SecureSIM: Rethinking Authentication and Access Control for SIM/eSIM

Jinghao Zhao, Boyan Ding, Yunqi Guo, Zhaowei Tan, Songwu Lu

University of California, Los Angeles

{jzhao,dboyen,luckiday,tan,slu}@cs.ucla.edu

ABSTRACT

The SIM/eSIM card stores critical information for a mobile user to access the 4G/5G network. In this work, we uncover three vulnerabilities of the current SIM practice. We show that the PIN-based access control may expose the in-SIM data to an adversary through both hardware and software. Once exposed, such in-SIM information can be used to reconstruct various keys used for device authentication, data encryption, etc. They thus enable a number of attacks, including traffic eavesdropping, man-in-the-middle attack, impersonation, etc. The fundamental problem is that, the current SIM design does not offer proper authentication and fine-grained access control to hundreds of in-SIM files for various in-card applets and off-card units. We next propose a new solution that offers both authentication and fine-grained access control. Our implementation and evaluation have confirmed the viability of our proposal.

CCS CONCEPTS

• **Security and privacy** → **Mobile and wireless security**; *File system security*.

KEYWORDS

Mobile Security, SIM Security, Access Control, SecureSIM

ACM Reference Format:

Jinghao Zhao, Boyan Ding, Yunqi Guo, Zhaowei Tan, Songwu Lu. 2021. SecureSIM: Rethinking Authentication and Access Control for SIM/eSIM. In *The 27th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '21)*, October 25–29, 2021, New Orleans, LA, USA. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3447993.3483254>

1 INTRODUCTION

The SIM card or the recent embedded SIM (eSIM) stores critical information for the mobile user to access the Internet or make voice calls through the 4G/5G cellular network. It contains the device/user identities, various security keys used for 4G/5G access, location status, or personal records (e.g., contacts and phone numbers), in the form of hundreds of in-SIM files. If hacked, such information leakage will compromise the entire 4G/5G system security.

In this paper, we study the topic of SIM/eSIM¹ security. We seek to achieve two goals. First, we devise practical attacks against

current SIM practices. We implement several attacks and assess their impacts on our 4G LTE testbed, as well as over two US mobile operators. Second, we propose new solutions to defend against these attacks and better protect the SIM for 4G/5G access.

The current SIM design has been prevailing for years due to the simple communication scheme and easiness of extending new services. However, the interoperability between various entities such as the SIM, modem, and protocol stacks exposes vulnerabilities that risk mobile network security. We identify three vulnerabilities in current SIM practices. First, the current SIM offers simple PIN-based access control, but it cannot provide enough protection. Adversaries may easily access sensitive in-SIM information through hardware or software. Second, the interaction between the SIM and the LTE protocol stack retains the security context that can be leaked and reused. Third, the communication channel between the modem and the SIM is not encrypted, thus eavesdropping is feasible via malicious hardware. The fundamental problem is that, the PIN-based, coarse-grained access control does not suffice to protect the SIM information from available hardware or malware. Furthermore, the current SIM does not offer proper authentication for various in-card applets and off-card units that legitimately seek to access the in-SIM information, as well as from adversaries.

We further devise several practical attacks that exploit the three vulnerabilities. Our attacks first retrieve the security context from SIM/eSIM. For SIM, this context can be acquired via one-time physical access to the SIM card; for eSIM, it is learned with malware on rooted devices. Given the security context, we next reconstruct the various keys used in 4G/5G data communications. We further apply the reconstructed decryption key upon the eavesdropped, yet ciphered data traffic between the device and the 4G/5G infrastructure, and recover the original plain-text content. We may launch the man-in-the-middle attack, which uses the one-time key access to authenticate itself to the victim device. Moreover, the adversary can further impersonate the victim's identity through the standardized fast reattach procedure. Our prototype and evaluation on the testbed, as well as two US mobile operators, have confirmed the feasibility of such attacks.

To defend against such attacks, we propose a novel solution that offers both authentication and fine-grained access control to SIM. We depart from the PIN and key based authentication, and devise a certificate-based authentication scheme that scales to hundreds of fine-grained entities from both in-SIM applets and off-card units. Our access control organizes all in-SIM files into six classes of profiles, and customizes the access policy schema of various in-SIM applets and off-card units into a multi-rooted tree hierarchy. Through this way, we enable the fine-grained policy specification and automated checking through the access graph. Our empirical evaluation shows that certificate verification introduces 0.9s (17%)

¹For simplicity of notation, our presentation is centered on SIM security; all results are equally applicable to eSIM.

This work is licensed under a Creative Commons Attribution International 4.0 License.

ACM MobiCom '21, October 25–29, 2021, New Orleans, LA, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8342-4/21/10.

<https://doi.org/10.1145/3447993.3483254>

extra latency when loading files at the start of the session, which typically lasts for several days. The amortized overhead is thus marginal. The encryption component ensures security with acceptable latency (5.5%) and energy (3.2%) overhead.

2 BACKGROUND

2.1 SIM and eSIM

The universal integrated circuit card (UICC), also known as the SIM card, facilitates a mobile user to access the 4G/5G cellular network. It stores the SIM file package, which contains various critical parameters for 4G/5G access. The embedded SIM (eSIM) is a recent technique to improve usability and programmability. It is soldered on the device's circuit board and provides the SIM functions with a programmable chip. eSIM supports remote SIM provisioning, allowing users to download SIM file packages remotely and change between operators' packages without switching physical SIM cards.

Both SIM and eSIM use various types of files as the basic units for storing the SIM package. A typical SIM includes around 200 files. Each file has a unique ID [3]. The operators² control SIM files and are responsible for managing the file access. The current SIM offers four access modes for files: (1) Always (ALW), with no restrictions. (2) PIN, where the card holder needs to verify the PIN unless PIN verification is disabled. (3) ADM, where the administrative key (owned by the card issuer, e.g., operators) needs to be verified. (4) Never (NEV), with no access. For example, the international mobile subscriber identity (IMSI) is stored with ID 6F07 with ADM access mode so that the modem can only read it but cannot update. Some other files that need updates from the modem during connection use PIN mode, such as the security context.

The modem and SIM communicate through application protocol data unit (APDU), which can be considered as a byte-stream interface. The modem utilizes it to load files and access the network with loaded files. The mobile operating system or applications can also send APDUs to the SIM through the modem. For example, operator applications could send APDUs to update network configurations in the SIM [64].

To provide add-on services, e.g., geofencing, Quality of Service reporting, and SIM status checking [44], operators or IoT service providers install applets in the SIM. The applet is a small application in the SIM, which can send APDUs to the modem to acquire device information, set up calls, send short messages (SMS), etc.

2.2 LTE Security Primer

The SIM plays a critical role in the LTE security design and operations. We thus review the LTE security design and how SIM files facilitate it.

LTE Key Derivation In the 4G LTE network, the mobile device, also known as user equipment (UE), gains data service through direct access to base stations (eNB). Figure 1 shows the underlying LTE protocol stacks. The control plane provides control functions such as authentication in the mobility management entity (MME). The user plane provides data services with routing packets to the Packet Data Network Gateway (P-GW). Both control and user plane traffic are protected with symmetric keys. The SIM and the network share the root key K . Operators install the K when initializing the

SIM, and users cannot acquire the K . All encryption and integrity keys, listed in Figure 1, are derived from K . Keys between the UE and MME protect the Non-access Stratum (NAS) signaling messages. Other keys are used between UE and eNB at Packet Data Convergence Protocol (PDCP) to provide security for Radio Resource Control (RRC) signaling messages and user data.

As a crucial step for key derivation, the Authentication and Key Agreement (AKA) procedure provides mutual authentication as shown in Figure 2(a). First, the UE's modem initializes a NAS Attach Request message, including IMSI as the identity. Upon receiving the request, MME sends a NAS Auth Request to the UE, including a random byte array (RAND) and authentication token (AUTN) generated based on the K and RAND. The UE's modem forwards the RAND and AUTH to the SIM. The SIM verifies the MME based on the K and derives Cipher Key (CK), Integrity Key (IK), and the response value (RES). The modem gets keys and RES through APDU. The modem derives the K_{ASME} with CK/IK and sends the RES to the MME. If it is as expected, the MME authenticates the UE and agrees on the shared K_{ASME} . Both sides derive NAS encryption and integrity key K_{NASenc}/K_{NASint} from the K_{ASME} with the algorithm identifier in the NAS Security Mode Command (SMC). Finally, both sides derive keys for protecting RRC signaling and user data.

LTE Fast Reattach Procedure To speed up the reattach procedure, the standard allows the UE to reattach with the cached context³ [4]. When the UE detaches (e.g., loses signal, enables the airplane mode, or turns off), the UE shall store the context into the SIM if the SIM supports the fast reattach service. The context includes the identifier called Globally Unique Temporary Identifier (GUTI) in file EPSLOC1, and NAS security context in file EPSNSC. When the UE detects a cached context, it will use the cached GUTI as the identity in the NAS Attach Request, as shown in Figure 2(b). The NAS Attach Request will be integrity-protected with the UE's cached context. If the MME finds a record corresponding to the context, the operators can bypass the AKA procedure [4]. In this case, the MME directly sends the NAS SMC to the UE. They authenticate mutually by checking the message's integrity with the cached K_{NASint} . Then they derive all the following keys and set up the network connection.

