Comparative Analysis of Robust Fault Attack Resistant Architectures for Public and Private Cryptosystems

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

Abstract

The adaptive and active nature of fault based side-channel attacks along with the large arsenal of fault injection methods complicates the design of effective countermeasures. To overcome the unpredictability of fault attackers protection methods based on robust codes were proposed which can provide uniform error detection against all errors eliminating possible weaknesses in the protection. In this paper we evaluate and compare the error detection properties and hardware overheads of architectures based on robust, partially robust, and minimum distance robust codes for both public and private key cryptosystems.

1. Introduction

Fault based cryptanalysis is one of the most powerful types of side-channel attacks. Unlike other forms of side-channel attack 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 complicates the design of secure devices.

To prevent fault attacks, redundancy, in the form of error detecting codes is often added to detect and react to errors and attacks. Most of the proposed system level protection methods make assumptions about the expected faults due to an attack. Typically these methods have been based on the use of linear minimum distance error detecting codes such as parity or Hamming codes. These codes concentrate their error detecting power on errors of a low multiplicity.

However, it has never been proven nor argued that it is sufficient to only detect a particular subset of faults or errors to prevent a fault attack. The spectrum of available fault injection methods and the adaptive nature of an attacker suggests that it would be possible to bypass such protection by injecting a class of faults or errors which the cryptographic device has not been anticipating. Considering even only inexpensive non-invasive or semi-invasive fault attacks, there is a wide spectrum of the types of faults and injection methods an attacker has at his disposal [2].

Robust codes have been proposed as solution to the limitation of minimum distance error detecting codes for detection of fault injection attacks [8]. These nonlinear codes are designed to provide equal protection against all errors thereby eliminating possible weak areas in the protection that can be exploited by an attacker. Several variants of robust codes have been used to protect both private and public cryptographic algorithms. These variants allow several tradeoffs in terms of robustness and hardware overhead for many architectures. *Robust*, *partially robust*, and *minimum distance partially robust* codes have been used for the protection for both private [7] [8] and public key cryptosystems [6].

In this work we compare the error detection characteristics and hardware overheads of the previously proposed robust codes and robust architectures. We also propose a new type of robust codes, the *minimum distance robust codes*, which combine robust properties with the minimum distance of traditional linear codes. We compare the properties of the proposed codes with fault simulations on a linear sub-circuit of the Advanced Encryption Standard (AES) and a multiplier as the exemplary sub-blocks for private and public key cryptosystems respectively.

In the next section we summarize the variants of robust codes. In Section 3 formalize their designs and propose constructions for minimum distance robust codes. Sections 4 and 5 we compare the different protection methods for an AES sub-circuit and a multiplier.

2. Definitions

A qualitative summary of the main properties and characteristics of the main types of robust codes compared to a

classical linear code are presented in Figure 1 and Table 1.

As shown in the Figure and Table, a main characteristic of traditional linear error detecting codes is that they concentrate their error detecting power on a small subset of errors which are assumed to be the most likely to occur or most likely to be injected by an attacker. Typically, such codes concentrate their error detection on errors of a small multiplicity. They are designed to guarantee detection of all errors with a multiplicity less than d. Error detection beyond the minimum distance of the code is typically not a part of the design criteria and can be unpredictable and ineffective. While for some classes of errors the codes provide 100% protection, for a very large class of errors linear codes offer no protection for all messages. For any q-ary linear systematic error detecting code of length n and dimension k there are q^k undetectable errors. Linear codes have the largest kernel (denoted by K), or rather the set of undetectable errors, of any class of systematic codes with the same n and k.

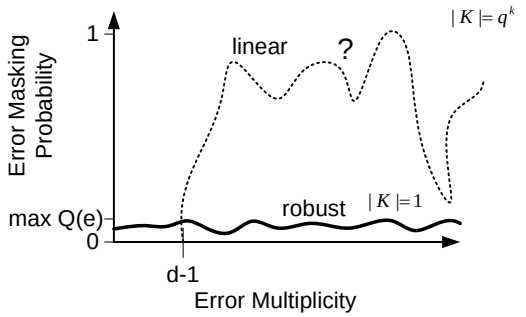

Figure 1. Error detection profile of a robust and linear code

Robust codes are designed to provide for a guaranteed level of detection against all error types and classes. The characteristics of an error profile of a robust code are shown in by a solid line in Figure 1. These codes are characterized by their error masking probability $Q(e)$, which is the fraction of codewords that mask each error.

