Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information and Computation 204 (2006) 1179–1193

Information and Computation

www.elsevier.com/locate/ic

On two DES implementations secure against differential power analysis in smart-cards ☆

Jiqiang Lv

National Key Lab of ISN, Xidian University, Xi'an City, Shaanxi Province 710071, China

Received 17 December 2004; revised 14 February 2006
Available online 5 June 2006

Abstract

Masking is one of the efficient and easily implemented countermeasures to protect cryptographic algorithms in such resource limited environments as smart-cards from differential power analysis as well as simple power analysis that were first introduced by Kocher et al. in 1999. To defend differential power analysis attacks, Akkar and Giraud presented a Transformed Masking Method and applied it to DES implementation in 2001. Unfortunately, in 2003, Akkar and Goubin showed a superposition attack that actually is a high-order differential power analysis attack on Akkar and Giraud's DES implementation using Transformed Masking Method, and finally they presented a DES implementation using their proposed Unique Masking Method to defend any order differential power analysis attacks, which was later improved by Akkar, Bévan and Goubin in 2004. In this paper, by exploiting a new artifice to classify the electric consumption curves, we show that Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method is still vulnerable to a high-order differential power analysis attack. Besides, we find it is also vulnerable to a superposition attack. We also present four new differential power analysis attacks on Akkar and Giraud's DES implementation using Transformed Masking Method.

Keywords: Data encryption standard (DES); Smart cards; Power analysis; Boolean masking

☆ The work was done when the author was with ONETS Wireless and Internet Security Co. Ltd. (CHINA).

E-mail address: lvjiqiang@hotmail.com.

0890-5401/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2006.04.002

1. Introduction

Lucifer, a block cipher designed by IBM in 1971, was selected as the Data Encryption Standard (DES) [13] by NIST [27] in 1977. Since then, DES has been extensively adopted to protect the privacy of the users and the transaction data in a large number of security service applications, say e-commerce, financial service and smart-cards. In the meantime, it has always been the target of many cryptanalysts and there have been quite a lot cryptanalytic results on it during the past nearly thirty years, of which differential cryptanalysis [4] and linear cryptanalysis [22] are two most well-known attacks. These attacks exploit mathematically statistical characteristics between a cipher's inputs and outputs, especially between inputs and outputs of its S-boxes, but do not take its software or hardware implementations into consideration.

However, electronic components are not usually perfectly tamperproof, and they may leak certain sensitive information on the algorithm from some side channels when an embedded cryptographic algorithm is executed, such as the timing of algorithm operations, power consumption and etc. In 1996, by carefully measuring the amount of time required to perform private key operations, Kocher [17] succeeded to exhibit the first side-channel attack that might find fixed Diffie–Hellman exponents [12], factor RSA keys [31], and break other cryptosystems. In 1997, Boneh et al. [7] presented another kind of side channel attack—fault analysis, which relies on the fact that hardware faults and errors that occur during the operations of a cryptographic device might leak information about the private key. Subsequently, by combining differential cryptanalysis and fault analysis, Biham and Shamir [6] presented a differential fault analysis attack, which is also applicable to secret key cryptosystems, for example, DES. In 1998, Kocher et al. [18] introduced a new kind of side channel attack—power analysis that includes simple power analysis (SPA) and differential power analysis (DPA), and they published them [19] in 1999. Power analysis starts from the fact that the attacker can get much more information than the knowledge of the inputs and the outputs during the execution of the algorithm, such as the electric consumption or electromagnetic radiations of the circuit devices, then tries to extract information about the secret key of a cryptographic algorithm by studying the power consumption of the electronic devices during the execution of the algorithm. Its initial focus was on DES, but soon was extended to other symmetrical cryptosystems and some public key cryptosystems, such as Advanced Encryption Standard (AES) candidates [8,15,23].

To secure cryptographic algorithms against DPA attacks, two main countermeasure categories have been presented so far. The first is the splitting method due to Goubin and Patarin [15] and Char et al. [9], which consists in "splitting" all the intermediate variables using some secret sharing principle; The second is the boolean masking method due to Messerges [25], which "masks" all the intermediate data if all the fundamental operations used in a given algorithm can be rewritten with masked input data, giving masked output data. A drawback of the splitting method is that it greatly increases the computation time and the memory required, which is a weakness in some constrained environments such as smart-cards, while the masking method is easy and efficient to be implemented in some algorithms, for example DES, and it has received extensive research [10,11,16]. In 2001, to counteract the DPA attack, Akkar and Giraud [1] presented a Transformed Masking Method and applied it to DES implementation. The main idea of this masking method is to perform all the computation such that all the data are XORed with a random mask. Moreover, the S-Boxes are modified such

that the output of a round is masked by the same mask as the input. Both the two main methods have been proven to be secure against the initial DPA attacks, and are now widely used in real-world implementations of many algorithms. Unfortunately, they do not take into consideration more elaborated attacks, called "High-Order DPA" [19,24,26], which consist in studying correlations between the secret data and several points of the electric consumption curves. In 2003, Akkar and Goubin [2] showed that Akkar and Giraud's DES implementation using the Transformed Masking Method was also vulnerable to such a High-Order DPA attack. To protect some secret-key cryptographic algorithms against any order DPA attack, they introduced a new countermeasure called Unique Masking Method, and applied it to DES implementation. However, recently, based on the fact that the output of the S-Box of the second round is unmasked, Akkar, Bévan and Goubin [3] presented an enhanced DPA attack on Akkar and Goubin's DES implementation using Unique Masking Method, and they finally gave an improved DES implementation using Unique Masking Method to avoid the enhanced DPA attack. Most recently, there were new advances in power analysis, as follows. Based on the Davies–Murphy attack [5,14], Kunz-Jacques et al. [20] presented a new kind of High-Order DPA attack on DES, called Davies–Murphy power attack that is more elaborated than ordinary High-Order DPA attacks. In [30], Prouff studied certain properties of S-boxes with respect to DPA attacks.

