# Small Tweaks do Not Help: Differential Power Analysis of MILENAGE Implementations in 3G/4G USIM Cards

Junrong Liu[1], Yu Yu[1,2,3], François-Xavier Standaert[4], Zheng Guo[1,5], Dawu Gu[1], Wei Sun[1], Yijie Ge[1], and Xinjun Xie[6]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
Email: {liujr,yyuu,guozheng,dwgu,ruudvn}@sjtu.edu.cn
[2] State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093)
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China
[4] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
Email: fstandae@uclouvain.be
[5] Shanghai Viewsource Information Science & Technology Co., Ltd
[6] Shanghai Modern General Recognition Technology Corporation

**Abstract.** Side-channel attacks are an increasingly important concern for the security of cryptographic embedded devices, such as the SIM cards used in mobile phones. Previous works have exhibited such attacks against implementations of the 2G GSM algorithms (COMP-128, A5). In this paper, we show that they remain an important issue for USIM cards implementing the AES-based MILENAGE algorithm used in 3G/4G communications. In particular, we analyze instances of cards from a variety of operators and manufacturers, and describe successful Differential Power Analysis attacks that recover encryption keys and other secrets (needed to clone the USIM cards) within a few minutes. Further, we discuss the impact of the operator-defined secret parameters in MILENAGE on the difficulty to perform Differential Power Analysis, and show that they do not improve implementation security. Our results back up the observation that physical security issues raise long-term challenges that should be solved early in the development of cryptographic implementations, with adequate countermeasures.

## 1 Introduction

The mathematical and physical security of cryptographic algorithms used in cellular networks has been a long standing concern. Starting with the reverse engineering of the COMP-128 algorithm (i.e. the A3/A8 algorithms used to authenticate GSM subscribers and generate session keys), Briceno, Goldberg and Wagner first showed that its compression function was fatally flawed due to a lack of diffusion. The resulting "narrow pipe attack" takes roughly 131,000 challenge-response pairs to recover a GSM SIM card master key [10]. Furthermore, several cryptanalytic results have been published about the A5 algorithm – i.e. the stream cipher used to encrypt the GSM communications based on a session key (see, e.g. [6,7,8,9,15]). Besides, different implementations of COMP-128

deployed in actual SIM cards have also been proved susceptible to Differential Power Analysis (DPA). For example, it was shown in [18] that a specialized (so-called partitioning) side-channel attack could lead to the cloning of 8-bit GSM SIM cards after monitoring its power consumption for only a couple of minutes. More recently, Zhou et al. reached a similar conclusion for implementations in 16-bit CPUs [20]. The latter reference also discussed the negative impact of closed-source algorithms (such as COMP-128) on physical security, as it limits the amount of research on dedicated countermeasures against physical attacks for these algorithms. As a result of this state-of-the-art, the move towards UMTS/LTE and the 3G/4G communication technology, whose security is based on standardized algorithms, was a very welcome improvement.

In this paper, we pay attention to the implementation of the MILENAGE algorithm in 3G/4G USIM cards, for which the recommended underlying primitive is the Advanced Encryption Standard (AES) Rijndael. MILENAGE is typically used for authentication and key agreement in UMTS/ LTE networks. As for previous works on side-channel analysis against SIM cards, this focus is motivated by the fact that breaking this part of the system is most damaging, since it allows eavesdropping, card cloning, and therefore bypassing the one-time-password authentication mechanism with mobile phones. In this context, we evaluated the security of eight commercial USIM cards, coming from a variety of operators and manufacturers, in order to tackle two main questions.

First, are the AES implementations used by MILENAGE systematically protected by state-of-the-art countermeasures against side-channel attacks? We answer this question negatively, as the different cards against which we performed experiments did not exhibit any particular mechanisms to prevent such attacks, leading to the same conclusions as [20] regarding the need to consider physical security issues early in the development of cryptographic products.

Second and more importantly, we analyzed the impact of small tweaks in MILENAGE – such as the use of secret (operator-defined) constants – regarding the difficulty of performing the attacks. As a main contribution, we show that these secrets have very limited impact on the attacks complexity. In particular, they do not bring the security improvements that would be expected from unknown-plaintexts, and allow successful divide-and-conquer key recoveries after a few minutes of power consumption measurements, as standard unprotected implementations of the AES in similar devices. The latter result is of more general interest, since it applies to any implementation of MILENAGE.

**Cautionary note.** The experiments presented in this paper were performed more than one year before submission to ESORICS 2015. We contacted the operators with feedbacks and suggestions (on countermeasures against side-channel analysis) before publication of the results. Upgrades towards more physically secure implementations are under development (or maybe already deployed). We do not claim that the USIM cards we measured and analyzed are reflective of the majority of deployed USIM cards and the paper does not contain specific details allowing to reveal the operators and manufacturers that we considered.

