# Defeating ISO9797-1 MAC Algo 3 by Combining Side-Channel and Brute Force Techniques

Benoit Feix and Hugues Thiebeauld

Underwriters Laboratories, UK Security Lab
Basingstoke, England.
firstname.familyname@ul.com

**Abstract.** Side-channel analysis is a well-known and efficient hardware technique to recover embedded secrets in microprocessors. Over the past years, the state-of-the-art side-channel attacks has significantly increased, leading to a myriad of vulnerability paths that secure codes must withstand. Nowadays most of the attacks target the cryptographic algorithms, but very few exploit the cryptographic protocol. In this paper, we present a new attack that exploits the information exchange at the cryptographic protocol level in order to disclose the secret key. This attack is applicable to the MAC calculations standardized in ISO/IEC 9797-1 especially the MAC algorithm 3 with the DES function. This protocol is spread in secure products nowadays, this is the case typically for some EMV implementations. By using a side-channel technique combined with a reasonable brute force effort, we show that the secret key can be fully retrieved even though the DES implementation seems to be well-protected against side-channel attacks.

**Keywords:** side-channel analysis, DES, MAC ISO/IEC 9797-1, exhaustive search.

## 1 Introduction

Nowadays cryptographic algorithms and protocols are at the heart of billions of electronic devices security. Information stored in embedded devices like secure elements and smartphones are potentially exposed to numerous attacks if the cryptographic algorithms are not thoroughly protected.

One well-known cryptographic protocol is the ISO 9797-1 MAC algorithm 3. It is widely spread over the world and provides a triple DES security for cryptographic operations in many security systems.

As a key element of the system security, the ISO9797-1 MAC algorithm 3 is mostly executed in secure hardware, called secure elements. They can be found in different form factors, such as payment cards, USIM devices or as an internal component of the mobile phone handset for embedded secure elements. The aim of the secure element is to combine a strong software and hardware security for a cost-effective solution. Such a device is designed to withstand fraudulent access or secret disclosure.

To achieve that, the secure device must embed a blend of protection against numbers of threat. One of this threat concerns the physical attacks. These techniques aim at taking advantage of the physical execution, that could result to malevolent secret extraction or to severe security downgrades of the payment transactions.

In the context of this paper, the spotlight will be focused on the side-channel technique. This technique introduced by Kocher et al. [8,10] exploits the physical leakage during a sensitive code execution with the aim to disclose secrets manipulated during the processing of a cryptographic operation. Side-channel analysis has shown a high efficiency to exhibit secret cryptographic keys when targeting not well-protected cryptographic implementations.

In the last two decades many scientific publications have introduced and improved side-channel attacks. It concerns all cryptographic algorithms that are being implemented in embedded devices. Numerous countermeasures have also been presented.

The attack introduced in this paper is focused on the ISO9797-1 MAC algorithm 3 [7] in the particular mode when the master key remains identical for all computations. For instance some implementations of the EMV standard fall into this case when others are not concerned by this attack as the master key is derived prior to the computation. It is dependent upon the way the EMV standard is implemented.

The paper is organized as follows. Section 2 reminds the structure of the ISO9797-1 MAC algorithm 3. The necessary knowledge and background on side-channel analysis to understand the new attack is presented. In Section 3 we describe our new attack to recover the secret key used for the MAC computation. We present practical results we have obtained in section 4. Section 5 is a discussion of the classical countermeasures and their efficiency to prevent our attack. Finally a conclusion is given in section 6.

## 2 Preliminaries

### 2.1 Background on ISO 9797-1 MAC Algo 3

The MAC Algorithm implementing the ISO/IEC-9797-1 standard is a CBC MAC calculation using a simple DES operation to process the blocks of message and ends up with a Triple DES computation using a 112 bits secret key. This algorithm represents a good trade-off to combine a cost-effective algorithm with the proven security of the Triple DES. Indeed, the known vulnerability of the simple DES algorithm to the brute force attacks is covered by the final Triple DES. Figure 1 describes the MAC algorithm 3 when the encryption algorithm selected is the standard DES.

