Multilinear Codes for Robust Error Detection

Zhen Wang, Mark Karpovsky
Boston University
Reliable Computing Laboratory
8 Saint Marys Street,Boston, MA, USA
{lark,markkar}@bu.edu

Berk Sunar
Worcester Polytechnic Institute
CRIS Laboratory
100 Institute Road, Worcester, MA 01609
sunar@wpi.edu

Abstract

We propose an efficient technique for the detection of errors in cryptographic circuits introduced by strong adversaries. Previously a number of linear and non-linear error detection schemes were proposed. Linear codes provide protection only against primitive adversaries which no longer represents practice. On the other hand non-linear codes provide protection against strong adversaries, but at the price of high overhead. Here we propose a novel error detection technique, based on the random selection of linear codes. Under mild assumptions the proposed construction achieves near non-linear code error detection performance at much lower cost due to the fact that no non-linear operations are needed for the encoder and decoder.

1. Introduction

Cryptography has matured to the point where the biggest threat is due to attacks that directly target the implementation. Side-channels such as the execution time, power signature, accoustic and electromagnetic emanations, [10, 9] leak dependent that may (and for most cases does) allow the attacker to recover the secret key. Even worse an attacker may inject faults into the device forcing an error in the computation or forcing the device into an undesired state. Due to their active and adaptive nature, fault based attacks are one of the most powerful types of side-channel attacks. For instance the attack proposed by Boneh et al. in [1] trivially recovers the RSA factors by introducing an arbitrary fault in one of two RSA signature computations. One of the most efficient fault injection attacks on AES-128, for example, requires only 2 faulty ciphertexts to retrieve all 128-bits of the secret key [8]. Without proper protection architecture against fault injection attacks, the security of the cryptographic devices can never be guaranteed.

Tamper detection can be achieved by embedding sensors into the device, covering the device with a protective coating [12, 13], secure key storage [11] or by implementing error detection codes (EDCs). Among these techniques, only EDC allows us to easily establish a precise metric that measures the error detection performance. Sensor integration and the protective coating approaches require significant changes in the manufacturing process. The secure key storage technique does not provide comprehensive protection.

There are numerous EDC proposals based on linear codes such as parity codes and Hamming codes in the literature. Protection architectures based on linear codes concentrate their error detecting abilities on errors with small multiplicities or errors of particular types, e.g. byte errors or errors with odd multiplicities. This will suffice if the adversary has limited temporal and spatial resolution. However, in recent work it was shown that an arbitary bit location in memory could be modified via an optical induction attack [14]. Against such capable attackers linear codes stand no chance.

In [5], algebraic robust codes were proposed as an alternative to classical linear codes to protect cryptographic devices implementing AES against fault injection attacks. In [3] and [7], robust arithmetic residue codes were proposed which can be used to design fault tolerant cryptographic devices performing arithmetic operations. Instead of concentrating the error detecting abilities on particular types of errors, robust codes provide nearly equal protection against all error patterns hence eliminate weaknesses that can be exploited by attackers. The main disadvantage of robust codes is the large hardware overhead of the encoding and decoding circuits due to the non-linear operations.

In this paper, we propose a new method to achieve robustness. Instead of using non-linear functions to generate the signature of the code, we randomly select a linear code from a set of linear codes at each clock

cycle. For a given error e, the error masking probability $Q(e)$ is defined to be the fraction of codewords that mask the error.

$$Q(e) = \frac{|\{c|c \in C, c+e \in C\}|}{|C|}. \quad (1)$$

We distinguish between two kinds of errors – errors that are masked with probability 1 and errors that are masked with probability less than 1 but larger than 0. The former is called **undetectable errors** and the latter is called **conditionally detectable errors**. The proposed method can achieve as small number of undetectable errors as classical robust codes while requiring much less hardware overhead.