## 2 Background

### 2.1 The UMTS/LTE infrastructure

The Universal Mobile Telecommunications System (UMTS) and Long-Term Evolution (LTE) are respectively third generation (3G) and fourth generation (4G) mobile cellular systems for networks based on the Global System for Mobile Communication (GSM) standard. The technologies have been developed and maintained by the 3rd Generation Partnership Project (3GPP), and they have been widely adopted in many countries in Asia, Europe and the USA (see [3,4] for a list of mobile operators who adopt the 3G/4G technologies). For convenience, we only provide a simplified overview of the infrastructure by considering only two parties (omitting intermediate nodes such as Visitor Location Registers), namely, the Universal Subscriber Identity Module (USIM), which is typically a smart card embedded in a subscriber's telephony device, and an Authentication Center (AuC), which is a security function running on the operator's server. The cryptographic protocol engaged between two parties is symmetric, so that USIM and AuC need to share necessary information such as a unique identifier IMSI (International Mobile Subscriber Identity), a symmetric master key K, and operator-defined secrets OPc (operand code), $r1, \ldots, r5$, $c1, \ldots, c5$.

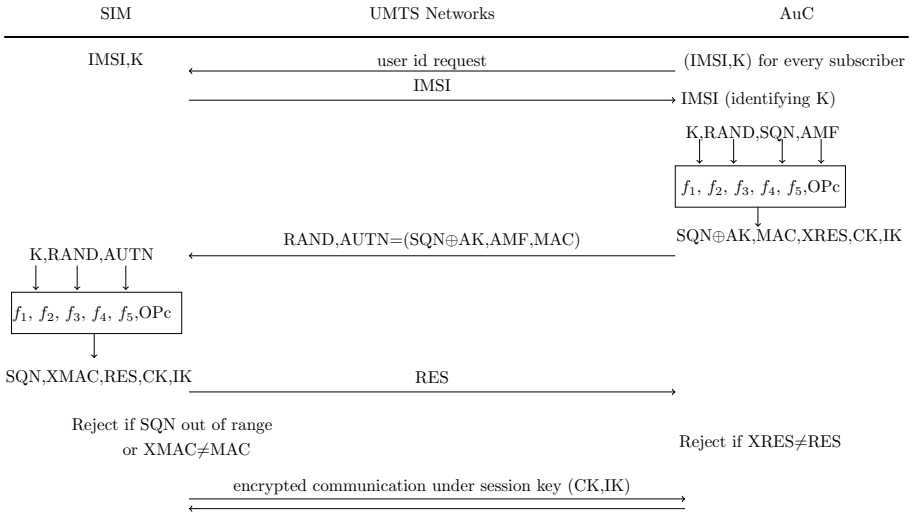


**Fig. 1.** Simplified AKA protocol between a USIM card and an AuC in 3G networks.

3G/4G AUTHENTICATION AND KEY AGREEMENT. Unlike GSM whose authentication was one-way and based on flawed algorithms, UMTS and LTE enforce a mutual authentication and key agreement (AKA) protocol, which in turn builds upon an AES-based algorithm called MILENAGE. As shown in Figure 1, the 3G authentication starts with a user id request and a response from USIM with

its unique IMSI. Upon the authentication request, the AuC samples a random RAND, assigns a sequence number SQN, and computes the MILENAGE algorithm (a suite of AES-based functions $f1$, ..., $f5$) with the symmetric key K and the AMF (Authenticated and key Management Field) constant to produce as output the masked (i.e. XORed with anonymity key AK) sequence number SQN⊕AK, tag MAC, expected response XRES, cipher key CK and integrity key IK. The USIM then receives RAND and AUTN=(SQN⊕AK,AMF,MAC), and computes with MILENAGE symmetrically to recover SQN, and to obtain XMAC (the expected MAC), response RES, CK and IK. The USIM rejects if the SQN is out of the expected range or the MAC is not the same as XMAC, and the AuC rejects if the response is not as expected (RES≠XRES). The 4G protocol slightly differs from the 3G one described in the figure (see, e.g. [5, Figure 1] for the details). However, none of its changes are relevant to our attacks. Note that while mostly based on public algorithms, MILENAGE still includes a slight amount of secrets in its specifications, e.g., the (fixed) parameter OPc is usually kept secret by mobile operators. Once an adversary recovers all the secrets stored in the USIM, he can clone it by loading the same configuration into a blank card. As mentioned in introduction, the next sections will investigate the impact of these secret parameters for physical security.

