usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments

Gabriel K. Gegenhuber and Florian Holzbauer, *University of Vienna;* Philipp É. Frenzel, *SBA Research;* Edgar Weippl, *University of Vienna and Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle* (*CDL-SQI*)*;* Adrian Dabrowski, *CISPA Helmholtz Center for Information Security*

https://www.usenix.org/conference/usenixsecurity24/presentation/gegenhuber

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

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.

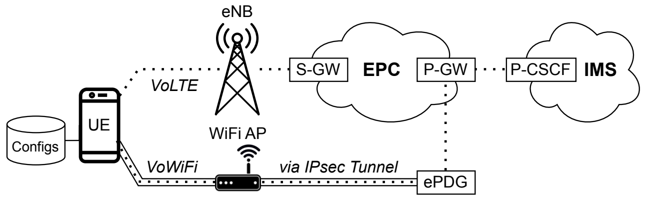


Figure 1: VoLTE compared to VoWiFi over an untrusted Internet connection – as relevant for this paper

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$^{768\text{ bits}}$ 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 support 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 over-provisioning 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.

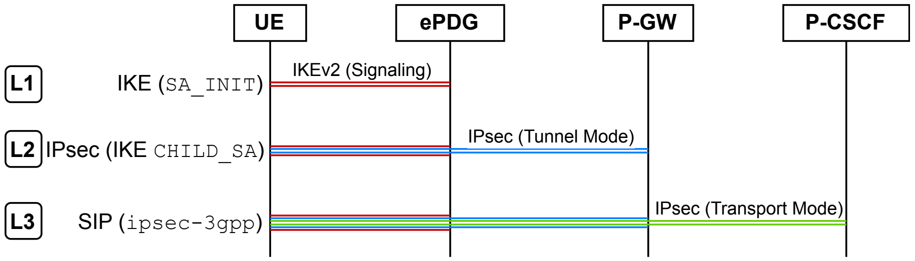


Figure 2: VoWiFi uses multiple tunnels to ensure security: L1 provides a trusted channel and manages the subsequent connections, L2 acts as a gateway to the internal infrastructure and L3 is used for the actual voice and messaging functionalities.

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
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]

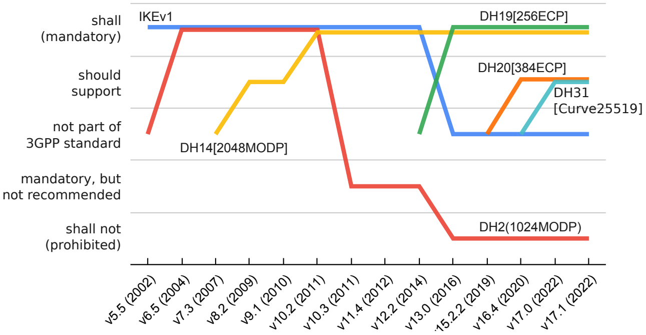


Figure 3: Development of the IPsec IKE Profile as defined in IETF TS 133.210 (IKE_SA_INIT) [23] Note: LTE started with v8, and IKEv1 has been phased out since v12.

2.5 IPsec/IKE Diffie-Hellmann 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.

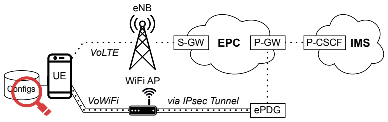
Further, any divergence from the privateness (secrecy) of a *private key* constitutes a broken encryption.

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.

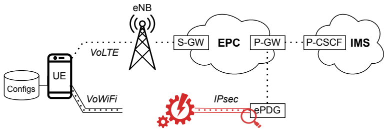


As stated in Section 2, critical information about the VoLTE and VoWiFi data bearer (or tunnels), as well as the IMS settings, need to be known to the UE in advance for each MNO. In lieu of a 3GPP autoconfigure protocol, a database of known settings for each operator is preloaded to the device. Due to 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–Hellmann groups), 2) rekeying timers, and 3) encryption, integrity algorithms & pseudo-random function (PRF).

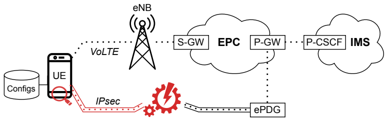
4.2 Active MNO-Side ePDG Scanning

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 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.

