SoK: Extracting secrets from UICCs
How to clone SIM cards in 2025

Daniel Kipp
TU Wien

Abstract

In this paper, we provide a systematic overview of the known research on extracting secrets stored on UICCs, commonly known as "SIM-cards". We survey all mobile network generations from 2G to 5G, compare the work matching our criteria, and then present all known attack vectors. Some early algorithms used in 2G, like COMP128-1, are cryptographically broken, allowing for easy key recovery through purely cryptographic means. Modern algorithms like the AES-based MILENAGE and Keccak-based TUAK utilize state-of-the-art cryptography, shifting the threat landscape towards side-channel exploitation. With side-channels, a sufficiently skilled and equipped attacker can break state-of-the-art Commercial Off-The-Shelf UICCs, even when the UICC utilizes proprietary side-channel protection mechanisms. The most efficient attacks for modern UICCs seem to be Correlation Power Analysis and NDDLA. By consolidating the existing body of research, this study provides a comprehensive assessment of attack feasibility and highlights knowledge gaps, for example, EM side-channels.

1 Introduction

Over the past decades, the Universal Integrated Circuit Card (UICC) has emerged as the cornerstone of subscriber authentication and secure key storage in mobile networks. Commonly referred to as the SIM card, the UICC plays a critical role in safeguarding the cryptographic secrets that enable secure communication across generations of cellular technology, from the early Global System for Mobile Communications (GSM) to today's fifth-generation (5G) networks. Almost everybody using a mobile phone is using a UICC as well. Theft of these secrets would enable an attacker to duplicate the UICC, impersonate the subscriber, and eavesdrop on his traffic.

While the cryptographic primitives used in modern networks have evolved substantially—transitioning from legacy algorithms like COMP128 to more robust constructions based on Advanced Encryption Standard (AES) and Keccak—the underlying assumption of UICC security has remained largely unchanged: that the card's tamper-resistant hardware prevents adversaries from retrieving the keys it protects. However, a growing body of research over the last two decades has challenged this assumption. In particular, advances in side-channel analysis have demonstrated that even well-designed cryptographic algorithms can be undermined by physical leakages such as power consumption side-channels.

This paper systematically examines the state of the art in extracting secret material from UICCs. We survey the known attacks across all major mobile network generations and algorithm families, highlighting both cryptographic weaknesses (such as the now well-documented vulnerabilities in COMP128-1) and practical side-channel exploits that target otherwise secure implementations. From early partitioning attacks to recent deep learning–assisted power analysis capable of bypassing proprietary countermeasures, these methods underscore the persistent challenges in securing embedded cryptographic hardware.

By consolidating and comparing existing studies, we aim to provide a clear, structured perspective on the feasibility of recovering UICC secrets in contemporary contexts. In doing so, we illustrate how advances in attack methodologies, particularly Correlation Power Analysis (CPA) and Non-Profiled Differential Deep Learning Attack (NDDLA) have extended the practical reach of adversaries even against commercially deployed UICCs. We hope that this synthesis will not only inform researchers and practitioners about current capabilities but also motivate further work on designing and evaluating effective countermeasures.

2 Background

2.1 Cellular Networking

Cellular networking is a vast and complex topic, featuring an extensive amount of different technologies and protocols.

Commonly, it is categorized into different generations, each representing a significant advancement in mobile communi-

cation technology. The first generation (1G) was launched in the 1980s, used analog technology without encryption, and had voice-only communication with limited capacity and poor voice quality and security. The second generation (2G), introduced in the early 1990s, used digital technology and got improved voice quality, capacity, and security via encryption. It also introduced Short Message Service (SMS) and basic data services like General Packet Radio Service (GPRS). Around 2000, the third generation (3G) was rolled out, featuring significantly higher data rates and improved security. Then in the late 2000s, the fourth generation (4G) was rolled out with even higher data rates with peak speed requirements as high as 1 $\frac{\text{Gbit}}{\text{s}}$ for low-mobility users like pedestrians. It also was designed to support all-IP communications, eliminating circuit switching in voice telephony. The fifth and to this date most recent generation (5G) started being rolled out in the late 2010s. Compared to 4G, it offers even higher speed with 10 $\frac{\text{Gbit}}{\text{s}}$ peak performance and reduced latency [17]. Multiple standards exist for each generation. A timeline of mobile network generations and the associated standards can be found in Figure 1.