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

Definition 2.1 *The code C is **robust** with respect to its error-masking probability iff $\max_{e \neq 0} Q(e) < 1$ or equivalently the kernel of the code only contains the zero vector $K = \{0\}$.*

The codes are considered optimum when the maximum $Q(e)$ for all errors is minimized. For a robust codes the error masking probability is bounded for every class and type of error.

Robust codes have no undetectable errors. That is, the kernel of a robust code is just one element, the zero vector. Traditional robust codes do not have a minimum distance larger than one and are typically not mean to guarantee 100% detection probability for any subset of errors. The error detection is as uniform as possible for all errors. A possible variant of the traditional robust codes is to include a minimum distance into the design criteria.

Definition 2.2 *Let $||e||$ denote the multiplicity of an error e. A robust code where the $Q(e) = 0$ for all errors where $||e|| < d$ is a **d-minimum distance robust code**.*

Minimum distance robust codes are fully robust codes but also have a minimum distance larger than one. Since these codes are robust they have no undetectable errors and the worst case error masking probability is bounded and predictable. However, unlike traditional robust codes they also provide for a guaranteed 100% probability of detection for a predefined class of errors. Such codes can be useful for providing the highest protection against the most likely or most dangerous threat while maintaining a detection guarantee in case of an unexpected behavior or attack.

	size of kernel K	$\max Q(e)$ $e \notin K$	min. distance
linear	q^k	1	d
robust	1	q^{-r}	1
partially robust	q^{k-r}	q^{-r}	1
min. dist. partially robust	$\ll q^k$	$\ll 1$	d
min. dist. robust	1	$\ll 1$	d

Table 1. characteristics of good systematic $(n,k)_q$ codes

For some applications the error characteristics of robust codes can be considered too pessimistic. Variants of robust codes which fill the gap between the optimistic linear codes and pessimistic robust codes are possible. *Partially robust* codes and *minimum distance partially robust* codes allow for a trade off between robustness, encoding complexity, and overheard.

Definition 2.3 *A systematic $(n,k)_q$ code with a kernel smaller than q^k is a **partially robust code**. If the code also has a minimum distance greater than one it is referred to as a **minimum distance partially robust code**. (For a code C, kernel K of C is defined as $K = \{c|c \in C, \forall c' \in C, c'+c \in C\}$. If C is linear $K = C$.)*

Partially robust codes reduce the number of undetectable errors while preserving some structures of linear codes

which can be exploited to build efficient prediction hardware that generates redundant bits of a message. Like linear codes, partially robust codes still have undetectable errors (hence they are not completely robust). The number of undetectable errors is reduced by many orders of magnitude. For practical partially robust constructions the number of undetectable errors can be reduced from q^k to q^{k-r} compared to a linear $(n,k)_q$ code [9]. The probability of masking for the errors that are detectable is bounded as in robust codes.

For partially robust and minimum distance partially robust codes two parameters are used for characterization, the size of the kernel and the maximum probability of masking for errors not in the kernel (see Table 1).

We next show constructions for these codes.

3. Constructions

3.1 Robust and Minimum Distance Robust Codes

A systematic robust code can be defined by the nonlinearity of its encoding function. The nonlinearity of the function can be measured by using derivatives $D_a f(x) = f(x+a) - f(x)$. The nonlinearity measure can be defined by (from [4])

$$P_f = \max_{0 \neq a \in GF(q^k)} \max_{b \in GF(q^r)} Pr(D_a f(x) = b) \qquad (2)$$

where $Pr(E)$ denotes the probability of occurrence of event E. The smaller the value of P_f, the higher the corresponding nonlinearity of f.

