# TOWARD SECURE AND DEPENDABLE MOBILE NETWORKS

By

Tian Xie

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2023

# ABSTRACT

Nowadays, the world has been mobilized. By the end of 2022, mobile networks have connected billions of mobile devices and provided billions of users with ubiquitous mobile services [1]. People can use the cellular network for voice and text communication, accessing the Internet, conducting monetary transactions, etc. With the development of cellular networks, lots of new services continue to be added and provided by the operators. As mobile networks continue to evolve, with billions of devices and users connected, ensuring the security of mobile networks becomes crucial.

However, it is challenging to secure mobile networks. The mobile network is a complex ecosystem comprising various components, such as eNodeBs, MMEs, HHS, AAAs, ePDGs, P-GWs, and S-GWs, encompassing a multitude of protocols including IP, NAS, RRC, PDCP, etc., and employing multiple generations of technologies of 2G, 3G, 4G/LTE, and 5G/NR. Furthermore, the introduction of new technologies and services, such as Voice over LTE (VoLTE), Voice over Wi-Fi (VoWi-Fi, a.k.a Wi-Fi calling), and the support of cellular IoT services further contributes to the complexity. Additionally, the wide range of devices (e.g., smartphones, tablets, IoTs) connected to mobile networks and the geographical distribution of mobile network components further complicate security measures. Any vulnerability in mobile networks may threaten the entire wireless ecosystem. Thus, there is a pressing need for security research to ensure the development of secure and dependable mobile networks, which is the motivation of this dissertation to conduct the security study on the essential cellular mobile network services including IMS services, wireless IoT services, and Internet Application Services.

First, the security research of cellular network IP Multimedia Subsystem (IMS) security in mobile networks is introduced. It is the first work that investigates the security of the operational VoWi-Fi services in three major U.S. operators' networks using commodity devices. We disclose that current VoWi-Fi (Voice over Wi-Fi) security is not bullet-proof and uncover three vulnerabilities. Two proof-of-concept attacks are devised and both of them can bypass the existing security defenses. We propose solutions to address all discovered vulnerabilities. Our discovered vulner-

abilities have been confirmed by GSMA. Our findings have been acknowledged by academia and industry and received positive recognition, including **IEEE CNS Best Paper Award** and **Google Security Reward**.

Second, we focus on securing wireless IoT services, specifically cellular IoT (CIoT). By conducting our empirical security research on cellular IoT service charging over the major U.S. carriers, we discover security vulnerabilities and analyze their root causes. To assess their real-world impact, proof-of-concept attacks are devised to allow adversaries to pay less for cellular data services. In the end, we analyze the challenges in addressing these vulnerabilities and develop an anti-abuse solution to mitigate attack incentives. The solution is standard-compliant and can be used immediately in practice. The prototype and evaluation confirm its effectiveness.

Third, to overcome the fundamental obstacle for Internet Application Service (IAS), which is that there is no scalable, dependable, reliable, and privacy-preserving method to verify the IAS users' identities, we propose a novel security framework, MPKIX, designated as Mobile-assisted PKIX (Public-Key Infrastructure X.509). MPKIX secures both IAS providers and users by leveraging the broadly used PKIX services and mobile networked systems. It provides a reliable and privacy protection user verification mechanism and largely mitigates the possibility of ID theft attacks and benefits other involved parties. The evaluation results based on the prototype confirm the effectiveness and efficiency of MPKIX with low overhead. In conclusion, the novel framework, MPKIX, integrates Internet Application Services into the wide-sense mobile networks and enables the mobile network to provide secure and dependable services to its users.

Lastly, the works introduced in this dissertation are summarized. Two future research topics are discussed. In conclusion, the security research on the mobile cellular network services (i.e., IP Multimedia Subsystem services, wireless IoT services, Internet Application Services) conducted in this dissertation contributes to the advancement of secure and dependable mobile networks. They secure the mobile ecosystem, facilitate the global deployment, and head toward secure and dependable mobile networks. Our findings and solutions have implications of billions of mobile users and pave the way for a safer mobile network ecosystem.

This dissertation is dedicated to my family. Thank you for walking with me during times of challenge and inspiring me to move forward.

# ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Guan-Hua Tu, for his enthusiastic support and insightful guidance throughout my Ph.D. study. The travel to earn my doctoral degree not only guides me how to do the scientific research as a scientist and scholar, but also teaches me never shying away from any challenge.

Next, I would like to express my gratitude to Dr. Guan-Hua Tu, Dr. Li Xiao, Dr. Mi Zhang, and Dr. Zhichao Cao, who serve on my committee for both the comprehensive examination and dissertation defense. Their feedback and support in helping me pursue the degree and complete the dissertation are invaluable.

I also want to thank all my colleagues and mentors at Michigan State University. Specifically, I am grateful to the support and help from Dr. Xinyu Lei, Sihan Wang, Jingwen Shi, Yiwen Hu, Min-Yue Chen, Haitian Yan, Jiawei Li, Dr. Tengfei Ma, Jie Wan, Dr. Nan Du, Dr. Jiao Chen, Dr. Xiaopeng Bi, Dr. Yan Gong, Dr. Bin Fan, and Dr. Wen Li. Thank you all for carrying me forward.

## TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

Nowadays, the world has been mobilized. By the end of 2022, mobile networks have connected billions of mobile devices and provided billions of users with ubiquitous mobile services [1]. People can use the cellular network for voice and text communication, accessing the Internet, conducting monetary transactions, controlling the smart appliances, etc. The cellular network has played an important role in our daily life. With the development of cellular network, lots of new services continue to be added and provided by the operators.

Considering such a great amount devices and people connected, it is very important to secure mobile networks. However, it is challenging to secure the mobile network for the following reasons. First, mobile network is a complex network. It contains various components (e.g., eNodeB, MME, HHS, AAA, ePDG, P-GW, S-GW), protocols (e.g., IP, NAS, RRC, PDCP, RLC, MAC, PHY), and technologies from multiple generations (e.g., 2G, 3G, 4G/LTE, 5G/NR). Second, new technologies and services comes with the rapidly evolving technology. For instance, starting from 4G, the voice and text services are transmitted via the packet switch (PS) instead of the circuit switching (CS). Using different Radio Access Network (RAN), there are different voice and text services including Voice over LTE (VoLTE), Voice over Wi-Fi (VoWi-Fi), and Voice over New Radio (VoNR). Third, mobile network connects wide range of devices such as smartphones, tablets, IoT devices. Fourth, different parts of the network are in different geographical locations. The User Equipment (UE), the base stations (e.g., eNB), and the servers in cellular Core Network are all distributed and connected with/without wires. In such complex mobile networks, any vulnerability can further threaten the entire wireless ecosystem. It is the motivation to conduct the security research on the mobile networks services for heading toward secure and dependable mobile networks.

## 1.1 Research Overview and Contributions

As shown in Figure 1.1, the author's security research on the mobile network services can be categorized as four projects on different essential mobile network services.

**Taming cellular network IP Multimedia Subsystem:** IMS (IP Multimedia Subsystem) is an

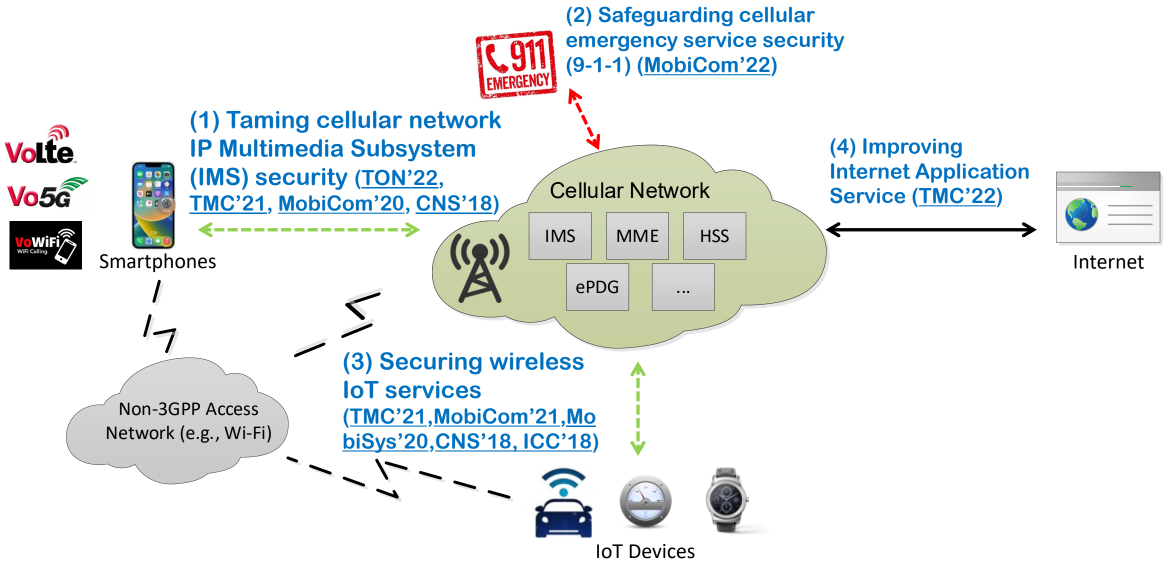

(2) Safeguarding cellular emergency service security (9-1-1) (MobiCom'22)
911 EMERGENCY
(1) Taming cellular network IP Multimedia Subsystem (IMS) security (TON'22, TMC'21, MobiCom'20, CNS'18)
VoLte
Vo5G
VoWiFi
Smartphones
Cellular Network
IMS
MME
HSS
ePDG
...
(4) Improving Internet Application Service (TMC'22)
Internet
Non-3GPP Access Network (e.g., Wi-Fi)
(3) Securing wireless IoT services (TMC'21,MobiCom'21,MobiSys'20,CNS'18, ICC'18)
IoT Devices

Figure 1.1 Research overview.

essential framework for providing 4G/5G multimedia services. It has been deployed worldwide to support three call services: VoLTE (Voice over LTE), VoNR (Voice over NR), and VoWi-Fi (Voice over Wi-Fi, a.k.a, Wi-Fi calling). Since 2016, all of three major U.S. operators have rolled out VoWi-Fi services, which enable telephony calls over the Wi-Fi networks to complement VoLTE and VoNR based on the 3GPP IMS technology. Compared with conventional cellular voice solutions, the major difference lies in that their traffic traverses untrusted Wi-Fi networks and the Internet. This exposure to insecure networks can cause the Wi-Fi calling users to suffer from security threats. Its security mechanisms are similar to the VoLTE and VoNR, because both of them are supported by the IMS. They include SIM-based security, 3GPP AKA, IPSec, etc.

However, are they sufficient to secure VoWi-Fi services? Unfortunately, after conducting the first security study on the operational VoWi-Fi services in three major U.S. operators' networks using commodity devices, we uncover that the VoWi-Fi security is not bullet-proof. Three vulnerabilities are uncovered. By exploiting the vulnerabilities, we devise three proof-of-concept attacks: telephony harassment or denial of voice service, user privacy leakage, and stealthy call DoS (Denial of Service) attack. All of them can bypass the existing security defenses. We have confirmed their feasibility using real-world experiments, as well as assessed their potential damages and proposed a solution to address all identified vulnerabilities. We actively reported and demonstrated the security threats to the industry including international telecommunication standard organizations (e.g., GSMA, 3GPP), U.S operators, device manufacturers. Our work received positive feedback in academia and industry. In academia, our work has received **IEEE CNS Best Paper Award**. In industry, the security team of **Google Android** has confirmed our findings and promised to address the vulnerability that coming from the device. Our research result can thus benefit billions of mobile phone users.

**Safeguarding cellular emergency service security:** Cellular networks that offer ubiquitous connectivity have been the major medium for delivering emergency services. In the U.S., mobile users can dial an emergency call with 911 for emergency uses in cellular networks, and the call can be forwarded to public safety answer points (PSAPs), which deal with emergency service requests. According to regulatory authority requirements for the cellular emergency services, anonymous user equipment (UE), which does not have a SIM (Subscriber Identity Module) card or a valid mobile subscription, is allowed to access them. Such support of emergency services for anonymous UEs requires different operations from conventional cellular services, and can therefore increase the attack surface of the cellular infrastructure.

In this work, we are thus motivated to study the insecurity of the cellular emergency services. We identify four vulnerabilities from cellular standard designs regarding emergency services, as well as validate them experimentally and analyze root causes. We next devise two proof-of-concept attacks with three variants each by exploiting the identified vulnerabilities (i.e., free data service attacks against cellular carriers, data DoS/overcharge and denial of cellular emergency service attacks against mobile users) and assess their real-world impact with three major U.S. cellular carriers. We finally propose a suite of standard-compliant solutions and evaluate them based on a prototype. The lessons learned can secure both cellular network carriers and mobile users. Our work received positive feedback from both academia and industry including **MobiCom Best Community Paper Runner-up** and **AT&T Security Award**.

**Securing wireless IoT services:** The User Equipment (UE) can be classfied into two categorizations, IoT devices and Non-IoT devices, according to the use scenarios. Based on the connected radio access networks, IoT devices are categorized as cellular IoT (CIoT) and Wi-Fi IoT. This dissertation focuses on our work about CIoT. Specifically, carriers are rolling out IoT services including various IoT devices and use scenarios from 2015. The support of cellular IoT services provide an alternative solution for people to access their smart devices. Compared with conventional non-IoT devices such as smartphones, IoT devices have limited network capabilities (e.g., low rates) and specific use scenarios (e.g., inside vehicles only). These specialized use scenarios lead to carries often offering cheaper device access fees for IoT devices. However, the aforementioned disparity of service charging between IoT and non-IoT devices may lead to security issues.

In this work, we conduct the first empirical security study on cellular IoT service charging over two major US carriers and make three major contributions. First, we discover four security vulnerabilities and analyze their root causes, which help us identify two significant security threats, IoT masquerading and IoT use scenario abuse. Second, we devise three proof-of-concept attacks and assess their real-world impact. We determine that they can be exploited to allow adversaries to pay 43.75%-80.00% less for cellular data services. Third, we analyze the challenges in addressing these vulnerabilities and develop an anti-abuse solution to mitigate attack incentives. The solution is standard-compliant and can be used immediately in practice. Our prototype and evaluation confirm its effectiveness.

**Improving Internet Application Service:** Nowadays, both Internet Application Service (IAS) providers and users face various security threats and legal issues. Due to the lack of reliable user information verification mechanisms, adversaries can abuse IASs to launch various cyberattacks, such as misinformation distributing and phishing, by using fake user accounts. IAS providers may thus inadvertently offer inappropriate content to restricted users, thereby suffering a serious risk of prosecution under local or international laws. Also, IAS users may suffer from nefarious ID theft attacks.

This chapter makes four contributions. First, we proposed a novel security framework, MPKIX, designated as Mobile-assisted PKIX (Public-Key Infrastructure X.509). MPKIX secures both IAS providers and users by leveraging the broadly used PKIX services and mobile networked systems.

MPKIX provides IAS providers with a reliable verification mechanism of user information while providing IAS users with cross-IAS privacy protection via the developed ppQuery mechanism. It can prevent various cyberattacks launched by false user accounts and distribution of improper content. Moreover, MPKIX secures IAS users from nefarious ID theft attacks without revealing unnecessary user information to IAS providers. By conforming to existing PKIX and cellular network standards, MPKIX has a small deployment cost. It can facilitate the delivery of accountable and secure online application services.

Second, the effectiveness of the proposed MPKIX framework is demonstrated experimentally. First, the MPKIX testbed is capable of processing up to 130,000 CSRs (Certificate Signing Requests) per minute and producing the corresponding CA-signed PKIX user certificates. Second, the terminal-side prototype of MPKIX is evaluated on both phones and computers. It is shown that MPKIX works well even on low/medium-end phone models. Third, MPKIX enables IAS providers to effectively verify the correctness of user information within less than 1 second without compromising user privacy. Fourth, the decision of the arbitration of a disputed IAS ID revocation/claim can be made within 4 seconds, whereas the current practice takes several business days or weeks.

Third, a security analysis of the MPKIX framework is conducted. It shows that MPKIX not only offers desirable security guarantees, such as data integrity, non-repudiation, user privacy, and accountability, but also defends against various attacks.

Fourth, MPKIX benefits all the involved parties. Specifically, *CAs* can expand their enterprise-based PKIX credential services to billions of mobile users. *cellular network operators* can make profit by answering the queries about user information from IAS providers. *IAS providers* can ensure the correctness of user information so that the risk of improper content distribution and cyberattacks can be minimized. *IAS users* have an efficient privacy-aware mechanism to claim/revoke impersonated IDs without revealing additional user information to IAS providers.

## 1.2 Dissertation Structure

The rest of the dissertation is structured as follows. Note that, this dissertation will not introduce the project of safeguarding cellular emergency service security in details because this work is not author's main contribution.

Chapter 2 introduce the background about the mobile network architecture, the voice call flow of the VoWi-Fi services, cellular IoT technologies, cellular IoT service charge, and PKIX (Public-Key Infrastructure (X.509) [2]).

Chapter 3 introduces the project taming cellular network IMS security with an emphasis on the operational VoWi-Fi services. Chapter 3.1 and Chapter 3.2 introduce the state-of-the-art works, the threat model, the methodology, and the ethical consideration. The discovered vulnerabilities are illustrated in Chapter 3.3. Two proof-of-concept attacks are demonstrated in Chapter 3.4 and Chapter 3.5. Chapter 3.6 presents the proposed solution to address the identified vulnerabilities.

Chapter 4 studies the new security threats in operational cellular networks coming with the cellular IoT services. Chapter 4.1 discusses the related works. Chapter 4.2 and Chapter 4.3 present the overview, the threat model, and the methodology of this work. Chapter 4.4 describes and validates the discovered vulnerabilities with the proof-of-concept attacks. Chapter 4.5 models mobile users bills, analyzes the adversary's maximum gain, and gives three attack instances to showcase real-world impact. The difficulties to secure cellular IoT service charging are introduced in Chapter 4.6. Finally, a standard-compliant solution that can rapidly mitigate the IoT attacks is presented in Chapter 4.7.

Chapter 5 introduces a novel security framework, MPKIX, designated as Mobile-assisted PKIX (Public-Key Infrastructure X.509) to improve Internet Application Service. Chapter 5.1 presents the related work first. Chapter 5.2 describes the threat model, assumptions, and offered security guarantees. Chapter 5.3 introduces the design of MPKIX. Chapter 5.4 gives the security analysis of MPKIX. Chapter 5.5 and Chapter 5.6 present the MPKIX implementation and performance evaluation, respectively. Chapter 5.7 discusses some remaining issues of MPKIX. In conclusion, the novel framework, MPKIX, demonstrated in Chapter 5 integrates Internet Application Services into the wide-sense mobile networks and enables the mobile network to provide secure and dependable services to its users.

Chapter 6 summarizes my dissertation and discusses my future research topics.

# CHAPTER 2

## BACKGROUND

In this chapter, the network architecture, the voice call flow of the VoWi-Fi services, cellular IoT technologies, cellular IoT service charge, and PKIX are introduced.

**Network architecture:** Figure 2.1 illustrates a simplified network architecture that supports both the Wi-Fi calling and VoLTE services. The UE (User Equipment), where the Wi-Fi calling and VoLTE applications are installed, connects to the similar network infrastructure including the RAN (Radio Access Network) and the CN (Core Network). For the RAN, VoLTE and Wi-Fi calling employ the eNodeB (Evolved Node B) and the Wi-Fi network, respectively. The 3GPP standard [4] classifies the Wi-Fi network into two types, namely *trusted* and *non-trusted*. For a cellular network operator, the Wi-Fi networks deployed by itself are considered trusted, whereas the others are non-trusted.

The CN consists of eight main components: the S-GW (Serving Gateway), the PDN-GW (Public Data Network Gateway), the IMS (IP Multimedia Subsystem) servers, the TWGA (Trusted Wireless Access Gateway), the ePDG (Evolved Packet Data Gateway), the HSS (Home Subscriber Server), the MME (Mobility Management Entity), and the AAA (Authentication, Authorization, and Authorization) server. For the IMS traffic delivered between the UE and the IMS servers, the VoLTE packets are routed by the S-GW and the PDN-GW; those of Wi-Fi calling are routed by the trusted Wi-Fi network, the TWAG, and the PDN-GW, or by the untrusted Wi-Fi network,

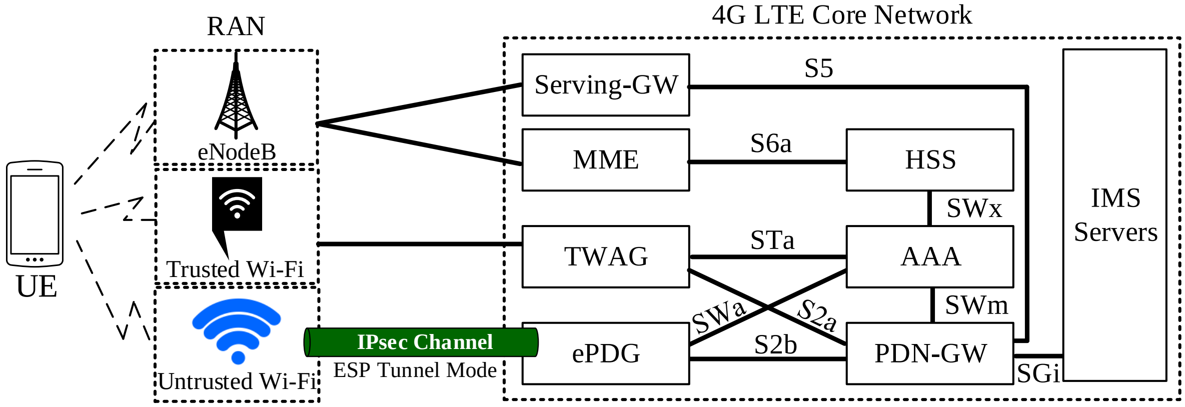


Figure 2.1 The 4G LTE network architecture that supports the Wi-Fi calling service [3].

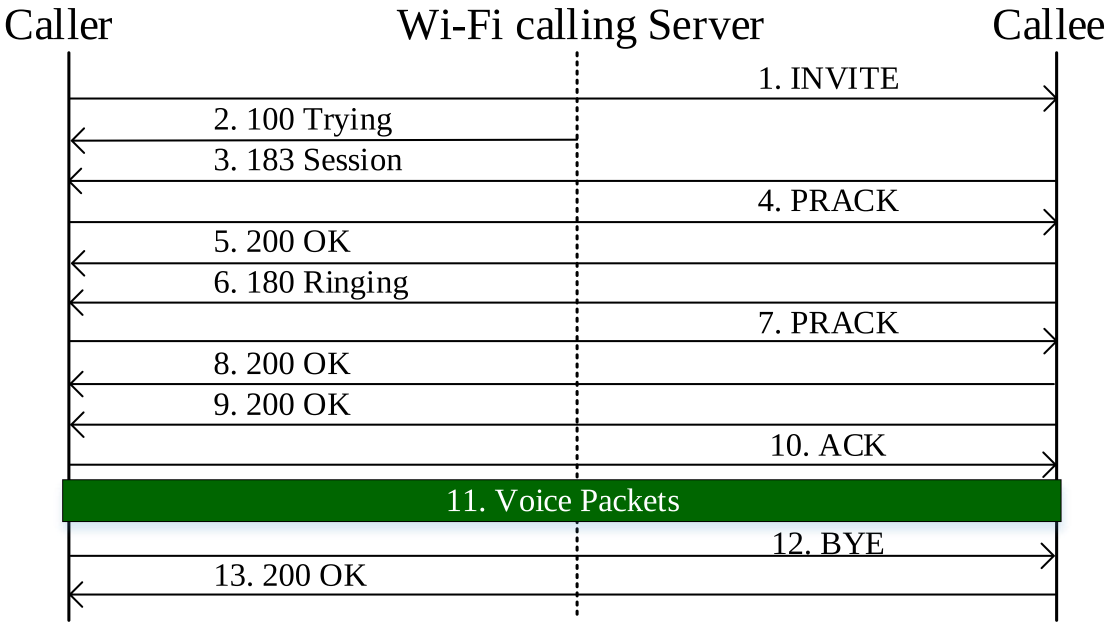


Figure 2.2 Wi-Fi calling call flow diagram.

the ePDG, and the PDN-GW. The IMS servers offer multimedia services such as voice and text services in the cellular network. The HSS stores user subscription data while, together with the AAA, providing the user authentication service. The MME takes care of user mobility and network resource reservation.

In order to protect the UE and the CN from the access of the non-trusted Wi-Fi network, the Wi-Fi calling standard [5] stipulates that the UE and the CN shall support the EAP-AKA (Extensible Authentication Protocol - Authentication and Key Agreement) procedure [6], IKEv2 (Internet Key Exchange version 2), and IPSec [7]. Specifically, they have to authenticate each other based on the EAP-AKA procedure and then establish a secure IPSec channel using the ESP tunnel mode [8, 9] between the UE and the ePDG for the Wi-Fi calling services.

**Wi-Fi calling call flow:** Figure 2.2 shows the normal call flow of Wi-Fi calling. To initiate a call, the caller sends an SIP `INVITE` message, which specifies the capabilities (e.g., voice codec) of the caller, to the callee. Afterwards, the Wi-Fi calling server at the IMS system replies to the caller with an `100 Trying` message, which indicates that the call setup is in progress. In the meantime, the callee replies to the caller with a list of available voice codecs in an `183 Session` message. After receiving the message, the caller sends a `PRACK` (Provisional Acknowledgement) message to

| | | **CAT-4 (R8)** | **CAT-1 (R8)** | **CAT-M1 (R13)** | **NB-IoT (R13)** |
|---|---|---|---|---|---|
| KPI | IoT types | Critical | Critical/Massive | Massive | Massive |
| | DL peak rate | 150 Mbps | 10 Mbps | 1 Mbps | 0.2 Mbps |
| | UL peak rate | 50 Mbps | 5 Mbps | 1 Mbps | 0.2 Mbps |
| | bandwidth | 20 Mhz | 20 Mhz | 1.4 MHz | 180 KHz |
| | battery life | day(s) | year(s) | >10 years | >10 years |
| Roll-out | Consumer IoT | ● | ● | ◐ (Few) | ◐ (Few) |
| | product carrier | ● | ● | ◐ (Partial) | ◐ (Partial) |

Table 2.1 Summary of cellular IoT technologies in operational LTE networks from US carriers [13, 10, 11, 12].

inform the callee of the selected codec. Once the `PRACK` is received, the callee phone starts to ring while sending back an `180 Ringing` message. The caller phone rings upon the arrival of the `180 Ringing` message. Whenever the callee answers the call, two call ends start to exchange voice packets for the voice call after the `200 OK` and `ACK` messages. A `BYE` message is sent from the end who terminates the call, and then the other end acknowledges it with a `200 OK` message.

**Cellular IoT technologies:** Cellular IoT is a newly emerging solution for IoT devices connected over cellular networks. They share network infrastructure with non-IoT devices (e.g, smartphones), but require special support, such as long sleep time and the delivery of small data over the control plane. Several technologies have been proposed to meet their diverse demands: CAT-4, CAT-1, CAT-M1, and NB-IoT (Narrowband IoT) [10, 11, 12], which are summarized in Table 2.1. These cellular IoT technologies support two major types of IoT applications: critical (e.g, traffic/safety control and mobile health) and massive (e.g, smart agriculture) applications. The critical IoT applications require ultra reliability, low latency, and high availability, whereas the massive IoT applications focus on low cost, low energy, and small data volumes. In the market, CAT-4 and CAT-1 have been widely deployed by US carriers, but other technologies have not (e.g., Verizon and AT&T support only CAT-M1 whereas T-Mobile supports only NB-IoT). Most consumer IoT devices, such as wearable devices, car-connected mobile hotspots, and tracking sensors, belong to CAT-4, CAT-1, and CAT-M1.

Figure 2.3 shows the 4G LTE network architecture with IoT support. The network architecture consists of two major components: Radio Access Network (RAN) and core network. The RAN

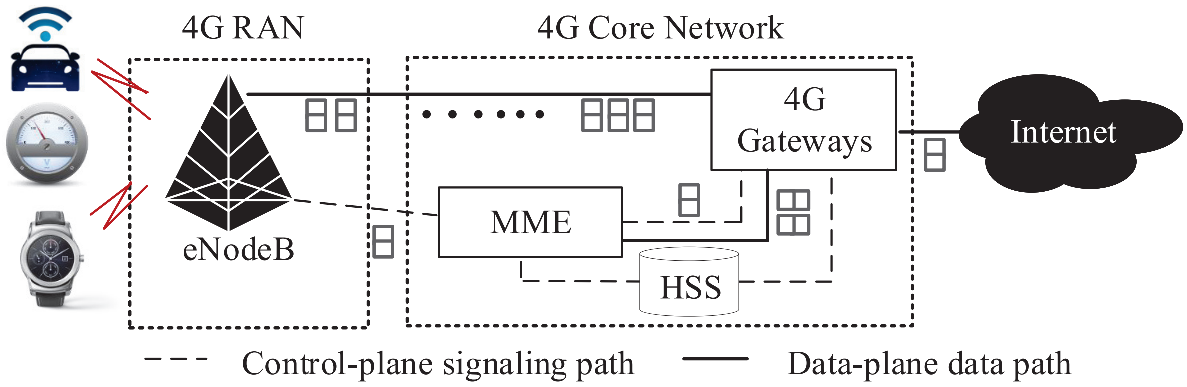


Figure 2.3 4G LTE network architecture with IoT support.

| Carriers | Data plan | Monthly Charge fees | Non-IoT devices | | IoT devices | | |
|---|---|---|---|---|---|---|---|
| | | | | | CAT-4 | | |
| | | | Smartphone | Portable Mobile Hotspot | Wearable | Car-connected Mobile Hotspot | CAT-1/CAT-M1 |
| OP-I | Limited data plan | Device access fee | $20 | $20 | $10 | $10 | $0 |
| | | Service access fee | $50 (3GB) | $0* | $0* | $0* | $0.99(0.5MB),$14(0.1GB),$22(1GB),$35(5GB) |
| | Unlimited data plan | Device access fee | $35 | No unlimited data plans | $10 | $20 | No unlimited data plans |
| | | Service access fee | $75 | | $0* | $0* | |
| OP-II | Limited data plan | Device access fee | $20 | $10 | $5, $10 (varying with models) | $10 | $0 |
| | | Service access fee | $35 (2GB) | $0* | $0* | $0* | $2(0.2MB),$18(0.15GB),$25(1GB),$50(5GB) |
| | Unlimited data plan | Device access fee | $0(1),$65(2),$75(3),$85(4) | $20 | $5, $10 (varying with models) | $20 | No unlimited data plans |
| | | Service access fee | $75 | $0* | $0* | $0* | |

Table 2.2 Data plans for IoT and non-IoT devices in two US carriers (studied in Dec. 2018). The price and volume cap are shown by per month unless explicitly specified. Many variants may not be included, for example, $60 for 10GB per 30 days for OP-I IoT sim cards [14] ($0*: Shared the fee with phones).

allows IoT devices to transmit IoT data to cellular network infrastructure using the aforementioned cellular IoT technologies. The core network includes three main network elements: Mobility Management Entities (MMEs), 4G gateways, and the Home Subscriber Server (HSS). The MMEs are responsible for user mobility, user authentication, and resource reservation. Additionally, the MMEs are responsible for new IoT functions [15], such as power saving mode and extended discontinuous reception [16]. The HSS stores user subscription data and user information profiles. The 4G gateways forward data between the RAN and the Internet, as well as collect device data usage.

**Cellular IoT service charge:** We investigate the service charges of IoT devices from two top-tier US carriers denoted as OP-I and OP-II and compare them with those of non-IoT service charges. Table 2.2 summarizes the comparison. The SIM card used for each device is associated with the owner's non-IoT or IoT data plan. For both device types, a device's charge includes two kinds of fees, device and service access fees. Its bill can be formulated as $B(u) = \alpha + u \otimes \beta$, where $\alpha$ is the device access fee and $u \otimes \beta$ represents the service access fee determined by actual data usage volume $u$ and unit price $\beta$. In most cases, unlimited voice and text services are offered, so the formula does not include them. The service charges vary not only with device types and models but also with limited and unlimited data plans.

The unlimited data plans often have higher device access fees than those of the limited data plans. For instance, the device access fees are \$20 and \$35 for a smartphone line in OP-I's limited and unlimited plans, respectively. The limited plans usually have service access fees increasing with capped data usage volumes, in contrast to fixed service fees in the unlimited data plans. For example, OP-II charges \$35, \$50 and \$70 for monthly volumes of 2 GB, 4 GB, and 8 GB, respectively. Note that the increase is not proportional for most carriers except Google Project Fi [17], which charges \$15 for each 1 GB, is one of few exceptions.

In terms of the service charging policies, IoT devices differ from non-IoT devices in two aspects. First, IoT device access fees are cheaper, since IoT devices require much smaller data usage volumes than non-IoT devices. The IoT device access fees may also vary with device models. For example, OP-II charges \$5 for an LG Watch Urbane2 and \$10 for an Apple Watch. Second, IoT service access fees are usually tied to non-IoT data plans, but there are still some IoT-specific data plans. The IoT-specific data plans offer lower service fees per data unit. For example, OP-I offers 5 GB [14] to IoT users at only \$35, but offers the same amount of data to smartphone users at \$50.

**PKIX:** PKIX (Public-Key Infrastructure (X.509) [2]) is built based on the asymmetric cryptography, in which the data encrypted by a public key can only be decrypted by its paired private key and vice versa. The public key is disseminated to the public, whereas the private key is known only by its owner. The PKIX certificate is usually formed in the format of X.509, which is an ITU-T (International Telecommunications Union) standard. The certificate contains three main elements, namely (1) the subject (owner) information (e.g., name, residence and age), (2) the owner's public key, and (3) the digital signature of the CA that issued the certificate. In practice, to obtain a CA-signed PKIX user certificate, the applicant needs to provide the CA with a government-issued photo ID and a CSR [18] request containing the applicant's subject information, public key, and digital signature. The CA confirms the applicant's identity by validating his/her digital signature using the public key and verifying the subject information by inspecting the photo ID. After the confirmation, the CA generates a PKIX user certificate and attaches a digital signature generated for the certificate.

# CHAPTER 3

# THE UNTOLD SECRETS OF WIFI-CALLING SERVICES: VULNERABILITIES, ATTACKS, AND COUNTERMEASURES

Since 2016, all of four major U.S. operators have rolled out Wi-Fi calling services. They enable mobile users to place cellular calls over Wi-Fi networks based on the 3GPP IMS technology. Compared with conventional cellular voice solutions, the major difference lies in that their traffic traverses untrusted Wi-Fi networks and the Internet. This exposure to insecure networks can cause the Wi-Fi calling users to suffer from security threats. Its security mechanisms are similar to the VoLTE, because both of them are supported by the IMS. They include SIM-based security, 3GPP AKA, IPSec, etc. However, they are not sufficient to secure Wi-Fi calling services. In this project, the first security study on the operational Wi-Fi calling services in three major U.S. operators networks using commodity devices is conducted and makes four contributions.

1. We conducted the first security study to explore the dark side of operational Wi-Fi calling services in five operational cellular networks in the U.S. and Taiwan using commodity devices. We identified three Wi-Fi calling vulnerabilities, each of which roots in a design defect of the Wi-Fi calling standard or an operational slip of the operators.

2. We devised two proof-of-concept attacks by exploiting the identified vulnerabilities and assessed their negative impacts in a responsive manner.

3. We developed a practical solution, Wi-Fi Calling Guardian, to address the identified vulnerabilities. Our experiments confirm that it can protect the Wi-Fi calling users from the proposed security threats.

4. We actively reported and demonstrated the security threats to the industry, and received a positive feedback. Specifically, the security team of Google Android has confirmed our findings and promised to address the vulnerability that coming from the device. Our research result can thus benefit billions of Android phone users.

## 3.1 Related Work

**Cellular Network Security:** Cellular network security is getting more attention in recent years. Christian et al. [19] proposed Sonar to detect SS7 redirection attacks with audio-based distance bounding. Reaves et al. [20] introduced AuthentiCall to protect voice calls made over traditional telephone networks by leveraging now-common data connections available to call endpoints. Another study [21] analyzed nearly 400,000 text messages sent to public online SMS gateways over the course of 14 months and offered insights into the prevalence of SMS spam and behaviors. Jover [22] summarized the current state of affairs in the 5G protocol security and discussed the related areas that can be improved further. He et al. [23] presented a comprehensive survey of the attacks including RF jamming, signaling attacks, various SIP attacks, etc., in the LTE network. The other three works[24, 25, 26] study various attacks for SIP on different levels, discuss a potential attack based on SIP signaling, and classify existing SIP attacks and defenses, respectively. Compared with them, our work focuses on the security of the newly deployed Wi-Fi calling service security, which has not been fully explored yet.

**VoIP and VoLTE Security:** The security problem of the VoIP and VoLTE system has attracted lots of attentions. Two studies [27, 28] examine side-channel attacks on VoIP traffic. McGann et al. [29] analyzed the security threats and tools in the VoIP system. Several security issues (e.g., Toll Fraud) of VoIP applications were discussed in [30]. Li et al. [31] examined the security implications of VoLTE, which include several vulnerabilities (e.g., improper charing policies). Dacosta et al. [32] proposed the use of a modified version of OpenSER to improve authentication performance of distributed SIP proxies. This chapter studies the Wi-Fi calling service from the perspectives of the standard, the implementation, and the operation, which are not covered by the prior arts.

**Side-Channel Attacks Against Mobile Systems:** The side-channel information leakage against mobile systems has been a popular research area in recent years. Current studies [27, 28] target the side-channel information leaked by mobile users' traffic, which is generated by some particular Internet services, and then seek to infer users' activities. The work [33] introduces the analysis on automatic fingerprinting of mobile applications for arbitrarily small samples of Internet traffic. Ali et al. [34] illustrated that each app leaves a fingerprint on its traffic behavior (e.g., transfer rates, packet exchanges, and data movement). Another work [35] demonstrates automatic fingerprinting and real-time identification of Android applications from their encrypted network traffic, which even could work when HTTPS/TLS is employed. Eskandari et al. [36] analyzed the personal data transfers in mobile apps and revealed that 51% of these apps did not provide any privacy policy. The paper [37] demonstrates discerning of mobile user location within commercial GPS resolution by leveraging the ability of mobile device magnetometers to detect externally generated signals in a permissionless attack. Reaves et al. [38] did the security analysis on the branchless banking applications. Different from them, we focus on the insecurity of the cellular Wi-Fi calling service, which is stipulated by the 3GPP and is going to be deployed globally on billions of mobile devices in the near future.

**Wi-Fi Security:** There are many novel studies related to Wi-Fi security. Liu et al. [39] used the fine-grained channel information to authenticate the user. Lee et al. [40] examined the limitations of the existing jamming schemes against channel hopping Wi-Fi devices in dense networks. Li et al. [41] inferred user demographic information by exploiting the meta-data of Wi-Fi traffic. Another study [42] proposes the system, the Wi-Fi Privacy Ticker, to improve participants' awareness of the circumstances in which their personal information is transmitted. Mikhail et al. [43] proposed an SBN model to effectively detect intrusions in the enterprise networks and the 802.11 wireless networks. Kolias et al. [44] categorized and evaluated popular attacks on the 802.11 networks, and applied different learning models to the collected dataset for the intrusion detection. Different from the prior art, our work investigates the insecurity of the Wi-Fi calling services, which have been deployed worldwide by cellular network operators, instead of new Wi-Fi vulnerabilities.