### 2.2 The MILENAGE Algorithm

The MILENAGE algorithm [13] is a suite of mathematical functions, $f1$, ..., $f5$, that are based on the AES-128. For the purposes of this paper, it suffices to consider the computation of this algorithm on the USIM side of the AKA protocol, as depicted in Figure 2. In particular, we will focus on $f5$. It is used to

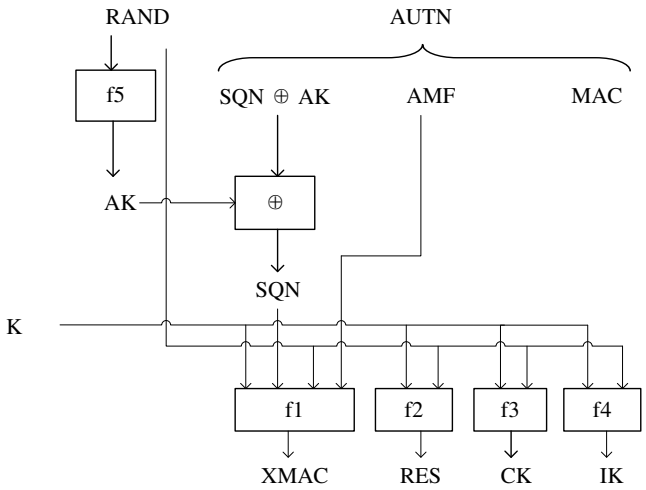


**Fig. 2.** Illustration of the computation of MILENAGE on a USIM.

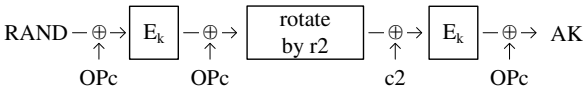


**Fig. 3.** Illustration of $f5$.

compute AK=$f5_k$(RAND) and thus allows to recover SQN = (SQN $\oplus$ AK) $\oplus$ AK, which is in turn used to compute XMAC = $f1_k$(SQN , RAND, AMF). Note that if XMAC does not equal to MAC, the USIM authentication will terminate and signal an error message, which means the rest of the functions (i.e. $f2$, $f3$ and $f4$) will not be computed. Therefore, $f5$ is a target of choice for our power analysis investigations. Yet, we mention that other functions $f1$, $f2$, $f3$ and $f4$ are similarly defined, and we refer to [16] for details on their specifications. As depicted Figure 3, $f5$ takes RAND, OPc and K as 16-byte inputs, and computes:

$$\begin{aligned} M_1 &= E_k(RAND \oplus OPc), \quad M_2 = Rotate_{r2}(M_1 \oplus OPc), \\ M_3 &= E_k(M_2 \oplus c2), \qquad\quad AK = M_3 \oplus OPc\ , \end{aligned} \tag{1}$$

with $\oplus$ a bitwise XOR, $Rotate_{r2}$ a rotate-by-$r2$-bits, and $E_k$ the AES-128 [12].

Operator-defined parameters. In our context, OPc is seen as a secret value chosen by the operator and fixed once for all its USIMs. Other parameters such as $r1, \ldots, r5$ and $c1, \ldots, c5$ have default values suggested by 3GPP specification [1], but they are also configurable (to secret values) by operators.

### 2.3 Side-channel attacks

Side-channel attacks generally exploit the existence of data-dependent and physically observable phenomenons caused by the execution of computing tasks in microelectronic devices. Typical examples of such information leakages include the power consumption and the electromagnetic radiation of integrated circuits. We will focus on the first one in the rest of this paper. The literature usually divides such attacks in two classes. First, Simple Power Analysis (SPA) attempts to interpret the power consumption of a device and deduce information about its performed operations. This can be done by visual inspection of the power consumption measurements in function of the time. SPA in itself does not always lead to key recovery, e.g. with block ciphers, distinguishing the encryption rounds does not reveal any sensitive information. Yet, it can be a preliminary step in order to reduce the computational requirements of more advanced attacks. Second, Differential Power Analysis (DPA) intends to take advantage of data-dependencies in the power consumption patterns. In its standard form [14], DPA is based on a divide-and-conquer strategy, in which the different parts of a

secret key (usually denoted as "subkeys") are recovered separately. The attack is best illustrated with an example. Say one targets the first round of a block cipher, where the plaintext is XORed with a subkey and sent through a substitution box $\mathsf{S}$. DPA is made of three steps:

1. For different plaintexts $x_i$ and subkey candidates $k^*$, the adversary predicts intermediate values in the implementation, e.g. the S-box outputs $v_i^{k^*} = \mathsf{S}(x_i \oplus k^*)$.