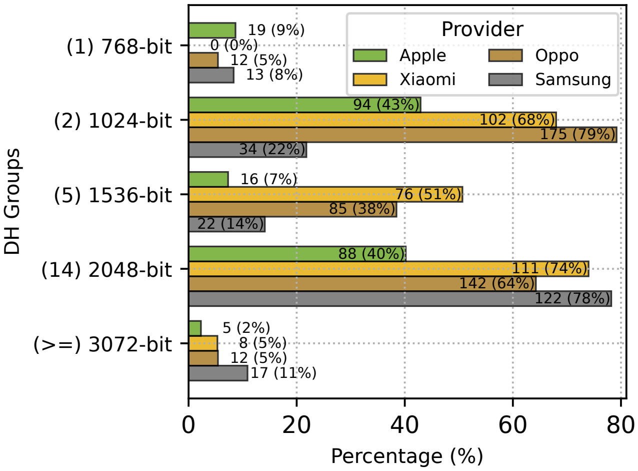


Figure 4: Number of MNOs per supported DH group (client side, grouped by device type).

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}, SHA$2^{256,384,512}$
	PRF	SHA2^{256}, SHA1, AES^{128}	*	SHA1, AES_XCBC^{128}, SHA$2^{256,384,512}$

* If no PRF is set, the PRF can be derived from the integrity algorithms. † Xiaomi Poco X3 NFC deprecated DH [23]

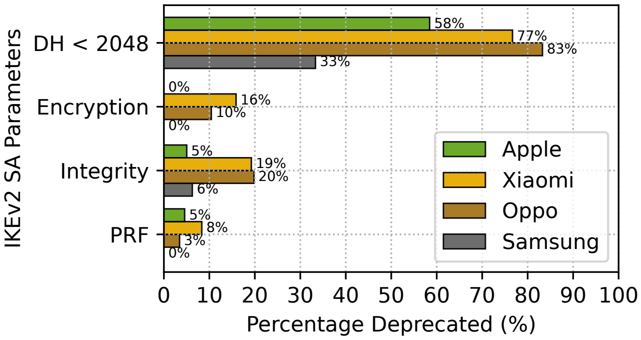


Figure 5: Share of deprecated IKEv2 parameters within all operator-specific VoWiFi settings, i.e., 83% of Oppo's configured DH settings include a deprecated DH group.

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.

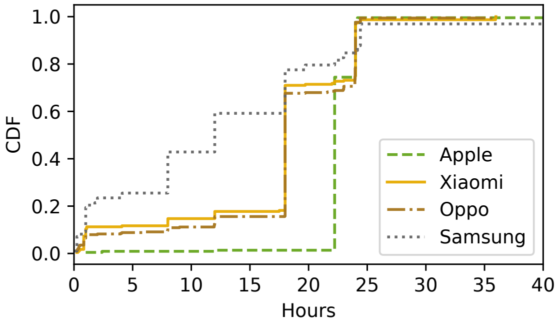


Figure 6: Operator-specific configuration of rekey timings. The majority of Apple devices is configured to renew the keys after 22 hours. For Xiaomi and Oppo, the graph represents the configured soft timers (peaking at 18 hours).

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).

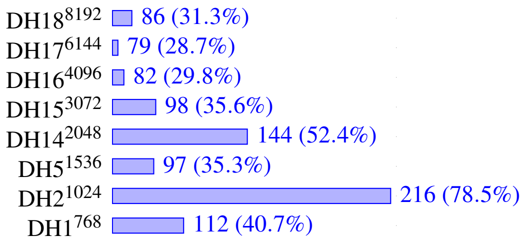


Figure 7: Number of MNOs per supported DH group

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%.