**Wi-Fi Calling Security:** Wi-Fi calling security is a new research area and has not been fully studied by the academic yet, since carriers just deployed their Wi-Fi calling services in recent years. Current researchers mainly focus on the security vulnerabilities on Wi-Fi calling devices. Specifically, Beekman et. al pointed out that T-Mobile Wi-Fi calling devices (e.g., Samsung S2) are vulnerable to invalid server certificates [45]. Chalakkal et. al studied SIM-related security issues on Wi-Fi calling devices [46]. However, our work examines the Wi-Fi calling security from two aspects: standards and operations.

### 3.2 Threat Model, Methodology and Ethical Considerations

**Threat model:** Compared to the limited deployment of trusted Wi-Fi networks, the non-trusted public Wi-Fi networks have been broadly deployed in practice, including those in campuses, libraries, grocery stores, coffee shops, to name a few. The present study mainly targets the security threats while users are using non-trusted public Wi-Fi networks. Adversaries are people or organizations which attack the Wi-Fi calling users. We consider the adversaries with the following capabilities: (1) they can *intercept*, *modify*, or *inject* any messages in the public communication channels (inside or outside connected Wi-Fi networks, e.g., Internet); (2) they adhere to all cryptographic assumptions, e.g., adversaries cannot decrypt an encrypted message without the decryption key; (3) they cannot compromise the Wi-Fi calling devices or the cellular network infrastructure, but may access/deploy surveillance cameras near the victims.

**Methodology:** We validate the vulnerabilities and the attacks on three major U.S. carriers, which together take about 75% of market share, and two Taiwan carriers, which together take 45% of market share. We conduct experiments using two Wi-Fi APs, a software-based AP based on a MacBook Pro 2014 laptop and an ASUS RT-AC1900 AP, and eight popular smartphones with the Wi-Fi calling service, which include Samsung Galaxy S6/S7/S8/J7, Apple iPhone6/iPhone7/iPhone8, and Google Nexus 6P. Apple and Samsung already take 74% share of the smartphone market [47]. The experiments are conducted in the Wi-Fi networks of several campuses, including Michigan State University, New York University, University of California Berkeley, and Northeastern University.

**Ethical considerations:** We understand that some feasibility tests and attack evaluations might be harmful to the operators and/or users. Accordingly, we proceed with this study in a responsible manner by running experiments in fully controlled environments. In all the experiments, victims are always our lab members. Our goal is to disclose new security vulnerabilities and provide effective solutions, instead of aggravating the damages.

## 3.3 Vulnerabilities

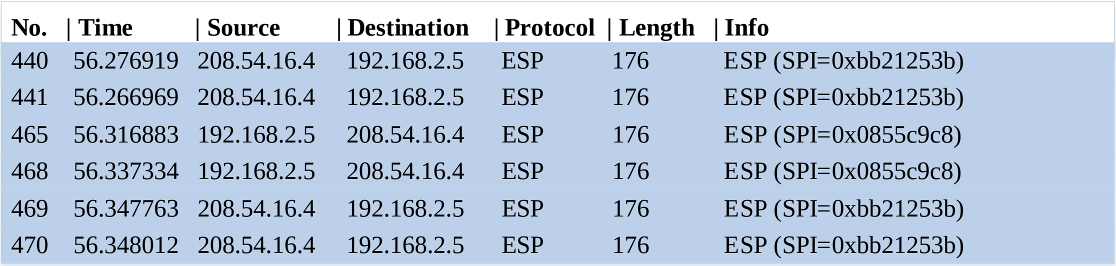
In this subchapter, we first introduce three security vulnerabilities discovered from operational Wi-Fi calling services in the U.S., and then present a study on non-U.S. operators and a feedback from the industry.

### 3.3.1 V1: WLAN selection mechanisms for Wi-Fi calling devices merely consider radio/connectivity capabilities of available Wi-Fi networks

The first vulnerability is that all studied Wi-Fi calling devices cannot exclude an insecure Wi-Fi network while enabling Wi-Fi calling services. According to Wi-Fi calling standards[5, 48], there are two Wi-Fi network selection modes: manual and automatic modes. In the manual mode, devices maintain a prioritized list of selected Wi-Fi networks, the implementation of which is vendor-specific. In the automatic mode, devices select their connected Wi-Fi networks by following the guidance from the network infrastructure based on the ANDSF (Access Network Discovery and Selection Function) procedure described in [3]. However, both modes do not consider security risks of available Wi-Fi networks but radio quality (e.g., ThreshBeaconRSSIWLANLow [48]) and connectivity capabilities, such as MaximumBSSLoad (i.e., the loading of Wi-Fi AP), Minimum-BackhaulThreshold (e.g., 2 Mbps in the downlink) [3, 49].

**Validation:** We deploy two Wi-Fi routers of the same model to test the Wi-Fi network selection of the Wi-Fi calling devices. The experiment is conducted with four steps as follows. First, those two routers are deployed 5 and 10 meters, respectively, away from the tested devices. All test Wi-Fi calling devices are pre-installed with the required credentials to access these two Wi-Fi routers. Second, the security mechanism against the ARP (Address Resolution Protocol) spoofing attack, which is the prerequisite of various MitM (Man-in-the-Middle) attacks, is enabled on the far router, but it is disabled on the near router. Third, we launch an ARP spoofing attack from a computer that connects to the near router, to perform an MitM attack against all the other devices connecting to the router. Fourth, we enable the Wi-Fi calling service on all the tested devices, and then make a Wi-Fi calling call on each device whenever the device successfully has a Wi-Fi network connected.

We have three observations from the experiment. First, all the test Wi-Fi calling devices connect



```
No.  | Time       | Source         | Destination   | Protocol | Length | Info
440   56.276919   208.54.16.4      192.168.2.5     ESP        176      ESP (SPI=0xbb21253b)
441   56.266969   208.54.16.4      192.168.2.5     ESP        176      ESP (SPI=0xbb21253b)
465   56.316883   192.168.2.5      208.54.16.4     ESP        176      ESP (SPI=0x0855c9c8)
468   56.337334   192.168.2.5      208.54.16.4     ESP        176      ESP (SPI=0x0855c9c8)
469   56.347763   208.54.16.4      192.168.2.5     ESP        176      ESP (SPI=0xbb21253b)
470   56.348012   208.54.16.4      192.168.2.5     ESP        176      ESP (SPI=0xbb21253b)
```

Figure 3.1 A trace of the Wi-Fi calling packets intercepted based on the ARP spoofing.

to the near Wi-Fi router. Second, all the Wi-Fi calling packets from the tested devices are intercepted by the computer based on the ARP spoofing attack, as shown in Figure 3.1. Third, none of the tested devices disconnects from the near router or terminates their Wi-Fi calling services; not any alerts or warnings are observed from the tested devices. This validation experiment confirms that current WLAN selection mechanisms do not prevent the Wi-Fi calling devices from connecting to an insecure Wi-Fi network, thereby causing them to suffer from the MitM attack. Note that the MitM attack does not need to compromise or control the near router.

**Security implications:** It is not without reasons that the WLAN selection mechanisms do not take security issues into consideration but consider only the radio quality or/and WLAN performance, since the Wi-Fi calling sessions have been protected by the IPSec tunnels with the end-to-end confidentiality and integrity protection. Although the security protection can prevent the Wi-Fi calling packets from being decrypted or altered, intercepting or discarding those packets for further attacks is still possible. We believe that 3GPP and GSMA shall revisit the Wi-Fi network selection mechanisms for the Wi-Fi calling service in terms of security; otherwise, the Wi-Fi calling users are being exposed to potential security threats.

### 3.3.2 V2: Potential Side-channel Inference

Given the security mechanisms of untrusted access, the packets of the cellular services under untrusted Wi-Fi networks can be securely delivered through the IPSec channel between the UE and the ePDG. However, we discover that for all the test operators, the Wi-Fi calling service is the only service carried by the IPSec channel. This monotonous operation may allow the adversary to monitor the channel and then launch a side-channel attack to infer user privacy from the Wi-Fi

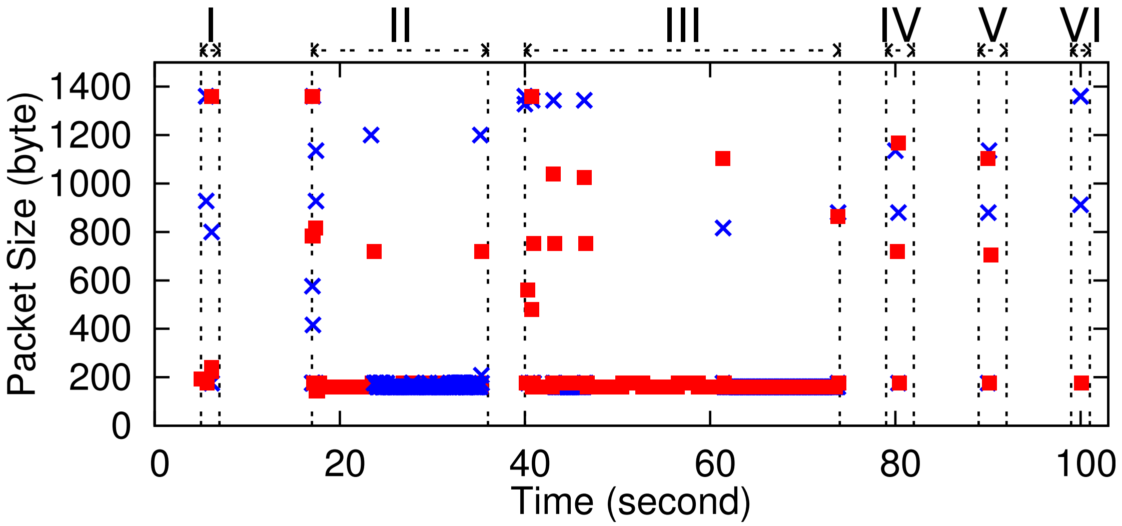
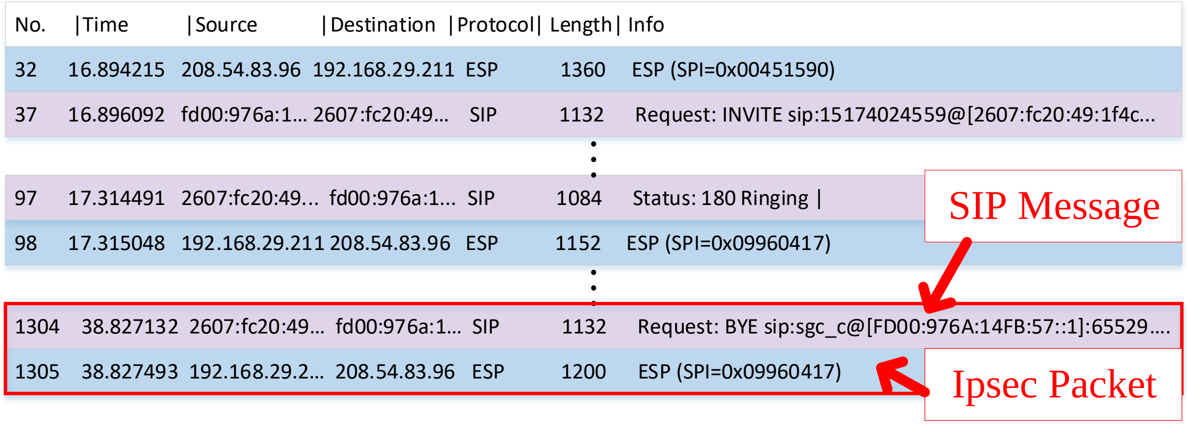


Figure 3.2 The IPSec packets of six Wi-Fi calling events over time (×: uplink packets; ■: downlink packets; I/VI: Activating/Deactivating Wi-Fi calling; II/III: Receiving/Dialing a call; IV/V: Sending/Receiving a text).



```
No.   |Time       |Source         |Destination   |Protocol| Length| Info
32     16.894215  208.54.83.96   192.168.29.211  ESP      1360    ESP (SPI=0x00451590)
37     16.896092  fd00:976a:1... 2607:fc20:49...  SIP      1132    Request: INVITE sip:15174024559@[2607:fc20:49:1f4c...
                                                                  ⋮
97     17.314491  2607:fc20:49...  fd00:976a:1... SIP      1084    Status: 180 Ringing |
98     17.315048  192.168.29.211 208.54.83.96    ESP      1152    ESP (SPI=0x09960417)
                                                                  ⋮
1304   38.827132  2607:fc20:49...  fd00:976a:1... SIP      1132    Request: BYE sip:sgc_c@[FD00:976A:14FB:57::1]:65529....
1305   38.827493  192.168.29.2...  208.54.83.96   ESP      1200    ESP (SPI=0x09960417)
```

Figure 3.3 A trace of the Wi-Fi calling packets collected on a test phone: SIP and IPSec packets.

calling events (e.g., call and text messaging statuses) and call statistics.

**Validation:** We examine whether any information can be inferred based on the intercepted Wi-Fi calling packets, which are encrypted by IPSec. After analyzing their patterns, we discover that for all the three operators, there are six service events of the Wi-Fi calling service, namely dialing/receiving a call, sending/receiving a text message, and activating/deactivating the service.

Figure 3.2 shows the IPSec packets captured on a Wi-Fi AP when the above six events are triggered on a test phone connecting to the AP. It is observed that all the events differ from each other in terms of traffic patterns, which are composed of packet direction (uplink or downlink),

| Test Device | US-I | US-II | US-III |
|---|---|---|---|
| Samsung J7 (US-III) | N/A | N/A | 100% |
| Samsung S6 (US-II) | N/A | 100% | N/A |
| Samsung S7 (US-I) | 100% | N/A | N/A |
| Samsung S8 (US-II) | N/A | 100% | N/A |
| Nexus 6P | 100% | N/A | N/A |
| iPhone 6 | 100% | 100% | 100% |
| iPhone 7 | 100% | 100% | 100% |
| iPhone 8 | 100% | 100% | 100% |

Table 3.1 Classification accuracy of the Wi-Fi calling events in various cross-phone and cross-carrier cases. N/A means that the test phone does not support the carrier's Wi-Fi calling service.

packet size, and packet interval. In order to automatically identify them based on the encrypted Wi-Fi traffic, we apply a decision tree method, the C4.5 algorithm [50]. To prepare a set of training data, we trigger those six events on the test phone with 50 runs each while collecting all the IPSec packets on the Wi-Fi AP. Based on the training data, a classification model can be generated by the C4.5 algorithm. We assess the classification accuracy of the model using 50 tests by comparing the model's output with the test phone's packet trace as shown in Figure 3.3. The result shows that the model can give 100% accuracy. Note that the test phone is Nexus 6P with the Wi-Fi calling service of US-I.

We next examine whether the classification model works for cross-phone and cross-carrier cases. We consider various devices with the Wi-Fi calling services of the three carriers. Table 3.1 summarizes the result. It is observed that those six events in all the test cases can be identified accurately. Accordingly, the model that is derived based on the training data collected from Nexus 6P with the US-I's Wi-Fi calling service can be applied to the other devices and carriers, which include the Samsung Galaxy J7/S6/S7/S8 and iPhone 6/7/8 devices with the US-II/US-III networks.

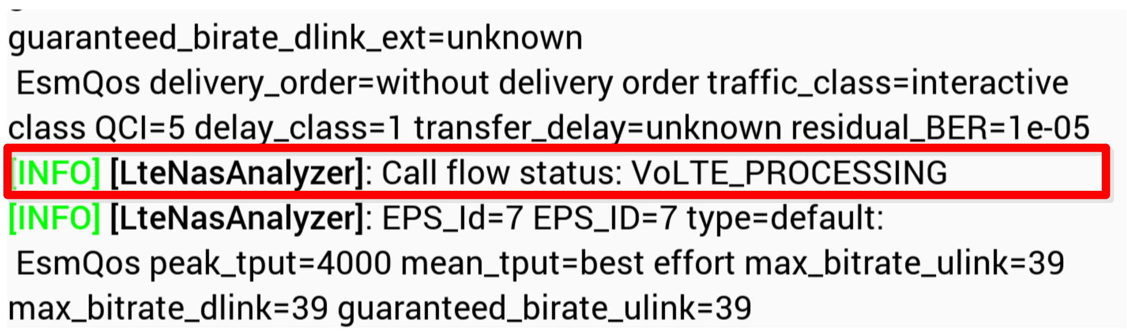
**Security implications:** The IPSec channel can prevent man-in-the-middle attackers from decrypting or altering the Wi-Fi calling packets, but does not block the side-channel inference attack. Its monotonous operation allows the adversary to collect '*clean*' Wi-Fi calling traffic, which simplifies the side-channel inference.

### 3.3.3 V3: the Inter-system Service Continuity Mechanism of Wi-Fi Calling can be Bypassed

The inter-system service continuity mechanism can seamlessly switch the voice service of Wi-Fi calling on a device back to the cellular-based voice service (e.g., VoLTE), when the device disconnects from its connected Wi-Fi network or it cannot be reached through the Wi-Fi network (e.g., no response from the device in the Wi-Fi calling service). The mechanism can be triggered by the device or the cellular network infrastructure, and mainly consists of two steps, namely an inter-system handover [3] between Wi-Fi and the cellular network, and a procedure of the IMS service continuity [51]. Its operation can inherently protect the device against a DoS attack on the Wi-Fi calling service. For example, when all the Wi-Fi calling packets are maliciously dropped, the device is unreachable. However, the operation is not bullet-proof and may be bypassed with a sophisticated attack.

**Validation:** We conduct experiments to examine whether the mechanism can be bypassed in any scenarios. We test a Wi-Fi calling device with the following four scenarios, together with their corresponding results. First, the device with an established voice call of Wi-Fi calling moves out of its connected Wi-Fi network. We observe that the ongoing voice call can successfully migrate from Wi-Fi calling to VoLTE without any call interruption. Second, the device is dialing a Wi-Fi calling call while all its Wi-Fi calling packets are discarded from the Wi-Fi AP. We find that the device successively sends a packet of `SIP INVITE` to the Wi-Fi calling server; after six attempts, it switches to initiating a VoLTE call, as shown in Figure 3.4. Third, while the device is having an incoming call, all the Wi-Fi calling packets are discarded. It is observed that the device switches to VoLTE for the incoming call. Fourth, the packets of a Wi-Fi calling call on the device are discarded right after the call is established. We observe that no voice can be heard from two call ends, but the inter-system switch is not triggered and the device keeps the connection of the Wi-Fi network.

In summary, the inter-system service continuity mechanism is triggered only when the radio quality of the connected Wi-Fi network becomes bad, or the device and the network infrastructure cannot reach each other in the Wi-Fi calling service. As in the above fourth case, where the device



```
guaranteed_birate_dlink_ext=unknown
 EsmQos delivery_order=without delivery order traffic_class=interactive
class QCI=5 delay_class=1 transfer_delay=unknown residual_BER=1e-05
[INFO] [LteNasAnalyzer]: Call flow status: VoLTE_PROCESSING
[INFO] [LteNasAnalyzer]: EPS_Id=7 EPS_ID=7 type=default:
 EsmQos peak_tput=4000 mean_tput=best effort max_bitrate_ulink=39
max_bitrate_dlink=39 guaranteed_birate_ulink=39
```

Figure 3.4 A trace shows that a device switches an ongoing call attempt from Wi-Fi calling to VoLTE after all the Wi-Fi calling packets are dropped. It is obtained on the test phone via the software MobileInsight [52].

and the network can reach each other but some packets are dropped, the adversary can attack a device's Wi-Fi calling call while keeping the device using the Wi-Fi calling service by preventing the inter-system switch from being triggered.

**Security implications:** Although the Wi-Fi calling standard [3, 51, 5] provides the inter-system switch mechanism for the Wi-Fi calling service continuity, it may suffer from some sophisticated attacks where the Wi-Fi calling packets can be intercepted. The interception is possible since the Wi-Fi calling traffic needs to traverse untrusted Wi-Fi networks and the Internet. To prevent the attacks, the service continuity mechanism should also take security concerns into consideration.

### 3.3.4 A Vulnerability Study on Non-U.S. Operators

We conduct a study of the Wi-Fi calling vulnerabilities on two Taiwan operators to examine whether they are limited to only U.S. operators or not. We summarize the result of the test phone, Samsung Galaxy S8, for each vulnerability as follows.

**V1:** We repeat the validation experiment of V1 on the phone with the Taiwan operators, and observe the same result that the WLAN selection mechanism does not prevent the device from connecting to an insecure Wi-Fi network, where an ARP spoofing attack is launched.

**V2:** For both Taiwan operators, we observe that the Wi-Fi calling service is also the only one service carried by the IPSec channel, and then apply the same classification method described in Chapter 4.4.1.2 into classifying the aforementioned six events. The result summarized in Table 3.2 confirms that the method can give 100% accuracy for the event inference.

| Operator | Act./Deact. Wi-Fi calling | Rec./Dialing a call | Sending/Rec. a text |
|---|---|---|---|
| TW-I | 100%/100% | 100%/100% | N/A |
| TW-II | 100%/100% | 100%/100% | 100%/100% |

Table 3.2 Classification accuracy of the six Wi-Fi calling events for two Taiwan operators. N/A means that the event is not supported.

**V3:** We test the device with the Wi-Fi calling services of the Taiwan operators for the inter-system service continuity mechanism. It is also observed that the mechanism is deployed and can be bypassed in the fourth scenario described in Chapter 4.4.2.

### 3.3.5 Industry Feedback

We have reported the vulnerabilities to the U.S. operators that are studied in this work and several device manufacturers including Google, Samsung, and Apple. In particular, the Google Android security team gives a positive feedback that the team has confirmed our findings after a security analysis of the vulnerabilities, and will address them in an upcoming security patch. We thus received a Google Security Reward in Jan. 2020. On the other hand, we are awaiting hearing from the other operators and manufacturers.

## 3.4 Telephony Harassment/Denial of Voice Service (THDoS) Attack

We next devise the THDoS attack, which can cause telephony harassment or denial of voice service on the Wi-Fi calling users. In the following, we describe the attack design, evaluation and real-world impact.

### 3.4.1 Attack Design

In this attack, the adversary manages to discard particular signaling or/and voice packets of Wi-Fi calling from the victim device, while preventing the inter-system service continuity mechanism from being triggered. The attack can cause damage on the device's voice service supported by Wi-Fi calling, and let the damage last by getting the device stuck with the Wi-Fi calling service. To discard particular packets between the device and the network infrastructure, the adversary needs to identify encrypted IPSec packets. We next start with an illustrative example of the Wi-Fi calling call, and then analyze the traffic patterns of the Wi-Fi calling messages and events based on the encrypted packets.

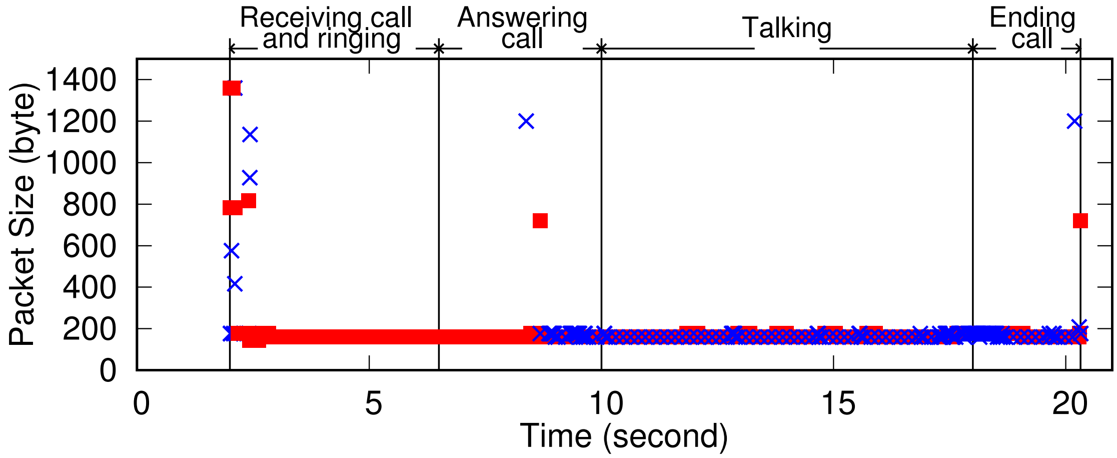
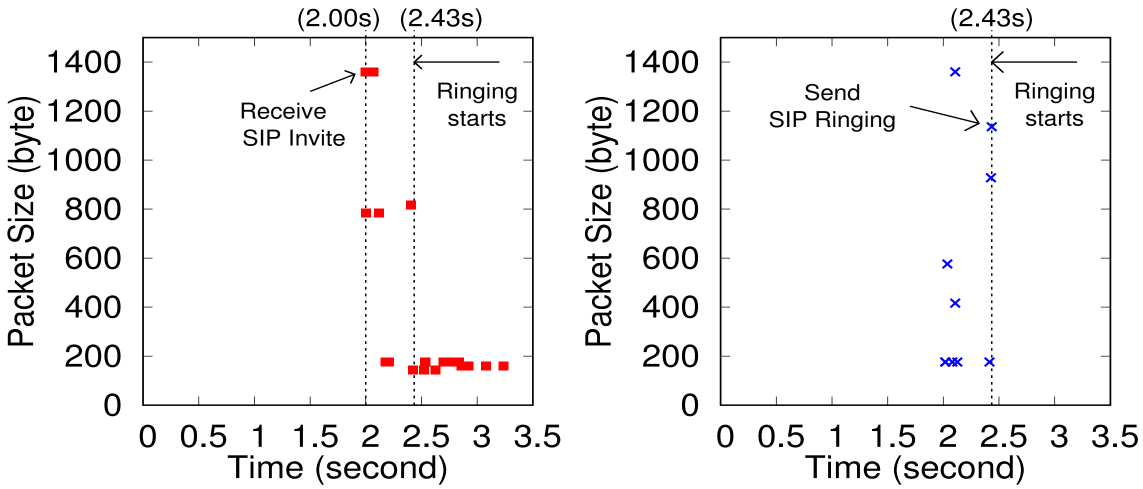


Figure 3.5 The IPSec packets of a Wi-Fi calling call, which are observed on the Wi-Fi AP to which the callee connects. (×: uplink packets; ■: downlink packets).



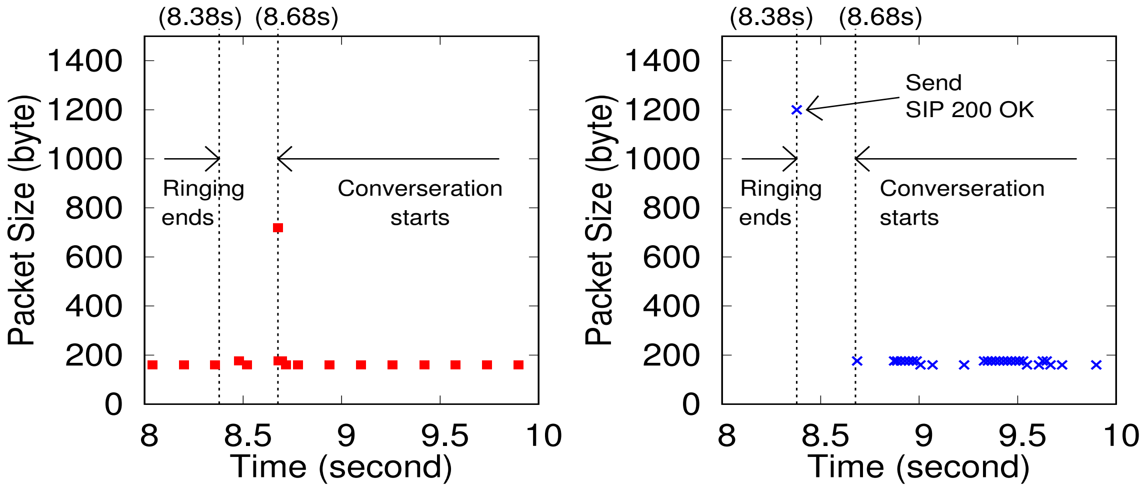
(a) Downlink (sent by the server)

(b) Uplink (sent by the callee)

Figure 3.6 The packet arrivals of the event 'receiving a call with a ringtone' on the Wi-Fi AP.

**An illustrative example: A device user receives an incoming Wi-Fi calling call and answers it around 6 seconds after its ringtone. Afterwards, the user has a voice conversation for around 12 seconds. Figure 3.5 shows the IPSec packets observed on the Wi-Fi AP to which the device connects. The following four events can be observed: (1) receiving a call with a ringtone; (2) answering a call; (3) talking; (4) hanging up a call.**

*Event 1: Receiving a call with a ringtone.* Figures 3.6a and 3.6b show the downlink and uplink packets of this event, respectively. The first incoming packet, which is intercepted at the 2nd second, is a 1360-byte IPSec packet. We decrypt it at the callee and identify it as an SIP



(a) Downlink (sent by the server)

(b) Uplink (sent by the callee)

Figure 3.7 The packet arrivals of the event ‘answering a call’ on the Wi-Fi AP.

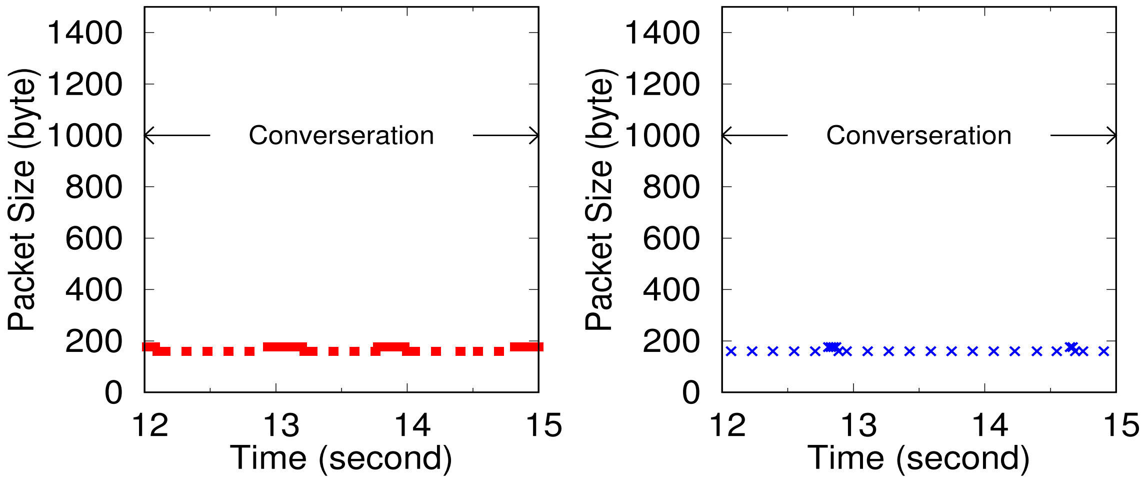
INVITE message, which indicates that a call attempt is coming. At the 2.43th second, the callee sends an 180 RINGING message to the Wi-Fi calling server. Afterwards, it is observed that several small IPSec packets with only 176 bytes are received by the callee, but the callee does not send any packets back. We discover that they are voice packets in the RTP (Real-Time Protocol) protocol.

*Event 2: Answering a call*. As shown in Figures 3.7a and 3.7b, the callee answers the call at the 8.38th second by sending a 200 OK message to the server, and then receives an acknowledgment at the 8.68th second. Afterwards, the call conversation starts and the callee begins to send/receive voice packets.

*Event 3: Talking*. The traffic pattern of this event is shown in Figure 3.8. During the call conversation, the callee keeps sending/receiving voice packets to/from the Wi-Fi calling server, but no SIP messages are observed. We further discover that the callee at least receives 10 voice packets every two seconds from the server.

*Event 4: Hanging up a call*. The callee sends a BYE message at the 20.19th second after the call is hanged up, as shown in Figure 3.9b. After the 20.32nd second, no more IPSec packets are observed. Note that if the caller hangs up the call first, the BYE message should be sent by the server.

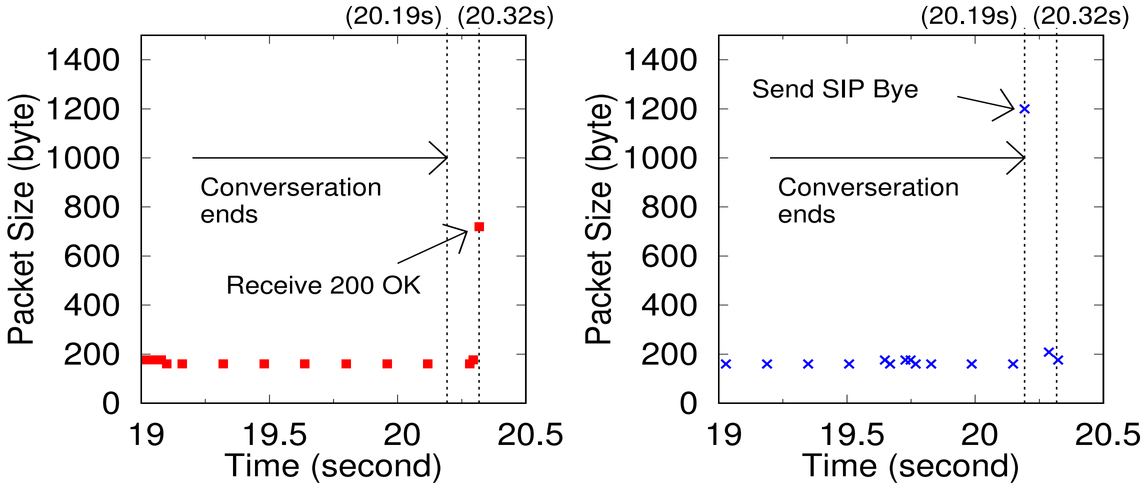
**Traffic Pattern Analysis:** We have five observations on the traffic patterns of the Wi-Fi calling



(a) Downlink (sent by the server)

(b) Uplink (sent by the callee)

Figure 3.8 The packet arrivals of the event ‘talking’ on the Wi-Fi AP.



(a) Downlink (sent by the server)

(b) Uplink (sent by the callee)

Figure 3.9 The packet arrivals of the event ‘hanging up a call’ on the Wi-Fi AP.

messages and events.

1. The sizes of the voice packets in IPSec are smaller than 200 bytes (e.g., 176 bytes).

2. The sizes of the SIP packets that contain signaling messages, including `INVITE`, `180 RINGING`, `200 OK`, and `BYE`), in IPSec are much larger than the voice packets (e.g., 800-1360 bytes).

3. The callee starts to receive the voice packets from the Wi-Fi calling server after the `180 RINGING` message is sent.

4. No voice packets are sent out by the callee before the call conversation starts.
5. The callee keeps receiving more than 10 voice packets every two seconds from the Wi-Fi calling server after the call conversation starts.

These patterns allow us to identify call events, e.g., an outgoing call is initiated, an incoming call attempt arrives, and an ongoing call ends. Moreover, by correlating them with the call flow of Wi-Fi calling (see Figure 2.2), the signaling messages of Wi-Fi calling can be identified purely based on the encrypted IPSec packets. Note that the third observation is made only from US-I and US-II; the others can be observed from all the test operators.

### 3.4.2 Attack Evaluation

We launch attacks by discarding different patterns of the signaling and voice packets for an outgoing call of Wi-Fi calling. Table 3.3 summarizes the results, which are observed on all the tested smartphones. We exploit the results to devise four attack variants as follows. Note that the damage that is caused to mobile phones may not be applied to other SIP phones (e.g., Cisco SPA 525G2).

**Annoying-Incoming-Call Attack:** The callee as the victim would receive multiple incoming calls from the caller. There are two approaches. First, the adversary drops the `183 Session Progress` message sent by the callee, and then the caller's Wi-Fi calling device would initiate another VoLTE call towards the callee. Second, the adversary discards the `180 Ringing` message sent by the callee, and then it would cause the caller's Wi-Fi calling device to get stuck in the dialing screen. The caller does not hear any alerting tone, but the callee's device would ring. The caller may thus keep redialling.

**Zombie-Call Attack:** The caller's device can be forced to get stuck in the dialing screen, when the adversary discards the `200 OK` message sent by the callee. The message indicates that the call has been answered, so without receiving the message, the caller's device gets stuck in the dialing screen and keeps hearing the alerting tone. The call conversation is thus never started.

**Intermittent Mute Call Attack:** Two parties of a Wi-Fi calling call are both victims. This attack

| No. | Dropped Packets | Sender | Results |
|---|---|---|---|
| 1 | INVITE | Caller | Caller initiates a cellular-based call. |
| 2 | 100 Trying | Server | No effect. |
| 3 | 183 Session | Callee | Two outgoing calls arrive at callee. |
| 4 | PRACK | Caller | No effect. |
| 5 | 200 OK | Callee | No effect. |
| 6 | 180 Ringing | Callee | Caller will not enter the conservation state. The caller phone gets stuck in the dialing screen. |
| 7 | PRACK | Caller | No effect. |
| 8 | 200 OK | Callee | Caller keeps hearing the alerting tone. |
| 9 | 200 OK | Callee | Caller keeps hearing the alerting tone. |
| 10 | ACK | Caller | No effect. |
| 11 | Voice Packets | Caller/ Callee | Call drops or voice quality downgrades. |
| 12 | BYE | Caller | Callee gets stuck in the conversation state for 20 s. Afterwards, the call is terminated. |
| 13 | 200 OK | Callee | No effect. |

Table 3.3 The results obtained when we drop different patterns of the signaling and voice packets for an outgoing Wi-Fi calling call.

| Drop Rate (%) | Voice Quality |
|---|---|
| below 20% | No clear impact. |
| 40-60% | Some noises. |
| 70-90% | Conversation is hardly continued. |
| 100% | Call is terminated by the network. |

Table 3.4 Voice quality varies with the drop rate of voice packets.

does not aim to terminate the call but only mute the victims' voice for a certain time. Our result shows that the adversary can mute the call up to 8 seconds by dropping voice packets. If the voice suspension time is longer than 8 seconds, the call would be terminated by the network. To prolong the attack period, the adversary can launch a cyclical attack that drops voice packets for 7 s and skip the packets for the next 1 s to mute the call intermittently.

**Telephony Denial-of-Voice-Service Attack:** Both the caller and the callee are victims. This attack downgrades the voice quality of a Wi-Fi calling call so that the conversation is hard to be continued; meanwhile, the inter-system service continuality mechanism is not triggered. It is achieved by controlling the drop rate of the intercepted Wi-Fi calling packets to/from the victim. Table 3.4 shows the negative impact on the voice quality with different drop rates. There are four findings. First, when the drop rate is below 20%, the caller/callee users do not complain about any voice quality downgrade. Second, when the drop rate increases to 40%-60%, some of the users may notice some noises. Third, when the drop rate becomes 70%-90%, the voice call is hardly continued. Fourth, when the drop rate is 100%, the call is terminated within 8 seconds. Note that when the drop rate is below 90%, the call termination is never triggered.

### 3.4.3 Real-world Impact

The impact of the THDoS attack can be significant in practice. Our studies show that the campus Wi-Fi networks, which most U.S. universities have deployed, are the best attack surfaces for the adversary. For example, the campus Wi-Fi (MSUNet) in Michigan State University provides students, the faculty, and the staff with free Wi-Fi access. In a 2-min experiment, we discover that more than 700 devices including smartphones, tablets, and computers, connect to MSUNet. All the devices are served by the same gateway which is vulnerable to an ARP spoofing attack, so their Wi-Fi calling packets can be intercepted if there are any. Therefore, it allows the adversary to launch the THDoS attack against the Wi-Fi calling devices under the gateway. Note that MSUNet is not the only Wi-Fi infrastructure that suffers from the ARP spoofing and THDoS attacks. We find that such vulnerability also exists in the campus Wi-Fi of many other universities, such as New York University, University of California Berkeley, Northeastern University, etc.