3 VULNERABILITY & THREAT MODELS

3.1 Vulnerabilities

The SIM plays a critical role in the 4G/5G connection for authentication and protecting the communication. The current SIM design has been prevailing for years due to its simplicity and extendability. Diverse schemes in LTE, such as the fast reattach service, are also designed to improve the network performance. However, the interoperability between various entities such as the SIM, modem, and protocol stacks exposes vulnerabilities:

First, the PIN-based access control cannot provide enough file protection (V1). The SIM cannot differentiate between various entities that access the SIM by PIN verification. Sensitive files, including identities, security-related keys, phonebooks, etc., can be accessed by modems, card readers, OS, or apps. LTE-related files such as the security context are only intended for the modem. Leaking such

²Operators represent the owner of commercial or private networks.

³In 3GPP, it is called EPS Mobility Management Information.

Figure 1: LTE Protocol Stack and Key Architecture

Figure 3: SIM/eSIM Usage Survey

sensitive files allows attackers to launch eavesdropping, man-in-the-middle (MitM), and impersonation attacks. Furthermore, the current PIN-based authorization is insufficient. Though all the files mentioned above are only accessible after PIN authorization, the SIM only verifies the PIN once, then unlocks access until SIM is unplugged or the device turns off. The attacker without the PIN can still access files during the power cycle, although the user enables the PIN authorization. Even worse, operators usually set the SIM PIN to an identical default number and keep SIM unlocked by default [27]. For example, AT&T and Verizon use 1111 and T-Mobile uses 1234 as the default PIN and need users to activate it manually [9, 63, 65]. We conducted an informal survey to assess mobile users' sensitivity to SIM security. We collected responses from 103 volunteers from the authors' social networks in the US and 86 responses from anonymous online participants, which were solicited from a US university's public forums on online social networks. The participation was open to everybody and there was no material incentive. The participants cover a wide range of age groups, as shown in Figure 3. Among participants, 94.2% of the users set the password for their phones, but only 10.6% set the SIM PIN. Our survey results indicate that the vast majority of mobile users do not enable the SIM PIN, exposing all protected files to attackers.

Second, the SIM and LTE protocol stack interoperation exposes vulnerabilities (V2). Storing the security context is mandatory, but the current design cannot provide enough protection. According to the standard, if the SIM supports the fast reattach service, the modem shall store the NAS security context into the SIM when the UE detaches from the network [4]. The modem would delete the context when it detects the SIM is changed during the power-on state. During the power-off state, however, the modem cannot detect if the SIM has been changed. The attacker can unplug the

Figure 2: LTE Authentication Procedure

SIM and acquire the context. Supposing that the attacker inserts the SIM card back when the modem's power is off, the UE and network will use the same context to perform the fast reattach, which allows various attacks, including traffic eavesdropping, man-in-the-middle (MitM), and impersonation.

Third, the channel between the modem and the SIM is unencrypted (V3). It enables malicious hardware such as a SIM sticker [58] to eavesdrop on and spoof the APDU traffic. The attacker can acquire all the files as plaintext when the modem reads them from the SIM. Multi-layer keys in LTE can also be accessed through the unencrypted APDU. Prior work leveraged CK/IK leakage during the AKA procedure to attack the VoLTE and VoWiFi [13]. We discover that the K_{ASME} leaks when the modem store the NAS security context into the SIM. Moreover, the attacker hardware can modify the APDU or actively send APDUs to the mobile device to acquire the device location, network conditions, device ID, calling information, etc. The attacker can also leverage APDUs to set up calls or send messages without notifying the victim.

3.2 Threat Model

We consider two types of attackers. A *passive attacker* can set up the standard-compliant SDR devices to eavesdrop the radio layer information. The passive attacker could log the ciphered signaling and data messages between the victim UE and eNB, which remains unperceivable. An *active attacker* can set up forged UE, eNB, and EPC to send standard-compliant signaling and data messages in the communication channel. The active attacker could also forge or alter packets from/to the victim at the forged eNB and EPC. Both types of attackers leverage low-cost SDR devices. We assume both attackers can acquire sensitive files such as the NAS security context from SIM with one-time physical access through customized hardware (such as the SIM sticker or card reader in §4.1) or malware installed on the victim's mobile devices.

Our experiments also observe ethical rules. We ensure that tests on commercial networks do not harm other users. We own the SIM cards used in the experiments. Our passive attacker does not send signals over the licensed band. We use the unlicensed bands for experiments on the MitM and impersonation attacks. We are also informing mobile operators of our findings.

4 ATTACKS

We next devise a novel two-step attack against the current SIM. First, we introduce a component attack that obtains sensitive files such as the security context from the SIM. We show that the attacker only needs one-time physical access to acquire sensitive files, which opens a door for other attacks. Second, based on the one-time

access attack, we devise three novel attacks: traffic eavesdropping, context-based MitM, and impersonation attack. We present the attack procedures for each attack and validate them on two US mobile operators, denoted as OP-I and OP-II.

4.1 One-time Access Attack

Current SIM with PIN-based access control cannot differentiate between access entities. Attackers could leverage cheap hardware or malware to acquire sensitive files, such as the victim's identities and the security context. The prerequisite for our attacks is to obtain the security keys via one-time physical access to the SIM card or malware on the rooted device. Table 1 shows requirements for attacks from different hardware and software. SIM is susceptible to both the hardware/software attack, and eSIM is only susceptible to the software attack.

Attackers can leverage the customized hardware, such as a SIM sticker, to extract sensitive files from the victim's SIM. The SIM sticker, as shown in Figure 4, is a small chip with a processor and storage. The SIM sticker has the same size as a SIM card and can be attached to the SIM. It is used to support multi-IMSI or unlock the carrier lock [28, 37]. The attacker can extend it to log the sensitive files. To attack the victim with the SIM sticker, the attacker first needs one-time access to the victim's device. The attacker unplugs the SIM, attaches the sticker to the SIM card, and inserts them back into the device. The attack through the sticker does not require the device password or SIM PIN. APDUs between the modem and SIM will go through the sticker. Even if the victim enables the PIN, the sticker could still eavesdrop on all the APDUs as plaintext (V3). The attacker does not need to reaccess the device or unplug the sticker to read data. The sticker can forge APDUs and send the captured files through SMS to the attacker's number, enabling the attacker to control the APDU channel and acquire SIM files remotely.

The attacker can also use a SIM card reader to obtain sensitive files. Reading files with the card reader does not require the device password but requires that the SIM PIN is not enabled or the attacker knows the PIN. The attacker needs one-time access to the victim's device first. To acquire the security context files, the attacker triggers the UE detach procedure by enabling the airplane mode from the status bar or turning off the device by pressing the power button. It forces the modem to store the security context into the SIM (V2). This procedure usually does not require unlocking the phone. The attacker unplugs the SIM, reads files with a card reader (V1), then plugs the SIM back into the victim's device.

To achieve the same purpose, the attacker can allure victims to install a malicious application on the phone or piggyback malicious codes in SDKs used by mobile applications [40]. Attackers extract sensitive files through system APIs (V1) and acquire Identities, phonebooks, SMS if the user grants privileges to the malware. It is feasible since most users cannot make security decisions correctly based on the prompt window's limited information [18, 52]. With root privilege, the attacker can further access all files with the PIN access mode by sending APDUs to the SIM or eSIM. The malware can send acquired files from various channels, including Bluetooth, Wi-Fi, LTE, SMS, to name a few.

The various approaches above target different scenarios. The malware, which can bypass the PIN authentication on rooted devices, is able to attack both the SIM and eSIM. For unrooted devices,

the attacker could launch the hardware attack to the SIM with one-time access. In addition to smartphones, cellular IoT devices using SIM cards are also vulnerable to such a one-time access attack. By default, the SIM PIN is disabled. The attacker can directly use a card reader to extract sensitive files. Even though the victim enables the PIN, the attacker can still insert the SIM sticker along with the SIM card into the device to capture all APDUs, including the PIN, as plaintexts. After getting the context with the one-time access attack, all the following attacks, including traffic eavesdropping, context-based MitM, and impersonation attacks, are feasible for both SIM and eSIM.