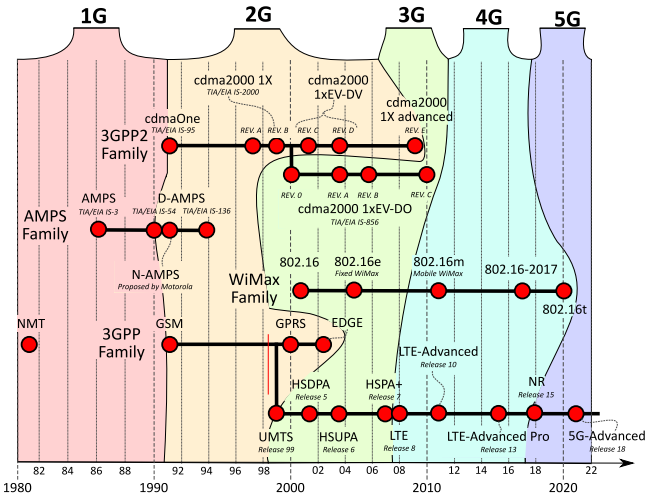

Figure 1: Cellular network standards and generation timeline[1]

The focus of this paper will be on the 3GPP family, including all the GSM-, UMTS-, LTE- and 5G NR-based standards since these are the most commonly used ones. All other standards like WiMax are considered out-of-scope. Since all standards of the GSM family (e.g. GPRS) build upon the GSM Authentication and Key Agreement (AKA) mechanism, we can focus solely on GSM in the 2G domain. The same holds true for UMTS in the 3G domain, LTE in the 4G domain, and 5G NR in the 5G domain. For these reasons, we will from now on use 2G as a synonym for GSM, 3G as a synonym for UMTS, 4G as a synonym for LTE, and 5G as a synonym for 5G NR.

[1] ©Michel Bakni, CC-BY-SA 4.0, unmodified, from Wikimedia Commons

2.2 The Universal Integrated Circuit Card (UICC)

Since mobile network security is based on symmetric cryptography, the operator and the subscriber need shared secrets which in turn need adequate protection. While on the operator side the key material is stored inside a secure database, on the subscriber side the UICC is responsible for the protection. A UICC is a smart card, featuring a processor, RAM, ROM, and NVRAM. It can host multiple applications like a Subscriber Identity Module (SIM) application for accessing 2G networks or a Universal Subscriber Identity Module (USIM) application for accessing 3G to 5G networks. Inside these applications, among other things like the International Mobile Subscriber Identity (IMSI), the cryptographic keys for subscriber authentication and session key derivation (the session keys are then used to encrypt the data actually transmitted over the air) are stored. The secrets are generated by the operator, stored in his database, and then burned into the card. The protection is accomplished by only allowing predefined cryptographic algorithms to be run (on user-supplied data) using the stored keys and providing no means to read the keys themselves. Since the secrets are stored in the microelectronic of the UICC and no reading interface is provided it is highly challenging to get ahold of them.

The confidentiality of the key material is crucial for mobile network security. Compromising it allows an attacker to effectively duplicate the card, impersonate the subscriber, or eavesdrop on the subscriber's traffic.

2.3 Authentication and Key Agreement (AKA)

In the following sections, the AKA mechanisms for the different technology generations are outlined. While conceptually similar, with each subsequent generation it becomes increasingly complex but also more secure. For the sake of brevity, we focus only on the interaction with the UICC. However, the AKA cheatsheets by Nakarmi [14] on which this section is partially based provide an excellent graphical overview of the full details of the different mechanisms in use.

2.3.1 2G

A simplified version of the authentication process, focusing on the interaction between the subscriber's device and the SIM is outlined in Figure 2. It starts with the SIM sending an Authentication Vector Request with its IMSI to the Authentication Center (AuC). Based on the IMSI, the operator looks up the authentication key K, generates a random number $RAND$, and computes an expected response $XRES$ based on these two values using the A3 algorithm. The $RAND$ value is then forwarded to the SIM, which computes its RES value individually with the same A3 algorithm using the same K authentication key it has stored. This value is then sent to the operator and if it matches the one from the AuC, the

subscriber is considered authenticated. If the authentication was successful, the server and the client proceed to generate a session key CK which is then used to encrypt further communication [7].