## 3.5 User Privacy Leakage Attack

In this subchapter, we devise a proof-of-concept attack that can leak the privacy of the Wi-Fi calling users. We exploit the discovered vulnerabilities to collect call statistics (e.g., call duration and number of dialing calls) for each Wi-Fi calling device with a specific IP address in an area, while using the nearby cameras to identify the person behaviors related to phone usages. By considering two information sources together, a device's call statistics can be correlated with a

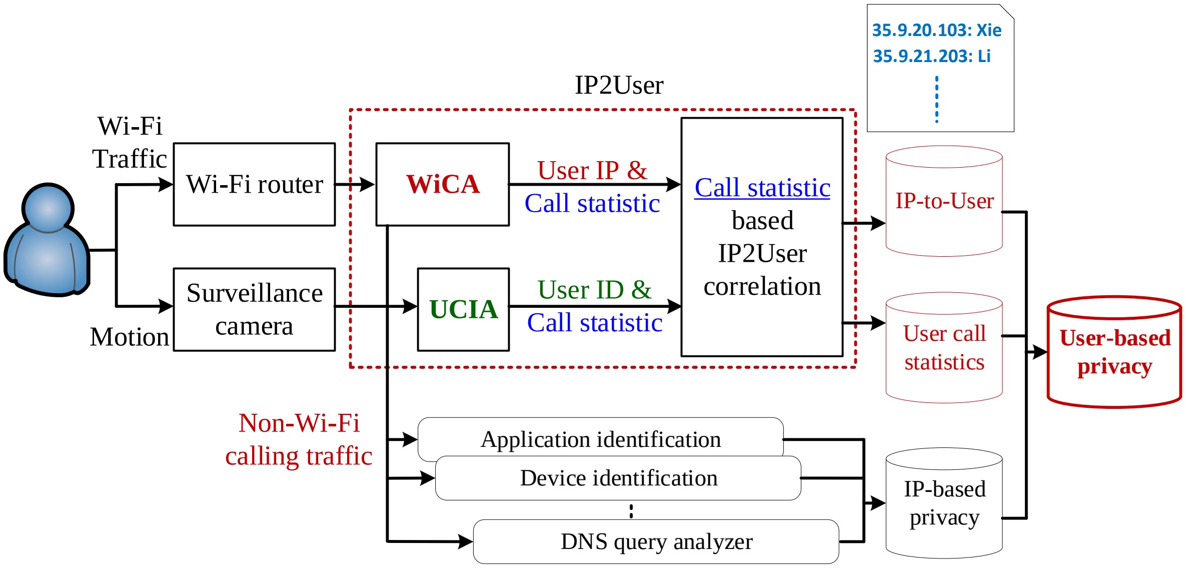


Figure 3.10 The UPIS system that infers user privacy of the Wi-Fi calling users.

person's behavior. For example, a device with 5-second call duration can be correlated with a person who holds his/her phone and speaks for 5 seconds. Based on such correlation, the adversary can obtain the IP address of a specific Wi-Fi calling user and then identify the user's packets. The adversary can thus inspect the packets to infer the user's privacy, including device activities (e.g., accessing gmail), device information (e.g., iPhone 7), running applications (e.g., WeChat), etc. In addition, several prior studies have demonstrated that the call statistics can be exploited to infer some user privacy information including mood (e.g., stressful [53]), personality (e.g., conscientiousness [54]), malicious behaviors (e.g., dialing spamming calls) [55], to name a few.

### 3.5.1 Overview of Attack Design

We launch this attack by developing a user privacy inference system, called UPIS, as shown in Figure 3.10. It consists of three major components: WiCA (Wi-Fi Calling Analyzer), UCIA (User Call and ID Analyzer), and CS-IP2U (Call Statistics based IP-to-User correlation) modules. WiCA intercepts all the Wi-Fi packets and then identifies the Wi-Fi calling ones. From the Wi-Fi calling packets, WiCA extracts call statistics (e.g., ringing time and call duration) for each device IP. The other packets are dispatched to a real-time traffic analyzer, which analyzes application identity and device information. UCIA identifies each phone user's call statistics based on a surveillance camera using the techniques of face recognition and human motion detection. CS-IP2U uses the call statistics from both WiCA and UCIA to correlate each phone user with an IP address. It generates a mapping table with IP and user identity, together with each user's call statistics. We next elaborate on the WiCA, UCIA, and CS-IP2U components, and finally evaluate the UPIS system.

### 3.5.2 WiCA: Wi-Fi Calling Analyzer

WiCA infers call statistics on a per-IP basis by analyzing the Wi-Fi calling traffic. Unlike the aforementioned THDoS attack where specific signaling messages of Wi-Fi calling need to be accurately identified, WiCA considers the extraction of only call statistics. Thus, it requires a relatively simple approach that consumes little resources. Figure 3.11 illustrates its finite state machine, where the initial state is IDLE. It works as follows.

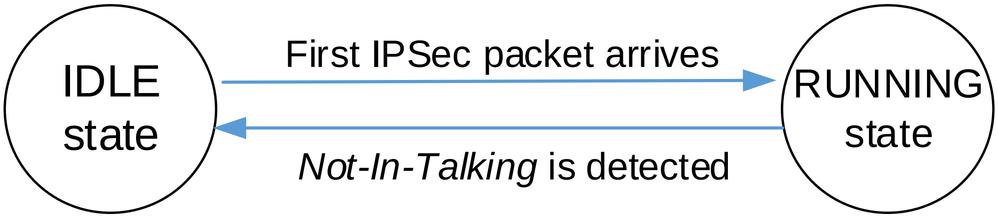


Figure 3.11 The state transition diagram of WiCA.

**Step 1:** At the initial IDLE state, whenever any IPSec packet belonging to a call event is received, WiCA moves to the RUNNING state. WiCA determines that kind of IPSec packets by checking whether they are sent to/from any Wi-Fi calling servers. WiCA records the forwarding direction to differentiate between two events, namely *dialing a call* and *receiving a call*. Their IPSec packets are sent to and from the servers, respectively.

**Step 2:** At the RUNNING state, WiCA uses a 2-second time window to group the collected IPSec packets and classifies them into three categories: *C-Large*, *C-Middle*, and *C-Small*. They include the packets with the sizes larger than 800 bytes, between 200 and 800 bytes, and smaller than 200 bytes, respectively. The *C-Large* category includes some critical SIP call messages (e.g., INVITE and RINGING), whereas the *C-Small* contains voice packets. Note that the 2-second packet collection is denoted as $Data_{2sec}(x)$, where $x$ is the sequence of a series of the 2-second collection sets.

| Conditions | | Identified Scenarios |
|---|---|---|
| $Num\_UL\_C_{Small}$ | $Num\_DL\_C_{Small}$ | |
| =0 | >10 | Ringing |
| >10 | >10 | Talking |
| =0 | =0 | Not in Talking |

Table 3.5 $Num\_UL\_C_{Small}$ and $Num\_DL\_C_{Small}$, which respectively represent numbers of uplink and downlink packets smaller than 200 bytes within 2 seconds, are used to determine *Ringing*, *Talking*, *Not in Talking* scenarios for US-I, US-II, US-III. Note that the rule of determining ringing event is only applicable to US-I and US-II but not to US-III, since which US-III does not send small voice packets to the Wi-Fi calling callee when his/her phone is ringing.

**Step 3:** WiCA identifies three scenarios, namely *Ringing*, *Talking* and *Not in Talking*, based on the number of uplink and downlink *C-Small* packets in $Data_{2sec}(x)$, which are denoted as $Num\_UL\_C_{Small}$ and $Num\_DL\_C_{Small}$, respectively. The rules are summarized in Table 3.5. When no event is identified in a collection set, $Data_{2sec}(x)$, it is buffered and WiCA moves back to `Step 2`. When any event is identified, WiCA takes subsequent actions for the event in the following.

- *Ringing:* WiCA revisits the last collection, $Data_{2sec}(x-1)$, and looks for the time when the last *C-Large* IPSec packet is captured, which is considered as when the ring starts. We denote the time as $T_{RingingStart}$.
- *Talking:* When no *Talking* scenarios are identified before this scenario, WiCA revisits the last collection, $Data_{2sec}(x-1)$, and finds the time when the first *C-Large* IPSec packet (i.e., `SIP 200 OK`, which indicates the event 'answering the call') is captured. This time, denoted as $T_{TalkingStart}$, is considered as the time when the talk starts.
- *Not In Talking:* WiCA revisits the last collection, $Data_{2sec}(x-1)$, to discover the time when the first *C-Large* IPSec packet (i.e., `SIP BYE`) is captured. This time, denoted as $T_{CallEnd}$, is considered as the time when the call ends. When the *C-Large* packet is sent by the Wi-Fi calling device, WiCA infers that the device user hangs up the call first. Otherwise, the other call end terminates the call first. When the call termination is observed, a pattern

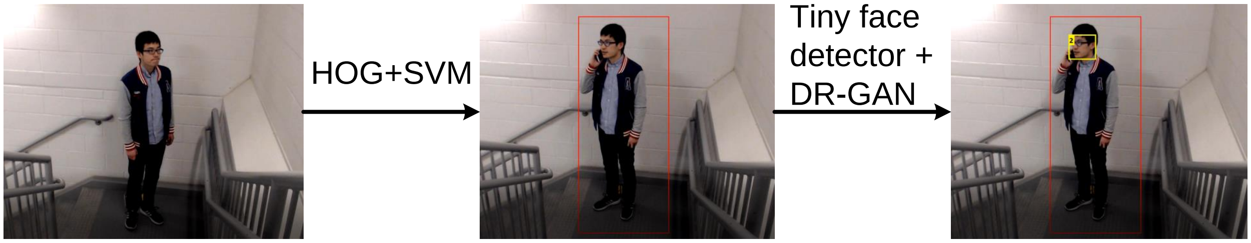


Figure 3.12 The UCIA working flowchart. The red bounding box denotes a detected calling/talking motion, whereas the yellow bounding box denotes a detected user face.

analyzer outputs a set of information including the call end initiating the call, ringing duration (i.e., $T_{TalkingStart} - T_{RingingStart}$ or $T_{CallEnd} - T_{RingingStart}$), talking duration (i.e., $T_{CallStop}$-$T_{TalkingStart}$), and the call end terminating the call. Afterwards, WiCA returns to the `IDLE` state. Note that the talking duration is not applicable to unestablished calls.

### 3.5.3 UCIA: User Call and ID Analyzer

UCIA is a visual recognition system which identifies users and their motions related to making calls (e.g., a user moves a phone close to his/her ear). It mainly leverages four computer vision techniques including a tiny face detector, which is designed to find small faces in a video, DR-GAN (Disentangled Representation learning-Generative Adversarial Network), HOG (Histogram of Oriented Gradient) [56], and SVM (Support Vector Machine). UCIA does not require the users to be still or use a high-resolution video. It can support the video in which face resolutions are as low as 25x10 [57].

Figure 3.12 illustrates the UCIA working flowchart, which analyzes videos on a per-frame basis. It consists of two modules: (1) calling/talking motion detection and (2) user face detection and recognition. In each video frame, UCIA uses the HOG and SVM models to detect calling/talking motions for all users, and labels those whose motions are detected using red bounding boxes. For each red bounding box, UCIA further uses the tiny face detector and the DR-GAN model to label the user face with a yellow bounding box, and identifies his/her identities (i.e., names). We next detail these two modules and then evaluate the performance of UCIA.

#### 3.5.3.1 Calling/talking Motion Detection

UCIA generates features of target motions using the HOG descriptor, and then classify them with an SVM model.

**SVM:** We train the SVM model to recognize two motions, namely ‘dialing a call’ and ‘talking in a call’. Since no video datasets contain them, we invite twenty volunteers to record videos of their dialing/talking motions. To differentiate those two motions from the others, we do the model training by mixing the recorded videos with those from 101 action categories in the UCF101 database [58].

**HOG:** Each frame in a surveillance video may contain many candidate bounding boxes within a sliding window. After all the persons are marked by the bounding boxes, the pre-trained SVM classifier determines whether any of those two motions happens in each bounding box based on the change of the gradient information described by the HOG descriptor. To implement the HOG descriptor, we first divide each image into different small connected components, called cells, and then collect the orientation histogram of gradients for each pixel within each cell. Finally, we concatenate all the histograms to be the HOG descriptor.

#### 3.5.3.2 User Face Detection and Recognition

We adopt a tiny face detector [57], which is based on the technique of deep convolution neural network (CNN), to detect user faces, since not all the surveillance cameras offer high video quality (e.g., 1080p). The detector is designed to detect small faces (e.g., a face with the size of $3 \times 3cm^2$) in a low-resolution video, but can also support large faces in a high-resolution video. Moreover, since people do not always face to the cameras with a frontal view, extracting pose-invariant feature representations is critical to the face recognition. We thus apply DR-GAN [59] that can generate those representations to recognizing user identities.

**Tiny face detector:** The working flowchart of this detector is illustrated in Figure 3.13. The detector first resizes each input image into other two images with different resolutions to construct an image pyramid for the training. It uses those three images with different resolutions as the input of the CNN model. We adopt a well-trained model provided by Hu et al. [57]. The trained model

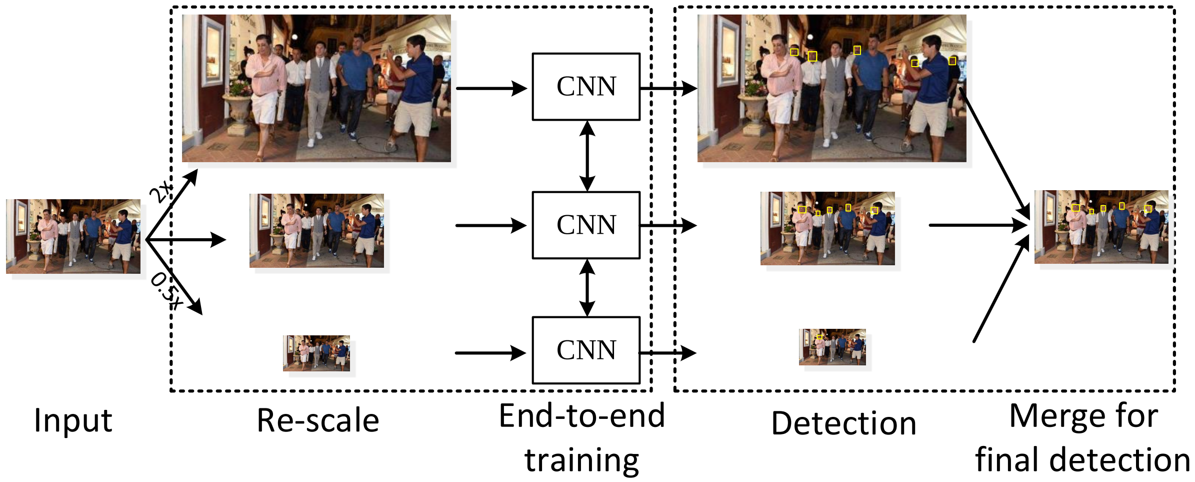


Figure 3.13 The working flowchart of the tiny face detector.

can be used to predict the bounding boxes on the image pyramid. All the detected bounding boxes are then selected and merged based on the non-maximum suppression (NMS) method [60], and then the final detection result on the original image can be obtained.

**DR-GAN-based face recognition:** To recognize user identities, it is challenging to deal with variations on the user faces (e.g., illumination conditions, poses, and expressions); especially, the pose changes can cause a big drop on the face recognition performance. We tackle this challenge by applying the DR-GAN model in the following two steps. First, we define face angles ranging from $-90°$ to $90°$. With the $0°$ face angle, the face is in the frontal view, which almost contains all the facial information. With the angle of $-90°$ or $90°$, only one side of the face is visible so that it is difficult for the model to do face recognition. Second, we leverage the DR-GAN model to extract the disentangled face representation by fine-tuning the GAN (Generative Adversarial Networks). The model can generate a representation for each face with personal identity information and then the representation can be used for the face verification and identification.

The face recognition flowchart of the DR-GAN model is shown in Figure 3.14. To train the DR-GAN model, several face images with different poses for the same user identity are used as the input. Each image will be fed into the encoder that uses VGG16 as the network structure. In addition to generating a 320-dim feature $f$ for each face, the encoder outputs a 1-dim coefficient $w$. A fused feature $f'$ can be then generated based on the following equation:

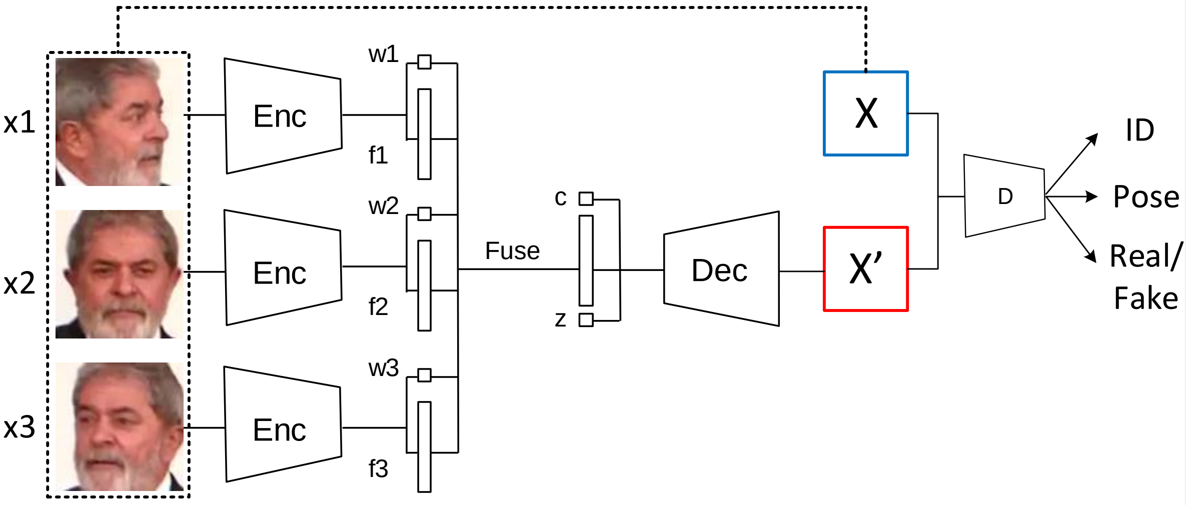


Figure 3.14 Overview of the DR-GAN model.

$$f' = \frac{\sum_i^n w_i f_i}{\sum_i^n w_i} \tag{3.1}$$

, where $f'$ is a weighed average over all the $f_i$. $f'$ can be fed into a decoder to generate an output image, called synthetic image, with the same size as the input. Accompanying the feeding of $f'$, a pose code $c$ and a random noise $z$ are also appended. The former can help the decoder generate a synthetic image with an arbitrary pose, whereas the latter can prevent the decoder from overfitting. We further use the combination of the original face images and the synthetic image to train a discriminator. After the adversarial training involving the encoder, the decoder, and the discriminator converges, an updated encoder can be derived. We finally use this trained encoder to generate the disentangled feature representations of all the input images for the face recognition.

### 3.5.4 CS-IP2U: Correlating IP with User Identity

The CS-IP2U module correlates user identities with IP addresses based on the call statistics extracted by WiCA and UCIA. It mainly considers two kinds of events, namely call start and call end. We denote the happening times of these two events as $TCStart$ and $TCEnd$, respectively. Ideally, for an identified Wi-Fi calling call, WiCA outputs $TCStart_w$, $TCEnd_w$, and $IP$, whereas UCIA outputs $TCStart_u$, $TCEnd_u$, and $UserID$. One correlation can be thus identified when $TCStart_w == TCStart_u$ and $TCEnd_w == TCEnd_u$. Nevertheless, in practice, it is not the case due to the errors of recorded timing in the call statistics. CS-IP2U thus considers not only time points but also time intervals in the correlation with the following three steps.

**Step 1:** We consider two time intervals, $TCStartInt_w = [TCStart_w - \sigma, TCStart_w + \sigma]$ and $TCEndInt_w = [TCEnd_w - \sigma, TCEnd_w + \sigma]$, for the call start and end events in WiCA, respectively. $\sigma$ is set to the maximum timing error observed in WiCA (i.e., 1 second).

**Step 2:** We further consider the other two time intervals, $TCStartInt_u = [TCStart_u - \epsilon, TCStart_u + \epsilon]$ and $TCEndInt_u = [TCEnd_u - \epsilon, TCEnd_u + \epsilon]$ for the call start and end events in UCIA, respectively. $\epsilon$ is set to the maximum timing error observed in UCIA (i.e., 1.5 seconds).

**Step 3:** When the following two conditions are met, the corresponding $IP$ and $UserID$ are correlated: $TCStartInt_w \bigcap TCStartInt_u \neq \emptyset$ and $TCEndInt_w \bigcap TCEndInt_u \neq \emptyset$.

Note that current CS-IP2U implementation does not support the cases that multiple Wi-Fi calling users start or end calls near-simultaneously (within the time interval of $max\{\sigma, \epsilon\}$ (i.e., 1.5 seconds). To address this issue, more fine-grained call statistics should be extracted by the WiCA and UCIA modules. For example, we can infer time periods that users are talking and those that user are listening by analyzing the uplink and downlink Wi-Fi calling voice packets at the WiCA and detecting who are talking [61] at the UCIA. We do not implement this advanced feature on our attack prototype, but only demonstrate the feasibility of the correlation between user identities and IP addresses.

### 3.5.5 Attack Evaluation

We next evaluate the performance of the UPIS system in a controlled setting (in our laboratory without passersby) and a wild setting (in an on-campus coffee shop with passersby). The WiCA is implemented using Python3 and the scikit-learn library [62] on a 2014 Macbook Pro laptop with a CPU, Intel I5-4278U, and an 8GB RAM. The UCIA is implemented using Python3 and other three computer vision and machine learning libraries, namely VLFeat [63], MatConvNet, and Tensorflow, on our campus computing servers (MSU HPCC) [64]. The CS-IP2U is also implemented on the Macbook laptop. Moreover, the CS-IP2U requires to associate time and events between the WiCA and the surveillance camera, so the clock synchronization between them is needed. The precision time protocol (PTP) [65] can be used for the synchronization.

| Module | Performance Metrics | | Controlled settings | | | | Wild settings | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | User1 | User2 | User3 | User4 | User1 | User2 | User3 | User4 |
| **WiCA** | **Call Event Time Estimation** | start time error (sec) | 0.25 | 0.55 | 0.15 | 0.08 | 0.33 | 0.48 | 0.22 | 0.15 |
| | | end time error (sec) | 0.17 | 0.23 | 0.37 | 0.10 | 0.27 | 0.38 | 0.54 | 0.31 |
| **UICA** | **Calling Motion Recognition** | ACC | 94.9% | 90.1% | 90.0% | 85.0% | 88.3% | 85.7% | 85.0% | 80.1% |
| | | FPR | 3.4% | 8.3% | 6.8% | 9.1% | 12.7% | 12.1% | 13.1% | 18.1% |
| | | FNR | 7.3% | 12.5% | 12.2% | 22.3% | 10.8% | 17.6% | 17.9% | 22.04% |
| | **Call Event Time Estimation** | start time error (sec) | 1.51 | 1.34 | 0.53 | 0.98 | 1.22 | 1.34 | 1.03 | 1.28 |
| | | end time error (sec) | 0.62 | 0.99 | 0.76 | 1.19 | 1.23 | 1.55 | 0.92 | 1.26 |
| | **User Identity Recognition** | ACC | 95.8% | 98.1% | 92.5% | 93.5% | 91.3% | 93.8% | 92.0% | 90.8% |
| | | FPR | 2.8% | 1.0% | 6.6% | 7.5% | 10.2% | 8.0% | 6.2% | 5.6% |
| | | FNR | 8.3% | 5.7% | 9.9% | 8.1% | 7.6% | 7.5% | 10.0% | 10.1% |
| **CS-IP2U** | **ID and IP Mapping** | ACC | 95.0%<br>(19/20) | 100%<br>(19/19) | 100%<br>(17/17) | 94.7%<br>(18/19) | 83.3%<br>(15/18) | 84.2%<br>(16/19) | 89.4%<br>(17/19) | 90%<br>(18/20) |

Table 3.6 Overall performance of the UPIS system.

#### 3.5.5.1 Evaluation Metrics

**WiCA:** The evaluation metric is the estimation error of the call event time, which is the difference between the time when a W-Fi calling call starts or stops, and the time that is estimated for the call event by the WiCA. Note that we can use a command, `logcat -b radio -v threadtime | grep "update phone state"`, on Android phones to obtain the times of the call start and stop events, which are the ground truth in the evaluation.

**UCIA:** We evaluate UCIA from three aspects, namely calling/takling motion recognition, user identity recognition, and the estimation error of the call event time. The evaluation metrics of the first two aspects include accuracy (ACC), false positive rate (FPR), and false negative rate (FNR).

For the calling/takling motion recognition, the video frames of a user can be classified into two categories: with and without a calling event. They are considered as positive and negative cases, respectively. For the user identity recognition, UCIA analyzes all the frames that are recognized with a calling event and looks for the user identity in the event from our database. The ACC, FPR, and FNR rates are calculated on a per-user basis.

**CS-IP2U:** The evaluation metric is the ratio of the accurate cases that the identity of the Wi-Fi calling user is correctly correlated with the user's device IP, to all the user's Wi-Fi calling calls.

#### 3.5.5.2 Experimental Results

We evaluate the performance of Wi-Fi calling user privacy inference system (UPIS) in the non-wild and wild settings as follows.

•**Using non-wild settings (without passersby):** The experiment is conducted in a on-campus space where there are no passersby but the experiment participants. We consider four participants in the experiment. In each test, each of them is requested to dial at least one call; they are allowed to do any random actions (e.g., looking at the ground). To emulate a real use scenario, we do not restrict the duration of each Wi-Fi calling call. The experiment includes 10 tests, and 10 videos are recorded.

The experimental result is summarized in Table 3.6. In the WiCA module, the errors of the call event time estimation are limited to at most 0.55 s. As for the UICA module, the ACC/FPR/FNR rates of the motion recognition are 85%~94.9%, 3.4%~9.1%, and 7.3%~22.3%, respectively; those of the identity recognition are 92.5%~98.1%, 1.0%~7.5%, and 5.7%~9.9%, respectively; the errors of the time estimation range between 0.53 s and 1.51 s. Although the identify recognition mechanism does not correctly recognize user identity in all the video frames, the 100% accuracy is not needed in practice. The reason is that the successful recognition of a Wi-Fi calling user requires only one video frame of the user. Lastly, the overall performance of the UPIS system is 97.33% (73/75) by considering the accuracy of the CS-IP2U module.

•**Using wild settings (with passersby):** We conduct the above experiment in an on-campus coffee shop where has not only experiment participants but also other customers. We compare the results of the wild experiment, which is also summarized in Table 3.6, with that of the controlled one. For the WiCA module, the performance is comparable to that of the controlled experiment. In the UICA module, the ACC/FPR/FNR rates of the motion recognition decrease to 80.1%~88.3%, increase to 12.1%~18.1%, and increase to 10.8%~22.04%, respectively. This downgrade performance hurts the accuracy of the call event time estimation; thus, the combined error of the start and end times increases from 2.33 seconds in the controlled experiment to 2.89 seconds. The similar trends are also observed in the identity recognition; its ACC/FPR/FNR rates decrease to 90.8%~93.8%, increase to 5.6%~10.2%, and increase to 7.5%~10.0%, respectively. As expected, the overall performance is reduced to 87% (66/76). The reason is that the unexpected passersby can affect the performance of the motion and identity recognition mechanisms. We leave the further improvement to our future work.

### 3.5.6 Real-world Impact

To the best of our knowledge, the UPIS is the first system which can correlate the identity of the Wi-Fi calling user with the user's device IP based on the call statistics of the Wi-Fi calling service. Seemingly, it needs a little strong assumption that the victims are in the visible area of a surveillance camera that can be accessed by the adversary and a face recognition technique can be applied. However, for the sake of public safety, such surveillance cameras with face recognition have been broadly deployed in several countries, e.g., United Kingdom [66], China [67], and U.S. (Chicago and Detroit) [68]. We thus believe that some use scenarios can benefit from the UPIS system in practice. For example, the UPIS can be deployed at airports to be against terrorists. It allows the law enforcement agents to identify suspects' phone models and IP addresses, and further remotely install the malware on their phones for monitoring. The remote installation can be achieved by exploiting public security vulnerabilities of the target devices. Note that we do not advocate any use scenarios compromising user privacy no matter whether the purpose is benign or not.

## 3.6 Solution

To completely address all the identified vulnerabilities, it is required to modify current Wi-Fi calling standard; the standard modification is too time consuming to be achieved in a short time. We thus propose a software-based security framework, *Wi-Fi Calling Guardian*, to largely mitigate the impact of the vulnerabilities without any modifications on the standard but only a phone-side software upgrade. In the following, we present the design and evaluate its performance.

### 3.6.1 Design

The architecture of Wi-Fi Calling Guardian consists of two network elements, namely the client on the Wi-Fi calling device and the server in a secure private network, as shown in Figure 3.15. There are mainly three security modules on the client and the server: (1) *Wi-Fi security examiner*, which examines whether the connected Wi-Fi network is secure for the Wi-Fi calling service; (2) *singularity rectifier*, which introduces noises to mix with the Wi-Fi calling traffic, thereby increasing the difficulty of the inference; (3) *service quality monitor*, which monitors whether the

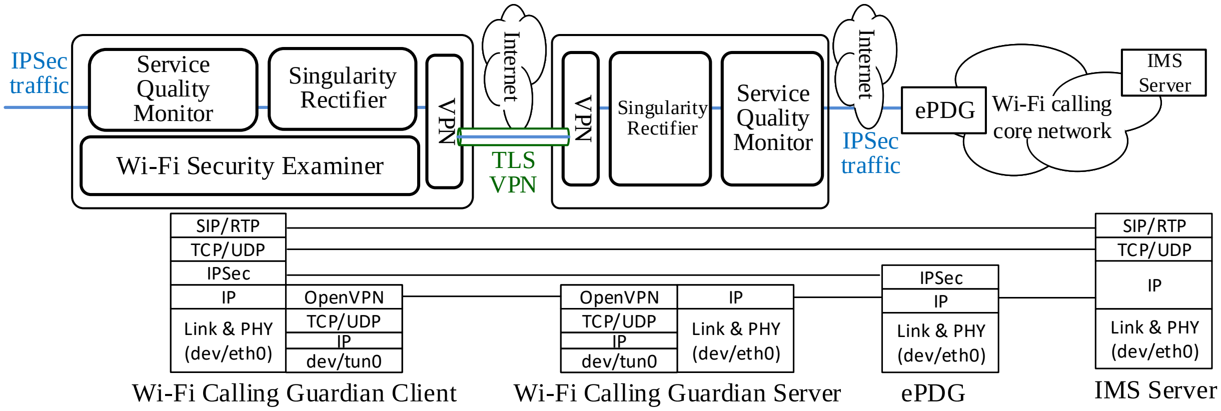


Figure 3.15 The network architecture and protocol stack of Wi-Fi Calling Guardian.

Wi-Fi calling user is suffering from the degradation of the service quality and then takes actions if needed.

Ideally, the Wi-Fi security examiner can help the Wi-Fi calling device stay away from insecure Wi-Fi networks, which are vulnerable to any known attacks (e.g., the ARP spoofing attack). However, the situation is far from simple in practice due to three reasons. First, not all the vulnerabilities can be identified using a passive approach in which the examiner operates (e.g., using detection only not launching attacks). Second, the Wi-Fi calling user may have no secure Wi-Fi networks to associate with. Third, the proposed attacks (e.g., the THDoS and user privacy leakage attacks) can be launched outside of the connected Wi-Fi network. Therefore, the Wi-Fi security examiner uses a passive approach to explore the insecurity of the connected Wi-Fi network on one hand; on the other hand, the other security modules, *singularity rectifier* and *service quality monitor*, protect the Wi-Fi calling device against potential attacks. We next elaborate on the details of these three security modules.

**Wi-Fi security examiner:** Two detection mechanisms are deployed to examine the insecurity of the connected Wi-Fi network. First, this module detects whether the WPA3 protocol [69] is enabled in the connected Wi-Fi network. It is because the WPA3 requires all the compliant devices to support the PMF (Protected Management Frames) feature, which provides integrity protection over management frames and can thus defend against some Wi-Fi attacks (e.g., deauthentication and rogue AP attacks). Second, this module detects whether the Wi-Fi calling device is being under an

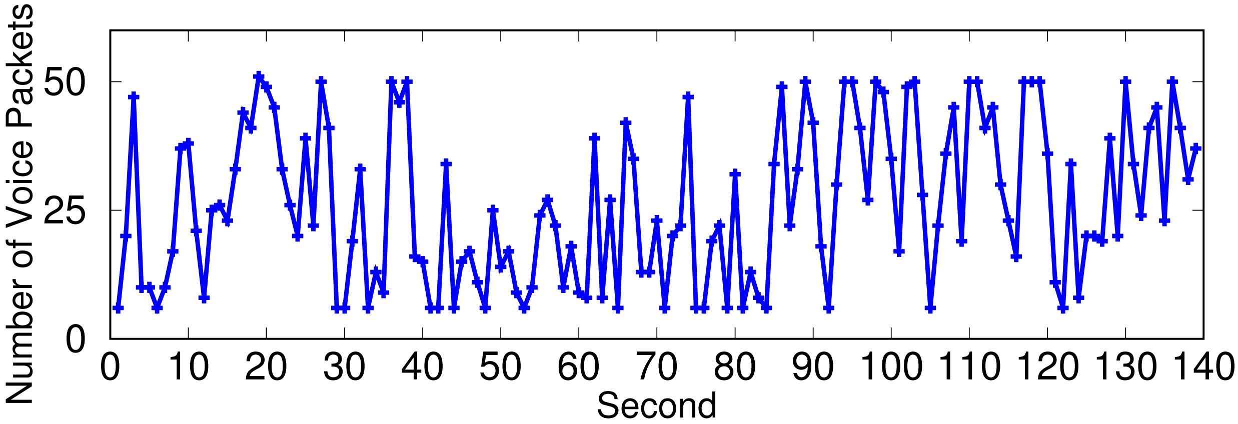


Figure 3.16 The voice packets sent from the phone per second.

ARP spoofing attack, which is a prerequisite of various MitM attacks, so that V1 can be prevented. It monitors the device's ARP table and checks whether two different IP addresses associate with the same MAC address.

**Singularity rectifier:** This module uses a *normalized data transmission* mechanism to prevent the Wi-Fi calling service from appearing as a singular service supported by the IPSec channel. The mechanism encapsulates all the Wi-Fi calling packets into UDP datagrams for the delivery. The UDP datagrams with a fixed packet size (e.g., 300 bytes) are generated by both the client and the server, and sent to the other end at a steady rate. This approach can remove two traffic patterns of Wi-Fi calling, namely *packet sizes* and *delivery directions*, at a low cost (e.g., consuming only the bandwidth of 0.032 MB/s while the rate is 50 pkts/s) so that V2 can be eliminated.

**Service quality monitor:** This module provides the Wi-Fi calling device with the inter-system service continuity mechanism driven by the service quality instead of the radio quality or the WLAN performance. V3 can be thus prevented. We estimate the voice quality based on the number of received voice packets per second on the device. Figure 3.16 plots the number of voice packets for a 140-second voice call. Since the Wi-Fi calling voice service uses the AMR (Adaptive Multi-Rate) audio codec, the packet rate varies with time. However, we observe that the packet rate is never smaller than 10 packets every two seconds. This rate can be thus used to detect whether the device is being under service degradation attacks. Once any suspicious attack is detected at the client or the server, the inter-system service continuity mechanism is triggered.

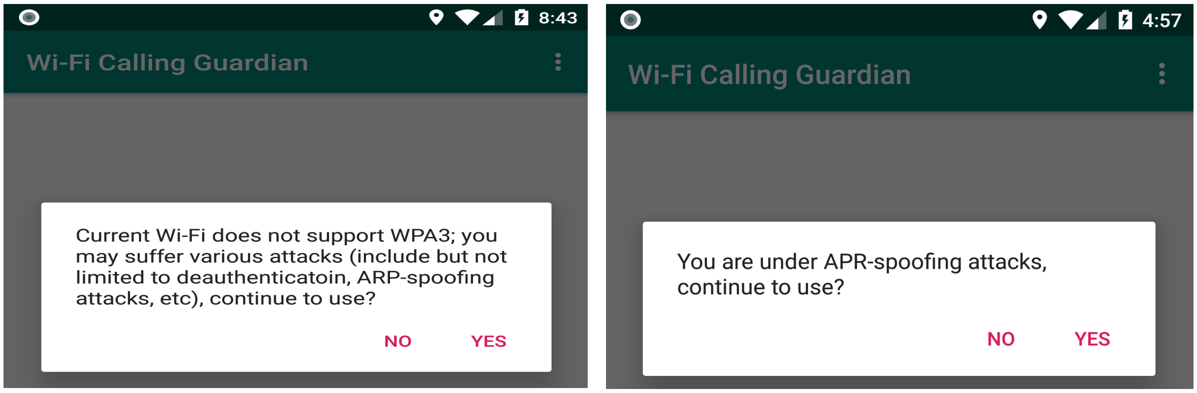
### 3.6.2 Implementation

The client of Wi-Fi Calling Guardian is an Android application written in Java and implemented on a Google Pixel XL with a CPU, Qualcomm Snapdragon 821, and a 4GB RAM; the server is a network program written in C++ and implemented on a Dell precision tower 5810 with a CPU, E5-1603, and an 8GB RAM. We next elaborate on the implementation of each key component.

**Wi-Fi security examiner:** This module is implemented in the client with two detection mechanisms. First, the module uses an Android class of `WiFiManager` to obtain the Wi-Fi connection status that indicates whether the WPA3 is enabled. Second, the module uses a command "`arp -a`" to access the ARP table of the client device, and then detects the ARP spoofing attack by checking whether any two entries share the same MAC address.

**VPN:** We use OpenVPN to set up the VPN tunnel between the client and the server. On the client side, only Wi-Fi calling packets are forwarded through the VPN connection, whereas the other packets are directly routed to their destinations. Since the Android system does not allow the OpenVPN client to redirect the packets from a system application (i.e., the Wi-Fi calling application), we deploy the OpenVPN client on a software-based Wi-Fi router to which the client device connects. Through the VPN tunnel, all the uplink packets of Wi-Fi calling are delivered to the service quality monitor on the server, whereas all the downlink packets of Wi-Fi calling are forwarded to the singularity rectifier on the client.

**Singularity rectifier:** Data padding or packet fragmentation is performed on each Wi-Fi calling packet so that the packets can be encapsulated into 300-byte UDP datagrams. This module is implemented using the Type-length-value encoding scheme for the packets. Specifically, five types of the UDP payload are developed: (1) *signaling-packet*, which specifies the start and stop of the normalized data transmission; (2) *original-packet*, which contains a complete IPSec packet; (3) *fragment-packet*, which contains a complete IPSec header, a fragment of an IPSec packet, and the fragment's sequence number; (4) *padding-data*, which contains padding data; (5) *inter-system-switch-request*, which carries an inter-system switch request for the Wi-Fi calling service. After the IPSec packets are restored from the UDP datagrams, they are forwarded to the service quality



Current Wi-Fi does not support WPA3; you may suffer various attacks (include but not limited to deauthenticatoin, ARP-spoofing attacks, etc), continue to use?

You are under APR-spoofing attacks, continue to use?

(a) WPA3 is not detected

(b) Under an ARP spoofing attack

Figure 3.17 The Wi-Fi security examiner detects the WPA3 usage and any ongoing ARP spoofing attack.

monitor.

