On the Need of Physical Security for Small Embedded Devices: a Case Study with COMP128-1 Implementations in SIM Cards (long version)

Yuanyuan Zhou[1], Yu Yu[2], Francois-Xavier Standaert[3]
and Jean-Jacques Quisquater[3]

[1] Brightsight, Delft, The Netherlands.
[2] East China Normal University and Tsinghua University, China.
[3] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.

Abstract. Ensuring the physical security of small embedded devices is challenging. Such devices have to be produced under strong cost constraints, and generally operate with limited power and energy budget. However, they may also be deployed in applications where physical access is indeed possible for adversaries. In this paper, we consider the case of SIM cards to discuss these issues, and report on successful side-channel attacks against several (old but still deployed) implementations of the COMP128-1 algorithm. Such attacks are able to recover cryptographic keys with limited time and data, by measuring the power consumption of the devices manipulating them, hence allowing cards cloning and communications eavesdropping. This study allows us to put forward the long term issues raised by the deployment of cryptographic implementations. It provides a motivation for improving the physical security of small embedded devices early in their development. We also use it to argue that public standards for cryptographic algorithms and transparent physical security evaluation methodologies are important tools for this purpose.

1 Introduction

Protecting present information systems requires considering both hardware and software security issues, with their specific risks and constraints. In general, software attacks are cheaper and tools for performing them can be rapidly disseminated. Yet, they are also easier to patch with code updates. By contrast, hardware attacks are more difficult to perform, as they require laboratory equipment that ranges from low-cost to highly expensive. But they can be more difficult to fix a posteriori, as hardware updates imply more expensive development processes, and usually take place in the longer term. Hence, finding the best balance between hardware and software security is a difficult task for system designers. This concern is particularly critical with cryptographic implementations that may be the target of fault insertion attacks [7] and side-channel attacks [26, 27, 44]. In the latter case (that will be our focus in this paper), the adversary exploits physical information leakage such as the power consumption of the device running a cryptographic algorithm, in order to extract secret information such as secret keys. As the power consumption of a device is expected to be correlated with the data it manipulates, these attacks essentially proceed by comparing key-dependent leakage predictions with actual measurements. When no particular care is taken, cryptographic implementations frequently turn out to be highly susceptible to side-channel attacks, as recently exhibited with results against the KeeLoq remote keyless entry systems (at CRYPTO 2009 [18]), the Mifare DESFire contactless smart cards (at CHES 2011 [38]), or Xilinx's FPGA bitstream encryption mechanisms (at ACM CCS 2011 [35]).

As a result of their implementation-specific nature, side-channel attacks are particularly difficult to prevent. That is, since these attacks do not target algorithms, but actual instances of their implementation in various technologies, it is hard to design general (and efficient) solutions that allow making *any* implementation of an algorithm secure. Hence, most state-of-the-art techniques to improve security against such attacks rely on heuristic assumptions (e.g. the masking and hiding principles in [29]), and need to be confirmed by empirical evaluation. Note that although this situation raises challenging research problems (e.g. discussed at the CHES workshops [13]), producing practically secure integrated circuits is not out of reach. Nowadays, most smart card companies have products evaluated by independent laboratories and granted with high security levels by certification authorities, e.g. [1, 10]. But this improved security usually comes at the cost of implementation overheads that may limit their practical deployment. In addition, and although having certificates may be a good selling point, obtaining them also takes time and money (see, e.g. the Common Criteria [15] and EMVco [19]). Hence, while such certificates are a frequent requirement for security products of government agencies and banking applications, they are much less usual in lower-cost applications using SIM or transport cards.

A typical example of this lack of general approaches for preventing side-channel attacks was actually given by a team from IBM in 2002, for implementations of the COMP128-1 algorithm used in GSM communications. In a paper from IEEE S&P [45], Rao et al. first showed that a straightforward application of Differential Power Analysis (DPA) was not successful against the instances of SIM cards they were analyzing (presumably because of some ad hoc countermeasures). Then, they observed that at the first round of COMP128-1's compression function, the substitution-box (S-box) consists of 512 values (i.e. are accessed by a 9-bit index). It implies that on low-speed SIMs (with 8-bit CPU) this S-box has to be implemented using two (typically equal-size) lookup tables. Knowing which table is being accessed (which could be identified from the power traces) could result in a key recovery with a maximum of 1000 random challenges, or 255 chosen ones, or just 8 adaptively chosen ones (i.e. as efficient as a binary search). This data corresponds to the monitoring of a few minutes of SIM card operations. In other words, while the standard DPA approach did not directly lead to successful key recoveries, a slightly modified path taking advantage of the implementation specificities did a perfect job. Fortunately, the attack (exploiting the 8-bit addressing) was only applicable to 8-bit-CPU SIM cards. Since 2003, the major operators have been gradually phasing out the use of legacy SIM by issuing products equipped with 16-bit CPU data bus, ruling out this possibility.

In this paper, we take advantage of this SIM card example to discuss the practical challenges raised by hardware security issues. For this purpose, we investigate the resistance of SIM cards from two different GSM operators and four different manufacturers against DPA. Our experiments target implementations of the COMP128-1 algorithm in 16-bit CPUs, that are secure against the IBM 2002 attack. They are also secure against the algorithmic collision attacks described in [8]. While COMP128-1 is progressively being replaced by improved versions, it is still deployed in commercial devices, and sometimes being distributed. We show how DPA can be used to recover its 128-bit secret key, allowing cards cloning and communications eavesdropping. Depending on the targets and measurement setup available to the adversary, the attacks require physical access to the device ranging from minutes to a couple of hours. Interestingly, our results can be seen as the methodological counterpart of the 2002 ones. While the previous analysis in [45] targets instances of SIM cards (presumably) secure against standard DPA attacks but weak against dedicated ones, our instances are robust against the IBM attack but weak against standard DPA.

The important conclusions of this work are methodological. First, our results exhibit the long term nature of physical security concerns. While cryptographic implementations are not deployed as long as algorithms, they may remain in service for a couple of years, and are not straightforward to upgrade. This observation makes a case for considering physical security as an important feature of small embedded devices in general. Technical solutions exist to make side-channel attacks significantly more difficult to perform, e.g. the previously mentioned masking and hiding. But they work best if considered early in a design process. Second, we observe that public standards for cryptographic algorithms are useful to improve the efficiency of countermeasures against physical attacks. By contrast, the closed-source nature of COMP128-1 has significantly limited the amount of research about its secure implementations. Finally, transparent and reproducible (possibly standardized) methodologies for physical security evaluations are required, in order to quantify physical security on a sound basis.

