Robust FSMs for Cryptographic Devices Resilient to Strong Fault Injection Attacks

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

Abstract

The security of the cryptographic device may be compromised if the FSM of the system is not properly protected [1]. FSM protection architectures based on linear codes cannot provide a guaranteed level of protection under the assumption of a strong attack model. In this paper, we propose secure FSM architectures based on the idea of randomly selecting one code from a set of codes for each encoding and decoding operation. Assuming that the attacker is able to inject specific error patterns, randomly selecting one code from L codes as described in the paper can reduce the chance for the attacker to conduct a successful attack by a factor of L. The proposed techniques can achieve much higher security level than architectures based on linear codes with reasonable hardware overhead for cryptography applications ($120\% - 130\%$ for the protection of the FSM for the Montgomery ladder algorithm).

1 Introduction

The security of the cryptosystems is threatened by side-channel attacks such as timing analysis attacks [2], power analysis attacks [3] and fault injection attacks [4]. Unlike other forms of side-channel attacks, fault based attacks are often active and hence adaptive. The adaptive nature combined with the vast arsenal of fault injection methods and techniques available to an attacker makes fault injection attacks one of the most powerful side-channel attacks and provides a big challenge for the design of cryptographic devices.

Most of the current research on protecting cryptographic devices against fault injection attacks target at the data path of the system. Concurrent error detection (CED) techniques, for example, are often used to verify the data integrity at the output of the cryptosystems for the purpose of detecting maliciously introduced errors [5, 6, 7]. On the contrary, few papers have been published on protecting the control circuit (e.g. FSM, pipeline) of the device. In a recent work [1], the author showed that by injecting faults into the FSM of the cryptographic device implementing the Montgomery ladder algorithm, the attacker can still reveal the secret key of the system even if the data path is properly protected. Thereby, the security of the FSMs should also be considered when designing cryptosystems resistant to fault injection attacks.

The design of reliable FSM architectures tolerant to naturally introduced errors (e.g. soft errors) are well studied in the community [8, 9, 10]. Most of these reliable FSM architectures are based on linear codes (e.g. TMR, parity prediction) and assume a specific error model where errors with small multiplicities are more probable. They cannot provide a guaranteed level of protection against fault injection attacks under strong attack models since errors introduced by an attacker can be unpredictable.

In [6, 11, 12], *robust codes* that can provide equal protection against all error patterns are proposed as an alternative to linear codes for the protection of cryptographic devices against fault injection attacks. However, the advantage of robust codes lies on the assumption that all codewords are equi-probable. For FSMs only some of the codewords correspond to valid states. Moreover, in most of the cases the probability of valid states is not uniformly distributed. Due to these two inherent characteristics of FSMs, protection architectures based on single robust codes cannot be directly applied to build secure FSMs.

As a solution for the protection of FSMs against strong attackers, the authors in [13] proposed to use fingerprints generated by physically unclonable functions (PUFs) to verify the transition of the FSM when it is in operation. However, the architecture can only be applied to known-path state machines where the state transitions do not depend on the external inputs. In [14], a secure FSM architecture based on nonlinear functions and randomized maskings was proposed. While interesting and efficient as a countermeasure against strong attackers, the method requires a high hardware overhead.

In this paper, we propose to use multi-code techniques to protect FSMs against fault injection attacks. The technique

is based on the idea of randomly selecting one code from a set of codes for each encoding and decoding operation. The proposed architectures can provide a guaranteed level of security under strong attack models and require less hardware overhead than other existing secure FSM architectures described in the literature.

The rest of the paper is organized as follows. In Section 2, the attack model we use in this paper is described. In Section 3, first, the general architecture of FSMs protected by linear codes is presented. Its limitations for the protection of cryptosystems against strong attackers are shown. Second, two secure FSM architectures based on multi-code techniques are presented and their error detection properties are analyzed. In Section 4, we compare the performance and the estimated overhead for different alternatives.