Experimental Validation We validate the feasibility of the one-time access attack with hardware and software. With a card reader, we implement a program with python that can automatically extract all the files readable with the PIN, including NAS security context, identities, phonebooks, to name a few. Current commercial SIM stickers do not provide open-source SDK. Thus, we use the SIMTrace [60] board to emulate the SIM sticker's behavior. Results show that it can eavesdrop on all the APDU communication to extract keys, identities, location updates, etc., and further send them out through SMS. We implement the software attacker as an Android application that acquires SIM files on the rooted devices with system APDU interfaces. We test it on Google Pixel, XIAOMI MIX2, and Google Pixel 4a. As shown in Figure 5, we can successfully send APDUs to the SIM and acquire sensitive files. We compare performances when acquiring files with a card reader for SIM and with software for both SIM and eSIM. Regarding the average time to read a file, it takes 10.5 ms for the SIM reading with a card reader. The software leveraging the Android API takes a similar time for SIM and eSIM, which are 11.3 ms (SIM) and 10.8 ms (eSIM).

We also conduct a survey to evaluate the SIM/eSIM usage and user security sensitivity. Our survey results from 189 responses show that 75.8% of participants' phones support eSIM, as shown in Figure 3. Among all participants, 54.1% only use the SIM, 15.3% only use the eSIM, and 30.6% use both SIM and eSIM for their devices. For the hardware attack, the SIM sticker can only be discovered when the user plugs out the SIM, e.g., switching the SIM cards for the current phone. However, it is an infrequent behavior for daily usage. 75.6% users plug out the SIM card less than once a year, which makes the sticker attack potentially last for a long time once the one-time access is successful. Also, 53.4% of participants admit that there are chances for other people to access the phone physically in offices, homes, etc., making them vulnerable to the one-time access attack.

4.2 Traffic Eavesdropping

Given the one-time access attack exploiting the interoperation vulnerability (V2), the adversary can eavesdrop and decipher victim's data and signaling messages. The attack includes two stages: the online trace collection stage and offline decoding stage.

At the online trace collection stage, the attacker camps on the victims' connected cell and eavesdrops on the victim's traffic. The attacker can infer the victim's connected cell through monitoring paging channel [57]. With the inferred cell ID, the attacker camps on the cell based on broadcasted configurations [2]. To locate the

Figure 4: SIM Sticker

```

11:39:30.128: SEND:00A40004026F07Select IMSI File (6F07)
11:39:30.147: RECV:9000 Success (9000)
11:39:30.149: SEND:00B0000009Read IMSI; Length 9B
11:39:30.165: RECV:08 IMSI File Content 9000
11:39:30.167: SEND:00A40004026FE4Select EPSNSC (6FE4)
11:39:30.183: RECV:9000
11:39:30.186: SEND:00B2010436Read EPSNSC; Length 54B
11:39:30.202: RECV:

```

EPSNSC File Content
9000

Figure 5: Malware Attack APDU Traces

Figure 6: Attacker Testbed Setup

Attack Method	Physical access?	Device rooted?	Need phone password?	Need SIM PIN?	Effective to SIM?	Effective to eSIM?
SIM Sticker	Yes	No	No	No	Yes	No
Card Reader	Yes	No	No	Yes	Yes	No
Malware	No	Yes	No	No	Yes	Yes

Table 1: Requirements of Hardware and Software Attacks

victim's traffic in the channel, the attacker needs to know the victim's Cell Radio Network Temporary Identifier (C-RNTI), which can be acquired by monitoring the unciphered signaling messages [38]. Then the attacker could log the ciphered signaling/data traces. At the same time, the attacker collects the bearer configurations for later decryption. Afterward, the attacker performs the one-time access attack. By acquiring the file EPSNSC storing the NAS security context, the attacker can get the K_{ASME} , uplink/downlink NAS counter, and NAS integrity/encryption algorithm identifiers.

At the offline decoding stage, the attacker leverages the NAS security context and logged signaling and data bearer configurations to decrypt the traces. The attacker derives K_{NASenc} , K_{NASint} , and K_{eNB} from the NAS security context. Note the decryption also needs to infer algorithm identifiers for RRC and data decryption. The identifiers are static for the cell, and attackers can infer them by another mobile with software such as MobileInsight [41]. In the worst case, there are three candidate algorithms [1] and the attacker can brute-force them to derive keys for RRC signaling and user data. Combining the keys, the frame counter in unciphered frame headers, and the bearer configurations from the online stage, all the traces eavesdropped can be deciphered offline.

Furthermore, the K_{ASME} can last for a long time, enabling the attack not only before the one-time access but also for later traffic. The K_{ASME} will only change when a new AKA procedure happens, or the 32-bit counter wraps around, which rarely happens due to the low volume of NAS messages. When the UE handover due to mobility, only the K_{eNB} will change, and the K_{ASME} remain the same. In the fast reattachment procedure, operators decide whether to perform the AKA procedure. If the operator does not run the AKA procedure, the network and UE will keep using the same K_{ASME} even the phone restarts.

Experimental Validation We perform the validation on the commercial network and phones. We implement the attacker node upon the open-source srsLTE stack [21] on an Ubuntu 18.04 server with i7-9700K CPU and use USRP B210 as the RF frontend, as shown in Figure 6. We deploy the attacker node with OP-I's configurations. As shown in Figure 7, it contains two components: the online tracer and offline decoder. First, the online tracer monitors the signaling and data frames at the physical layer and logs the ciphered messages as MAC traces. It also collects the bearer configurations

for the decryption. Then we perform the one-time access attack. We read the NAS security context file EPSNSC from the SIM. The offline decoder automatically parses the file to generate K_{NASenc} , K_{RRCEnc} , and K_{UPenc} . It performs the RLC and PDCP decoding and decrypts the UE's traffic. Experiments show the attacker node can successfully eavesdrop on the victim UE connecting with OP-I network. Figure 8 shows the eavesdropped traffic before and after the decryption. The OP-II in the experimental area uses a different modulation and coding scheme in the physical layer, which the current open-source LTE testbed cannot support. We perform a trace analysis and a similar attack procedure can also work with OP-II theoretically.

We perform a user study of the K_{ASME} 's lifetime on OP-I and OP-II. During the testing, devices follow daily usage scenarios with various mobilities, including static, walking, and driving. For both operators, the same K_{ASME} can last for more than one week. Experiments show that OP-II do not perform the AKA procedure when the UE reattaches to the eNB. The UE keeps using the same K_{ASME} after the restart or turning on/off the airplane mode. The attacker can acquire all the traffic without end-to-end encryption, such as DNS, HTTP, or IoT sensory data [31].

4.3 MitM Attack

The attacker can leverage the SIM vulnerabilities and exploit the fast attachment procedure to perform the MitM attack. Prior work exploited the missing integrity protection of the user plane on the physical layer to introduce bit flips and redirect traffic to another destination [53]. The fake base station (FBS) performs attacks including denial of service, network downgrade, UE identities, location tracking attacks, etc [57, 62, 67]. They all use a relay node between the victim UE and commercial eNB, which cannot perform decryption. We propose the context-based MitM, the first attack that can perform the decryption for both the control and data plane and further manipulate signaling and user data. We next introduce our context-based MitM attack.

The attack requires the NAS security context and GUTI, which can be acquired from EPSNSC and EPSLOCI by the one-time access attack. When the victim UE tries to reattach, it scans all supported RF bands, filters out cells of its operator's Public Land Mobile Network (PLMN), and selects the cell with the strongest signal to attach.

Figure 7: Eavesdropping Impl.

Figure 8: Eavesdropped Traffic Decryption

Figure 9: Context-based MitM

Simultaneously, the attacker sets up a fake eNB and EPC with the same PLMN as the victim's operator and loads the context into it. The attacker node increases the transmit power to be the strongest among all victim's candidate cells. The UE will send the Attach Request with GUTI to the attacker node to perform the fast reattachment.

After receiving the Attach Request, the attacker node returns the NAS Secure Mode Command (SMC) with corresponding 4-byte NAS-MAC derived from the stolen security context. After the UE verifies the NAS SMC's integrity, it connects the Internet with the fake eNB and EPC. All the data packets are forwarded to the Internet by the P-GW in the fake EPC. Attackers can eavesdrop on all the traffic and perform heterogeneous MitM attacks such as IP/ARP/DNS spoofing, SSL hijacking, SSL stripping, to name a few. It risks the victim's email, bank, or social media accounts and further causes financial losses [14, 35]. The attacker can also derive the integrity key so that previous solutions, e.g., adding integrity protection for the 4G/5G data plane, cannot prevent this attack [53, 54].

Experimental Validation We implement the attacker node as shown in Figure 9 upon the srsLTE, which includes a fake eNB and EPC, the dynamic context loader, and the traffic MitM module. We validate the attack on OP-I and OP-II with commodity phones including Google Pixel 4a, XIAOMI MIX2, and iPhone 7 as the victim UE. When performing the one-time access attack, the attacker sets the victim to airplane mode and configures the attacker node with the same PLMN as the victim's operator on an unlicensed spectrum, following the ethical guidelines. We ensured that experiments do not harm real-world users.

The dynamic context loader loads the acquired EPSLOCI and EPSNSC and updates the attacker node's cached context. It will also derive the K_{NASenc} and K_{NASint} . After receiving the Attach Request from the victim UE, the attacker node will generate the NAS SMC integrity protected by the K_{NASint} . Experiments show that the victim validates the NAS SMC's integrity and the attacker node passes the UE authentication. The victim completes the following attach procedure and transmit all the data and signaling through the attacker node. With the USRP B210 as the front end, the attacker node can maintain the highest signal strength among all the candidate cells within 10 meters from the victim. We believe the distance can be larger with more powerful RF devices.