The rest of the paper is organized as follows. Background about the GSM infrastructure, the COMP128-1 hash algorithm and side-channel attacks is given in Section 2. Section 3 contains the technical description of the different attacks we mounted, as well as our experimental results. Countermeasures are briefly discussed in Section 4. Eventually, we conclude the paper in Section 5, by discussing lessons learned and possible directions for future research.

Contact with the operators. Our experiments have been performed in 2010. The different operators exploiting the SIM cards that we discuss in this paper have been contacted before publication of our results. Updates towards implementations of COMP128-2 and COMP128-3, including protections against side-channel attacks, are under development (or maybe already deployed).

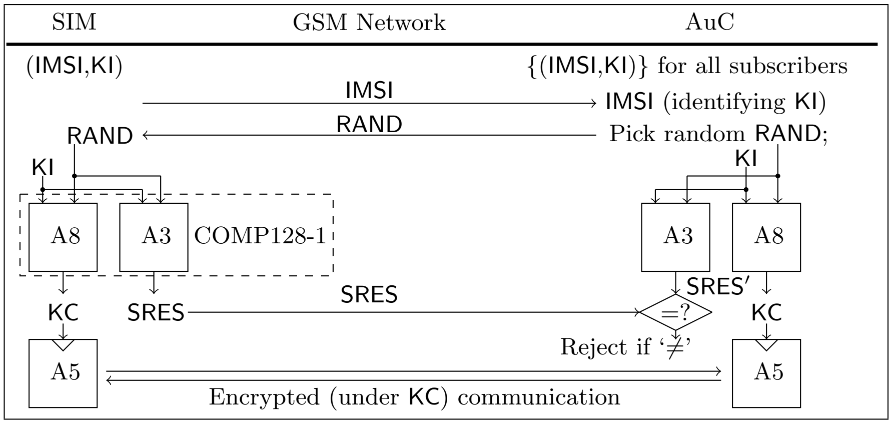

Fig. 1. Illustration of the protocol between a SIM card and an authentication center.

2 Background

2.1 GSM infrastructure and COMP128-1

GSM-Network. Despite the migration to 3G networks being a major trend, the *Global System for Mobile Communication* (GSM) remains the current dominant technology for mobile communications worldwide [53], especially in many developing countries. The (oversimplified) infrastructure of a GSM network is represented in Figure 1. It consists of a *subscriber identity module* (SIM) which is an integrated circuit located in a mobile telephony device, and the operator's network *authentication center* (AuC). A SIM stores necessary credentials to identify itself and authenticate to the network, including a symmetric secret key KI (*key identifier*), a serial number ICCID (*Integrated Circuit Card ID*) and a unique user number IMSI (*International Mobile Subscriber Identity*).

GSM Authentication and Encryption. The GSM specification allows network operators to choose their algorithms for authentication, and only mentions COMP128-1 (aka A3/A8) as an example. COMP128-1 is a cryptographic hash function that was not available to the public until it was reverse engineered by Briceno, Goldberg and Wagner [8] in 1998. As depicted in Figure 1, each SIM has its ID-key pair (IMSI,KI) while the AuC stores the keys for all registered SIMs (indexed by IMSI). So a SIM has to first send its IMSI to AuC in order to identify the corresponding KI. Then, both parties engage in a challenge-response protocol: the AuC sends a random challenge RAND to the SIM. The SIM then hashes the concatenated string RAND‖KI into digest SRES‖KC, part of which (i.e. SRES) is sent back to the AuC as a response. The AuC rejects the authentication if the received SRES differs from its own version. Otherwise, the parties establish communication encrypted using stream cipher A5 under session key KC.

COMP128-1 is a cryptographic hash function that takes a 32-byte input (i.e. a 16-byte challenge RAND and a 16-byte secret key KI), and produces a 12-byte output (a 4-byte response SRES and an 8-byte session key KC). We list the pseudo-code of COMP128-1 below, where we slightly abuse the notation by using RAND[j] (resp. KI[j]) to denote the $(j+1)^{-th}$ byte of RAND (resp. KI).

```
function COMP128-1(RAND, KI)
begin
    for $j$=16 to 31 do    {load RAND}
        $X[j]$ := RAND[$j-16$];
    for $i$=0 to 7 do    {8 iterative rounds}
    begin
        for $j$=0 to 15 do    {load KI}
            $X[j]$ := KI[$j$];
        call Compress;    {5-subround compression}
        call FormBitsFromBytes;    {assemble result}
        if $i < 7$ then    {permute except the last round}
            call Permute;
    end;
end;
```

The algorithm first loads RAND into the upper half of a 32-byte vector $X[]$, namely $X[16-31]$. Then it iterates eight rounds, where one loads KI into $X[0-15]$ and compresses the 32-byte $X[0-31]$ into 16 bytes (i.e. after compression each $X[j]$ consists of 4 useful bits), which are then assembled by FormBitsFromBytes into $X[16-31]$, followed by a permutation on $X[16-31]$ (except for the last round). The first 12 bytes of $X[16-31]$ are produced as output. For all known attacks (and our DPA attacks), it is sufficient to consider the code up to the first invocation of the compression function. As detailed in the pseudo-code below and illustrated in Figure 7 (in Appendix), the compression function consists of 5 (sub-)rounds of table look-ups using S-boxes $T_0[512]$, $T_1[256]$, $T_2[128]$, $T_3[64]$ and $T_4[32]$ respectively, where each T_j replaces $(9-j)$-bit strings with $(8-j)$-bit ones. We often refer to the pairwise substitution structure as "butterfly". In each of the 5 levels, compression is performed on 2 equal sized sections, and two input bytes are used to calculate the index for the table. The result is the output byte. More precisely, at each (sub-)round $j+1$, for every pair of $X[m]$ and $X[n]$ with $n = m + 4^{4-j}$, two intermediate values y and z are computed as in the pseudo-code, and the values of $X[m]$ and $X[n]$ are replaced by $T_j[y]$ and $T_j[z]$.

```
function Compress($X[0-31]$)
begin
    for $j = 0$ to 4 do    {5 sub-rounds in Figure 7}
        for $k = 0$ to $2^j - 1$ do
            for $l = 0$ to $2^{4-j} - 1$ do
            begin
                $m := l + k \cdot 2^{5-j}$;
                $n := m + 2^{4-j}$;
                $y := (X[m] + 2{\cdot}X[n]) \mod 2^{9-j}$;
                $z := (2{\cdot}X[m] + X[n]) \mod 2^{9-j}$;
                $X[m] := T_j[y]$;
                $X[n] := T_j[z]$;
            end;
end;
```

