Algebraic Manipulation Detection Codes and Their Applications for Design of Secure Cryptographic Devices

Zhen Wang and Mark Karpovsky
Reliable Computing Laboratory, Boston University, Boston , USA
{*lark, markkar*}*@bu.edu*

***Abstract*—Cryptographic devices are vulnerable to fault injection attacks. All previous countermeasures against fault injection attacks based on error detecting codes assume that the attacker cannot simultaneously control the fault-free outputs of a device-under-attack and error patterns. For advanced attackers who are able to control both of the above two aspects, traditional protections can be easily compromised. In this paper, we propose optimal algebraic manipulation detection (AMD) codes based on the nonlinear encoding functions and the random number generators. The proposed codes can provide a guaranteed high error detecting probability even if the attacker can fully control the fault-free outputs of a device-under-attack as well as the error patterns. As a case study, we present the protection architectures based on AMD codes for multipliers in Galois fields used for the elliptic curve cryptography. The results show that the proposed architecture can provide a very low error masking probability at the cost of a reasonable area overhead. The protected multiplier has no latency penalty when the predictor is pipelined.**

***Keywords*-Security; Error Detecting Codes; Fault Injection Attacks; Cryptographic Hardware;**

I. Introduction

The security of modern cryptographic devices is threatened by side-channel attacks such as timing analysis attacks [1], power analysis attacks [2] and fault injection attacks [3], [4]. Unlike other forms of side-channel attacks, fault injection 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 complicate the design of secure cryptographic devices. The attack described in [5], for example, requires only 2 faulty cipher-texts to retrieve from AES all 128-bits of the secret key.

To protect cryptographic devices against fault injection attacks, redundancy based on error detecting codes (EDC) is often added. Most of the existing proposals of EDC architectures are based on linear codes such as 1-d parity codes [6], duplication codes and Hamming codes. Proposals for secure architectures based on linear codes usually make assumptions about how the faults manifest as non-zero errors at the output of a device-under-attack. The error detecting capabilities and the security level of these architectures largely depend on the accuracy of the error model. For example, cryptographic devices protected by linear 1-d parity codes can provide 100% error detection probabilities if all injected faults manifest as errors with odd multiplicities at the output of the device. However, if the attacker can inject faults that manifest as errors with even multiplicities, the protection based on linear 1-d parity codes can be easily bypassed. Similarly, for the protection based on duplication [7], identical errors in both copies are undetectable.

Due to the above inherent weakness, secure architectures based on linear codes can only provide satisfactory protection assuming a weak attacker model, in which the attacker can not precisely control the location and the timing of the injected faults. As more advanced fault injection mechanisms are proposed, protection architectures based on linear codes are no longer sufficient. For example, the author in [3] showed that any arbitrary bit location in memory could be modified via an optical induction attack. Against such attackers linear codes stand no chance.

As an alternative to linear codes, robust codes based on nonlinear encoding functions were proposed [8], [9], [10], [11], [12]. Robust codes can provide nearly equal protection against all error patterns. The error masking probabilities for robust codes are upper-bounded by very small numbers for all non-zero errors. Compared to architectures based on linear codes, architectures based on robust codes can provide a guaranteed protection even if the attacker has a high spatial resolution of fault injection and can inject faults which generate specific error patterns. Variants of robust codes – partially robust and minimum distance robust codes – were proposed in [11], [12], which allow tradeoffs in terms of robustness and hardware overhead.

One limitation of robust codes is that these codes assume the output of the cryptographic device is almost uniformly distributed and is not controllable by the attacker, i.e. the attacker cannot select the fault-free outputs of the device-under-attack during fault injection attacks.

In this paper, we present constructions of **algebraic manipulation detection codes** based on nonlinear encoding functions and true random number generators. We show the architectures based on these codes, for which the probability

The work of the second author has been partially supported by the NSF grant CNS 1012910.

of a successful fault injection attack is upper-bounded by a very small number even assuming the above strong attacker model.

The rest part of the paper is organized as follows. In Section II, we describe the conception of **strong security** and present the attacker model used throughout the paper. In Section III, the definitions and bounds for AMD codes are described. In Section IV, constructions of optimal and near optimal AMD codes are presented. In Section V, case studies of applying AMD codes to the design of secure serial multipliers used for elliptic curve public key cryptography are shown. The hardware overhead and the security level of different alternatives are compared.