2 Attack Model

We assume that the attacker knows the detailed implementation of the secure FSM architectures. To reveal the secret information of the system, the attacker tries to force the FSM into a faulty but valid state by injecting faults into system registers or combinational networks resulting in additive errors in the system registers. Denote by x the content of a system register and e the error vector introduced by the attacker, the distorted content is $\tilde{x} = x \oplus e$, where $\oplus$ is the bit-wise XOR operation.

We further assume that the attacker cannot first read the contents from the registers and then decide how the faults will be injected during the same clock cycle. However, the attacker may be able to inject any specific error patterns, which is possible using the advanced fault injection mechanisms [15]. Moreover, the attacker may know the next state of the FSMs before he injects faults. This is probable for some known-path FSMs where the state transitions are not dependent upon the external inputs [13].

To conduct a more comprehensive comparison of different alternatives, we analyze their error detection capabilities for three different attack models.

A1: The attacker injects random errors with uniform distribution.

A2: The attacker has high spatial fault injection resolution and is able to inject any specific error patterns. But he does not know the next state of the FSM before he injects faults.

A3: The attacker has high spatial and temporal fault injection resolution. He knows the next state before he injects faults and is able to introduce any specific error patterns.

We further notice that the attacker may have different goals of conducting fault injection attacks.

G1: Force the FSM into an arbitrary faulty(but valid) state.

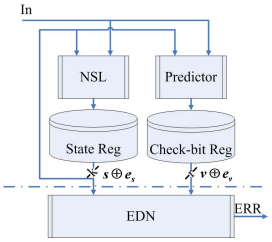


Figure 1: Secure FSM Architectures Based on Linear Error Detection Codes

G2: Force the FSM into a certain faulty state as desired by the attacker.

We denote by Q_1 and Q_2 the probability that the attack is successful for the above two situations respectively. The performance of the protection architectures will be evaluated by computing Q_1 and Q_2 under all the three attack models.

3 Secure FSM Architectures

Throughout the rest of the paper, we denote by (n, k) a binary systematic code with length n and dimension k. Let S be the set of binary vectors representing the valid states of the FSM and s an element in S. For situation $G2$, let s' be the certain state that the attacker wants to force the FSM into. Let $p(s)$ be the probability that the next state of the FSM is s, assuming the external inputs are uniformly distributed. (When treating FSMs as Markov chains, $p(s)$ is the stationary distribution of the chain.) We will only present the protection architectures for the computation of the next state functions. The computation of the output functions can be protected using similar techniques.

3.1 Architectures Based on Linear Codes

Figure 1 shows the general secure FSM architecture based on a (n, k) systematic code C, which consists of two registers and three combinational networks. The NSL block computes the next state vector based on the current state and the external inputs. The predictor computes the redundant bits $v \in GF(2^r)$ of the code, where $r = n - k$. The state register stores the next state vector $s \in GF(2^k)$. The check-bit register stores the redundant bits v. The non-distorted outputs of the two registers compose a codeword of C. Denote by e_s and e_v the additive errors occurring to the state vectors and the redundant bits respectively. At the beginning of each clock cycle, EDN will verify whether $(s \oplus e_s, v \oplus e_v)$ is a codeword of C and $s \oplus e_s$ is a valid state vector. If either of the two verifications is failed, errors are detected and ERR will be asserted.

Although architectures based on linear codes can provide a satisfactory protection against most of the naturally introduced errors, when facing an attacker with advanced fault injection mechanisms, the security level of the system cannot be guaranteed. (Due to the lack of space, the proof for Theory 3.1 is omitted.)