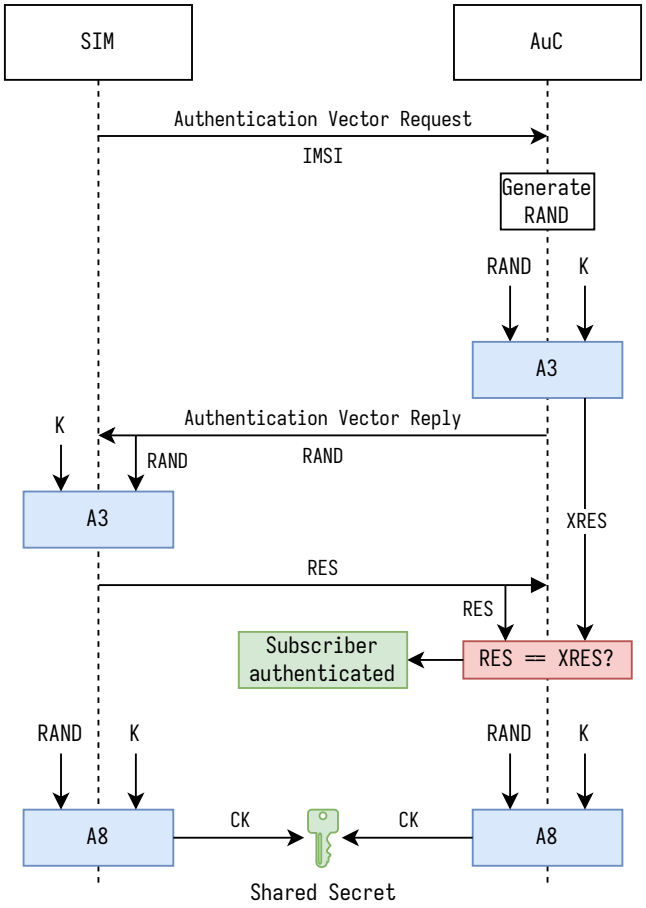

Figure 2: Simplified version of the 2G authentication process

Note that the operator is free to choose the concrete implementation of the A3 and A8 algorithms. However, the secret COMP128-1 cipher was widely used in the past. But in 1998 it was reverse-engineered and a cryptoanalysis was performed, which uncovered major vulnerabilities [19]. This led to the development of COMP128-2 and COMP128-3 as secure replacement. The two ciphers are virtually identical, except for the fact that COMP128-2 sets the 10 rightmost bits of the generated shared key to zero [6]. Initially also kept secret, they were reverse-engineered in 2013, but to the best of our knowledge, no cryptoanalysis of them has been published to this date. Nowadays the GSM Association recommends the AES-based GSM-MILENAGE algorithm set (although COMP128-3 is also rated "acceptable") [6] and most operators will probably follow this recommendation.

2.3.2 3G

Liu et al. [12] cover the 3G authentication mechanism very well. Unlike GSM where only the network authenticates the client and not the other way around, UMTS enforces a mutual AKA protocol, which builds upon a set of cryptographic functions $f_1, ..., f_5$. The authentication again starts with an Authentication Vector Request containing the subscriber's IMSI [7]. The AuC then samples a random number $RAND$, assigns a sequence number SQN, and computes f_1 with the symmetric key K, SQN, $RAND$ and the AMF (Authenticated and key Management Field) constant to produce the MAC. Then f_5 is computed with $RAND$ and K, yielding the AK (Anonymity Key). Then the SQN is XORed with AK (if SQN was sent plain an attacker could intercept and manually increment it, enabling replay attacks) and together with AMF and MAC send as $AUTN$ (AUthentication TokeN) together with $RAND$ to the USIM. The USIM then computes the AK itself, employing the same f_5 function as the operator. With the AK the USIM then unmasks the SQN by computing $(SQN \oplus AK) \oplus AK$ and continues to use it together with AMF, $RAND$ and K to compute the expected MAC $XMAC$. Then it is checked whether SQN is outside the expected range (the USIM compares it to an internal counter to prevent replay attacks) and if $XMAC == MAC$ (to authenticate the base station). If one of the checks fail, the USIM aborts the authentication. Otherwise, it uses $RAND$ and K to compute the $XRES$ value with f_2, which is then sent to the AuC. The AuC has performed the same calculation and compares the two values. If they match, the client is considered authenticated and both proceed to calculate the cipher key CK and integrity key IK using functions f_3 and f_4 respectively, taking $RAND$ and K as arguments [12]. The process is somewhat simplified and again focuses on the interaction with the USIM. A graphical representation of this flow can be found in Figure 3.

Like with 2G, the operator is free to choose the concrete algorithms for the $f_1, ..., f_5$ functions in his home network and USIMs. However, the GSM Association recommends that operators implement both the MILENAGE algorithm set based on AES and the TUAK algorithm set based on the Keccak permutation (which also serves as the basis of the SHA-3 cryptographic hash function) [6]. The MILENAGE algorithm, besides K, takes some operator defined parameters $r_1, ..., r_5$, OP_C and $c_1, ..., c_5$. With the exception of OP_C, all parameters have default values suggested in the 3GPP specification, although operators can configure them to secret values [12]. TUAK on the other hand has only one operator-defined parameter named TOP_C, which serves the same role as OP_C in MILENAGE. Currently, MILENAGE seems to be the most commonly used algorithm set.

2.3.3 4G

4G defines three different authentication schemas: AKA, EAP-AKA, and EAP-AKA'. They are used for authentication