2. For each predicted value, the adversary models the leakages. For example, if the target block cipher is implemented in a CMOS microcontroller, the model can be the Hamming weight (HW) of the predicted values[7]: $m_i^{k^*} = \mathsf{HW}(v_i^{k^*})$.

3. For each subkey candidate $k^*$, the adversary compares the modeled leakages with actual measurements, produced with the same plaintexts $x_i$ and a secret subkey $k$. In the univariate DPA attacks (that we will apply next), each $m_i^{k^*}$ is compared independently with many single points in the traces, and the subkey candidate that performs best is selected by the adversary.

Different statistical tools have been proposed to perform this comparison. In our experiments, we will consider a usual DPA distinguisher, namely Pearson's correlation coefficient [11]. In this case, and denoting a leakage sample produced with plaintext $x_i$ and subkey $k$ as $l_i^k$, the adversary selects the subkey candidate as:

$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \frac{\sum_i (m_i^{k^*} - \overline{m}^{k^*}) \cdot (l_i^k - \overline{l}^k)}{\sqrt{\sum_i (m_i^{k^*} - \overline{m}^{k^*})^2 \cdot \sum_i (l_i^k - \overline{l}^k)^2}}, \tag{2}$$

where $\overline{m}^{k^*}$ and $\overline{l}^k$ are the sample means of the models and leakages. By repeating this procedure for every subkey (possibly exploiting smart enumeration strategies if needed [19]), the complete master key is finally recovered.

## 3 DPA against MILENAGE implementations

### 3.1 Measurement setup and target USIM cards

As depicted in Figure 4, we used a self-made card reader (with a resistor inserted for power acquisition) and ran some open source software [2] on a PC to control the test cards and execute the MILENAGE algorithm. At the same time, we used a LeCroyScope oscilloscope to acquire the power traces, and connected it with a Card-to-Terminal adapter providing an external DC power (+5V). Finally, we used MP300 SC2 to intercept the authentication messages between USIM and AuC, which provides useful information for our experiments (e.g. whether

[7] This assumption relates to the observation that in CMOS circuits, a significant part of the power consumption is dynamic, i.e. caused by the switching activity. A first-order approximation of this switching activity is given by the Hamming weight of the intermediate values produced when performing the cryptographic computations.

authentication succeeds or not). The different target USIM cards we considered in our experiments are listed in Table 1. They all include secret OPc. As for the other configurable parameters ($r1$, $c1$, ..., $r1$, $c5$), some of the USIM cards use standard (public) suggested values, and the rest use secret ones.

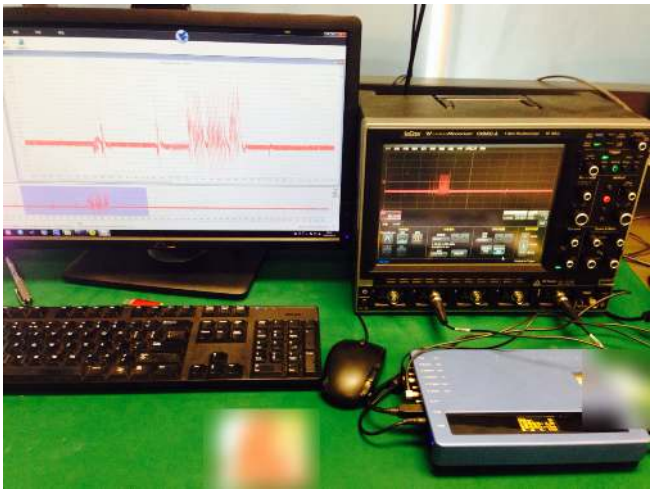

**Fig. 4.** The actual measurement setup for our experiments.

**Table 1.** List of target USIM cards with anonymized operators, manufacturers and countries of origin. (C1-1 stands for continent 1, country 1). The data and time complexity are measured respectively by the number of power traces and the total amount of time needed for the attack (including power acquisition, data processing and DPA).

| USIM | operator | manufacturer | technology | secrets | # of traces | time |
|---|---|---|---|---|---|---|
| #1 | C1-1 | C1-I | 3G UMTS | K,OPc | 200 | 10 mins |
| #2 | C1-1 | C2-II | 3G UMTS | K,OPc | 200 | 10 mins |
| #3 | C1-1 | C1-III | 3G UMTS | K,OPc | 200 | 10 mins |
| #4 | C1-2 | C3-I | 3G UMTS | K,OPc,$r_1$, ..., $r_5$,$c_1$,..., $c_5$ | 1000 | 60 mins |
| #5 | C2-1 | C2-I | 3G UMTS | K,OPc,$r_1$, ..., $r_5$,$c_1$,..., $c_5$ | 1000 | 70 mins |
| #6 | C1-3 | C1-IV | 4G LTE | K,OPc,$r_1$, ..., $r_5$,$c_1$,..., $c_5$ | 1000 | 60 mins |
| #7 | C1-3 | C1-II | 4G LTE | K,OPc,$r_1$, ..., $r_5$,$c_1$,..., $c_5$ | 1000 | 60 mins |
| #8 | C2-2 | C2-II | 4G LTE | K,OPc,$r_1$, ..., $r_5$,$c_1$,..., $c_5$ | 1000 | 80 mins |