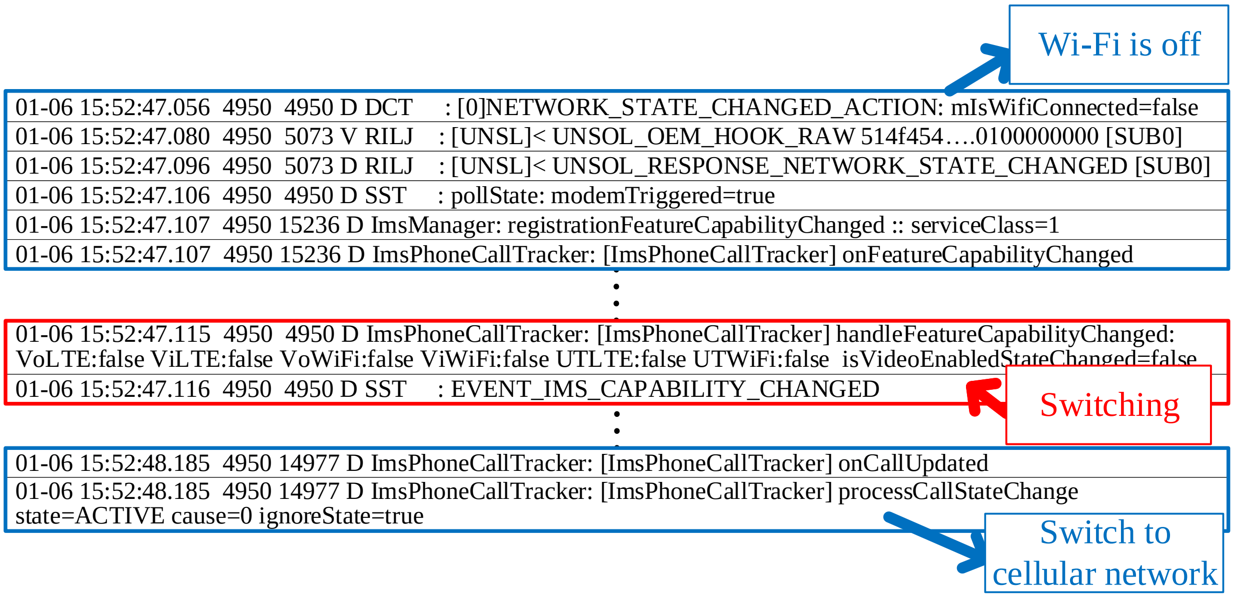
**Service quality monitor:** When the number of received small Wi-Fi calling packets is smaller than 10 during two seconds or a request of the inter-system switch is received, this module triggers the inter-system switch by disabling the device's Wi-Fi interface via an Android class of `WiFiManager`. Without the Wi-Fi access, the Wi-Fi calling device can be automatically switched back to the cellular network.

### 3.6.3 Evaluation

We next evaluate the performance of those three key components and present a small-scale user study.

**Wi-Fi security examiner:** We deploy a test Wi-Fi network which does not support the WPA3 protocol, and make the smartphone of Google Pixel XL connect with the Wi-Fi network. We further launch an ARP spoofing attack against all the devices from a computer in the Wi-Fi network. Figure 3.17 shows the evidence that the client of Wi-Fi Calling Guardian on the smartphone can successfully detect a lack of WPA3 and the ARP spoofing attack.

**Singularity rectifier:** We evaluate whether the singularity rectifier can defend against the THDoS and user privacy leakage attacks. The experiment is conducted as follows. First, we dial a Wi-Fi calling call from one device to another device with the client of Wi-Fi Calling Guardian, where the singularity rectifier is enabled. Second, we launch the annoying-incoming-call attack that discards



```
01-06 15:52:47.056  4950  4950 D DCT     : [0]NETWORK_STATE_CHANGED_ACTION: mIsWifiConnected=false
01-06 15:52:47.080  4950  5073 V RILJ    : [UNSL]< UNSOL_OEM_HOOK_RAW 514f454….0100000000 [SUB0]
01-06 15:52:47.096  4950  5073 D RILJ    : [UNSL]< UNSOL_RESPONSE_NETWORK_STATE_CHANGED [SUB0]
01-06 15:52:47.106  4950  4950 D SST     : pollState: modemTriggered=true
01-06 15:52:47.107  4950 15236 D ImsManager: registrationFeatureCapabilityChanged :: serviceClass=1
01-06 15:52:47.107  4950 15236 D ImsPhoneCallTracker: [ImsPhoneCallTracker] onFeatureCapabilityChanged
                                        :
01-06 15:52:47.115  4950  4950 D ImsPhoneCallTracker: [ImsPhoneCallTracker] handleFeatureCapabilityChanged:
VoLTE:false ViLTE:false VoWiFi:false ViWiFi:false UTLTE:false UTWiFi:false  isVideoEnabledStateChanged=false
01-06 15:52:47.116  4950  4950 D SST     : EVENT_IMS_CAPABILITY_CHANGED
                                        :
01-06 15:52:48.185  4950 14977 D ImsPhoneCallTracker: [ImsPhoneCallTracker] onCallUpdated
01-06 15:52:48.185  4950 14977 D ImsPhoneCallTracker: [ImsPhoneCallTracker] processCallStateChange
state=ACTIVE cause=0 ignoreState=true
```

Figure 3.18 A log from the Android logcat shows that a voice call over Wi-Fi calling is switched to the cellular-based voice based on the Wi-Fi disabling.

the `180 Ringing` message and causes the caller device to get stuck in the dialing screen (see Chapter 3.4). Third, we use the WiCA module to infer the call statistics of this call from the callee's connected Wi-Fi network. Our experimental result shows that the singularity rectifier can well defend against those two attacks. Specifically, in the first attack, the `180 Ringing` message cannot be identified because no large IPSec packets (800-1360 bytes) are observed. In the second attack, $T_{RingingStart}$ and $T_{TalkingStart}$ are not identified due to a lack of the *C-Large* IPSec packets. Therefore, the ringing time and the call conversation time cannot be inferred.

**Service quality monitor:** We evaluate whether the service quality monitor can detect an attack of the service quality degradation and then initiate a inter-system switch of the Wi-Fi calling service continuity. We launch a telephony denial-of-voice-service attack which discards 70% packets of a Wi-Fi calling call against a device after the call conversation starts. Our experimental result shows that the service quality monitor can detect the service quality degradation within 2 seconds after the attack is launched, and immediately trigger the inter-system switch, which is finished within 1 second as shown in Figure 3.18.

**User study:** To examine whether the VPN-based approach can significantly downgrade the voice quality of the Wi-Fi calling calls, we invite 10 students to participate in a user study experiment. In the experiment, we dial two Wi-Fi calling calls to each participant. One call is made with enabling Wi-Fi Calling Guardian, whereas the other is performed without it. The participants should report which one's voice quality is better or they are indistinguishable. Our experimental result shows that all participants cannot distinguish VPN-enabled Wi-Fi calling calls from original Wi-Fi calling calls (they think that both types of calls are the same in terms of voice quality), which means that Wi-Fi Calling Guardian does not downgrade the voice quality to a noticeable extent.

# CHAPTER 4

# HOW CAN IOT SERVICES POSE NEW SECURITY THREATS IN OPERATIONAL CELLULAR NETWORKS?

Carriers are rolling out Internet of Things (IoT) services including various IoT devices and use scenarios. Compared with conventional non-IoT devices such as smartphones and tablets, IoT devices have limited network capabilities (e.g., low rates) and specific use scenarios (e.g., inside vehicles only). These specialized use scenarios lead to carries often offering cheaper device access fees for IoT devices. However, the aforementioned disparity of service charging between IoT and non-IoT devices may lead to security issues. In this work, we make four major contributions.

- This work conducts the first empirical security study on cellular IoT service charging over two major US carriers.
- This project discovers four security vulnerabilities summarized in Table 4.1 and analyze their root causes, which help identify two significant security threats, IoT masquerading and IoT use scenario abuse.
- In this work, three proof-of-concept attacks are devised and their real-world impact are assessed. It is determined that they can be exploited to allow adversaries to pay 43.75%-80.00% less for cellular data services.
- In this project, we analyze the challenges in addressing these vulnerabilities and develop an anti-abuse solution to mitigate attack incentives. The solution is standard-compliant and can be used immediately in practice. Our prototype and evaluation confirm its effectiveness.

## 4.1 Related Work

**Mobile Security.** Mobile security has been an active research area in recent years. Researchers mainly study the security vulnerabilities of mobile data service charging, mobile devices, mobile network infrastructure, and mobile applications/services. Some interesting findings are reported, which include the anonymization of the SIP protocol [70], design flaws of mobile operating systems (e.g., Android and iOS) [71, 72, 73], charging attacks of mobile data services [74, 75, 76, 31, 77], spam and fraudulence attacks through text and voice services [78, 79], vulnerable usage of Android Internet sockets [80], vulnerabilities of VoWiFi [81] to name a few. Most of the early research works target non-IoT devices (e.g., smartphones), as well as their mobile applications and services. However, our work focuses on cellular IoT devices instead of smartphones, tablets or other non-IoT mobile devices.

**IoT Security.** Current research studies can be categorized into three dimensions: (1) device software and hardware, (2) network protocols, and (3) security architecture. In the first dimension, a study [82] shows that an IoT botnet based on the Mirai malware [83] is able to launch a 600 Gbps traffic attack. Another work [84] presents a threat that adversaries can compromise smart meters to reduce their utility bills. Liu *et al.* [85] propose an ARM TrustZone based virtual sensing system to enable a safe, isolated environment for IoT devices. Gao *et al.* [86] develop an easy access solution for authenticated users to access the voice-based assistants. Ding *et al.* [87] discover possible physical interactions and generates all potential interaction chains across applications in IoT environment.

For the IoT network protocols, Sastry *et al.* [88] discover several security vulnerabilities and pitfalls in IEEE 802.15.4, which is designed for wireless communication among low-power IoT devices. Soltan *et al.* [89] and Herwig *et al.* [90] study the IoT botnet and analyze its attacks on power grids and peer-to-peer networks.

For the IoT security architecture, some novel security mechanisms have been proposed, e.g., data-origin authentication, integrity verification, privacy preserving, and identity-based encryption. Jia *et al.* [91] propose `ContexIoT`, a context-based permission system for IoT platforms. It provides contextual integrity [92] and implements it on the Samsung SmartThings platform. Das *et al.* [93] propose a deep-learning based classifier for IoT authentication. Harris *et al.* [94] propose to protect user data against leakage by adopting the CryptoCoP-based encryption and a unique MAC address rotation mechanism. Wang *et al.* [95] conduct an analysis of the IFTTT and enumerate the inter-rule vulnerabilities that exist within trigger-action platforms. Haddadi *et al.* [96] introduce

| Category | Vulnerability | Type | Root Cause |
|---|---|---|---|
| **Device** | V1: an IoT SIM card can be used for a non-IoT device. | Design Defect | No mutual authentication between the SIM card and the device is stipulated in cellular IoT standards. |
| **Infrastructure** | V2: the infrastructure is unable to correctly identify IoT and non-IoT devices. | Design Defect | No device authentication mechanism is stipulated in cellular IoT standards. |
|  | V3: the infrastructure does not impose any restrictions on IoT data services. | Operational Slip | Operators merely rely on the hardware constraints of IoT devices instead of imposing restrictions from the infrastructure. |
|  | V4: the infrastructure is unable to confine IoT devices to their pre-defined use scenarios. | Operational Slip/ Implementation Issue | Operators restrict the IoT use scenarios by device-based security mechanisms and constraints. However, they are not bullet-proof. |

Table 4.1 Summary of security vulnerabilities and root causes.

the SIOTOME architecture between the network edge and the ISP to defend against attacks from compromised IoT devices. Memos *et al.* [97] study the security challenges of the upcoming IoT network architecture, and media security and privacy in wireless sensor networks (WSNs) and develop an efficient algorithm for media-based surveillance systems in IoT network for smart city framework. Stergiou *et al.* [98] do the security survey of IoT and Cloud Computing and show the security challenges of the integration of IoT and Cloud Computing. Celik *et al.* [99] present a policy-based enforcement system IoTGuard for IoT, which protects users from unsafe and insecure states. Moreover, some researchers focus on improving the efficiency of the systems leveraging the blooming of IoT devices (e.g., media-based IoT devices such as security camera and senors) and cloud computing to secure our society. For example, Psannis *et al.* [100] develop an efficient algorithm for encoding advanced scalable media-based smart big data on intelligent cloud computing systems, which can efficiently process the smart big data generated by a great number of media-based IoT devices (e.g., security camera). Stergiou *et al.* [101] leverage the blooming of IoT in cloud computing to develop a new type of network for intelligent media-data transfer. Plageras [102] investigates new systems for efficiently collecting and managing sensors' data in a smart building by leveraging IoT, big data, cloud computing, and monitoring technologies. Different from them, we here focus on the security of cellular IoT devices and their charging functions in the operational 4G LTE networks.

## 4.2 Overview

We aim to explore the dark side of the emerging IoT service charging scheme and its technical support from a security perspective. Any of its vulnerabilities may cause cellular users and/or carriers to suffer monetary losses. We start from an observation that IoT devices have cheaper data plans than non-IoT devices, which can be attributed to their distinct use scenarios. For example, smartwatches are designed for simple voice/data services, and car-connected hotspots are used only inside vehicles. It appears to be reasonable, but one question arises: *are the underlying technologies sufficient to secure this differential charge?* We answer it by starting with the following questions.

**Q1.** Given different charges for the same data service of an IoT device and a smartphone, can the smartphone masquerade as the IoT device to pay less?

**Q2.** If yes, can the smartphone retain its data service quality (e.g.,, no speed downgrade)?

**Q3.** Can adversaries abuse IoT devices in unanticipated use scenarios so as to take advantage of operators?

Unfortunately, we disclose that the IoT charging, as well as the technical support behind it, is not bullet-proof. The answers to the above three questions are all yes. Specifically, we uncover four vulnerabilities from design, implementation, and network operation aspects. The cellular network standards, network operators/vendors, and device manufacturers all share the blame for these vulnerabilities. The fundamental problems are rooted not in how to charge IoT and non-IoT devices, but in how to provision and safeguard IoT services.

## 4.3 Threat Model and Methodology

**Threat model.** In this work, the adversary is a mobile user who uses only commodity devices: smartphones and IoT devices available on the market. To launch attacks, (s)he needs to either know how to install tools on smartphones and modify their settings, or rely on some one-click software/hardware package, the development of which is not our main focus. In all cases, (s)he has no access to the cellular network infrastructure or other devices. Moreover, the network infrastructure and the device hardware are not compromised. Given this model, the identified security loopholes can be translated into realistic attacks against carriers.

**Methodology.** We validate vulnerabilities and attacks in two top-tier US carriers, OP-I and OP-II. They, together, take more than 65% of market share [103] in the U.S. We conduct experiments using IoT devices including two popular smartwatch models and two car-connected hotspots, as well as non-IoT devices including four Android phone models, with the two carriers' SIM cards. The two smartwatch models are LG Watch Urbane 2nd edition with Android 6.1.1 and Samsung Gear S3 frontier with Tizen OS 2.3.2. The four phone models include Samsung Galaxy S5/S6, LG G3 and Google Nexus 6P, which run Android 4.4.4, 5.0.2/6.0.1, 4.4.2 and 7.1.1, respectively. Note that all the results can be applied to both carriers, unless explicitly stated otherwise.

**Responsible experiments.** We understand that some feasibility tests and attack evaluations might be harmful to carriers, so we proceed with this study in a responsible manner. We run experiments in fully controlled environments. We purchase plans with sufficient data/voice/text quotas, so the carriers do not get hurt. We seek to disclose new security vulnerabilities and effective attacks on cellular IoT services, but not to aggravate the damages caused by them.

## 4.4 Vulnerabilities

In this subchapter, we answer the three aforementioned questions by considering two potential threats, IoT masquerading and IoT use scenario abuse. We validate vulnerabilities and devise proof-of-concept attacks for each threat, as well as evaluate a long-term IoT attack to show real-world impact.

### 4.4.1 IoT Masquerading

We first introduce three vulnerabilities and then devise an IoT masquerading attack.

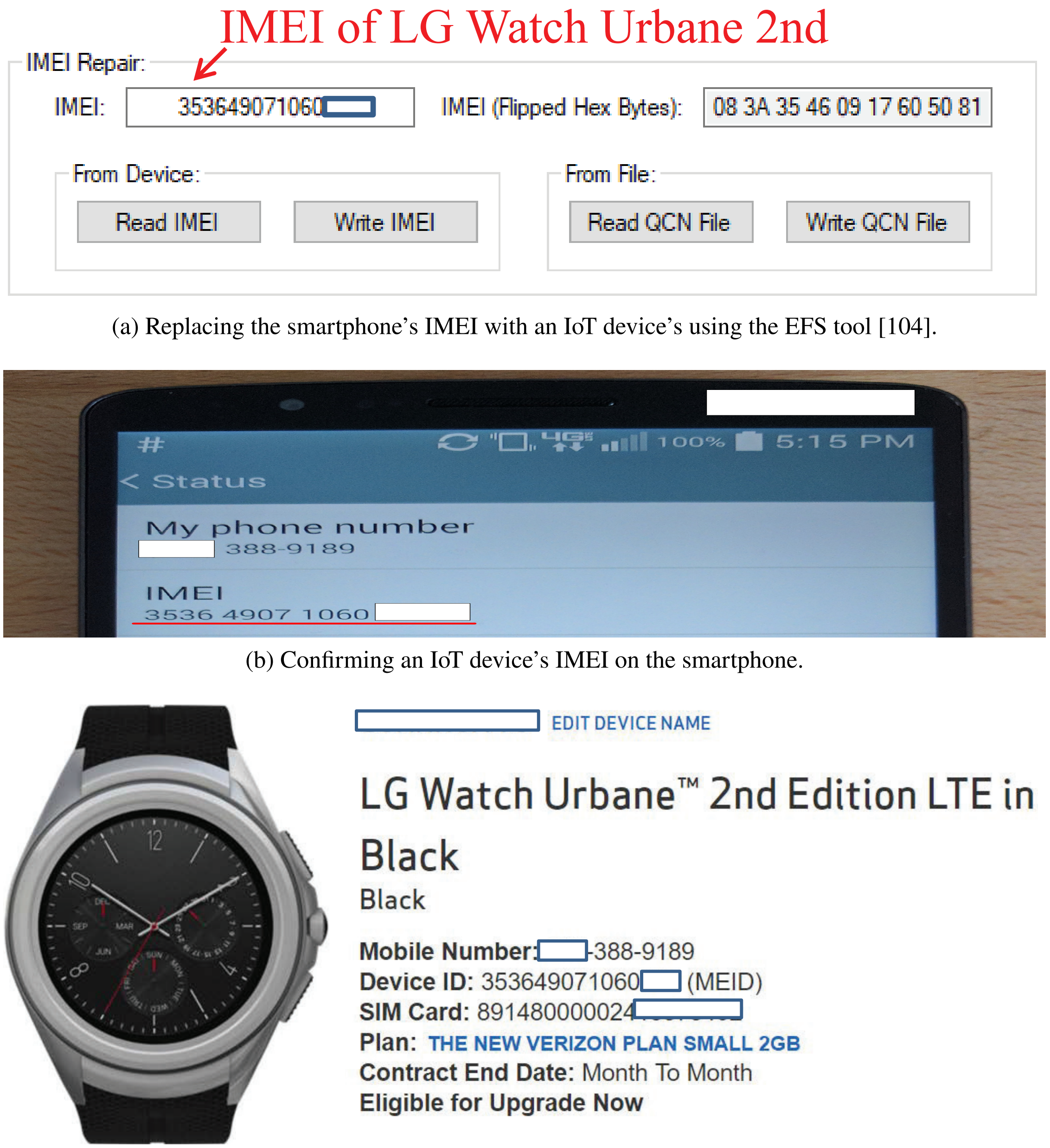
#### 4.4.1.1 Can Non-IoT Devices Masquerade as IoT Ones?

The answer is yes, due to two vulnerabilities discovered within the 3GPP security design. Each of vulnerabilities corresponds to a lack of mutual authentication between two parties. One is between IoT SIM cards and mobile devices, so an IoT SIM card can be used for a non-IoT device (V1). The other is between mobile devices and the infrastructure, so the latter is unable to correctly identify IoT and non-IoT devices (V2). These two vulnerabilities allow non-IoT devices to masquerade as IoT devices without being detected by SIM cards or the infrastructure.

The cellular authentication solely relies on the Authentication and Key Agreement (AKA) procedure [15], where users and the infrastructure are mutually authenticated. Each user is identified by his/her International Mobile Subscriber Identity (IMSI) and authenticated based on a secret key. Both the IMSI and the secret key are stored in the SIM card. However, neither the SIM card nor the infrastructure authenticates the used device; the former does not differentiate types of mobile devices in its operation, whereas the latter identifies a connected device purely based on its reported information, which may be fake without the device authentication and thus lead to wrong identification. By current design, the non-IoT/IoT data plan to which each user subscribes is bound to the IMSI or the SIM card, so the used device is not restricted by the subscribed plan. That is, an IoT SIM card, which is associated with an IoT data plan, can also work on non-IoT devices. This allows the IoT masquerade to be possible. Moreover, differential non-IoT/IoT charges, where the IoT plans are cheaper, can be a strong incentive for the masquerade.

**Validation.** We first validate V1 by showing that IoT SIM cards work for non-IoT devices. We purchase IoT SIM cards used for CAT-4, CAT-1, and CAT-M1 IoT devices. We insert each of them into our test smartphones, properly configure their networking settings, and then restart the phones. Our experimental results, collected from OP-I and OP-II, show that all the smartphones successfully obtain IP addresses from the cellular networks and access the Internet without any issues.

We next validate V2 by examining whether the infrastructure can be fooled into thinking that a smartphone is an IoT device. Initially, we discover that the OP-I and OP-II networks can correctly identify connected non-IoT or IoT devices, and show the device information on their web pages. We then analyze the control-plane protocol traces by using cellular diagnosis tools (e.g., MobileInsight [52]). It is observed that the infrastructure identifies a connected device based on the IMEI (International Mobile Equipment Identity) carried in its `IDENTITY RESPONSE` message [15].



IMEI of LG Watch Urbane 2nd

IMEI Repair:

IMEI: 353649071060[redacted]

IMEI (Flipped Hex Bytes): 08 3A 35 46 09 17 60 50 81

(a) Replacing the smartphone's IMEI with an IoT device's using the EFS tool [104].

[redacted]

My phone number
[redacted] 388-9189

IMEI
3536 4907 1060[redacted]

(b) Confirming an IoT device's IMEI on the smartphone.

[redacted] EDIT DEVICE NAME

LG Watch Urbane™ 2nd Edition LTE in Black

Black

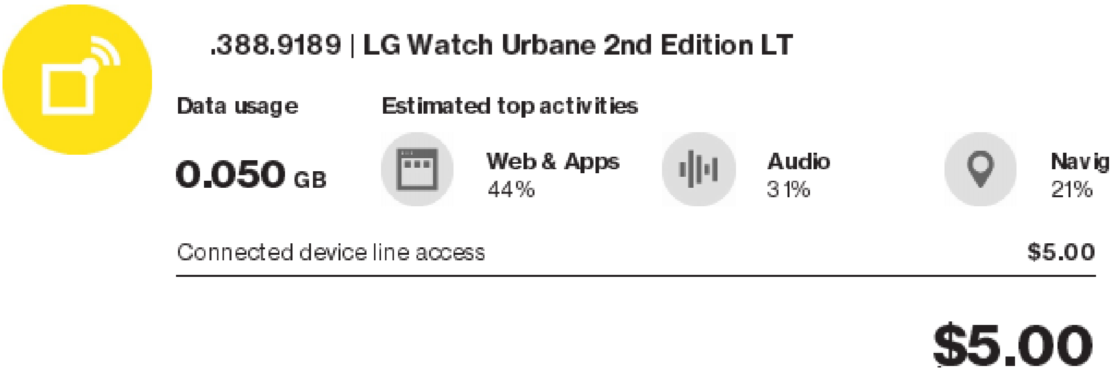
**Mobile Number:**[redacted]-388-9189
**Device ID:** 353649071060[redacted] (MEID)
**SIM Card:** 891480000024[redacted]
**Plan:** THE NEW VERIZON PLAN SMALL 2GB
**Contract End Date:** Month To Month
**Eligible for Upgrade Now**

(c) A snapshot of the OP-II's web page shows that the smartphone is recognized as an IoT device, a smartwatch.

Figure 4.1 Making IMEI spoofing on a smartphone (LG G3) to masquerade as an IoT device (LG Watch Urbane 2nd).

When connecting to the network, the device reports its IMEI in the response to the message of IDENTITY REQUEST. This implies that if the device reports a fake IMEI, the spoofing can happen.

We then investigate how to make a mobile device report a fake IMEI. The IMEI is stored in the non-volatile memory of the device modem, and the memory can be modified by some tools (e.g.,



.388.9189 | LG Watch Urbane 2nd Edition LT

Data usage

0.050 GB

Estimated top activities

Web & Apps 44%

Audio 31%

Navigation 21%

Connected device line access $5.00

$5.00

Figure 4.2 When the IMEI spoofing on the smartphone lasts for one month, OP-II still recognizes it as an IoT device, the LG smartwatch, with a $5 charge as the IoT device access fee.

EFS Professional [104]). We here show that the IMEI of a smartphone, LG G3, can be spoofed as that of an IoT device, LG Watch Urbane 2nd, in the OP-II network; the same result is also observed in OP-I. The validation consists of four steps. First, we connect to the smartphone's modem via the EFS tool [104] and replace its IMEI with the IoT device's IMEI (i.e., `353649071060XXX`), as shown in Figure 4.1a. Second, we confirm the IMEI replacement on the smartphone as shown in Figure 4.1b. Third, we reboot the smartphone to let it report the spoofed IMEI to the network. We then confirm its IMEI change on OP-II's web page, as shown in Figure 4.1c. It shows that the smartphone has been recognized as the IoT device. Last, we keep the IMEI spoofing on the smartphone for a monthly billing cycle and discover that OP-II does not detect this abuse but still charges the IoT device's access fee (e.g.,, $5), as shown in Figure 4.2. From an extended experiment with eight months (the results are elaborated on in Chapter 4.4.3), we find that the operator cannot detect the spoofing, even though several hundred megabytes of mobile data are consumed on the smartphone spoofing the IMEI of the IoT device.

**Security implications.** As new cellular IoT service charging demands arise, current security mechanisms for cellular IoT support in the 3GPP standards are not sufficient to secure carriers. We believe that addressing V1 and V2 requires revisiting these security mechanisms.

### 4.4.1.2 Any Limitations Imposed on IoT Data Services?

The answer is expected to be yes when the infrastructure offers differential data services to IoT and non-IoT devices. However, this is not the case for the tested carriers. We discover that a

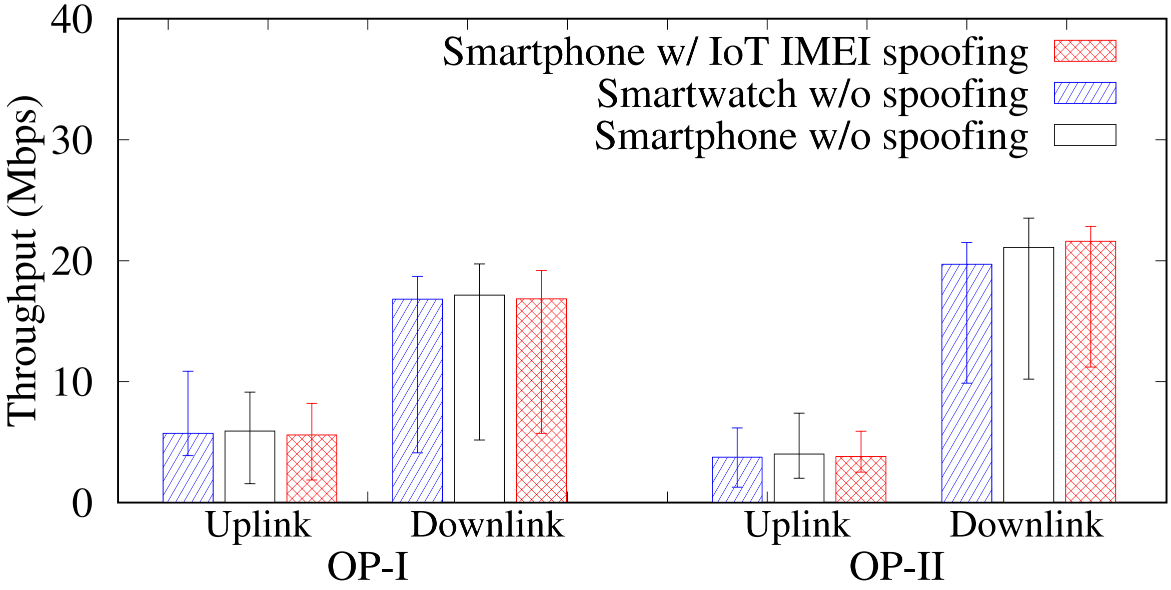


Figure 4.3 The uplink and downlink TCP throughput at the $10^{th}$, $50^{th}$, and $90^{th}$ percentiles for an IoT device (*i.e.*, LG Watch Urbane 2nd), a smartphone (*i.e.*, Samsung S5), and the smartphone with the spoofing of the IoT device's IMEI in the OP-I and OP-II networks.

non-IoT device masquerading as an IoT device can still retain the same data service quality while paying less (V3). This allows adversaries to take advantage of the carriers by purchasing cheaper IoT device access for their non-IoT devices.

**Validation.** We validate this vulnerability by using `iPerf` to examine TCP throughput performance on three devices: (1) an IoT device (*i.e.*, LG Watch Urbane 2nd) equipped with an IoT SIM card, (2) a smartphone (*i.e.*, Samsung S5) with a non-IoT SIM card, and (3) the smartphone spoofing the IoT device's IMEI with an IoT SIM card. We consider both uplink and downlink cases and test each case for 10 runs. Figure 4.3 shows the $10^{th}$, $50^{th}$, and $90^{th}$ percentiles of the throughput results of those three devices in the OP-I and OP-II networks. We observe that all the three devices have comparable performance on the uplink and downlink throughput in each operator's network. For example, in the OP-I network, the median uplink/downlink throughout speeds for the IoT device, the smartphone, and the smartphone masquerading as an IoT device are 5.73/16.82 Mbps, 5.91/17.15 Mbps, 5.59/16.85 Mbps, respectively. This shows that the networks do not enforce any noticeable restrictions on IoT devices in terms of data transmission rates. Besides, we do not observe that any restrictions are imposed on IoT data usage volumes.

| Operator | Device | Data Service | Voice Service | Text Service | Dedicated number? | Charge (per month) |
|---|---|---|---|---|---|---|
| OP-I | Smartphone w/ spoofing | √ | √ | √ | √ | $10 |
| | Smartwatch | √ | √ | √ | √ | $10 |
| | Smartphone | √ | √ | √ | √ | $20 |
| OP-II | Smartphone w/ spoofing | √ | √ | √ | √ | $5 |
| | Smartwatch | √ | √ | √ | × | $5 |
| | Smartphone | √ | √ | √ | √ | $20 |

Table 4.2 Offered services and charges vary with the devices with or without the IMEI spoofing in the OP-I and OP-II networks based on limited data plans.

**Security implications.** Seemingly, carriers just make a simple operational mistake, but this may not be the case. This vulnerability can be attributed to two possible reasons. First, carriers may not have incentives to restrict IoT data services for IoT devices due to its limited benefits. For example, for limited IoT data plan users, the more data that IoT users use, the more profit that carriers can make. Second, carriers may impose service restrictions based on the theoretical maximum uplink and downlink rates of IoT device categories (e.g., CAT-4: 50Mbps/150Mbps, CAT-1: 5Mbps/10Mbps), but they do not take any effect. This is because wireless resources are shared by multiple devices and the theoretical maximum rates are usually much higher than the actual rates available to the networks.

#### 4.4.1.3 A Proof-of-concept Attack

We devise an IoT masquerading attack based on the vulnerabilities V1, V2, and V3. We consider that an adversary has subscribed to a cellular network service with a limited or unlimited data plan. (S)he adds a smartwatch to his/her account and obtains its IoT SIM card from the carrier. Afterwards, (s)he can start to launch the attack by letting his/her smartphone masquerade as the smartwatch based on the IMEI spoofing. We test three main cellular network services on the smartphone: data, voice and text. The results are summarized in Table 4.2. With the attack smartphone, the adversary can make voice calls, send/receive short messages and access the Internet at 10 different locations, but only pay the IoT device access fee. The adversary can save 50% and 75% of the smartphone device access charges in the OP-I and OP-II networks, respectively. Note that OP-II does not assign a dedicated phone number to the smartwatch for voice and text services; the user has to use the phone number belonging to the paired smartphone's SIM card. However, the attack smartphone can obtain a dedicated phone number. It may be because OP-II prevents the IoT SIM card from registering the VoLTE system on the smartwatch, but it is not prohibited on the smartphone.

### 4.4.2 IoT Use Scenario Abuse

We next investigate whether IoT devices can work in unanticipated use scenarios. Current carriers offer cheaper device access fees to some IoT devices due to their limited use scenarios. However, we discover the fourth vulnerability (V4) that those IoT devices may not be restricted to their anticipated use scenarios. We validate this vulnerability on two different types of popular IoT devices: car-connected mobile hotspots and smartwatches.

#### 4.4.2.1 Car-connected Hotspots: Not Limited to only Vehicles

Car-connected hotspots are, by default, designed for using only inside vehicles. However, when they are fully controlled by adversaries, some malicious manipulations can be performed to bypass the usage restriction. We discover that the adversary may turn these car-connected hotpots into common mobile hotspots, which offer mobile data services.

We observe that two hardware features of car-connected hotspots restrict their usage to only inside operating vehicles. First, its power supply is from the diagnostic connector of OBD-II (On-Board Diagnostics II), which is a system for the status report of various vehicle subsystems. The OBD-II connector is not used for other non-vehicle systems, so the car-connected hotspot is hardly powered on outside vehicles.

Second, the hotspot automatically enters a sleep mode after the vehicle has been turned off for a period of time. The hotspot detects whether the vehicle is operating based on voltage changes of the OBD-II connector. According to the hotspot's specification, it operates normally when the voltage of the OBD-II connector is higher than 11.7 V. The voltage of the OBD-II connector can increase up to 15.5 V at the moment that the vehicle engine is ignited. The device disables its hotspot function and enters the sleep mode, when the voltage of the OBD-II connector drops to

11.7 V and 9 V, respectively. Once the adversary makes a power supply with the OBD-II connector interface and then sets its voltage to be higher than 11.7 V, the device can be turned into a common portable hotspot.

**Validation** We validate this vulnerability by testing whether a car-connected hotspot can continue to be used outside vehicles with our customized power supply. To keep the device's hotspot function active, the power supply is made to output 12 V from a power bank through a voltage regulator. We then connect the power bank's ground and power pins to the fourth and sixteenth pins of the OBD-II connector, respectively, via the regulator. After powering on the hotspot, we connect a Wi-Fi client to the hotspot and use the client to keep generating traffic to/from the Internet using `ping`. We run the test for a whole day, and the traffic is not interrupted.

#### 4.4.2.2 Smartwatches: Not Constrained by Hardware or Software

Smartwatches with hardware constraints (e.g., small screen) are mainly developed to assist mobile users in getting voice/text services, simple data services (e.g., voice assistants), and notifications from their paired smartphones. Therefore, by design, there is only a small number of smartwatch applications, and their functions are more limited than smartphone applications. For example, Google wearable devices are not allowed to install standalone Gmail (e.g., working without paired smartphones), Chrome browser, and Youtube. However, these hardware/software constraints are not sufficient to restrict the real-world usage of the smartwatch. Specifically, the smartwatch can be turned into a mobile data gateway, which forwards data packets between a Wi-Fi device and the Internet, to provide Internet access over Wi-Fi. Note that the Wi-Fi device connects to the smartwatch via Wi-Fi and the smartwatch connects to the Internet via the cellular network.

**Validation.** We validate this vulnerability by examining whether the network forbids a smartphone's data packets which are forwarded by the smartwatch. We develop a data forwarding application on the smartwatch. It first receives the smartphone's data packets from the Wi-Fi interface and sends them to our external UDP server on the Internet through the cellular network interface. In our test, the smartphone transmits about 500 MB traffic to the forwarding smartwatch. Our experiment shows that all the transmitted packets are received by our Internet server. No restrictions from the

OP-I and OP-II networks are observed for this usage.

**Security implications.** This vulnerability can be attributed to two potential issues. First, there are various cellular IoT use scenarios, so it is challenging for the infrastructure to identify all possible use scenarios. Second, even though carriers deploy some constraints or security mechanisms on the IoT devices, they can be easily bypassed at low cost. For example, car-connected devices have to be powered on via the OBD-II interface, and smartwatch users are not allowed to install the applications that smartphone users can install.

#### 4.4.2.3 Two Proof-of-concept Attacks

We devise two proof-of-concept attacks to assess the real-world damages of the vulnerability V4.

**Car-connected IoT abuse: portable mobile hotspot** In this attack, we turn a car-connected IoT hotspot (i.e., Mobley) into a mobile hotspot and then compare its performance with an ordinary mobile hotspot (i.e., Velocity). We here present the results obtained in the OP-I network, but skip that of OP-II because of similar phenomena. We connect a laptop with an 802.11ac Wi-Fi card to each of those two hotspots, Mobley and Velocity, and gauge its uplink/downlink performance. In the test, the hotspots are located at the same location, and the laptop is placed at six different locations, which are spaced at 2-meter intervals, for a total range of 10 meters (i.e., S1-S6). S1 is the closest to the hotspot location, whereas S6 is the farthest from the hotspots.

We test uplink and downlink throughput for 10 runs in each case and plot $10^{th}$, $50^{th}$, and $90^{th}$ percentiles of the throughput results in Figure 4.4. We observe that the two hotspots have comparable performance for both uplink and downlink throughput at each location. Specifically, the differences between their median throughput results are within only 5.62% and 2.03% for all the cases in the OP-I and OP-II networks respectively. Neither of the hotspots always outperforms the other. Take the OP-I's limited data plans as an example for the gain estimation of this attack. $10 and $20 device access fees are charged for the car-connected and normal mobile hotspots. With this attack, the adversary can gain a hotspot service for 50% cheaper. The gains can vary with different carriers and data plans, as shown in Table 2.2.

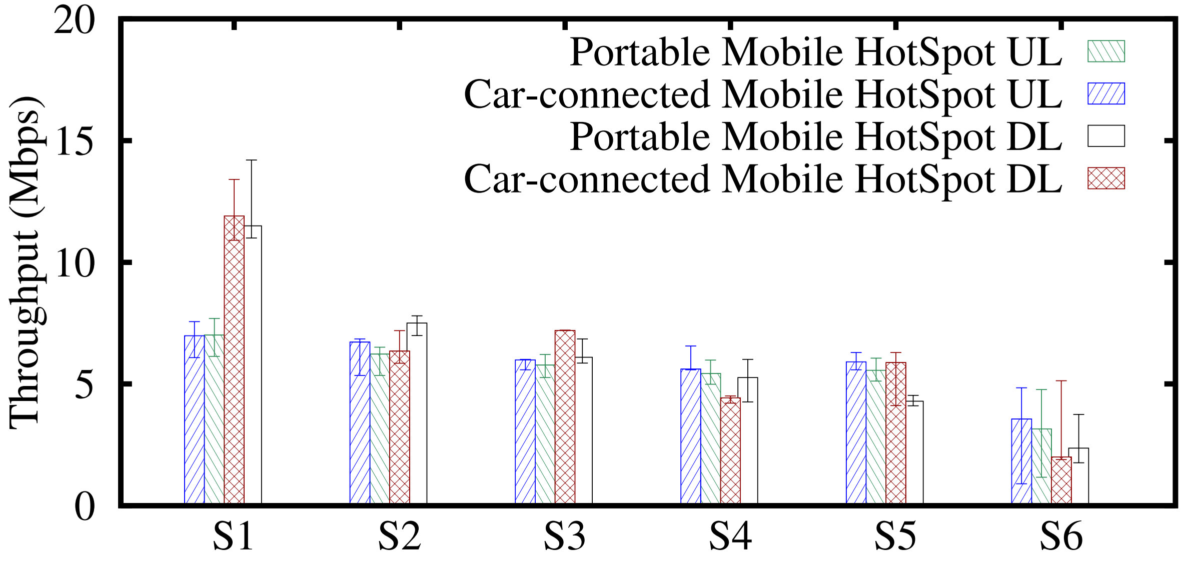


Figure 4.4 The uplink/downlink TCP throughput results at the $10^{th}$, $50^{th}$, and $90^{th}$ percentiles are plotted for an IoT-masqueraded hotspot (*i.e.*, Mobley) and a normal mobile hotspot (*i.e.*, Velocity) in the OP-I network, given a laptop client placed at six indoor locations in our campus.