The traffic MitM module can perform heterogeneous MitM attacks. Results show the victim UE for both OP-I and OP-II will connect to the attacker node. We build the showcase MitM attack including DNS spoofing and SSL stripping. Leveraging the DNS spoofing, the attacker could forge the DNS packets to make the victim connect to phishing websites. As shown in Figure 10(a), the victim will enter the phishing bank website with the correct domain name. The attacker could set up the phishing page layouts identical to the

(c) Captured Account Username and Password

Figure 10: MitM Attack Validation

real login page, which makes the account information vulnerable. Even for websites that enable HTTPS, it is still feasible to attack the user session with SSL stripping. The SSL stripping could downgrade the connection from HTTPS to HTTP to be able to perform the MitM attack [30]. Figure 10(b) shows that the HTTPS connection to the email webpage is downgraded to HTTP. The attacker node could capture the username and password from the victim's request with an HTTP proxy, as shown in Figure 10(c).

4.4 Impersonation

The attacker can further impersonate the victim's identity by leveraging the vulnerabilities in SIM and fast reattachment service. A recent study reveals that the lack of physical layer integrity protection along with ICMP reflection mechanisms allows the impersonation attack [54]. The prior attack is limited to the user plane and requires the attacker's relay node to be close to the victim. We leverage the SIM vulnerabilities with the fast reattachment to impersonate the victim on both the control and user plane. This attack can be launched remotely during the impersonation, thus more challenging to be detected.

The impersonation attack includes three steps. First, the attacker acquires the fast reattachment context files (EPSLOCI and EPSNSC) from the SIM by the one-time access attack. Second, the attacker derives the K_{NASint} and K_{NASenc} with K_{ASME} and integrity algorithm identifier from the EPSNSC. The UE fast reattachment requires the victim's NAS keys, GUTI, and network capability. The network capability of a device is public information. Finally, the attacker sets up a fake UE with SDR devices to send the NAS Attach Request with GUTI from the EPSLOCI, which is integration protected by the NAS-MAC derived from K_{NASint} . After the MME authenticates the fake UE and grants it access to the Internet, the attacker can


```

19:14:29.730205 [S1AP] [I] Received Initial UE message -- Attach Request
19:14:29.730211 [NAS] [I] Attach request -- M-TMSI: 0xd9bc07a GUTI attach
19:14:29.730213 [NAS] [I] Attach request -- eNB-UE S1AP Id: 5 request with
19:14:29.730214 [NAS] [I] Attach request -- Attach type: 2 victim's context
...
19:14:29.730231 [NAS] [I] Attach Request -- Found previously attached UE.
19:14:29.730238 [NAS] [I] Integrity check ok. Local: count=6, Received: count=6
19:14:29.730241 [NAS] [I] GUTI Attach -- NAS Integrity OK. UL count 6, DL count 3
19:14:29.730247 [NAS] [I] Generating KeNB with UL NAS COUNT: 6
...

```

Successfully attached as the victim

Figure 11: Impersonation Validation

drain the victim's data plan or attack other entities with the victim's identity. Some operators verify device access to applications or services through the cellular identity/keys [54]. It protects users when the application or SMS one-time passwords are leaked. However, the SIM impersonation attack could still bypass the checking and cause financial losses. Note that the attacker with the context can launch attacks far from the victim, which makes locating attackers challenging for law enforcement agencies.

Experimental Validation We perform the validation on the SDR testbed with the same hardware configurations before. With one USRP B210 as the eNB, we use commodity phones with programmable SIM cards that support fast reattach service to connect to our private LTE network. We test Google Pixel, Google Pixel 4a, XIAOMI MIX2, and iPhone 7. Through the one-time access, the attacker acquires the EPSLOC and EPSNSC from the SIM. Another USRP B210 runs srsUE as the rogue UE. The rogue UE loads the context, derives K_{NASint} and K_{NASenc} , and sends NAS Attach Request with the valid GUTI, UE network capability, and NAS-MAC derived from K_{NASint} . The results show that the rogue UE will be authenticated as the victim's identity and set up the LTE connection. Figure 11 shows corresponding logs at the eNB. We also perform trace analysis on the commercial network successfully. Traces show that OP-II do not run a new AKA procedure when the UE performs the fast reattach, allowing attackers to impersonate the victim's identity.

5 SOLUTION

We design SecureSIM to enable flexible access control and ensure proper authentication at SIM. Our solution seeks to address two issues:

- **How to enable fine-grained file access control for heterogeneous entities?** The current SIM includes about 200 files, and different entities have various access requirements for these files. We devise a novel access graph to organize the SIM files. Moreover, we examine common scenarios that require SIM access, including all off-card units and in-card applets. We consequently design a multi-rooted tree in the access graph to offer fine-grained access control to various entities. To improve usability, we design APIs for the access graph, which provides a flexible way for operators to manage file access. Operators specify the intended access policies for various entities. SecureSIM detects conflicts and generates the access graph to perform its fine-grained access control.
- **How does the SIM distinguish various entities and ensure security?** Fine-grained access control itself cannot prevent attackers from acquiring files. Attackers may pretend to be another entity for file access. Proper authentication is thus needed (V1). It should defend against authentication attacks from the card reader, SIM sticker, malware, physical thefts, etc. SecureSIM devises the

Profile Type	Included Files
Identity	IMSI (6F07), Phone number (6F40), etc.
Security	NAS Security Context (6FE4), CK/IK (6F08), etc.
Location	EPS location Info (6FE3), Location Area (6F7E), etc.
User Information	Phonebook (5F3A), SMS (6F3C), etc.
Operator Information	Operator Name (6F46), Icon (6FDE), etc.
Network Configurations	Services Table (6F56), Access Technology (6F60), etc.

Table 2: SIM Profile Node Types

Type	Class	Access Schema
Off-card	Modem	/Offcard/Modem/[UE Category]/[Modem ID]
	System App	/Offcard/App/SystemApp/[App ID]
	User App	/Offcard/App/UserApp/[App Type]/[App ID]
	Operator	/Offcard/Reader/Operator/[Operator ID]
	Subscriber	/Offcard/Reader/Subscriber/[Subscriber ID]
In-card	Services	/Incard/[Service Type]/[Applet ID]

Table 3: Access Schema

certificate-based solution to ensure authentication, flexibility, and scalability. We design the certificate chain for SecureSIM to distinguish various entities. The certificate ID binding protects the SIM from physical thefts. To protect the APDU communication channel (V3), we propose selective encryption to ensure data secrecy with negligible overhead. With the authentication and APDU channel protection, SecureSIM protects SIM files such as identities, security context, etc., to prevent traffic eavesdropping, MitM, and impersonation attacks (V2). Most users do not enable the current PIN-based protection. SecureSIM makes an effort to be more user-friendly. We thus co-design automated verification and revocation with the modem to protect users without degrading usability.

5.1 Control SIM Access with Graph

To address the first issue, we leverage two types of relations in our design. The relation among files enables a flexible approach to manage these SIM files. The relation between entities facilitates the access to be fine-grained for various scenarios. We consequently devise a novel access graph to model the relations in the SIM access control.

5.1.1 Access Graph. The access graph, as shown in Figure 12, has three types of nodes: file node, profile node, and schema node. Each file node denotes a single file stored in the current SIM. Each profile node denotes a category of files. Each schema node contains the policy blueprint for a given entity.

We design the profile node to organize files into a small number of categories for systematic access management. With more services added to the SIM, the first issue is how to manage diverse SIM files. Different from servers or smartphones, which have enough processing capabilities and storage to manage a large number of files, SIM's processing speed and storage are limited. It is burdensome to manage each file's access control for more than 200 files in the SIM, which requires an access control design that fits the hardware capabilities. To better manage files with the same requirements, we examine current SIM files, and classify them into six categories (shown in Table 2). Each category is with a profile node. In the access graph, each file node connects to only one profile node. Every profile node contains multiple files sharing the same access control permissions.

Figure 12: Managing SIM File Access for Off-card Units and In-card Applets with Access Graph

We design the schema node to classify heterogeneous entities. Organizing various entities systematically and flexibly is challenging. By reviewing current usage scenarios, we classify entities into off-card units and in-card applets, which are organized as a multi-rooted tree. The schema node provides a convenient and intuitive way to construct the tree based on practical scenarios. All profile nodes connect to the tree to check entities' privilege. Each schema node contains the node name and access privileges. We prescribe the tree structure with access schema in Table 3. Symbol / splits schema nodes into multiple layers. Symbol [] denotes extensible schema nodes. For example, the UE category indicates the modem's capabilities. A schema node Modem links [UE Category] schema nodes, enabling operators to customize the access control for specific devices such as NB-IoT. The applications contain system applications and user-installed applications, which are further classified based on application types. The SIM grants access to applets based on the service type. The finest granularity for all entities is the unique ID, such as the application ID [6, 7], the modem ID [49], and the applet ID [32]. The schema nodes contain each profile node's privileges, including *Read (R)*, *Update (U)*, *Delete (D)*, etc., which complies with the current standard [33].