To initiate the authentication, the PC (which plays the role of AuC) typically communicates with the USIM in the language of application protocol data unit (APDU) as follows:

```
00 A4 08 04 02 2F 00 select file with 2(0x02)-byte argument 2F 00
00 C0 00 00 1C       get response of 29(0x1C) bytes
00 B2 01 04 26       read records
00 A4 04 04 10 A0 00 00 00 87 10 02 FF 86 11 04 89 FF FF FF FF
                     select file with 16(0x10)-byte argument A0**FF
00 C0 00 00 35       get response of 53(0x35) bytes
00 A4 00 04 02 6F 07 select file with 2(0x02)-byte argument 6F 07
00 C0 00 00 19       get response of 25(0x19) bytes
00 B0 00 00 09       read binary
00 88 00 81 22 10 AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
---------------10 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
run authentication on 16(0x10)-byte RAND=(AA**AA),AUTN=(BB**BB)
```

where the '−'s are padded for alignment only. Roughly speaking, one needs to apply a sequence of "select file" APDUs from the master file (through the directory tree) to reach the application that invokes MILENAGE. The last APDU runs MILENAGE on two 16-byte arguments "AA...AA" and "BB...BB" (highlighted in blue), which can be replaced with any values for RAND and AUTN.

Note finally that the structure of the APDU is defined by ISO/IEC 7816-4, but the "command data" fields (highlighted in red in the previous example) of some APDUs may vary for different manufacturers. In the latter cases, we used some brute force search in order to remove the uncertainties.

### 3.2 Attack strategy

In order to recover OPc and K from the USIM, we interact with the card and execute the AKA protocol based on full knowledge of the inputs being processed (i.e. RAND and AUTN), which allows us to collect power consumption traces for the implementation of MILENAGE. We then perform DPA using the Hamming weight model with the following steps.

1. *Recovering K⊕OPc.* As illustrated in Figure 5, the (known) RAND is XORed with (secret) OPc before going through $E_k$ (i.e. the AES-128 encryption [12]). In this step, we therefore focus on the first round of $E_k$, where the 16-byte plaintext RAND⊕OPc is parsed as a $4 \times 4$ byte state matrix. This 16-byte plaintext is first bitwise XORed with 16-byte secret key in *AddRoundKey*. Then, each updated state byte is replaced by another one using the S-box (16 invertible lookup tables) in *SubBytes*. As a result, a simple DPA attack can be performed by considering the output of *SubBytes* as target intermediate value, viewing RAND as plaintext (instead of RAND⊕OPc as in a basic DPA setting), and OPc ⊕ K (rather than K) as the first round key.
2. *Recovering K (and OPc).* Given that K⊕OPc is already known, we just need to recover either K or OPc. A straightforward way to do this is to target

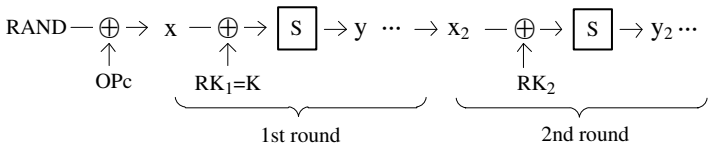


**Fig. 5.** AES S-box lookups for the first two rounds.

at the XOR operation between RAND and OPc[8] , but DPA usually works better after a non-linear operation (as explained in [17]). Therefore, a much better approach (i.e. requiring less measurements) is to attack the second block cipher round. That is, upon successful key recovery in the first round, we obtain the output of the first round (i.e. $x_2$ in Figure 5), which enables us to perform another DPA on the second round to recover the 2nd round key $RK_2$, from which we compute the corresponding encryption key K.