**Wearable IoT abuse: mobile data gateway** We devise an attack that abuses a wearable IoT, smartwatch, to be a mobile data gateway, which can provide a local area network with Internet access through the mobile data service. This IoT-masqueraded gateway can cooperate with a Wi-Fi AP to supply Internet access to Wi-Fi devices. We enable it to work for all the applications on Wi-Fi devices by taking a VPN approach.

Figure 4.5 shows the network architecture that turns a smartwatch to a mobile data gateway. It consists of four components: (1) a VPN server deployed on the Internet, (2) an IoT device supporting both Wi-Fi and cellular networks (e.g., LG Watch Urbane and Samsung Gear S3), (3) a Wi-Fi AP, and (4) a VPN client installed on the Wi-Fi device (here, a smartphone). Both the smartphone and the smartwatch connect to the AP. The VPN client on the smartphone establishes a VPN tunnel with the VPN server, and the smartwatch forwards data between the VPN client and the VPN server through its Wi-Fi and LTE interfaces.

Our experimental results show that the smartphone's applications can access the Internet and work as usual without any changes. Figure 4.6 shows the smartwatch's data usage. The 91% traffic volume consumed by the web and applications is mostly used by the application that forwards data between the Wi-Fi and LTE networks. We further examine the smartwatch's forwarding bandwidth

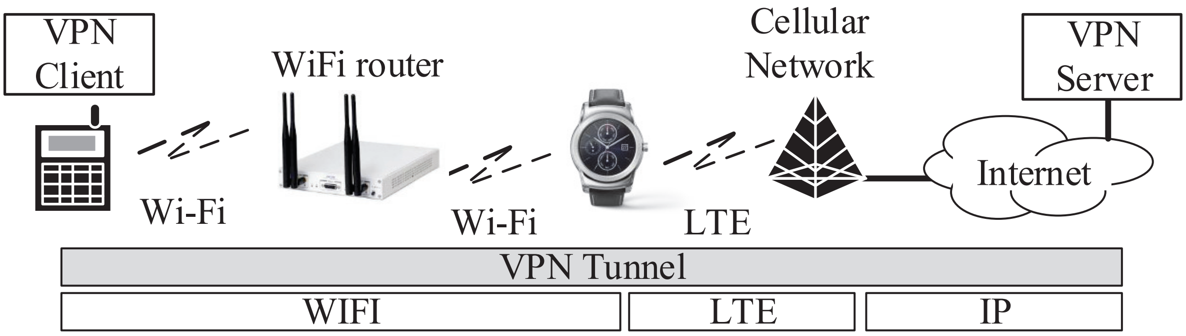
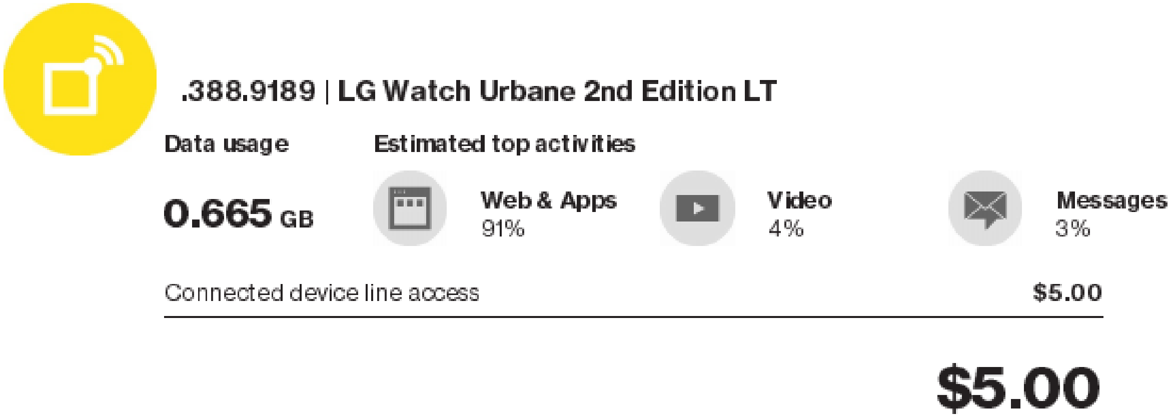


Figure 4.5 The network architecture that turns a smartwatch to a mobile data gateway based on a VPN approach.



.388.9189 | LG Watch Urbane 2nd Edition LT

Data usage

0.665 GB

Estimated top activities

Web & Apps 91%

Video 4%

Messages 3%

Connected device line access $5.00

$5.00

Figure 4.6 The data usage of the smartwatch that masquerades as a mobile gateway. The 91% traffic volume of the total 665 MB, which is consumed by the web and applications, is mostly used by the gateway application.

based on TCP traffic using `iPerf`. It is observed that the median of the TCP throughput over 10 runs can achieve 4.1 Mbps. Note that this attack can work without the Wi-Fi AP in two cases. First, the IoT device supports the Wi-Fi direct technology, which enables Wi-Fi devices to connect to each other directly. Second, the Wi-Fi device can run the VPN and Wi-Fi AP functions simultaneously. As a result, this attack allows the adversary to pay 50% and 75% less in the OP-I and OP-II networks, respectively. Both operators are not capable of detecting or preventing this attack.

### 4.4.3 Long-term IoT Attack Evaluation

We conduct a long-term attack evaluation on the IoT masquerading for eight months, in order to examine whether carriers deploy any anomaly detection mechanism for IoT attacks. In the experiment, we subscribe to a 2 GB data plan and then add a smartphone (i.e., Samsung J7) and

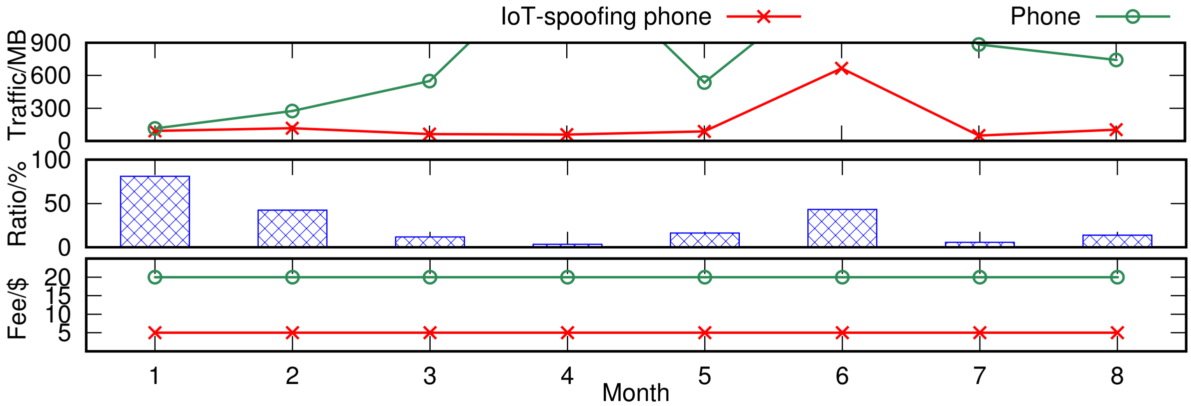


Figure 4.7 An 8-month evaluation of the IoT masquerading attack: a smartphone and an IoT device which another smartphone masquerades as (i.e., IoT-spoofing phone) subscribe to the same 2 GB mobile data plan. Top: monthly data usage volumes; middle: the ratio of the IoT-spoofing phone's data usage to the normal phone's; bottom: monthly device access fees from OP-II.

an IoT device (i.e., LG Watch Urbane 2nd) to this plan. Their device access fees are $20 and $5, respectively. We use another smartphone (i.e., Samsung S5) to masquerade as the IoT device (i.e., LG Watch Urbane 2nd) with IMEI spoofing. During the 8-month duration, the IoT-spoofing phone is scheduled to access the Internet at least once every day.

Figure 4.7 shows monthly data usage volumes for both the smartphone and the IoT-spoofing phone (top), monthly usage ratios (the ratio of the data usage of the IoT-spoofing phone to that of the normal smartphone) (middle), and device access fees charged by carriers (bottom). We make three observations. First, the data usage volumes of the IoT-spoofing phone range from 50 MB to 650 MB, whereas those of the normal smartphone are from 115 MB to more than 900 MB. Second, the ratio of the data usage of the IoT-spoofing phone to the normal smartphone ranges from 3.36% to 80.87%. Third, the tested carrier keeps treating the IoT-spoofing phone as an IoT device according to its persistent IoT device access fee of $5. This result shows that current anomaly detection mechanisms are not able to detect the attack, even though the IoT-spoofing phone's monthly usage volume can be as high as 650 MB or the ratio of its usage to that of the normal smartphone is 80.87%.

## 4.5 Attack Incentive Modeling

In this subchapter, we model mobile user bills and analyze the adversary's maximum gain, as well as give three attack instances to showcase real-world impact.

### 4.5.1 Mobile User Bills Modeling

Suppose that there are $s$ different monthly service plans from an operator, and a mobile user has a subscribed service plan $j$, monthly data usage $u$, and $n_t$ devices from each device type $t$. Given $i$ different device types, the number of devices owned by the user can be represented by $\Sigma_{t=1}^{i} n_t$. The user's monthly bill can thus be modeled as follows:

$$\begin{aligned} Bill_j(u, n_1, \cdots, n_i) = \Sigma_{t=1}^{i} n_t \cdot \alpha_{j,t} + \\ max\{\beta_{j,1}, \beta_{j,2} \cdot u \cdot \mathrm{I}(u \leq cap_j), \beta_{j,2} \cdot cap_j \cdot \mathrm{I}(u > cap_j)\} + \\ \beta_{j,3} \cdot (u - cap_j) \cdot I(u > cap_j), \end{aligned}$$

where $\alpha_{j,t}$ is the device access fee of device type $t$ in plan $j$, $\beta_{j,1}$ is the minimal data service fee in plan $j$ (*e.g.*, \$35 in the OP-II's 2GB plan), $\beta_{j,2}$ is the unit price when $u$ is lower than $cap_j$, which is the maximum data usage for the unit price $\beta_{j,2}$, $\beta_{j,3}$ is the unit price after $u$ exceeds $cap_j$, and $I(x > y)$ is a boolean value (0 or 1) indicating if $x$ is larger than $y$.

**Maximal attack gain.** Suppose that the adversary uses a service plan $j$ before launching an attack. To maximize the attack gain, the adversary can choose the best service plan for his overall usage and the best device type to masquerade as for each device. The gain can be represented as follows:

$$Bill_j(u, n_1, \cdots, n_i) - min\{Bill_k(u, n'_1, \cdots, n'_i)\}$$

where $\Sigma_{t=1}^{i} n_t = \Sigma_{t=1}^{i} n'_t$ and $k = 1, \cdots, s$. By considering all the possible service plans and the charges of all the device types, the adversary can identify an attack policy that maximizes the gain.

### 4.5.2 Three Attack Instances

**Example I: light usage (Saving:\$70→\$14).** Bob usually has free Wi-Fi access and thus requires only small volume of mobile data service on his smartphone. Assume that the required volume is less than 1 GB per year. According to OP-I's monthly data plans, he needs to subscribe to at least a 3 GB data plan with a monthly service fee of $50 and adds his smartphone to the plan with a monthly device access fee of $20. For a one-year time period, he should pay $840 ($70×12). Based on the analysis of maximum gain, the best attack policy is to purchase a monthly 100 MB IoT CAT-1/CAT-M1 plan, which has a *monthly* service fee $14, and then launch the IoT masquerading attack on his smartphone. The attack can reduce his annual bill from $840 to $168 , offering an 80% saving.

**Example II: moderate usage (Saving:$70→$22).** Bob usually uses around 3 GB mobile data per month. The OP-I's 3 GB monthly data plan can be a perfect match for him. The monthly fee is $70 including $50 service access and $20 device access fees. The best attack policy for him is to purchase a 3 GB monthly IoT data plan, which only charges $22, and then launch the IoT masquerading attack on his smartphone. His monthly bill can have a 68.5% reduction, from $70 to $22.

**Example III: heavy usage (Saving:$160→$90).** Bob and his three family members together use more than 8 GB mobile data per month. The OP-II's unlimited data plan is a good match for them. With four smartphone lines, a monthly fee $160 is charged for the unlimited data plan. By launching the IoT masquerading attack, the cost can be reduced to $90, where $75 comes from one smartphone line in the unlimited plan and $15 ($5×3) comes from three smartwatch lines that can be used to masquerade for their three smartphones. This results in a 43.75% saving for Bob's family; on the other hand, there is a 43.75% revenue loss for OP-II on this account.

### 4.6 Difficulties Secure Cellular IoT Service Charging

To secure cellular IoT service charging, the network infrastructure needs to accurately identify IoT devices and use scenarios. However, this can be challenging in practice. We next analyze several potential and existing solutions.

#### 4.6.1 Identifying Devices is Challenging

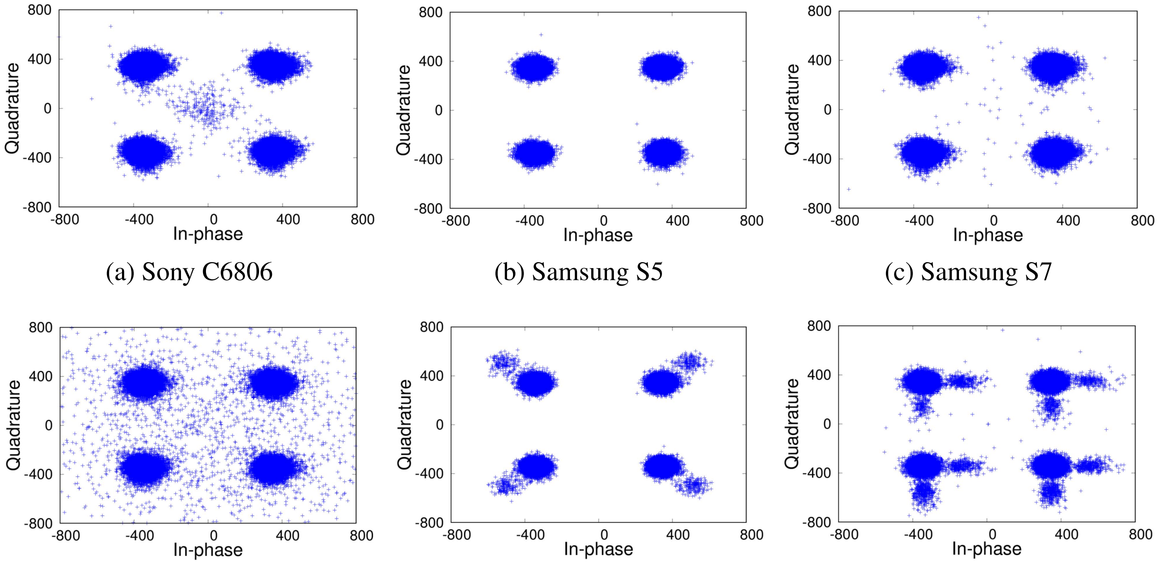
Current cellular networks identify a device based on the IMEI reported by the device itself. When the adversary has full control over the device, it is challenging to prevent its IMEI from being altered. We next introduce four possible remedies for the device identification and discuss their drawbacks.

**Profiling-based device identification.** Cellular IoT devices usually have limited software/hardware capabilities, so the usage volume of their mobile data services can be expected to be low. For example, due to the smartwatch's small display, its Android OS does not support standalone browser and Youtube applications. This can prevent IoT devices, such as smartwatches, from consuming as much data traffic as smartphones. The infrastructure may thus be able to identify the IoT devices based on such low-traffic profiles.

However, this approach has two potential technical issues. First, data usage patterns can vary with users. Given that an IoT device's daily usage volume exceeds a specified threshold, which may be determined based on some statistical usage results, the carrier is still unable to ensure whether the IoT masquerading attack is indeed happening. Second, various IoT devices can have different data usage patterns. Profiling all IoT device types can lead to non-negligible overhead for carriers since there will be more and more new IoT devices in the near future.

**Hardware-based device identification.** Potential hardware-based solutions include the ARM TrustZone and the hardware-based public-key cryptography. The ARM TrustZone has been supported by many popular Cortex-A class processors, crypto chips and secure elements with tamper-proof blocks. Carriers can leverage it to protect the IMEI from being modified by the adversary. However, not all the user devices support this feature. The adversary can easily bypass the protection by using the mobile devices developed on top of the SDR (Software Defined Radio) platforms, which lack the ARM TrustZone support.

With public-key cryptography, each mobile device needs to be assigned a key pair of private and public keys, and an X.509 certificate which is signed by a CA (Certification Authority). The infrastructure can identify each device based on its response to a challenge. Nevertheless, this approach has two major issues. First, not all of IoT devices can support public-key cryptography due to resource constraints (e.g., there is no enough storage space to install security libraries). Second, enabling the public-key cryptography support for the device identification requires modifications to current cellular network standards.



(a) Sony C6806 (b) Samsung S5 (c) Samsung S7

(d) Nexus 6P (e) LG G3 (f) LG Watch

Figure 4.8 QPSK constellation diagrams collected on six mobile devices.

**RF fingerprint-based device identification.** Another possible solution is to identify devices based on their different RF fingerprints. The differences come from device types and the imperfections of device hardware. This has been proposed to address some security issues such as intrusion detection [105], access control [106], wormhole detection [107], and to improve inter-cellular security [108], to name a few. To assess the effectiveness of this approach, we conduct experiments using the OpenAirInterface (OAI) platform, a software-defined 4G LTE infrastructure [109]. We collect the RF signals transmitted by various mobile devices that connect to the OAI eNodeB. The experiment starts after the tested device is powered on and stops after the RRC (radio resource control) connection between the device and the OAI eNodeB is established. Note that we take two measures to prevent the experiment from affecting other normal mobile users. First, we configure the eNodeB to use the LTE band 7, which is not used by carriers in North America. Second, we put the OAI platform and the mobile devices in a RF shielded enclosure box. Figure 4.8 shows the QPSK (Quadrature Phase-Shift Keying) constellation diagrams of six mobile devices including IoT and non-IoT devices. Seemingly, we can identify these devices by analyzing their constellation diagrams, especially for the LG G3, LG Watch Urbane 2nd and Sony C6806. However, this approach is not scalable as it requires the eNodeB to collect all the IoT devices' RF fingerprints.

**Tethering-detection-based device identification.** Tethering detection has been deployed by operators to detect if users provide their PCs with Internet access by enabling Wi-Fi or USB tethering on their smartphones. However, it still requires significant modifications since it is designed for smartphones rather than IoT devices, and some studies have reported that they can be bypassed (e.g., faking OS signatures).

### 4.6.2 Identifying Use Scenarios is Challenging

The network is capable of identifying abnormal use scenarios of an IoT device to some extent. Take car-connected hotspots as an example. The network has cell-level mobility information of each hotspot, and can keep track of its mobility patterns. However, it is still difficult to identify whether the hotspot is being used inside a vehicle or not. Even if the hotspot keeps staying within a cell for a long time period, it is not necessarily outside the vehicle. It may be due to a serious traffic jam. With the proliferation of cellular IoT devices in the near future, there may be more unprecedented IoT use scenarios. It can be very challenging for the network infrastructure to identify the use scenario of each device.

## 4.7 Solution

We seek for a standard-compliant solution that can rapidly mitigate the IoT attacks. We thus consider eliminating V3, and it can also mitigates the attack incentives on the other three vulnerabilities. We leave the solutions for V1, V2, and V4, which require time-consuming standard modifications and cannot be done shortly, to the future design. Specifically, two new mutual authentication mechanisms are required to address V1 and V2: one is between an IoT SIM card and an IoT device, as well as the other is between the device and the infrastructure. The mutual authentication based on the public-key cryptography can be a potential solution option, but it requires modifications to 3GPP standards, which is time-consuming and cannot be done in a short time. To address V4, a new security mechanism shall be introduced to confine IoT devices to their specific use scenarios. It not only requires standard support but also is challenging for carriers.

To this end, we propose an anti-abuse service model to address V3. This can also largely mitigate the attack incentives on other vulnerabilities. Specifically, our approach ensures that no

IoT users can get better service quality than non-IoT users when the IoT users pay less, which does not require any modifications to SIM cards, mobile devices, and cellular network standards but minimal support from the infrastructure. Moreover, our model is scalable to support various IoT devices and use scenarios and achieves both data service fairness and spectrum utilization efficiency. We finally implement and evaluate it using the OAI platform.

### 4.7.1 Anti-Abuse Service Model

The major idea of this service model is to serve each cellular-connected device with service quality based on its cellular IoT technology category and the device access fee paid by its owner. This can prevent different charges on the same quality of services that the adversary can abuse. Our model consists of two components: operational IoT service consistency and charge-aware service access control. They together ensure that no IoT users can get better service quality than non-IoT users when the IoT users pay less. Note that this assurance cannot be achieved by simple IoT service throttle mechanisms (e.g., limiting data rates to 1 Mbps), since the available data rates of all the devices can be smaller than the IoT rate limits in practice.

#### 4.7.1.1 Operational IoT Service Consistency

With distinct cellular IoT technologies, IoT devices have different capabilities in terms of theoretical maximum uplink/downlink speed. For example, for an IoT device supporting CAT-M1, the theoretical maximum uplink/downlink speed is 1 Mbps/1 Mbps, whereas for an IoT device supporting CAT-1, the theoretical maximum uplink/downlink speed is 5 Mbps/10 Mbps. However, in practice, different entities including IoT devices, SIM cards, and the network infrastructure do not operate in consistency with the cellular IoT profiles. That is, the network may not restrict the performance of the IoT SIM cards based on their profiles. This leads to the gains which the adversary can get by the IoT masquerading. We thus propose that all the parties in the cellular ecosystem shall be consistent with the support of the IoT profiles. For example, when an IoT user subscribes to an IoT sim card for his/her CAT-1 IoT device, the maximum uplink/downlink speed of the CAT-1 IoT SIM card shall be limited to 5 Mbps/10 Mbps by the network no matter what device is used for the SIM card. Therefore, even if the adversary performs the IoT masquerading on a non-IoT device using the IoT SIM card, the device can get only 10 Mbps as its maximum speed.

This service consistency mechanism contains two major tasks in the core network operation. First, the network infrastructure should maintain maximum uplink/downlink speed information for each IoT service subscription based on its subscribed cellular IoT technology category. Second, it should apply the maximum speed to the EPS bearer context activation procedure [15], which is initiated when an IoT device accesses the IoT service with which the SIM card is associated.

#### 4.7.1.2 Charge-aware Service Access Control

Due to fewer resources needed for IoT services, carriers inevitably provide them with cheaper charge plans than conventional non-IoT plans. However, they do not restrict the IoT services from the network but only rely on the inherent constraints of IoT devices. This is why the adversary can abuse the IoT devices to have non-IoT services with cheaper charges. We argue that these differential charges shall be reflected in the service quality which includes traffic priority and maximum transmission rate. This causes the gaps between IoT and non-IoT services to correlate with their charges, thereby reducing attack incentives. We next elaborate on how to correlate the charges with the priority and the maximum rate.

In the LTE network, there are 9 priority levels, which are assigned to different types of traffic [110]. The level number decreases with the increase of priority. For example, the signaling and voice traffic flows of VoLTE (Voice over LTE) respectively have levels 5 and 1, whereas the flows of mobile data services on non-IoT devices are usually given the level 9, which is the lowest priority. Since IoT services are cheaper, their traffic flows should have lower priority than level 9. We then propose to use the level ranging from 9 to 10 to set priority for IoT services and correlate it to their differential charges.

The priority value for an device $X$ can be formulated as:

$$Priority_X = 10.0 - \frac{Charge_X}{Charge_{Highest}}, \tag{4.1}$$

where $Charge_X$ is the device access fee of device X and $Charge_{Highest}$ is the highest device access fee among the devices in the same type of data plan (e.g., limited or unlimited data plan) in the same network. For example, in a 2GB limited data plan, \$20 and \$5 are charged for a smartphone and an LG smartwatch, respectively. Their priority levels should be set to 9 and 9.75 (i.e., 10.0 – \$5/\$20). The cheaper a device's access fee a user pays, the lower the priority of device traffic flows (s)he can receive. Note that we elaborate on how to set various priority levels in Chapter 4.7.2.

We next restrict maximum uplink/downlink transmission rates for IoT devices. We determine the maximum transmission rate of each device by considering both its priority value and the maximum rate given by the operational IoT service consistency. Assume that the maximum rate for non-IoT devices in the same type of data plan is $NonIoTMaxRate$ and the maximum IoT rate from the service consistency is $InitialMaxRate_X$. Then the maximum rate for the IoT device $X$ is formulated as:

$$\begin{aligned} MaxRate_X = Min(&NonIoTMaxRate \times (10 - Priority_X), \\ &InitialMaxRate_X). \end{aligned} \tag{4.2}$$

Take the LG smartwatch as an example. Since its device access fee is \$5 and the smartphone's is \$20 in the OP-II network, its service priority and maximum rate are respectively 9.75 and 25% of the maximum rate that the smartphone can receive when $NonIoTMaxRate \times 0.25$ is smaller than $InitialMaxRate_{watch}$.

#### 4.7.1.3 Computational Complexity Analysis

We next analyze computational complexity of the operational IoT service consistency and the charge-aware service control. We consider the time complexity of associating a new IoT service subscription with its transmission capability. After the association, the network can easily apply the transmission capability to an IoT device based on its SIM profile when it attaches to the network. In the analysis, we assume that (1) the GSMA's and operators' IMEI and SIM card databases are maintained based on the B+ tree [111] (B+ tree is a common data structure used by database systems, such as mySQL), (2) the time complexity of performing an arithmetic operation, such as subtraction, multiplication, division, is $O(1)$, and (3) the time complexity of reading/writing an item in GSMA or operators' databases is $O(1)$.

**Operational IoT service consistency:** Making the service consistent consists of three main steps.

First, the network obtains the information of cellular IoT technologies that the device can support based on its IMEI, which is collected from the device owner. It can be queried from the GSMA's global central IMEI database (`https://imeidb.gsma.com`), which stores all the IMEIs with device profiles, such as manufacturers and software/hardware capabilities. The time complexity of the search operation on a B+ tree database is $O(\log n)$ [111], where $n$ is the number of global mobile devices stored in GSMA's IMEI database. Second, the network identifies the theoretical maximum uplink/downlink speed of the supported IoT technologies. It takes $O(\alpha)$, where $\alpha$ is the number of various cellular IoT technologies. Third, the network associates the device's IoT SIM card with the transmission capability, and adds it into the SIM card database. The time complexity of a B+ database insertion is $O(\log\beta)$, where $\beta$ is the number of active SIM cards stored in the operator's SIM card database. In summary, the total time complexity is $O(\log n) + O(\alpha) + O(\log\beta)$. Since the time complexity related to the number of global mobile devices can dominate in practice, the time complexity for operational IoT service consistently mechanism can be reduced to $O(\log n)$.

**Charge-aware service access control:** This module takes three major steps to add a new IoT service subscription. First, it obtains the highest charge among all the devices in the same type of data plans. The time complexity is $O(\beta')$, where $\beta'$ is the number of active SIM cards in the type of data plan to which the IoT user subscribes (e.g., limited data plan). Since, in practice, $\beta'$ is smaller than $\beta$ (i.e., the number of all active SIM cards that the operator currently support), we can reduce $O(\beta')$ to $O(\beta)$. Second, it obtains the maximum rate of non-IoT devices. It takes only a constant time $O(1)$, since carriers, including OP-I and OP-II, usually apply the same maximum rate to all non-IoT devices. Third, it calculates the IoT subscription's priority value and then determines the final maximum rate according to Equation 4.2. The calculation costs only a constant time $O(1)$. In summary, the total time complexity is $O(\beta) + O(1) + O(1)$ and can be reduced to $O(\beta)$.

**Overall complexity:** As a result, the overall time complexity is $O(\log n) + O(\beta)$, where $n$ is the number of global mobile devices including cellular IoT devices and $\beta$ is the number of active SIM cards that the operator currently support. In practice, $n$ is much larger than $\beta$.

#### 4.7.1.4 Merits

We next summarize three major merits of the anti-abuse service model. First, the model does not require any modification to cellular IoT standards or devices, since its two components can be carried out in the standard EPS bearer context activation procedure [15], which is initiated by the infrastructure when an IoT device accesses the IoT services. Second, it can be scalable to support a variety of devices and use scenarios, as it does not require calibration of the IoT service rates for various devices and use scenarios. This is especially relevant with more and more devices being introduced in the future. Third, it can achieve both data service fairness and spectrum utilization efficiency. For the fairness, it can guarantee that no IoT devices can get better services than non-IoT devices when the IoT owners pay less. For the efficiency, IoT devices still have chances to achieve their maximum speeds when radio resources are sufficient (e.g., no contention comes from non-IoT devices). Note that for limited IoT data plan users, the more data that IoT users use, the more profit that carriers can make.

### 4.7.2 Implementation

We implement the anti-abuse model on the OAI platform. It consists of the 4G core network and RAN. The 4G core network runs on a laptop (Acer Aspire E5-575-53EJ). The RAN contains the eNodeB on a PC (Dell Inspiron 3268) and a software-defined radio (USRP B210). We mainly modify three entities: the HSS, the MME, and the eNodeB (see Figure 2.3).

**HSS.** We add two types of new information in the user subscription data, which are associated with each SIM card: user equipment profile and charge rate class. The former indicates the highest technology category (e.g., CAT-4) that the SIM card can support. The latter represents the operator-specific charge rate class (e.g., 25% off, 50% off) to which the SIM belongs. These are used by the MME to determine service priority for the SIM. We add the delivery of this information to the normal procedure that the MME has to obtain user authentication information from the HSS. The new information entries are included in an element `UE-Usage-Type` of the response to the request `Authentication Information Retrieval`, which is sent from the MME to the HSS.

**MME.** The maximum uplink/downlink rates and the service priority are set for each SIM card based on that new information provided by the HSS. We introduce new QoS to the EPS radio access bearer (E-RAB). During the `E-RAB Setup` procedure [112], the MME specifies those two restrictions in the fields, `UE Aggregate Maximum Bit Rate` [112] and `E-RAB Level QoS Parameters` [112], respectively in the `E-RAB Setup Request` message, which is sent to the eNodeB.

**eNodeB.** We support new priority levels (e.g., 9.25 and 9.5) by defining new QoS Class Identifier (QCI) values, which are used to represent QoS classes in the LTE network. Each QCI value is an 8-bit unsigned octet. The QCI values, ranging from 128 to 254, are reserved for operator-specific usage, so new QCI values can be added in this range. In our implementation, we define two new priority levels 9.25 and 9.5 by adding new QCI values 129 and 130, respectively. Note that the eNodeB in the current OAI implementation does not support full QCI functions specified by the standards. We thus add a Service Control Entity (SCE), which is a Linux server, between the eNodeB and the 4G core to fulfill the regulation of the maximum rates and the service priority.

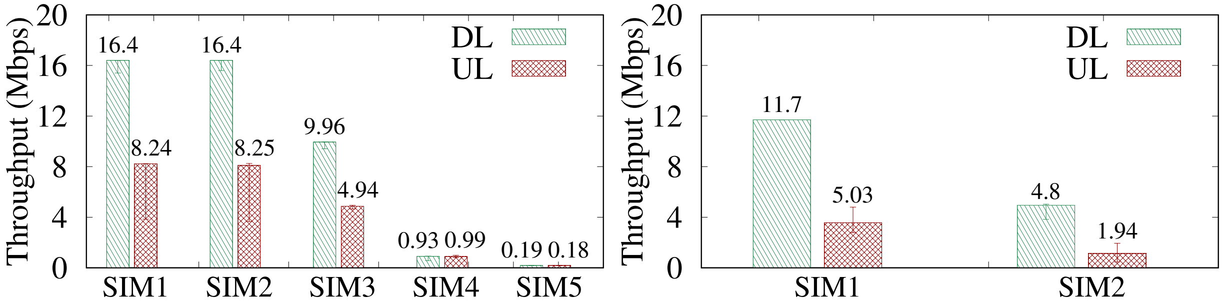
### 4.7.3 Evaluation

We evaluate our solution based on the OAI-based prototype. We use 5 sysmoUSIM-SJS1 SIM cards, which are standard-compliant, and add their information to the HSS database. They are configured to have five different categories (*i.e.*, CAT-10, CAT-4, CAT-1, CAT-M1 and NB-IoT), and classified into three priority classes: 9, 9.5, and 9.75. The device access fees of those three priority classes are respectively 0%, 50% and 75% cheaper than non-IoT devices. These configurations are summarized in Table 4.3. We use the `iPerf` tool to assess throughput of user devices.

**Operational IoT service consistency.** We use one device (Nexus 6p) with different SIM cards shown in Table 4.3 to assess the operational consistency for IoT profiles. We test both uplink and downlink speed performance. The test on each SIM card has 10 runs with 30 seconds each. Figure 4.9a shows maximum, median and minimum downlink/uplink speed results for the SIM cards. There are two observations. First, the maximum throughput results of SIM1 and SIM2 are similar (*i.e.*, 8 Mbps and 16 Mbps for uplink and downlink, respectively), because they are bound

| SIM | Highest Theoretical UE DL/UL speed | Priority Value | Mapped Operator Plan |
|---|---|---|---|
| SIM1 | CAT-10 (450 Mbps/150 Mbps) | 9 | Non-IoT, $20 |
| SIM2 | CAT-4 (150 Mbps/50 Mbps) | 9.5 | IoT, $10 |
| SIM3 | CAT-1 (10 Mbps/5 Mbps) | 9.5 | IoT, $10 |
| SIM4 | CAT-M1 (1 Mbps/1 Mbps) | 9.75 | IoT, $5 |
| SIM5 | NB-IoT (0.2 Mbps/0.2 Mbps) | 9.75 | IoT, $5 |

Table 4.3 The configurations of our test SIM cards.



(a) Single connected device

(b) Multiple connected devices

Figure 4.9 Maximum, median and minimum uplink/downlink speeds vary with SIM cards.

by the OAI platform's maximum throughput, which is smaller than their maximum speeds. Second, for the other three SIM cards, the maximum uplink/downlink speeds are 4.94 Mbps/9.96 Mbps (SIM3), 0.99 Mbps/0.93 Mbps (SIM4) and 0.18 Mbps/0.19 Mbps (SIM5), respectively. They are bound by the regulated maximum speeds of the cellular IoT technologies.

**Charge-aware service control.** We next examine whether the service priority control can take effect in the prototype. We use two phones, Nexus 6p and Samsung S5, with SIM1 and SIM2, respectively. Both phones have much larger maximum downlink/uplink throughput than the OAI platform's throughput bottleneck. The service priority levels assigned to them are respectively 9 and 9.5 based on the priority classes. We have 10 runs for each test. In each run, we generate traffic to gauge throughput performance on them simultaneously, and examine how they affect each other. Figure 4.9b plots maximum, median and minimum uplink/downlink results. It is observed that the maximum throughput results for Nexux 6p with SIM1 and Samsung S5 with SIM2 are 5.03 Mbps/11.7 Mbps and 1.94 Mpbs/4.8 Mbps, respectively. It confirms that the service flows of

Nexus 6P with priority level 9 have higher priority than those of Samsung S5.

# CHAPTER 5

# MPKIX: TOWARDS MORE ACCOUNTABLE AND SECURE INTERNET APPLICATION SERVICES VIA MOBILE NETWORKED SYSTEMS

Nowadays, both Internet Application Service (IAS) providers and users face various security threats and legal issues. Due to the lack of reliable user information verification mechanisms, adversaries can abuse IASs to launch various cyberattacks, such as misinformation distributing and phishing, by using fake user accounts. IAS providers may thus inadvertently offer inappropriate content to restricted users, thereby suffering a serious risk of prosecution under local or international laws. Also, IAS users may suffer from nefarious ID theft attacks. In this chapter, a novel security framework, MPKIX, designated as Mobile-assisted PKIX (Public-Key Infrastructure X.509) is proposed. MPKIX secures both IAS providers and users by leveraging the broadly used PKIX services and mobile networked systems. It not only provides IAS providers with a reliable user verification mechanism while simultaneously enabling cross-IAS user privacy protection, but also largely mitigates the possibility of ID theft attacks and benefits other involved parties, such as cellular network operators and PKIX service providers.

This chapter makes four contributions.

- MPKIX provides IAS providers with a reliable verification mechanism of user information while providing IAS users with cross-IAS privacy protection via the developed ppQuery mechanism. It can prevent various cyberattacks launched by false user accounts and distribution of improper content. Moreover, MPKIX secures IAS users from nefarious ID theft attacks without revealing unnecessary user information to IAS providers. By conforming to existing PKIX and cellular network standards, MPKIX has a small deployment cost. It can facilitate the delivery of accountable and secure online application services.

- The effectiveness of the proposed MPKIX framework is demonstrated experimentally. First, the MPKIX testbed is capable of processing up to 130,000 CSRs (Certificate Signing Requests) per minute and producing the corresponding CA-signed PKIX user certificates. Second, the terminal-side prototype of MPKIX is evaluated on both phones and computers. It is shown that MPKIX works well even on low/medium-end phone models. Third, MPKIX enables IAS providers to effectively verify the correctness of user information within less than 1 second without compromising user privacy. Fourth, the decision of the arbitration of a disputed IAS ID revocation/claim can be made within 4 seconds, whereas the current practice takes several business days or weeks.

- A security analysis of the MPKIX framework is conducted. It shows that MPKIX not only offers desirable security guarantees, such as data integrity, non-repudiation, user privacy, and accountability, but also defends against various attacks.
- MPKIX benefits all the involved parties. Specifically, *CAs* can expand their enterprise-based PKIX credential services to billions of mobile users. *cellular network operators* can make profit by answering the queries about user information from IAS providers. *IAS providers* can ensure the correctness of user information so that the risk of improper content distribution and cyberattacks can be minimized. *IAS users* have an efficient privacy-aware mechanism to claim/revoke impersonated IDs without revealing additional user information to IAS providers.

## 5.1 Related Work

**Side-channel inference/verification:** Several methods have been proposed to infer/verify user demographics (e.g., age, gender and education level) using side-channel information (e.g., HTTPS packets and social network activities). Specifically, Wang *et al.* [113] developed a tensor factorization based method, *Dinfer*, for inferring user demographic attributes from WiFi AP trajectories; Li *et al.* [41] applied machine learning to analyzing campus WiFi traffic and inferred the user's gender and education level; Neal *et al.* [114] devised a multimodal-based approach to predict user gender based on usage records of Bluetooth and Wi-Fi. However, these schemes have several common issues. First, the error rates are not negligible (e.g., 22% in [41] and 9% [114]). The erroneous inference results for IAS users may lead to unnecessary suspension or mistaken operations of IAS services. Second, the above inference methods can only be applied to registered users, so they do not protect IAS providers from numerous ID-related attacks during user registration.

