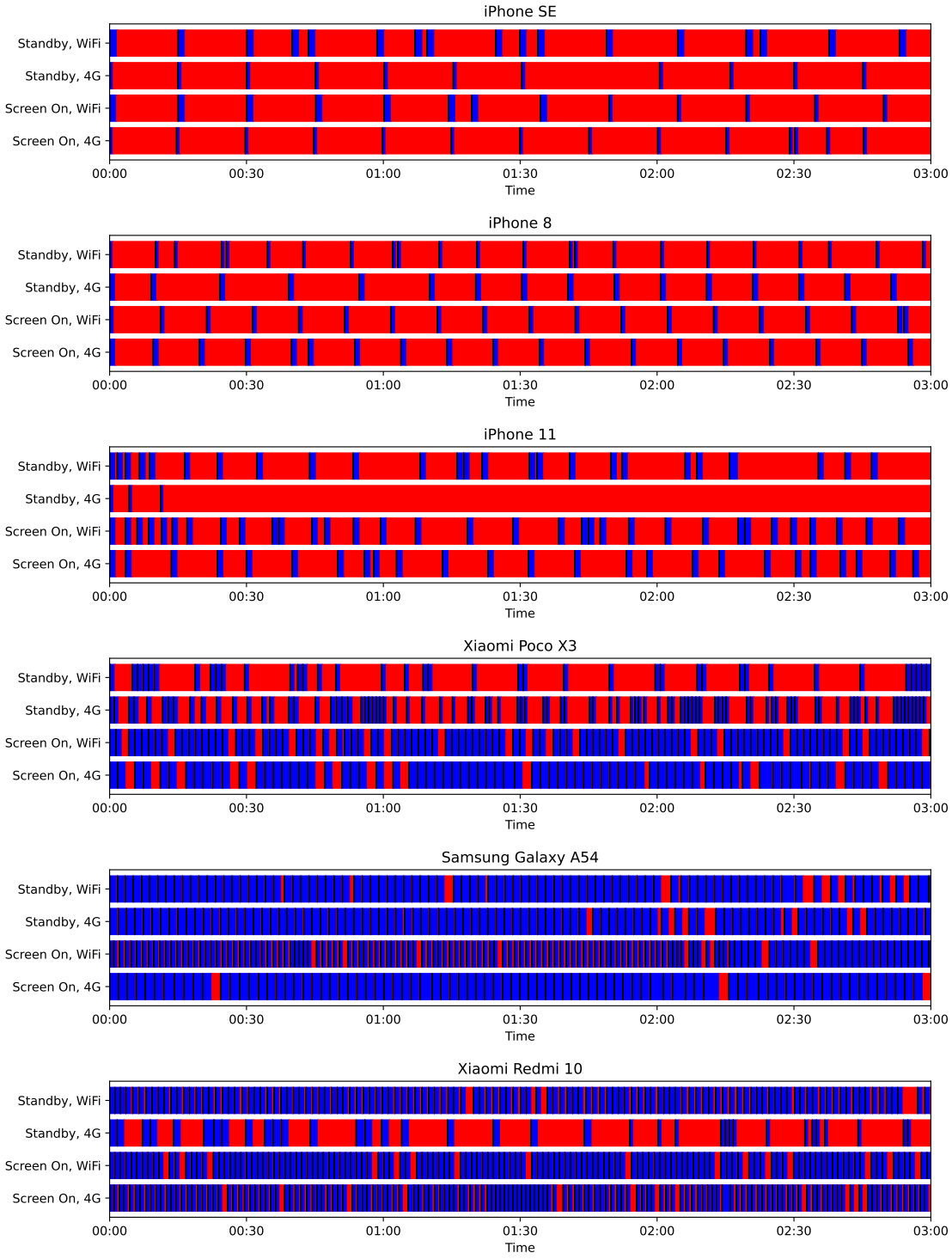


Figure 4: Characteristic refill behavior of different smartphone models in various device and connection states. Time where no one-time prekeys are available is shown in red.

B Signal Protocol in More Detail

The Signal protocol actually consists of an entire family of protocols [19–22, 27] which been studied in a variety of works [3, 6, 9–12, 32].