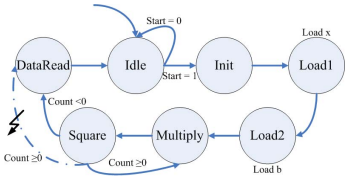


Figure 2: State Transition Diagram of the FSM for the Montgomery Ladder Algorithm [1]

Theorem 3.1 *Let $|S|$ be the number of valid states of the FSM. Let $S_{e_s} = \{s_1 \in S | \exists s_2 \in S, s_1 \oplus s_2 = e_s\}$.For architectures based on any (n, k) linear error detection code, Q_1 and Q_2 for different attack models described in Section 2 are as stated below.*

A1: $Q_1 = \frac{|S|-1}{2^n}$, $Q_2 = \frac{1-p(s')}{2^n}$;
A2: $Q_1 = \max_{e_s}\{\sum_{s \in S_{e_s}} p(s)\}$, $Q_2 = \max_{s \neq s'}\{p(s)\}$;
A3: $Q_1 = Q_2 = 1$.

We next present an example of utilizing linear codes for the protection of the FSM for the Montgomery ladder algorithm, which is widely used in RSA and elliptic curve cryptosystems. The state transition diagram of the Montgomery ladder algorithm is shown in Figure 2. The algorithm is for the computation of $y = x^b \text{mod} N$, where x is the original message and b is the m-bit secret key of the cryptosystems. After loading x and b into system registers, the FSM takes m clock cycles to finish the computation. A possible attack scenario was presented in [1]. The author showed that by forcing the FSM into the *DataRead* state before all the computation is completed, b can be easily revealed one bit per time. (Fore more details about the Montgomery ladder algorithm and the attack scenario, please refer to [1, 16]).

The state assignment for the FSM is shown in Table 1. $S = \{001, 010, 011, 100, 101, 110, 111\}$. S_{e_s} can be derived from the definition in Theorem 3.1, e.g. $S_{001} = \{010, 011, 100, 101, 110, 111\}$. Assuming a public key size of 17-bit for RSA ($m = 17$), $p(s)$ is shown in the last column of Table 1.

Example 3.1 *The FSM for the Montgomery ladder algorithm can be protected using a $(6, 3)$ linear Hamming code whose parity check matrix is*

$$H = \begin{bmatrix} 100101 \\ 010110 \\ 001011 \end{bmatrix}.$$

Table 1: State Assignment of the FSM for the Montgomery Ladder Algorithm

Valid State	State Vector	$p(s)$
Idle	001	1/39
Init	010	1/39
Load1	011	1/39
Load2	100	1/39
Multiply	101	17/39
Square	110	17/39
DataRead	111	1/39

To reveal the secret key b of the cryptosystem, the attacker tries to force the FSM into state 111 (DataRead) before the computation is completed. Thereby $s' = 111$. Given the state assignment in Table 1, $\max_{s \neq s'}\{p(s)\} = 17/39 = 0.4359$ (s can be either 101 or 110). $\max_{e_s}\{\sum_{s \in S_{e_s}} p(s)\} = 38/39 = 0.9744$. e_s can be any vector in $\{001, 010, 011, 100, 111\}$. Q_1 and Q_2 under different attack models are shown in Table 2 (Section 4). Under attack models $A2$ and $A3$, Q_1 is close to 1 and Q_2 is at least 0.4359. Obviously, FSM protection architectures based on single linear codes cannot provide enough protection in these situations.

Remark 3.1 • *We note that for the above FSM some mistransitions of the states may not reveal secret information of the cryptosystem. However, to compare different alternatives for more general situations, we still use the definition of Q_1 and Q_2 given in Section 2.*
• *Since m clock cycles are required to finish the computation after x and b are loaded, an extra counter is needed to store the number of passed clock cycles (see $Count$ in Figure 2). This counter should be protected using similar techniques presented in this paper.*

From Theorem 3.1 it is clear that for FSM protection architectures based on single linear codes, Q_1 and Q_2 do not depend on the code type. When the attacker can only inject random errors ($A1$), Q_1 and Q_2 are affected by n and $p(s)$. When the attacker is able to inject specific error patterns ($A2$), Q_1 and Q_2 are affected by $p(s)$ and S_{e_s}. In general, to achieve a higher security level of FSMs, $\max_{s_1 \neq s_2}\{p(s_1) - p(s_2)\}$ should be as small as possible. If the attacker also knows the next sate of the FSM ($A3$), any protection architectures based on single linear codes stand no chance.