2 Multilinear Algebraic Codes

For the remainder of this paper we denote by $\{C_i, 1 \leq i \leq l, l \geq 2\}$ the set of linear codes, where l is the number of different codes in the set. We first propose several methods of constructing linear algebraic codes $C_i, 2 \leq i \leq l$ from C_1 in such a way that randomly selecting $C_i, 1 \leq i \leq l$ have much less or even no undectable errors. For the randomization a standard low-rate true random number generator is used, which is available in most cryptographic devices.

Construction 2.1 *Let C_1 be a (n, k) linear code with Hamming distance larger than 2. $C_i, 2 \leq i \leq l, l = k$ is constructed by swapping the first and the i_{th} information bits of C_1. If we randomly select $C_i, 1 \leq i \leq l = k$ to encode the original messages with equal probability, the only undetectable error is the codeword of C_1 with all $1's$ in the information part. Errors that have the same value for the first $k-1$ information bits will be masked with probability $\frac{k-1}{k}$, which is the maximum conditional error masking probability.*

Proof Undetectable errors are codewords that belong to all of the $l = k$ linear codes $C_1, C_2, \cdots, C_l$. Because the Hamming distance of C_1 is larger than 2, the intersection of these codes contains only the vector (also a codeword) with all $1's$ information bits and the vector with all $0's$ information bits. Errors which have the same value for the first $k-1$ information bits belong to $k-1$ linear codes. So it will be masked with probability $\frac{k-1}{k}$ if we select the codes with equal probability. Obviously, this is the maximum conditional error masking probability. ■

When implemented in hardware, the overhead of Construction 2.1 may be excessive. To reduce the hardware overhead, we can use only $l, 2 \leq l < k$ linear codes. Generally speaking, when we randomly select $l, 2 \leq l \leq k$ linear codes to encode the messages, the number of undetectable errors is 2^{k-l+1} (including the all $0's$ vector). The maximum error masking probability for all conditionally detectable errors is $\frac{l-1}{l}$. The simplest case is to use only C_1 and C_2 to encode the messages, where C_2 is built by swapping the first and the second information bits of C_1. This method requires only 2 more $2:1$ multiplexer for the encoder while the number of undetectable errors is reduced by 50% compared with the method using only C_1. All conditionally detectable errors will be detected with probability 0.5.

Another variation of Construction 2.1 is to swap the redundant bits instead of the information bits of C_1. Suppose $C_i, 2 \leq i \leq r$ is constructed by swapping the first and the i_{th} redundant bits of C_1. Assume that all 2^r binary vectors are possible for the redundant part of C_1. If we randomly select $l, 2 \leq l \leq r$ linear codes to encode the original messages with equal probability, the number of undetectable errors is 2^{k-l+1}. The maximum error masking probability for conditionally detectable errors is $\frac{l-1}{l}$. For this variation the smallest possible number of undetectable errors is 2^{k-r+1} which is larger than that can be achieved by swapping information bits due to the fact that only r different codes can be constructed.

Example 2.1 *We compare the hardware complexity for the encoder and the number of undetectable errors for the architectures that utilize different numbers of linear codes constructed by swapping information bits of the original code. Let C_1 be a $(39, 32)$ Hsiao code [17] whose parity check matrix is in the standard form $H = [I, P]$, where I is a 7×7 identity matrix and P is a 7×32 predictor matrix defined as follows. $C_i, 2 \leq i \leq 32$ is constructed by swapping the first and the i_{th} information bits of C_1*

$$P = \begin{bmatrix} 11111111000000100001001010000011 \\ 00001001111111110010010010000100 \\ 00010000000100001111111100110110 \\ 00100010001001011000000011111111 \\ 01100101010010010000111101101000 \\ 10000110100011101111100000001000 \\ 11011000111100000100000101010001 \end{bmatrix}.$$

The hardware complexity for the encoder, the number of undetectable errors and the maximum error masking probability of conditionally detectable errors for four schemes are shown in Table 1. The first column is the number of codes we randomly select to encode the messages. Row 1 corresponds to the case when only C_1 is used. 70 2-input gates and inverters are required to build the encoder. When randomly select 2