**Public-key infrastructure:** The PKI has been widely developed and studied in recent years. Specifically, two studies [115, 116] conducted a large-scale analysis of current PKI-based certificate ecosystem, whereas another study [117] used practical symbolic execution to expose noncompliance in X.509 certificates. Moreover, Aas *et al.* [118] introduced an automated certificate authority, *Let's Encrypt*, for free issuance of HTTPS certificates. Wang *et al.* [119] distributed the trust for certificate authenticity between the corresponding CA and the certificate owner by letting them co-sign the certificate. Wang *et al.* [120] employed cache spaces on IoT devices as a large pool to store validated certificates. Hoglund *et al.* [121] introduced a lightweight profile for X.509 digital certificates for resource-constrained IoT devices. Rashid *et al.* [122] and Papageorgiou *et al.* [123] developed a blockchain-based public key infrastructure for the decentralized issuance and management of digital certificates.

Different from the above studies, MPKIX aims to leverage cellular networked systems to provide IAS providers with authentic user information and protect IAS users from nefarious ID theft attacks while preserving user privacy. Notably, this problem has not been addressed.

**Mobile Connect:** Mobile Connect[124] enables mobile users to log onto IAS services using mobile phones. Specifically, when an IAS user accesses an IAS service, a cellular-network-initiated user authentication is conducted on the user's mobile phone. The authentication result is then returned to the IAS server. The IAS user can choose to provide the IAS provider with nothing, Mobile Connect identity (i.e., phone number) only, or Mobile Connect identity and other user information (e.g., birthday).

Compared with MPKIX, Mobile Connect has the following limitations. First, an IAS user using Mobile Connect is required to use his/her mobile phone and have cellular network connectivity on it while accessing an IAS service; this requirement may decrease the applicability of Mobile Connect. However, MPKIX does not have this limitation, since it supports not only computers connecting to mobile phones with the MPKIX service but also an offline mode in which a CA-issued ppCert and its corresponding private key are exported to other cryptographic tokens (e.g., Yubikey). Second, Mobile Connect does not provide users with a cross-IAS querying mechanism with fine-grained privacy-preserving configuration. It allows IAS users to disclose only least information for user verification. Third, Mobile Connect does not support a privacy-aware ID claim/revocation mechanism, which prevents users from disclosing additional information to IAS providers for the ID claim or revocation. However, the above two mechanisms are supported by MPKIX.

## 5.2 Threat Model, Assumptions, and Security Guarantees

**Threat Model:** In this study, adversaries are people or organizations who aim to impersonate IAS users, abuse IASs with false user information, or infer undisclosed information of IAS users. Two different types of adversaries are considered, namely semi-trusted IAS providers, which are interested in disclosing user identity and information, and network adversaries. The adversary capabilities are assumed to be the same as the Dolev-Yao model [125]; that is, adversaries can overhear, intercept, and synthesize any messages, but are constrained by the cryptographic methods in use (e.g., adversaries cannot decrypt ciphered messages without corresponding cipher keys). Moreover, we bear in mind to conduct this study in a responsible manner. All experiments and evaluations were conducted conforming to the IRB policy; no human subjects were involved.

**Assumptions:** MPKIX makes the following assumptions: (1) cellular network operators follow local/international information privacy laws (e.g., Code of Federal Regulations: Title 47 [126]) to protect user information from being leaked to other parties without user consent; and (2) the adversaries adhere to all cryptographic assumptions; e.g., they cannot restore an original message from its hashed value or decrypt an encrypted message without its decryption key.

**Security Guarantees:** MPKIX offers four security guarantees: (1) *data integrity*, which guarantees accuracy and consistency of the user information provided by an IAS user to an IAS provider; (2) *non-repudiation*, which guarantees that an IAS user cannot dispute authorship of the information revealed by the user to an IAS provider; (3) *user privacy*, which guarantees that an IAS user can reveal only partial user information to an IAS provider while accessing the IAS and initiating ID claim/revocation arbitration, and moreover, the undisclosed user information cannot be inferred (e.g., adversaries cannot correlate any IAS user with a particular individual or a small group based

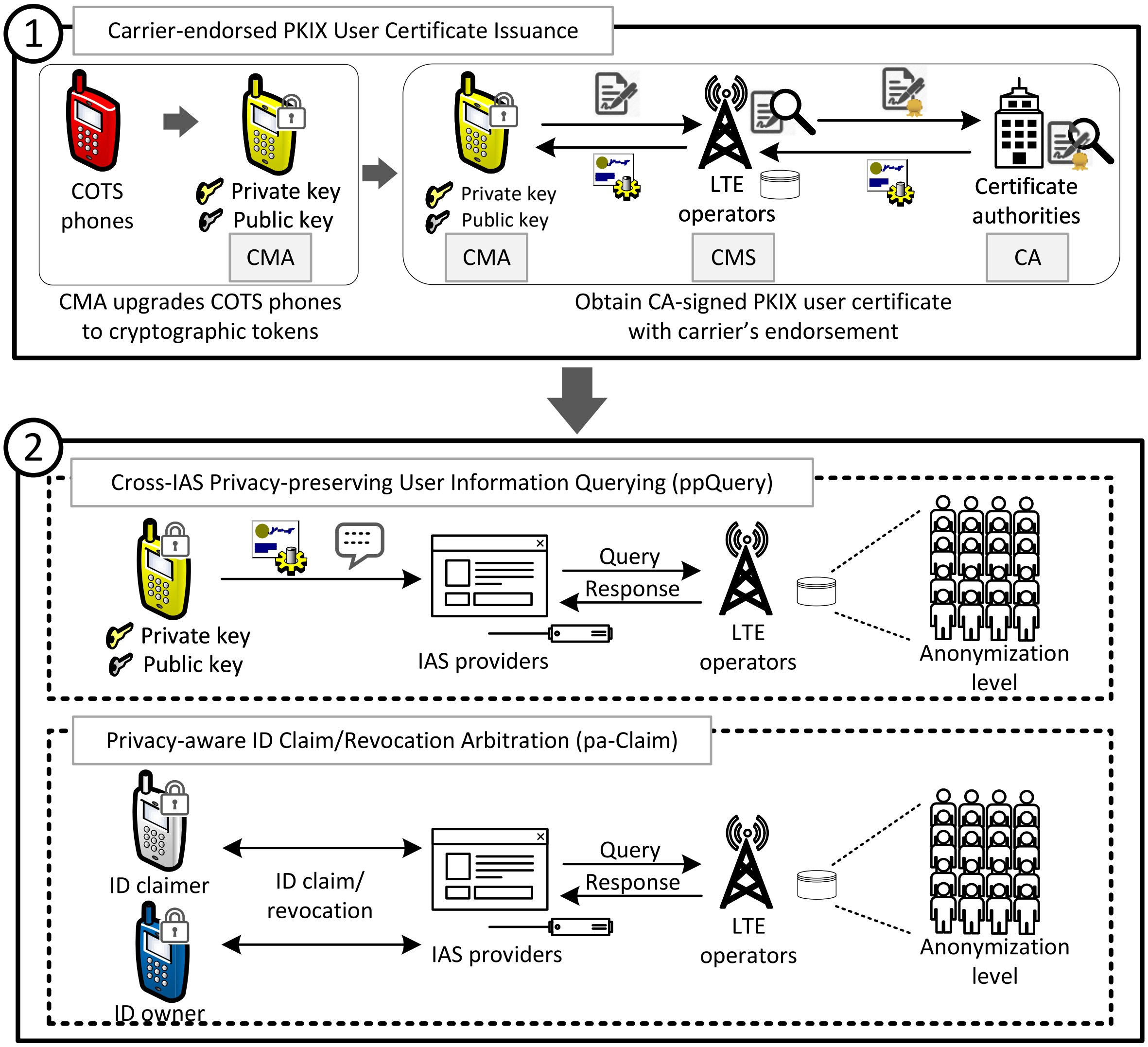

1
Carrier-endorsed PKIX User Certificate Issuance
COTS phones
Private key
Public key
CMA
CMA upgrades COTS phones to cryptographic tokens
Private key
Public key
CMA
LTE operators
CMS
Certificate authorities
CA
Obtain CA-signed PKIX user certificate with carrier's endorsement
2
Cross-IAS Privacy-preserving User Information Querying (ppQuery)
Private key
Public key
IAS providers
Query
Response
LTE operators
Anonymization level
Privacy-aware ID Claim/Revocation Arbitration (pa-Claim)
ID claimer
ID owner
ID claim/ revocation
IAS providers
Query
Response
LTE operators
Anonymization level

Figure 5.1 The overview of MPKIX.

the user's information; (4) *accountability*, which guarantees that, given an IAS-based cyberattack, the law enforcement authority can discover real identities of the IAS provider and user.

## 5.3 MPKIX Design

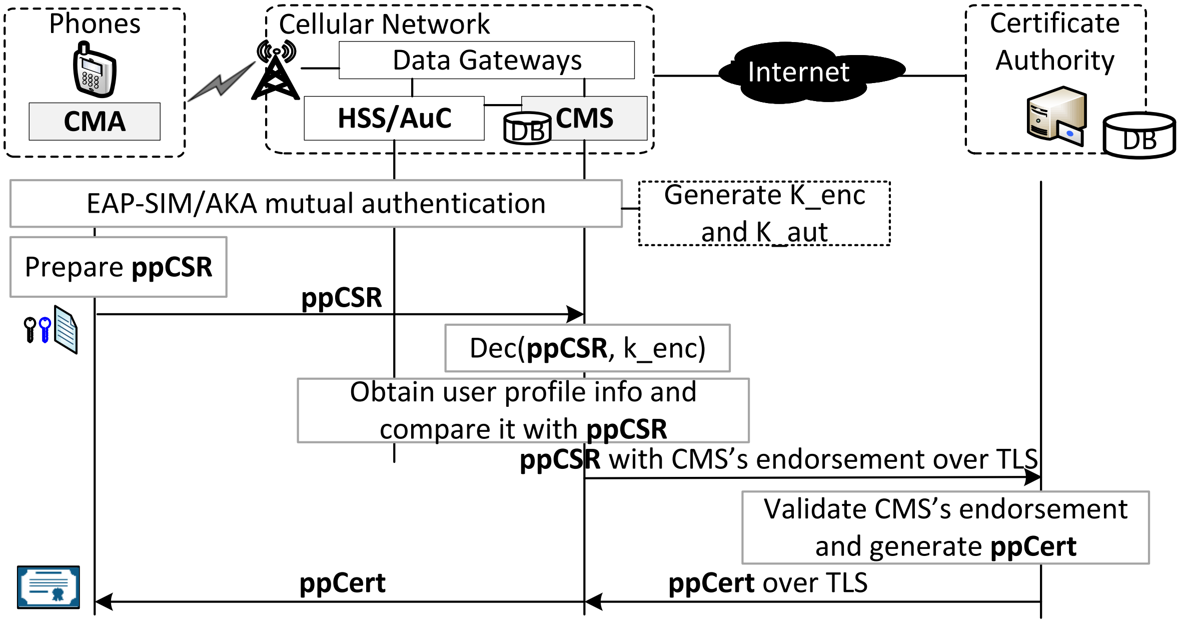
MPKIX enables a mobile user to securely access IAS services while preserving user privacy from semi-trusted IAS providers and providing the IAS providers with a reliable means to verify the user information essential to IASs. Figure 5.1 shows an overview of MPKIX containing three major service components, namely carrier-endorsed PKIX user certificate issuance (**ceIssuance**), cross-IAS privacy-preserving user information querying (**ppQuery**), and privacy-aware ID claim/revocation

| Category | Symbol | Description |
|---|---|---|
| **ceIssuance** | CMA | Certificate Management Application |
| | CMS | Certificate Management Server |
| | ppCSR | Privacy-preserving Certificate Signing Request |
| | ppCert | Privacy-preserving User Certificate |
| | $K_{enc}$ | An encryption key used to encrypt data. |
| | $K_{aut}$ | An authentication key to calculate message authentication code for integrity protection. |
| **ppQuery** | $\mathbb{IA}$ | Anonymization factor. |
| | $S$ | A subject attribute (e.g., name). |
| | $f_S$ | The anonymization function of a given $S$. |
| | $V_S$ | The value of a given $S$ (e.g., Smith). |
| | $m$ | The number of subject attributes. |
| | $\|DB\|$ | The number of users in the database. |
| | $H_{u,i}$ | The highest anonymization level used by user $u$ for the value of $S_i$. |
| **paClaim** | $V_i^{claimer}$ | ID claimer's value for $S_i$. |
| | $V_i^{owner}$ | *Owner*'s value for $S_i$. |
| | $W_i$ | The weight of the Levenshtein distance for $S_i$ |

Table 5.1 Summary of abbreviations, symbols, and parameters in MPKIX.

arbitration (**paClaim**). To enable the MPKIX service, a mobile user must apply to **ceIssuance** for an MPKIX user credential including a CA-signed PKIX user certificate, where user information is encrypted and has been verified by a cellular carrier, and a key pair of public and private keys. With the MPKIX user credential, the user can securely access IAS servers with the support of PKIX-based mutual authentication, which is supported by most mainstream security protocols (e.g., HTTPS, SSL/TLS, and IPSec). To preserve user privacy, **ppQuery** enables IAS providers to verify the user certificate through a cellular carrier for user authentication without decrypting user information in the certificate. **paClaim** enables MPKIX users to claim/revoke an IAS ID that an adversary forges from IAS providers without disclosing any additional user information.

We next elaborate on each of the three service components, where abbreviations, symbols, and parameters are summarized in Table 5.1.



Phones
CMA
Cellular Network
Data Gateways
HSS/AuC
DB
CMS
Internet
Certificate Authority
DB
EAP-SIM/AKA mutual authentication
Generate K_enc and K_aut
Prepare ppCSR
ppCSR
Dec(ppCSR, k_enc)
Obtain user profile info and compare it with ppCSR
ppCSR with CMS's endorsement over TLS
Validate CMS's endorsement and generate ppCert
ppCert
ppCert over TLS

Figure 5.2 MPKIX carrier-endorsed user certificate issuance.

### 5.3.1 ceIssuance: Carrier-endorsed PKIX User Certificate Issuance

The ceIssuance mechanism was developed to facilitate the issuance process of PKIX-based user certificates while satisfying diverse demands of privacy protection from IAS users. It leverages mobile user information that has been verified by cellular network operators during mobile service activation[1], and introduces privacy-preserving certificate signing request (ppCSR) and certificate (ppCert).

Figure 5.2 presents an overview of the proposed mechanism involving four key parties: (1) Certificate Management Application (**CMA**), which is an MPKIX application running on the applicant's mobile phone; (2) HSS/AuC (Authentication Center), where HSS stores verified user information (e.g., names, ages) and subscriptions (e.g., service plans) of mobile users, and AuC is a subsect of the HSS that maintains secret keys shared with mobile users and generates a pair of challenge and expected response to HSS for user authentication; (3) Certificate Management Server (**CMS**), which is an application server (AS) [128] deployed in the cellular network and can obtain user information from the HSS over the cellular-specific Sh interface [129] with secure communications based on the 3GPP-stipulated Diameter protocol [130] over TLS; and (4) MPKIX-supported CA,

[1] Verifying mobile user information has been required by the law in many areas (e.g., China and Thailand) and is becoming a mandatory policy [127].

which collaborates with cellular network operators to issue PKIX user certificates. Notably, the CMS is a standard-compliant AS accessing the HSS based on the 3GPP-stipulated interface and secure communication protocol, so its deployment does not cause new security threats to cellular networks.

The **ceIssuance** service comprises three parts: (1) secure mutual authentication between CMA and CMS; (2) ppCSR preparation, validation, and endorsement; and (3) ppCert issuance. We describe them in detail below.

#### 5.3.1.1 Secure Mutual Authentication

We deployed a mechanism of secure mutual authentication between CMA and CMS to defend against the attacks of certificate applicant masquerading and rogue infrastructure. It is based on mobile Extensible Authentication Protocol (EAP), which relies on cellular-specific symmetric cryptography with a secret key $K$ shared between UE (in the (U)SIM card) and HSS. It has two methods, namely EAP-SIM [131] and EAP-AKA [6], which are used by 2G and 3G/4G/5G networks, respectively. They were adopted to enable the secure mutual authentication in MPKIX, and two 128-bit security keys were thus derived and shared between CMA and CMS: (1) $K_{aut}$, an authentication key used to calculate message authentication code for integrity protection; and (2) $K_{enc}$, an encryption key used to encrypt data.

In particular, CMA and CMS authenticate each other and derive the above two keys as follows:

**Step 1:** CMA provides CMS with the user's subscriber identity, i.e., international mobile subscriber identity (IMSI), through an exchange of EAP-Request and EAP-Response identity messages.

**Step 2:** As an EAP authenticator, CMS obtains a user authentication vector from HSS for the authentication purpose of CMA. The authentication vector contains a random number serving as a challenge, an expected challenge response, a transient master secret key, and a network authentication token, which consists of an ownership proof of the secret key $K$ and a configuration of 3GPP authentication and key generation functions [132]. Note that all the above functions require the secret key $K$.

**Step 3:** After receiving the user authentication vector, CMS sends an EAP-Request message carrying the challenge and the network authentication token to CMA.

**Step 4:** After receiving the EAP-Request message, CMA first validates the ownership proof of the secret key $K$ to authenticate CMS, then generates an answer to the challenge, and finally produces a shared transient master secret key using the configured security functions and the shared secret key $K$ within the (U)SIM card. The transient master secret key is then fed as a seed to an EAP-defined pseudo-random number function [133], and then the function generates a pair of the $K_{enc}$ and $K_{aut}$ security keys. Afterwards, CMA replies an EAP-Response message to CMS with the answer to the challenge.

**Step 5:** On receipt of the EAP-Response message, CMS verifies the answer and then generates the pair of the $K_{enc}$ and $K_{aut}$ security keys based on the transient master secret key shared with CMA. Note that the $K_{enc}$ and $K_{aut}$ will be generated once when applying for the MPKIX user credential via ceIssuance service.

We further use the $K_{enc}$ and $K_{aut}$ security keys to generate the ppCSR, as described below.

#### 5.3.1.2 ppCSR preparation, validation, and endorsement

To request an MPKIX certificate, i.e., ppCert, CMA prepares a certificate request, ppCSR, and sends it to CMS for validation and endorsement. For the ppCSR preparation, CMA first generates a pair of private and public keys, and then produce four major elements: (1) subject: containing user information attributes such as name, address, and phone number; (2) subject extension: domain name of the MPKIX CMS server (e.g., cms.mpkix.att.com); (3) public key information: the generated pubic key and key algorithm; and (4) digital signature. For each attribute, a hash value of the attribute value is generated based on the SHA-1 algorithm and the authentication key ($K_{aut}$), and then the hash value is encrypted by the AES encryption algorithm and the encryption key ($K_{enc}$), as illustrated in the upper part of Figure 5.3.

After receiving the ppCSR from CMA, CMS first verifies the digital signature and then validates the encrypted hash value of each attribute by using the same keys and algorithms shared with CMA and checking authentic user information from HSS. If any error occurs, CMS rejects the ppCSR; otherwise, it endorses the ppCSR by attaching its digital signature and then sends the endorsed

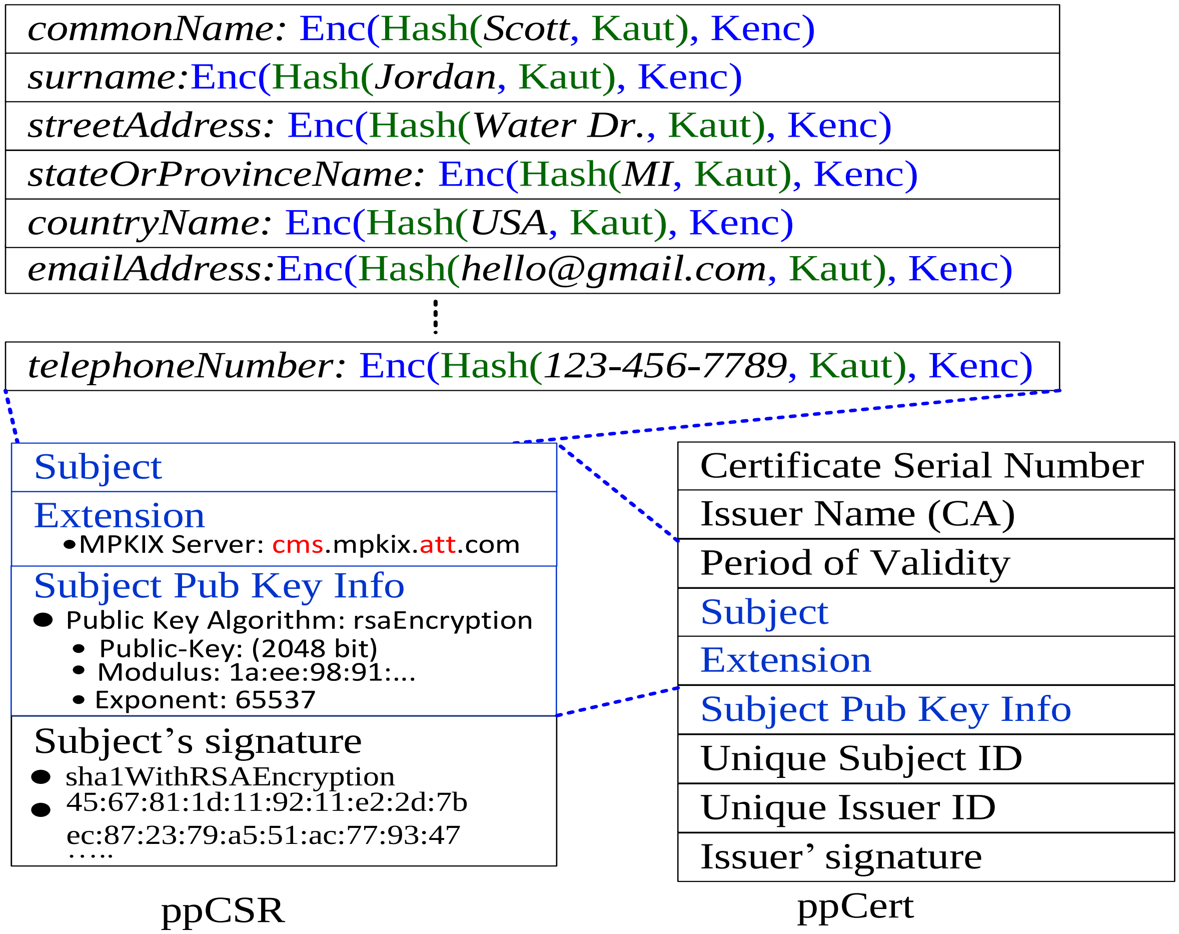

commonName: Enc(Hash(Scott, Kaut), Kenc)
surname:Enc(Hash(Jordan, Kaut), Kenc)
streetAddress: Enc(Hash(Water Dr., Kaut), Kenc)
stateOrProvinceName: Enc(Hash(MI, Kaut), Kenc)
countryName: Enc(Hash(USA, Kaut), Kenc)
emailAddress:Enc(Hash(hello@gmail.com, Kaut), Kenc)
telephoneNumber: Enc(Hash(123-456-7789, Kaut), Kenc)
Subject
Extension
•MPKIX Server: cms.mpkix.att.com
Subject Pub Key Info
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus: 1a:ee:98:91:...
Exponent: 65537
Subject's signature
sha1WithRSAEncryption
45:67:81:1d:11:92:11:e2:2d:7b
ec:87:23:79:a5:51:ac:77:93:47
.....
ppCSR
Certificate Serial Number
Issuer Name (CA)
Period of Validity
Subject
Extension
Subject Pub Key Info
Unique Subject ID
Unique Issuer ID
Issuer' signature
ppCert

Figure 5.3 The formats of ppCSR and ppCert.

ppCSR to an MPKIX-supported CA over a secure channel (e.g., TLS connection).

### 5.3.1.3 ppCert Issuance

The MPKIX-supported CA issues a privacy-preserving PKIX user certificate (ppCert) with its digital signature for each valid carrier-endorsed ppCSR from the CMS, as shown in the lower right part of Figure 5.3. It validates each ppCSR by verifying the digital signatures of both the CMS and the applicant in the ppCSR. The ppCert is then issued to the CMA via the CMS. Note that once the ppCert issuance succeeds, those two security keys ($K_{aut}$ and $K_{enc}$) associated with the ppCert are recorded in the CMS. They are further used to answer queries from IAS providers when the ppCert is used to access IASs, as described in Chapter 5.3.2.

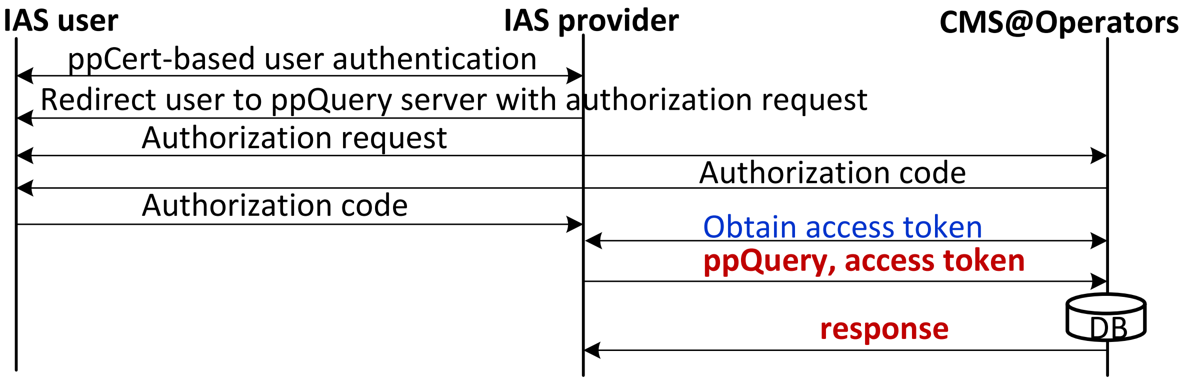


Figure 5.4 MPKIX: privacy-preserving query and verification of the user information.

#### 5.3.1.4 Compared with Conventional PKIX User Certificates

ppCert has two key advantages over conventional PKIX user certificates. First, the conventional certificate application process requires applicants to provide the CA with their user information, but the ppCert applicants do not need to take this action. The reason is that MPKIX leverages the user information that has been verified in the serving cellular network. Second, the conventional certificates, which carry user information in plain text, are delivered without the protection of secure channels[134], so the user information may be leaked; however, only the hash values of encrypted user information are given in the ppCert.

### 5.3.2 ppQuery: Privacy-preserving User Information Querying

ppQuery is a carrier-certified service that not only allows IAS providers to query/verify IAS user information but also protects IAS users from the leakage of user information. The ppQuery service comprises three parts: ppCert-based user acquisition, ppQuery access token acquisition, and carrier-certified user information querying, as shown in Figure 5.4. We elaborate on each of them below.

#### 5.3.2.1 ppCert-based User Acquisition

An IAS user can send his/her ppCert to an IAS provider and the provider verifies the ppCert based on the CA signature. This ppCert-based user acquisition between the IAS user and the IAS provider can be protected based on one of mainstream security protocols (e.g., HTTPS and SSL/TLS), since ppCert conforms to the PKIX standard, the authentication mechanism of which has been broadly supported in the mainstream security protocols. If the verification fails, the IAS provider may still offer the IAS user anonymous or unrestricted services.

#### 5.3.2.2 ppQuery Access Token Acquisition

The ppQuery service is provided based on the common OAuth [135] framework. To consume the service, the IAS provider needs to obtain an access token from CMS through the IAS user. As shown in Figure 5.4, the acquisition procedure of the access token is described below. First, the IAS provider obtains the IAS user's serving CMS server address (e.g., cms.mpkix.att.com) and a unique subject ID from its received ppCert, generates an authorization request including user information for a query, and then redirects the IAS user to the CMS with the authorization request. Second, upon the redirection, the IAS user logs onto the CMS server, reviews the authorization request, and decides if the authorization is granted. Third, given a granted authorization request, the IAS user obtains an authorization code from the CMS server and then forwards it to the IAS provider. Fourth, the IAS provider can receive a ppQuery access token for the IAS user from the CMS by presenting the authorization code to the CMS.

#### 5.3.2.3 Carrier-certified User Information Query

For each IAS user with a granted authorization request, the IAS provider can use the corresponding access token to query the CMS about the user's information via GSMA OneAPI [136], which is a set of standard APIs designed for external service providers to access cellular network services and user profiles. The CMS responds to the query in accordance with the policy of user-specific privacy protection. The key idea of the privacy protection is to allow an IAS user to specify an anonymization degree of user information in terms of which attributes (e.g., age) can be disclosed.

Moreover, a **minimum individual anonymization level** ($\mathbb{IA}_{min}$) is adopted for each IAS user to guarantee that the user's real identity cannot be discovered or narrowed down to a small group of possible candidates, even though adversaries collect all the user information that the user ever revealed to different IAS providers. Specifically, an IAS user's $\mathbb{IA}_{min}$ represents the minimum percentage of the users with the same disclosed user information as the user in the database of the cellular operator. Thus, for example, if $\mathbb{IA}_{min}$ is set to 20% for an IAS user, adversaries cannot discover the user's real identity but can only narrow down the user identity to a group of possible candidates that take a percentage no smaller than 20% of all the users. Notably, any modification on an anonymization degree that violates the desirable $\mathbb{IA}_{min}$ is denied.

In the following, we first introduce how to anonymize a given subject attribute in the ppCert certificate for a user and then present how an individual privacy protection, i.e., minimum individual anonymization level, spans multiple subject attributes.

**Anonymization of a Subject Attribute:** MPKIX anonymizes attribute data using the Domain Generalization Hierarchy (DGH) approach [137]. Given a subject attribute, $S$, and its value, $V_S$, there is an anonymization function $f_S : (V_S, n) \rightarrow V_S^n$, where $n$ lies in the range between 0 and $L_S - 1$, and $L_S$ indicates the number of anonymization levels for $S$. $S$ has $L_S$ different attribute values, namely $V_S^0, V_S^1, ..., V_S^{L_S-1}$. $V_S^0$ is equivalent to $V_S$ and indicates the complete attribute value, whereas $V_S^{L_S-1}$ provides only a minimum detail. Notably, the number of anonymization levels can vary with subject attributes. In some cases, there are only two anonymization levels: disclosed and undisclosed. Each IAS user is allowed to set their preferred number on the anonymization level of each subject attribute.

Consider two examples on the anonymization of subject attributes. The first example attribute is user address. Given $L_S = 4$, there are four different attribute values: $V_{Addr}^0$ = {State-City-Street-StreetNumber}, $V_{Addr}^1$ = {State-City-Street-***}, $V_{Addr}^2$ = {State-City-****-***}, and $V_{Addr}^3$ = {State-****-****-***}. The second one is cell number. Given $L_S = 3$, three different attribute values are generated as $V_{Phone}^0$ = 323-111-2222, $V_{Phone}^1$ = 323-111-****, and $V_{Phone}^2$ = 323-***-****.

**Minimum Individual Anonymization Level ($\mathbb{IA}_{min}$):** Although the anonymization level of each subject attribute can be customized by an IAS user, the user may not know which level is sufficiently secure. Moreover, the secure degree of each level depends on the disclosed information itself. For example, if an IAS user's first or last name is rarely used, adversaries may be able to narrow down the user's identity to a small group of candidates. Once more information is given from other attributes, the user's identity may be further inferred. As a result, the IAS user can be more interested in the anonymization levels that can have at least a certain percentage of the users with the same disclosed user information as the user so that adversaries cannot tell the user's identity among those users, which makes MPKIX less useful in practice.

To address the above concern, we propose a minimum individual anonymization level, $\mathbb{IA}_{min}$, to provide individuals with cross-attributes user privacy protection, thereby preventing adversaries from identifying the real user identities by analyzing all user information attributes that (s)he ever disclosed (partially or fully) to IAS providers. Specifically, $\mathbb{IA}_{min}$ is a configurable parameter indicating the minimum level of $\mathbb{IA}$ for each user; the $\mathbb{IA}$ is an individual anonymization factor representing the current anonymization degree of permitted information disclosure across attributes for individuals. The $\mathbb{IA}$ factor of a user $u$ is defined as:

$$\mathbb{IA}_u = \frac{\sum_{j=1}^{|DB|} \bigcap_{i=1}^{m} (V_{j,S_i}^{H_{u,i}} == V_{u,S_i}^{H_{u,i}})}{|DB|}$$

, where $m$ and $|DB|$ are the number of subject attributes and the number of users, respectively, in the database, and $H_{u,i}$ is the lowest anonymization level ever used by the user for the value of attribute $S_i$ in response to the queries of IAS providers. In other words, the numerator in the above equation indicates the number of users with the same disclosed values of all the attributes as the user $u$. Intuitively, the higher value the $\mathbb{IA}_u$ has, the more difficult it is for adversaries to identify the user's identity.

Consider an example to calculate $\mathbb{IA}$ for a user $u$ whose first name is John and birth year is 1951 in the Michigan Voter database with 121,489 qualified voters (see more details in Chapter 5.5). For the two subject attributes, first name and age, two anonymization levels are adopted; the former has the undisclosed and fully disclosed levels, whereas the latter is with the undisclosed level and a disclosed level on whether the user age is over 21. Assume that the user is willing to disclose both first name and age, the $\mathbb{IA}_u$ is calculated as $\frac{\text{3,766 (\#users whose first names are John} \bigcap \text{ages over 21)}}{\text{121,489 (\#voters in database)}} = 3.1\%$, which indicates that 3.1% of users in the database or more than 3,760 users have the same values of both subject attributes as the user $u$.

The $\mathbb{IA}_u$ is calculated for each query from the IAS provider or when any modifications are made to anonymization levels of subject attributes. Whenever $\mathbb{IA}_u$ is smaller than $\mathbb{IA}_{min,u}$, an alert is sent to the user and his/her approval is required. In this study, the default value of $\mathbb{IA}$ is set to 1.6% (see details in Chapter 5.6). To increase the diversity of applicable use scenarios, the current MPKIX prototype is designed to maximize the number of subject attributes without the highest anonymization level, i.e., the least information disclosure, for each user while satisfying his/her desirable $\mathbb{IA}_{min}$.

#### 5.3.2.4 Compared with Conventional User Information Verification

ppQuery not only offers IAS providers a reliable means to verify user information but also protects user privacy for the access of different IASs. It differs from conventional approaches of user information verification from two aspects. First, ppQuery allows IAS users to disclose verified user information based on different degrees of data anonymization. For example, it is unnecessary for a user to reveal his/her full birthday to Google during account registration since Google only needs to verify if the user is over 18 years old. Second, ppQuery allows IAS users to control information disclosure based on the $\mathbb{IA}$ factor so that the leakage of user identity can be prevented. With conventional approaches, an IAS user may inadvertently reveal different kinds of user information while accessing different IASs; it may allow an adversary to discover the user's identity and then keep track of his/her activities.

Note that we admit that ppQuery may fail to prevent the identity leakage in some cases, e.g., a user reveals an attribute value which is unique or ignores a privacy leakage alert and agrees to reveal critical information to IAS providers. Some data perturbation techniques may be adopted to address this problem. We leave this improvement to our future work.

### 5.3.3 paClaim: Privacy-aware ID Claim/Revocation Arbitration

We developed the paClaim mechanism to improve the efficiency of ID dispute resolution based on the ppQuery service. Figure 5.5 shows an overview of this mechanism involving two main procedures: (1) ID claimer pre-qualification and (2) order-preserving-encryption (OPE)-enabled ID Levenshtein Distance [138] comparison.

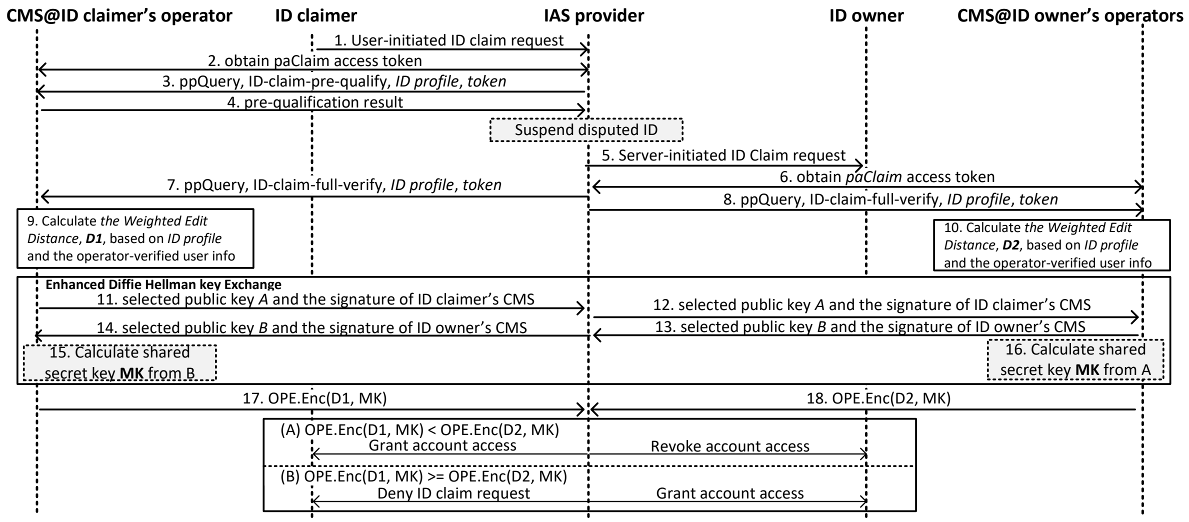

CMS@ID claimer's operator
ID claimer
IAS provider
ID owner
CMS@ID owner's operators
1. User-initiated ID claim request
2. obtain paClaim access token
3. ppQuery, ID-claim-pre-qualify, ID profile, token
4. pre-qualification result
Suspend disputed ID
5. Server-initiated ID Claim request
6. obtain paClaim access token
7. ppQuery, ID-claim-full-verify, ID profile, token
8. ppQuery, ID-claim-full-verify, ID profile, token
9. Calculate the Weighted Edit Distance, D1, based on ID profile and the operator-verified user info
10. Calculate the Weighted Edit Distance, D2, based on ID profile and the operator-verified user info
Enhanced Diffie Hellman key Exchange
11. selected public key A and the signature of ID claimer's CMS
12. selected public key A and the signature of ID claimer's CMS
13. selected public key B and the signature of ID owner's CMS
14. selected public key B and the signature of ID owner's CMS
15. Calculate shared secret key MK from B
16. Calculate shared secret key MK from A
17. OPE.Enc(D1, MK)
18. OPE.Enc(D2, MK)
(A) OPE.Enc(D1, MK) < OPE.Enc(D2, MK)
Grant account access
Revoke account access
(B) OPE.Enc(D1, MK) >= OPE.Enc(D2, MK)
Deny ID claim request
Grant account access

Figure 5.5 Privacy-aware ID claim/revocation arbitration mechanism.

#### 5.3.3.1 ID Claimer Pre-qualification

The ID claimer needs to pass the ID claimer pre-qualification before initiating an ID claim/ revocation request to the IAS provider. It can filter out unnecessary or malicious ID claim/revocation requests by examining whether the carrier-verified user information of the ID claimer is equivalent to those of the disputed ID to some extent. The IAS provider first selects some subject attributes (e.g., the first and last names) for pre-qualification and the ID claimer then needs to prove that his/her name values are similar enough to those of the disputed ID.

In this study, we use the ID Levenshtein Distance (*IDLevDist*) to quantify the similarity; the Levenshtein Distance is the minimum number of single-character edits required to change one word into the other (e.g., the Levenshtein distance between "Alex" and "Alexa" is 1). Notably, for certain subject attributes (e.g., address), different values may still represent the same information (e.g., HK and Hong Kong), and additional formatting functions (e.g., translating a user-entered address to a USPS-suggested address) for attribute values are thus required (more details will be discussed in Chapter 5.5).