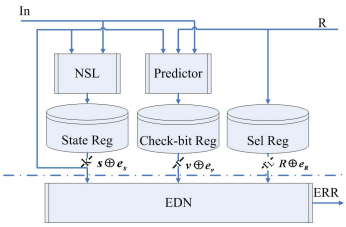
In the next section, we will show a possible solution for the protection of FSMs against strong attackers, which is based on multi-code techniques.

3.2 Architectures Based on Multi-Code Techniques

The basic idea of multi-code techniques is to randomly select one code from a set of codes for each encoding and decoding operation. The general secure FSM architecture based on multi-code techniques is shown in Figure 3. At each clock cycle, a selection signal $R \in GF(2^{r_L})$ is generated by a (pseudo) random number generator, which is integrated in most of the cryptographic devices. Based on the value of R, the predictor selects a code from L, $L \leq 2^{r_L}$ different codes to encode the next state vector s. R is stored in a separate register and is used to verify s at the beginning of the next clock cycle.

Remark 3.2 *We assume that countermeasures are implemented in the cryptographic devices to prevent the attacker*

Figure 3: Secure FSM Architectures Based on Multi-code Techniques



from tampering with the clock signals or the random number generators.

3.2.1 Design of Secure FSMs Using Multilinear Codes

Multilinear codes were proposed in [17, 18] as an alternative to robust codes for the protection of cryptographic devices. These codes have similar error detection capabilities to robust codes and require less hardware overhead. In this section, we will show that multilinear codes can also be applied to build secure FSMs resistant to fault injection attacks assuming a strong attack model.

Given a set of linear codes, if the intersection of any two of these codes contains only the all-zero vector, we say that these codes are **non-overlapping**. The next theorem shows that by randomly selecting one code from a set of non-overlapping linear codes for each encoding and decoding operation, we can effectively reduce the chance for the attacker to conduct a successful fault injection attack.

Theorem 3.2 *Let $C_1, C_2, \cdots, C_L$ be L different (n,k) linear codes satisfying $C_i \cap C_j = \{\mathbf{0} \in GF(2^n)\}, i \neq j, 1 \leq i,j \leq L \leq 2^{r_L}$. Let $R \in GF(2^{r_L})$ be the randomly generated selection signal with uniform distribution and $e_R \in GF(2^{r_L})$ be the additive error in the register storing the value of R. Let Γ be the set of all valid selection signals, $|\Gamma| = L$. Assume that for every nonzero e_R, there is at most one pair of R, R' satisfying $R, R' \in \Gamma, R \oplus R' = e_R$. If we randomly select C_i for each encoding and decoding operation, Q_1 and Q_2 for different attack models are as stated below.*

A1: $Q_1 = \frac{L(|S|-1)}{2^{n+r_L}}$, $Q_2 = \frac{L(1-p(s'))}{2^{n+r_L}}$;
A2: $Q_1 = \frac{1}{L}\max_{e_s}\{\sum_{s\in S_{e_s}} p(s)\}$,
$Q_2 = \frac{1}{L}\max_{s\neq s'}\{p(s)\}$;
A3: $Q_1 = Q_2 = \frac{1}{L}$.

Proof Denote by H_i the parity check matrix of the i_{th} linear code. Let (e_s, e_v) be the error injected to the codeword and e_R be the error injected to the register storing the selection signal R. $G1$ is achieved iff $H_{R\oplus e_R}(s\oplus e_s, v\oplus e_v) = 0$ and $s \oplus e_s$ is a valid state, $e_s \neq 0$. $G2$ is achieved iff $H_{R\oplus e_R}(s \oplus e_s, v \oplus e_v) = 0$ and $s \oplus e_s = s'$.