In this paper, we investigate the security of two DES implementations, the Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method and the Akkar and Giraud's DES implementation using Transformed Masking Method, against High-Order DPA attacks. By exploiting a new artifice to classify the electric consumption curves corresponding to the inputs, we show that Akkar Bévan and Goubin's improved DES implementation using Unique Masking Method is still vulnerable to a DPA attack that uses the outputs of the S-Boxes of the first two rounds. Besides, we find that it is also vulnerable to a superposition attack. Finally, by using the outputs of the S-Boxes of the first two rounds, or the last two rounds, or the second round and the last round, or the first round and the last second round, we present four new DPA attacks on Akkar and Giraud's DES implementation using Transformed Masking Method.

The rest of the paper is organised as follows. In the next section, we describe DPA and High-Order DPA attacks. In Section 3, we briefly review Akkar and Giraud's DES implementation using Transformed Masking Method, Akkar and Goubin's DES implementation using Unique Masking Method, and Akkar Bévan and Goubin's improved DES implementation using Unique Masking Method. In Section 4, we show our High-Order DPA attacks on Akkar Bévan and Goubin's improved DES implementation using Unique Masking Method. In Section 5, we present four new High-Order DPA attacks on Akkar and Giraud's DES implementation using Transformed Masking Method. Section 6 concludes this paper.

2. Description of DPA and High-Order DPA attacks

DPA is an attack that allows to obtain information about the secret key (contained in a smartcard for example), by performing a statistical analysis of the electric consumption records measured for a large number of computations with the same key.

The DPA attack on the DES can be performed as follows (cited from [15]):

Step 1: We measure the consumption on the first round, for 1000 (for example) DES computations. We denote by $M_1, \ldots, M_{1000}$ the input values of those 1000 computations. We denote by $C_1, \ldots, C_{1000}$ the 1000 electric consumption curves measured during the computations. We also compute the mean curve MC of those 1000 consumption curves.

Step 2: We focus for instance on the first output bit (as the target bit) of the first S-Box during the first round. Let b be the value of that bit. It is easy to see that b depends on only 6 bits of the secret key. We make an hypothesis on the involved 6 bits. We compute the expected (theoretical) values for b from those 6 bits and from the M_i ($i = 1, \ldots, 1000$). This enables us to separate the 1000 inputs $M_1, \ldots, M_{1000}$ into two categories: those giving $b = 0$ and those giving $b = 1$.

Step 3: We now compute the mean MC_0 of the curves corresponding to inputs of the first category (i.e., the one for which $b = 0$). If MC and MC_0 show an appreciable difference in a statistical meaning (i.e., a difference much greater than the standard deviation of the measured noise), we consider that the chosen values for the 6 key bits were correct. If MC and MC_0 do not show any sensible difference, we repeat step 2 with another choice for the 6 bits.

Step 4: We repeat steps 2 and 3 with a "target" bit b in the second S-Box, the third, ..., until the eighth S-Box. As a result, we finally obtain 48 bits of the secret key.

Step 5: The remaining 8 bits can be found by exhaustive search.

This attack does not require any knowledge about the individual electric consumption of each instruction, nor about the position in time of each of these instructions. It applies exactly the same way as soon as the attacker knows the outputs of the algorithm and the corresponding consumption curves. It only relies on the following fundamental hypothesis [2]:

Fundamental Hypothesis 1 (*Order 1*)**.** There exists an intermediate variable, that appears during the computation of the algorithm, such that knowing a few key bits (in practice less than 32 bits) allows to decide whether two inputs (respectively two outputs) give or not the same value for a known function of this variable.

High-Order DPA attacks generalize the DPA: the attacker now compute statistical correlations between the electrical consumptions considered at several instants. More precisely, an nth order DPA attack takes into account n values of the consumption signal, which correspond to n intermediate values occurring during the computation. These attacks rely on the following fundamental hypothesis [2],

Fundamental Hypothesis 2 (*Order n*)**.** There exists a set of n intermediate variables, that appear during the computation of the algorithm, such that knowing a few key bits (in practice less than 32 bits) allows to decide whether two inputs (respectively two outputs) give or not the same value for a known function of these n variables.

3. Review of the DES implementations using Transformed Masking Method and Unique Masking Method

3.1. Akkar and Giraud's DES implementation using Transformed Masking Method and Following Attacks

In this section, we will briefly describe Akkar and Giraud's DES implementation using Transformed Masking Method [1] and Akkar and Goubin's attack [2]. We refer the reader to [1,2] for details if our description is hard to follow.

3.1.1. Akkar and Giraud's DES implementation using Transformed Masking Method

Transformed Masking Method, introduced by Akkar and Giraud [1], is to perform all the computation that all the data are XORed with a random mask. By using suitably modified S-Boxes, it is possible to have the output of a round masked by exactly the same mask that masks the input. The computation is thus divided into two main steps: the first one consists in generating the modified S-Boxes, and the second one consists in applying the usual computation using these modified S-Boxes with the initial input being masked before starting DES and the final output being unmasked after DES.