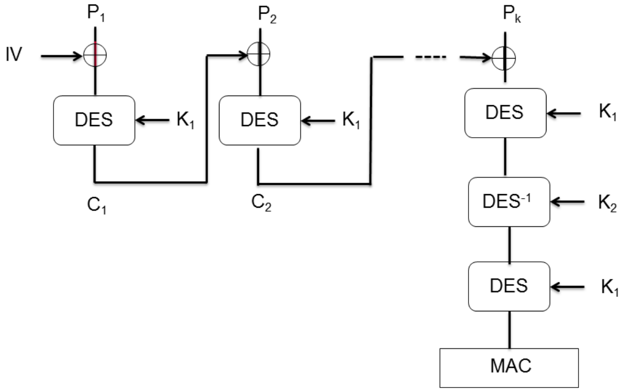

**Fig. 1.** ISO 9797-1 MAC Algo 3 Description with DES cipher

We define the following notation to be used in the rest of the paper. Let us denote $i$ an occurrence of the MAC computation on the $k$ 8-byte blocks input message $P_{i,1}P_{i,2}\dots P_{i,k}$. Let $\mathrm{MAC}_i$ be the result value CBC-MAC-Algo3($P_{i,1}P_{i,2}\dots P_{i,k}$).

$$
\begin{aligned}
&IV = 0\\
&C_{i,1} = \mathrm{DES}_{k_1}(P_{i,1} \oplus IV)^1\\
&C_{i,2} = \mathrm{DES}_{k_1}(P_{i,2} \oplus C_{i,1})\\
&\dots\\
&C_{i,k-1} = \mathrm{DES}_{k_1}(P_{i,k-1} \oplus C_{i,k-2})\\
&MAC_i = C_{i,k} = \text{3-DES}_{k_1,K_2}(P_{i,k} \oplus C_{i,k-1})
\end{aligned}
$$

For the sake of simplicity, in this paper we denote by transaction an operation using the MAC Algo 3.

### 2.2 Side-channel attack background

**Side-channel analysis** has been studied for years since it has been introduced by Kocher et al. [9]. Many attack paths have been published

[1] $\oplus$ represents the bitwise exclusive OR operation

on the different cryptosystems like DES and RSA which are widely used in the majority of the embedded devices like Banking or Identity products. In the same time many statistical attack techniques have improved the original Differential Side-Channel Analysis (DSCA) (ie. Difference of Mean - DoM) from Kocher et al. [11]. We can for instance mention the Correlation Side-Channel Analysis (CSCA) introduced by Brier et al. [2], the Mutual Information Side-Channel Analysis (MISCA) from Gierlichs et al. [5] or the Linear Regression Side-Channel Analysis [4,17].

**Correlation side-channel analysis** relies upon a linear leakage model in the Hamming weight of a sensitive manipulated data:

$$W = a \cdot HW(D) + b + \epsilon \tag{1}$$

where $a$ and $b$ are real values characteristic of the hardware targeted and $\epsilon$ is a white gaussian of mean 0 and standard deviation $\sigma$. In order to measure the dependency between the estimated value of a sensitive data and the corresponding value manipulated and represented in the physical traces measurement, the linear correlation factor from Bravais-Pearson is classically used.

Let $\mathcal{C}^{(i)}$ with $1 \leqslant i \leqslant \ell$ a set of $\ell$ side-channel traces captured from a device processing the targeted computations with input value $X^{(i)}$ whose processing occurs at time sample $t$ with $l$ the number of points acquired at time sample $t$. We consider $\Theta_0 = \{\mathcal{C}^1(t), \ldots, \mathcal{C}^\ell(t)\}$. We denote $S^{(i)}$ with $1 \leqslant i \leqslant \ell$ a set of $\ell$ guessed intermediate sensible values based on a power model, which is generally linear in the Hamming weight of the data. Let $f(X^{(i)}, \hat{K})$ be a function of the input value $X^{(i)}$ and (a part of) the targeted guessed secret $\hat{K}$. All $l$ points in the leakage trace are equal to this value $f(X^{(i)}, \hat{K})$ for the time sample $t$. We then consider $\Theta_1 = \{S^{(1)}, \ldots, S^{(\ell)}\}$. The objective is to evaluate the dependency between both sets $\Theta_0$ and $\Theta_1$ by using the linear correlation factor $\rho_{\Theta_0,\Theta_1}$.