A1: The probability that $e \oplus e_s$ is a valid state is $\frac{|S|-1}{2^k}$. The probability that $s \oplus e_s = s', e_s \neq 0$ is $\frac{1-p(s')}{2^k}$. For each e_s, there are L pairs of e_R and e_v satisfying $H_{R\oplus e_R}(s \oplus e_s, v \oplus e_v) = 0$. Thereby, $Q_1 = \frac{L(|S|-1)}{2^{n+r_L}}$, $Q_2 = \frac{L(1-p(s'))}{2^{n+r_L}}$.

A2: The maximum probability that $s \oplus e_s$ is a valid state for a given e_s is $\max_{e_s}\{\sum_{s\in S_{e_s}} p(s)\}$. The maximum probability that $s \oplus e_s = s'$ is $\max_{s\neq s'}\{p(s)\}$. If $e_R = 0$, $Q_1 = \frac{1}{L}\max_{e_s}\{\sum_{s\in S_{e_s}} p(s)\}$, $Q_2 = \frac{1}{L}\max_{s\neq s'}\{p(s)\}$. If $e_R \neq 0$, under the assumption of Theorem 3.2, there exists only one $R' \in \Gamma$ such that $R \oplus R' = e_R$. Rewrite H_i into the standard form $H_i = (I, P_i)$, where I is the $r \times r$ identity matrix and P_i is the $r \times k$ encoding matrix of C_i. Suppose $e = (e_s, e_v)$ converts an encoded state s in C_R into another encoded state s' in $C_{R'}$. Then $e_v = (P_R \oplus P_{R'})s \oplus P_{R'}e_s$. (The same $e = (e_s, e_v)$ can also convert s' in $C_{R'}$ back into s in C_R). For each $e = (e_s, e_v)$, there is at most one s satisfying the above equation. If the attacker inject nonzero e_R, it is easy to show that Q_1 and Q_2 will be no larger than the case when $e_R = 0$. Thereby, in this situation $Q_1 = \frac{1}{L}\max_{e_s}\sum_{s\in S_{e_s}} p(s)$, $Q_2 = \frac{1}{L}\max_{s\neq s'}\{p(s)\}$,

A3: In the last situation, following similar analysis we can show that $Q_1 = Q_2 = \frac{1}{L}$. ■

Remark 3.3 *Γ is a 2-robust code [6], i.e. each nonzero error is masked by at most two codewords of the code.*

Compared to the architectures based on single linear codes, architectures based on multilinear codes reduce Q_1 and Q_2 by a factor of L under attack models $A2$ and $A3$ thus largely increase the security level of the system.

Theorem 3.2 raises the question of how to construct multiple non-overlapping linear codes. Several constructions of multilinear algebraic codes were presented in [17]. For the simplest case of constructing two non-overlapping linear codes, one possible solution derived from [17] is as stated below.

Construction 3.1 *Let $H_1 = (I, P_1)$ be the parity check matrix of a $(2k, k)$ linear code C_1, where I is the $k \times k$ identity matrix and P_1 is a $k \times k$ encoding matrix. Let $H_2 = (I, P_1 \oplus I)$ be the parity check matrix of the second $(2k, k)$ linear code C_2. Then $C_1 \cap C_2 = \{\mathbf{0} \in GF(2^{2k})\}$.*

Remark 3.4 *The dimension of the intersection of any two (n,k) linear codes C_1 and C_2 is at least $k - r, r = n - k$. Thereby, if C_1 and C_2 are non-overlapping, $r \geq k$.*

A more general case of Construction 3.1 is as follows. Denote by $H_i = (I, P_i)$ the parity check matrix of a $(2k, k)$ linear code $C_i, 1 \leq i \leq L$. Let $P_{ij} = P_i \oplus P_j$. $C_i, 1 \leq i \leq L$ are non-overlapping as long as the rank of P_{ij} is k.