Cryptanalysis of COMP128-1 and A5. The most severe cryptanalytic weakness in the GSM infrastructure was identified together with the reverse engineering of the algorithm in 1998 [8]. Briceno et al. showed that COMP128-1 was fatally flawed due to a lack of diffusion in its compression function, which leads to a collision attack (also called Narrow Pipe Attack). It takes roughly 131, 000 challenge-response pairs to recover KI, and about 7.5 hours to acquire the necessary data given physical access to the SIM. Quite naturally, recovering the key identifier completely cancels the security of the infrastructure. As a reaction, the GSM association developed newer (but still *proprietary*[1]) versions, namely COMP128-2 and COMP128-3. While these newer versions are already widely deployed in Europe, many SIM cards implementing COMP128-1 remain in service in other countries. Besides, several strong cryptanalysis results have also been published against various versions of the A5 algorithm, including [3–6, 34], leading to real-time and low-cost attacks demonstrated by Karsten Nohl and Sylvain Munaut at the 2010 Chaos Communication Congress. Here as well, the move towards adopting the A5/3 algorithm is slowly taking place [42].

SIM Cloning Fraud and Countermeasures. For unprotected (and weakly protected) implementations of COMP128-1, SIM card cloning kits are available from eBay for about $10 which typically include a USB SIM reader/writer, a programmable wafer card, and a software tool, where the tool extracts the KI by realizing collision attacks. Depending on the key recovery tools ("SimScan", "WoronScan", "SimMaster" to name a few) and their randomized computation, the time spent on key extraction can range from half an hour to 36 hours. Although physical access to the SIM is required, a practical scenario could be that a retailer makes duplicates of the SIMs in stock, and later makes fraudulent calls and payments. Alternative scenarios include access to security sensitive locations, where guests are required to hand over their mobile phones to a security officer, and get them back when checking out. Beyond the direct consequences of cloning for the security of the GSM communications, one can mention possible consequences for other security infrastructures relying on SIM card security. As a typical illustration, and as part of the multi-factor authentication for Internet banking, some commercial banks send one-time passwords to customers' mobile phones rather than to issue additional secure hardware tokens. In order to prevent frauds, most SIM cards implementing COMP128-1 are now deployed with a combination of protections against cloning attacks based on collisions.

[1] We recall the Kerckhoffs' principle that a cryptosystem should be secure even if everything about the algorithm, except the secret key, is public knowledge. In this respect, an advantage of the 3G technology (over GSM) is that its authentication protocol is based on the (public and well-studied) Advanced Encryption Standard.

For this purpose, a natural measure is to set a maximal number of challenge requests before the SIM locks itself. However, this limit has to be above the number of requests a SIM receives during its lifetime (under normal operation) in order not to trouble legitimate users. For example, it is set to 65,535 by many U.S. operators [24]. Hence and as a complement, the so-called "Indexed Challenges" can be implemented: it essentially pre-stores a few byte patterns that cause 2R-collisions, and upon successful pattern-match of a requested challenge, proceeds with the computation by replacing the true KI with a fake one (pre-stored on the SIM) which will eventually lead to a false output. These Indexed Challenges turn out to be insufficient as they neither "punish" any suspected malicious behavior, nor do they handle any collision attacks beyond the second sub-round. To address this problem, from 2009 some operators started to put in place a new countermeasure referred to as "Collision Free" in the rest of the paper. In this case, the SIM stores N (e.g. 50, typically) records of previously queried challenges in an Elementary File (EF). In case the current challenge RAND matches any record in 5 or more bytes (which presumably captures the characteristics of collision attacks at 2R, 3R and above), it is counted as an attack. The SIM is locked if more than 255 attacks are detected. Otherwise, RAND is passed to COMP128-1 as input. A Random Number Generator (RNG) is used to provide randomness for deciding whether to store each challenge RAND or not, and which existing record to replace with. This countermeasure considers not only 2R- collision attacks, but also those at subsequent sub-rounds, with a good chance of causing a SIM lock. To the best of our knowledge, it is the start-of-art countermeasure to deter SIM cloning attacks on COMP128-1 implementations.

2.2 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 present 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. For example with block ciphers, distinguishing the encryption rounds does not reveal any sensitive information. Yet, it is usually an important 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 [30], 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 S. DPA is made of three main steps:

1. For different plaintexts x_i and subkey candidates k^*, the adversary predicts intermediate values in the target implementation. For example, one could predict S-box outputs and get values $v_i^{k^*} = \mathsf{S}(x_i \oplus k^*)$.
2. For each of these predicted values, the adversary models the leakages. For example, if the target block cipher is implemented in a CMOS-based microcontroller, the model can be the Hamming weight (HW) of the predicted values[2]. One then obtains modeled leakages $m_i^{k^*} = \mathsf{HW}(v_i^{k^*})$.
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), 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 [9]. 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:

$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \frac{\sum_i (m_i^{k^*} - \overline{m}^{k^*}) \cdot (l_i^k - \overline{l}^k)}{\sqrt{\sum_i (m_i^{k^*} - \overline{m}^{k^*})^2 \cdot \sum_i (l_i^k - \overline{l}^k)^2}}, \tag{1}$$

where $\overline{m}^{k^*}$ and $\overline{l}^k$ are the sample means of the models and leakage samples. By repeating this procedure for every subkey, the complete master key is finally recovered. Other distinguishers will be discussed in Section 3.4.

3 DPA attacks against implementations of the COMP128-1 algorithm in SIM cards

3.1 Target SIM cards & measurement setup

In this section, we perform DPA attacks on four representative SIM cards denoted as #1,#2, #3 and #4. As summarized in Table 1, they correspond to different operators and manufacturers and implement different countermeasures against collision attacks: SIM#1 and SIM#2 are susceptible to collision attacks in 20 000 and more queries, SIM#3 and SIM#4 are immune against them.

We used a LeCroy WavePro 950 oscilloscope to acquire the power traces, via a small resistor of 25 Ohm between the GND of power supply and the GND of a self-made Card-to-Terminal adapter. The Card-to-Terminal adapter was tweaked to provide an external DC power to the test card via a Kenwood P18A power supply (+5V), and to provide an external clock to the card via an Agilent

[2] 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.

Table 1. Target SIM cards.

	Manufact.	Operator	Countermeasure(s)
SIM#1	I	A	Not Available
SIM#2	II	B	I-C
SIM#3	III	B	I-C + C-F
SIM#4	IV	B	I-C + C-F

33120A function generator(5MHz Frequency, 2.2V Amplitude and 1.1V Offset). We used a commercially available card reader and software to control the test card during the acquisitions. In addition, we used a Keithley 488 GPIB card (i.e. a PCI card installed inside a PC) to communicate with the oscilloscope.