II. Attacker Models and Strong Security

Robust codes [11], [12] are designed to provide a guaranteed level of detection against all error types and classes, assuming the attacker cannot control the fault-free outputs of the cryptographic devices. These codes can be easily compromised when the above assumption is not valid.

Example 2.1: Suppose the 32-bit linear block of AES is protected by a robust duplication code $C = \{u, f(u)\}$, where $u, f(u) \in GF(2^{32})$, $f(u) = u^3$ and all operations are in $GF(2^{32})$. It is easy to prove that any non-zero error e will be masked by at most two out of 2^{32} codewords [8]. If the attacker has the ability to control the inputs (hence the fault-free outputs) and can inject arbitrary error patterns at the output, let (u, y) be an input-output pair, i.e. y is the output of the AES linear block when the input to the device is u. The attacker can easily derive an error pattern $e = (e_y, e_f)$, $e_y, e_f \in GF(2^{32}), e_y \neq 0$ that will be masked by the codeword with y as the information bits. During the attack, the attacker can simply input u to the linear block and inject $e = (e_y, e_f), e_y \neq 0$ at the output of the block. In this case, the attack will always be successful.

The above example assumes an advanced attacker model, where the attacker knows every detail of the cryptographic device including the error detecting code used to protect the device. The attacker can select specific inputs to the device during fault injection attacks. Moreover, the attacker is also able to inject any specific error pattern at the output of the device. In this case, the attacker has full control of not only the non-zero error $e = (e_y, e_f)$, but also the fault-free output y and the faulty output $\tilde{y} = y \oplus e_y$. This situation is probable for the modern fault injection techniques [3]. Under this attacker model, all previous protection architectures based on error detecting codes will not be sufficient. We will call an architecture that can still provide a guaranteed fault detection probability under the above attacker model **strongly secure cryptographic architecture**. Correspondingly, a coding technique that can be used to build strongly secure cryptographic devices is called **algebraic manipulation detection code** (AMD). The strongly secure architectures provide also a high level of protection when the attacker can force the output of the protected device into any specific value.

In this paper, we present constructions of AMD codes based on introducing randomness into the information bits of the code. We describe the architecture of strongly secure cryptographic devices protected by these codes. In the presented architecture, the redundant bits of the code are determined not only by the output y of the original device but also by the random data x generated by a true random number generator, which is incorporated into most cryptographic devices by default for key initialization, random pad computation, challenge generation, etc [13]. We assume that both the original cryptographic devices and the true random number generator may be attacked. We will show that under the most advanced attacker model described in this Section, the cryptographic devices protected by the presented AMD codes can still have a high error (fault) detecting probability.

III. Definitions and Bounds for Algebraic Manipulation Detection Codes

Throughout the paper we denote by $\oplus$ the addition in $GF(q), q = 2^r$. Due to the lack of space, all proofs are omitted. (All the results presented in the paper can be easily generalized to the case where $q = p^r$ (p is a prime).)

Definition 3.1: (**Security Kernel**) For any (k, m, r) error detecting code V with the encoding function $f(x, y)$, where $y \in GF(2^k)$ are information bits, $x \in GF(2^m)$ are random bits and $f(x, y) \in GF(2^r)$ are redundant bits, the **security kernel** K_S is the set of errors $e = (e_y, e_x, e_f), e_y \in GF(2^k), e_x \in GF(2^m), e_f \in GF(2^r)$, for which there exists y so that $f(x \oplus e_x, y \oplus e_y) \oplus f(x, y) = e_f$ is satisfied for all x.

$$K_S = \{e | \exists y, f(x \oplus e_x, y \oplus e_y) \oplus f(x, y) = e_f, \forall x\}. \quad (1)$$

Nonzero errors $e \in K_S$ can be used by an advanced attacker to bypass the protection based on the error detecting codes with security kernel K_S. By injecting faults that manifest as $e \in K_S$ at the output of the device and selecting y for which e is always masked as the input to the device, the attacker can assure that the error (thus the injected fault) will never be detected by the code. Thereby a strongly secure code resilient to advanced fault injection attacks should have no errors in the security kernel except for the all zero vector in $GF(2^n)$, where $n = k + m + r$ is the length of the code.

Definition 3.2: A (k, m, r) error detecting code is called an Algebraic Manipulation Detection code (AMD) iff $K_S = \{\mathbf{0}\}$, where $\mathbf{0}$ is the all zero vector in $GF(2^n), n = k+m+r$.