Akkar and Giraud's DES implementation using Transformed Masking Method is as follows.

One chooses a 64-bit random mask X that will be XORed with the 64-bit message M at the beginning of the DES. Then he starts DES with the value $M \oplus X$. When it passes the Initial Permutation, the output value will become $IP(M) \oplus IP(X)$, where IP represents the Initial Permutation. At this point, the right and left 32 bits will respectively be $IP(M)_{32\text{-}63} \oplus IP(X)_{32-63}$ and

$$IP(M)_{0-31} \oplus IP(X)_{0-31}. \tag{1}$$

Just before the S-Box after E permutation, there will be an intermediary mask $E(IP(X)_{32-63})$, where E represents the Expansive Permutation of a DES round. To reestablish the mask $IP(X)$ at each round, Akkar and Giraud used a modified S-Box, denoted by SM-Box. The output of the SM-Box, after the permutation P following S-Box and after being XORed with the left part of the masked message, must have a mask equal to $IP(X)_{32-63}$. To meet this requirement, Akkar and Giraud defined the SM-Box as:

$$\text{SM-Box(A)} = S(A \oplus E(IP(X)_{32-63})) \oplus P^{-1}(IP(X)_{0-31} \oplus IP(X)_{32-63}),$$

where A is the input of SM-Box, S represents the original DES S-Box function and P^{-1} denotes the inverse of the permutation P following the S-Box. Therefore, after the input $E(IP(M)_{32-63}) \oplus E(IP(X)_{32-63}) \oplus K_1$ passes the SM-Box, the value will be

$$\begin{aligned}&\text{SM-Box}(E(IP(M)_{32-63}) \oplus E(IP(X)_{32-63}) \oplus K_1)\\ &\quad = S(E(IP(M)_{32-63}) \oplus K_1) \oplus P^{-1}(IP(X)_{0-31} \oplus IP(X)_{32-63}).\end{aligned} \tag{2}$$

After the value of Eq. (2) passes the P permutation and XORed with the left 32 bits, the value will become

$$P(S(E(IP(M)_{32-63}) \oplus K_1) \oplus P^{-1}(IP(X)_{0-31} \oplus IP(X)_{32-63}))$$
$$\oplus IP(M)_{0-31} \oplus IP(X)_{0-31}$$
$$= P(S(E(IP(M)_{32-63}) \oplus K_1)) \oplus IP(M)_{0-31} \oplus IP(X)_{32-63}.$$

At the same time, the right 32 bits $IP(M)_{32-63} \oplus IP(X)_{32-63}$ will become the new left 32 bits. Note that the new left 32-bit value has a mask $IP(X)_{32-63}$ that is different from the previous left 32-bit mask $IP(X)_{0-31}$ in Eq. (1). To implement easily in the following rounds, Akkar and Giraud XORed this new left 32-bit value $IP(M)_{32-63} \oplus IP(X)_{32-63}$ with $IP(X)_{0-31} \oplus IP(X)_{32-63}$ before executing further, so that the left 32-bit value has the same mask as in Eq. (1).

Similarly, after executing the left fifteen rounds, the output of the final round will have a mask $IP(X)_{0-31}||IP(X)_{32-63}$, where $\|$ denotes string concatenation. Since the left right 32 bits and the right 32 bits will interchange before the Final Permutation, so again for the easy implementation, Akkar and Giraud XORed both the right and left 32 bits of the final round with $IP(X)_{0-31} \oplus IP(X)_{32-63}$. So the mask just before the Final Permutation will become $IP(X)_{0-31}||IP(X)_{32-63}$, which will become X after Final Permutation IP^{-1}. Finally, by taking XOR of the value after Final Permutation with the mask X, one can recover the output of the message as the same output in a DES without countermeasures.

Note that there is always a random mask during each round, so it could prevent the initial DPA attack.

However, Akkar and Goubin [2] showed recently that it cannot withstand a High-Order DPA attack. Now let's briefly describe Akkar and Goubin's attacks in the following section.

3.1.2. Akkar and Goubin's attacks on Akkar and Giraud's DES implementation using Transformed Masking Method

Usual Second-Order DPA. In [2], Akkar and Goubin pointed out that their DES Implementation using Transformed Masking Method is subject to a second-order DPA attack. And the real output of the S-Boxes is correlated to the masked value and the random value, as a result, after getting the electrical trace of these two values, one can combine them and get a trace on which a classical DPA attack will work. In order to perform efficiently such an attack, the attacker should know precisely where the interesting values are manipulated.

Superposition Attack. Akkar and Goubin's superposition attack is a second-order DPA attack in theory, but in practice, it is nearly as simple as an usual DPA attack. The idea is as following: in a second order DPA attack, the most difficult thing is to localize the time when the precise needed values are manipulated, but on the contrary, localizing a whole DES round is often quite easy. So instead of correlating precise part of the consumption traces, the attacker will just correlate the whole trace of the first and the last round. With this method, one can notice that the attacker will have the following value T that is the XOR of the two values of the S-Boxes in the first and the last rounds:

$$T = (S(E(R_{15}) \oplus K_{16}) \oplus P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31}))$$
$$\oplus (S(E(IP(M)_{32-63}) \oplus K_1) \oplus P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31}))$$
$$= S(E(R_{15}) \oplus K_{16}) \oplus S(E(IP(M)_{32-63}) \oplus K_1),$$

where R_{15} are the right part of the output (corresponding to the input M and the same keys) of the 15th round in a DES without countermeasures.