Internally, the signal protocol uses two key derivation functions (KDF), one to derive ratchet keys, which we denote by $KDF_r(\cdot)$ and one to derive the message keys, denoted by $KDF_m(\cdot)$. The KDFs are implemented with HKDF and HMAC-SHA256. The thereby generated derived keys are used for authenticated encryption with associated data (AEAD), using AES256 in CBC mode for encryption and HMAC-SHA256 for authentication. We denote the symmetric encryption function by $E(\text{key}, \text{plaintext}, \text{associated data})$ and the decryption function by $D(\text{key}, \text{ciphertext}, \text{associated data})$. Moreover, the signal protocol relies on Diffie-Hellman key exchange to compute shared keys and achieve its design goals. We denote the key exchange algorithm by $DH(\cdot)$, which is implemented over Curve25519 [5] in practice

The Signal protocol uses three different types of Diffi-Hellman public keys to ensure forward secrecy right from the start: Long-term identity keys, medium-term (signed) prekeys and short-term (one-time) ephemeral prekeys (see Table 6 for an overview). In our scenario Alice is the *initiator* and wants to establish a secure connection with Bob (the *responder*). The Signal protocol, as also implemented by WhatsApp, uses prekey bundles deposited by every user at a central server to allow any *initiator* to negotiate a shared secret (via Diffi-Hellman Key exchange) even if the *responder* is currently not online/available using the X3DH protocol [22] [8].

The initial handshake works as depicted in figure 5 in the appendix and here starting with formula 13. First the prekey bundle of the responder (in our case Bob) is fetched from the server by the initiator (in our case Alice). The information from the prekey bundle is verified by Alice through checking the signature on the *signed prekey* using the (long-term) identity pubic key of Bob. As within other works [9], it is assumed that Alice has already verified out-of-band that the long-term identity public key indeed belongs to Bob.

Then the public keys from Bob's prekey bundle are used to compute shared keys for the ratcheting and message encryption and authentication. Here now we have to distinguish between the case where an ephemeral (one-time) prekey $eprepk^B$ of Bob is available or not. If no ephemeral prekey is available, the DH invocation dh_4 in formula 17 is omitted.

In any case, before initiating the session and sending the first message to Bob, Alice generates two ephemeral key pairs: The *ephemeral handshake key pair* denoted (epk^A, ek^A) and the *ephemeral ratchet key pair* denoted $(rchpk^A, rchk^A)$.

[8] Since late 2023 Signal replaced X3DH by PQXDH [19] and started a process to use PQXDH for new sessions if supported by both peers. On a high level, PQ3DH is comparable to X3DH with the difference that an additional key from a CRYSTALS-Kyber [1] key encapsulation mechanism (KEM) is used in the KDF.

Those are used for the initial handshake and to initialize the DH ratchet construction, also referred to as the *asymmetric ratchet*.

$$dh_1 \leftarrow DH(ik^A, prepk^B) \tag{13}$$
$$dh_2 \leftarrow DH(ek^A, ipk^B) \tag{14}$$
$$dh_3 \leftarrow DH(ek^A, prepk^B) \tag{15}$$
$$[dh_4 \leftarrow DH(ek^A, eprepk^B)] \tag{16}$$
$$rk_0 \leftarrow KDF_r(dh_1 \,||\, dh_2 \,||\, dh_3[||\, DH4]) \tag{17}$$
$$DH_{ratchet} \leftarrow DH(rchk_0^A, prepk^B) \tag{18}$$
$$rk_1, ck_{0,0}^{i\rightarrow r} \leftarrow KDF_r(rk_0, DH_{ratchet}) \tag{19}$$
$$ck_{0,1}^{i\rightarrow r}, mk_{0,0}^{i\rightarrow r} \leftarrow KDF_m(ck_{0,0}^{i\rightarrow r}) \tag{20}$$

The derived message key $mk_{0,0}^{i\rightarrow r}$ is then used to encrypt and authenticate (AE) the first chat *message* from Alice to Bob, as well as to authenticate some associated data AD consisting of the ephemeral public keys (epk^A and $rchpk_0^A$) generated previously by Alice.