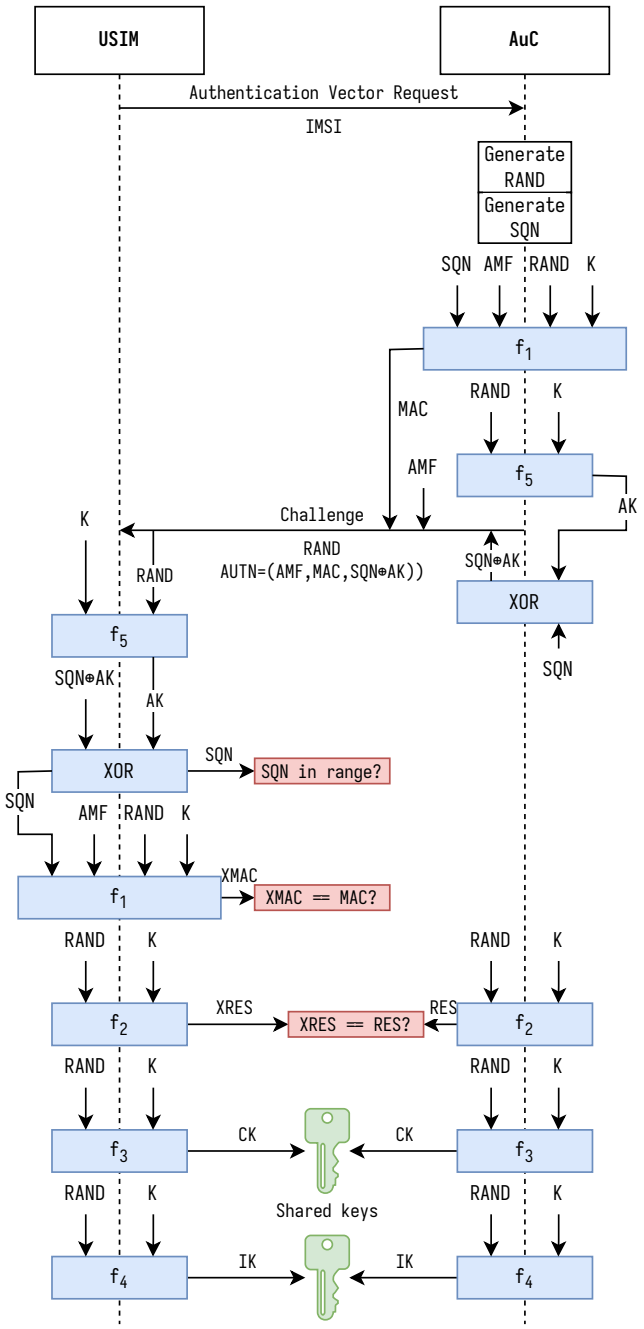

Figure 3: Simplified version of the 3G authentication process

with 3rd Generation Partnership Project (3GPP) networks, untrusted Non-3GPP networks, and trusted Non-3GPP networks respectively. However, concerning I/O and calculations performed by the UICC, the three schemas have negligible differences and are also almost identical to 3G. The main differences lie after the AKA has taken place and the *CK* encryption key and *IK* integrity key have already been passed from the UICC to the Mobile Equipment (ME). [14] Thus, the process for extracting secrets from 4G UICCs should be the same as for 3G UICCs.

2.3.4 5G

5G defines three different authentication schemas: AKA, EAP-AKA' and EAP-TLS with the latter being only dedicated to "limited use cases such as private networks and IoT environments" [4], so we can ignore it. The other two, similar to 4G-AKA, feature no differences in interactions with and computations performed by the UICC, so again the extraction process should be the same as for 3G UICCs.

2.3.5 Similarities

Since the interactions with the UICC are virtually the same for the 3G, 4G, and 5G AKA mechanisms, we will group them and refer to them as 3G+. We now can focus on just two authentication processes, the 2G and 3G+ ones.

2.4 Side-Channel Attacks (SCAs)

A side-channel attack, in the broad sense, is "an attack enabled by leakage of information from a physical cryptosystem" [15]. Exploitable characteristics include timing (how long it takes to perform a cryptographic operation), power consumption, electromagnetic emissions, and even acoustic emissions. In this section, we will concentrate on power side-channels, as they are the primary side-channel employed for UICC secret recovery. They exploit the fact that changes in voltage (flipping bits) require small movements of electric charges which consume power and produce electromagnetic radiation.

2.4.1 Simple Power Analysis (SPA)

SPA involves directly measuring and reading the power consumption of a device. It is good at revealing the sequence of instructions executed and thus works very well for breaking implementation on which the execution path depends on secret data. A good example for this is the DES key schedule, which involves rotating key registers and where a conditional branch is used to check if the bit shifted off the edge was a 1 so that it can be wrapped around. Thus the power consumption traces are easily distinguishable for 1 and 0 bits [11]. It is rather limited, however, and it is not always possible to reconstruct information with SPA, because the execution path does not always depend on sensitive values.

2.4.2 Differential Power Analysis (DPA)

DPA exploits the fact that the power consumption is not only dependent on the execution sequence but also on the exact values manipulated. These variations are usually very small and can easily be overshadowed by noise and measurement errors. But by statistically analyzing power consumption over

multiple rounds of execution it is often still possible to recover information.