The graph's characteristics benefit access control management. Schema nodes follow the least privilege principle: an entity is given the minimum levels of access it needed. The fine-grained entities should have higher or equal privileges than coarse-grained entities [10, 19]. The more fine-grained child node can possess higher privileges than its parent. For example, the system application can *Read* identities, but a general application cannot. Each schema node inherits its parent privilege by default. When a schema node's privilege changes, all children's default privileges also change accordingly. When an entity accesses files, SecureSIM first checks the file's profile nodes, then follows schema nodes and grants access based on the corresponding leaf node's privilege.

The access graph is flexible for future extensions. When adding new files, the access graph can link each file to one of the current profile node. If new files cannot fit current profiles, the access graph supports extending new profile nodes for them. The access graph supports heterogeneous scenarios by extensible schema nodes. In rare cases, If the SIM wants to further control the access for a specific file, it can add a new profile node, change the file's profile node to the new one, and set access privileges in schema nodes.

5.1.2 Access Policy. The access graph provides fine-grained SIM access control. However, directly setting up the privilege for every

Access Policy	ID	Sec	Loc	UserInfo	OpInfo	NetConf
/Offcard/Modem/Cat(1,3)/*	R	R,U	R,U	R,U	R	R,U
/Offcard/App/SystemApp/*	R		R	R,U	R	
/Offcard/App/UserApp/Operator/{ID1,ID2}	R		R	R	R,U	
/Offcard/App/UserApp/*						
/Offcard/Reader/Operator/OP-I	All ⁴	All	All	All	All	All
/Offcard/Reader/Subscriber/0001	R	R	R	R,U	R	R
/Incarn/Identity/*	R,U					

Table 4: Access Policy Example

schema node is error-prone. Privileges may violate the least privilege principle. Duplicate sibling nodes with conflicting privileges may also exist. We provide the access policy as APIs for operators to grant privileges.

Operators need a flexible way to grant the same privilege to many entities. For example, the SIM may grant *Read* identity privilege to all system applications without specifying the application ID. The access policy use notation similar to the regular expression. The * matches any name. {...} to represent a list of names. (a, b) represents a range from a to b. Table 4 shows an example access policy for SIM files. Note that SecureSIM forbids all applications to acquire the security files to prevent software attackers. The modem plays a critical role in network connection and could read or update most of the files. A wildcard covers all system applications without specifying the ID. The operator can restrict the SIM usage to only a range of UE categories. A list of operator applications can read the operator information for add-on services.

The operator grants privileges by access policies with the following steps: First, the operator specifies policies for each profile type; Second, SecureSIM checks policy conflicts. The conflicting access control could lead to several damages. A higher privilege to an entity leaks keys or privacy, and an insufficient privilege blocks normal functions. SecureSIM leverages the graph's characteristics to assist the automated privilege checking. SecureSIM checks privileges with the least privilege principle for the parent and children nodes, detects duplicate or conflicting privileges for sibling nodes, and verifies the tree structure correctness with the access schema. Upon detecting conflicts, SecureSIM notifies the operator to fix till no conflict exists; Finally, SecureSIM generates the access graph on the SIM with access policies. With the access graph, SecureSIM controls various entities' access as described in Section 5.1.1.

5.2 Rethink SIM with Certificates

The second question for the SIM design is: **How does the SIM distinguish various entities and ensure security?**

To identify various entities, the SIM requires authentication before the entity access files or performs functions such as AKA procedure. The current SIM takes password-based authentication

⁴All grant privileges for all operations including Read, Update, Append, Delete, Create, Activate and Deactivate.

Figure 13: SIM Certificate Chain

for years, and the eSIM follows the same approach. The PIN verification is easy to deploy considering the SIM's limited storage and computation capability. SIM only allows limited tries (usually three times), making it robust to brute force and side-channel attacks. However, the current PIN only contains 4-8 digits and lacks enough strength. Furthermore, simply extending the PIN mode to multiple modes cannot distinguish various entities. It is not flexible for heterogeneous entities.

Another solution is to improve the strength with a shared key between the SIM and each entity. Nevertheless, the shared-key authentication is not scalable. Entities must own different keys to be distinguished. The SIM needs to store all keys and corresponding entity names for the fine-grained access control. Although the SIM with 50K free memory could hold more than one hundred keys, managing the keys is still challenging. Whenever the operator adds a new entity, all SIM cards need to update keys remotely. The cumbersome distribution procedure limits the scalability and cannot perform the fine-grained identity checking and access control.

SecureSIM takes the certificate-based authentication, which is much more scalable than the password or shared key authentication. The SIM does not have to store keys for each entity but only a single certificate or several certificates. The certificate is flexible for heterogeneous entities. We design the certificate chain for SecureSIM with ID binding to ensure security and flexibility. To protect the plaintext APDU channel, we design selective encryption for SIM-modem communication with negligible overheads. Considering SIM's limited processing capability, performing all tasks on the SIM will affect the user experience. We leverage the modem to accelerate SecureSIM. We devise the automated certificate verification and revocation procedure to improve security without hindering usability. SecureSIM design cannot be directly applied to existing phones and require operators and modem manufacturers to update the SIM applets and modem firmware. All functions can be updated over the air and do not need extra hardware. We discuss how SecureSIM could leverage the current infrastructure for each scheme to be deployed gradually. Now we introduce the details of certificate-based SecureSIM design.

5.2.1 Certificate Chain with ID Binding. We design the certificate chain as shown in Figure 13 for fine-grained access control (V1). In the certificate chain, the root certificate is self-signed and organized by a trustworthy third party such as GSMA [23]. Operators collaborate with SIM vendors to manufacture SIM cards and sign certificates for SIMs, subscribers, and applets. Independent of operators, the modem original equipment manufacturer (OEM) manufactures modems according to 3GPP standards. The root certificate signs the modem OEM certificates, and each OEM signs modems. The current eSIM system also leverages the certificates to validate

Figure 14: SIM-Modem APDU Encryption

operators' SIM packages and eSIM chips. GSMA defines specific Public Key Infrastructure (PKI) for eSIM remote provisioning, and signs certificates to operators and OEMs [24]. The current eSIM infrastructure can be extended to support the SecureSIM certificate chain. The root certificate also signs certificates for operating systems, which further sign applications. Current operating systems such as Android require the application is signed before installation. For example, when the developer publishes the application to the google play store, Google provides the platform to sign and publish the application [22]. SecureSIM could leverage the platform pipeline to sign applications automatically.

The certificates include a customized attribute *SchemaName* that follows the access schema for access control. SecureSIM leverages the *ModemID* in the *SchemaName* to prevent physical thefts. Each modem possesses a unique modem ID. The user needs to send the modem ID to the operator when requesting the SIM for the first time. The operator sets the ID into a whitelist in the SIM. Only whitelisted modems can pass the authentication. If the user wants to change the device, he can add more modem IDs into the list through authenticated devices or request to the operator. For IoT devices, the SIM can store a modem ID pattern as the access policy to whitelist a collection of modems. Through the Modem ID binding, the SIM can prevent physical thefts from stealing the data plan or the identity. In current eSIM deployments, operators also require users to submit the eSIM chip ID before activating the eSIM usage. The modem ID could also be submitted simultaneously. Regarding usability, registering the Modem ID does not require extra steps comparing with the existing deployment.

5.2.2 Selective Encryption. Limiting the SIM access to certified roles cannot prevent eavesdropping on the plaintext APDU channel (V3). We design the APDU encryption mechanism between the SIM and modem to prevent the SIM sticker attack. Generally, asymmetric encryption takes relatively more time than the symmetric encryption [45, 48]. Considering SIM's limited resources, we first perform a Diffie-Hellman (DH) key exchange and later use the generated key to perform the symmetric encryption for later APDU traffic.

As shown in Figure 14, the SIM generates a random SIMChallenge and sends it to the modem. The modem generates the DH Key pair and uses its secret key of the certificate to sign the SIMChallenge and the modem DH public key. Then the modem sends the

Figure 15: Automated Verification Procedure

signature ModemSign with signed information and its certificate to the SIM. The SIM prevents the replay attack by a random SIM-Challenge every time. After receiving the information, the SIM first verifies the modem certificate and SIMChallenge, then authenticates the ModemSign with the certificate's public key. If the modem passes all the checking, the SIM follows a similar way to send its DH public key with the signature SIMSign and its certificate. After the modem validates the SIM, both sides derive the key from the shared secrets and encrypt all later APDUs with the symmetric encryption algorithm such as AES. We further devise a selective scheme to reduce the overhead. Most of the APDUs sent by the modem are standardized and do not contain sensitive information, such as selecting a file. These APDUs can be plaintext so that the SIM can eliminate the decryption latency. The selective encryption only ciphers the sensitive information, including file contents returned by the SIM, updated file contents sent by the modem, and authentication procedures such as LTE AKA. The selective encryption ensures security with marginal overheads. Current eSIM performs the package decryption and certificate verification inside the eSIM chip. Considering the modem's hardware is more powerful than eSIM chips, it is feasible to deploy the functions inside a modem by firmware updates.