$$AD \leftarrow \langle rchpk_0^A, epk^A, \text{ id of } prepk^B, [eprepk^B] \rangle \tag{21}$$
$$AE_{mk_{0,0}^{i\rightarrow r}}, AD \leftarrow E(mk_{0,0}^{i\rightarrow r}, message, AD) \tag{22}$$

Once Bob receives this initial message, he can compute the same shared keys using his identity key ik^B and his prekey $prek^B$, as well as the public keys of Alice consisting of her identity key ipk^A, her ephemeral handshake key epk^A and her ephemeral ratchet key $rchpk^A$. Since the later two are transmitted in the associated data AD, they are authenticated, but not encrypted.

$$dh_1 = DH(ipk^A, prek^B) \tag{23}$$
$$dh_2 = DH(epk^A, ik^B) \tag{24}$$
$$dh_3 = DH(epk^A, prek^B) \tag{25}$$
$$rk_0 \leftarrow KDF_r(dh_1 \,||\, dh_2 \,||\, dh_3) \tag{26}$$
$$DH_{ratchet} \leftarrow DH(rchpk_0^A, prek^B) \tag{27}$$
$$rk_1, ck_{0,0}^{i\rightarrow r} \leftarrow KDF_r(rk_0, DH_{ratchet}) \tag{28}$$
$$ck_{0,1}^{i\rightarrow r}, mk_{0,0}^{i\rightarrow r} \leftarrow KDF_m(ck_{0,0}^{i\rightarrow r}) \tag{29}$$

The received message is then decrypted using the previously computed shared message key $mk_{0,0}^{i\rightarrow r}$:

$$message, AD \leftarrow D(mk_{0,0}^{i\rightarrow r}, AE_{mk_{0,0}^{i\rightarrow r}}, AD) \tag{30}$$

If no ephemeral prekeys have been fetched by Alice, this initial message sent by Alice has no forward secrecy if observed by an attacker. Therefore, an attacker who is able to

Keys		Description
ipk^A	ik^A	Long-term identity key pair of Alice
ipk^B	ik^B	Long-term identity key pair of Bob
$prepk^A$	$prek^A$	Medium-term prekey pair of Alice, aka. *signed prekey*
$prepk^B$	$prek^B$	Medium-term prekey pair of Bob, aka. *signed prekey*
$eprepk_n^A$	$eprek_n^A$	Short-term prekey pair number n of Alice, aka. *ephemeral prekey* or *one-time prekey*
$eprepk_n^B$	$eprek_n^B$	Short-term prekey pair number n of Bob, aka. *ephemeral prekey* or *one-time prekey*
$\langle ipk^A, prepk^A, Sig(ik^A, prepk^A), [eprek_n^A]\rangle$		A prekey bundle deposited by Alice on the server
$\langle ipk^B, prepk^B, Sig(ik^B, prepk^B), [eprek_n^B]\rangle$		A prekey bundle deposited by Bob on the server
epk^A	ek^A	Ephemeral handshake key pair of Alice
$rchpk_0^A$	$rchk_0^A$	Ephemeral ratchet key pair of Alice
rk_x		Symmetric (shared) root key of ratchet number x
$ck_{x,y}^{i\to r}$		Symmetric (shared) chaining key number y, in the x^{th} initiator to responder ratchet
$mk_{x,y}^{i\to r}$		Symmetric (shared) message key number y, in the x^{th} initiator to responder ratchet

Table 6: Main cryptographic keys of the signal protocol relevant for our attacks. Public keys in asymmetric schemes always end in pk. The naming convention of the keying material is according to Cohn-Gordon et al. [9]. The ephemeral prekeys, which are considered optional in the prekey bundle, are depicted in red. The secret keys of Bob which an attacker has to compromise to benefit from the violation of forward secrecy, i.e., if no ephemeral prekeys can be used, are depicted in orange.