3.2 Preprocessing of the traces

As usual when implementing side-channel attacks, we started by applying SPA in order to identify the relevant parts of the power traces. This task is easy for SIM#1 and SIM#2. As shown in Figure 2, we can identify the 8 iterative rounds of COMP128-1 by visual inspection. Next, by further zooming on the different iterations, we can even observe the 5 sub-rounds of the COMP128-1 compression function, as illustrated in Figure 3. Therefore, it is directly possible to extract the parts of the power traces where to apply DPA for these two targets.

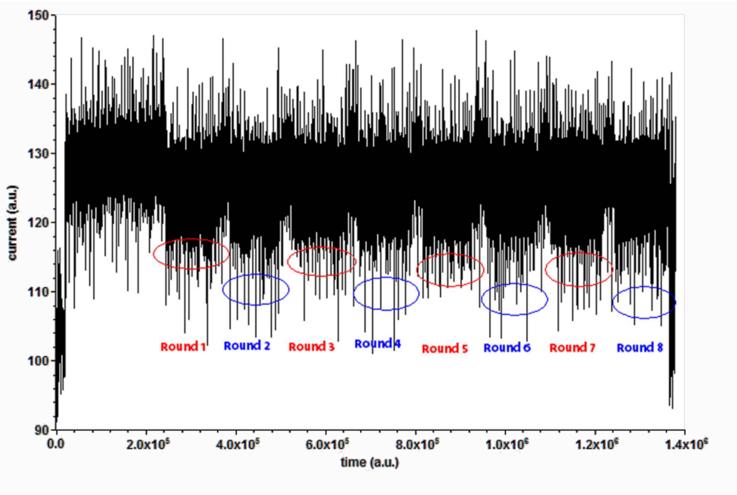

Fig. 2. A power trace from SIM#1.

The situation slightly differs for SIM cards #3 and #4, where the Collision Free countermeasure was implemented. As illustrated in Figure 4 (and Figure 8 in Appendix), it is again possible to identify the COMP128-1 operations (as

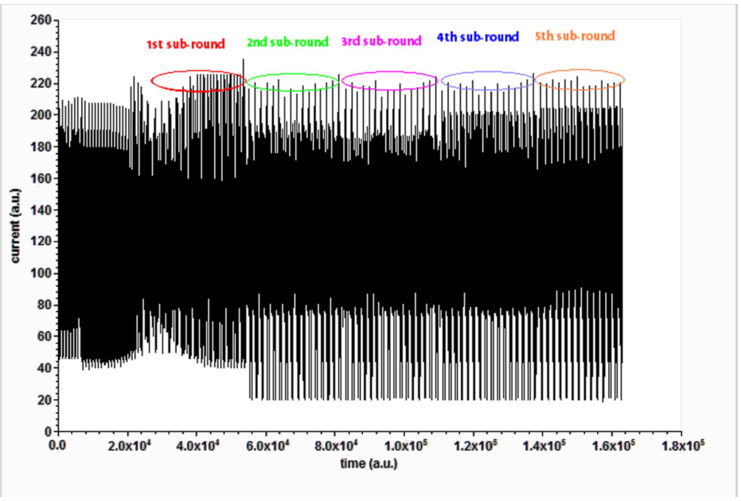

Fig. 3. Zoom on a power trace from SIM#2.

well as the Indexed Challenges) in the power traces. Yet, the Collision Free countermeasure includes a randomized memory writing operation (i.e. it uses randomness to decide whether to store a current request or not). Therefore, the length of the power traces varies for different inputs, which requires special care for aligning the traces after acquisition. In order to deal with this situation, a simple solution is to apply pattern matching techniques. That is, we selected a characteristic pattern including the samples of interest for our DPA attacks, and then systematically identified them in following traces using cross-correlation. As the noise level in our measurements was relatively low, such a simple heuristic was sufficient for performing successful key recoveries, as will be described next.

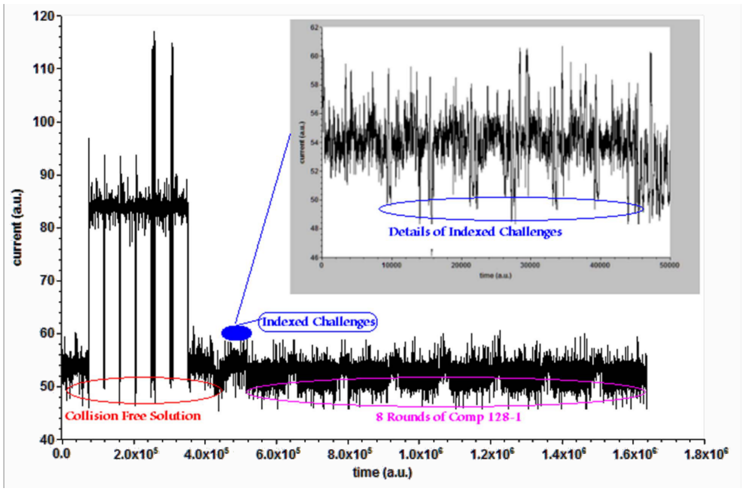

Fig. 4. A power trace from SIM#3.

3.3 DPA attack results

First, we mention that no countermeasures in our target SIM cards prohibit random queries. Therefore, we generated our traces by repeatedly executing the COMP128-1 algorithm with such inputs. Next, we applied exactly the divide-and conquer strategy described in Section 2.2. That is, we performed DPA with Pearson's correlation coefficient and used the Hamming weight power model. We applied our distinguisher to all the leakage samples l_i^k that were selected using the SPA in the previous section. Finally, and as target values $v_i^{k^*}$, we focused on the intermediate values y and z at the first sub-round of the first round in the implementation of COMP128-1, namely:

$$\begin{aligned} y &= (X[m] + 2{\cdot}X[n]) = (\mathsf{KI}[m] + 2{\cdot}\mathsf{RAND}[m]) \ \mathsf{mod}\ 2^{9-j}, \\ z &= (2{\cdot}X[m] + X[n]) = (2{\cdot}\mathsf{KI}[m] + \mathsf{RAND}[m]) \ \mathsf{mod}\ 2^{9-j}. \end{aligned}$$

For each $0 \leq m \leq 15$, we built predictions for the 256 possible values of $\mathsf{KI}[m]$ and performed the comparison. The result of such a comparison for one of the 16 COMP128-1 subkeys is given in Figure 5 for SIM#2 (and in Appendix, Figure 5 for SIM#1). The figures plot the value of Pearson's correlation coefficient over time, using y as a target value. We observe that a significant peak is distinguishable at the time samples where the computation of y actually takes place, and this peak only appears for the correct subkey candidate. As expected, the situation is slightly more challenging for SIM#3 (for which the result is given in Figure 6) and SIM#4 (for which the result is given in Appendix, Figure 10). This is due to more noisy traces and the previously mentioned synchronization issue. Yet, in both cases, a DPA peak remained clearly distinguishable, and we could always identify the COMP128-1 subkeys. Finally, we consistently recovered the full key of SIM#1 and SIM#2 with an amount of traces in the hundreds range, and this number extends to the thousands range for SIM#3 and SIM#4.