AMD codes for the special case $m = r$ and $k = br$ were introduced in [14] and were used in [15] for robust secret sharing scheme and for robust fuzzy extractors.

For an AMD code V, let $v = (y, x, f(x, y))$ be the original codeword and $\tilde{v} = (y \oplus e_y, x \oplus e_x, f(x, y) \oplus e_f)$ be the distorted codeword. Denote by $Q_V(y, e)$ the probability

of missing an error e once y is fixed, which can be computed as

$$Q_V(y,e) = 2^{-m}|\{x \,|\, v \in V, \tilde{v} \in V\}| \quad (2)$$

The optimal AMD code should minimize the maximum value of $Q_V(y,e)$ among all codes with the same parameters. Thus the criterion we use to construct good AMD codes is

$$\min_{V \in V_{k,m,r}} \max_{y,e\neq 0} Q_V(y,e), \quad (3)$$

where $V_{k,m,r}$ is the set of all (k,m,r) error detecting codes. In the rest part of the paper, we denote $\max_{y,e\neq 0} Q_V(y,e)$ by Q_V and denote $\min_{V\in V_{k,m,r}} Q_V$ by $Q(k,m,r)$.

A lower bound on $Q(k,m,r)$ for (k,m,r) AMD codes is shown in the next Theorem.

Theorem 3.1: For any (k,m,r) AMD code, where k is the number of information bits, m is the number of random bits and r is the number of redundant bits,

$$\begin{aligned} Q(k,m,r) &= \min_{V \in V_{k,m,r}} \max_{y,e\neq 0} Q_V(y,e) \\ &\geq 1 - 2^{-m} d_q(n,M), \end{aligned} \quad (4)$$

where $d_q(n,M)$ is the maximum possible distance of a q-ary code C_V ($q = 2^r$) with length $n = 2^m$ and $M = |C_V| = 2^{k+m+r}$.

Theorem 3.1 shows the relationship between the worst case error masking probability Q_V for the AMD code V and the Hamming distance of the corresponding traditional error detecting code C_V. We note that $d_q(n,M)$ can be estimated by classical bounds such as the Hamming bound, the Johnson bound, the Singleton bound, the Plotkin bound, etc [16]. When $d_q(n,M)$ is estimated by the Singleton bound, $Q(k,m,r)$ can be written in the compact form.

Corollary 3.1: For any (k,m,r) AMD code,

$$Q(k,m,r) \geq \lceil \frac{k+m}{r} \rceil 2^{-m}. \quad (5)$$

Remark 3.1: It follows from Corollary 3.1 that there are no AMD codes when $k \geq r2^m - m$.

Definition 3.3: A (k,m,r) AMD code V is **optimal** iff

$$\max_{y,e\neq 0} Q_V(y,e) = 1 - 2^{-m} d_q(n,M),$$

where $q = 2^r, n = 2^m, M = 2^{k+m+r}$.

Example 3.1: Let $k = m = 3$ and $r = 1$. According to (5), $Q(3,3,1) \geq \frac{6}{8}$. Let V be the code composed of all vectors $(y, x, f(x,y))$, where $y, x \in GF(2^3)$ and

$$f(x,y) = x_1 \cdot x_2 \cdot x_3 \oplus x_1 \cdot y_1 \oplus x_2 \cdot y_2 \oplus x_3 \cdot y_3. \quad (6)$$

The error masking equation is $f(x \oplus e_x, y \oplus e_y) \oplus f(x,y) = e_f$, which is a polynomial of x with degree 2. The function on the left hand side of the error masking equation corresponds to a codeword in the second order binary Reed-Muller code $RM_2(2,3)$ [16] with 3 variables. It is well known that any codeword of $RM_2(2,3)$ has a Hamming weight of at least 2. Thus the number of solutions for the error masking equation is upper bounded by 6. V is a AMD code with $Q_V = \frac{6}{8}$. This code is optimal.

IV. Constructions of AMD Codes

The codewords of a (k,m,r) AMD code V are in the format of $(y, x, f(x,y))$, where $y \in GF(2^k)$ are information bits, $x \in GF(2^m)$ are random bits and $f(x,y) \in GF(2^r)$ are redundant bits. Let us re-write $f(x,y)$ as $f(x,y) = A(x) \oplus B(x,y)$, where $A(x)$ is independent of y. We next show that by selecting $A(x)$ and $B(x,y)$ based on different classical error detecting codes such as the Generalized Reed-Muller codes and the extended Reed-Solomon codes, we can construct good (and in many cases optimal) AMD codes for different k and different $Q_V = \max_{y,e\neq 0} Q_V(y,e)$ when m and r are given.