Table 1. Hardware complexity for the encoder, number of undetectable errors and maximum conditional error masking probabilities for schemes using different number of codes from Construction 2.1 ($n = 39, k = 32$)

Number of Codes	Number of Gates	Number of Undetectable Errors	Maximum conditional error masking probability
1	70	2^{32}	-
2	77	2^{31}	0.5
4	96	2^{29}	0.75
8	153	2^{25}	0.875

linear codes, we only need 7 *extra gates and the number of undetectable errors is reduced by* 50% *compared with the case when only a single linear code is used. Increasing the number of codes can further decrease the number of undetectable errors. However, this is at the cost of larger hardware overhead and worse conditional error masking probability.*

Another simple way to construct $C_i, 2 \leq i \leq l$ from C_1 is to circularly shift the redundant bits of C_1 as outlined below.

Construction 2.2 *Let C_1 be a (n, k) linear code with $r = n-k \leq k$ redundant bits. Denote by $H = [I, P]$ the parity check matrix of C_1, where I is a $r \times r$ identity matrix and P is a $r \times k$ predictor matrix. Assume that the rank of P is r. Construct C_2 by circularly shifting the redundant part of C_1 by 1 bit. If we randomly select C_1 and C_2 to encode the original messages with equal probability, the number of undetectable errors is 2^{k-r+1}. In addition, there are $2^{k+1} - 2^{k-r+2}$ errors which will be detected with probability* 0.5.

Proof Denote by $x = (x_1, x_2, \cdots, x_n)$ the codeword of a (n, k) linear code and assume that the first r bits are redundant bits. If $(x_1, x_2, \cdots, x_r, x_{r+1}, \cdots, x_n)$ belongs to both C_1 and C_2, then from the construction method of C_2 we know that $(x_2, x_3, \cdots, x_r, x_1, x_{r+1}, \cdots, x_n)$ also belongs to C_2. Hence the sum $(x_1 + x_2, x_2 + x_3, \cdots, x_r + x_1, 0, \cdots, 0)$ is another codeword of C_2. Because the information part is all $0's$, the redundant part should also be all $0's$. Thereby $x_1 + x_2 = 0, x_2 + x_3 = 0, \cdots, x_r + x_1 = 0$. So $x_1 = x_2 = \cdots = x_r \in \{0, 1\}$. Given the assumption that the rank of the predictor matrix P is r, all 2^r values are possible for the redundant part of the code. There are 2^{k-r} codewords that can generate each value of the redundant part. Hence the number of undetectable errors is equal to the size of the intersection of the two code which is 2^{k-r+1}. ■

Example 2.2 *$(x, (Px)^3)$ is a partially robust code, where $x \in GF(2^k)$, P is a $r \times k$ matrix in $GF(2)$, $Px \in GF(2^r)$ and $y^3 (y = Px)$ is a cube operation in Galois Field $GF(2^r)$ [6]. The number of undetectable errors of $(x, (Px)^3)$ code is 2^{k-r}. All conditionally detectable errors are masked with probability 2^{-r+1}. Compared with $(x, (Px)^3)$ code, Construction 2.2 has nearly the same order of the number of undetectable errors but requires much less hardware overhead to implement. As an illustrative example, Table 2 compares the hardware overhead for the encoder, number of undetectable errors as well as the maximum conditional error masking probability for these two codes when $n = 39, k = 32$. P is selected to be the same matrix as in Example 2.1. Only* 95 *2-input gates and inverters are required for the encoder of circularly shifting method while $(x, (Px))$ needs* 514. *The gap will become even larger when r increases.*

Construction 2.2 can be further improved to reduce the maximum conditional error masking probability. Let C_1 be a (n, k) linear code. Assume that we can find m numbers $s_i, 1 \leq i \leq m < r = n-k$, such that $s_i, 1 \leq i \leq m$ and r are mutually prime. $C_i, 2 \leq i \leq m+1$ is constructed by circularly shifting the redundant part of C_1 by s_{i-1} bits. If we randomly select $C_i, 1 \leq i \leq m+1$ to encode the original messages with equal probability, the number of undetectable errors is 2^{k-r+1}, assuming the rank of the predictor matrix of C_1 is r. In addition, $(m+1) \cdot (2^k - 2^{k-r+1})$ errors will be detected with probability $\frac{1}{m+1}$.