compromise Bobs medium-term and long-term secret keys $prek^B$ and ik^B later on, can recompute the same message key $mk_{0,0}^{i\to r}$, which highlights that there is no forward secrecy for this message. If Alice sends multiple messages before receiving any response from Bob, all these messages are affected as well, as the keys for these messages come from the symmetric ratchet. This is illustrated by the following example starting with formula 31, which depicts encrypting a second message from Alice to Bob. Here, y is 0 at the beginning and later set to $y = 1$ for the second message and so forth:

$$ck_{0,2}^{i\to r}, mk_{0,1}^{i\to r} \leftarrow KDF_m(ck_{0,1}^{i\to r}) \tag{31}$$
$$y \leftarrow y + 1 \tag{32}$$
$$AD \leftarrow (rchpk_0^A, ipk^A, ipk^B, y) \tag{33}$$
$$AE_{mk_{0,1}^{i\to r}}, AD \leftarrow AE(mk_{0,1}^{i\to r}, message, AD) \tag{34}$$

Forward secrecy is restored through the asymmetric ratchet, when Bob responds to a message. If Bob responds to one of Alice messages, he also computes a new ephemeral ratchet key pair $(rchpk_1^B, rchk_1^B)$, s.t. $x = 1$, and thereby advances the asymmetric ratchet as follows:

$$DH_{ratchet} \leftarrow DH(rchpk_{x-1}^A, rchk_x^B) \tag{35}$$
$$tmp, ck_{x,0}^{r\to i} \leftarrow KDF_r(rk_x, DH_{ratchet}) \tag{36}$$
$$ck_{x,1}^{r\to i}, mk_{x,0}^{r\to i} \leftarrow KDF_m(ck_{x,0}^{r\to i}) \tag{37}$$
$$AD \leftarrow (rchpk_x^B) \tag{38}$$
$$AE_{mk_{x,y}^{r\to i}} \leftarrow E(mk_{x,y}^{r\to i}, message, AD) \tag{39}$$

As soon as these ephemeral ratchet keys are deleted, forward secrecy is regained for this as well as subsequent messages. Note, that even if forward secrecy is regained in a chat session, the initial messages sent from Alice to Bob have been encrypted using the symmetric ratchet only. Therefore, they remain vulnerable for the entire lifetime of the signed prekey $prek^B$ of Bob. According to the specifications, the signed prekey should be periodically rotated [2,22,23], where suggested intervals reach from once a week to once a month [9].

[9] In practice Signal rotates the signed prekey every two days https://github.com/signalapp/Signal-Android/blob/481dc162d80292a046b4229cceba2ac2f2a73f36/app/src/main/java/org/thoughtcrime/securesms/jobs/PreKeysSyncJob.kt#L57-L66