$$\begin{aligned}
\rho_{\Theta_0,\Theta_1} &= \frac{\mathrm{Cov}(\Theta_0, \Theta_1)}{\sigma_{\Theta_0}\sigma_{\Theta_1}} \\
&= \frac{\ell \sum (\mathcal{C}^{(i)}(t) \cdot S^{(i)}) - \sum \mathcal{C}^{(i)}(t) \sum S^{(i)}}{\sqrt{\ell \sum (\mathcal{C}^{(i)}(t))^2 - (\sum \mathcal{C}^{(i)}(t))^2}\sqrt{\ell \sum (S^{(i)})^2 - (\sum S^{(i)})^2}},
\end{aligned}$$

where summations are taken over $1 \leqslant i \leqslant \ell$.

The correlation value between both series is equal to 1 when the simulated model perfectly matches with the measured power traces. It then indicates that the guess on the secret corresponds to the correct key value handled by the device in the computations.

These different side-channel techniques (i.e. DSCA and CSCA) may compromise an implementation of the ISO9797-1 MAC Algo 3 protocol. State of the art attack paths are focused on the DES algorithm executions by applying classical DSCA on DES intermediate computation values. They aim at exhibiting directly some parts of the secret master key. A first attack path may target the first block processed. This attack requires the knowledge of the 8 byte-long input message value. As a result of such a successful attack one of the first round key can be retrieved. Similarly a correlation attack may target the final Triple DES computation. This requires the knowledge of the cryptogram value that can be straightforwardly obtained from the collection of the MAC computations output. However it is not possible to have the control of the ciphertexts, and consequently to run a chosen ciphertext attack unlike the first attack path that can be designed using chosen plain texts.

Nowadays it is worth mentioning that most of the recent devices are well secured against these attacks thanks to the numerous countermeasures proposed in the literature or patented by card makers like masking techniques [1,11,15,16] or hardware countermeasures [3,6].

However, despite a good expertise to tackle the side-channel threat in the DES implementations, we demonstrate in the following that the protections shall not be confined in the cryptographic layer only. They shall be extended to the cryptographic protocol level as well. The attack introduced in this paper is a good illustration of this. In the ISO9797-1 MAC algorithm 3 context we show that the secret key can be exposed by defeating the cryptogram implementation whereas the DES implementation has been proven secure against state of the art side-channel techniques.

## 3 A New Side-Channel Attack on MAC Algo 3

The attack introduced in this paper targets a 112 bit-length master key when it is being used for a MAC computation implementing ISO 9797-1. As a result of this, the whole master key may be disclosed providing an effort of brute-force. In the following, the different steps for implementing this attack are being described.

### 3.1 Attack Pre-conditions

The first step of the attack targets the recovery of one couple of value (plaintext P, ciphertext C) from a single DES computation value amongst a set of $\ell$ executions of the ISO9797-1 MAC algorithm 3 with the attacked hardware device.

In the following to extract the ciphertext C, the attacker requires the input message is structured in a particular form. The first 8 byte-long block value $P_{i,1}$ must be set to a constant value $P$ for $\ell$ MAC computation $MAC_1, \ldots, MAC_\ell$. The constant value in input shall also be the same for all ISO9797-1 MAC algorithm 3 computations considered for the attack. However the choice of the value itself does not matter. As a result the ciphertext values $C_{i,1}$ of the first DES calculation using the 56 bit-long key $K_1$ remains identical across all MAC calculations. We denote this value in the following $C = DES(P, K_1)$.

Another condition concerns the second block $P_{i,2}$, that must be chosen randomly or at least with a high entropy across the transaction set. The values itself does not matter.

It is worth noticing that the previous conditions in the data format can be easily extended. Indeed, the attack can target any block of simple DES computation within the MAC. For attacking the $k^{th}$ block of the MAC, the conditions become to have ($k$-1) 8-byte long input blocks fixed to the same value and the subsequent block chosen as a random. In that context, the side-channel technique will target the ciphertext value resulting from the ($k$-1) first blocks DES CBC encryptions.