Example 3.2 *In this example, we use four non-overlapping $(6,3)$ linear codes $C_i, 1 \leq i \leq 4$ to protect the FSM for the*

Montgomery ladder algorithm. Let

$$P_1 = \begin{bmatrix} 101 \\ 110 \\ 011 \end{bmatrix}, P_2 = P_1 \oplus \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}$$

$$P_3 = P_1 \oplus \begin{bmatrix} 010 \\ 001 \\ 110 \end{bmatrix}, P_4 = P_1 \oplus \begin{bmatrix} 001 \\ 101 \\ 111 \end{bmatrix}$$

be the encoding matrices of the codes. It is easy to verify that $P_{ij}, 1 \le i, j \le 4$ has rank 3 and the intersection of any two of these codes contains only the all-zero vector. Let $r_L = 3$ and $001, 010, 100, 011$ be the selection signal for $C_i, 1 \le i \le 4$ respectively. ($\Gamma = \{001, 010, 100, 011\}$.) For every $e_R \neq 0$, there is at most one pair of $R, R' \in \Gamma$ satisfying $R \oplus R' = e_R$. Given the state assignment in Table 1, if we randomly select C_i for every encoding and decoding operation, Q_1 and Q_2 under different attack models are shown in Table 2 (Section 4). Compared to Example 3.1, the described method can reduce Q_1 and Q_2 by a factor of four under attack models $A2$ and $A3$. Higher security level can be achieved by randomly selecting from more linear codes (increase k and r if necessary).

3.2.2 Design of Secure FSMs Using Multirobust Codes

Similar multi-code techniques can also be applied to nonlinear robust codes. We next describe a secure FSM architecture based on **multirobust** codes, which has highly regular structures for the encoder and the decoder and results in comparable hardware overhead to architectures based on multilinear codes. The error detection capability of the proposed architecture will be analyzed and compared to other alternatives.

Theorem 3.3 *Let $C_i = \{(s, v) | s \cdot v = i - 1\}, 1 \le i \le 2^k$, where $s, v \in GF(2^k)$ and $\cdot$ is the multiplication in $GF(2^k)$. Assume that $\mathbf{0} \in GF(2^k)$ is not a valid state ($\mathbf{0} \notin S$). If we randomly select $C_i, 1 \le i \le 2^k$ for each encoding and decoding operation ($L = 2^k$), Q_1 and Q_2 under different attack models are as stated below.*

A1: $Q_1 = \frac{|S|-1}{2^{2k}}$, $Q_2 = \frac{1-p(s')}{2^{2k}}$;
A2: $Q_1 = \frac{1}{2^k} \max_{e_s} \{\sum_{s \in S_{e_s}} p(s)\}$,
$Q_2 = \frac{1}{2^k} \max_{s \neq s'} \{p(s)\}$;
A3: $Q_1 = Q_2 = \frac{1}{2^k}$.

Proof An error $e = (e_s, e_v)$ will be missed iff $(s \oplus e_s) \cdot (v \oplus e_v) = R \oplus e_R$. Since $v = \frac{R}{s}$, $s \neq 0$, the above equation can be re-written as $e_v \cdot s^2 \oplus (e_s \cdot e_v \oplus e_R) \cdot s \oplus e_s \cdot R = \mathbf{0}$.

A1: The probability that $e_s \oplus s$ is a valid state is $\frac{|S|-1}{2^k}$. The probability that $e_s \oplus s = s', s \neq s'$ is $\frac{1-p(s')}{2^k}$. For each $e = (e_s, e_v)$, there is only one e_R satisfying the above equation. Thereby $Q_1 = \frac{|S|-1}{2^{2k}}$. $Q_2 = \frac{1-p(s')}{2^{2k}}$.