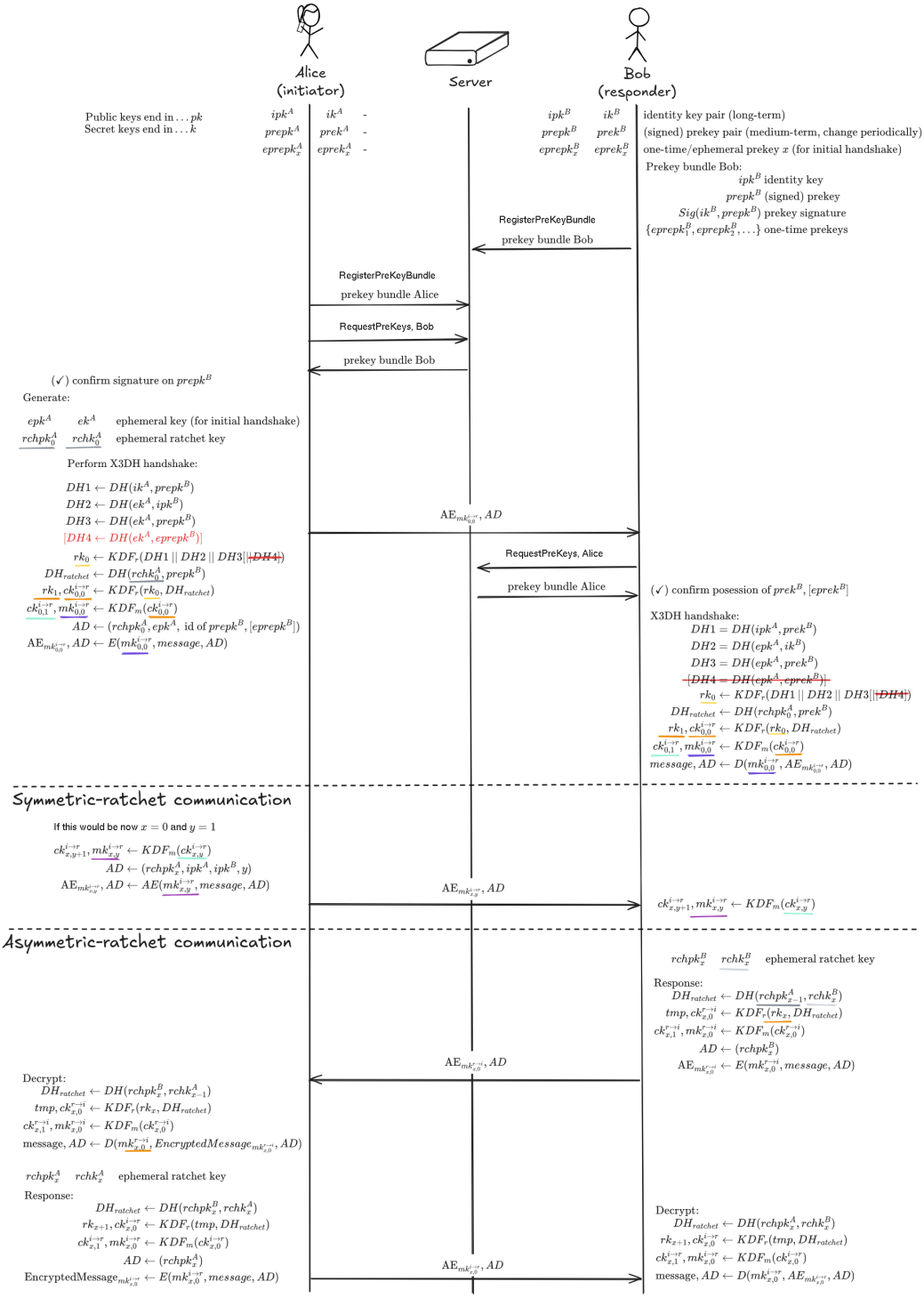


Figure 5: Signal protocol layout, when no *ephemeral (one-time) prekeys* are available on the server. Identical keys a highlighted with the same color.

9 Conclusion

Cellular networks form a resilient global communication infrastructure due to their inherently distributed architecture. Radio access networks and VoWiFi services are operated by independent entities across geographic and administrative boundaries, reducing the impact of failures to a limited region and user base.

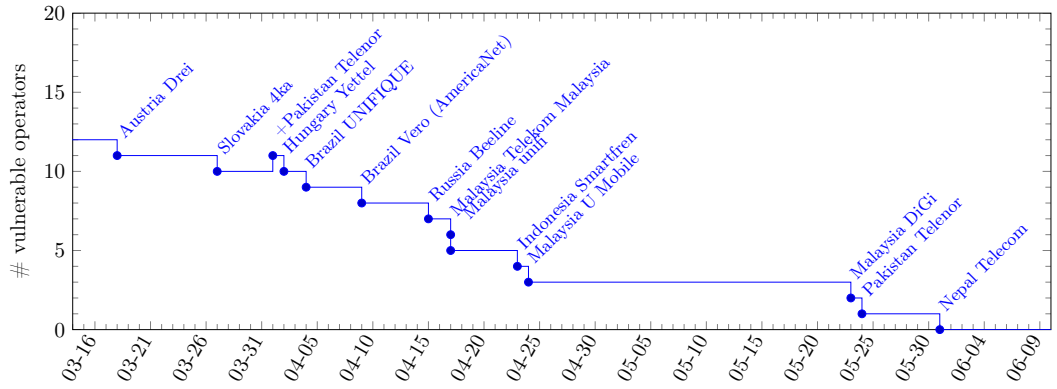
At the same time, this dissertation shows that many components of the mobile communication ecosystem have become increasingly vulnerable and centralized. Internet-accessible operator gateways expand the attack surface and are frequently operated with outdated or insecure configurations. Moreover, globally dominant OTT instant messaging platforms consolidate functionality that was previously distributed, introducing systemic risks by concentrating failure domains and attack surfaces at a massive scale. This risk is not only theoretical: recent work reports that roughly 3.5 billion WhatsApp accounts could be enumerated, illustrating how a single service can create ecosystem-wide exposure [GFG+26].

