An Efficient Hardware-Based Fault-Tolerant Method for SMS4

Hanguang Luo, *Guangjun* Wen and *Jian* Su

University of Electronic Science and Technology of China, 611731 West Hi-Tech Zone, China

Abstract. The SMS4 cryptosystem has been used in the Wireless LAN Authentication and Privacy Infrastructure (WAPI) standard for providing data confidentiality in China. So far, reliability has not been considered a primary objective in original version. However, a single fault in the encryption/decryption process can completely change the result of the cryptosystem no matter the natural or malicious injected faults. In this paper, we proposed low-cost structure-independent fault detection scheme for SMS4 cryptosystem which is capable of performing online error detection and can detect a single bit fault or odd multiple bit faults in coverage of 100 percent. Finally, the proposed techniques have been validated on Virtex-7 families FPGA platform to analyze its power consumption, overhead and time delay. It only needs 85 occupied Slices and 8.72mW to run a fault-tolerant scheme of SMS4 cryptosystem with 0.735ns of detection delay. Our new scheme increases in minimum redundancy to enhance cryptosystem's reliability and achieve a better performance compared with the previous scheme.

1 Introduction

With the rapid development of network and digitization, the information protection has nowadays became an absolutely requirement. Encryption technology is not only be used for computer network, but also used on some special digital equipment and chip cryptographic services to protect information. Due to the limited resources and highly-constrained of the equipment or embedded chip, time constraint, electricity constraint etc., lightweight encryption algorithm is widely used in numerous data privacy places. SMS4 is one of the lightweight block cipher using a Unbalance Feistel Network(UFN) structure, designing for the resource-constrained platforms, which is mandatory in wireless networks by China in 2006[1] .

As the hardware implementations of SMS4 algorithm, stresses, temperature changes and electromagnetic fields affect electronic devices causing it abnormal operation. Furthermore, as a result of rapid diffusivity of its security function, even a bit fault can cause complete decryption failure. In addition, it had not been known that different error output could be used to carry out an attack on the cryptographic algorithm until 1996 by Boneh, DeMillo, and Lipton [2] from Bellcore. Soon afterwards, Differential Fault Analysis(DFA) was extended to DES-like secret key cryptosystems by Biham and Shamir[3].Since SMS4 was published, several authors mounted DFA attacks on it[4,5,6]. The main idea of these attacks is induced several byte fault into the register of the round function, then use DFA attack to retrieve secret key. It is reported in [6] that a successful attack on SMS4 requires only a single random byte faulty responses.

Therefore, an effective error detection scheme is required to improve SMS4 algorithm of fault tolerance. There have been some traditional methods applied to the fault detection of other cipher, e.g. AES. A scheme is put forward in [7], which used reverse functions to recover the original input for the encryption/decryption process; [8,9] can applied to the internal function, round function, or entire encryption process levels, based on the hardware-redundant-based concurrent error detection schemes. Another category of fault detection methods is based on parity codes [10,11,12]. This kind of methods utilized the parity check codes to each byte of the state matrix. It can also be applied to each transformation, each round, or the encryption process levels. To the best of our knowledge, [13] is the only one article which worked to investigate fault-tolerant methods for SMS4 and used reverse functions in the process levels.

The organization of this paper is as follows: In Section 2, we review the basics of SMS4. In Section 3, we introduce the traditional simple method of faults detection and implemented it to the SMS4. Then we present the proposed less redundancy faults detection scheme. Section 4 presents the experimental results and hardware implementation. Finally, conclusions are made in Section 5.

2 Description of SMS4 algorithm

SMS4 is a 128-bit block cipher, which supports 128-bit block size and 128-bit key lengths. It iterates a simple round function 32 times. The 128-bits plaintext is divided into 4 words each has 32-bit length, $P=(P_0,P_1,P_2,P_3)\in(¡^{32}_{2})^4$. The ciphertext and round

subkeys are also divided into 4 words denoted as $C=(C_0,C_1,C_2,C_3)\in(¡\ ^{32}_{2})^4$ and $RK_i\in¡\ ^{32}_{2}(i=0,1,2,...,31)$, respectively. Note that the first round is referred to Round 0.