3. *Recovering the other secret parameters.* One of our target USIMs contained secret values for $r1, \ldots, r5$ and $c1, \ldots, c5$, which can also be recovered with a divide-and-conquer side-channel attack, as we now explain for $c2$ and $r2$ (the same techniques applies to the other secret parameters). Based on the previous attacks, we now know the 128-bit intermediate result prior to the $Rotate_{r2}$ operation (illustrated in Figure 3), say $v_0 v_1 \ldots v_{127}$. $Rotate_{r2}$ is simply a right cyclic shift of this known value by $r_2$ bits. In order to recover $r_2$, we first write it as a multiple of some $i$ plus remainder $j$, i.e. $r2 = 8i + j$. Then we consider the sequence:

$$\underbrace{(v_j v_{j+1} \ldots v_{j+7})}_{\text{byte } 0} \ldots \underbrace{(v_{j+120} \ldots v_{127} v_0 \ldots v_{j-1})}_{\text{byte } 15} \tag{3}$$

which represents $v_0 v_1 \ldots v_{127}$ rotated by $j$ bits. Assuming that the power trace is correlated with the Hamming weight of every individual byte of this rotated value, we can simply make guesses about $j$ (which has only 8 possibilities) and test these guesses with a correlation analysis between the Hamming weight of any byte in (3) and the power trace. Once we recovered $j$, we then take into account the fact that the sequence was shifted by $8i + j$ bits, which means that the correlations for the 16 bytes above should appear in the order of byte numbers $15-i+1, \ldots, 15, 0, \ldots, 15-i$. Therefore, we can again identify the value of $i$ by doing a correlation analysis (with Hamming weight model) for every byte, and finding out the order in which significant correlations appear for those bytes. Eventually, an attack as previously described can be used against the second $E_k$ in Figure 3 to recover $c2$.

[8] Which succeeded as well, but less efficiently than the following proposal.

### 3.3 Experimental results

For conciseness, we show power traces and coefficients plots for one of the USIM cards we investigated (with secret parameters). Results were essentially similar for the other USIM cards. In all cases, successful attacks were obtained based on several hundred power traces that were acquired in a few minutes, and pre-processed with a low-pass filter. We began with SPA to identify the relevant parts of the power traces, sent the USIM card authentication commands with randomized inputs for this purpose, and as expected received a "*user authentication reject*" error due to the mismatch of XMAC and MAC (or SQN out of range). For illustration, Figure 6 gives a view of an entire power trace collected, where we identify 4 similar parts, and each part has 10 rounds. We observe that the last round is less obvious to spot than the other ones, which can be justified by the fact that AES-128 computes no *MixColumns* in its last round. Note that in this case, the power consumption we measured only corresponds to that of $f_1$ (to compute XMAC for verification) and $f_5$ (to obtain AK and thus recover SQN) since $f_2$, $f_3$ and $f_4$ are not computed on an authentication failure. We could therefore safely assume that the first two (resp. last two) parts represent the two AES executions of $f5$ (resp. the two AES executions of $f1$).

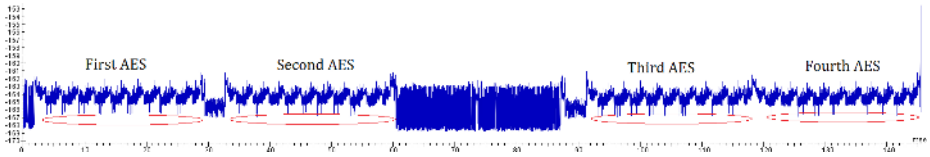


**Fig. 6.** An overview of a power trace.

We aligned the traces corresponding to the randomized inputs with a standard pattern matching method. That is, we choose a unique pattern close to the part of the traces of our interest (e.g. the header part of Figure 6), used cross-correlation tools to identify this pattern in all the traces in an automated manner, and then aligned the traces based on those cross-correlations. This simple technique was sufficient in our experiments, due to the relatively low noise level of our traces. Thanks to the iterative nature of the AES, we could then divide the traces into segments that correspond to their respective rounds. Furthermore, for each round we identified the parts of the *AddRoundKey* and *MixColumns* operations, by comparing the differences between that round and the 10-th round. Figure 7 is a zoomed-in view of the trace segment nearby the first round, where the four operations *AddRoundKey*, *SubBytes, ShiftRows*, and *MixColumns* are identified. We could also verify with correlation analysis that the part of trace prior to *AddRoundKey* corresponds to the operation of RAND⊕OPc. The fact that all operations appeared to be carried out sequentially gave us hints that the MILENAGE algorithm was implemented in software. This suggested that trying an attack with a Hamming weight leakage model might be a good option.

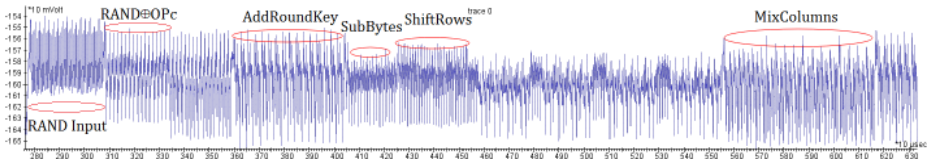


**Fig. 7.** A zoomed-in view of part of a power trace.

We finally discuss the key recovery following the strategies in Section 3.2.