Correlation Power Analysis (CPA) CPA is a variation of DPA. It works by observing n power traces of parts of a cryptographic operation involving the secret to extract. Then the secret is split into parts (commonly byte-sized ones) called subkeys and the expected power consumption of the device is modeled based on the input and a guess of the subkey. For every possible guess it is then calculated how well the estimated power traces correlate with the actually observed ones. Usually, all the data (including all n power traces) is combined into one correlation coefficient for every subkey guess. The guess with the highest coefficient is the most likely to be correct. The correlation coefficient of the correct guess should be clearly visible as a peak when plotting all correlation coefficients. For UICCs, popular choices for the power-consumption model and correlation coefficient are the Hamming Weight model (the Hamming Weight of a binary number is the amount of 1 digits) and Pearson's Correlation Coefficient respectively.

As an example, consider attacking AES-128 with CPA. Let's assume we measure T power traces consisting of D data points where $d_{t,j}$ is the j-th data point ($1 \leq j \leq D$) in power trace t ($1 \leq t \leq T$). We split the 128-bit key into 16 different subkeys each 8 bit long, meaning we have 256 guesses for each individual subkey. The power estimate in trace t of an individual guess i ($0 \leq i \leq 255$) is denoted by $g_{t,i}$. We model the power consumption with the Hamming Weight model and calculate the correlation with Pearson's Correlation Coefficient. One way to calculate the correlation coefficient for each guess i and time j is

$$r_{i,j} = \frac{\sum_{t=1}^{T}[(g_{t,i} - \overline{g_i}) \cdot (d_{t,i} - \overline{d_j})]}{\sqrt{((\sum_{t=1}^{T} g_{t,i})^2 - T\sum_{t=1}^{T} g_{t,i}^2)((\sum_{t=1}^{T} d_{t,j})^2 - T\sum_{t=1}^{T} d_{t,j}^2)}}$$

where $\overline{g_i}$ is the mean power consumption over all traces of a guess i and $\overline{d_j}$ is the mean measured power consumption over all traces of a time j. The correlation coefficient for each individual guess r_i is then calculated with $r_i = max(|r_{i,j}|)$, selecting the maximum value over all time indexes j. The guess i with the highest coefficient r_i is then the most likely candidate for the subkey [8].

High-order DPA / CPA Generally speaking high-order DPA is an advanced form of DPA where multiple data sources are combined in a single trace.

As an example, we are going to look at high-order CPA used to break AES with a specific side-channel countermeasure known as masking, which it can defeat. The basic idea behind masking is to split up the cryptographic operation (including the plaintext and key material) into d independent parts known as shares and run them independently. This theoretically eliminates the correlation between the key and the power measurements if the attacker uses fewer than d signals simultaneously. So the attacker needs to combine d samples simultaneously with a combination function like *normalized product combining* before analyzing the traces. This is then called d-th order CPA [9].

Non-Profiled Differential Deep Learning Attack (NDDLA) NDDLA is a type of partition-based non-profiled DPA attack that utilizes deep learning training to verify the key guesses.

A partition-based attack builds upon the attacker guessing a part of the key (subkey). He then partitions the set of traces according to hypothetical intermediate values based on the guess and then uses a statistical distinguisher (e.g. difference of means) to measure the consistency of each partition. For the correct guess, the partitioning should be consistent (high difference of means), whereas for incorrect guesses, the partitioning is basically random, and one should observe no consistency (difference of means close to 0).

Now with NDDLA, the idea is that the attacker trains a deep neuronal network with the traces as training data and the partitions as classification labels. Only for the correct guess the partition and labels used for the training will be consistent with the corresponding trace, leading to more efficient training. So the attacker can select the correct guess as the one with the best training metrics [18].

NDDLA performs equally well as a high-order CPA attack and is also able to break masking countermeasures [9].

3 Methods

3.1 Collection

We employed a somewhat unstructured and flexible approach to collect papers on this topic. We utilized various generic and specialized search utilities like Google[2], Google Scholar[3], the ACM Digital Library[4], arXiv[5] or Elicit[6] and varying (English only) search queries to collect all papers matching our criteria about extracting secrets from UICCs. Our indicator for when to stop searching for papers was when we felt like we could not find any more work on the topic. Since the amount of matching papers turned out to be rather limited, we decided against discarding a fixed portion of the papers which is sometimes done in SoK papers to select only the highest quality works.

3.2 Criteria

All papers were evaluated if they meet the following criteria and discarded if not:

[2] https://www.google.com/
[3] https://scholar.google.com/
[4] https://dl.acm.org/
[5] https://arxiv.org/
[6] https://elicit.com/

- **Attack Feasibility**: The paper must present concrete and practically feasible attacks.
- **Attack Focus**: The paper must attack an UICC and extract secret values from the SIM or USIM applications. Attacks against other types of smart cards are out of scope.
- **Attack Target**: The paper must target the secret values of the UICC. Attacks against just the wireless communication (e.g. the cipher key *CK*) are out of scope.
- **Attack Success**: At least one of the cryptographic secrets stored on the UICC must get compromised. Papers about failed attempts are out of scope.
- **Technology**: The attack must be mounted against 2G or 3G+. Attacks against other standards like WiMax are out of scope.
- **Algorithms**: The attack must be mounted against a known and commonly used algorithm set like COMP128 (any version), MILENAGE, or TUAK. SIM or USIM applications employing unknown or esoteric algorithms are out of scope.
- **Writing Format**: We only include scientific papers that adhere to basic quality standards, such as the correct use of the English language. Papers with poor English are deemed out of scope, and we focus solely on scientific works; thus, otherwise valuable resources like blog posts are also excluded.