When r is prime, s_i can be any integer in the range of $[1, r-1]$. In this case the maximum conditional error masking probability is $\frac{1}{r}$.

Example 2.3 *Let C_1 be a* $(7, 4)$ *linear perfect Hamming code. $r = n - k = 3$ is prime. Construct $C_i, 2 \leq i \leq 3$ by circularly shifting the redundant part of C_1 by $i - 1$ bits. Figure 1 shows the experimental error masking properties for all $2^7 - 1$ nonzero errors. For each error we encode* 2000 *arbitrarily selected messages. A C_i is randomly chosen to encode each of the message. As we can see from Figure 1, $2^{k-r+1} = 4$ errors are undetectable (including the all $0's$ vector). All*

Table 2. Comparison of $(x, (Px)^3)$ code and Construction 2.2

Codes	Number of Gates	Number of Undetectable Errors	Maximum conditional error masking probability
$(x, (Px)^3)$	514	2^{25}	2^{-6}
Construction 2.2	95	2^{26}	0.5

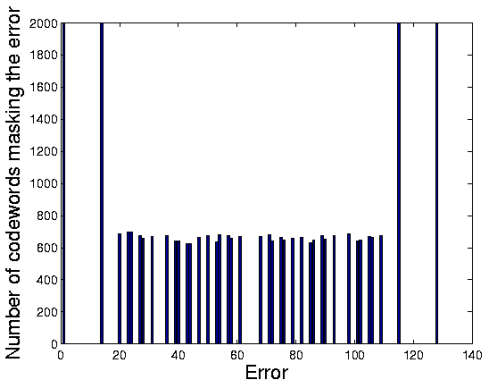

Figure 1. Error detection properties of circularly shifting and randomly selecting $(7, 4)$ Hamming code

the conditionally detectable errors are masked by about 660 *codewords, which means they are masked with probability* $\frac{1}{r} = \frac{1}{3}$.

The more codes we randomly select from, the better the error detection ability we can achieve. Randomly selecting more codes can result in either less undetectable errors (Example 2.1) or smaller maximum conditional error masking probability (Example 2.3). On the other hand, randomly selecting more linear codes means more complicated encoding and decoding strategies which may result in larger hardware overhead. An apparent question is how to randomly select codes to optimize either the number of undetectable errors or the maximum conditional error masking probability. If we randomly select l different linear codes, the best conditional error masking probability is $\frac{1}{l}$ which can be achieved when there are no errors belonging to more than one code except for the undetectable errors. The conditions for minimizing the number of undetectable errors when randomly selecting $l \leq \lfloor \frac{k}{r} \rfloor + 1$ linear codes is derived as follows.

It is easy to show that the smallest possible dimension of the intersection of $l \leq \lfloor \frac{k}{r} \rfloor + 1$ different (n, k) linear codes is $k-(l-1)r$, where $r = n-k$ is the number of redundant bits. For linear codes, every single redundant bit can be written as a seperate function of the information bits. Denote by $f_{i,j}$ the encoding function for the i_{th} code to generate the j_{th} redundant bit, where $1 \leq i \leq l, 1 \leq j \leq r$. The errors belonging to the intersection of all l linear codes should satisfy the following equations: $f_{1,j} = f_{2,j} = \cdots = f_{l,j}$ or equivalently $f_{1,j} + f_{2,j} = 0, f_{1,j} + f_{3,j} = 0 \cdots f_{1,j} + f_{l,j} = 0$ where $j = 1, 2, \ldots, r$. There are in total $(l-1)r$ equations and k unknowns (information bits of the code). The smallest possible dimension of the intersection is $k-(l-1)r$ which can be achieved when all these $(l-1)r$ equations are linearly independent. We next give a construction which can optimize the number of undetectable errors for any given $l \leq \lfloor \frac{k}{r} \rfloor + 1$.