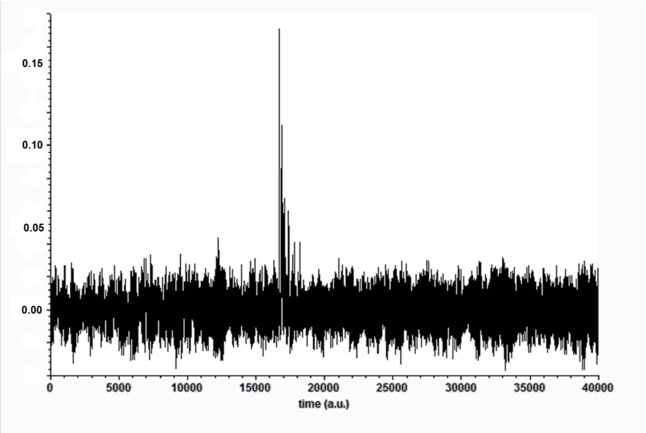

Fig. 5. DPA result against SIM#2.

These estimated data complexities are in accordance with the work of Mangard at CT-RSA 2004 [28], where it is shown that the number of measurement traces needed to recover a subkey is inversely proportional to the square of the correlation coefficient estimated for the correct key candidate. In practice, these data complexities corresponds to a few minutes to a couple of hours of acquisition, depending on the target and speed of the setups available to an adversary.

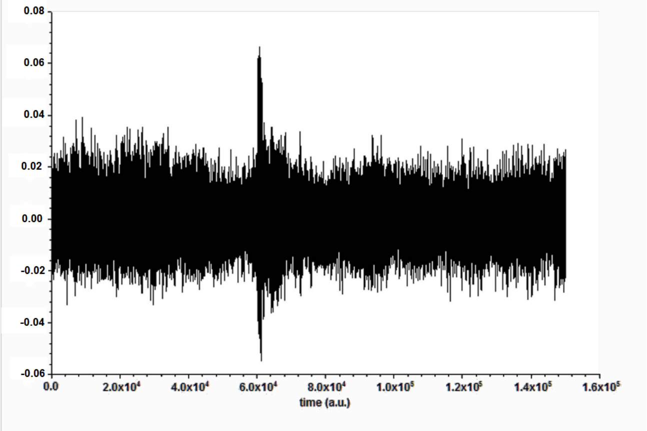

Fig. 6. DPA result against SIM#3.

3.4 Possible improvements of the attacks

Although the tools used in this paper are sufficient to demonstrate the existence of exploitable leakage in different SIM cards, there are many tracks for improving them and reducing data complexity. A better preprocessing including advanced synchronization methods and possible filtering of the noise comes in the first place [14, 56]. Besides, the Hamming weight model that we exploit could certainly be refined. Using profiling such as in template attacks is the optimal solution for this purpose [12]. Regression-based approaches are another more flexible option [48], that can possibly be applied in a non-profiled scenario [17]. Moving towards multivariate attacks, taking advantage of several leakage points concurrently, and possibly exploiting dimensionality reduction techniques, is yet another possibility [2, 50]. Finally, in case side-channel attacks are limited by a bounded data complexity, it is also possible to trade a lack of measurements for more offline computations, e.g. using enumeration algorithms [57].

4 Countermeasures

Numerous solutions to improve the security of embedded devices against side-channel have been proposed in the open literature. In general, the state-of-the-art intuition is that none of them is sufficient to completely prevent the threat of physical adversaries. Hence, modern security chips usually combine different types of protections, at different abstraction levels. From the application and usability point of view, these countermeasures can roughly be classified among two essentially orthogonal axes. On the one hand, they can be hardware or software. Hardware-based countermeasures offer the most direct way to prevent the leakage, as they tackle the problem directly where it lies. Examples include the dual-rail logic styles introduced in [54], or masked computations [11]. The main limitation of these solutions is the difficulty to control the design process, e.g. in order to balance the capacitances of rails in a logic style [22, 55], or to avoid detrimental effects such as glitches that may lead to easy-to-exploit leakages in masked implementations [31, 32]. Hence, software-based countermeasures bring a more flexible complement, e.g. exploiting time randomizations [16, 23] or data masking [39, 47], at the cost of possibly higher performance overheads. Both for hardware and software countermeasures, security evaluations usually reveal that attacks remain possible if high number of measurements are available (see, e.g. [43, 52] for the case of masking). On the other hand, protections can be more or less transparent to the global infrastructure. For example, the previously listed countermeasures aim to protect cryptographic algorithms, independent of the protocol using them. But in order to prevent the exploitation of side-channel leakages, it is ultimately useful to also limit the number of times a secret key is manipulated to encrypt with leakage. Modes of operation that ensure such a condition have been considered early after the publications of side-channel attacks, e.g. by Paul Kocher [40]. They are also at the cores of several recent leakage-resilient constructions, e.g. [41, 59]. Summarizing, the public literature contains a wide range of techniques for improving the security of cryptogaphic implementations that could apply to SIM cards. Yet, improving the understanding of their strength and weaknesses in order to obtain the best security with minimum performance overheads remains an important scope for further research.