Let $x = (x_1, x_2, \cdots, x_t), x_i \in GF(q)$, $q = 2^r$. A b^{th} order q-ary Generalized Reed-Muller code [17] with t variables $GRM_q(b,t)$ consists of all codewords $(f(0), f(1), \cdots, f(q^t-1))$, where $f(x)$ is a polynomial of $x = (x_1, x_2, \cdots x_t)$ of degree up to b. When $b \leq q-3$, the dimension of the code is $\binom{t+b+1}{t}$ [17]. The distance of the code is $q^t - bq^{t-1}$ [17]. Let

$$A(x) = \begin{cases} \bigoplus_{i=1}^{t} x_i^{b+2} & \text{if } b \text{ is odd;} \\ \bigoplus_{i=2}^{t-1} x_1 x_i^{b+1} & \text{if } b \text{ is even and } t > 1; \end{cases}$$

where $\bigoplus$ is the accumulated sum in $GF(2^r)$. Let

$$B(x,y) = \bigoplus_{1 \leq j_1 + j_2 + \cdots + j_t \leq b+1} y_{j_1, j_2, \cdots, j_t} \prod_{i=1}^{t} x_i^{j_i}, \quad (7)$$

where $\prod_{i=1}^{t} x_i^{j_i}$ is a monomial of x of a degree between 1 and $b+1$ and $\prod_{i=1}^{t} x_i^{j_i} \notin \Delta B(x,y)$ defined by

$$\begin{cases} \{x_1^{b+1}, x_2^{b+1}, \cdots, x_t^{b+1}\} & \text{if } b \text{ is odd;} \\ \{x_2^{b+1}, x_1 x_2^b, \cdots, x_1 x_t^b\} & \text{if } b \text{ is even and } t > 1; \end{cases}$$

Suppose $f(x,y) = A(x) \oplus B(x,y)$, it is easy to verify that the left hand side of the error masking equation $f(x \oplus e_x, y \oplus e_y) \oplus f(x,y) \oplus e_f = 0$ is always a non-zero polynomial of x of a degree up to $b+1$. It corresponds to a codeword in the $(b+1)^{th}$ order q-ary Generalized Reed-Muller code. The parameters of the AMD code with $f(x,y)$ as the encoding function are shown in the next Theorem.

Theorem 4.1: Let $f(x,y) = A(x) \oplus B(x,y)$ be a q-ary polynomial with $y \in GF(q^s)$ as coefficients and $x \in GF(q^t)$ as variables, where $1 \leq b \leq q-3$ and $q = 2^r$. Then the code V composed of all vectors $(y, x, f(x,y))$ is an (k,m,r) AMD code with $m = tr$, $k = (\binom{t+b+1}{t} - 1 - t)r$ and $Q_V = (b+1)2^{-r}$.

Remark 4.1: When k is not a multiple of r, 0's can be appended to y before $f(x,y)$ is computed. The resulting AMD code will have the same Q_V as the AMD code with the same $f(x,y)$, for which k is a multiple of r.

When $b = 1$ $B(x, y)$ is the quadratic form $x_1 \cdot y_1 \oplus x_2 \cdot y_2 \oplus \cdots \oplus x_t \cdot y_t$, where all the operations are in $GF(2^r)$. If $e_y \neq 0$, it is easy to verify that the number of solutions for the error masking equation $B(x \oplus e_x, y \oplus e_y) \oplus B(x, y) \oplus e_f = 0$ is upper bounded by q^{t-1}. In this case no $A(x)$ is required and the parameters of the AMD codes are shown in the following Corollary.

Corollary 4.1 ($b = 1, e_y \neq 0$ in Theorem 4.1): If $e_y \neq 0$, the code composed of all vectors $(y, x, f(x, y))$, where $y, x \in GF(q^t)$ and $f(x, y) = x_1 \cdot y_1 \oplus x_2 \cdot y_2 \oplus \cdots \oplus x_t \cdot y_t, f(x, y) \in GF(2^r), q = 2^r$ is an optimal (tr, tr, r) AMD code with $Q_V = 2^{-r}$.