Construction 2.3 *Suppose we want to construct* $l \leq \lfloor \frac{k}{r} \rfloor + 1$ *linear systematic codes such that the dimension of the intersection of these codes is minimum. Denote by* H_i *the parity check matrix of the* i_{th} *linear code. Without loss of generality, assume that the first* r *bits of any codeword are the redundant bits and the parity check matrices are in standard form* $H_i = [I_r, P_i]$, *where* I_r *is a* $r \times r$ *identity matrix and* P_i *is a* $r \times k$ *predictor matrix. Given* P_1, P_i $2 \leq i \leq l$ *can be constructed as follows.*

$$\begin{aligned} P_2 &= P_1 \oplus [I_r, 0_{r,k-r}] \\ P_3 &= P_1 \oplus [0_{r,r}, I_r, 0_{r,k-2r}] \\ &\vdots \\ P_l &= P_1 \oplus [0_{r,(l-2)r}, I_r, 0_{r,(k-(l-1)r)}] \end{aligned}$$

where I_r *is a* $r \times r$ *identity matrix and* $0_{i,j}$ *is a* $i \times j$ *all zero matrix.*

Example 2.4 *In this example, we construct 2* $[10, 5]$ *linear systematic codes such that the intersection of the two codes contains only the* $0's$ *vector. We select the first code* C_1 *to be a shortened Hamming code with the following parity check matrix.*

$$H_1 = \begin{bmatrix} 1000010101 \\ 0100011111 \\ 0010011011 \\ 0001000111 \\ 0000100101 \end{bmatrix}.$$

According to Construction 2.3, the parity check matrix of the second code can be computed as follows:

$$H_2 = \begin{bmatrix} 1000010101 \\ 0100011111 \\ 0010011011 \\ 0001000111 \\ 0000100101 \end{bmatrix} + \begin{bmatrix} 0000010000 \\ 0000001000 \\ 0000000100 \\ 0000000010 \\ 0000000001 \end{bmatrix}.$$

It is easy to verify that the dimension of the intersection is $k - r = 0$. The only vector belonging to both codes is the all $0's$ vector.

In Construction 2.3, $P_i, 2 \leq i \leq l$ is built by flipping r bits of r columns in the original predictor matrix P_1, one bit for each column. Generally speaking, this method cannot guarantee that all the linear codes have the same distance as the original code C_1. In the above example, the distance of C_2 is 2 instead of 3. However, by carefully selecting the parity check matrix for C_1 or adjusting the flipping positions, it is possible to make the other linear codes have the same distance as C_1. For instance, we can construct another $(10, 5)$ linear code C_2^* with the parity check matrix computed as follows. The minimum distance of C_2^* becomes 3 in this case while the intersection of C_1 and C_2^* still contains only the all $0's$ vector.

$$H_2^* = \begin{bmatrix} 1000010101 \\ 0100011111 \\ 0010011011 \\ 0001000111 \\ 0000100101 \end{bmatrix} + \begin{bmatrix} 0000010000 \\ 0000000010 \\ 0000000100 \\ 0000001000 \\ 0000000001 \end{bmatrix}.$$

3 General Analysis of Fault Detecion Ability of Multilinear Codes

A big difference between linear codes and the proposed constructions based on randomly selecting multiple linear codes is that our method has conditionally detectable errors. The detection of errors is message dependent. If the error is masked by one codeword, it is still possible that it will be detected by another codeword of a different code at the next moment. The longer the same error stays, the higher the detection probability is. Thereby in channels where errors tend to repeat themselves, our method has higher error detection ability than classical linear codes.