Specifically, the pre-qualification process works as follows. First, the ID claimer provides the IAS provider with the access token of a ppQuery service. Second, the IAS provider sends a query to the CMS server of the ID claimer using the access token. The query message comprises three key elements: (1) a subset of provider-selected subject attributes and values for the disputed ID, $Owner = \{S_1, V_1, S_2, V_2, ..., S_n, V_n\}$, where $S_i$ is the *i*th subject attribute and $V_i$ is the value of $S_i$; (2) a set of Levenshtein distance weights, $W = \{W_1, W_2, ..., W_n\}$, where $W_j$ is the weight of the Levenshtein distance between $V_j$ and the ID claimer's value for $S_j$; (3) the maximum of the *IDLevDist* values that are allowed to pass the ID pre-qualification. *IDLevDist* is calculated as $\sum W_i * LevDist(V_i^{claimer}, V_i^{owner})$. Note that, to prevent the IAS provider from inferring the ID claimer's user information, it is suggested that the maximum number of the compared attributes is set to 3. Moreover, the recommended $Owner$ contains the first name, the last name, and an additional provider-selected subject attribute (e.g., address).

Third, the CMS first checks if the computed *IDLevDist* exceeds the maximum value and then sends back the pre-qualification result to the provider. If the ID claimer passes the pre-qualification, the IAS provider initiates the ID claim/revocation arbitration and may temporarily suspend the disputed ID accordingly.

#### 5.3.3.2 OPE-enabled IDLevDist Comparison

After the ID claimer is pre-qualified for the ID claim/revocation arbitration, the IAS provider initiates it by sending a ppQuery message for full ID verification to the ID claimer's CMS server and the ID owner's (Steps 5-8). Then, each of them computes its own *IDLevDist* (Steps 9-10). Similar to the ppQuery message previously introduced in the pre-qualification, the ppQuery message comprises $Owner$ and $W$. But, there are two major differences. First, the number of subject attributes specified in $Owner$ is not limited. Second, the maximal *IDLevDist* that is allowed to pass the verification is not specified.

Given those two *IDLevDist* values, the IAS provider can easily determine which of the ID claimer and the ID owner has more operator-verified user information corresponding to the disputed ID. The one with a shorter distance (i.e., smaller *IDLevDist* value) wins and is allowed to access or revoke it. However, the *IDLevDist* value in plain-text may allow the IAS provider to infer additional user information of the ID claimer and the ID owner. For example, the *IDLevDist* given by the ID owner's CMS indicates how close the user information that the ID owner left on the IAS provider is to the operator-verified information of the disputed ID.

To prevent this inference attack, the ID Levenshtein distances computed by the CMSs are not directly returned to the IAS provider; instead, only the distances encrypted by the OPE (Order Preserving Encoding) method [139], which is an encryption algorithm ensuring the order of plain-text numbers to be equal to that of encrypted numbers, are delivered for the comparison. For the secure distribution of the encryption key shared between the CMSs, the Diffie Hellman Key Exchange (DHKE) protocol [140] was adopted; DHKE is a method of enabling the secure exchange of cryptographic keys over public channels. By exchanging security parameters (e.g., two DHKE public keys $A$ and $B$), DHKE enables the CMSs of ID claimer and ID owner to derive a shared secret key $MK$ using their DHKE private keys for the further OPE-based ID Levenshtein distance encryption (Steps 15-18). With OPE-based ID Levenshtein distance comparison, the IAS provider can identify the one with a shorter distance while preserving the privacy of the ID owner.

Note that current paClaim service only supports one ID claimer in each ID claim/revocation arbitration; if there is more than one user claiming the same IAS ID, multiple arbitrations are required. For example, by assuming that IAS users A and B both claim the ownership of a disputed ID, whose owner is user C currently, and the IAS provider receives A's request first, the IAS provider arranges the first arbitration between users A and C, and then does the second arbitration between user B and the winner of the first arbitration.

#### 5.3.3.3 Compared with Conventional ID Claim/revocation Mechanisms

The *paClaim* has two key advantages. First, the *paClaim*-based ID claim arbitration can be done in seconds, but the existing mechanisms may take several days or even longer. Second, the *paClaim* does not require current ID owners or claimers to disclose additional operator-verified user information to the IAS provider, whereas current mechanisms (e.g., uploading government-issued ID documents) can inevitably cause an excessive information disclosure.

## 5.4 Security Analysis

In this subchapter, we analyze the desirable security guarantees provided by MPKIX and the common attacks against which MPKIX can defend.

### 5.4.1 Security Guarantees

**Integrity and non-repudiation:** MPKIX leverages the merits of current PKIX practice and cellular network security to achieve both data integrity and non-repudiation of the ppCert certificate. To obtain a ppCert certificate, an IAS user needs to create a ppCSR request and attach his/her digital signature. After validating the user's information and digital signature, the serving operator endorses the ppCSR with its digital signature. The operator's signature allows MPKIX-supported CAs to validate the user's ppCSR and digitally sign it. Thus, the accuracy of the user information is guaranteed by the serving operator, and the data integrity is then guaranteed by both the cellular symmetric cryptography with the key $K_{aut}$ (see Chapter 5.3.1) and PKIX asymmetric cryptography with the CA's private key; these two keys are hardly to be stolen. Regarding the non-repudiation property, in many countries/areas, e.g., the European Union and the U.S., previously described digital signatures have legal significance [141]. Therefore, IAS users and operators cannot dispute the authorship/validity of their digital signatures.

**Privacy:** MPKIX provides IAS users with a multitude of privacy protection. First, MPKIX allows a user to freely determine which subject attributes in the ppCert are disclosed to the IAS provider through the ppQuery service. Since the attribute values in the ppCert are hashed and encrypted, neither the IAS provider nor adversaries can infer the values without the encryption and integrity keys (i.e., $K_{aut}$ and $K_{enc}$). Second, MPKIX guarantees that adversaries cannot infer the identity of an IAS user or narrow it down to a small group of possible candidates. Third, MPKIX allows an IAS user to create his/her IAS user account with false information due to privacy concerns; however, if the IAS user suffers from ID theft attacks, where an adversary impersonates the user's identity, MPKIX empowers the IAS user to claim/revoke the impersonated ID without revealing more verified user information to the IAS provider.

**Accountability:** MPKIX allows law enforcement authorities to discover the real identity of an IAS provider or an IAS user, when an IAS-based cyber attack/crime occurs. The IAS provider's identity can be revealed from its CA-signed PKIX server certificate, whereas although the IAS user's identity may not be disclosed in his/her ppCert, the law enforcement authorities can discover it from the user's serving operator, which can be identified through the CA.

### 5.4.2 MPKIX's Resilience Against Possible Attacks

We next analyze the resilience of those three MPKIX services against various cyberattacks and discuss how MPKIX deals with other possible attacks (e.g., stealing mobile phones) beyond the adversary model of this study, where the Dolev-Yao model [125] is considered (see details in Chapter 5.2).

**Assumptions:** Two assumptions are made. First, we assume that the cellular infrastructure is secure, and operators deploy security patches timely. Second, we assume that all security and service protocols (e.g., TLS and OAuth) used by MPKIX are properly configured and with recommended security patches (e.g., eliminating obsolete TLS configurations, such as ECDHE with custom curves [142]).

**Notations:** We denote an IAS user with a mobile phone having Certificate Management Application (CMA) installed by $\mathbb{A}$, the Certificate Management Server (CMS) by $\mathbb{S}$, the MPKIX-supported certificate authority by $\mathbb{CA}$, the IAS provider by $\mathbb{I}$, the encryption function by $Enc$, the decryption function by $Dec$, the function producing message authentication code by $Mac$, the signature function by $Sig$, the private key by $Pri$, and the public key by $Pub$.

**ceIssuance Analysis:** We model the ceIssuance service and analyze it in terms of security as follows:

1. $\mathbb{A}$ and $\mathbb{S}$ conduct EAP-SIM/AKA-based mutual authentication and obtain two shared security keys, $K_{enc}$ and $K_{aut}$.

2. $\mathbb{A}$ sends $Enc(ppCSR|Mac(ppCSR, K_{aut}), K_{enc})$ to $\mathbb{S}$, where '|' is a concatenation operator.

3. $\mathbb{S}$ decrypts the encrypted $ppCSR$ and verifies the MAC using $K_{enc}$ and $K_{aut}$, respectively. Given a valid $ppCSR$, $\mathbb{S}$ obtains the verified user information from the HSS through Diameter over TLS and compares it to the user information in $ppCSR$.

4. $\mathbb{S}$ sends $Enc(ppCSR|Sig(ppCSR, Pri_S), Pub_{CA})$ to $\mathbb{CA}$.

5. $\mathbb{CA}$ verifies $\mathbb{S}$'s signature using $Pub_S$. If valid, $\mathbb{CA}$ generates a CA-signed ppCert with its signature $Sig(ppCert, Pri_{CA})$ and sends $Enc(ppCert), Pub_S)$ to $\mathbb{S}$.

6. $\mathbb{S}$ sends the CA-signed $Enc(ppCert, Pub_A)$ to $\mathbb{A}$.

At Step 1, the messages exchanged between $\mathbb{A}$ and $\mathbb{S}$ are plain-text. Adversaries can thus intercept and synthesize those messages to launch Man-in-the-Middle (MitM) attacks. However, Alt *et al.* [143] have proven that the cellular AKA protocol with unique server identifiers attains the properties (e.g., state-confidentiality and soundness) that can defend against MitM attacks even in the presence of corrupted servers. Thus, adversaries cannot compromise the mutual authentication, and further infer $K_{enc}$ and $K_{aut}$.

At Steps 2-3, adversaries may apply for a CA-signed PKIX user credential on behalf of $\mathbb{A}$ by launching an impersonation attack. However, without $K_{enc}$ and $K_{aut}$, the adversaries cannot generate a valid request message, $Enc_(ppCSR|Mac(ppCSR, K_{aut}), K_{enc})$.

At Steps 4-6, all the message exchanges of $ppCSR$ and $ppCert$ are provided with confidentiality and integrity protection. Thus, without the private keys of $\mathbb{CA}$ and $\mathbb{S}$, adversaries cannot decrypt any intercepted ciphertext messages or fabricate digital signatures of $\mathbb{CA}$ and $\mathbb{S}$.

**ppQuery Analysis:** We model the ppQuery service and do security analysis on it below.

1. $\mathbb{A}$ sends *Client Hello* to $\mathbb{I}$.

2. $\mathbb{I}$ sends *Server Hello*, *Certificate*, *Server Key Exchange*, *Certificate Request*, and *Server Hello Done* to $\mathbb{A}$.

3. $\mathbb{A}$ sends **ppCert**, *Server Key Exchange*, *Certificate Verify*, *Change Cipher Spec*, and *Finished* to $\mathbb{I}$.

4. $\mathbb{I}$ sends *Change Cipher Spec* and *Finished* to $\mathbb{A}$.

5. $\mathbb{I}$ obtains the CMS address (i.e., $\mathbb{S}$) and the subject ID of $\mathbb{A}$ from **ppCert** for further ppQuery operations, which verify correctness of the user information provided by $\mathbb{A}$.

6. $\mathbb{I}$ initiates an OAuth-based ppQuery access token acquisition with $\mathbb{S}$ over TLS.

7. $\mathbb{I}$ sends an encrypted ppQuery message, $Enc(ppQuery|Token|Mac(ppQuery|Token, K_{int_{TLS}}), K_{enc_{TLS}})$, to $\mathbb{S}$, where $K_{enc_{TLS}}$ and $K_{int_{TLS}}$ are the encryption key and integrity key derived from the establishment of TLS connection between $\mathbb{I}$ and $\mathbb{S}$.

8. $\mathbb{S}$ sends an encrypted response, $Enc(Response|Mac(Response, K_{int_{TLS}}), K_{enc_{TLS}})$ to $\mathbb{I}$.

At Steps 1-4, $\mathbb{A}$ and $\mathbb{I}$ establish a TLS connection while authenticating each other. In particular, $\mathbb{A}$ provides $\mathbb{I}$ with *ppCert* during the TLS connection establishment. Adversaries may intercept and synthesize those handshake messages including *ppCert* to launch MitM attacks, infer the verified user information, or conduct long-term user tracking attacks. However, MPKIX is immune to these attacks due to the following three reasons. First, according to a recent NSA (National Security Agency) report [142], an established TLS connection is considered as a secure communication channel against various MitM attacks (e.g., MitMProxy and SSLSplit attacks) when obsolete TLS configurations are avoided. Second, the values of subject attributes in *ppCert* are encrypted hashed values (see Figure 5.3), and the used keys, $K_{enc}$ and $K_{aut}$, are hardly obtained from the ceIssuance service. Third, the real-world risk of ppCert-based user tracking attacks is limited since MPKIX guarantees that adversaries cannot discover the real identity of a ppCert owner or narrow it down to several possible individuals.

At Steps 5-6, adversaries may attempt to launch various attacks against token acquisition and usage, but Fett *et al* [144] have proven that the OAuth protocol establishes strong authorization, authentication, and session integrity guarantees, which can well defend potential attacks.

At Steps 7-8, $\mathbb{I}$ sends a ppQuery message with the granted access token to $\mathbb{S}$, and $\mathbb{S}$ replies a response to $\mathbb{I}$ based on $\mathbb{A}$'s privacy protection setting. To defend against possible cyberattacks, the ppQuery request and response messages are protected with confidentiality and integrity using those two keys, $K_{enc_{TLS}}$ and $K_{int_{TLS}}$.

**paClaim Analysis:** The paClaim service is comprised of three ppQuery request-response transactions over TLS for the qualification examination of the ID claimer, the collection of the OPE-encoded

ID Levenshtein distance from the ID claimer's CMS, and that from the ID owner's CMS, respectively. Since the user information verification and message exchange in the ppQuery service have been analyzed, we here focus on the security analysis of deriving the shared OPE security keys at CMSs (i.e., Steps 11-18 in Figure 5.5).

1. $\mathbb{S}_{\text{Claimer}}$ selects a DHKE (Diffie Hellman Key Exchange) public key, $X_{S1}$ and a DHKE private key, $Y_{S1}$, generate a signature, $Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ for $X_{S1}$ using its $Pri_{S_{\text{Claimer}}}$, and sends $X_{S1}|Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ to $\mathbb{I}$.

2. $\mathbb{I}$ forwards $X_{S1}|Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ to $\mathbb{S}_{\text{Owner}}$.

3. $\mathbb{S}_{\text{Owner}}$ selects a DHKE public key, $X_{S2}$ and a DHKE private key $Y_{S2}$, generate a signature, $Sig(X_{S2}, Pri_{S_{\text{Owner}}})$, for $X_{S2}$ using its $Pri_{S_{\text{Owner}}}$, and sends $X_{S2}|Sig(X_{S2}, Pri_{S_{\text{Owner}}})$ to $\mathbb{I}$.

4. $\mathbb{I}$ forwards $X_{S2}|Sig(X_{S2}, Pri_{S_{\text{Owner}}})$ to $\mathbb{S}_{\text{Claimer}}$.

5. $\mathbb{S}_{\text{Claimer}}$ calculates the shared OPE security key using DHKE algorithm[2] as: $(X_{S2})^{Y_{S1}} \bmod q$, where $q$ is a prime number shared by all CMSs MPKIX.

6. $\mathbb{S}_{\text{Owner}}$ calculates the shared OPE security key using DHKE algorithm as: $(X_{S1})^{Y_{S2}} \bmod q$.

Different from the ppQuery service, where outside adversaries are considered, the paClaim service may suffer from an inside adversary, the IAS provider (i.e., $\mathbb{I}$), which may be interested in discovering the plain-text ID Levenshtein distances from $\mathbb{S}_{\text{Claimer}}$ and $\mathbb{S}_{\text{Owner}}$ to infer more user information. Thus, it can motivate $\mathbb{I}$ to compromise the procedure of the OPE security key exchange by launching an MitM attack [145]. To this end, $\mathbb{I}$ first selects two DHKE key pairs: (1) $X_{I\leftrightarrow S_{\text{Claimer}}}$ and $Y_{I\leftrightarrow S_{\text{Claimer}}}$ and (2) $X_{I\leftrightarrow S_{\text{Owner}}}$ and $Y_{I\leftrightarrow S_{\text{Owner}}}$, intercepts $X_{S1}$ and $X_{S2}$, and then sends $X_{I\leftrightarrow S_{\text{Claimer}}}$ and $X_{I\leftrightarrow S_{\text{Owner}}}$ to $\mathbb{S}_{\text{Claimer}}$ and $\mathbb{S}_{\text{Owner}}$, respectively. In the unmodified DHKE protocol, $\mathbb{I}$ can obtain two shared OPE security keys: one is for $\mathbb{I}$ and $\mathbb{S}_{\text{Claimer}}$ (i.e., $(X_{S1})^{Y_{I\leftrightarrow S_{\text{Claimer}}}} \bmod q$), and the other is for $\mathbb{I}$ and $\mathbb{S}_{\text{Owner}}$ (i.e., $(X_{S2})^{Y_{I\leftrightarrow S_{\text{Owner}}}} \bmod q$), and further discover the plain-text ID Levenshtein distances.

[2] The DHKE algorithm is based on the discrete logarithm problem; given $\alpha$ and $a$, find $b$ so that $\alpha^b = a$.

However, the paClaim service is immune to the above MitM attack. This is because $S_{\text{Claimer}}$ and $S_{\text{Owner}}$ attach their digital signatures while transmitting $X_{S1}$ and $X_{S2}$ to $\mathbb{I}$ at Steps 1 and 3, respectively. Without their private keys, $Pri_{S_{\text{Claimer}}}$ and $Pri_{S_{\text{Owner}}}$, $\mathbb{I}$ cannot produce the digital signatures and have them accepted the fabricated DHKE keys.

**Other potential attacks:** We next discuss how MPKIX defends against several potential attacks beyond the Dolev-Yao adversary model.

- *(U)SIM Card Compromising Attacks:* By compromising mobile users' (U)SIM cards, adversaries can apply for ppCerts on behalf of them. There have been several SIM-based attacks, which include inferring the secret key $K_i$ by abusing A3 algorithm COMP128v1 [146], rooting SIM cards via insecure OTA [147], and launching a SIM swap attack [148]. The root causes mainly lie in improper configurations of the cellular network [146], security flaws from SIM card manufacturers [147], and social engineering attacks [148]. Most of these attacks can be addressed with proper configurations and timely security patches.

- *Mobile Phone Compromising Attacks:* An adversary may infer user information from a pre-compromised mobile phone by eavesdropping on the issuance of carrier-endorsed PKIX user certificate. However, MPKIX is immune to this attack since no plain-text user information is sent over the air. Moreover, such attack requires root privilege of the compromised phone, which has been shown with a significant technical challenge [149, 150, 151].

- *Stealing Phones:* If an IAS user's phone is stolen, his/her PKIX user credential may be abused. However, this problem can be largely mitigated by an action that the user promptly updates the public CRL (Certificate Revoke List) to revoke his/her credential. Furthermore, compared with traditional cryptographic tokens (e.g., YubiKey), the MPKIX phone-based cryptographic tokens provide better security of user credentials. Specifically, modern smartphones support a variety of bio-based security mechanisms, such as fingerprint and facial recognition, which can prevent adversaries from abusing user credentials on stolen/lost phones.

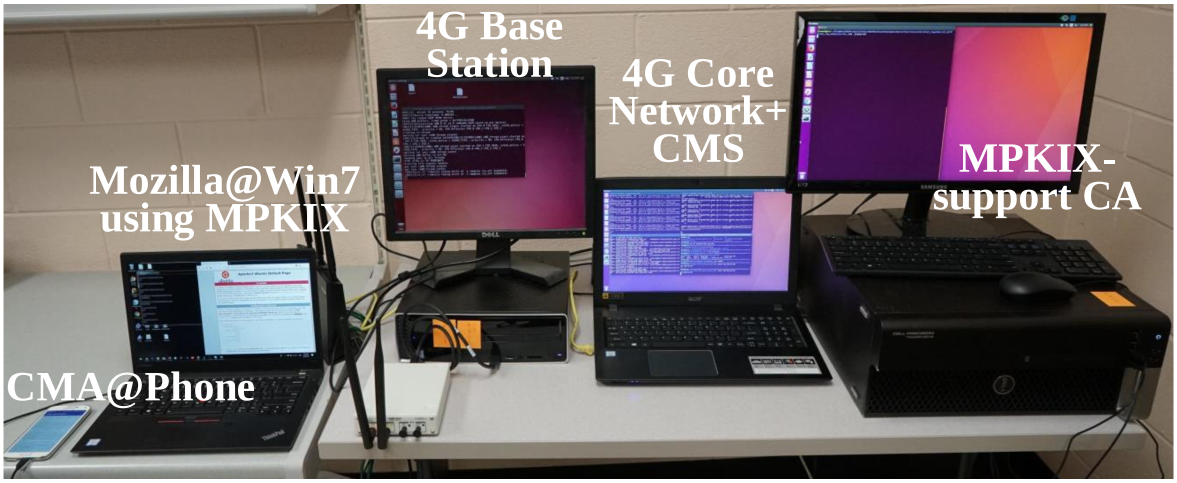


Figure 5.6 MPKIX prototype.

## 5.5 Implementation of MPKIX

Figure 5.6 illustrates three key entities of the MPKIX prototype: CMA on a mobile phone, an MPKIX-enabled 4G LTE infrastructure with CMS, and an MPKIX-supported CA. Each of them is elaborated below. Notably, the secure communications between MPKIX-supported network elements were enabled by TLSv1.2 using the ciphersuite of ECDHE-RSA-AES256-SHA, and the cryptographic key operations were implemented using the OpenSSL [152] library.

**CMA** was written in Java and implemented on four low/mid-end smartphones including Samsung S2 (2011), Samsung S5 (2014), Sony Xperia Z (2013), and Google Pixel XL (2016). Notably, successfully deploying CMA on these old phone models with fewer computing resources than modern ones indicates that CMA works for most phone models. CMA uses the credential services provided by an IsoApplet [153] to generate public and private keys, prepare ppCSR, and obtain/maintain CA-issued ppCert. Moreover, to increase the applicability of the MPKIX credential service, the PKCS#11 (Public Key Cryptography Standards [154]) interface, which is a standard platform-independent API to access diversified cryptographic tokens and has been broadly supported by many operating systems, was implemented on CMA. It enables CMA to transform an IAS user's phone to a cryptographic token with the PKCS#11 interface.

**MPKIX-enabled 4G LTE infrastructure** was set up using the SDR(software-defined radio)-based OpenAirInterface (OAI) platform, which comprises a 4G LTE core network and a base station. The core network was deployed on a Lenovo desktop with Intel i7-9700k and 16GB RAM, whereas the base station was built on a PowerSpec desktop with Intel i7-9700K and 16GB RAM connecting to an Ettus USRP B210. In the core network, a CMS server supporting *ceIssuance*, *ppQuery*, and *paClaim* services was deployed. We next introduce implementation details about CMS.

- **CMS services.** Three MPKIX services were implemented: (1) the *ceIssuance* service used OpenSSL [152] to implement cryptographic key operations, and employed Node-diameter [155], a diameter protocol over TLS, to enable secure communications with HSS; (2) the *ppQuery* service was implemented on top of oauth2-server [156] and enabled to support OAuth (using scribejava-6.9.0 [157]) and OneAPI (using an open GSMA OneAPI library [136]); and (3) the *paClaim* service used the Boost Algorithm [158] and Fast OPE [159] to calculate the ID Levenshtein distances and perform the order-preserving encryption, respectively.
- **CMS database.** The SQL database was built on top of the CryptoDB library [160] to store and anonymize user information obtain from HSS. To emulate real mobile user data, the HSS's database contained the information of 120,531 users, which was purchased from a voter registration database [161] with 120,531 voters. The user attributes included name, gender, birthday, address, and phone number. In the current prototype, the data anonymization levels for each attribute are as follows (the information specified at each level was disclosed): (1) *name*: two levels (L0: full name; L1: none); (2) *gender*: two levels (L0: gender; L1: none); (3) *birthday*: six levels (L0: year, month, and day; L1: year and month; L2: year; L3: small age ranges ({0-17, 18-40, 41-60, 61-80, >80}); L4: large age ranges ({0-40, 40-80, >80}); L5: none); (4) *phone number*: four levels (L0: phone number; L1: last seven digits; L2: last four digits; L3: none). (5) *addresses*: five levels (L0: street number, street name, city, and state; L1: street name, city, and state; L2: city and state; L3: state; and L4: none). Notably, to tackle the different addresses that have the same legal semantics (e.g., HK and Hong Kong), we will reformat all address inputs to unified ones using Google Geocoding API [162] prior to the data processing.

Moreover, the value of $\mathbb{IA}_{min}$ was set as $\frac{2{,}000}{120{,}531} \simeq 1.66\%$ for all the users; it indicates that the number of mobile users who have the same disclosed user information cannot be smaller than 2,000. With the given $\mathbb{IA}_{min}$, the MPKIX prototype can automatically adjust the number of subject attributes and the anonymization level of each attribute, if needed, for each user. In particular, to improve the diversity of applicable use scenarios, the current MPKIX prototype is designed to maximize the number of a user' subject attributes that do NOT apply the highest anonymization levels while ensuring desirable $\mathbb{IA}_{min}$

**MPKIX-supported CA** was written in Java and used the bouncycastle-v1.6 [163] library, a lightweight cryptography library, to validate ppCSR and generate ppCert. It was deployed on a Dell 5810 precision tower.

## 5.6 Evaluation of MPKIX

In this subchapter, we evaluated the effectiveness and performance of the three key MPKIX services.

### 5.6.1 ceIssuance

We evaluated the ceIssuance service by two metrics: (1) certificate issuance time, which is the time required by an IAS user to generate a pair of public and private keys, prepare a ppCSR request, and obtain a ppCert certificate on a mobile phone, but does not include the time required by the user to input user data; and (2) certificate issuance rate, which indicates the maximum number of PKIX user certificates issued by the MPKIX prototype per unit of time.Finally, the overhead of supporting varied anonymization levels was evaluated.

**Experimental settings:** For the issuance time, the experiment was intentionally conducted on a low-end mobile phone, Samsung Galaxy S5 equipped with Qualcomm Snapdragon801 CPU and 2GB RAM, and had 20 runs. For the issuance rate, a program was developed to keep sending ppCSR to the MPKIX infrastructure. The experiment lasted for one hour, where each new ppCSR was sent right after the ppCert of the last ppCSR, if there was any, was received.

**Experimental results:** Figure 5.7 plots the CDF of the time spent on the overall issuance and its three events including ppCSR preparation (T1), ppCSR validation (T2), and ppCert generation (T3). We have three observations. First, an IAS user can obtain a ppCert certificate within 5 s

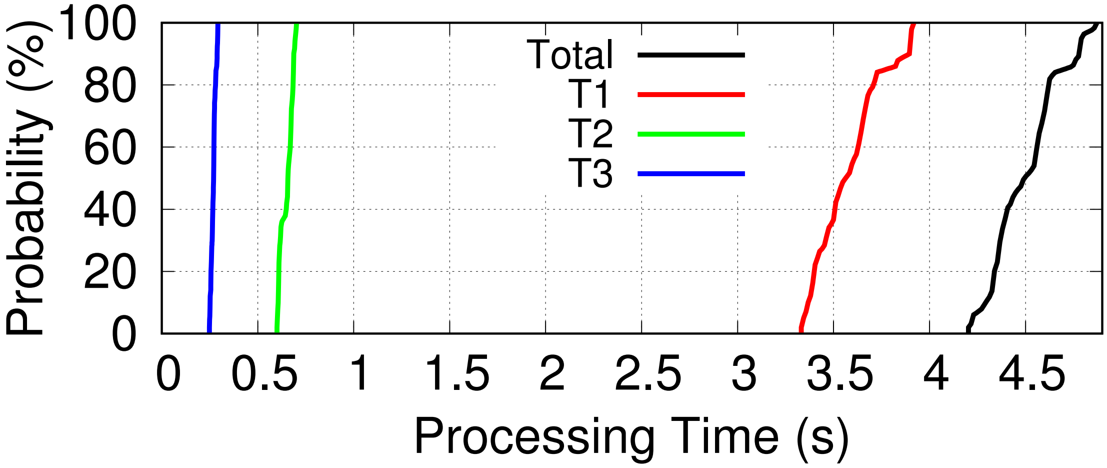


Figure 5.7 PKIX user certificate issuance time. *T1* is the time between the generation of a public/private key pair and that of ppCSR; *T2* is the time between when ppCSR is sent by the phone and when the carrier-endorsed ppCSR is received by the CA; *T3* is the period from the time right after the end of T2 to that ppCert is received by the phone.

even on a low/medium-end smartphone, but typical CAs, e.g., GlobalSign and DigiCert [164, 165], require several days for a certificate application. Second, T1 ranges from 3.3 s to 3.8 s, whereas T2 and T3 take only 1.2-1.7 s. The main reason is that the IsoApplet used by the current CMA prototype required more actions to carry out the credential service functions because of its data length limitation, no larger than 256 bytes, for communicating with external applications. The usage of the IsoApplet, a lightweight Java applet offering credential services, is to support low/medium-end resource-constrained phones. Notably, the maximum values of the observed instant RAM and CPU usages in the experiment for the CMA are 57 MB and 27%, respectively. Our experiment results show that even on a low/medium-end smartphone, a user is still able to obtain his/her CA-issued PKIX user certificate within less than 5 seconds, whereas the typical CAs, e.g., GlobalSign and DigiCert, require several days [164, 165].

Figure 5.8 plots the issuance rate of the number of ppCerts issued per minute. It is observed that the MPKIX infrastructure issued around 130,000 ppCerts per minute and issued a total of 7.82 million ppCerts without any significant variance within an hour. It shows that the MPKIX infrastructure has a stable issuance performance.

For the overhead of supporting varied anonymization levels, CMS produces the values of all

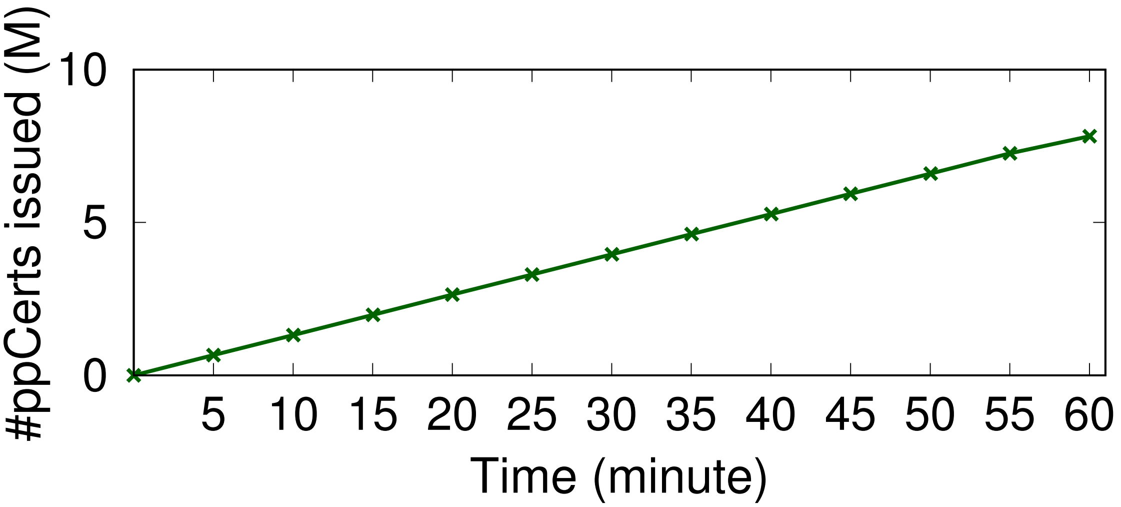


Figure 5.8 #ppCerts issued by the MPKIX infrastructure.

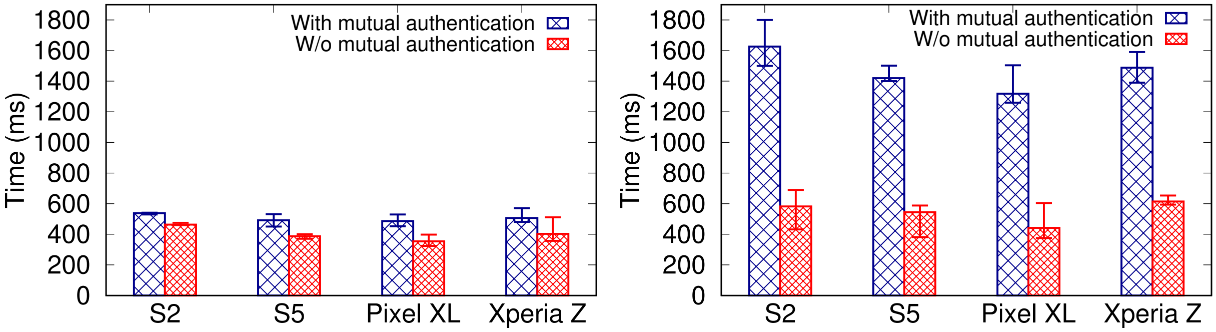
the subject attributes based on given anonymization levels for each ppCSR request before sending the carrier-endorsed ppCSR to CA. For example, four values are produced for the *phone number* attribute with four anonymization levels. In this experiment, we used a global variable $\alpha$ as the maximum anonymization level for all the subject attributes and then examined whether varied anonymization levels would affect T2 (ppCSR validation time) by varying $\alpha$. The experiment was conducted with 20 runs for each level.

The result shows that T2 was increased by 42 ms, 48 ms, 54 ms, and 59 ms when $\alpha$ was set to 1, 2, 3, and 4, respectively, compared with the case with $\alpha = 0$. Although T2 is observed to increase with anonymization level, producing values for all the anonymization levels is conducted only once at CMS for each carrier-endorsed ppCSR, and does not affect the subsequent ppQuery response times regardless of user privacy settings.

### 5.6.2 ppQuery

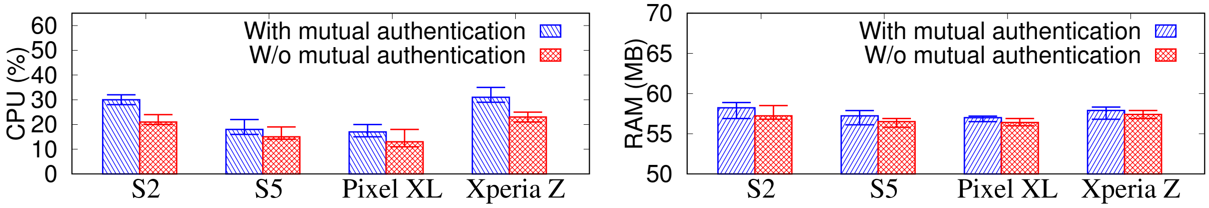
We evaluated the ppQuery service based on not only correctness, but also two metrics: (1) IAS access time, which is the time required by an IAS user to establish a secure TLS connection with an IAS server using his/her CA-issued ppCert; (2) IAS query time, which is the time spent by the IAS server on receiving a query response after submitting the query.

**Experimental settings:** We randomly selected two IAS users from the user database at HSS: one



(a) WebKit on phones (b) Mozilla on computer

Figure 5.9 The IAS access time (max/med/min) involving the establishment of a TLS connection with or without mutual authentication varies with MPKIX-enabled phones and a computer connecting to those phones.

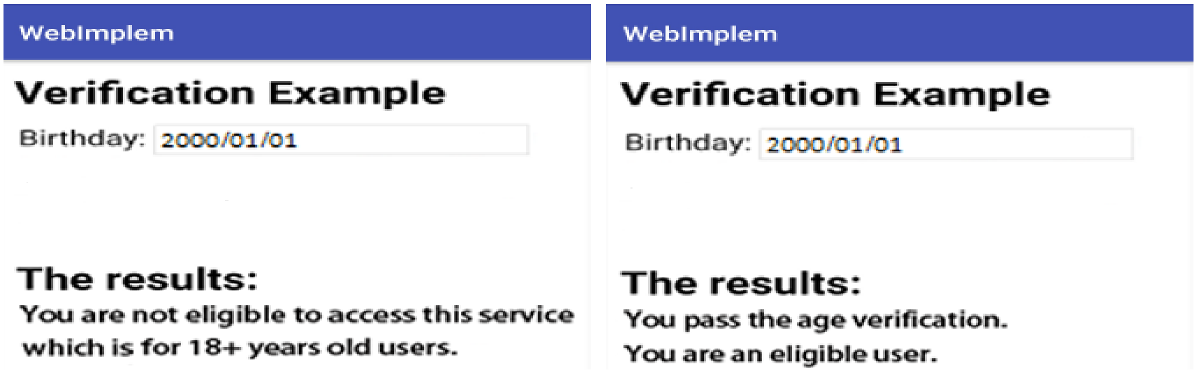


(a) CPU usage (b) RAM usage

Figure 5.10 The peak CPU and RAM usage statistics (max/med/min) of the MPKIX-enabled phones for the ppQuery service.

user is older than 18 years old, whereas the other is not. After obtaining their CA-issued ppCerts through MPKIX, these two selected users attempted to connect with an IAS server, which allowed only users older than 18 years old, using browsers on phones and computers. During the connection of each user, the IAS server examined the age eligibility of the user by querying the CMS server and then determined whether the user is allowed to have the access. The experiment was conducted with 20 runs.

**Experimental results:** Figure 5.9 plots the statistics of the IAS access time for the WebKit browser on different MPKIX-enabled phones and the Mozilla browser on a Windows computer connecting to those phones for the MPKIX service. It is observed that the Mozilla browser requires 1.5s on average for the IAS access time On the contrary, the Webkit browser takes only 0.5s on average. The reason is that the WebKit can access the CMA locally on the phones, whereas the Mozilla cannot. Moreover, compared to the case without TLS mutual authentication as shown in Figure 5.10, the


WebImplem

Verification Example

Birthday: 2000/01/01

The results:
You are not eligible to access this service which is for 18+ years old users.

(a) User: no older than 18.

WebImplem

Verification Example

Birthday: 2000/01/01

The results:
You pass the age verification.
You are an eligible user.

(b) User: older than 18.

Figure 5.11 The IAS connection results based on the ppQuery.

peak CPU and RAM usages are increased by 2-10% and 1-2 MB, respectively.

Figure 5.11 confirms that the IAS server successfully verified the ages of the IAS users using the ppQuery service. The result showed that the IAS took 1.2s averagely, where 0.5s IAS access time and 0.7s IAS query time, on the query of a single attribute. Notably, it was observed that no full birthday information was returned to the IAS server in the experiment.

### 5.6.3 paClaim

We finally evaluated the effectiveness and performance of the *paClaim* service. Three performance metrics were used: (1) $T_{pre}$, the time required to perform the ID pre-qualification (Steps 1-4 in Figure 5.5); (2) $T_{dis}$, the time required to calculate the full ID Levenshtein distances (Steps 5-10 in Figure 5.5); (3) $T_{ope}$, the time required to perform the order-preserving encryption with key exchange (Steps 11-18 in Figure 5.5).