Note that the value T does not depend on the random masking value and that R_{15} can be deduced from the output by applying the inverse of the Final Permutation IP^{-1}. Therefore, it is easy to see that after making a hypothesis on the 2×6 bits of the sub-key of the first and last round, it is possible to determine the XORed value of the output of the S-Boxes of the first and last round. After that one can perform an usual DPA attack and find the values of the different sub-keys of K_1 and K_{16}.

3.2. Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method

In this section, we will briefly describe the Unique Masking Method proposed by Akkar et al. and its application to DES implementation to defend DPA attack. We refer the reader to [2] for details.

3.2.1. Akkar and Giraud's DES implementation using Unique Masking Method

Unique Masking Method aims at providing a generic protection against any order DPA. The two principles of this method is first to mask only the values that depend on less than 32 bits of the key in order to prevent DPA, and second intermediate independent variables depending on less than 32 bits of the key should not be masked by the same value in order to thwart High-Order DPA.

Given any 32-bit value α, Akkar et al. first defined two new functions $\widehat{S}1$ and $\widehat{S}2$ based on the original DES S-Box function S:

$$\begin{cases} \forall x \in [0,1]^{48} : \widehat{S}_1(x) = S(x \oplus E(\alpha)) \\ \forall x \in [0,1]^{48} : \widehat{S}_2(x) = S(x) \oplus P^{-1}(\alpha) \end{cases}.$$

Then, they defined f_{K_i} to be the composition of the four transformations E, the XOR with the ith round subkey K_i the S-Box and the permutation P. Finally, they defined $\widehat{f}_{1,K_i}$ and $\widehat{f}_{2,K_i}$ by replacing S in f_{K_i} with $\widehat{S}_1$ and $\widehat{S}_2$.

Using the function f_{K_i}, $\widehat{f}_{1,K_i}$ and $\widehat{f}_{2,K_i}$, they obtained the following five different rounds using masked or unmasked values:

- A: The left and the right parts of the input are unmasked, and the function is f_{K_i}. Therefore, the left and the right parts of the output will also be unmasked.
- B: The left and the right parts of the input are unmasked, but the function is $\widehat{f}_{2,K_i}$. Therefore, the left part of the output will be unmasked, but the right part will be masked.
- C: The left part of the input is unmasked, but the right part is masked, and the function is $\widehat{f}_{1,K_i}$. Therefore, the left part of the output will be masked while the right part will be unmasked.
- D: The left part of the input is masked, but the right part is unmasked, and the function is f_{K_i}. Therefore, the left part of the output will be unmasked while the right part will be masked.
- E: The left part of the input is masked, but the right part is unmasked, and the function is $\widehat{f}_{2,K_i}$. Therefore, the left or the right part of the output will be unmasked.

To defend any order DPA attack, they gave a compatible 16 round DES implementation as follows: $IP - B_{\alpha_1}C_{\alpha_1}D_{\alpha_1}C_{\alpha_1}D_{\alpha_1}C_{\alpha_1}E_{\alpha_1}B_{\alpha_2}C_{\alpha_2}D_{\alpha_2}C_{\alpha_2}D_{\alpha_2}C_{\alpha_2}D_{\alpha_2}C_{\alpha_2}E_{\alpha_2} - FP$, where FP represents the Final Permutation of DES without countermeasures and B_{α_i} ($C_{\alpha_i}, D_{\alpha_i}$) denotes that the round is a B-type (respectively, C-type and D-type) with the mask α_i ($i = 1, 2$).

Furthermore, they pointed out that if one wants the mask never to appear several times, even on values depending on more than 36 bits of the key, one can use the following combination instead of the above one: $IP - B_{\alpha_1}C_{\alpha_1}E_{\alpha_1}AAAAAAAAAAB_{\alpha_2}C_{\alpha_2}E_{\alpha_2} - FP$. It is even possible to add two new masks and to mask every values depending on less than 56 bits of the key.

However, Akkar et al. [3] showed in FSE'04 that the above DES implementation using Unique Masking Method is vulnerable to an enhanced DPA attack, and finally they gave an improvement, which will be briefly described in the following section.

3.2.2. Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method

For all the proposed sequences of rounds in last section, the second round is always a C-type round. The output of the S-Box of this second round is

$$\begin{aligned} &S\left(E(P(S(K_1 \oplus E(IP(M)_{32-63}))) \oplus IP(M)_{0-31} \oplus \alpha_1) \oplus K_2 \oplus E(\alpha_1)\right) \\ &= S(\underline{E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus K_2} \oplus E(IP(M)_{0-31})), \end{aligned} \tag{3}$$

which is unmasked and stay unmasked after being XORed with the left part of the message.

Akkar et al. [3] pointed out that the fact that the output of the second round S-Boxes is unmasked will be vulnerable, for one can take the underlined value in Eq. (3) as the data to be acquired by a DPA attack. Based on this point, they presented a DPA attack on the above DES implementation using Unique Masking Method. The main idea of the attack is to retrieve two intermediate values which are not protected against DPA, and then to get the key bits by solving an equation involving the two intermediate values. The attack includes the following three parts:

- First Part:
  (1) The attacker performs DES computations with some chosen messages M_i ($\imath = 1, 2, \ldots, 1000$) for which the right part $IP(M_i)_{32-63}$ of the message M_i after IP will be set to an arbitrary but constant R_0. The left part $L_{0,i}$ will be random.
  (2) The attacker then performs a first-order DPA attack on the input of each S-Box of the second round. Because the output of the S-Boxes is unmasked, he will determine the value of the second round key XORed with the unknown but constant output of the S-Boxes of the first round. The found value will be:

  $$\delta = K_2 \oplus E(P(S(K_1 \oplus E(R_0)))).$$

- Second Part:
  (1) Similarly, the attacker performs another first-order DPA with other messages with a different known constant value R_0^*, which will provide:

  $$\delta^* = K_2 \oplus E(P(S(K_1 \oplus E(R_0^*)))).$$

- Final Part:
  (1) By taking XOR of the two values found at last two parts, the attacker can obtain the following value:

$$\delta \oplus \delta^* = (K_2 \oplus E(P(S(K_1 \oplus E(R_0))))) \oplus (K_2 \oplus E(P(S(K_1 \oplus E(R_0^*))))).$$

The value K_2 vanishes and the linearity of functions E and P gives the attacker the equation:

$$S(K_1 \oplus E(R_0)) \oplus S(K_1 \oplus E(R_0^*)) = P^{-1}(E^{-1}(\delta \oplus \delta^*)), \tag{4}$$

where E^{-1} is the inverse of E permutation.

  (2) Because the attacker knows R_0 and R_0^*, doing a exhaustive search on each 6-bit subkey of K_1, will give him all the possible values for the subkey K_1. On average, the differential properties of S will give him about 4 possibilities for each subkey. Since there are 8 subkeys and he also needs to find the 8 bits that are not in K_1, this gives him $4^8 \cdot 2^8 = 2^{24}$ possibilities on the key. So an exhaustive search with one known plaintext/ciphertext pair will take only a few seconds on a PC.

Finally, to improve the DES implementation by masking the output of the second round, they pointed out that one can use a different mask but the use of α_1 is not forbidden since the bits that are masked by the same value depends on 42 bits of the key, so they defined one more function $\widehat{f}_{3,K_i}$ with the modified S-Boxes $\widehat{S}_3(x)$ such that $\forall x \in [0,1]^{48} : \widehat{S}_3(x \oplus E(\alpha_1)) = S(x) \oplus P^{-1}(\alpha_1)$. Hereafter, the output of the S-Boxes of the second round in the improved DES implementation will be

$$\begin{aligned} &S(E(P(S(K_1 \oplus E(IP(M)_{32-63}))) \oplus IP(M)_{0-31} \oplus \alpha_1) \oplus K_2) \\ &= S(E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus E(IP(M)_{0-31}) \oplus E(\alpha_1) \oplus K_2) \\ &= S(E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus K_2 \oplus E(IP(M)_{0-31})) \oplus P^{-1}(\alpha_1). \end{aligned} \tag{5}$$

Note that every encryption there will be a random and different value $P^{-1}(\alpha_1)$ that is unknown to the attacker in Eq. (5), so the attacker cannot any longer classify correctly the message M_i into two groups, which seems to disable the above attack.

4. Our attacks on Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method

By using the outputs of the S-Boxes of the first two rounds in Akkar et al.'s improved DES implementation using Unique Masking Method, we could perform a DPA attack on it. Our attack is a chosen plaintext attack. Besides, it was also vulnerable to a superposition attack similar to the one in Section 3.1.2.

4.1. Main idea of our attack

Based on the fact that there is the same mask during the outputs of the S-Boxes of the first two rounds in Akkar et al.'s improved DES implementation using Unique Masking Method, our attacks are also to retrieve two intermediate values which are not protected against DPA by adopting a new technique to classify the electric consumption curves corresponding to the inputs, and then to get the key bits by solving an equation involving the two intermediate values. The new technique is crucial to successfully perform our attacks.

During Akkar et al.'s improved DES implementation using Unique Masking Method in Section 3.2.2, we can see that:

Step 1: The output of the S-Box of the first round is

$$S(K_1 \oplus E(IP(M)_{32-63})) \oplus P^{-1}(\alpha_1). \tag{6}$$

Step 2: The output of the S-Box of the second round is Eq. (5).

Step 3: By taking XOR of the two values of Eqs. (5) and (6) (i.e. XOR the outputs of the S-Boxes of the first and second rounds), then we get the following Eq. (7):

$$\begin{aligned} & S\ (E(P(S(K_1 \oplus E(IP(M)_{32-63}))) \oplus IP(M)_{0-31} \oplus \alpha_1) \oplus K_2) \\ & \quad \oplus S(K_1 \oplus E(IP(M)_{32-63})) \oplus P^{-1}(\alpha_1) \\ & = S(\underline{E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus K_2} \oplus E(IP(M)_{0-31})) \oplus S(K_1 \oplus E(IP(M)_{32-63})). \end{aligned} \tag{7}$$

Therefore, the random value $P^{-1}(\alpha_1)$ vanishes.

From Section 2, we learn that during a DPA attack, the attacker has to compute the value of the target bit and then classifies the electric consumption curves according to this value. Note that in Akkar et al.'s enhanced DPA attack in Section 3.2.2, the attacker is so lucky that he can explicitly get the value of the target bit in Eq. (3) corresponding to the message M after he makes an hypothesis on the six bits of the underlined value in Eq. (3). However, it is obvious that he will not be so lucky to get the value of the target bit depending on less than 32 key bits in Eq. (7), for after he makes an hypothesis on the six bits of the underlined value in Eq. (7), he has to compute $S(K_1 \oplus E(IP(M)_{32-63}))$ from this hypothesized underlined value when he computes the value of some target bit in Eq. (7), which will depend on more than 32 bits of key. On the other hand, only after he knows all the 32 bits of $S(K_1 \oplus E(IP(M)_{32-63}))$ could he compute the value of the target bit in Eq. (7). This incurs a main difference between our following attack and Akkar et al.'s enhanced DPA attack in Section 3.2.2.