### 3.2 Side-channel Attack: Recovering a Single DES Ciphertext Value

With a secret ciphertext $C_{i,1} = C$ for all $i$ and a set of random inputs $\{P_{1,2}, \ldots, P_{\ell,2}\}$ as second message input blocks, a side-channel technique can be performed in order to exhibit $C$. This requires to run a significant amount of transactions with the same device using the same secret key. The number of transactions necessary to succeed the attack will mostly depend on the way the hardware is behaving and on the level of countermeasures implemented. Typically several dozens of thousands transactions will be necessary at least.

For each execution $i$ a physical trace $T^{(i)}$ needs to be collected when the cryptogram computations is being processed, more precisely when the secret $C$ is being processed to feed the next block of computation. This can be achieved by measuring the power fluctuations or the electromagnetic radiations.

The side-channel will target the processing when the secret ciphertext $C$ is being XORed with the input $P_{i,2}$. As per the pre-condition defined for this attack, the input $P_{i,2}$ is a random value known or controlled by the attacker. As a result, it is possible to guess the value $D_{i,2} = C_{i,1} \oplus P_{i,2}$ from the physical traces. It is then possible to exhibit the 8-byte long secret $C$ with side-channel technique. The same value can leak at different timing, either when the XOR operation is being executed or when the next block input is being loaded for the next DES operation.

The data extraction does not require any specific technique in side-channel, as a classical correlation analysis or even a differential analysis may turn out to be very efficient. The factor of success is the level of hardware or software countermeasures that would not lead to any exploitable leakage in a reasonable amount of traces. The maximum amount of traces that a device can execute is tied with the application and the corresponding configuration.

The divide-and-conquer approach can be applied by guessing $C_{i,1}$ byte per byte in order to limit the computation efforts. In this case the attacker is performing CSCA on each of the 256 guesses for the targeted byte. He computes the correlation factor $\rho_{\Theta_0,\Theta_1}$ with $\Theta_0 = \{T^{(1)}, \ldots, T^{(\ell-1)}\}$ and $\Theta_1 = \{HW((D_{i,2})_j), \ldots, HW((D_{\ell,2})_j)\}$ for each byte $1 < j < 8$ of the ciphertext $C$. The highest correlation value indicates the associated guess is the right ciphertext byte value.

**Correlation Analysis Result Interpretation** When considering the standard Hamming weight leakage model the correlation attack described previously is leading theoretically to two possible secret bytes and not only one. Indeed as the leaking operation is a XOR for any secret byte both $K$ and the value $K \oplus FFh$ are leading to the same correlation factor value but with opposite signs. We can assume that it is not possible to do any straightforward guess upfront to determine which sign corresponds the correlation with the right value. As a result, in this leakage model the complete correlation analysis is then leading to two possible 8-bytes ciphertexts: $C = ((C)_0, \ldots, (C)_7)$ and $\overline{C} = ((C)_0 \oplus FFh, \ldots, (C)_7 \oplus FFh)$.

This assumption has been verified with the following tests. Relying on the linear leakage model given in section 2.2 we designed simulation traces for the MAC Algo3 operation for all possible $C_0$ byte value $(0, \ldots, 255)$ and a random set of values $\{P_{1,2}, \ldots, P_{\ell,2}\}$. We then observed that for each case that the highest correlation values were obtained with $(C)_0$ and $\overline{(C)}_0$. Test results are given as examples in figures 2 and 3 where correlation values for $(C)_0$ are plotted in black color, those for $\overline{(C)}_0$ in red and other 254 cases in grey. Hence it has been demonstrated that only two ciphertext values are obtained at the end of the attack.