The one round structure of SMS4 is depicted in Figure.1 and the encryption procedure of SMS4 is as follows:

1) For i =0 to 31, the words are updated as $X_{i+4}=X_i\oplus F(X_{i+1}\oplus X_{i+2}\oplus X_{i+3},\ RK_i)$.

2) The 128-bit ciphertext with output of the last round is generated by applying the switch transformation R denoted as

$$\begin{aligned}(Y_0,Y_1,Y_2,Y_3)&=R(X_{32},X_{33},X_{34},X_{35})\\&=(X_{35},X_{34},X_{33},X_{32})\end{aligned}\quad(1)$$

The Addroundkey is a process that let three input X_{i+1}, X_{i+2}, X_{i+3} do XOR operation, then put the 32-bit output XOR with the round key RK_i. The non-linear transformation S is constituted of four equal S-box in parallel to the 32-bit input. The transformation S is defined as follows.

$$S(B)=(Sbox(a_0),Sbox(a_1),Sbox(a_2),Sbox(a_3))\quad(2)$$

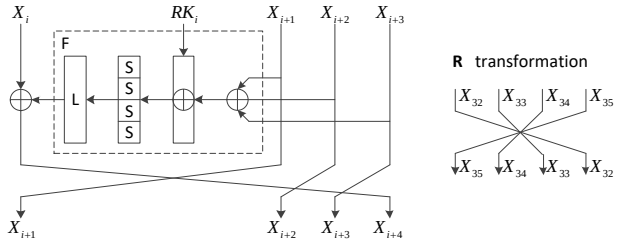

Figure 1. The i-th round of SMS4

Where $a_i\in¡\ ^{8}_{2}, i\in0,1,2,3$ denotes the four 8-bit input of S-box which achieved by the output of the Addroundkey process and divided into four equal parts. $B\in¡\ ^{32}_{2}$ is the output of the transformation S.

The diffusion transformation L is a simple linear function which follows the transformation S. Its input is the output of the transformation S and defined as follows.

$$\begin{aligned}L(B)=B\oplus(B<<<2)\oplus(B<<<10)\\\oplus(B<<<18)\oplus(B<<<24)\end{aligned}\quad(3)$$

The Key schedule of the SMS4 is similar to the encryption function and it only has 128-bit key version. The main idea of the Key schedule is to generate 32-bit round key by a 128-bit master key in 32 rounds. As the key schedule algorithm is not involved in our research we omit the concrete process. And interested readers may refer to [14] for the details.

3 Error detection methods

3.1. Error model

The pathway of discussed error model in this paper can be divided into two categories: deliberately introduced error in a malicious attempt and internal error. The first one is usually inserted by an adversary for using Fault Analysis (FA) or Differential Fault Analysis (DFA) in order to retrieve secret key of the algorithm. And the faults can be inserted into the register unit in each round. The latter one occurs during the process of encryption/decryption. This class of fault happens in a natural consequence of the module's normal operation, duration of use or environment, which takes into account a global, physical and electrical failure approach. Both of them can be temporary or permanent faults.

Although SMS4 algorithm allows the 32 iteration round unfold into line structure, the hardware overhead is not suitable for most of the embedded application. Due to the different applications, the demand for encryption/decryption speed, and power consumption, area resources, the structure of SMS4 algorithm is also different. Without loss of generality, we take the 32 iteration round structure into account. We also assume that an error can both be inducted into the register and the state between transformations in each round.

Since most of embedded systems, data are based on bytes of storage and processing, we also assume that the error happened only in a single bit or odd multiplies of bits, whether it is a malicious attempt insert error or internal occur error. Errors, however, the number of bytes and location are random.

3.2 Fault detection techniques

1) Redundancy-Based Techniques (RBT). Traditional fault-tolerant methods have been widely investigated for the implementation of cipher. One of the most simple methods is Redundancy-based Techniques (RBT) [15]. The main idea of the method is as follows. Consider an arbitrary function $y=f(x)$. If $f(*)$ exists an invertible function $z=f^{-1}(y)$ in the domain. We can safely come to the conclusion that must equal to z. i.e. $x=z=f^{-1}(y)$. We can also apply this method to SMS4 fault detection. It not only can be applied at the algorithm level but also at the round function level. The detail can be seen in the Figure.2. at (a) and (b), respectively.