3.3 Processing

With all the papers collected, we compiled a comprehensive overview of them (see Table 1). We then proceeded to compare the papers and developed a summary of all the possible attacks on UICCs.

4 Results

4.1 Overview

We found eight studies matching our criteria, which are presented in Table 1. Almost all of them use side-channel attacks, only one paper presents a purely cryptographic attack. The reason for this could be that besides the completely broken COMP128-1 algorithm, all other popular algorithms are based on solid cryptographic principles and side-channels are the only meaningful way to extract information from UICCs utilizing these algorithms. Note that we could not include the original cryptographic attack on the COMP128-1 cipher from Wagner, Goldberg, and Briceno [19] because they never published a formal paper about their attack but just an informal entry on the website of their research group.

Three of the eight studies focus on 2G technology, while the rest focus on 3G+ technology. All 2G papers focus on the broken COMP128-1 cipher and to the best of our knowledge there is no work on 2G SIMs using the newer COMP128-2/3 or GSM-MILENAGE algorithms. However, it should be possible to apply the side-channel attacks from Zhou et al. [21], Liu et al. [12], or Jin et al. [9] to those algorithms, although further work is needed to confirm this. Apart from that the three studies are quite distinct. Rao et al. [16] invent the "partitioning attack", a new way to exploit side-channels, and utilize it with a power side-channel to break a COMP128-1 implementation with some unknown side-channel protections in place. Wray [20] improves the original cryptographic attack from Wagner, Goldberg, and Briceno [19], reducing the expected number of challenges before success from 150 000 to 60 000 and making attacks against some "strong" keys for which the original approach would not work possible. Zhou et al. [21] use plain DPA to recover the key from UICCs with unknown protections against the partitioning attack and the original cryptographic attack in place.

With one exception focusing on TUAK, all the 3G+ papers use the MILENAGE algorithm set. The reason for this is probably that MILENAGE is more commonly used. Also in the paper using TUAK the researchers implemented the algorithms themselves on a UICC with a "32-bit processor core running at up to 25 MHz" [13]. To the best of our knowledge no research on breaking TUAK in a real product has been published so far, probably because, again, MILENAGE seems to be more common.

The works from Devine, San Pedro, and Thillard [5] and Brisfors, Forsmark, and Dubrova [2] are heavily based on Liu et al. [12]. While Devine, San Pedro, and Thillard [5] are simply reproducing their results, Brisfors, Forsmark, and Dubrova [2] extend them by training a Convolutional Neural Network (CNN) on the acquired power traces and then use the CNN to enable very efficient and easy attacks on similar UICCs. Note that for the training they need to know the secret values, which are extracted using the method from Liu et al. [12]. One notable paper is the one from Jin et al. [9]. They use high-order CPA and NDDLA to successfully attack Commercial Off-The-Shelf (COTS) UICCs employing proprietary side-channel countermeasures that were not known beforehand to the researchers.

Interestingly, all 3G+ papers use some form of CPA to recover the secrets, suggesting that it is a very effective attack method. However, as shown by Jin et al. [9] first-order CPA seems to be ineffective against targets with side-channel countermeasures. High-order CPA or NDDLA is able to bypass these measures nevertheless.

Another interesting fact is that all papers that utilize side-channels are using power side-channels. To the best of our knowledge, there is no paper using alternative side-channels to extract UICC secrets.