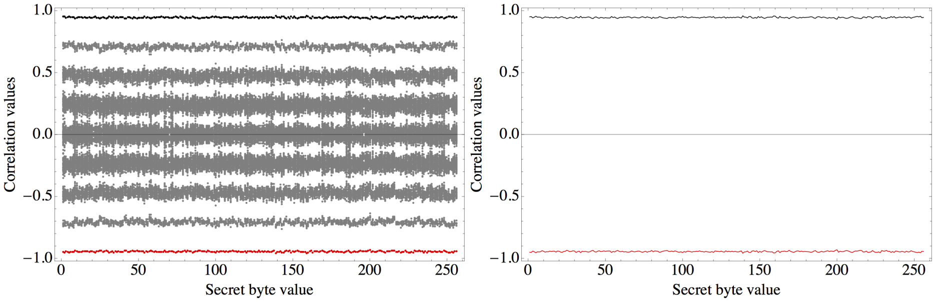

**Fig. 2.** Test example for a = 1, b = -1.026, noise $\sigma$ = -0.490 and $\ell$ = 500

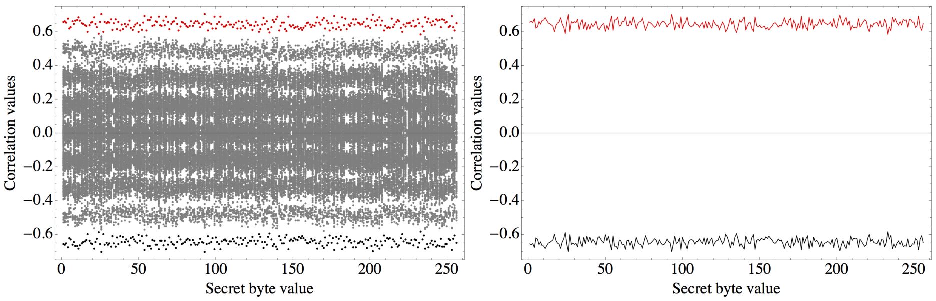

**Fig. 3.** Test example for a = -0.434, b = 0.689, noise $\sigma$ = 0.729 and $\ell$ = 500

Obviously different leakage models (or if HW contribution in the leakage model is very low) could lead to different interpretation. However all the observations so far on real hardware confirmed the validity of assuming such a Hamming weight model without being able to guess the sign upfront.

Therefore this leads to an uncertainty between two potential values in that model.

### 3.3 Brute-force: Recovering the Secret Key

Once the value $C_{i,1}$ is retrieved, the 56 bit-long secret key $K_1$ can be disclosed by the mean of a brute-force with the knowledge of the couple ($P_{i,1}$,$C_{i,1}$). As demonstrated in many recent publications [12,13,14], modern albeit affordable equipment can be easily purchased to crack simple DES keys. Indeed, whereas the time processing remains tedious with classical computers, the brute force attack can be efficiently implemented using specific hardware. Such a device can be designed to run parallel computation or even been commercially available at a reasonable cost, around 10 thousand dollars. The most famous example is the Copacobana [12] hardware that was proven to crack a simple DES in few days several years ago.

More recently, K. Nohl in BlackHat conference [13] illustrated how rainbow techniques from Oechslin [14] can efficiently decrease the complexity for cracking simple DES and consequently significantly improve the brute-force efficiency. With an excellent success rate, it was shown that the cracking time for a simple DES key could be retrieved in less than 1 minute, providing an investment of 1500 dollars and a large pre computing time of 1 year approximately. Using rainbow tables require the attacker can select the plaintext value that is often possible. Assuming that the outcome of the precomputation is available publicly, on the web typically, retrieving the key in less than 1 minute appears extremely efficient. This is only possible when the format of the data is defined, which can be the case for most of the standards using the MAC Algo3.

Ideally, the global brute-force effort includes four consecutive simple DES cracks for retrieving respectively $K_1$ and $K_2$ (ie. two each possible ciphertext $C$ and $\overline{C}$ recovered with DSCA). Hence the cumulative effort remains very reasonable and affordable for lots of parties nowadays, like agencies, universities or illegal organisations. Practically, if the whole $C_{i,1}$ cannot be disclosed, the brute-force complexity grows together with the missing information. Even though the computation time will be increased significantly, it shall remain affordable within certain boundaries. These boundaries follow the state of the art developments in parallel computing and dedicated hardware.

## 4 Practical Results

Several practical experimentations have been conducted to ascertain the attack feasibility on different real devices.

**Targeted Implementations** First attack has been done on simulated power traces relying on a standard Hamming weigth leakage model with a white Gaussian noise as described previously. Several noise deviation values were applied in order to verify that the attack still remains efficient even in the case of a high noise added to the signal. In all cases ciphertexts $C$ and $\overline{C}$ were recovered amongst the side-channel traces with correlation analysis. In this ideal context, the right value correlated with a positive sign and could subsequently be easily discriminated.