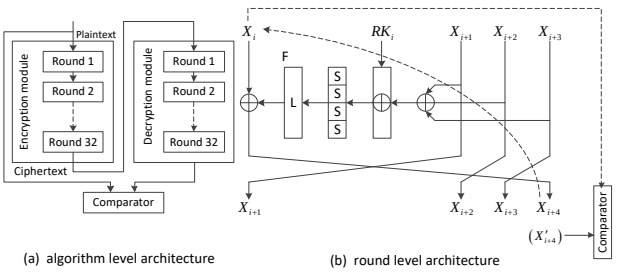

Figure 2. Method of Redundancy-based Techniques

As shown in Figure 2. (a), when RBT is employed at the algorithm level, the ciphertext is decrypted to the plaintext then compared with the original plaintext. If faults are produced in the process of encryption, the result will not equal. When it comes to the round level, due to the structure of SMS4 round function, the encryption and decryption are almost the same. The four input data $X_i, X_{i+1}, X_{i+2}, X_{i+3} \in ¡ ^{32}_{2}$ which divided from the plaintext, is operated with round key RK_i result in $X_{i+1}, X_{i+2}, X_{i+3}, X_{i+4} \in ¡ ^{32}_{2}$ (see Figure 2.(b) with the encryption). The imaginary line describes the change data of decryption process in a round, i.e. the round key is also be RK_i , the input of decryption is $X_{i+4}, X_{i+1}, X_{i+2}, X_{i+3} \in ¡ ^{32}_{2}$ and the output is $X_{i+1}, X_{i+2}, X_{i+3}, X'_{i+4}, \in ¡ ^{32}_{2}$. If no error occurs in the encryption process the decryption output data X'_{i+4} must equal to X_i , or it is incorrect.

Although the principle of RBT method is quite simple and easy to be implemented, the time expense on the fault detective and the power consumption is almost equivalent to the encryption process. This can't be tolerated with some of the resources limited embedded device, e.g. RFID. At the same time we cannot guarantee, as a part of error detection process, the decryption is exactly correct. If some faults happen to the detection process, the whole error detection progress fails. Therefore to find a suitable new redundancy scheme is necessary.

2) Error Detecting Codes (EDC). Bertoni et al. proposed in [16] in AES algorithm used a parity bit to resist fault analysis. They use a single parity bit to represent the 8 bit word in the middle of the status or register unit.

$$P_{i,j} = \bigoplus_{k=0}^{7} S_{i,j}^{(k)} \tag{4}$$

The same method can also be used in SMS4 algorithm. We check on the single parity bit in order to know whether there is a bit or odd bit fault in each register unit. Meanwhile, it can be extended to arbitrarily bit situation to check more bits in each time. i.e.

$$P = \bigoplus_{i=0}^{n} W^i \tag{5}$$

For the convenience of description and easy to understand the fault detection scheme, the round function is divided into four parts：1) “AddRK” short for Addroundkey: $\left(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus RK_i\right)$; 2) “Sbox” short for S-box transformation; 3) “ Ltranf ” short for L transformation; 4) “ OUT- X_{i+4} ” represented the output X_{i+4} : $X_{i+4} = X_i \oplus L(B) \in ¡ ^{32}_{2}$. And we define the corresponding parity bit denoted with symbol “$\mathrm{I}\ *$”. e.g. The corresponding parity bit of X_i denoted as $\mathrm{I}\ X_i = \bigoplus_{i=0}^{3} p^i \in ¡ _2$, where p^i is the parity bit to the corresponding register unit, i.e. $p^0 = \bigoplus_{k=0}^{7} x_i^k, \mathrm{L}\ , p^3 = \bigoplus_{k=24}^{31} x_i^k$.

To our scheme, the detection can be applied at round function level or at the operation level. Here we only show the round function level scheme, see Figure 3. The operation level scheme is similar and omitted. The S-boxes of SMS4 algorithm are used in a look-up table (LUT) method.