5 Conclusions & future work

Technically, it is not a surprise that weakly protected chips can be defeated by side-channel attacks. Yet, our results exhibit (or recall) that such attacks are relatively easy to implement, and are certainly accessible to determined adversaries. Taking the example of SIM cards, this can have severe consequences for the security of GSM communications. Overall, the security of a system is always as strong as its weakest point. Hence, distributing cryptographically-enhanced chips without a sufficient care for physical security leads to unbalanced situations, as side-channel attacks may constitute a trapdoor to circumvent mathematical security. This is especially important for small embedded devices, for which physical access may sometimes be granted to adversaries. In this respect, it is more surprising that (somewhat) security sensitive applications do not always build on certified chips (following what is done, e.g. for bank cards). Admittedly, the target SIM cards investigated in this paper implement old versions of the GSM algorithms, in old technologies. Nevertheless, some of these cards are still in circulation and cards cloning is an important concern that could prevent the adoption of new services [20]. Hence, this situation illustrates the long term nature of hardware security issues. It provides a general motivation for considering them as an important element to take into account early in cryptographic developments. In this respect, we note that the use of proprietary algorithms in commercial products significantly slows down progresses in securing their implementation. In view of the implementation-specific nature of physical attacks, it frequently turns out that protection mechanisms that are tailored to certain cryptographic algorithms provide the best efficiency vs. security tradeoffs. For example, secure implementations of the AES have been the subject of a large literature over the last 10 years. By contrast, no similar analysis is available for COMP128-1. Worse, the use of large (e.g. 512-bit) tables makes it hardly suitable for implementation of countermeasures such as software masking [21]. Following this observation and in the long term, considering protections against physical attacks as a design criteria for cryptographic algorithms could be useful.

While resorting to certification would be an important step in improving the security of SIM cards (or other devices), we also note that the procedures used by evaluation laboratories could benefit from an improved transparency. That is, currently certified chips certainly rule out the possibility of simple attacks as we describe in this paper. But it remains that the exact security level they guarantee is opaque for the end-users, and this opaqueness generally increases as countermeasures (e.g. as listed in Section 4) are added to the chips. Proposals of worst-case security evaluations aiming at limiting the risks of a "false sense" of security could improve this situation [49, 51]. Considering the strongest available adversaries and taking advantage of the latest cryptanalytic progresses during evaluations of cryptographic hardware appears important in view of the difficulty to fix physical security breaches a posteriori. Eventually, better reflecting side-channel evaluation tools and methodologies in public standards would be highly beneficial too. In this respect, it is noticeable that the recent ISO 19790 draft standard on "security requirements for cryptographic modules" (aka. FIPS-140-3 [36]) leaves the entire section on non-invasive attack methods essentially optional to vendors, with little details about the exact evaluation procedures.

Additional scopes for further research naturally include the development of countermeasures against physical attacks under strong cost and performance constraints (e.g. for contactless smart cards and RFIDs). Other security protocols relying on implementations in embedded devices could be investigated too. Keeping the example of mobile communications, 3G networks and smart phones appear as natural targets. In the latter case, recent news suggest that the side-channel issue is not limited to SIM cards, and extends to cryptographic libraries inside the phones [37]. Hence, security partitioning appears as an important direction for improving the security of such embedded information systems.

Acknowledgements. Yu Yu was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61172085, 61061130540, 61073174, 61103221, 11061130539, 61021004 and 61133014. François-Xavier Standaert is an associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in part by the ERC project 280141 on CRyptographic Algorithms and Secure Hardware (acronym CRASH).

References

1. ANSSI. Agence nationale de la securite des systemes d'information, http://www.ssi.gouv.fr/en/products/certified-products/, retrieved on feb. 1, 2012.
2. Archambeau, C., Peeters, E., Standaert, F.-X., and Quisquater, J.-J. Template attacks in principal subspaces. In *CHES* (2006), L. Goubin and M. Matsui, Eds., vol. 4249 of *Lecture Notes in Computer Science*, Springer, pp. 1–14.
3. Barkan, E., Biham, E., and Keller, N. Instant ciphertext-only cryptanalysis of gsm encrypted communication. In *CRYPTO* (2003), D. Boneh, Ed., vol. 2729 of *Lecture Notes in Computer Science*, Springer, pp. 600–616.
4. Biham, E., and Dunkelman, O. Cryptanalysis of the a5/1 gsm stream cipher. In *INDOCRYPT* (2000), B. K. Roy and E. Okamoto, Eds., vol. 1977 of *Lecture Notes in Computer Science*, Springer, pp. 43–51.
5. Biryukov, A., Shamir, A., and Wagner, D. Real time cryptanalysis of a5/1 on a pc. In *FSE* (2000), B. Schneier, Ed., vol. 1978 of *Lecture Notes in Computer Science*, pp. 1–18.
6. Bogdanov, A., Eisenbarth, T., and Rupp, A. A hardware-assisted realtime attack on a5/2 without precomputations. In *CHES* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *LNCS*, Springer, pp. 394–412.
7. Boneh, D., DeMillo, R. A., and Lipton, R. J. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT* (1997), W. Fumy, Ed., vol. 1233 of *Lecture Notes in Computer Science*, Springer, pp. 37–51.
8. Briceno, M., Goldberg, I., and Wagner, D. GSM Cloning. `http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html`, 1998. Retrieved on Oct. 14, 2011.
9. Brier, E., Clavier, C., and Olivier, F. Correlation power analysis with a leakage model. In *CHES* (2004), M. Joye and J.-J. Quisquater, Eds., vol. 3156 of *Lecture Notes in Computer Science*, Springer, pp. 16–29.
10. BSI. Federal office for information security, https://www.bsi.bund.de/en/topics/certification/certification_node.html, retrieved on feb. 1, 2012.
11. Chari, S., Jutla, C., Rao, J. R., and Rohatgi, P. Towards sound approaches to counteract power analysis attacks. In Wiener [58], pp. 398–412.
12. Chari, S., Rao, J. R., and Rohatgi, P. Template attacks. In *CHES* (2002), B. S. K. Jr., Çetin Kaya Koç, and C. Paar, Eds., vol. 2523 of *Lecture Notes in Computer Science*, Springer, pp. 13–28.
13. CHES. Workshop on cryptographic hardware and embedded systems, http://www.chesworkshop.org/.
14. Clavier, C., Coron, J.-S., and Dabbous, N. Differential power analysis in the presence of hardware countermeasures. In *CHES* (2000), Çetin Kaya Koç and C. Paar, Eds., vol. 1965 of *LNCS*, Springer, pp. 252–263.
15. Common Criteria. http://www.commoncriteriaportal.org/. Retrieved on February 20, 2012.
16. Coron, J.-S., and Kizhvatov, I. Analysis and improvement of the random delay countermeasure of ches 2009. In Mangard and Standaert [33], pp. 95–109.
17. Doget, J., Prouff, E., Rivain, M., and Standaert, F.-X. Univariate side channel attacks and leakage modeling. *J. Cryptographic Engineering 1*, 2 (2011), 123–144.

18. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., and Shalmani, M. T. M. On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In *CRYPTO* (2008), D. Wagner, Ed., vol. 5157 of *Lecture Notes in Computer Science*, Springer, pp. 203–220.
19. EMVco. http://www.emvco.com/. Retrieved on April 11, 2012.
20. Extreme Tech. http://www.extremetech.com/mobile/105683-nfc-enabled-sim-cards-to-become-a-worldwide-standard. Retrieved on February 18, 2012.
21. Goubin, L., and Patarin, J. Des and differential power analysis (the "duplication" method). In *CHES* (1999), Çetin Kaya Koç and C. Paar, Eds., vol. 1717 of *Lecture Notes in Computer Science*, Springer, pp. 158–172.
22. Guilley, S., Hoogvorst, P., Mathieu, Y., and Pacalet, R. The "backend duplication" method. In Rao and Sunar [46], pp. 383–397.
23. Herbst, C., Oswald, E., and Mangard, S. An aes smart card implementation resistant to power analysis attacks. In *ACNS* (2006), J. Zhou, M. Yung, and F. Bao, Eds., vol. 3989 of *Lecture Notes in Computer Science*, pp. 239–252.
24. Hulton, D. Smart card security (by h1kari). DEFCON 2004, `http://www.defcon.org/html/links/dc-archives/dc-12-archive.html`. Retrieved on October 14, 2011.
25. Joux, A., Ed. *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings* (2009), vol. 5479 of *Lecture Notes in Computer Science*, Springer.
26. Kocher, P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology—CRYPTO '96* (18–22 Aug. 1996), N. Koblitz, Ed., vol. 1109 of *LNCS*, Springer-Verlag, pp. 104–113.
27. Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In Wiener [58], pp. 388–397.
28. Mangard, S. Hardware countermeasures against dpa ? a statistical analysis of their effectiveness. In *CT-RSA* (2004), T. Okamoto, Ed., vol. 2964 of *Lecture Notes in Computer Science*, Springer, pp. 222–235.
29. Mangard, S., Oswald, E., and Popp, T. *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.
30. Mangard, S., Oswald, E., and Standaert, F.-X. One for all – all for one: unifying standard differential power analysis attacks. *IET Information Security 5*, 2 (2011), 100–110.
31. Mangard, S., Popp, T., and Gammel, B. M. Side-channel leakage of masked cmos gates. In *CT-RSA* (2005), A. Menezes, Ed., vol. 3376 of *Lecture Notes in Computer Science*, Springer, pp. 351–365.
32. Mangard, S., Pramstaller, N., and Oswald, E. Successfully attacking masked aes hardware implementations. In Rao and Sunar [46], pp. 157–171.
33. Mangard, S., and Standaert, F.-X., Eds. *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings* (2010), vol. 6225 of *Lecture Notes in Computer Science*, Springer.
34. Maximov, A., Johansson, T., and Babbage, S. An improved correlation attack on a5/1. In *Selected Areas in Cryptography* (2004), H. Handschuh and M. A. Hasan, Eds., vol. 3357 of *Lecture Notes in Computer Science*, Springer, pp. 1–18.
35. Moradi, A., Barenghi, A., Kasper, T., and Paar, C. On the vulnerability of fpga bitstream encryption against power analysis attacks: extracting keys from xilinx virtex-ii fpgas. In *ACM CCS* (2011), Y. Chen, G. Danezis, and V. Shmatikov, Eds., ACM, pp. 111–124.

36. National Institute of Standards and Technologies. http://csrc.nist.gov/publications/PubsDrafts.html. Retrieved on March 25, 2012.
37. NetworkWorld. http://www.networkworld.com/news/2012/ 012612-rsa-crypto-keys-255379.html, retrieved on february 13, 2012.
38. Oswald, D., and Paar, C. Breaking mifare desfire mf3icd40: Power analysis and templates in the real world. In *CHES* (2011), B. Preneel and T. Takagi, Eds., vol. 6917 of *Lecture Notes in Computer Science*, Springer, pp. 207–222.
39. Oswald, E., and Schramm, K. An efficient masking scheme for aes software implementations. In *WISA* (2005), J. Song, T. Kwon, and M. Yung, Eds., vol. 3786 of *Lecture Notes in Computer Science*, Springer, pp. 292–305.
40. Paul Kocher. Leak resistant cryptographic indexed key update. US Patent 6539092.
41. Pietrzak, K. A leakage-resilient mode of operation. In Joux [25], pp. 462–482.
42. Preneel, B. The cryptographic year in review. isse 2011 keynote talk, available from http://homes.esat.kuleuven.be/preneel/preneel_isse11v1.pdf.
43. Prouff, E., Rivain, M., and Bevan, R. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers 58*, 6 (2009), 799–811.
44. Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Smart Card Programming and Security (E-smart 2001) Cannes, France* (Sept. 2001), vol. 2140 of *LNCS*, pp. 200–210.
45. Rao, J. R., Rohatgi, P., Scherzer, H., and Tinguely, S. Partitioning attacks: Or how to rapidly clone some gsm cards. In *IEEE Symposium on Security and Privacy* (2002), pp. 31–44.
46. Rao, J. R., and Sunar, B., Eds. *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings* (2005), vol. 3659 of *LNCS*, Springer.
47. Rivain, M., and Prouff, E. Provably secure higher-order masking of aes. In Mangard and Standaert [33], pp. 413–427.
48. Schindler, W., Lemke, K., and Paar, C. A stochastic model for differential side channel cryptanalysis. In Rao and Sunar [46], pp. 30–46.
49. Standaert, F.-X. Some hints on the evaluation metrics & tools for side-channel attacks. proceedings of the nist non-invasive attacks testing workshop, nara, japan, september 2011.
50. Standaert, F.-X., and Archambeau, C. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *CHES* (2008), E. Oswald and P. Rohatgi, Eds., vol. 5154 of *Lecture Notes in Computer Science*, Springer, pp. 411–425.
51. Standaert, F.-X., Malkin, T., and Yung, M. A unified framework for the analysis of side-channel key recovery attacks. In Joux [25], pp. 443–461.
52. Standaert, F.-X., Veyrat-Charvillon, N., Oswald, E., Gierlichs, B., Medwed, M., Kasper, M., and Mangard, S. The world is not enough: Another look on second-order dpa. In *ASIACRYPT* (2010), M. Abe, Ed., vol. 6477 of *Lecture Notes in Computer Science*, Springer, pp. 112–129.
53. The Global mobile Suppliers Association (GSA). GSM Market Share. `http://www.gsacom.com/gsm_3g/market_update.php4`, March 2011. Retrieved on October 14, 2011.
54. Tiri, K., and Verbauwhede, I. Securing encryption algorithms against dpa at the logic level: Next generation smart card technology. In *CHES* (2003), C. D. Walter, Çetin Kaya Koç, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer, pp. 125–136.