Paper	Year	Target	Method	Result	Limitations	Notes
Rao et al. [16]	2002	2G SIM COMP128-1	Partitioning attack with power side-channel	Recovery of K		They invent a new way to exploit side-channels called partitioning attack and use it to break a COMP128-1 implementation with unknown side-channel protections
Wray [20]	2003	COMP128-1	Cryptographic	Recovery of K	Works only with COMP128-1	Purely theoretical work, improves the original cryptographic attack on the COMP128-1 cipher of Wagner, Goldberg and Briceno [19]
Zhou et al. [21]	2013	2G SIM COMP128-1	DPA	Recovery of K		They use plain DPA to extract the keys from UICCs employing unknown protections against the partitioning attack of Rao et al. [16] and the original attack of Wagner, Goldberg and Briceno [19]
Liu et al. [12]	2015	3G+ USIM MILENAGE	CPA	Recovery of K, OP_C, $r_1,...,r_5$ and $c_1,...,c_5$	Unprotected MILENAGE implementation	They perform CPA even though they say DPA
Maghrebi and Bringer [13]	2017	TUAK	CPA	Recovery of K, TOP_C	Unprotected TUAK implementation Custom TUAK implementation loaded onto an UICC	They do not break a real product but demonstrate that attacks against unprotected TUAK implementation are in principle possible
Devine, San Pedro and Thillard [5]	2018	3G+ USIM MILENAGE	CPA	Recovery of K, OP_C	Unprotected MILENAGE implementation Default values for $r_1,...,r_5$ and $c_1,...,c_5$ are assumed	Reproduction of the work from Liu et al. [12]
Brisfors, Forsmark and Dubrova [2]	2021	3G+ USIM MILENAGE	CPA	Recovery of K, OP_C	Unprotected MILENAGE implementation Default values for $r_1,...,r_5$ and $c_1,...,c_5$ are assumed	The papers focuses on training a CNN and using it to recover secrets of similar UICCs with knowledge of K and OP_C as prerequisite. They recover them with the method from Liu et al. [12]
Jin et al. [9]	2021	3G+ USIM MILENAGE	CPA and NDDLA	Recovery of K, OP_C, $r_1,...,r_5$ and $c_1,...,c_5$		Targets COTS UICCs with unknown proprietary side-channel countermeasures

Table 1: Overview of the found studies

4.2 Attacks

4.2.1 Cryptographic attacks

As demonstrated by Wagner, Goldberg, and Briceno [19] and Wray [20] the COMP128-1 cipher, historically used for 2G AKA, suffers from severe vulnerabilities, allowing to reconstruct the secret key K with about 60 000 challenge-response pairs in the common case or 675 000 challenge-response pairs in the case of a "strong" key. Approximately 2^{76} out of 2^{128}, or 22.20×10^{-15} %, of the possible keys, are such "strong"

keys.

Practically the attack has been implemented by Kaljevic in his SimScan application [10], which utilizes independent optimizations and appears to be capable of extracting the full key with about 20 000 queries to the UICC [3]. We do not know if the application is able to recover "strong" keys.

4.2.2 Side-Channel Attacks

Side-channels provide an effective way of extracting secrets from the UICC if no protections against them are deployed. In addition to cryptographic attacks, the COMP128-1 algorithm is also vulnerable to multiple types of side-channel attacks like DPA [21] or the partitioning attack from Rao et al. [16]. It is unknown whether the newer COMP128-2/3 and GSM-MILENAGE algorithms are vulnerable to side-channel attacks. It appears quite likely that they are, but further research is needed to confirm this. Unprotected implementations of MILENAGE [12] and TUAK [13] seem to be vulnerable to first-order CPA. Even protected implementations of MILENAGE are vulnerable to high-order CPA or NDDLA attacks [9]. It is currently unknown whether the same is possible for protected implementations of TUAK.

5 Conclusions

Overall, the UICC storage seems to be quite secure. The secrets being stored within the microelectronic of the chip without a direct reading interface makes them very hard to obtain. However, there are still some methods that allow a properly skilled and equipped attacker to steal them.

Except for cryptographic weaknesses in the COMP128-1 algorithm, no other direct (as in, not involving side-channels) vulnerabilities have been found in the implementation of modern UICCs utilizing state-of-the-art algorithms. So currently, side-channel attacks appear to be the most promising way to accomplish secret recovery, with the most notable methods being first-order CPA for unprotected implementations and high-order CPA or NDDLA for ones with unknown proprietary side-channel countermeasures. This is a considerable risk since extracting the secrets allows for illegal cloning of the card and thus impersonating the mobile subscriber or eavesdropping on his traffic. Further research is necessary to learn how to provide more effective mitigations against side-channel attacks.

Also besides the power side-channel used in all studies, there are other types of side-channels, particularly EM side-channels, which utilize the emitted electromagnetic radiation of the chip. They do not appear to have been studied in the context of UICC secret extraction but have been successfully applied against other smart cards [1]. Further research is necessary to gain insight into how alternative side-channels might lead to a bypass of power analysis protections.

Other things which yet need to be investigated are how secure the COMP128-2/3 and GSM-MILENAGE algorithms are against side-channel attacks and the cryptographic security of the COMP128-2/3 algorithms. To the best of our knowledge, no work on this has been published so far.

The security of the TUAK algorithm in real implementation also remains uncertain, as to the best of our knowledge, it has only been studied using a self-written implementation loaded onto a not further specified UICC [13]. If the approach of Maghrebi and Bringer [13] works on real products and if, like with MILENAGE, side-channel measures can be broken with the right techniques still requires investigation.

Another rather minor blind spot is the security of 5G UICCs, the 3G+ papers all use either 3G or 4G cards. However, since 5G in theory features the exact same interactions with the UICC in its AKA process as 4G and 3G, there should be no difference.