Fortunately, we exploit a new technique to correctly classify the 1000 electric consumption curves corresponding to some 1000 (for example) inputs. Note in Eq. (7) that given K_1, if $IP(M)_{32-63}$ is set to some arbitrary but fixed value, then $S(K_1 \oplus E(IP(M)_{32-63}))$ will also be fixed. So if we classify the 1000 electric consumption curves corresponding to the 1000 inputs (the right 32 bits of each message after IP is fixed to a constant) according to some target bit in Eq. (7), we can also classify them to the same two groups according to the corresponding bit of

$$S(E(P(S(K_1 \oplus E(IP(M)_{32-63}))))) \oplus E(IP(M)_{0-31}) \oplus K_2). \tag{8}$$

Therefore, similar to Akkar et al.'s DPA attack in Section 3.2.2, we can perform a DPA attack with some chosen messages to acquire the value of $K_2 \oplus E(P(S(K_1 \oplus E(IP(M)_{32-63}))))$ and then perform another DPA attack with some different chosen messages to acquire a similar value. Finally, we can retrieve the key K_1 by taking XOR of the two acquired values.

We will show the details of our attack in the following section.

4.2. Our concrete attack

Step 1: During Eq. (8), letting

$$\theta = K_2 \oplus E(P(S(K_1 \oplus E(IP(M)_{32-63})))),$$

we now study on the following equation

$$\lambda = S(\theta \oplus E(IP(M)_{0-31})).$$

Step 2: We fix the right 32 bits $IP(M_i)_{32-63}$ of a message M_i after the initial IP to an arbitrary but constant M_R, and choose 1000 (for example) random 32-bit M_{Li} $(i = 1, 2, \ldots, 1000)$ as the left 32 bits of the 1000 inputs after IP. As what we describe in Section 2, by using these 1000 inputs, we can obviously apply a DPA attack to acquire θ_R,

$$\theta_R = K_2 \oplus E(P(S(K_1 \oplus E(M_R)))). \tag{9}$$

Step 3: By changing M_R to another different one M_{R^*}, we can acquire the corresponding θ_{R^*},

$$\theta_{R^*} = K_2 \oplus E(P(S(K_1 \oplus E(M_{R^*})))). \tag{10}$$

Step 4: By taking XOR of Eqs. (9) and (10), we get the following equation,

$$\begin{aligned}\theta_R \oplus \theta_{R^*} &= K_2 \oplus E(P(S(K_1 \oplus E(M_{R^*})))) \oplus K_2 \oplus E(P(S(K_1 \oplus E(M_R))))\\ &= E(P(S(K_1 \oplus E(M_{R^*})))) \oplus E(P(S(K_1 \oplus E(M_R)))).\end{aligned} \tag{11}$$

Step 5: From Eq. (11), we get

$$S(K_1 \oplus E(M_{R^*})) \oplus S(K_1 \oplus E(M_R)) = P^{-1}(E^{-1}(\theta_R \oplus \theta_{R^*})). \tag{12}$$

Note that Eq. (12) is similar to Eq. (4) except the values of the four known parameters $M_{R^*}, M_R, \theta_{R^*}$ and θ_R, so this again gives us $4^8 \cdot 2^8 = 2^{24}$ possibilities on the key. Consequently, as mentioned by Akkar et al. in [3], an exhaustive search with one known plaintext/ciphertext pair will take only a few seconds on a PC.

Therefore, Akkar et al.'s improved DES implementation using Unique Masking Method is still vulnerable to DPA attack.

`Note:` By fixing the right 32 bits of each message after *IP* to some arbitrary value and letting the left 32 bits change to get the enough inputs, we can correctly get the underlined value in Eq. (7) and K_1 simultaneously by performing a superposition attack similar to the one in Section 3.1.2.

5. Our four new attacks on Akkar and Giraud's DES implementation using Transformed Masking Method

Instead of using the outputs of the S-Boxes of the first round and last round of Akkar and Giraud's DES implementation using Transformed Masking Method, our new attack uses the outputs of the S-Boxes of the first two rounds, or the last two rounds, or the second round and the last round, or the first round and the last second round. The main idea of the attack using the first two rounds or the last two rounds is similar to the attack in Section 4.1, while the attack using the second round and the last round, or the first round and the last second round, is somewhat similar to the superposition attack in Section 3.1.2.

5.1. Attacks using the first two rounds or the last two rounds

During Akkar and Giraud's DES implementation using Transformed Masking Method in Section 3.1.1, we can see that,

Step 1: The output of the SM-Box of the first round is

$$S(K_1 \oplus E(IP(M)_{32-63})) \oplus P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31}). \tag{13}$$

Step 2: The output of the SM-Box of the second round is

$$\begin{aligned} & S\ (E(P(S(K_1 \oplus E(IP(M)_{32-63}))) \oplus IP(M)_{0-31} \oplus IP(X)_{32-63}) \\ & \oplus K_2 \oplus E(IP(X)_{32-63})) \oplus P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31}) \\ & = S(E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus E(IP(M)_{0-31}) \oplus K_2) \\ & \oplus P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31}). \end{aligned} \tag{14}$$

Step 3: By taking XOR of the two values of Eqs. (13) and (14)(that is, XOR the outputs of the S-Boxes of the first and second rounds), then we get the following

$$S\ (\underline{E(P(S(K_1 \oplus E(IP(M)_{32-63})))) \oplus K_2} \oplus E(IP(M)_{0-31})) \oplus S(K_1 \oplus E(IP(M)_{32-63})).$$

Therefore, the random value $P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31})$ vanishes.