**Experimental settings:** We randomly selected 11 users from the user database at HSS; the first 10 users were assumed to be the victims of an ID theft attack and denoted as benign users, whereas the last user was an attacker of the ID theft. We obtained ppCerts for all the users through MPKIX. On the IAS server, the attacker created 10 accounts impersonating those 10 benign users, respectively, with their first and last names. An ID claim/revocation arbitration was performed for each benign user. For the ID claim pre-qualification at the IAS server, $Owner$ comprised two subject attributes {first name, last name}, $W = \{1, 1\}$ indicated the IDLevDist weights of those two attributes, and the maximal *IDLevDist* value was set to 5. For the ID claim full verification, $Owner$ and $W$ were

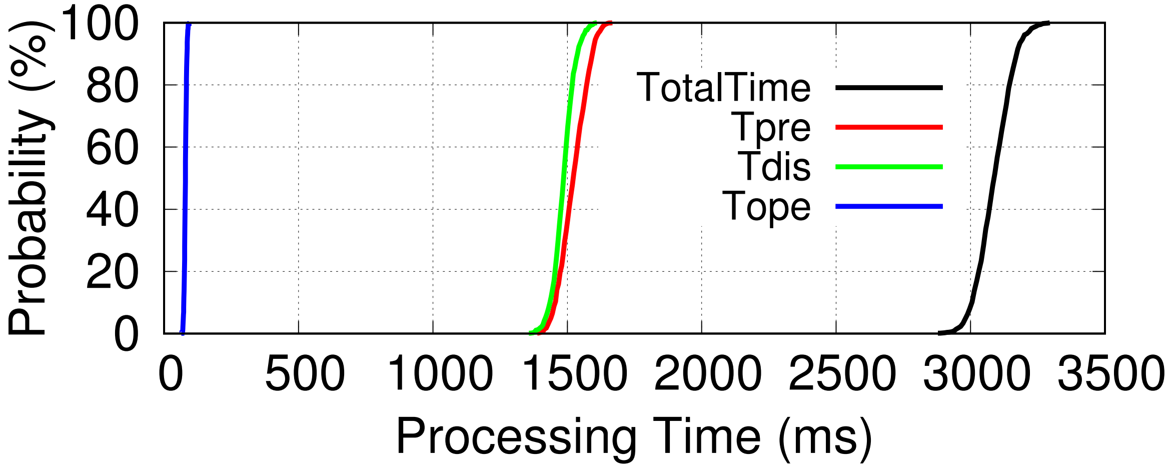
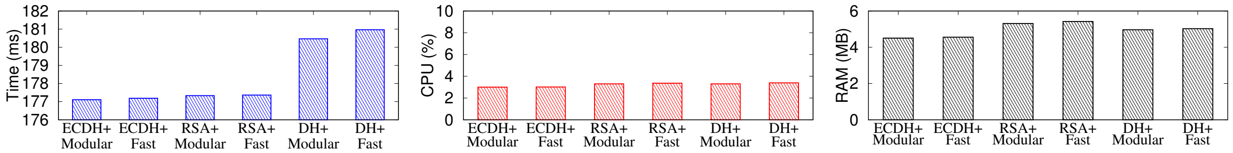


Figure 5.12 The CDF of the ID claim/revocation arbitration time.



(a) $T_{ope}$: order-preserving encryption time

(b) CPU usage

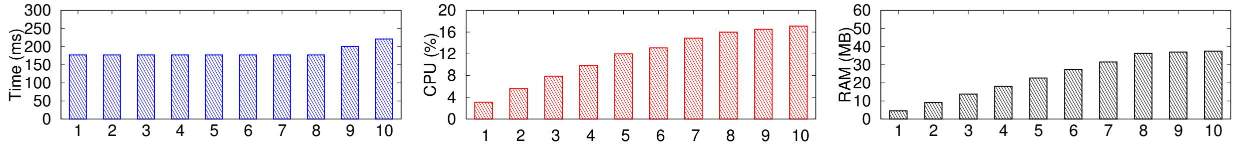
(c) RAM usage

Figure 5.13 The statistics of $T_{ope}$, CPU usage and RAM usage at CMS for different cryptographic schemes.

set to {first name, last name, year of birth, state of address} and $\{3, 3, 2, 1\}$, respectively. The experiment was conducted with 100 runs.

**Experimental results:** It was observed that each of those 10 benign users passed the ID claim pre-qualification and won the arbitration. The statistics of the total arbitration time and the time spent on each stage, $T_{pre}$, $T_{dis}$ and $T_{ope}$, are plotted in Figure 5.12. We have two observations. First, the overall ID claim arbitration process could be finished within 3.4s under the condition that the ID owner can timely respond to the arbitration request. Second, the 90th percentile values of $T_{pre}$, $T_{dis}$, and $T_{ope}$ are less than 1.63s, 1.58s, and 0.18s, respectively. The results have confirmed the effectiveness and efficiency of the paClaim service.

We further studied the impact of different cryptographic schemes on the performance of the paClaim service, especially for the OPE key exchange and encryption mechanisms (Steps 11-18 in Figure 5.5). Specifically, we considered three key exchange schemes, Diffie-Hellman (DH), Elliptic Curve Diffie-Hellman (ECDH), and RSA, and two OPE algorithms, Modular OPE [166] and Fast



(a) $T_{ope}$: order-preserving encryption time

(b) CPU usage

(c) RAM usage

Figure 5.14 The statistics of $T_{ope}$, CPU usage and RAM usage at CMS for different numbers of concurrent ID claim requests.

OPE [159]. For each combination of a key exchange scheme and an OPE algorithm, we initiated ID claim requests and measured $T_{ope}$, CPU usage, and RAM usage on average at CMS; there are totally six combinations of the cryptographic schemes. We make three observations from the experimental results, as shown in Figure 5.13. First, the ECDH-based schemes are faster than the others; the EDCH scheme plus modular OPE achieves the smallest $T_{ope}$, whereas the DH scheme plus fast OPE leads to the largest $T_{ope}$. Second, all the cryptographic schemes have comparable CPU usages, but the CPU usages of the RSA-based schemes are slightly higher than the others. Third, the ECDH-based schemes consume about 500 KB RAM less than the others.

We finally studied the performance and overhead of the paClaim service with a varying number of concurrent ID claim requests. The above experiment was repeated with two modifications: (1) the most efficient combination of cryptographic schemes, ECDH plus modular OPE, was used for MPKIX; and (2) the number of concurrent ID claim requests initiated ranges from 2 to 10. The results show that $T_{ope}$, CPU usage, and RAM usage are increased from 180 ms to 210 ms, from 3% to 18%, from 4 MB to 38 MB, respectively, when the number of concurrent requests increases from 2 to 10, as shown in Figure 5.14.

In summary, our experimental results confirm not only the effectiveness of the paClaim service, where concurrent ID claim requests can be processed efficiently, but also its merit that largely reduces the time of ID claim/revocation arbitration, compared with current technologies, while preserving IAS user privacy.

### 5.7 Discussion

In this subchapter, we discuss potential concerns about deploying and using MPKIX, as well as its limitations.

**Incentives for MPKIX deployment.** We believe that all the involved parties can benefit from the MPKIX deployment. The reasons are four-fold. First, *CAs* can expand their enterprise-based PKIX credential services to billions of mobile users, and the cost of user information verification can be greatly reduced without the need for photo ID inspection. Second, *cellular network operators* can make profit on new services including offering MPKIX to IAS providers and providing mobile users with emerging services (e.g., facilitating the aircraft boarding process and short-term keyless apartment rental) based on user certificate. Third, *IAS providers* can ensure the correctness of user information so that the risk of cyberattacks with potential legal issues and complaints can be reduced, e.g., those from governments [167] and online advertisers [168], where there is a total of $1.3 billion loss due to fake followers. Fourth, *IAS users* can transform their phones to PKCS#11-supported cryptographic tokens supporting a variety of PKIX credential services, with an efficient privacy-aware mechanism of ID claim/revocation arbitration.

Note that an IAS user may not need to pay the serving CA for the PKIX user certificate issuance if a reciprocal agreement between the connected cellular operator and the CA is signed. This business model is commonly observed in practice. For example, Google provides users with free cloud services but makes profit from online advertisers.

**Enforcing users to disclose more information?** People may think that MPKIX enforces IAS users to disclose more user information to cellular network operators, CAs, and IAS providers, compared with traditional mechanisms of user certificate. However, it is not true due to three major reasons. First, MPKIX leverages only the existing user information that has been verified by the operators. Second, MPKIX prevents IAS users from revealing user information to the CAs by encrypting data in carrier-endorsed ppCSRs. Third, MPKIX provides the service of ID claim/revocation arbitration to IAS users without the need of disclosing additional information to IAS providers. This privacy protection mechanism does not exist in current solutions [169].

**Why use cellular network infrastructure?** People may wonder why build MPKIX with cellular network operators but not the other institutions (e.g., banks and insurance companies) that also have verified user information. The reasons are two-fold. First, the cellular network infrastructures are built based on GSMA and 3GPP standards with a unified framework, but those institutions have diversified network systems. The standardized and unified cellular framework allows MPKIX to be developed on top of the GSMA OneAPI [136], which is generally supported by cellular operators, so that MPKIX can be easily deployed in operational cellular networks. Second, the CMS server is deployed as a 3GPP-defined application server (AS) [128], which can securely access HSS via the standardized Sh interface [129], but the other institutions may be afraid that deploying a new server in their network infrastructures may cause new security threats, especially for its access of their user information databases.

**Why not use email certificates?** Several CAs can issue a user with an X.509 email certificate within a few minutes. However, this kind of certificates can prove only the access of a particular email account for the user, but not other user information such as age and address.

**How about family-plan users?** In some countries, operators offer mobile services with family plans that contain more than one user. Some of them verify only the ID of the primary user; such a case is currently not supported by MPKIX. We leave it to our future work.

**How about photo-based ID theft attack?** An ID thief may impersonate an IAS user by using only the user's personal photo without other user information such as name and birthday. MPKIX can be extended to effectively defend against this attack due to three reasons. First, cellular network operators can easily obtain verified user photos while verifying each user's government-issued photo ID for service activation. Second, ppCert can carry any type of octet data including an encrypted user photo by using X.509 certificate extensions [2]. Third, given the encrypted user photo in ppCert, MPKIX can verify a provided user photo based on face recognition techniques, and thus prevent the photo-based ID theft attack.

**Is the paClaim service better than current solutions?** The common approach against ID theft attacks is to do the manual inspection on government-issued photo IDs or other proof documents provided by ID claimers. Seemingly, by involving the investigators of IAS providers with more user information, this approach is error-free and more rigorous than MPKIX. However, it may not be the case due to three reasons. First, this approach is not considered to proceed in a scientific way. The efficiency and accuracy of the ID claim process highly depend on the investigators, and they may delay or have a bias due to human factors. For example, one police's personal information was abused to create a Facebook ID by an adversary; however, the disputed ID still remained valid for a long time after a revocation request corresponding to the ID was submitted to Facebook [170]. Second, the current approach may be still vulnerable to ID owner masquerading attacks. With a stolen photo ID or a utility bill from a benign IAS user (e.g., accessing mailbox), an adversary can submit a request to an IAS provider by masquerading as the ID owner, and successfully claim the ownership of the benign IAS user account. However, with MPKIX, an IAS user cannot submit any ID claim/revocation request without passing the pre-qualification based on verified user information from CMS. For example, Bob cannot claim the ownership of Alice's account, even when Bob possesses Alice's driver license. Third, the photo-IDs and proof documents provided by ID claimers may compromise user privacy by leaking more user information. On the contrary, the proposed paClaim service is a scientific, rigorous and privacy-protected approach.

**Support 5G/6G cellular networks?** Current MPKIX prototype is built on top of 4G cellular infrastructure. However, to support new cellular network systems (e.g., 5G and 6G), its core functions do not require any modifications. Only the interfaces that MPKIX uses to communicate with cellular network elements/functions that store subscriber information, e.g., UDM (Unified Data Management)/UDR (Unified Data Repository) in 5G, need to be updated.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

This chapter provides conclusions of three aforementioned studies. Then, a few topics for future research are introduced.

## 6.1 Conclusion

The cellular network has played an important role in our daily life. With the development of cellular network, lots of new services continue to be added and provided by the operators. Considering a great amount devices and people connected via cellular networks, it is very important to secure mobile networks. Three security research projects of the most essential cellular network services including IMS services, wireless IoT services, and Internet Application services are conducted and introduced in this dissertation, which helps head toward the secure and dependable mobile networks. Next, the specific conclusion of each project is presented.

**Taming cellular network IP Multimedia Subsystem.** The VoWi-Fi service is thriving and being deployed worldwide. In this work, we conduct the first study on the security implication of the operational VoWi-Fi service over five operational networks, three in U.S. and two in Taiwan, using commercial VoWi-Fi devices (e.g., Google Nexus 6P, Apple iPhone 8, Samsung Galaxy S8). We discover three security vulnerabilities which stem from design defects of the Wi-Fi calling standard (V1 and V3) and an operational slip of the Wi-Fi calling services (V2). By exploiting the vulnerabilities, adversaries are able to launch the telephony harassment or denial of voice service attack and infer the Wi-Fi calling user's privacy.

The fundamental issue is that the conventional security defenses well examined in cellular network services are simply applied to the VoWi-Fi service without considering its specific security threats. We thus develop a solution, called Wi-Fi Calling Guardian, which alleviates real-world damage by getting to the root of the vulnerabilities. The lessons learned from the operational VoWi-Fi service operators can help secure mobile ecosystem and facilitate the global deployment, as well as provide new design insights for upcoming next-gen networks.

**Securing wireless IoT services.** The cellular IoT is thriving and being deployed worldwide. The security of the cellular IoT is playing an important role in its development, but has not been fully explored yet. In this work, we examine the security implications in the service charging scheme of the cellular network. We show that the cellular IoT charging can be exploited to launch attacks against carriers. The adversary can gain 43.75%-80.00% cheaper bills on cellular data services by masquerading non-IoT devices as IoT devices and abusing them in unanticipated use scenarios. The fundamental issue lies in that no sufficient security manners, which include mutual authentications between involved cellular entities, support differential charges between non-IoT and IoT devices. In light of heavy burdens on standard modification, we propose an anti-abuse solution to mitigate attack incentives instead of addressing the vulnerabilities directly. It can be used immediately in practice so as to benefit carriers on securing the cellular IoT ecosystem.

**Improving Internet Application Service.** Both IAS providers and users face various security threats nowadays. IAS providers are abused by adversaries based on fake user accounts, since they have no reliable means to verify correctness of user information. IAS users suffer from nefarious ID theft attacks, which lead to both financial losses and emotional/physical health damages. To address these security threats, we proposed the MPKIX framework to improve the security and accountability of IAS. MPKIX offers IAS providers with a general, reliable mechanism of user information verification, which makes IAS users accountable while largely preserving user privacy. Specifically, three novel mechanisms with privacy protection are introduced for user verification, namely carrier-endorsed PKIX user certificate issuance, privacy-preserving user information querying, and privacy-aware ID claim/revocation arbitration. Our evaluation results have shown that MPKIX is an effective and scalable approach. MPKIX not only provides a potent solution to secure the present-day IAS, but also benefits all the involved parties. Last but not least, the novel framework, MPKIX, integrates Internet Application Services into the wide-sense mobile networks. It enables the mobile network to provide secure and dependable services to users.

## 6.2 Future Work

Along the line of this thesis, there are two topics that are worth more research efforts in the future.

**Efficient and Secure Cellular-Satellite Communication.** Mobile networks (5G/4G) have successfully served billions of users. But the heavy deployment and operation costs in rural areas, aircraft, oceans, or even the areas suffered from the disasters (e.g., earthquakes, tornados, and wars) limit the cellular network to cover all global users everywhere. Thus, the industry and academia are collaborating to extend 5G and beyond network the constellation. Since this technique is still in early stage, many challenges would come with the new technology. For instance, the network congestion on the anchor gateway can be a limitation in the network. Due to the mobility of the satellite, it is expected to have frequent handover with high signaling overhead considering billions of users may access the network via satellite in the future. For security, a potential topic is moving parts of the cellular infrastructure (e.g., AUSF (Authentication Server Function) and UDM (unified data management)) to space, user authentication data leakage can become a problem for the harsh environment of outer space and the difficulty of repair. Therefore, there is a pressing need for studying the standardization for secure and efficient cellular satellite networks.

**Cellular IoT Charging on 5G.** As discussed in Chapter 4 *Securing wireless IoT services*, the unremarkable standard design defects, carrier operational slips, and implementation issues on the IoT device and infrastructure can cause the extremely appalling charging threat to the cellular users and operators. Compared to the cellular services for smartphones, the services for cellular IoTs is still on the way being mature. The service plans/mechanisms are changed frequently due to the development of IoT devices. Moreover, cellular IoTs can be more popular in 5G network. 5G is much more than just fast downloads; its unique combination of high-speed connectivity, very low latency, and ubiquitous coverage will better support smart vehicles and transport infrastructure such as connected cars, trucks, and buses. The operators will have specific plans to support those cellular IoT devices. Any design defect, operational slip, or implementation issue can cause the devastating problems. Thus, the study of cellular IoT charging vulnerabilities in 5G network is still a valuable research topic for future work.

# BIBLIOGRAPHY

[1] GSMA, “The Mobile Economy 2023,” Feb. 2023.

[2] “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” 2008, RFC 5280.

[3] 3GPP, “Ts23.402: Architecture enhancements for non-3gpp accesses;.”

[4] ——, “TS23.401: GPRS Enhancements for E-UTRAN Access,” 2011.

[5] GSMA, “IR.51 IMS OVER WI-FI V5.0,” May 2017.

[6] J. Arkko, V. Lehtovirta, P. Eronen, R. G. Authentication, K. Agreement *et al.*, “Extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka)", rfc 4187,” 2006.

[7] 3GPP, “TS33.402: Security aspects of non-3GPP accesses,” Jun. 2018.

[8] V. Devarapalli and F. Dupont, “Mobile ipv6 operation with ikev2 and the revised ipsec architecture,” RFC 4877, April, Tech. Rep., 2007.

[9] GSMA, “Wi-Fi Roaming Guidelines Version 12,” September. 2017.

[10] T. Tirronen, “Cellular IoT Alphabet Soup,” Feb. 2016, https://goo.gl/HmEgN7.

[11] AT&T, “Low cost LTE modules for the Internet of Things,” 2016, https://goo.gl/4g8AJW.

[12] Nokia, “LTE evolution for IoT connectivity,” 2017, https://onestore.nokia.com/asset/200178.

[13] A. Leckie, “LTE Category-0 & LTE-M low power M2M device roadmaps,” May 2015, https://goo.gl/cEGHDE.

[14] AT&T, “At&t iot plan,” https://marketplace.att.com/iot-connectivity, 2018.

[15] 3GPP, “TS 24.301: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3,” 2011.

[16] GSMA, “LTE-M Deployment Guide to Basic Feature Set Verison 2.0,” April 2018. [Online]. Available: https://gsma.com/newsroom/wp-content/uploads//CLP.29-v2.0.pdf

[17] “Google project fi,” https://fi.google.com/about/plan/, 2018.

[18] “Trusted digital signatures,” https://www.globalsign.com/en/digital-signatures/, 2021.

[19] P. Christian, A. Hadi, S. Nolen, B. Jasmine, T. Patrick, R. Bradley, and B. Kevin, “Sonar: Detecting ss7 redirection attacks with audio-based distance bounding,” in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 86–101.

[20] B. Reaves, L. Blue, H. Abdullah, L. Vargas, P. Traynor, and T. Shrimpton, “Authenticall: Efficient identity and content authentication for phone calls,” in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 575–592.

[21] B. Reaves, N. Scaife, D. Tian, L. Blue, P. Traynor, and K. R. Butler, “Sending out an sms: Characterizing the security of the sms ecosystem with public gateways,” in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 339–356.

[22] R. P. Jover, “The current state of affairs in 5g security and the main remaining security challenges,” *arXiv preprint arXiv:1904.08394*, 2019.

[23] L. He, Z. Yan, and M. Atiquzzaman, “Lte/lte-a network security data collection and analysis for security measurement: a survey,” *IEEE Access*, vol. 6, pp. 4220–4242, 2018.

[24] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, and D. Sisalem, “Survey of security vulnerabilities in session initiation protocol,” *IEEE Communications Surveys and Tutorials*, vol. 8, no. 1-4, pp. 68–81, 2006. [Online]. Available: https://doi.org/10.1109/COMST.2006.253270

[25] D. Geneiatakis, G. Kambourakis, T. Dagiuklas, C. Lambrinoudakis, and S. Gritzalis, “Sip security mechanisms: A state-of-the-art review,” in *In Proc. 5th International Network Conference (INC*. ACM, 2005, pp. 147–155.

[26] U. U. Rehman and A. G. Abbasi, “Security analysis of voip architecture for identifying sip vulnerabilities,” in *Emerging Technologies (ICET), 2014 International Conference on*. IEEE, 2014, pp. 87–93.

[27] A. Compagno, M. Conti, D. Lain, and G. Tsudik, “Don’t skype & type!: Acoustic eavesdropping in voice-over-ip,” in *AsiaCCS*. ACM, 2017, pp. 703–715.

[28] J. Fang, Y. Zhu, and Y. Guan, “Voice pattern hiding for voip communications,” in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–9.

[29] S. McGann and D. C. Sicker, “An analysis of security threats and tools in sip-based voip systems,” in *Second VoIP security workshop*, 2005.

[30] P. C. Hung and M. V. Martin, “Security issues in voip applications,” in *Electrical and Computer Engineering, 2006. CCECE’06. Canadian Conference on*. IEEE, 2006, pp. 2361–2364.

[31] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, “Insecurity of voice solution volte in lte mobile networks,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 316–327.

[32] I. Dacosta, V. Balasubramaniyan, M. Ahamad, and P. Traynor, “Improving authentication performance of distributed sip proxies,” in *Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications*. ACM, 2009, p. 1.

[33] E. Bocchi, L. Grimaudo, M. Mellia, E. Baralis, S. Saha, S. Miskovic, G. Modelo-Howard, and S.-J. Lee, “Magma network behavior classifier for malware traffic,” *Computer Networks*, vol. 109, pp. 142–156, 2016.

[34] A. I. Ali-Gombe, B. Saltaformaggio, D. Xu, G. G. Richard III *et al.*, "Toward a more dependable hybrid analysis of android malware using aspect-oriented programming," *computers & security*, vol. 73, pp. 235–248, 2018.

[35] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 63–78, 2018.

[36] M. Eskandari, B. Kessler, M. Ahmad, A. S. de Oliveira, and B. Crispo, "Analyzing remote server locations for personal data transfers in mobile apps," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 1, pp. 118–131, 2017.

[37] K. Block and G. Noubir, "My magnetometer is telling you where i've been?: A mobile device permissionless location attack," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2018, pp. 260–270.

[38] B. Reaves, J. Bowers, N. Scaife, A. Bates, A. Bhartiya, P. Traynor, and K. R. Butler, "Mo (bile) money, mo (bile) problems: analysis of branchless banking applications," *ACM Transactions on Privacy and Security (TOPS)*, vol. 20, no. 3, p. 11, 2017.

[39] H. Liu, Y. Wang, J. Liu, J. Yang, Y. Chen, and V. Poor, "Authenticating users through fine-grained channel information," *IEEE Transactions on Mobile Computing*, no. 1, pp. 1–1, 2018.

[40] I.-G. Lee, H. Choi, Y. Kim, S. Shin, and M. Kim, "Run away if you can: Persistent jamming attacks against channel hopping wi-fi devices in dense networks," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2014, pp. 362–383.

[41] H. Li, Z. Xu, H. Zhu, D. Ma, S. Li, and K. Xing, "Demographics inference through wi-fi network traffic analysis," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[42] S. Consolvo, J. Jung, B. Greenstein, P. Powledge, G. Maganis, and D. Avrahami, "The wi-fi privacy ticker: improving awareness & control of personal information exposure on wi-fi," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 2010, pp. 321–330.

[43] J. W. Mikhail, J. M. Fossaceca, and R. Iammartino, "A semi-boosted nested model with sensitivity-based weighted binarization for multi-domain network intrusion detection," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 3, p. 28, 2019.

[44] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2015.

[45] J. Beekman and C. Thompson, "Man-in-the-middle attack on t-mobile wi-fi calling," https://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-18.pdf, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Tech. Rep., 2013.

[46] S. Chalakkal, H. Schmidt, and S. Park, "Practical attacks on volte and vowifi," *ERNW Enno Rey Netzwerke, Tech. Rep*, 2017.

[47] CISION, "comScore Reports June 2017 U.S. Smartphone Subscriber Market Share," 2017, https://www.prnewswire.com/news-releases/comscore-reports-june-2017-us-smartphone-subscriber-market-share-300498296.html.

[48] 3GPP, "TS24.302:Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks," 2017.

[49] W.-F. Alliance, "Hotspot 2.0 Specification," 2019. [Online]. Available: https://www.wi-fi.org/downloads-registered-guest/Hotspot_2.0_Specification_Package_v2.0.zip/29728

[50] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[51] 3GPP, "TS23.237:IP Multimedia Subsystem (IMS) Service Continuity; Stage 2," 2017.

[52] UCLA, "Cellular Network Trace Collector: Spurring In-Phone Mobile Network Intelligence," 2017, http://www.mobileinsight.net/.

[53] S. Thomée, A. Härenstam, and M. Hagberg, "Mobile phone use and stress, sleep disturbances, and symptoms of depression among young adults-a prospective cohort study," *BMC public health*, vol. 11, no. 1, p. 66, 2011.

[54] Y.-A. de Montjoye, J. Quoidbach, F. Robic, and A. S. Pentland, "Predicting personality using novel mobile phone-based metrics," in *International conference on social computing, behavioral-cultural modeling, and prediction*. Springer, 2013, pp. 48–55.

[55] V. Balasubramaniyan, M. Ahamad, and H. Park, "Callrank: Combating SPIT using call duration, social networks and global reputation," in *CEAS'07*, 2007.

[56] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[57] P. Hu and D. Ramanan, "Finding tiny faces," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1522–1530.

[58] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[59] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, vol. 3, no. 6, 2017, p. 7.

[60] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 850–855.

[61] M. Cristani, A. Pesarin, A. Vinciarelli, M. Crocco, and V. Murino, "Look at who's talking: Voice activity detection by automated gesture analysis," in *Constructing Ambient Intelligence*, R. Wichert, K. Van Laerhoven, and J. Gelissen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 72–80.

[62] "scikit-learn: Machine Learning in Python," http://scikit-learn.org/stable/.

[63] VLFeat, "VLFeat 0.9.21," 2018, http://www.vlfeat.org/.

[64] MSU, "MSU High Performance Computing."

[65] IEEE, "Precision Time Protocol," 2008, https://en.wikipedia.org/wiki/Precision_Time_Protocol.

[66] S. CARLO, "Britain has more surveillance cameras per person than any country except china. that is a massive risk to our free society," https://time.com/5590343/uk-facial-recognition-cameras-china/, 2019.

[67] zhihu, "Do you know what level of domestic face recognition monitoring is achieved," https://zhuanlan.zhihu.com/p/39868461, 2019.

[68] G. Barber, "Some us cities are moving into real time facial surveillance," https://www.wired.com/story/some-us-cities-moving-real-time-facial-surveillance/, 2019.

[69] W.-F. Alliance, "Wpa3 specification version 1.0," 2019.

[70] I. Leontiadis, C. Delakouridis, L. Kazatzopoulos, and G. F. Marias, "Anosip: anonymizing the sip protocol," in *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. ACM, 2012, p. 4.

[71] H. Zhang, D. She, and Z. Qian, "Android ION hazard: the curse of customizable memory management system," in *ACM CCS*, 2016.

[72] Y. Shao, Q. A. Chen, Z. M. Mao, J. Ott, and Z. Qian, "Kratos: Discovering inconsistent security policy enforcement in the android framework," in *IEEE NDSS*, 2016.

[73] X. Zhou, Y. Lee, N. Zhang, M. Naveed, and X. Wang, "The peril of fragmentation: Security hazards in android device driver customizations," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 409–423. [Online]. Available: http://dx.doi.org/10.1109/SP.2014.33

[74] Y. Go, E. Jeong, J. Won, Y. Kim, D. F. Kune, and K. Park, "Gaining control of cellular traffic accounting by spurious TCP retransmission," in *NDSS*. The Internet Society, 2014.

[75] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real threats to your data bills: Security loopholes and defenses in mobile data charging," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 727–738.

[76] C. Peng, G.-h. Tu, C.-y. Li, and S. Lu, "Can we pay for what we get in 3g data access?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 113–124.

[77] C. Peng, G. Tu, C. Li, and S. Lu, "Can we pay for what we get in 3g data access?" in *ACM Mobicom*, 2012.

[78] G.-H. Tu, C.-Y. Li, C. Peng, Y. Li, and S. 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*. ACM, 2016, pp. 1118–1130.

[79] H. Tu, A. Doupé, Z. Zhao, and G. Ahn, "Sok: Everyone hates robocalls: A survey of techniques against telephone spam," in *IEEE S&P*, 2016.

[80] W. Bu, M. Xue, L. Xu, Y. Zhou, Z. Tang, and T. Xie, "When program analysis meets mobile security: an industrial study of misusing android internet sockets," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 2017, pp. 842–847.

[81] T. Xie, G.-H. Tu, C.-Y. Li, C. Peng, J. Li, and M. Zhang, "The dark side of operational wi-fi calling services," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018.

[82] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb 2017.

[83] "Mirai Malware for Botnet," 2017, https://en.wikipedia.org/wiki/Mirai_(malware).

[84] B. Krebs, "FBI: Smart Meter Hacks Likely to Spread," 2012, https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[85] R. Liu and M. Srivastava, "Virtsense: Virtualize sensing through arm trustzone on internet-of-things," in *Proceedings of the 3rd Workshop on System Software for Trusted Execution*. ACM, 2018, pp. 2–7.

[86] C. Gao, V. Chandrasekaran, K. Fawaz, and S. Banerjee, "Traversing the quagmire that is privacy in your smart home," in *Proceedings of the 2018 Workshop on IoT Security and Privacy*. ACM, 2018, pp. 22–28.

[87] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 832–846.

[88] N. Sastry and D. Wagner, "Security considerations for ieee 802.15. 4 networks," in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 32–42.

[89] S. Soltan, P. Mittal, and H. V. Poor, "Blackiot: Iot botnet of high wattage devices can disrupt the power grid," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 15–32.

[90] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet." in *NDSS*, 2019.

[91] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash, "ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms," in *IEEE NDSS*, 2017.

[92] H. Nissenbaum, “PRIVACY AS CONTEXTUAL INTEGRITY,” 2004, https://crypto.stanford.edu/portia/papers/RevnissenbaumDTP31.pdf.

[93] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, “A deep learning approach to iot authentication,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[94] A. F. Harris, H. Sundaram, and R. Kravets, “Security and privacy in public iot spaces,” in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–8.

[95] Q. Wang, P. Datta, W. Yang, S. Liu, A. Bates, and C. A. Gunter, “Charting the attack surface of trigger-action iot platforms,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019, pp. 1439–1453.

[96] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig, “Siotome: An edge-isp collaborative architecture for iot security,” *Proc. IoTSec*, 2018.

[97] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, “An efficient algorithm for media-based surveillance system (eamsus) in iot smart city framework,” *Future Generation Computer Systems*, vol. 83, pp. 619–628, 2018.

[98] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, “Secure integration of iot and cloud computing,” *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[99] Z. B. Celik, G. Tan, and P. D. McDaniel, “Iotguard: Dynamic enforcement of security and safety policy in commodity iot.” in *NDSS*, 2019.

[100] K. E. Psannis, C. Stergiou, and B. B. Gupta, “Advanced media-based smart big data on intelligent cloud systems,” *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 77–87, 2018.

[101] C. Stergiou, K. E. Psannis, A. P. Plageras, Y. Ishibashi, B.-G. Kim *et al.*, “Algorithms for efficient digital media transmission over iot and cloud networking,” *Journal of Multimedia Information System*, vol. 5, no. 1, pp. 27–34, 2018.

[102] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, “Efficient iot-based sensor big data collection–processing and analysis in smart buildings,” *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

[103] Statista, “Market share of mobile network carriers in the U.S,” 2017, http://www.statista.com/statistics/199359/market-share-of-wireless-carriers-in-the-us-by-subscriptions/.

[104] lyriquidperfection, “EFS Professional,” Aug. 2011, https://goo.gl/wQKJ59.

[105] T. M. Khoshgoftaar, S. V. Nath, S. Zhong, and N. Seliya, “Intrusion detection in wireless networks using clustering techniques with expert analysis,” in *Fourth International Conference on Machine Learning and Applications (ICMLA’05)*. IEEE, 2005, pp. 6–pp.

[106] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 116–127.

[107] K. B. Rasmussen and S. Capkun, “Implications of radio fingerprinting on the security of sensor networks,” in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*. IEEE, 2007, pp. 331–340.

[108] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improving intra-cellular security using air monitoring with rf fingerprints,” in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[109] “Openairinterface,” 2018, http://www.openairinterface.org/.

[110] 3GPP, “TS23.203: Policy and charging control architecture,” Dec. 2017.

[111] “B+ tree,” https://en.wikipedia.org/wiki/B

[112] 3GPP, “TS36.413: Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP),” 2011.

[113] P. Wang, F. Sun, D. Wang, J. Tao, X. Guan, and A. Bifet, “Inferring demographics and social networks of mobile device users on campus from ap-trajectories,” in *Proceedings of the 26th International Conference on World Wide Web Companion*.

[114] T. J. Neal and D. L. Woodard, “A gender-specific behavioral analysis of mobile device usage data,” in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*.

[115] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, “Towards a complete view of the certificate ecosystem,” in *Proceedings of the 2016 Internet Measurement Conference*.

[116] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, J. Li, and Z. Zhang, “Rusted anchors: A national client-side view of hidden root cas in the web pki ecosystem,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[117] S. Y. Chau, O. Chowdhury, E. Hoque, H. Ge, A. Kate, C. Nita-Rotaru, and N. Li, “Symcerts: Practical symbolic execution for exposing noncompliance in x. 509 certificate validation implementations,” in *Security and Privacy (SP), 2017 IEEE Symposium on*.

[118] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla *et al.*, “Let’s encrypt: an automated certificate authority to encrypt the entire web,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[119] X. Wang and M. El-Said, “Domainpki: Domain aware certificate management,” in *Proceedings of the 21st Annual Conference on Information Technology Education*, 2020.

[120] M. Wang, C. Qian, X. Li, and S. Shi, “Collaborative validation of public-key certificates for iot by distributed caching,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019.

[121] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, “Pki4iot: Towards public key infrastructure for the internet of things,” *Computers & Security*, vol. 89, 2020.

[122] A. Rashid, A. Masood, H. Abbas, and Y. Zhang, “Blockchain-based public key infrastructure: A transparent digital certification mechanism for secure communication,” *IEEE Network*, vol. 35, no. 5, 2021.

[123] A. Papageorgiou, A. Mygiakis, K. Loupos, and T. Krousarlis, “Dpki: a blockchain-based decentralized public key infrastructure system,” in *2020 Global Internet of Things Summit (GIoTS)*, 2020.

[124] GSMA, “Gsma mobile connect,” https://mobileconnect.io/, 2020.

[125] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on information theory*, vol. 29, no. 2, 1983.

[126] F. C. Commission *et al.*, “Code of federal regulations: Title 47–telecommunications: Universal service,” 2008.

[127] GSMA, “Mandatory registration of prepaid sim cards,” https://www.gsma.com/publicpolicy/wp-content/uploads/2016/04/Mandatory-SIM-Registration.pdf, 2016.

[128] 3GPP, “TS23.002: Network architecture,” Mar. 2017.

[129] ——, “TS29.329: Sh interface based on the Diameter protocol; Protocol details,” Jul. 2020.

[130] P. Calhoun, “Diameter framework document,” *Internet draft*, 2001.

[131] H. Haverinen and J. Salowey, “Rfc 4186: Extensible authentication protocol method for global system for mobile communications (gsm) subscriber identity modules (eap-sim),” 2006.

[132] 3GPP, “TS35.909: Specification of the MILENAGE algorithm set: an example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*.” July 2020.

[133] NISF, “FIPS Publication 186-2: Digital Signature Standard (DSS),” January 2000.

[134] T. Dierks and E. Rescorla, “RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2.”

[135] “Oauth 2.0,” https://oauth.net/2/, 2021.

[136] “Gsma oneapi,” https://github.com/GSMADeveloper/GSMA-OneAPI/wiki, 2019.

[137] U. Hengartner and P. Steenkiste, “Exploiting hierarchical identity-based encryption for access control to pervasive computing information,” in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM’05)*, 2005.

[138] F. P. Miller, A. F. Vandome, and J. McBrewster, *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau-Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009.

[139] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, “Order preserving encryption for numeric data,” in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004.

[140] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE transactions on Information Theory*, vol. 22, no. 6, 1976.

[141] “Digital signature,” https://en.wikipedia.org/wiki/Digital_signature, 2019.

[142] NSA, “Eliminating obsolete transport layer security (tls) protocol configurations.”

[143] S. Alt, P.-A. Fouque, G. Macario-rat, C. Onete, and B. Richard, “A cryptographic analysis of umts/lte aka,” in *Applied Cryptography and Network Security*, 2016.

[144] D. Fett, R. Küsters, and G. Schmitz, “A comprehensive formal security analysis of oauth 2.0,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[145] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, 3rd ed. USA: Prentice Hall Press, 2014.

[146] “Sim cards: attack of the clones,” https://www.kaspersky.com/blog/sim-card-history-clone-wars/11091/, 2016.

[147] K. Nohl, “Rooting sim cards,” *BlackHat Briefings*, 2013.

[148] “Sim swap attack,” https://whatis.techtarget.com/definition/SIM-swap-attack-SIM-intercept-attack.

[149] Y. J. Jia, Q. A. Chen, Y. Lin, C. Kong, and Z. M. Mao, “Open doors for bob and mallory: open port usage in android apps and security implications,” in *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, 2017.

[150] L. Armasu, “How google improved android security in 2017?” https://www.tomshardware.com/news/google-android-security-improvements-2017,36673.html, 2018.

[151] Y. Acar, M. Backes, S. Bugiel, S. Fahl, P. McDaniel, and M. Smith, “Sok: Lessons learned from android security research for appified software platforms,” in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

[152] OpenSSL, “OpenSSL: Cryptography and SSL/TLS Toolkit,” 2021, https://www.openssl.org/.

[153] P. Wendland, “A java card pki applet aiming to be iso 7816 compliant,” https://github.com/philipWendland/IsoApplet, 2015.

[154] OASIS, “Pkcs #11 cryptographic token interface base specification version 2.40,” http://docs.oasis-open.org/pkcs11/pkcs11-base/v2.40/os/pkcs11-base-v2.40-os.html, 2015.

[155] node diameter, “nodediameter,” 2021, https://github.com/node-diameter/node-diameter/.

[156] “Oauth 2.0 server,” https://github.com/thephpleague/oauth2-server, 2021.

[157] ScribeJava, “faker.js,” https://github.com/scribejava/scribejava, 2019.

[158] boost, “Boost c++ libraries,” http://erikerlandson.github.io/algorithm/index.html, 2020.

[159] Y. H. Hwang, S. Kim, and J. W. Seo, “Fast order-preserving encryption from uniform distribution sampling,” in *Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop*.

[160] MIT CCS reseach group, “Cryptdb,” http://css.csail.mit.edu/cryptdb/, 2013.

[161] “Voter registration records,” https://raidforums.com/Announcement-Database-Index-CLICK-ME, 2019.

[162] Google, “Geocoding api,” https://developers.google.com/maps/documentation/geocoding/overview, 2021.

[163] B. Castle, “The legion of the bouncy castle,” https://www.bouncycastle.org/java.html, 2020.

[164] “Globalsign authentication,” https://www.globalsign.com/en/authentication/, 2021.

[165] “Digicert client certificate,” https://www.digicert.com/client-certificates/, 2021.

[166] A. Boldyreva, N. Chenette, Y. Lee, and A. O’neill, “Order-preserving symmetric encryption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2009.

[167] H. M. Elanine Moore, “Facebook’s massive fake numbers problem.”

[168] M. Graham, “Cnbc: Fake followers in influencer marketing will cost brands $1.3 billion this year, report says,” 2019.

[169] Facebook, “Report an impostor account,” https://www.facebook.com/help/contact/295309487309948, 2021.

[170] M. C. Jo Ling Kent, “Nbc news: Fake facebook profiles cause heartbreak for families and colleagues.”