1. *Recovering K⊕OPc.* We focused on the *SubBytes* part of Figure 7 and performed our DPA attack exactly as described in the previous section. The result of the correlation analysis for the first byte is shown in Figure 8. Note that the peak was clearly sufficient to recover all key bytes without ambiguity. Furthermore, the time at which they appeared were in line with our previous assumptions regarding when the S-box computations take place.

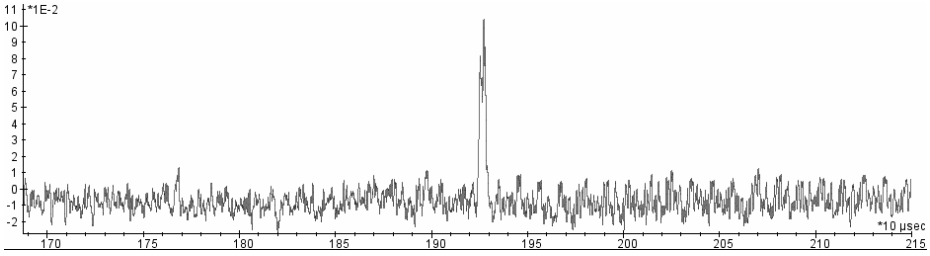


**Fig. 8.** DPA result on SubBytes to recover K⊕OPc.

2. *Recovering K (and OPc).* As previously detailed, a straightforward DPA against the second AES round allowed us to recover the second subkey, from which K (and OPc) can be derived. Correlation plots (and hence, attack efficiencies) were similar as for recovering K⊕OPc.
3. *Recovering the other secret parameters.* As mentioned in Section 3.2, we can write $r2 = 8i + j$ and find out the value of $j$ by correlating the Hamming weight of any single byte from (3) (with different hypothetical values about $j$) with the power trace. As depicted in Figure 9, we indeed obtained high correlations upon correct guesses about $j$. We then correlated all bytes in (3) (based on the correct value of $j$) to the power trace, and we expected to see that correlations occur in sequential order for bytes $15-i+1, \ldots, 15, 0, \ldots, 15-i$. For instance, the value of $i$ in Figure 9 should be 8. Eventually, we performed an additional DPA against the second $E_k$ in Figure 3 to recover $c2 \oplus$K (and thus $c2$), which yielded not particular challenge. The process for recovering other parameters $r1$, $c1$, $r3$, $c3$, …, $r5$ and $c5$ is identical.
4. *Correctness verification.* We used MP300 SC2 to acquire the actual messages (RAND and AUTN) communicated between the USIM card and the AuC.

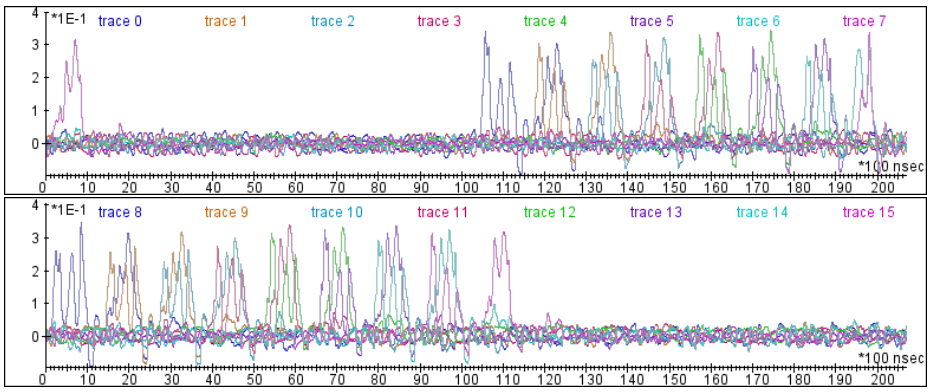


**Fig. 9.** Correlation traces between the Hamming weight of the bytes in (3) and the power trace, where traces 0, 1, ..., 15 correspond to bytes 0, 1, ..., 15.

Based on the K and OPc values we recovered thanks to side-channel analysis, XMAC can be calculated accordingly. We could therefore verify that our calculated XMAC equals to the MAC contained in AUTN, and thus confirm that the key recovery of K and OPc was successful in all cases.