5.2.3 Automated Verification. SecureSIM enables an automated verification scheme, as shown in Figure 15. Considering the limited RAM and computation resources, performing all certificate authentication in the SIM increases the processing latency and affects the user experience. To speed up the verification procedure, SecureSIM offloads the certificate validation to the modem after authenticating its certificates. Comparing with OS or applications, which are vulnerable on root devices, the attacker is hard to compromise a modem. Thus, the modem can be a trustworthy device for application authentication. After the modem validates the application's certificate, it sends the application's *SchemaName* to the SIM and enables the APDU communication with selective encryption. For in-card applets, SecureSIM verifies their certificates automatically before they access files. Through automated verification, SecureSIM can distinguish various entities with negligible overheads.

For both the SIM and the modem, SecureSIM's design does not change the file contents, the APDU interfaces, or require extra hardware. All functions can be updated through SIM or modem firmware updates over the air. SecureSIM schemes could be enabled based on the SIM/modem's capability, which provides backward compatibility for current devices and can be rolled out gradually. Considering the usability, the current SIM PIN is disabled by default and requires users to enable it manually. SecureSIM does not require users to manually enable the PIN or enter the PIN every time the phone starts. All the verifications are automated through certificates and do not affect users' daily usage.

5.2.4 Certificate Revocation. To ensure security when certificates are compromised, the operator maintains a Certificate Revocation

List (CRL), including revoked certificates before the expiration date. The SIM should not trust the revoked certificates. The SIM performs the automated revocation checking with two stages. First, SecureSIM periodically updates the CRL from the operator over-the-air to check the validity of modems, in-card applets, or subscribers. Then the modem is responsible for maintaining, updating, and handling the application CRL considering the limited storage on the SIM. If the CRL is too large to hold in the SIM, it can further extend to standardized Online Certification Revocation Protocol (OCSP) to support real-time certificate checking [51]. Current eSIM PKI also provides certificate revocation schemes through CRL or OCSP. SecureSIM could leverage the current infrastructure for deployments. Researchers proposed other certificate revocations schemes for IoT, which can also apply to the SIM [5, 12]. It is challenging for the SIM to verify the certificate's validity period because the current SIM does not possess a clock. Thus, it must reliably acquire the time. SecureSIM acquires the time through APDU from the modem, which is one of the standardized commands supported by commercial modems [17]. SecureSIM only updates the timestamp from whitelisted modems. The SIM also records the most recent timestamp retrieved before to approximate the current time when validating through other hardware such as card readers. It can provide enough security considering the difficulty of faking the certificates or reusing outdated certificates.

5.3 Security Analysis

We analyze the security of SecureSIM in comparison to the current SIM design.

LTE Key Leakage The LTE key leakages allow attackers to perform eavesdropping, MitM, impersonation attacks. SecureSIM provides fine-grained access control and certificate verification. Only whitelisted modems can acquire the security-related keys. The attacker cannot read them by hardware such as a card reader or a random modem out of the whitelist. The malware cannot pass the automated verification even with the root privilege.

APDU MitM SecureSIM protects the APDU communication with selective encryption between the SIM and modem. Both sides sign the APDUs in the key agreement procedure so that the attacker cannot forge or manipulate them. The symmetric encryption for APDUs protects all sensitive files.

Physical Thefts SecureSIM leverages the modem whitelist to restrict the SIM usage on credible devices. Physical thefts cannot use the stolen SIM on other devices for data service or impersonate the victim's identity.

SIM Cloning The 2G/3G SIM algorithm's weakness allows SIM cloning attack [50, 61]. LTE SIM cards improve the algorithm security, but attackers can still get the master K to clone the SIM from side-channel attacks like differential power analysis [34, 42]. Improved implementations with anti-cloning mechanisms can defend these attacks [8, 29]. SecureSIM only grants whitelisted devices to perform AKA authentication, increasing the difficulty of collecting traces for compromising. Even the master K leaks, attackers cannot compromise the certificates and clone the SIM.

SIM Swapping SIM swapping occurs when the attacker tricks the operator to port the victim's phone number to the attacker's SIM.

For example, the attacker acquires the victim’s personal information through phishing emails or social engineering and claim that they have lost their phone. While SecureSIM prevents the attacker access the SIM from unauthorized devices, defending SIM swap attack still requires the operator to improve the security checking like a mandatory strong account password.

Abused Modem If attackers extract modem keys and impersonate a certified modem, the SIM security is risked. However, it is hard to be achieved in practice. First, the modem firmware is usually close-sourced and hard to be attacked. Modems also provide separate hardware and software stacks from the host device, making it difficult to execute malicious codes and be abused. Second, even if the attacker abuses a specific modem, critical files such as keys in SecureSIM could only be accessed by whitelist modems. Abused modem with other Modem ID cannot access keys successfully.

6 IMPLEMENTATION

We implement SecureSIM with Javacard, one of the most prominent platforms for smart card development. Javacard has similar hardware capabilities to the commercial SIM. Operators could build SIM with smart cards and install an applet running SIM functions. Javacard defines a subset of the Java programming language and provides APIs for smart cards [36]. We develop a SIM card applet as a platform that can support standard SIM file operations and the MILENAGE algorithm to perform the LTE AKA mutual authentication. It consists of 5263 lines of Java code. To the best of our knowledge, there is no open-source SIM implementation on the smart card. We release the applet at [20].

We deploy the SecureSIM based on our SIM applet. We develop an access policy parser with python. It parses access policies provided by operators, checks for conflicts, and generates the corresponding access graph for the SIM applet. The SIM applet holds files as standardized in [3]. In the standard, each file contains a header including file format, length, etc. We extend the header with the file’s profile category. For each profile, schema nodes encode granted operations with one byte, which complies with the standard [33]. When an entity starts a new APDU session, the SIM matches the *SchemaName* from the certificate verification procedure. By checking the schema tree, SecureSIM could acquire the entity’s privileges for each profile type. When the entity performs file operations such as Read or Update, SecureSIM grants access based on the file’s profile.

SecureSIM adopts the certificate-based authentication and communication. We generate the certificate chain with X.509 v3 standards and add customized attribute *SchemaName*. For certificate algorithms, we select ECDSA for its smaller key size compared with RSA. The current phone modem firmware is closed-source. Thus, we deploy a module with Java in an Android application to simulate the modem’s selective encryption, automated verification, and certificate revocation procedure. We use the Elliptic-curve Diffie-Hellman (ECDH) over the secp256k1 curve to perform the key agreement described in section 5.2.2 and generate shared secrets. The later symmetric encryption uses the AES-128-CBC algorithm. For automated verification, the SIM first authenticates the module with the modem certificate CERT.Modem.ECDSA, then the module handles all the other certificate verifications automatically. The

Figure 16: Performance comparison of the commercial SIM card (LTESIM) and SecureSIM

Figure 17: Cert Verification

Figure 18: Encryption

module also collaborates with SIM for storing and updating the certificate revocation list.

7 EVALUATION

We evaluate the performance of SecureSIM on J3R180 Javacard supporting Java Card 3.0.5 specifications with 180KB EEPROM and 8KB RAM. We assess the performance with Google Pixel 4a with Qualcomm Snapdragon 730G running Android 10. We first compare the overall performance of the commercial SIM card (LTESIM) and SecureSIM. We then evaluate each module’s performance, including certificate verification, selective encryption, and access policy checking. We further measure the energy overhead introduced by new mechanisms in SecureSIM. The results show that the overhead of SecureSIM is marginal compared with the current SIM design.

Overall Evaluation We use SIMTrace to record all the APDUs when the modem loads SIM files once the SIM is inserted into the phone. Then we replay the APDUs and measure the overall performance at the SIM initial stage between the commercial SIM card and SecureSIM. The results show that SecureSIM takes $6.3 \pm 0.1s$, while the commercial SIM takes $5.4 \pm 0.2s$, as shown in Figure 16(a). The execution time increases mainly because of certificate verification. It only happens once at the start of the APDU session and will not incur overhead for follow-up communication. Our implementation is based on the generic smart card API. It cannot directly control the hardware resource. Customized devices could further improve execution efficiency. Moreover, the eSIM chip with more powerful computational resources can also reduce the overhead.

Certificate Verification We measure the time for the automated verification procedure. SecureSIM delegates its authentication task to the modem after the SIM validates its certificates. We compare the SIM-modem and modem-application verification time. As shown in Figure 17, the SIM-modem authentication time is $639 \pm 23ms$. It only happens once, when the modem starts the APDU session. The session can last within the SIM power cycle, which can be hours or days. SecureSIM leverages the modem to accelerate application authentication. The automated verification can reduce the modem-application verification time to $6.5 \pm 1.2ms$. It only happens once when the app requires SIM access and incurs marginal overhead.