When $t = 1$ and b is odd, $A(x) = x^{b+2}$ and $B(x, y) = x \cdot y_1 \oplus x^2 \cdot y_2 \oplus \cdots \oplus x^b \cdot y_b$. The left hand side of the error masking equation $f(x \oplus e_x, y \oplus e_y) \oplus f(x, y) \oplus e_f = 0$ corresponds to a codeword in an extended q-ary Reed-Solomon code, $q = 2^r$ [16]. For this case, the code generated by Theorem 4.1 coincides with the construction shown in [14].

Corollary 4.2 ($t = 1$ in Theorem 4.1): When $b \leq q - 3$ is an odd number, the code V composed of all vectors $(y, x, f(x, y))$, where $y \in GF(q^{bt}), x \in GF(q)$ and $f(x, y) = x^{b+2} \oplus x \cdot y_1 \oplus x^2 \cdot y_2 \oplus \cdots x^b \cdot y_b, f(x, y) \in GF(q), q = 2^r$, is an optimal (br, r, r) AMD code with $Q_V = \max_{y, e \neq 0} Q_V(y, e) = \frac{b+1}{q}$.

Compared to codes with $t > 1$ generated by Theorem 4.1, codes generated by Corollary 4.2 have higher data rate and are more suitable for applications where the number of redundant bits are critical, e.g. for secure memories where the overall overhead is determined by the number of redundant bits.

Example 4.1: Let $y = (y_1, y_2, y_3, y_4), y_i \in GF(2^8)$ be the output of a 32-bit linear block of AES. Let $r = 8$ and $t = 1$ and $b = 4$. Let $x \in GF(2^8)$ be the 8-bit random data generated by the random number generator. The linear block of AES can be protected by verifying the redundant bits computed as described in Corollary 4.2.

$$f = y_1 \cdot x \oplus y_2 \cdot x^2 \oplus y_3 \cdot x^3 \oplus y_4 \cdot x^4 \oplus x^7.$$

In this case the probability of conducting a successful fault injection attack is at most $6 \cdot 2^{-8}$.

As a case study, in the next section we will present architectures based on the proposed AMD codes for secure multipliers in $GF(2^k)$, which are commonly used blocks in cryptographic devices implementing the elliptic curve cryptographic algorithms [18], etc.

V. Protection of Normal Base Serial Multipliers in $GF(2^k)$

The general architecture based on AMD codes for the protection of cryptographic devices against fault injection attacks is shown in Figure 1. In addition to the original device, two extra blocks, the predictor and the error detecting network (EDN) are needed. The extended outputs of the fault-free device are codewords of the AMD code. As in most works discussing the protection of data-path in cryptographic devices [9], [10], we assume that the EDN is tamper resistant and cannot be attacked by the attacker. Otherwise, an advanced attacker can easily bypass any kind of protection mechanism based on error detecting codes by forcing the error flag signal *Error* to be 0 (Figure1). In most cases EDN is much smaller than the original device and the predictor. For example, for a 6-cell 64M-bit SRAM with $k = 64$ bits per word and $Q_V = 3 \times 2^{-16}$, the number of transistors in EDN divided by the number of transistors in the SRAM array is less than 10^{-5}. We also note that to improve the security of EDN, it may be implemented as a self-checking checker using the dual rail design [19].

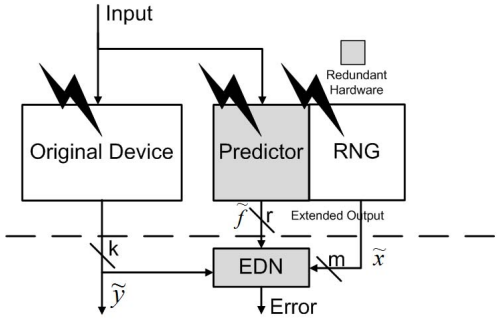

Figure 1. General architecture of a device protected by a (k, m, r) AMD code. RNG is an m-bit random number generator.

In this section we assume all Galois fields are represented by normal bases. The hardware implementations of multipliers in $GF(2^k)$ can be categorized as parallel multipliers and serial (sequential) multipliers. Compared to parallel multipliers, serial multipliers are more area efficient and are more practical in hardware for multiplications in a large Galois field especially in small digital devices, e.g. smart phones. A digit-serial Massey-Omura multiplier can output one digit of the product per clock cycle. Suppose the length of the digit is r-bit and the output of the multiplier is k-bit. The multiplication in $GF(2^k)$ is completed in $\lceil \frac{k}{r} \rceil$ clock cycles. The digit-serial Massey-Omura multiplier [20] can be implemented by using r identical combinatorial blocks with cyclically shifted inputs for normal base multiplication in $GF(2^k)$.