References

[1] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The em side—channel(s). In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[2] Martin Brisfors, Sebastian Forsmark, and Elena Dubrova. How deep learning helps compromising usim. In Pierre-Yvan Liardet and Nele Mentens, editors, *Smart Card Research and Advanced Applications*, pages 135–150, Cham, 2021. Springer International Publishing.

[3] Billy Brumley. A3/a8 & comp128. http://www.tcs.hut.fi/Studies/T-79.514/slides/S5.Brumley-comp128.pdf, November 2004.

[4] CableLabs. A comparative introduction to 4g and 5g authentication. https://go.cablelabs.com/hubfs/InformED%20Insights/A%20Comparative%20Introduction%20of%204G%20and%205G%20Authentication.pdf. Accessed: 2025-06-02.

[5] Christophe Devine, Manuel San Pedro, and Adrian Thillard. A practical guide to differential power analysis of usim cards. In *SSTIC 2018*, Rennes, June 2018.

[6] GSMA FASG. Security algorithm deployment guidance. Technical Report FS.35, GSM Association, April 2022. https://www.gsma.com/solutions-and-impact/technologies/security/wp-content/uploads/2022/09/FS.35-v3.0.pdf.

[7] Jérôme Härri and Christian Bonnet. *Security in Mobile Telecommunication Networks*, chapter 9, pages 315–360. John Wiley & Sons, Ltd, 2009.

[8] NewAE Technology Inc. Correlation power analysis - chipwisperer wiki. http://wiki.newae.com/Correlation_Power_Analysis. Accessed: 2025-07-11.

[9] Chengbin Jin, Yongbin Zhou, Xinkuan Qiu, Qi Feng, and Qian Zhang. Breaking real-world cots usim cards with unknown side-channel countermeasures. *Computers & Security*, 113:102531, 2022.

[10] Dejan Kaljevic. Simscan v2.00. https://web.archive.org/web/20100313050316/http://users.net.yu/~dejan/download/sim_scan.zip. Accessed: 2025-07-07.

[11] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[12] Junrong Liu, Yu Yu, François-Xavier Standaert, Zheng Guo, Dawu Gu, Wei Sun, Yijie Ge, and Xinjun Xie. Small tweaks do not help: Differential power analysis of milenage implementations in 3g/4g usim cards. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, pages 468–480, Cham, 2015. Springer International Publishing.

[13] Houssem Maghrebi and Julien Bringer. Side-channel analysis of the tuak algorithm used for authentication and key agreement in 3g/4g networks. In Kerstin Lemke-Rust and Michael Tunstall, editors, *Smart Card Research and Advanced Applications*, pages 39–56, Cham, 2017. Springer International Publishing.

[14] Prajwol Kumar Nakarmi. Cheatsheets for authentication and key agreements in 2g, 3g, 4g, and 5g, 2021.

[15] National Institute of Standards and Technology (NIST). Side-channel attack - glossary | csrc. https://csrc.nist.gov/glossary/term/side_channel_attack. Accessed: 2025-07-11.

[16] J.R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely. Partitioning attacks: or how to rapidly clone some gsm cards. In *Proceedings 2002 IEEE Symposium on Security and Privacy*, pages 31–41, May 2002.

[17] Ahmed Amin Ahmed Solyman and Khalid Yahya. Evolution of wireless communication networks: from 1g to 6g and future perspective. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(4):3943–3950, 2022.

[18] Benjamin Timon. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2):107–131, Feb. 2019.

[19] David Wagner, Ian Goldberg, and Marc Briceno. Gsm cloning. http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html, April 1998. Accessed: 2025-07-06.

[20] Stuart Wray. Comp128: A birthday surprise. https://www.stuartwray.net/comp128-a-birthday-surprise-rev.pdf, May 2003.

[21] Yuanyuan Zhou, Yu Yu, François-Xavier Standaert, and Jean-Jacques Quisquater. On the need of physical security for small embedded devices: A case study with comp128-1 implementations in sim cards. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 230–238, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

A Acronyms

3GPP 3rd Generation Partnership Project

5G NR 5G New Radio

AES Advanced Encryption Standard

AKA Authentication and Key Agreement

AK Anonymity Key

AMF Authenticated and key Management Field

AUTN AUthentication TokeN

AuC Authentication Center

CNN Convolutional Neural Network

COTS Commercial Off-The-Shelf

CPA Correlation Power Analysis

DES Data Encryption Standard

DPA Differential Power Analysis

GPRS General Packet Radio Service

GSM Global System for Mobile Communications

IMSI International Mobile Subscriber Identity

LTE Long-Term Evolution

ME Mobile Equipment

NDDLA Non-Profiled Differential Deep Learning Attack

NVRAM Non-Volatile Random Access Memory

RAM Random Access Memory

ROM Read-Only Memory

SCA Side-Channel Attack

SIM Subscriber Identity Module

SMS Short Message Service

SPA Simple Power Analysis

UICC Universal Integrated Circuit Card

UMTS Universal Mobile Telecommunications System

USIM Universal Subscriber Identity Module