Initialization, each storage unit is upgraded from the original 8 bit to 9 bit. The actual data is kept in the first 8 bit, and the parity bit is deposited in the last bit. To the look-up table of the S-box, the same as the input data, each corresponding parity bit is calculated and reserved behind the actual data in advance. The round key is also be calculated by the secret key through the key schedule beforehand. Then calculated the corresponding parity bit and reserved it.

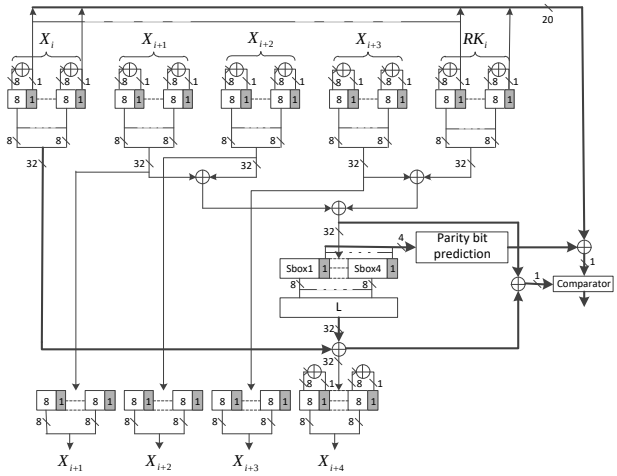

Figure 3. Round function level of parity bit prediction

To the Addroundkey part, as shown in figure 3, we can obtained that

$$\mathrm{I}\ (AddRK) = \left(\mathrm{I}\ X_{i+1} \oplus \mathrm{I}\ X_{i+2} \oplus \mathrm{I}\ X_{i+3} \oplus \mathrm{I}\ RK_i\right) \tag{6}$$

The S-box is the only nonlinear device in the SMS4 algorithm. When comes to the S-box transformation, the output and the corresponding parity bit can both be obtained from the look-up table.

L transformation is a linear device denoted in (3). It is obtained by the S-box output with cyclic shift several times, then XOR each other. As the left rotation operations do not affect the parity bit of its outputs, meanwhile the XOR operation times is odd. The parity bit of L transformation is equal to S-box’s. So the parity bit prediction of the L transformation is as follow:

$$\begin{aligned} \mathrm{I}\ (\mathrm{Ltranf}) &= \mathrm{I}\ L(B) \\ &= \mathrm{I}\ \left\{ \begin{matrix} B \oplus (B <<< 2) \oplus (B <<< 10) \oplus \\ (B <<< 18) \oplus (B <<< 24) \end{matrix} \right\} \\ &= \mathrm{I}\ B \oplus \mathrm{I}\ (B <<< 2) \oplus \mathrm{I}\ (B <<< 10) \\ &\quad \oplus \mathrm{I}\ (B <<< 18) \oplus \mathrm{I}\ (B <<< 24) \\ &= \mathrm{I}\ (Sbox) \in ¡ _2 \end{aligned} \tag{7}$$

Finally, the X_{i+4} is achieved by X_i XOR with L transformation output. The parity bit is

$$\mathrm{I}\ (OUT - X_{i+4}) = \mathrm{I}\ (X_i \oplus L(B)) = \mathrm{I}\ X_i \oplus \mathrm{I}\ L(B) \quad (8)$$

From what has been discussed above, if no faults happen in the encryption process, all parts of the parity bits will equal to all the parts of the actual data XOR. i.e.

$$\begin{aligned} &\bigoplus_{k=0}^{31}\left[\left(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus RK_i\right) \oplus X_{i+4}\right]^k \\ &= \bigoplus_{k=0}^{31}\left[\left(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus RK_i\right) \oplus X_i \oplus L(B)\right]^k \\ &= \mathrm{I}\ \left(X_i \oplus X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus RK_i\right) \oplus \mathrm{I}\ Sbox \\ &= \mathrm{I}\ X_i \oplus \mathrm{I}\ X_{i+1} \oplus \mathrm{I}\ X_{i+2} \oplus \mathrm{I}\ X_{i+3} \oplus \mathrm{I}\ RK_i \oplus \mathrm{I}\ Sbox \in ?_2 \end{aligned} \quad (9)$$

where $\mathrm{I}\ X_i$, $\mathrm{I}\ X_{i+1}$, $\mathrm{I}\ X_{i+2}$, $\mathrm{I}\ X_{i+3}$, $\mathrm{I}\ RK_i$, $\mathrm{I}\ Sbox$ are the parity bit of the input date and Keys saved before in the last bit of storage unit.