55. Tiri, K., and Verbauwhede, I. Place and route for secure standard cell design. In *CARDIS* (2004), J.-J. Quisquater, P. Paradinas, Y. Deswarte, and A. A. E. Kalam, Eds., Kluwer, pp. 143–158.
56. van Woudenberg, J. G. J., Witteman, M. F., and Bakker, B. Improving differential power analysis by elastic alignment. In *CT-RSA* (2011), A. Kiayias, Ed., vol. 6558 of *Lecture Notes in Computer Science*, Springer, pp. 104–119.
57. Veyrat-Charvillon, N., Gerard, B., Renauld, M., and Standaert, F.-X. An optimal key enumeration algorithm and its application to side-channel attacks. Cryptology ePrint Archive, Report 2011/610, 2011. `http://eprint.iacr.org/`.
58. Wiener, M., Ed. *Advances in Cryptology—CRYPTO '99* (15–19 Aug. 1999), vol. 1666 of *LNCS*, Springer-Verlag.
59. Yu, Y., Standaert, F.-X., Pereira, O., and Yung, M. Practical leakage-resilient pseudorandom generators. In *ACM Conference on Computer and Communications Security* (2010), E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds., ACM, pp. 141–151.

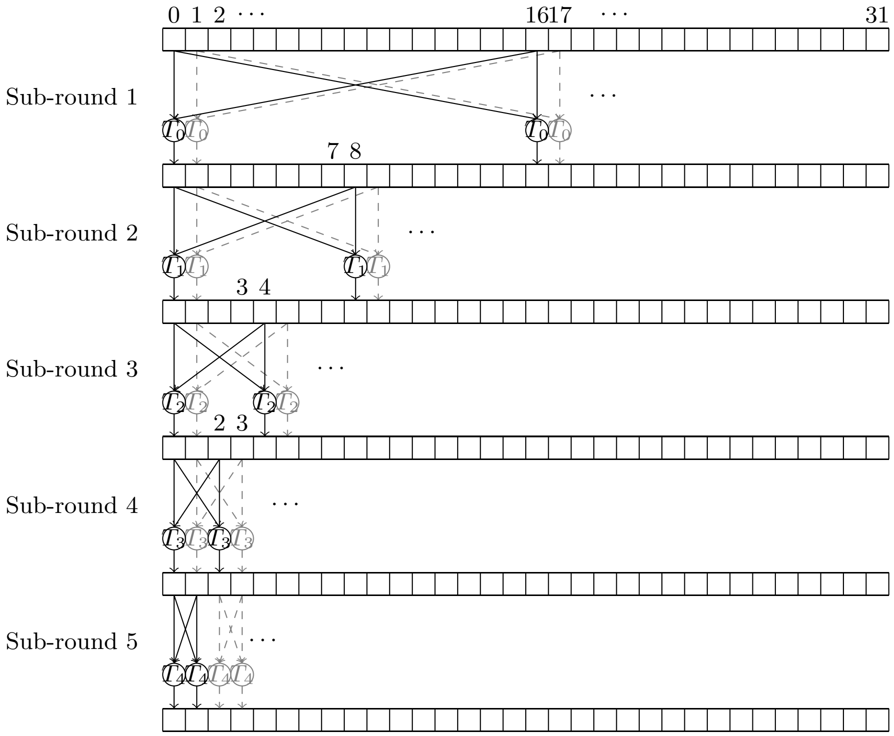

Fig. 7. Butterfly structure of COMP128-1's compressing function.

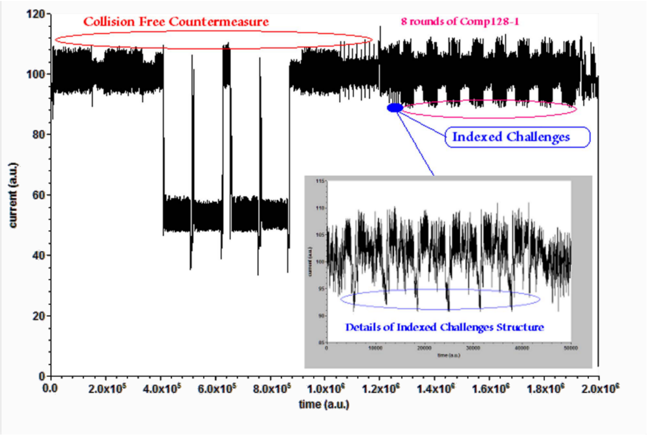

Fig. 8. A power trace from SIM#4.

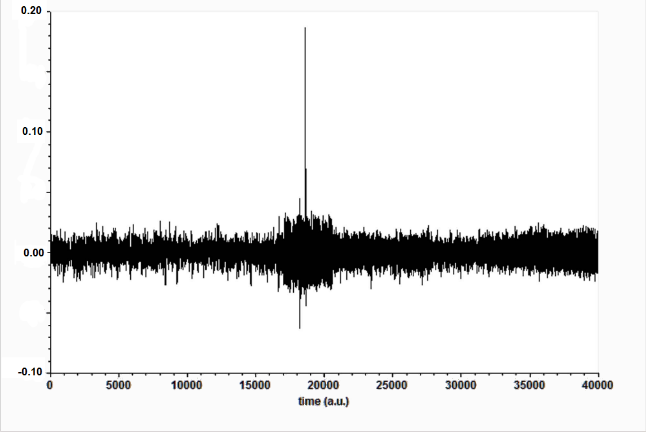

Fig. 9. DPA result against SIM#1.

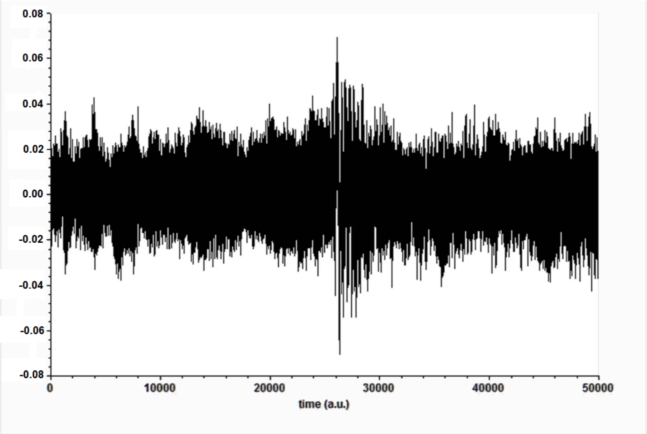

Fig. 10. DPA result against SIM#4.