We next estimate the area overhead for a digit-serial Massey-Omura multiplier in $GF(2^k)$ protected by AMD codes generated by Corollary 4.1 ($b = 1$ in Theorem 4.1) and Corollary 4.2 ($t = 1$ in Theorem 4.1).

When $b = 1$ in Theorem 4.1,

$$f(x, y) = \bigoplus_{i=1}^{t} x_i^3 \oplus x_i y_i = \bigoplus_{i=1}^{t} x_i (x_i^2 \oplus y_i). \qquad (8)$$

If $e_y \neq 0$, $\bigoplus_{i=1}^{t} x_i^3$ can be omitted and $f(x, y)$ can be simplified to be $\bigoplus_{i=1}^{t} x_i y_i$. The structure of the predictor for $(((\binom{t+2}{2} - 1 - t)r, tr, r)$AMD codes with $b = 1$ for the protection of a digit-serial multiplier in $GF(2^k)$ is shown

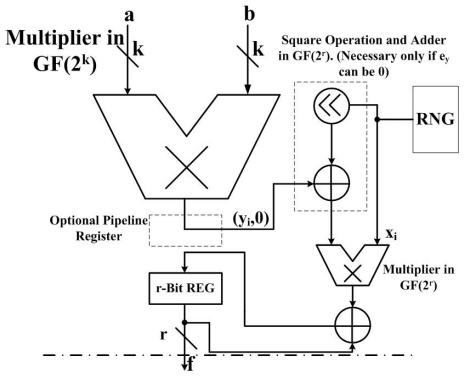

Figure 2. Predictor for serial Massey-Omura multiplier in $GF(2^k)$ protected by codes with $b = 1$ generated by Theorem 4.1

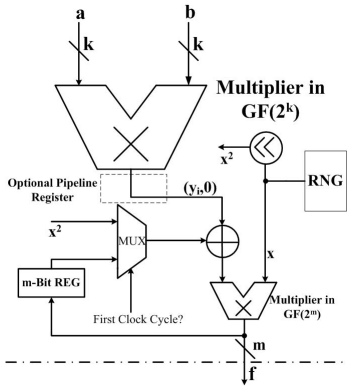

Figure 3. Predictor for serial Massey-Omura multiplier in $GF(2^k)$ protected by codes with $t = 1$ based on Theorem 4.1

in Figure 2. $x_i \in GF(2^r)$ is the random data generated by the true random number generator. y_i is the i^{th} component of the product. (The square operation can be implemented by cyclically shifting in normal base Galois fields [21].) The parallel multiplier in $GF(2^r)$ can be implemented as described in [22]. At every clock cycle, the digit generated by the digit-serial multiplier in $GF(2^k)$ is added to x_i^2 and then multiplied by x_i (see (8)). The result is cumulatively added and saved in the r-bit register. After t clock cycles, the redundant bits will be available in the r-bit register and will be verified by EDN to detect errors. To reduce the latency of the predictor, an optional pipeline register can be added between the original digit-serial multiplier in $GF(2^k)$ and the parallel multiplier in $GF(2^r)$ as shown by the dotted block in Figure 2.

When $t = 1$ and b is odd in Theorem 4.1,

$$\begin{aligned} f(x,y) &= x^{b+2} \oplus \bigoplus_{i=1}^{b} y_i x^i \\ &= x(y_1 \oplus x(y_2 \oplus \cdots \oplus x(y_b \oplus x^2)\cdots)) \end{aligned}$$

Table I
HARDWARE COMPLEXITY FOR PARALLEL AND DIGIT-SERIAL MASSEY-OMURA MULTIPLIERS

Type	AND	XOR	Latency
Digital-Serial MO[20]	rC_N	$r(C_N-1)$	$T_A + \lceil log_2 C_N \rceil T_X$
Parallel RR-MO[22]	k^2	$\frac{k}{2}(C_N+k-2)$	$T_A + \lceil log_2(C_N+1) \rceil T_X$
Parallel RR-MO*	k^2	k^2-1	$T_A + (1+\lceil log_2(k-1) \rceil) T_X$

*: Type I optimal normal base (ONB) generated by irreducible all-one polynomials exists [22].