In a second phase we performed the attack on different hardware devices. First device was a ISO9797-1 MAC algorithm 3 implementation using a FPGA based implementation of the DES operation with the aim to validate our assumption about the leakage model. We also tested the attack on 8 and 32-bit cores implementations with hardware accelerators. We obtained interesting attack results as detailed in the following.

**Equipment** To implement this attack in a straightforward configuration, the equipment remains relatively basic. Indeed, it requires an efficient way to run a sequence of consecutive ISO9797-1 MAC algorithm 3 encryptions. A smartcard reader and the corresponding piece of software playing the command including the targeted MAC computation would typically be enough for the analysis. Besides, the attack requires the use of a good equipment for the measurement. In a classical setup configuration a good digital oscilloscope only is needed with a decent bandwidth and some good probes to measure the power fluctuations or the EM radiations as shown in figure 4. All in all, the total cost for a basic setup to exhibit the vulnerability never exceeds few dozens of thousands of dollars. To go to the exploitation, the brute force may need more specialised equipment whereas it remains affordable. We do think that this attack could be potentially carried out in a garage by anyone with a minimum access of lab equipment and a good hands-on experience in the field.

**Results Analysis** Our practical testing on real devices showed that the main factor of success of this attack relies on the capacity to successfully run the side-channel part. To achieve this, the trace collection must be successfully performed with measurements carrying information that can be subsequently exploited. The quality of the side-channel traces requires most of the times a close access to the device, or at least the access to the physical resources such as the power lines. When this remains relatively straightforward for a smartcard device, it could turn out to be more deli-

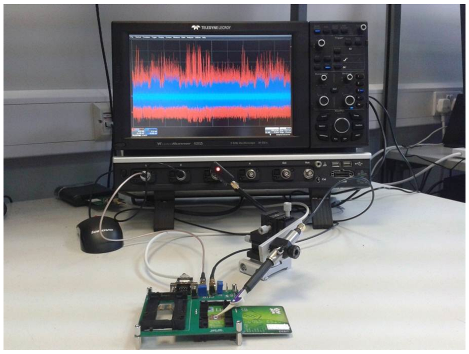

**Fig. 4.** Side-channel measurement bench exemple

cate for mobile handset or any device with a complex architecture formed of several hardware elements. On the other hand, the quality of the trace is heavily tied with the hardware behaviour and the level of embedded software countermeasures implemented in the device. Indeed, as for most of side-channel attacks, a thorough alignment of the set of traces will be necessary for being in position to disclose the secret information. Our practical testing showed that the attack realisation on modern devices was rarely straightforward as internal protections changing randomly the processing time were efficiently implemented, like jitters or random codes.

In order to overcome the difficulties to align together the set of traces, advanced techniques have shown a good efficiency. A first technique can be applied at a the measurement level by using smart triggers engines and filtering. The aim is to stick the measurements to a noticeable event and consequently to reduce the effects of random time execution in some cases. Such engines require some more specialised equipment and an expertise to set up. In a second stage, the rest of the work can be performed during the post processing of the collected traces. This work represents time and requires a good expertise in the field. Even though such countermeasures made the attack more complicated to perform, our experimentations showed that the device can still be subject to the attack when the information is available in the physical traces.

Lastly it is worth mentioning that a parameter can significantly reduce the effectiveness of the attack. Indeed, when the number of MAC computations using the same master key is limited, by a secure counter typically, this mechanically reduces the size of the set of collected traces.

Even though the same set of traces can be used for addressing the 8 bytes of the secret, we experienced that it obviously represents a strong restriction when the level of hardware and software countermeasure is high. When we managed to extract the secret, it was most of the time at the price of a good expertise in the field.

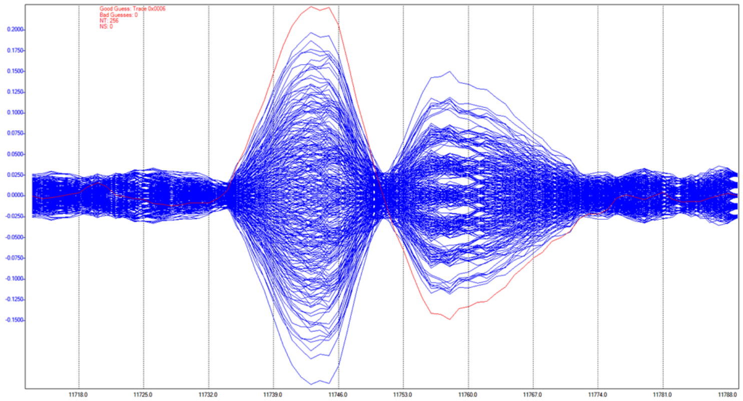

**Fig. 5.** Successful cryptogram byte recovery with CSCA

## 5 Protection and Countermeasures

As a consequence of the main findings of the practical experimentations, the shape of an efficient way to protect products against this attack can be guessed.