Across all settings, this dissertation demonstrates that active, independent measurements are essential to uncover and address such risks. By introducing MobileAtlas and leveraging Internet-based measurement vectors, this dissertation enables controlled and reproducible studies across cellular access networks, operator-managed services, and third-party communication platforms. The presented measurements show that while decentralization enhances resilience, centralization simplifies observation, control, and, consequently, exploitation.

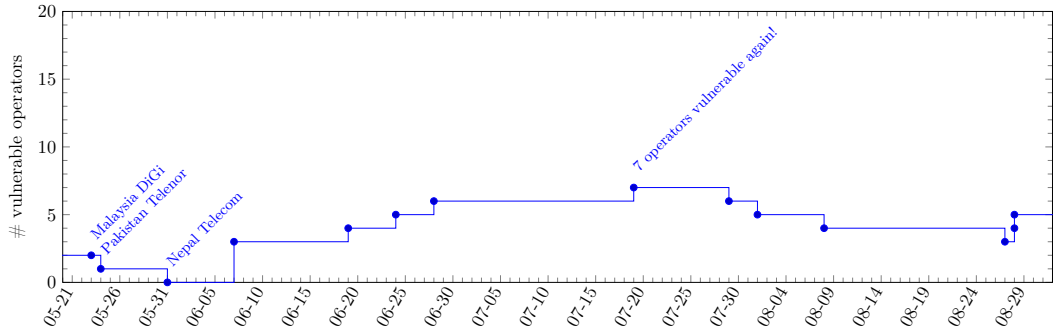
Beyond individual vulnerabilities, this work challenges the assumption that standardization alone provides security and privacy guarantees in mobile communication systems. The presented measurements demonstrate that without independent verification, such assumptions can mask systemic weaknesses across operators and platforms.

An interesting trade-off that was observed throughout this work concerns mitigation and patch deployment. Centralized services set clear responsibilities, allow for rapid patching and coordinated security updates, often mitigating vulnerabilities within several days. In contrast, decentralized cellular infrastructures require patches and configuration changes to be rolled out across many independently operated networks, resulting in slower and more uneven adoption, as we observed for *CVE-2024-22064* (shown in Figure 9.1). While this longer rollout slope can delay mitigation, it also limits the blast radius of individual failures.

Future directions. This dissertation does not aim to provide an exhaustive survey of mobile standards or deployments. Consequently, future measurement work should also cover the protocol and service alternatives that increasingly carry everyday communication: Apple-controlled iMessage has become a de facto default messaging channel on iOS; Rich Communication Services (RCS), originally developed as an open standard but nowadays heavily influenced (and, in practice, often controlled) by Google (e.g., because many operators rely on Google-hosted backends via Jibe); and



(a) **Initial remediation timeline**. After coordinated disclosure via the GSMA, vulnerable operators gradually replaced the reused private-keys used for the IKE/IPSec handshake over a period of several months, eventually eliminating the issue across all observed deployments.



(b) **Subsequent regression timeline**. Follow-up ePDG scans of global operators revealed a reappearance of reused private-keys during the IKE/IPsec handshake shortly after full remediation, affecting both previously remediated operators and newly observed vulnerable deployments.

Figure 9.1: Global reuse of static private keys in ePDG deployments: remediation and regression over time. Without continuous, worldwide scans of operator infrastructures, neither the original vulnerability nor its subsequent reoccurrence would have been discovered.

the email ecosystem, which has consolidated around a few large providers (e.g., Gmail, Outlook) while smaller providers continue to diminish and often lag in deploying available security mechanisms (e.g., DNSSEC) [HULF22]. Beyond technical risk, the concentration of these backends outside the EU raises digital sovereignty concerns by creating dependencies on external providers and jurisdictions. Extending the dissertation's methodology to map backend centralization, configuration hygiene, and metadata leakage across these systems, and to track mitigation rollouts longitudinally, would strengthen our ability to detect emerging systemic risks early.

In summary, this dissertation highlights a fundamental tension in modern mobile communication systems: decentralization strengthens resilience, while centralization

amplifies both risk and control. Independent, scalable measurement capabilities are indispensable to understanding this balance, detecting emerging systemic vulnerabilities, and supporting informed decisions that improve the security, privacy, and robustness of global communication infrastructures.