For applications utilizing cryptographic devices, we are more concerned about the fault detection ability of the code. The same fault may manifest itself as different error patterns at the output of the devices. When the same fault stays for t consecutive clock cycles, a general analysis of the fault detection abilities of the proposed method is shown in the next theorem.

Theorem 3.1 *Let $C_1, C_2 \cdots C_L$ be L different linear (n, k) codes. Assume that single or multiple faults stay for $t = aL + b, a \geq 0, 0 \leq b \leq L - 1$ consecutive clock cycles and may manifest themsevles as s different error patterns e_i, $1 \leq i \leq s \leq 2^n$ with probability $p(e_i)$ respectively.(We assume e_i may be the all $0's$ vector.) $P_j = \sum_{e_i \in C_j, 1 \leq i \leq s} p(e_i), 1 \leq j \leq L$ is the probability that faults manefest themselves as errors which are codewords of C_j. Denote by W_t the probability that faults are not detected after t clock cycles. If we circularly select $C_1, C_2 \cdots C_L$ to encode the message at every clock cycle, $W_t = \prod_{1 \leq j \leq L} P_j^{a+1-H(j-b-1)}$, where $H(j - b - 1)$ is the unit step function. If we randomly select the codes, $W_t = (\frac{1}{L} \sum_{i=1}^{L} P_i)^t$.*

Circularly selecting codes at each clock cycle is not suitable for cryptography applications because the attackers can circumvent the protection schemes if he knows what codes are used at each clock cycle. The advantage of circularly selecting, however, is that every error staying for at least L consecutive clock cycles can be 100% detected if the intersection of the L codes contains only the all $0's$ vector. When $L = 2$, all nonzero errors can be detected after staying for at most two clock cycles as long as C_1 and C_2 are non-overlapping and this is very useful for applications related to many communication channels and some computational channels where errors tend to repeat themselves with high probability, e.g. linear computational network consisting of XOR gates only.

To demomstrate the advantage of the proposed method, we compare the fault masking probability after t consecutive clock cycles for three error protection schemes for linear networks. The first one is based on a single $(20, 15)$ shortened Hamming code. Denote it by C_1. The second one utilizes four $(20, 15)$ linear codes $C_i, 1 \leq i \leq 4$ whose intersection contains only the all $0's$ vector. For the construction of $C_i, 2 \leq i \leq 4$, please refer to Construction 2.3. The third method is based on the $(x, (Px)^3)$ partially robust code. We can select P to be the same predictor matrix as for C_1.

To simplify the analysis. We assume that a stuck at fault occurs in the linear network. The fault manifests itself as the same nonzero error e at the output of the network with probability 0.5. If e is a codeword of C_1, after t clock cycles the error will be masked by the shortened Hamming code with probability 1. For method 2, we randomly select $C_i, 2 \leq i \leq 4$ with equal probability. If e belongs to the intersection of 3 codes, it will be masked with probability 0.875^t after t clock cycles according to Theorem 3.1. If e only belongs to one code, the error masking probability after t clock cycles is 0.625^t. The partially robust code

$(x, (Px)^3)$ has $2^{k-r} = 2^{10}$ undetectable errors. If e is undetectable by $(x, (Px)^3)$, it will be masked with probability 1 regardless of t. If e is conditionally detectable by $(x, (Px)^3)$, the error masking probability after t clock cycles is $(0.5 + 0.5 \cdot 2^{-r+1})^t$.

Figure 2 plots the fault masking probabilities after 10 clock cycles for the three alternatives. As expected, when considering the worst case fault masking probabilities, linear code is much worse than the other two. The fault will be masked no matter how many clock cycles it stays if it manifests as a codeword of the linear code. The method based on multilinear codes is much better than that based on single linear code. The performance of multilinear codes also depends on how the fault manifests itself. The less codes the manifested error belongs to, the better the fault detection ability is. One disadvantage of $(x, (Px)^3)$ code is that it still has undetectalbe errors. Even if the manifested error is conditionally detectable by $(x, (Px)^3)$, the fault masking probability is only a bit smaller than the best fault masking probability of multilinear codes. Given the fact that $(x, (Px)^3)$ code requires much more hardware overhead to implement, we claim multilinear codes are more promising alternatives in practice.