A2: The maximum probability that $s \oplus e_s$ is a valid state for a given e_s is $\max_{e_s} \{\sum_{s \in S_{e_s}} p(s)\}$. The maximum probability that $s \oplus e_s = s'$ is $\max_{s \neq s'} \{p(s)\}$. For any fixed $e_s \neq 0, e_v$ and e_R, there is only one R for each s satisfying the error masking equation. Hence $Q_1 = \frac{1}{2^k} \max_{e_s} \{\sum_{s \in S_{e_s}} p(s)\}$. $Q_2 = \frac{1}{2^k} \max_{s \neq s'} \{p(s)\}$.
A3: Without the knowledge of R, Q_1 and Q_2 are at most $\frac{1}{2^k}$. ■

For a given number of redundant bits $r = k$, the architecture based on Theorem 3.3 has the maximum L thus can minimize Q_1 and Q_2 under attack models $A2$ and $A3$. The encoder of the proposed multirobust codes mainly contains an inverse operation and a multiplication in $GF(2^k)$. The EDN of the multirobust codes requires only one multiplication in $GF(2^k)$. Architectures based on Theorem 3.3 require less overhead than other existing secure FSM architectures utilizing nonlinear robust codes. (For instance, the architecture proposed in [14] requires at least 4 cubings and 2 multiplications in $GF(2^k)$).

Example 3.3 *The FSM of the Montgomery ladder algorithm can also be protected using eight (6.3) nonoverlapping nonlinear codes as described in Theorem 3.3. We still use the state assignment in Table 1. (Note that the all-zero vector is not a valid state.) If we randomly select from $C_i = \{(s, v) | s \cdot v = i - 1\}, 1 \le i \le 8$ for every encoding and decoding operation, Q_1 and Q_2 will be reduced by a factor of 8 under attack models $A2$ and $A3$ compared to architectures based on linear codes (Table 2). Similar to protection methods based on multilinear codes, higher security level can be achieved by randomly selecting from more codes (increase k and r).*

4 Comparison of Secure FSM Architectures

The error detection capabilities of different secure FSM architectures (Example 3.1, 3.2, 3.3) for the Montgomery ladder algorithm are shown in Table 2. All the data are verified via simulation in MATLAB. When the attacker is able to inject specific error patterns (attack models $A2$ and $A3$), architectures based on multilinear codes and multirobust codes can reduce Q_1 and Q_2 by a factor of L and 2^k

Table 2: Q_1 and Q_2 of Secure FSM Architectures for the Montgomery Ladder Algorithm

Attack Models	Code	Q_1	Q_2
$A1$	Linear(Exp. 3.1)	0.0938	0.0152
	Multilinear(Exp. 3.2)	0.0469	0.0076
	Multirobust(Exp. 3.3)	0.0938	0.0938
$A2$	Linear(Exp. 3.1)	0.9744	0.4359
	Multilinear(Exp. 3.2)	0.2436	0.1090
	Multirobust(Exp. 3.3)	0.1218	0.0545
$A3$	Linear(Exp. 3.1)	1	1
	Multilinear(Exp. 3.2)	0.2500	0.2500
	Multirobust(Exp. 3.3)	0.125	0.125

Table 3: Structure, Hardware and Power Consumption Overhead of Encoders and EDN of Different Alternatives

code	Structure		Overhead	
	Linear	Nonlinear	Hardware	Power
Linear	$\surd$	–	44.37%	37.66%
Multilinear	$\surd$	–	129.0%	94.13%
Multirobust	–	$\surd$	119.5%	133.57%

respectively compared to architectures based on single linear codes (in our case $L = 4$, $2^k = 8$). Thereby, under the assumption of a strong attack model, the proposed methods can provide a guaranteed level of security which cannot be achieved using architectures based on single linear codes.

To estimate the overhead, the three designs were modeled in VERILOG, synthesized using Cadence RTL Compiler and placed & routed using Cadence Encounter based on Nangate 45nm open cell library [19]. The structure and the overhead of the encoder and EDN of different alternatives are shown in Table 3. All the structures are linear except for the encoder and EDN of architectures based on multirobust codes. Generally speaking, the implementation of nonlinear operations requires more hardware overhead than the implementation of linear operations. However, due to the regular structure of the codes, the secure FSM architecture based on multirobust codes has comparable hardware overhead to architectures based on multilinear codes for small k. When k is larger, architectures based on multilinear codes can have less hardware overhead than those based on multirobust codes and can be used as a tradeoff between overhead and security level of the system.