The structure of the predictor for the resulting AMD codes is shown in Figure 3. During the first clock cycle of every multiplication, the output digit y_b ($0's$ are appended if necessary) is added to x^2 and then multiplied by x. For each of the following clock cycles, xy_{b-i+1} is accumulated added to the contents stored in the r-bit register. The predictor for AMD codes with $t = 1$ requires nearly same overhead in area and latency as the predictor for AMD codes with $b = 1$.

A. Estimations of the Hardware Overhead for the Protection of Multipliers in Galois Fields Recommended for Elliptic Curve Cryptographic Algorithms

Table I summarizes the hardware complexity for the reduced redundancy parallel [22] (Row 2 and 3) and the digit-serial (Row 1) Massey-Omura multipliers in $GF(2^k)$, where r is the bit-width of the digit, C_N is the complexity of the normal base [22] and T_A, T_X are the delays due to one AND gate and one XOR gate respectively.

Table II shows the overhead for the predictor and EDN of the AMD codes with $b = 1$ or $t = 1$ generated by Theorem 4.1 for multipliers in $GF(2^{239})$ and $GF(2^{409})$, which are among the recommended Galois fields for elliptic curve cryptographic algorithms [18]. There are two columns for the number of AND and XOR gates for each k. The left column shows the number of gates that is required to implement the original multiplier in $GF(2^k)$. The right column shows the number of gates that is required to implement the other parts of the predictor and EDN. To estimate the area overhead, we only consider multipliers and ignore the adder, the multiplexer in $GF(2^r)$ and the r-bit register in the predictor since the space complexity of an r-bit multiplier is of the order of $O(r^2)$ and the complexity of an r-bit multiplexer and an r-bit adder is of the order of $O(r)$. We consider the cases where the multiplication in $GF(2^k)$ is completed in 2, 4 or 8 clock cycles. For each case, we select r in such a way that there is an optimal normal base of Type I in $GF(2^r)$ for the purpose of minimizing the hardware complexity of multipliers in $GF(2^r)$. When computing the percentage overhead of the predictor and EDN, we assume that the area of a XOR gate is about 1.5 times of the area of a AND gate according to the data of the 45nm NANGATE library [23].

Generally speaking, the area overhead for the predictor and EDN for codes based on Theorem 4.1 decreases as the number of clock cycles needed to finish one multiplication in $GF(2^k)$ increases. For the three cases shown in Table II, the overall area overhead for the predictor and EDN is about $110\% \sim 160\%$. In addition to the area overhead, the

Table II
ESTIMATION OF THE TOTAL AREA OVERHEAD FOR THE PREDICTOR AND EDN FOR DIGIT-SERIAL MULTIPLIERS IN $GF(2^k)$ PROTECTED BY CODES GENERATED BY THEOREM 4.1

Cycles	$k = 239^{(a)}$						$k = 409$					
	r	AND	XOR	Percentage	$log_2 Q_V$		r	AND	XOR	Percentage	$log_2 Q_V$	
2	$130^{(b)}$	$62,010$	$33,800$	$61,880$	$33,798$	154.5%	$-129(-128)$	$226^{(b)}$	$> 184,642$	$102,150$	$> 184,416$	$102,148$	155.4%	$-225(-224)$
4	$60^{(b)}$	$28,620$	$7,200$	$28,560$	$7,198$	125.2%	$-59(-57.4)$	$106^{(b)}$	$86,602$	$22,470$	$86,496$	$22,468$	126.0%	$-105(-103.4)$
8	$36^{(b)}$	$17,172$	$2,592$	$17,136$	$2,590$	115.1%	$-35(-32.7)$	$52^{(b)}$	$42,484$	$5,406$	$42,432$	$5,404$	112.8%	$-51(-48.7)$

(a): There exists an optimal normal base of Type II for $GF(2^r)$ [22].

(b): There exists an optimal normal base of Type I for $GF(2^r)$ [22].

[*]: We assume that the area of a XOR gate is approximately 1.5 times of the area of the AND gate according to the data of the 45nm NANGATE library[23].

protection architectures based on the proposed AMD codes will also increase the latency of the multiplier due to the longer critical path in the predictor. For example, when $k = 409$ and $r = 106$, the latency of the serial reduced-redundancy Massey-Omura multiplier is $T_A + 9T_X$. In the predictor, for codes with $b = 1$ shown in Figure 2, the critical path contains a digit-serial multiplier in $GF(2^{409})$, a parallel multiplier in $GF(2^{106})$ and a 2-level XOR network, assuming $e_y \neq 0$. Thus the latency of the predictor is $2T_A + 19T_X$ and is twice larger than the latency of the original multiplier. To reduce the latency, optional pipeline registers can be added between the duplicated multiplier in $GF(2^k)$ and the multiplier in $GF(2^r)$ as shown by the dotted blocks in Figure 2. In this case the same latency as for the original multiplier can be achieved for the predictor. Similar strategy can also be applied for codes with $t = 1$ (Figure 3).