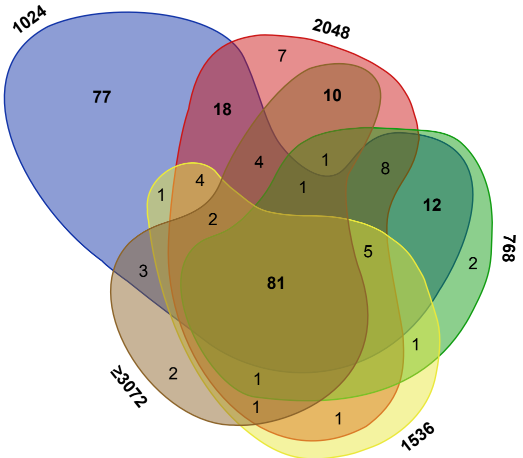


Figure 8: Number of MNOs that support a specific combination of DH key exchange groups. 3072-8192 bit groups are combined because of their low diversity.

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-ke-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 $\boxed{\text{L2}}$ 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*.

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.

[14] After the manufacturer provided a fix, a 15th ePDG/13th MNO appeared.

[15] https://www.shodan.io/

[16] https://github.com/strongswan/strongswan

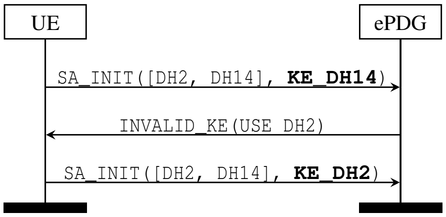


Figure 9: An ePDG server can switch from the initially selected DH group (DH14^{2048}) to a different group that is offered by the client within the proposal (DH2^{1024}).

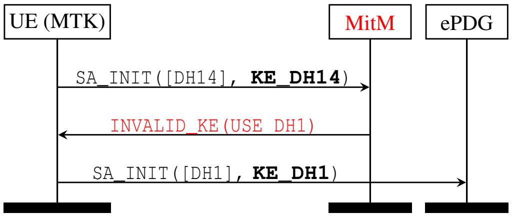


Figure 10: Some of MediaTek's Dimensity basebands are vulnerable to severe downgrade attacks, allowing the selection of DH groups that are known to be weak and were never part of the initial `IKE_INIT`-proposal nor 3GPP specification.

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.

Diffie-Hellmann 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,

[17] The standard allows both peers to trigger a rekeying prematurely, but we have not tested that.

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 reusage 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

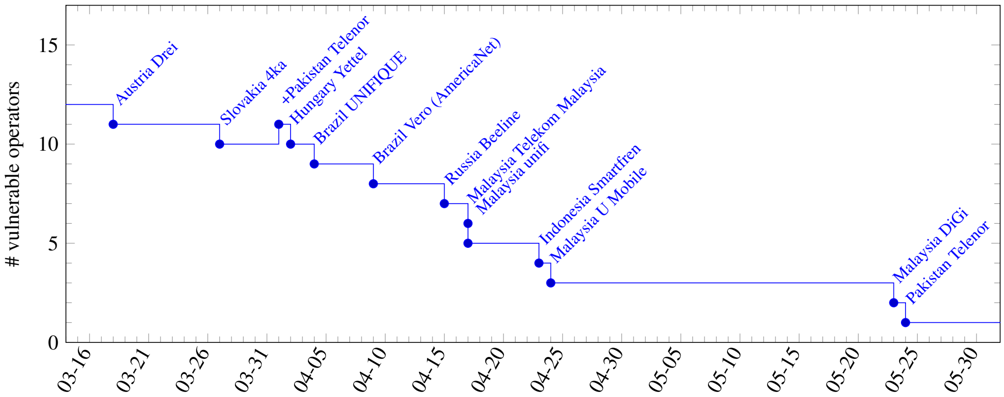


Figure 11: Globally Static Set of DH Keys: Remediation over Time

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
1f57f25e95eaeba86e5004b03058433378367e5db9126483b10b9a9262cd25b1
855aa6a8bb2b2327f52ed5d791f7ef211cc3177e50fa47c907f9a9004e91d002
b0b84567c90008babed048914325b15ff016d72b5aa46283447a6ea0b16a8fe5
8fdc2945a14dd9e7f6646107b9d324ab16a5d378e138c282c679f1de343fe447
516fb8ca4462ff067cd0611f391a289d1aaae6e73b2d96a5d7ad8444ce714a6f
a9fc5fabd3b4d94c2d11eb4ece812548a93ecb87a4ce82f883b55e6f683a5bc8
```