Step 4: In the following, we can perform an attack similar to the one in Section 4.2.

Similarly, by using the outputs of the S-Boxes of the last two rounds, we can perform another attack on Akkar and Giraud's DES implementation using Transformed Masking Method, which is a chosen ciphertext attack. Since the right part of the final output of Akkar and Giraud's DES implementation using Transformed Masking Method is still required to be set to a arbitrary but constant value as in the above attack, the attacker could succeed only if he could collect the required enough outputs that have the same right 32 bits. Anyway, this attack threatens Akkar and Giraud's DES implementation using Transformed Masking Method.

5.2. Attacks using the second round and the last round or the first round and the last second round

We assume that C is the output corresponding to the input M. Then the value before the Final Permutation is $IP^{-1}(C)$, therefore we can get the output, $L_{16}||R_{16}$, of the last round as $R_{16} = IP^{-1}(C)_{0-31}, L_{16} = IP^{-1}(C)_{32-63}$. Finally, we can deduce out the output $L_{15}||R_{15}$ of the 15th round and the output $L_{14}||R_{14}$ of the 14th round as follows:

$$\begin{aligned} &R_{15} = IP^{-1}(C)_{32-63}, \\ &L_{15}(= R_{14}) = P(S(K_{16} \oplus E(IP(C)_{32-63}))) \oplus IP^{-1}(C)_{0-31}, \\ &L_{14} = P(S(E(P(S(K_{16} \oplus E(IP(C)_{32-63})))) \oplus K_{15} \oplus E(IP^{-1}(C)_{0-31}))) \oplus IP^{-1}(C)_{32-63}. \end{aligned} \tag{15}$$

By using Eq. (15), we can get the XOR of the outputs of S-Boxes of the second round and the last round in Akkar and Giraud's DES implementation using Transformed Masking Method as follows:

$$\begin{aligned} &S\ (K_2 \oplus E(R_1)) \oplus S(K_{16} \oplus E(R_{15})) \\ &\quad = S(\underline{K_2 \oplus E(P(S(K_1 \oplus E(IP(M)_{32-63}))))} \oplus E(IP(M)_{0-31})) \oplus S(K_{16} \oplus E(IP^{-1}(C)_{32-63})). \end{aligned} \tag{16}$$

Therefore, the random value $P^{-1}(IP(X)_{32-63} \oplus IP(X)_{0-31})$ vanishes, again.

Then, after by fixing the right 32 bits of each message after IP to some arbitrary value and letting the left 32 bits change to get the enough inputs, we can easily get the correct underlined value in Eq. (16) and K_{16} simultaneously by performing a High-Order DPA attack similar to the superposition attack in Section 3.1.2, given that we could choose the inputs and get their respective outputs.

The case for the first and the last second rounds is similar except that we should get the enough outputs that have the same right 32 bits, which may be impossible in practice, but in theory it is feasible.

6. Conclusions

In CHES'01, Akkar and Giraud presented a Transformed Masking Method to defend the DPA attack and applied it to DES implementation. Unfortunately, by using the outputs of the S-Boxes of the first and last rounds, Akkar and Goubin showed in FSE'03 a High-Order DPA attack on Akkar and Giraud's DES implementation using Transformed Masking Method, and finally they presented a DES implementation using their proposed Unique Masking Method to defend any order DPA attacks, which was later improved by Akkar, Bévan and Goubin in [3]. However, in this paper, we show that Akkar, Bévan and Goubin's improved DES implementation using Unique Masking Method is still vulnerable to DPA attacks. We also presented four new DPA attacks on Akkar and Giraud's DES implementation using Transformed Masking Method. A new technique to classify the electric consumption curves corresponding to the inputs is introduced in this paper.

As a further work, Lv et al. [21] summarized and proved five basic requirements for a DES implementation using masking methods to defense High-Order DPA attacks, and then presented an enhancement on Akkar et al.'s DES implementation using Unique Masking Method. The enhanced

DES implementation requires only three random 32-bit masks and six additional S-Boxes to be generated every computation, which was proved to be the minimal cost for a DES implementation masking all the outputs of the S-Boxes of the sixteen rounds to be secure against High-Order DPA attacks.

In November 2001, NIST declared the advanced encryption standard—AES [29] for the next generation. Nowadays, just as the referee mentioned, DES is becoming older and older for regular computing applications, though it is still alive in the smart-card world with its extremely limited computational resources. We hope those results obtained on DES so far could be taken on AES.

Acknowledgments

The author was very grateful to the anonymous referees for their helpful comments to improve this work, and also very grateful to the editor-in-chief and Becky Shepardson for their editorial efforts in the process of this paper.