VI. CONCLUSIONS

In this paper, we present general constructions of strongly secure algebraic manipulation detection (AMD) codes. The proposed codes can provide a guaranteed level of protection against fault injection attacks even if an attacker can fully control the fault-free outputs of the device and the error patterns. The same characteristic cannot be achieved by any previously proposed protection countermeasures based on error detecting codes in the literature. As a case study, we present the protection architectures based on the AMD codes for multipliers in Galois fields recommended for elliptic curve cryptography. The area overhead for the protection architectures is between $110\% - 160\%$, which is much smaller than for widely used duplication approach and as opposed to duplication, for the ADM based architectures there are no undetectable errors even for the strong attacker model where both the error pattern and the fault-free outputs of the device are controllable by the attacker. When the predictor is pipelined, the protected multiplier has no latency penalty and can achieve the same performance as the original device.

REFERENCES

[1] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *KOBLITZ, N.(ed.) CRYPTO, Lecture Notes in Computer Science*, vol. 1109, 1996, pp. 104–113.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology CRYPTO 99*, ser. Lecture Notes in Computer Science, M. Wiener, Ed., vol. 1666. Springer Berlin / Heidelberg, 1999, pp. 789–789.

[3] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," in *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*. Springer-Verlag, 2003, pp. 2–12.

[4] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," vol. 94, no. 2, 2006, pp. 370 –382.

[5] G. Piret and J.-J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, ser. Lecture Notes in Computer Science, C. Walter, e. Ko, and C. Paar, Eds., vol. 2779. Springer Berlin / Heidelberg, 2003, pp. 77–88.

[6] 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, no. 4, pp. 492–505, 2003.

[7] T. Malkin, F.-X. Standaert, and M. Yung, "A comparative cost/security analysis of fault attack countermeasures," in *Fault Diagnosis and Tolerance in Cryptography*, ser. Lecture Notes in Computer Science, L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, Eds. Springer Berlin / Heidelberg, 2006, vol. 4236, pp. 159–172.

[8] M. G. Karpovsky and A. Taubin, "A new class of nonlinear systematic error detecting codes," *IEEE Trans Info Theory*, vol. 50, no. 8, pp. 1818–1820, 2004.

[9] M. Karpovsky, K. J. Kulikowski, and A. Taubin, "Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard." IEEE Computer Society, 2004, p. 93.

[10] G. Gaubatz, B. Sunar, and M. Karpovsky, "Non-linear residue codes for robust public-key arithmetic," in *Fault Diagnosis and Tolerance in Cryptography*, ser. Lecture Notes in Computer Science, L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, Eds., vol. 4236. Springer Berlin / Heidelberg, 2006, pp. 173–184.

[11] 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.

[12] Z. Wang, M. Karpovsky, and K. Kulikowski, "Design of memories with concurrent error detection and correction by nonlinear SEC-DED codes," *Journal of Electronic Testing*, pp. 1–22, 2010.

[13] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 109–119, 2007.

[14] Y. Dodis, J. Katz, and L. Reyzin, "Robust fuzzy extractors and authenticated key agreement from close secrets," in *Cryptology CRYPTO*. Springer, 2006, pp. 232–250.

[15] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs, "Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors," in *Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, ser. EUROCRYPT'08, 2008, pp. 471–488.

[16] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1998.

[17] E. F. Assmus, Jr., and J. D. Key, *Polynomial Codes and Finite Geometries*, 1995.

[18] "Standards for efficient cryptography, sec 2: Recommended elliptic curve domain parameters," 2000.

[19] T. R. N. Rao and E. Fujiwara, *Error-control coding for computer systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[20] J. Massey and J. Omura, *Computational Method and Apparatus for Finite Field Arithmetic*, 1986.

[21] S. Gao, "Normal bases over finite fields," Ph.D. dissertation, University of Waterloo, 1993.

[22] A. Reyhani-Masoleh and M. A. Hasan, "A new construction of Massey-Omura parallel multiplier over GF(2^m)," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 511–520, 2002.

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