Finally, before the output X_{i+4} is stored, the data of X_{i+4} should be calculated to the corresponding parity bit then preserved. The fault detection scheme of operation level version is to detect of the above four parts, respectively. From what has been discussed above, both the malicious faults and internal faults can be detected.

4 Simulation results

4.1. Fault detection rate

In order to test the faults detection rate of the EDC scheme, we implemented the detection scheme in C++ and tested the fault detective scheme from two types:

i) Randomly selected 10000 plaintexts data blocks and injected a single bit error (into round key or round plaintexts) into random position and random rounds. However, the secret key of key schedule was remained the same. Though our error detection scheme, all the faults were detected.

ii) Randomly selected 10000 plaintexts data blocks with the same secret key. An odd bit error (into round key or round plaintexts) was injected into random position and random rounds. Here too, all the faults were detected.

Through the experiment above, the proposed scheme can accurately detected the single and multiple odd bit error in SMS4. In order to determine the power consumption and overhead, further hardware experiment should be taken.

4.2. FPGA implementations and comparison

In this section, we used FPGA to verify practicability of the proposed schemes. The SMS4 algorithm and fault detective scheme are both designed in 32 iteration round structure. The S-boxes is designed in LUT-based method, parity bits of the S-boxes were calculated and preserved beforehand. The round keys and corresponding parity bits were also calculated and preserved beforehand. This scheme not only saved the time spent by extension of round keys but also reduced the overhead. We have used Verilog HDL as the design entry for ISE version 14.1 and the synthesis is performed using Xilinx Synthesis Tool on Virtex-7 families (XC7V585T).

As shown in Table 1, the used numbers of Slice LUTs in this paper were 169 compared with the 118 to the original scheme. The Slice Registers were 168 compared with the 129 to the original one. Although the total percentage of increased overhead is approximately 51.79%, the cut down of round key extension must have saved a large part of the overhead compared with the one which has the extension of key schedule. And then we compared with fault detection scheme of [13], the total occupied logic of resources were less than one third of [13].

Table 1. Overhead of different scheme of the SMS4 algorithm on Xilinx Virtex-7 FPGAs

Structure	overhead				Increase Overhead (%)
	Slice LUTs	*Slice Registers*	*IOBs*	*occupied Slices*	
Original	118	129	259	56	-
This paper	169	168	261	85	51.79%
RBT[13]	-	-	-	270	382.14%

Table 2. Power consumption of different scheme of SMS4 algorithm on Xilinx Virtex-7 FPGAs

Structure	Power(mw)				Increase (%)	Delay (ns)
	Logic	*Signals*	*IOs*	*Total*		
Original	1.15	3.07	1.74	5.96	-	-
This paper	2.86	4.12	1.74	8.72	46.30%	0.735

With the increase of overhead, the corresponding power consumption was also increased. We compared the original scheme (don't have the extension of key schedule) with our scheme of SMS4 cryptosystem's power consumption in Table 2. Total power consumption of our scheme was 8.72mW, increased 2.76mW compared with the original scheme. Although [13] didn't show the power consumption of the fault detection scheme, the cost of the power consumption of our scheme must be lower than the cost of [13], since the method of it only did the inverse process of encryption and the S-boxes process is the most power consuming part. From what has been discussed above, considering the contribution of faults detection, the little bit of increasing redundancy is completely acceptable. At the same time, the delay time of 0.735ns which caused by the additional circuit is also acceptable.

5 Conclusion

In this paper we did some research on the SMS4 cryptosystem of fault detection. We showed the traditional scheme of error detective method on SMS4 and presented a new fault detection scheme based on parity code. The simulations showed that our new fault detection structure reached the error detected coverage of 100 percent.