The first model will make sure that the data (ciphertext) information contained in the traces is not workable. This would mean that the intermediate information between each block of DES operations does not appear in plain and that no bias can be exploited by statistical techniques. Well implemented, this logical protection appears a very efficient way to tackle this attack. Developers must however keep in mind that classical higher order attack techniques could be used to defeat such a countermeasure if several values or masks could be combined in the side-channel trace.

A second way of protection can make use of mechanisms to hide the information in the side-channel traces rendering the signal no longer exploitable to the attacker. Hence the hardware traces would appear too noisy or impossible to properly align together. This would certainly require a minimum of testing to gauge the effectiveness of such implementation. Indeed, the physical behaviour of real devices may reserve some surprises.

Capping the number of MAC computations appears a temporary solution to limit the risk on some existing products that could be threatened by this attack. Obviously it shall not be considered as an ultimate protection. It only offers the insurance that the attack capability will remain tighten for all products implementing the specifications regardless their implementation. The maximum value for such a limit counter shall be chosen as a trade-off between the product capabilities and the need of security.

Regarding the nature of the attack, all implemented-based protection shall go through a minimum of validation relying on real product testing.

Finally the ideal solution consists in using session keys instead of master key when computing ISO9797-1 MAC algorithm 3 cryptograms.

## 6 Conclusion

This article introduces a new attack on MAC algorithm implementing ISO/IEC-9797-1 specification. Whereas it concerns only products implementing the MAC Algorithm 3 protocol using a fixed master key, this attack remains relevant for number of products that are present in our day to day life. By combining a side-channel technique followed with a brute force, this new technique could lead to the exposure of the whole master key resulting potentially to a severe breach in the system. As an additional reason to consider this threat, this attack only needs a classical equipment and remains in the "garage attack" category when applied in the basic form. As a good learning point it showed that the security shall not be confined in the cryptographic algorithms only but shall be extended to the cryptographic protocol as well. This paper introduced ways to efficiently thwart this attack at the product configuration and the implementation level. However a product can be deemed secure when it passed successfully a thorough practical validation on real devices.

### Acknowledgements

The authors would like to thank the UL crypto team: Arnaud Dambra, Andjy Ricart, Loic Thierry and Lucille Tordella for the fruitful discussions and the experiments on this subject.

## References

1. Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
2. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
3. D. Canright and L. Batina. A Very Compact "Perfectly Masked" S-Box for AES. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 446–459, 2008.
4. J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert. Univariate side channel attacks and leakage modeling. *IACR Cryptology ePrint Archive*, 2011:302, 2011.
5. B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual Information Analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
6. S. Guilley, L. Sauvage, J-L. Danger, T. Graba, and Y. Mathieu. Evaluation of power-constant dual-rail logic as a protection of cryptographic applications in fpgas. In *SSIRI*, pages 16–23. IEEE Computer Society, 2008.
7. ISO/IEC. Information technology - security techniques - message authentication codes (macs). *ISO/IEC Standards*, 1999.
8. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
9. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
10. P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
11. P.C. Kocher, J.M. Jaffe, and B.C. June. DES and Other Cryptographic Processes with Leak Minimization for Smartcards and other CryptoSystems. *US Patent 6,278,783*, 1998.
12. S-S. Kumar, C.Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler. Breaking with copacobana - a cost-optimized parallel code breaker. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 101–118. Springer, 2006.
13. Karsten Nohl. Rooting sim cards. *Black Hat Conference*, 2013.
14. Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer, 2003.
15. E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen. A Side-Channel Analysis Resistant Description of the AES S-Box. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 413–423. Springer, 2005.

16. M. Rivain and E. Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
17. W. Schindler, K. Lemke, and C. Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.