5 Conclusions

In this paper, we presented secure FSM architectures based on multi-code techniques which can provide a guaranteed level of protection under the assumption of strong attack models. We proved that if the attacker is able to inject specific error patterns, randomly selecting among L codes for each encoding and decoding operation as described in the paper can reduce the chance for the attacker to conduct a successful attack by a factor of L compared to architectures based on linear codes. The proposed techniques were utilized to protect the FSM of the Montgomery ladder algorithm, which can reduce the chance for the attacker to conduct a successful attack by a factor of up to 8 assuming $k = r = 3$. The hardware overheads of the proposed architectures are $120\% - 130\%$ and are less than other secure FSM architectures based on nonlinear codes which are resistant to strong attackers. The security level of systems protected by multi-code techniques can be further improved by increasing L.

References

[1] B. Sunar, G. Gaubatz, and E. Savas, "Sequential circuit design for embedded cryptographic applications resilient to adversarial faults," *IEEE Transactions on Computers*, vol. 57, pp. 126–138, 2007.

[2] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Lecture Notes in Computer Science*, 1996.

[3] P. Kocherand, J. Jaffe, and B. Jun, "Differential power analysis," in *Lecture Notes in Computer Science*, 1999.

[4] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerers apprentice guide to fault attacks," 2002.

[5] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "A parity code based fault detection for an implementation of the advanced encryption standard," ser. Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems(DFT'02), 2002.

[6] K. Kulikowski, Z. Wang, and M. Karpovsky, "Comparative analysis of robust fault attack resistant architectures for public and private cryptosystems," in *Fault Diagnosis and Tolerance in Cryptography, 2008. FDTC '08. 5th Workshop on*, Aug. 2008, pp. 41–50.

[7] 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)*, 2006.

[8] R. Rochet, R. Leveugle, and G. Saucier, "Analysis and comparison of fault tolerant FSM architecture based on SEC codes," in *Defect and Fault Tolerance in VLSI Systems, 1993., The IEEE International Workshop on*.

[9] A. Krasniewski, "Concurrent error detection for finite state machines implemented with embedded memory blocks of SRAM-based FPGAs," *Microprocessors and Microsystems*, 2008.

[10] S. Baranov, I. Levin, O. Keren, and M. Karpovsky, "Designing fault tolerant FSM by nano-PLA," *IEEE International On-Line Testing Symposium*, vol. 0, pp. 229–234, 2009.

[11] M. Karpovsky and A. Taubin, "New class of nonlinear systematic error detecting codes," *Information Theory, IEEE Transactions on*, Aug. 2004.

[12] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin, "Robust codes and robust, fault-tolerant architectures of the advanced encryption standard," *J. Syst. Archit.*, vol. 53, no. 2-3, pp. 139–149, 2007.

[13] G. Hammouri, K. Akdemir, and B. Sunar, "Novel puf-based error detection methods in finite state machines," in *Information Security and Cryptology(ICISC) 2008*, 2009, pp. 235–252.

[14] K. D. Akdemir, G. Hammouri, and B. Sunar, "Nonlinear error detection for finite state machines," in *WISA*, 2009, pp. 226–238.

[15] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," in *CHES, 2002*.

[16] M. Joye and S.-M. Yen, "The montgomery powering ladder," in *CHES 2002, LNCS*.

[17] Z. Wang, M. Karpovsky, and B. Sunar, "Multilinear codes for robust error detection," in *On-Line Testing Symposium, IOLTS 2009.*

[18] Z. Wang, M. Karpovsky, B. Sunar, and A. Joshi, "Design of reliable and secure multipliers by multilinear arithmetic codes," in *Information and Communications Security*, 2009, pp. 47–62.

[19] "Nangate 45nm open cell library," `http://www.nangate.com`.