Bibliography

[AWR14] Collin Anderson, Philipp Winter, and Roya. Global Network Interference Detection Over the RIPE Atlas Network. In *Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[blu03] Bluetooth SIM Access Profile Specification. Technical report, 3GPP, 2003.

[Geg25] Gabriel K. Gegenhuber. Security and Privacy Measurements in Cellular Networks: Novel Approaches in a Global Roaming Context. In *32nd ACM Conference on Computer and Communications Security (CCS)*, 2025.

[GF25] Gabriel K. Gegenhuber and Philipp É. Frenzel. Scanywhere: Distributed Internet Scanning Leveraging Commercial VPN Subscriptions. In *9th Network Traffic Measurement and Analysis Conference (TMA)*, 2025.

[GFD25] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Adrian Dabrowski. SIMulator: SIM Tracing on a (Pico-)Budget. In *18th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2025.

[GFG+26] Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, Johanna Ullrich, and Aljosha Judmayer. Hey there! You are using WhatsApp: Enumerating Three Billion Accounts for Security and Privacy. In *33rd Annual Network and Distributed System Security Symposium (NDSS)*, 2026.

[GFGJ25] Gabriel K. Gegenhuber, Philipp É. Frenzel, Maximilian Günther, and Aljosha Judmayer. Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp's Handshake Mechanism. In *19th USENIX WOOT Conference on Offensive Technologies (WOOT)*, 2025.

[GFW24a] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. Never Gonna Give You Up: Exploring Deprecated NULL Ciphers in Commercial VoWiFi Deployments. In *17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2024.

[GFW24b] Gabriel K. Gegenhuber, Philipp É. Frenzel, and Edgar Weippl. Why E.T. Can't Phone Home: A Global View on IP-based Geoblocking at VoWiFi. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2024)*, 2024.

[GGM+25] Gabriel K. Gegenhuber, Maximilian Günther, Markus Maier, Aljosha Judmayer, Florian Holzbauer, Philipp É. Frenzel, and Johanna Ullrich.

Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers. In *28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2025.

[GHF+24] Gabriel K. Gegenhuber, Florian Holzbauer, Philipp É. Frenzel, Edgar Weippl, and Adrian Dabrowski. Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments. In *33rd USENIX Security Symposium (USENIX Security)*, 2024.

[GLHS25] Gabriel K. Gegenhuber, Leonid Liadveikin, Florian Holzbauer, and Sebastian Strobl. A Relay a Day Keeps the AirTag Away: Practical Relay Attacks on Apple's AirTags. In *41st Annual Computer Security Applications Conference (ACSAC)*, 2025.

[GMH+23] Gabriel K. Gegenhuber, Markus Maier, Florian Holzbauer, Wilfried Mayer, Georg Merzdovnik, Edgar Weippl, and Johanna Ullrich. An extended view on measuring tor as-level adversaries. *Computers & Security*, 132:103302, 2023.

[GMW22] Gabriel K. Gegenhuber, Wilfried Mayer, and Edgar Weippl. Zero-Rating, One Big Mess: Analyzing Differential Pricing Practices of European MNOs. In *IEEE Global Communications Conference (GLOBECOM)*, 2022.

[GMWD23] Gabriel K. Gegenhuber, Wilfried Mayer, Edgar Weippl, and Adrian Dabrowski. MobileAtlas: Geographically Decoupled Measurements in Cellular Networks for Security and Privacy Research. In *32nd USENIX Security Symposium (USENIX Security)*, 2023.

[HULF22] Florian Holzbauer, Johanna Ullrich, Martina Lindorfer, and Tobias Fiebig. Not that Simple: Email Delivery in the 21st Century. In *USENIX Annual Technical Conference (USENIX ATC)*, 2022.

[ISO06] ISO/IEC 7816-3:2006 - Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols. Standard, International Organization for Standardization, Geneva, CH, November 2006.

[RIP15] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal*, 18(3), 2015.

[SB24] Matthew Shanahan and Kalvin Bahia. The State of Mobile Internet Connectivity 2024. Technical report, GSM Association, 2024.