Encryption We measure APDUs that read files from the SIM with and without the selective encryption. With symmetric encryption, reading a file needs on average $11.5 \pm 0.6ms$, comparable with the LTESIM without encryption ($10.9 \pm 0.6ms$), as shown in Figure 18. Furthermore, we compare the performance of encrypting all APDUs (naive encryption) and selective encryption in SecureSIM. Experiments show that selective encryption can reduce 15% of the average time (from $13.6 \pm 0.9ms$ to $11.5 \pm 0.6ms$). We measure the benefits of selective encryption at the initial stage. The results show that selective encryption could reduce the average package loading time by 1.1s (from $7.4 \pm 0.1s$ to $6.3 \pm 0.1s$).

Access Policy Checking SecureSIM matches the entity's *SchemaName* with the access graph once and gets corresponding access policies. Then SecureSIM only needs to check the file's profile type to grant access. We measure the operation execution time of SecureSIM access policy checking and the LTESIM with access mode checking. Figure 16(b) shows that SecureSIM ($9.2 \pm 0.4ms$) will not introduce extra overhead compared with LTESIM ($9.5 \pm 0.7ms$).

Energy Consumption We further measure the energy consumption of SecureSIM and LTESIM. We measure the 1-hour energy consumption of an application that reads SIM files through APDU every 50ms. The results show that SecureSIM consumes comparable power (0.064%) as LTESIM (0.062%), as shown in Figure 16(c). The increased energy consumption (3.2%) caused by encryption is marginal. We measure the modem's daily APDU traffic pattern with SIMTrace. During the whole session, the modem sends 0.2 APDU/s. The power consumption will be more insignificant. We also measure the energy overhead caused by automated verification. Our test application launch the certificate verification every 100ms. The 1-hour test consumes 1.1% of the power. Usually, the application only needs to authenticate once at the start, thus incurring negligible amortized energy overhead.

8 DISCUSSION

Here we discuss whether our findings apply to 5G. The 5G SIM does not transmit the permanent identifier, such as IMSI, during the attach to improve privacy protection. However, it still stores the security context and temporary identities for fast reattachment. 5G defines three authentication methods: 5G-AKA, EAP-AKA, and EAP-TLS. The 5G-AKA and EAP-AKA utilize a similar pre-shared key scheme as the LTE AKA. 5G introduces EAP-TLS for private networks or IoT environments to handle the devices without SIM to perform the AKA authentication, such as laptops or IoT sensors. EAP-TLS sets up the TLS connection based on devices' certificates. EAP-TLS does not rely on the symmetric master key K but generates keys from the TLS session. However, current vulnerabilities still exist in 5G. All three methods derive a shared key called K_{AMF} , which is still stored in the SIM. Furthermore, the 5G SIM adds support to store intermediate authentication keys. 5G also supports the fast reattachment with the cached security context. These files are under current PIN-based protection and put 5G users at risk of traffic eavesdropping, MitM, and impersonation attacks.

The 4G/5G trust models require the long-term symmetric key. All the keys depend on the master key K , which exposes high operational risk during the multi-layer key derivation and transmission. As future work, certificates in the SIM make a case to redesign the

5G authentication scheme. Certificates can derive one-time keys to set up security between the UE, serving network, and home network. Multi-layer keys can be independent and flexible for heterogeneous devices and services.

9 RELATED WORK

The current SIM design has been largely inherited from the legacy 2G/3G networks, where end devices do not assume much intelligence. Prior studies have exploited its implementation vulnerabilities to clone SIM cards [42], track user location [59], or attack VoLTE calls [13]. Recent efforts exploit the vulnerability in the remote SIM provisioning procedure to exhaust the memory at eSIM [15, 46]. In contrast, we look into the vulnerability of in-card file access for both SIM and eSIM, and use the obtained information to reconstruct the keys for 4G/5G attacks. We thus allow for attacks against both the victim device and the 4G/5G infrastructure within the standards. Therefore, we differ from existing studies that use the LTE design or implementation vulnerabilities to attack the victim user [43, 53, 54, 57] or the network [26, 39, 66].

To the best of our knowledge, we report the first effort on authentication and access control for SIM/eSIM. We take a novel approach with the access graph to enable fine-grained access control on SIM files. It differs from the recent graph-based vulnerability analysis in account access [25]. Prior efforts analyzed the mobile operating system's access control and vulnerability for both Android [56] and iOS [16]. They further proposed access control designs for mobile operating systems [11, 47, 55] and IoT applications [56, 68]. Such OS or IoT access control schemes cannot prevent the information leakage at SIM/eSIM.

10 CONCLUSION

The security of SIM/eSIM is critical to both mobile users and 4G/5G infrastructure operations, since it contains various IDs, security keys, and personal information of a mobile user. Despite its importance, SIM/eSIM security is a largely unaddressed topic in the research community. This is partly because SIM/eSIM is widely considered as a closed module only accessible to mobile operators and vendors. However, as recent initiatives seek to open up SIM and make it more programmable (e.g., adding new over-the-air services into SIM/eSIM [17]), the conventional wisdom probably will not hold. As a result, the current SIM protection is not sufficient. The fundamental problem is that, SIM/eSIM does not offer proper authentication to various in-SIM applets and off-card units. Moreover, it does not provide fine-grained access control to hundreds of in-SIM files that store keys and personal data. Consequently, various attacks are made feasible. In this work, we uncover three such vulnerabilities, devise new attacks against both the victim device and the 4G/5G infrastructure, and propose solutions. Our prototype and evaluation have validated both the attacks and our defenses.

Acknowledgements. We greatly appreciate our anonymous shepherd and reviewers for their insights and constructive comments. We thank all survey participants for their valuable feedback. This work is partly supported by NSF CNS-1910150 and NSF CNS-2008026.