In addition, the proposed fault detection scheme has been implemented on the recent Xilinx Virtex-7 families FPGAs. The overhead, delay and power consumption have all achieved very good performance compared with the original and prevenient scheme. The reason why new scheme of fault detecting structure used so less resource is that we only utilized XOR operation to checkout fault. And the time delay is also in an absolutely acceptable range since XOR operation is easy and fast to be implemented in hardware. At the same time, the reduced of the round key extension part also greatly reduces the cost. Therefore, we can conclude that from our proposed scheme it can well done on the fault detection and defensed Fault Attack on SMS4 cryptosystem.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under project contracts No.61601093, No.61791082, No. 61701116 and No.61371047, in part by Sichuan Provincial Science and Technology Planning Program of China under project contracts No.2016GZ0061, 2017GZ0336 and No.2018HH0034, in part by Guangdong Provincial Science and Technology Planning Program of China under project contracts No.2015B090909004 andNo.2016A010101036, in part by the fundamental research funds for the Central Universities under project contract No.ZYGX2016Z011, and in part by Science and Technology on Electronic Information Control Laboratory.

References

1. Office of State Commercial Cipher Administration. "Block Cipher for WLAN Products-SMS4", http://www.oscca.gov.cn/UpFile/200621016423197990.pdf.
2. Dan Boneh, Richard A. DeMillo, Richard J. Lipton, “On the importance of checking cryptographic protocols for faults”, in: EUROCRYPT’97, in: LNCS, vol. 1233, Springer 1997, pp. 37–51.
3. Eli Biham, Adi Shamir, “Differential fault analysis of secret key cryptosystems”, in: CRYPTO 1997, in: LNCS, vol. 1294, Springer, 1997, pp. 513–525.
4. Wei Li, Dawu Gu, “An improved method of differential fault analysis on the SMS4 cryptosystem", in: ISDPE 2007, IEEE Computer Society, 2007, pp. 175–180.
5. Wei Li, Dawu Gu, “Differential fault analysis on the SMS4 cipher by inducing faults to the key schedule”, Chinese Journal on Communications 29 (10) , 2008, pp.135–142.
6. Ruilin Li, Bing Sun, Chao Li, Jianxiong You, “Differential fault analysis on SMS4 using a single fault, Information Processing Letters”, vol.111, 2011, pp.156–163.
7. R. Karri, W. Kaijie, P. Mishra, and K. Yongkook, “FaultBased Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture,” in Proc. DFT’01, 2001, pp. 418–426.
8. R. Karri and X. Guo “Invariance-based concurrent error detection for advanced encryption standard,” in Proc. DAC, 2012, pp. 573–578.
9. M. Mozaffari-Kermani and A. Reyhani-Masoleh, “Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard,” IEEE Trans. Computers, vol. 59, no. 5, May 2010, pp. 608–622.
10. J. Mathew, et al., “On the design of different concurrent EDC schemes for s-box and gf(p),” in Proc. ISQED’10, 2010, pp. 211-218.
11. Ting An, Lirida Alves de Barros Naviner and Philippe Matherat, “Evaluation of Fault-tolerant Composite Field AES S-Boxes under Multiple Transient Faults,” in Proc. NEWCAS’13, 2013, pp 1–4.
12. A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache e, “Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures,” Proc. IEEE, vol. 100, no. 11, Nov. 2012,pp. 3056–3076.
13. Xin Xiaoxia, Wang Yi, Li Renfa, “FPGA-based implementation for fault detection of SMS4”, Journal of Computer Applications, vol. 35, no. 2, pp. 420-3, 10 Feb. 2015.
14. Specification of SMS4, Block cipher for WLAN products–SMS4, http://www.oscca.gov.cn/UpFile/200621016423197990.pdf
15. Abdel Alim Kamal, Amr M. Youssef, “An FPGA Implementation of AES with Fault Analysis Countermeasures Analysis Countermeasures”, in: ICM 2009, Vol.978, May 2009, pp.217-220.
16. G. Bertoni, L. Breveglieri,I. Koren, P. Maistri, and V. Piuri, “Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard”, IEEE Transactions on Computers, Vol. 52, April 2003, pp. 492-505.