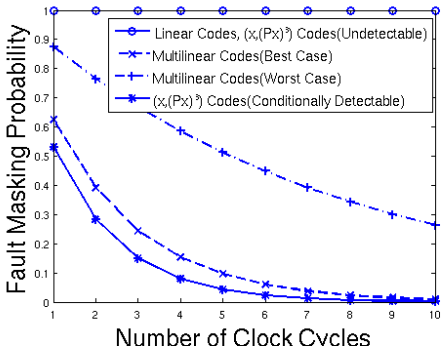

Figure 2. Comparison of fault masking probability after t clock cycles

4 Conclusion

We presented a new of class of error detection codes against adversarial errors. Compared to robust codes the proposed codes achieve similar error detection capabilities at much reduced cost due to their linearity.

References

[1] D. Boneh, R. A. Demillo, and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. In *Journal of Cryptology 14(2)*, pages 101–119, 2001.

[2] C. Carlet and C. Ding. Highly nonlinear mappings. *Journal of Complexity*, 20(2-3), 2004.

[3] G. Gaubatz, B. Sunar, and M. G. Karpovsky. Nonlinear residue codes for robust public-key arithmetic. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC '06)*, 2006.

[4] M. Karpovsky, K. Kulikowski, and A.Taubin. Differential fault analysis attack resistant architectures for the advanced encryption standard. Proc. IFIP World Computing Congress, Cardis, pages 177–193, Aug 2004.

[5] M. Karpovsky, K. Kulikowski, and A.Taubin. Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard. Proc. Int. Conference on Dependable Systems and Networks (DNS 2004), July 2004.

[6] M. G. Karpovsky and A. Taubin. A new class of nonlinear systematic error detecting codes. *IEEE Trans Info Theory*, 50(8):1818–1820, 2004.

[7] M. G. K. Konrad J. Kulikowski, Zhen Wang. Comparative analysis of robust fault attack resistant architectures for public and private cryptosystems. In *FDTC*, pages 41–50, 2008.

[8] G. Piret and J.-J. Quisquater. A differential fault attack technique against spn structures, with application to the AES and KHAZAD. In *CHES*, pages 77–88, 2003.

[9] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.

[10] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Lecture Notes in Computer Science*, 1109:104–113, 1996.

[11] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Delay-based Circuit Authentication and Applications. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 294–301, 2003.

[12] R. Posch. Protecting Devices by Active Coating. *Journal of Universal Computer Science*, 4(7):652–668, 1998.

[13] B. Skoric, S. Maubach, T. Kevenaar, and P. Tuyls. Information-theoretic Analysis of Coating PUFs. Cryptology ePrint Archive, Report 2006/101, 2006.

[14] Sergey Skorobogatov, Optical Fault Induction Attacks. Cryptographic Hardware and Embedded Systems Workshop (CHES-2002), LNCS 2523, Springer-Verlag, ISBN 3-540-00409-2, pp. 2–12, 2007.

[15] Sergey Skorobogatov, Data Remanence in Flash Memory Devices. Cryptographic Hardware and Embedded Systems Workshop – CHES 2005: 7th International Workshop, Edinburgh, UK, August 29-September 1, pp. 339–353, 2005.

[16] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, Claire Whelan, The Sorcerers Apprentice Guide to Fault Attacks, URL: `http://eprint.iacr.org/2004/100.pdf`.

[17] M.Y.Hsiao, A Class of Optimal Minimum Odd-weight-column SEC-DED Codes. IBM Journal of Research and Development, 14(4):395-401, 1970.