## 4 Conclusions

Technically, the results in this work are essentially based on known techniques (i.e., differential power analysis attacks). Yet, they are useful to illustrate that the move to AES-based encryption algorithms in 3G/4G USIM cards did not systematically take advantage of state-of-the-art countermeasures against side-channel attacks. Indeed, the USIM cards we analyzed essentially relied on plain (unprotected) software implementations of the AES. Besides, it is interesting to observe that the (minor) obfuscation of the MILENAGE specification with operator-defined secrets has essentially no impact on side-channel security (which was never claimed but is interesting to confirm). Needless to say, it would be interesting to exploit the broad literature on secure AES implementations and countermeasures against side-channel attacks to improve this situation.

**Acknowledgments.** This research work was supported in parts by the National Basic Research Program of China (Grant 2013CB338004) and the European Commission through the ERC project 280141 (acronym CRASH). Yu Yu was supported by the National Natural Science Foundation of China Grant (Nos. 61472249, 61103221). François-Xavier Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). Zheng Guo was supported by the National Natural Science Foundation of China Grant (Nos. 61402286,

61202371). Dawu Gu was supported by the National Natural Science Foundation of China Grant (No. 61472250), the Doctoral Fund of Ministry of Education of China (No. 20120073110094), the Innovation Program by Shanghai Municipal Science and Technology Commission (No. 14511100300), and Special Fund Task for Enterprise Innovation Cooperation from Shanghai Municipal Commission of Economy and Informatization (No. CXY-2013-35).

## References

1. 3GPP specification: 35.206 (Specification of the MILENAGE algorithm set), http://www.3gpp.org/DynaReport/35206.htm
2. Cryptography for mobile network - C implementation and Python bindings, https://github.com/mitshell/CryptoMobile
3. List of LTE networks, http://en.wikipedia.org/wiki/List_of_LTE_networks
4. List of UMTS networks, http://en.wikipedia.org/wiki/List_of_UMTS_networks
5. Security Technology for SAE/LTE, https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/rd/technical_journal/bn/vol11_3/vol11_3_027en.pdf, Retrieved on Jan. 6, 2015.
6. Barkan, E., Biham, E., Keller, N.: Instant ciphertext-only cryptanalysis of GSM encrypted communication. In: Advances in Cryptology - CRYPTO 2003. pp. 600–616 (2003)
7. Biham, E., Dunkelman, O.: Cryptanalysis of the A5/1 GSM stream cipher. In: 1st International Conference on Cryptology in India (INDOCRYPT 2000). pp. 43–51 (2000)
8. Biryukov, A., Shamir, A., Wagner, D.: Real time cryptanalysis of A5/1 on a PC. In: FSE. pp. 1–18 (2000)
9. Bogdanov, A., Eisenbarth, T., Rupp, A.: A hardware-assisted realtime attack on A5/2 without precomputations. In: 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007). pp. 394–412 (2007)
10. Briceno, M., Goldberg, I., Wagner, D.: GSM Cloning. http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html (1998), retrieved on Jan. 6, 2015.
11. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004). pp. 16–29 (2004)
12. Daemen, J., Rijmen, V.: The design of Rijndael: AES – the advanced encryption standard. Springer (2002)
13. Gindraux, S.: From 2G to 3G: a guide to mobile security. In: 3rd International Conference on 3G Mobile Communication Technologies. pp. 308–311 (2002)
14. Mangard, S., Oswald, E., Standaert, F.: One for all - all for one: unifying standard differential power analysis attacks. IET Information Security 5(2), 100–110 (2011)
15. Maximov, A., Johansson, T., Babbage, S.: An improved correlation attack on A5/1. In: 11th International Conference on Selected Areas in Cryptography (SAC 2004). pp. 1–18 (2004)
16. Niemi, V., Nyberg, K.: UMTS Security. Wiley Online Library (2003)
17. Prouff, E.: DPA attacks and S-boxes. In: 12th International Workshop on Fast Software Encryption (FSE 2005). pp. 424–441 (2005)
18. Rao, J.R., Rohatgi, P., Scherzer, H., Tinguely, S.: Partitioning attacks: Or how to rapidly clone some GSM cards. In: 2002 IEEE Symposium on Security and Privacy, Berkeley, California, USA. pp. 31–41 (2002)

19. Veyrat-Charvillon, N., Gérard, B., Renauld, M., Standaert, F.: An optimal key enumeration algorithm and its application to side-channel attacks. In: 19th International Conference on Selected Areas in Cryptography (SAC 2012). pp. 390–406 (2012)
20. Zhou, Y., Yu, Y., Standaert, F., Quisquater, J.: On the need of physical security for small embedded devices: A case study with COMP128-1 implementations in SIM cards. In: 17th International Conference on Financial Cryptography and Data Security (FC 2013). pp. 230–238 (2013)