References

[1] M. Akkar, C. Giraud, An implementation of DES and AES secure against some attacks, in: Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems CHES'01, Lecture Notes on Computer Science, vol. 2162, Springer-Verlag, Berlin, 2001.
[2] M. Akkar, L. Goubin, A generic protection against high-order dierential power analysis, in: Proceedings of the Fast Software Encryption 2003 FSE'03, Lecture Notes on Computer Science, vol. 2887, Springer-Verlag, Berlin, 2003.
[3] M. Akkar, R. Bévan, L. Goubin, Two power analysis attacks against one mask method, in: Proceedings of the Fast Software Encryption 2004 FSE'04, Lecture Notes on Computer Science, vol. 3017, Springer-Verlag, Berlin, 2004.
[4] E. Biham, A. Shamir, Differential cryptanalysis of the Data Encryption Standard, Springer-Verlag, Berlin, 1993.
[5] E. Biham, A. Biryukov, An improvement of Davies' attack on DES, in: Advances in Cryptology—EUROCRYPT'95, Lecture Notes on Computer Science, vol. 950, Springer-Verlag, Berlin, 1995.
[6] E. Biham, A. Shamir, Differential fault analysis of secret key cryptosystems, in: Advances in Cryptology—CRYPTO'97, Lecture Notes on Computer Science, vol. 1294, Springer-Verlag, Berlin, 1997.
[7] D. Boneh, R.A. DeMillo, R.J. Liption, On the importance of checking cryptographic protocols for faults, in: Advances in Cryptology—EUROCRYPT'97, Lecture Notes on Computer Science, vol. 1233, Springer-Verlag, Berlin, 1997.
[8] S. Char, C. Jutla, J. Rao, R. Rohatgi, A cautionary note regarding evaluation of AES candidates on smart-cards, in: Proceedings of the Second Advanced Encryption Standard Candidate Conference, 1999.
[9] S. Char, C. Jutla, J. Rao, R. Rohatgi, Towards sound approaches to counteract power-analysis attacks, in: Advances in Cryptology—CRYPTO'99, Lecture Notes on Computer Science, vol. 1666, Springer-Verlag, Berlin, 1999.
[10] J. Coron, L. Goubin, On boolean and arithmetic masking against differential power analysis, in: Proceedings of the workshop on Cryptographic Hardware and Embedded Systems CHES'00, Lecture Notes on Computer Science, vol. 1965, Springer-Verlag, Berlin, 2000.
[11] J. Coron, A. Tchulkine, A new algorithm for switching from arithmetic to boolean masking, in: Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems CHES'03, Lecture Notes on Computer Science, vol. 2779, Springer-Verlag, Berlin, 2003.
[12] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (6) (1976) 644–654.
[13] Data Encryption Standard, FIPS-46, National Institute of Standards and Technology, 1979. Available from: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>.
[14] D. Davies, S. Murphy, Pairs and triplets of DES S-boxes, Journal of Cryptology 8 (1) (1995) 1–25.

[15] L. Goubin, J. Patarin, DES and differential power analysis—the duplication method, in: Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems CHES'99, Lecture Notes on Computer Science, vol. 1717, Springer-Verlag, Berlin, 1999.
[16] L. Goubin, A sound method for switching between boolean and arithmetic masking, in: Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems CHES'01, Lecture Notes on Computer Science, vol. 2162, Springer-Verlag, Berlin, 2001.
[17] P. Kocher, Time attacks on implementation of Diffie–Hellman, RSA, DSS, and other systems, in: Advances in Cryptology—CRYPTO'96, Lecture Notes on Computer Science, vol. 1109, Springer-Verlag, Berlin, 1996.
[18] P. Kocher, J. Jaffe, B. Jun, Introduction to differential power analysis and related attacks, Technical Report, Cryptography Research Inc., 1998. Available from <http://www.cryptography.com/dpa/technical/index.html>.
[19] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Advances in Cryptology—CRYPTO'99, Lecture Notes on Computer Science, vol. 1666, Springer-Verlag, Berlin, 1999.
[20] S. Kunz-Jacques, F. Muller, F. Valette, The Davies–Murphy power attack, in: Advances in Cryptology—ASIACRYPT'04, Lecture Notes on Computer Science, vol. 3329, Springer-Verlag, Berlin, 2004.
[21] J. Lv, Y. Han, Enhanced DES implementation secure against high-order differential power analysis in smartcards, in: Proceedings of the Tenth Australasian Conference on Information Security and Privacy ACISP'05, Lecture Notes on Computer Science, vol. 3574, Springer-Verlag, Berlin, 2005.
[22] M. Matsui, Linear cryptanalysis method for DES cipher, in: Advances in Cryptology—EUROCRYPT'93, Lecture Notes on Computer Science, vol. 765, Springer-Verlag, Berlin, 1994, pp. 386–397.
[23] T. Messerges, A. Dabbish, R. Sloan, Power analysis attacks of modular exponentiation in smartcards, in: Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems CHES'99, Lecture Notes on Computer Science, vol. 1717, Springer-Verlag, Berlin, 1999.
[24] T. Messerges, Using second-order power analysis to attack DPA resistant software, Lecture Notes on Computer Science, Springer-Verlag, Berlin, 2000.
[25] T. Messerges, Securing the AES finalists against power analysis attacks, in: Proceedings of the Fast Software Encryption 2000 FSE'00, Lecture Notes on Computer Science, vol. 1978, Springer-Verlag, Berlin, 2001.
[26] T. Messerges, A. Dabbish, R. Sloan, Examining smart-card security under the threat of power analysis attack, IEEE Transactions on Computers 51 (4) (2002).
[27] http://www.nist.gov.
[29] National Institute of Standards and Technology, Advanced encryption standard FIPS 197, US Department of Commerce, November 2001.
[30] E. Prouff, DPA attacks and S-Boxes, in: Proceedings of the Fast Software Encryption 2005 FSE'05, Lecture Notes on Computer Science, vol. 3557, Springer-Verlag, Berlin, 2005.
[31] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (1978) 120–126.