REFERENCES

- [1] 3GPP. TS24.301: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3, Jun. 2013.
- [2] 3GPP. TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, Sep. 2019.
- [3] 3GPP. TS31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application, Dec. 2020.
- [4] 3GPP. TS33.401: 3GPP System Architecture Evolution (SAE); Security architecture, Jul. 2020.
- [5] ALRAWAIS, A., ALHOTHAILY, A., HU, C., AND CHENG, X. Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing* 21, 2 (2017), 34–42.
- [6] Android application id. <https://developer.android.com/studio/build/application-id>.
- [7] Apple bundle id. https://developer.apple.com/documentation/appstoreconnectapi/bundle_ids.
- [8] ARAVAMUTHAN, S., AND THUMPARTHY, V. R. A parallelization of ecdsa resistant to simple power analysis attacks. In *2007 2nd International Conference on Communication Systems Software and Middleware* (2007), IEEE, pp. 1–7.
- [9] At&t pin setting. <https://www.att.com/support/article/wireless/KM1000485/>.
- [10] AU, K. W. Y., ZHOU, Y. F., HUANG, Z., AND LIE, D. Pscout: analyzing the android permission specification. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), pp. 217–228.
- [11] BUGIEL, S., HEUSER, S., AND SADEGHI, A.-R. Flexible and fine-grained mandatory access control on android for diverse security and privacy policies. In *22nd USENIX Security Symposium (USENIX Security 13)* (2013), pp. 131–146.
- [12] CEBE, M., AND AKKAYA, K. Efficient certificate revocation management schemes for iot-based advanced metering infrastructures in smart cities. *Ad Hoc Networks* 92 (2019), 101801.
- [13] CHALAKKAL, S., SCHMIDT, H., AND PARK, S. Practical attacks on volte and vowifi. *ERNW Enno Rey Netzwerke, Tech. Rep* (2017).
- [14] CHIEW, K. L., YONG, K. S. C., AND TAN, C. L. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications* 106 (2018), 1–20.
- [15] CHITROUB, S., ZIDOUNI, N., AOUADIA, H., BLAID, D., AND LAOUAR, R. Sim card of the next-generation wireless networks: Security, potential vulnerabilities and solutions. In *2018 2nd European Conference on Electrical Engineering and Computer Science (EECS)* (2018), IEEE, pp. 502–509.
- [16] DESHOTELS, L., CARABAS, C., BEICHLER, J., DEACONESCU, R., AND ENCK, W. Kobold: Evaluating decentralized access control for remote nxpc methods on ios. In *2020 IEEE Symposium on Security and Privacy (SP)* (2020), IEEE, pp. 1056–1070.
- [17] ETSI. TS 102 223: Smart Cards;Card Application Toolkit (CAT), Jul. 2019.
- [18] FELT, A. P., HA, E., EGELMAN, S., HANEY, A., CHIN, E., AND WAGNER, D. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the eighth symposium on usable privacy and security* (2012), pp. 1–14.
- [19] FERRAILOLO, D. F., SANDHU, R., GAVRILA, S., KUHN, D. R., AND CHANDRAMOULI, R. Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4, 3 (2001), 224–274.
- [20] Flora esim applet. <https://github.com/Project-Flora/Flora-eSIM>.
- [21] GOMEZ-MIGUELEZ, I., GARCIA-SAAVEDRA, A., SUTTON, P. D., SERRANO, P., CANO, C., AND LEITH, D. J. srLTE: An open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization* (2016), pp. 25–32.
- [22] Google play app signing. <https://developer.android.com/studio/publish/app-signing#generate-key>.
- [23] Gsma. <https://www.gsma.com/esim/gsma-root-ci/>.
- [24] GSMA. SGP.22 Technical Specification v2.2.2, Jul. 2020.
- [25] HAMMANN, S., RADOMIROVIĆ, S., SASSE, R., AND BASIN, D. User account access graphs. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), pp. 1405–1422.
- [26] HE, L., YAN, Z., AND ATIQUZZAMAN, M. Lte/lte-a network security data collection and analysis for security measurement: A survey. *IEEE Access* 6 (2018), 4220–4242.
- [27] HE, S., AND PAAR, I. C. Sim card security. In *Seminar Work, Ruhr-University of Bochum* (2007).
- [28] Heicard sim. <https://www.heicard.com/en/>.
- [29] HERBST, C., OSWALD, E., AND MANGARD, S. An aes smart card implementation resistant to power analysis attacks. In *International conference on applied cryptography and network security* (2006), Springer, pp. 239–252.
- [30] HOSSAIN, M. S., PAUL, A., ISLAM, M. H., AND ATIQUZZAMAN, M. Survey of the protection mechanisms to the ssl-based session hijacking attacks. *Netw. Protoc. Algorithms* 10, 1 (2018), 83–108.
- [31] IoTNow. Why 98% of iot traffic is unencrypted. <https://www.iot-now.com/2020/06/04/103245-why-98-of-iot-traffic-is-unencrypted/>.
- [32] ISO/IEC. Identification cards - integrated circuit cards - part 5: Registration of application providers. *International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 7816-5* (2004).
- [33] ISO/IEC. Identification cards—integrated circuit cards—part 4: Organization, security and commands for interchange. *International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 7816-4* (2013).
- [34] ITOH, K., IZU, T., AND TAKENAKA, M. Address-bit differential power analysis of cryptographic schemes ok-ecdh and ok-ecdsa. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2002), Springer, pp. 129–143.
- [35] JAIN, A. K., AND GUPTA, B. B. Phishing detection: analysis of visual similarity based approaches. *Security and Communication Networks* 2017 (2017).
- [36] Java card platform, classic edition 3.0.5. <https://docs.oracle.com/javacard/3.0.5/>.
- [37] Knowroaming travel sim sticker. <https://nakedsecurity.sophos.com/2019/02/19/thousands-of-android-apps-bypass-advertising-id-to-track-users/>.
- [38] KOHLS, K., RUPPRECHT, D., HOLZ, T., AND PÖPPER, C. Lost traffic encryption: fingerprinting LTE/4G traffic on layer two. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks* (2019), pp. 249–260.
- [39] LI, C.-Y., TU, G.-H., PENG, C., YUAN, Z., LI, Y., LU, S., AND WANG, X. Insecurity of voice solution volte in lte mobile networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 316–327.
- [40] LI, L., LI, D., BISSYANDÉ, T. F., KLEIN, J., LE TRAON, Y., LO, D., AND CAVALLARO, L. Understanding android app piggybacking: A systematic study of malicious code grafting. *IEEE Transactions on Information Forensics and Security* 12, 6 (2017), 1269–1284.
- [41] LI, Y., PENG, C., YUAN, Z., LI, J., DENG, H., AND WANG, T. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (2016), pp. 202–215.
- [42] LIU, J., YU, Y., STANDAERT, F.-X., GUO, Z., GU, D., SUN, W., GE, Y., AND XIE, X. Small tweaks do not help: Differential power analysis of milenage implementations in 3g/4g usim cards. In *European Symposium on Research in Computer Security* (2015), Springer, pp. 468–480.
- [43] LU, Y.-H., LI, C.-Y., LI, Y.-Y., HSIAO, S. H.-Y., XIE, T., TU, G.-H., AND CHEN, W.-X. Ghost calls from operational 4g call systems: Ims vulnerability, call dos attack, and countermeasure. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020), pp. 1–14.
- [44] M2m applets. <https://10t.mobi/resources/blog/iot-hacking-series-6-what-is-a-sim-applet-and-why-is-it-important-for-iot-m2m>.
- [45] MARKANTONAKIS, K. Is the performance of smart card cryptographic functions the real bottleneck? In *IFIP International Information Security Conference* (2001), Springer, pp. 77–91.
- [46] MEYER, M., QUAGLIA, E. A., AND SMYTH, B. Attacks against gsma’s m2m remote provisioning (short paper). In *International Conference on Financial Cryptography and Data Security* (2018), Springer, pp. 243–252.
- [47] NAUMAN, M., KHAN, S., AND ZHANG, X. Apex: extending android permission model and enforcement with user-defined runtime constraints. In *Proceedings of the 5th ACM symposium on information, computer and communications security* (2010), pp. 328–332.
- [48] PATIL, P., NARAYANKAR, P., NARAYAN, D., AND MEENA, S. M. A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish. *Procedia Computer Science* 78 (2016), 617–624.
- [49] Qualcomm modem id. <https://thingspace.verizon.com/documentation/apis/qualcomm-cat-m1-lte-modem-device-api.html>.
- [50] RAO, J. R., ROHATGI, P., SCHERZER, H., AND TINGUELY, S. Partitioning attacks: or how to rapidly clone some gsm cards. In *Proceedings 2002 IEEE Symposium on Security and Privacy* (2002), IEEE, pp. 31–41.
- [51] RAO, L. J., RAO, M. V., AND SARADHI, T. V. How the smrtcard makes the certification verification easy. *Journal of Theoretical and Applied Information Technology* 83, 2 (2016), 180.
- [52] REARDON, J., FEAL, Á., WIJESKERA, P., ON, A. E. B., VALLINA-RODRIGUEZ, N., AND EGELMAN, S. 50 ways to leak your data: An exploration of apps’ circumvention of the android permissions system. In *28th USENIX Security Symposium (USENIX Security 19)* (2019), pp. 603–620.
- [53] RUPPRECHT, D., KOHLS, K., HOLZ, T., AND PÖPPER, C. Breaking lte on layer two. In *2019 IEEE Symposium on Security and Privacy (SP)* (2019), IEEE, pp. 1121–1136.
- [54] RUPPRECHT, D., KOHLS, K., HOLZ, T., AND PÖPPER, C. Imp4gt: Impersonation attacks in 4g networks. In *Symposium on Network and Distributed System Security (NDSS). ISOC* (2020).
- [55] RUSSELLO, G., CONTI, M., CRISPO, B., AND FERNANDES, E. Moses: supporting operation modes on smartphones. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies* (2012), pp. 3–12.
- [56] SCHUSTER, R., SHMATIKOV, V., AND TROMER, E. Situational access control in the internet of things. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018), pp. 1056–1073.
- [57] SHAIK, A., BORGAONKAR, R., ASOKAN, N., NIEMI, V., AND SEIFERT, J.-P. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. *NDSS* (2015).
- [58] Rebelsim. <http://osmocom.org/projects/simtrace/wiki/RebelSIM>.
- [59] Simjacker technical paper. <https://simjacker.com/>.
- [60] Simtrace 2. <https://osmocom.org/projects/simtrace2/wiki>.

- [61] SINGH, J., RUHL, R., AND LINDSKOG, D. Gsm ota sim cloning attack and cloning resistance in eap-sim and usim. In *2013 International Conference on Social Computing* (2013), IEEE, pp. 1005–1010.
- [62] STROBEL, D. Imsi catcher. *Chair for Communication Security, Ruhr-Universität Bochum 14* (2007).
- [63] T-mobile pin setting. <https://www.t-mobile.com/support/devices/android/t-mobile-revvlry-plus/sim-pin-t-mobile-revvlry-plus>.
- [64] Uicc carrier privileges. <https://source.android.com/devices/tech/config/uicc>.
- [65] Verizon pin setting. <https://www.verizon.com/support/knowledge-base-206865/>.
- [66] XIE, T., TU, G.-H., LI, C.-Y., AND PENG, C. How can iot services pose new security threats in operational cellular networks? *IEEE Transactions on Mobile Computing* (2020).
- [67] YANG, H., BAE, S., SON, M., KIM, H., KIM, S. M., AND KIM, Y. Hiding in plain signal: Physical signal overshadowing attack on LTE. In *28th USENIX Security Symposium (USENIX Security 19)* (2019), pp. 55–72.
- [68] ZHANG, W., MENG, Y., LIU, Y., ZHANG, X., ZHANG, Y., AND ZHU, H. Homonit: Monitoring smart home apps from encrypted traffic. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018), pp. 1074–1088.