Theorem 3.1 ***([10])*** *Let f be a function with nonlinearity P_f that maps $GF(q^k)$ to $GF(q^r)$ where $k \geq r$, the set of vectors resulting from the concatenation of $a, b : \{(a, b = f(a))\}$ where $a \in GF(q^k)$ and $b \in GF(q^r)$ forms a robust systematic error-detecting code where $\max_{e \neq 0} Q(e) \leq P_f$).*

Example 3.1 ***(Robust Parity)*** *The code $C = \{(x, f(x))\}$ where $x \in GF(2^{32})$ and $f : GF(2^{32}) \rightarrow GF(2)$ is a perfect nonlinear function with $P_f = 0.5$ defined by $f(x = (x_1 x_2 ... x_{32})) = x_1 x_2 + x_3 x_4 + ... + x_{31} x_{32}$, is a robust error-detecting code where $Q(e) \leq 0.5$ for any error e.*

We note that for unidirectional $0 \rightarrow 1$ errors where $\tilde{w} = w \vee e$ where $\tilde{w}$ is a distorted value for w and $\vee$ is a componentwise OR operation we have for robust parity codes $Q(e) \leq 0.5(1 - 2^{-||e||})$ for any e (where$||e||$ is the number of ones in e).

Minimum distance robust codes for algebraic errors can be constructed by appending a nonlinear signature to any systematic minimum distance code.

Theorem 3.2 *Let V be a systematic $(n, q^k, d)_q$ code and let $f : GF(q^k) \rightarrow GF(q^r)$ be nonlinear function with nonlinearity P_f. The code*

$$C = \{(x, \phi(x), f(x)) | (x, \phi(x)) \in V\}, \qquad (3)$$

where ϕ is the encoding function for code V, is a $(n + r, q^k, d)$ minimum distance robust code where $\max_{e \neq 0} Q(e) \leq P_f$.

Proof Appending extra nonlinear bits does not change the minimum distance of the code. Any error which will affect only the redundant bits of V will clearly be immediately detected. Any other error will be detected by the robust code. The robustness R of the code follows from Theorem 3.1.

Example 3.2 ***(Minimum Distance Robust Parity)*** *The code $C = \{(x, p(x), f(x))\}$ where $x \in GF(2^{32})$, $f : GF(2^{32}) \rightarrow GF(2)$ is a perfect nonlinear function defined by $f(x = (x_1, x_2, ..., x_{32})) = x_1 x_2 + x_3 x_4 + ... + x_{31} x_{32}$ and $p(x)$ is the linear parity function of x, is a $d = 2$ minimum distance robust error-detecting code. For this code $Q(e) = 0$ when $||e|| = 1$ and $Q(e) \leq 0.5$ when $||e|| > 1$.*

Similar constructions of robust codes, presented above for algebraic error models, can be extended to provide protection against arithmetic errors. Interestingly, the same non-repetative quadratic form used as the encoding function used for protection of algebraic errors can also be adapted for robust detection of arithmetic errors. We consider arithmetic errors that are modulo additive and $|e + x|_T$ is used to denote operations modulo T.

Theorem 3.3 *Let $x = x_1 + 2^t x_2 + ... + 2^{(s-2)(t)} x_{s-1} + 2^{(s-1)(t)} x_s$, where $x_i \in \mathbb{Z}_{2^t}$, s is even and $st = k$ be an output of a arithmetic device. Let $f(x) = |x_1 x_2 + x_3 x_4 + ... + x_{s-1} x_s|_p$ where $p > 2^t$ is a prime. The code*

$$C = \{(x, w = f(x))\} \qquad (4)$$

is a robust code with respect to additive errors $e = (e_x, e_w), e_x \in \mathbb{Z}_{2^k}, e_w \in \mathbb{Z}_p$ where $c+e = (|x+e_x|_{2^k}, |w+e_w|_p), c \in C$. The maximum error masking probability for the code is bounded by $\max_{e \neq 0} Q(e) \leq 2^{-t+1}$

Proof Let the error $e_x = e_1 + 2^t e_2 + ... + 2^{(s-2)(t)} e_{s-1} + 2^{(s-1)(t)} e_s$ For a fixed error e assume, without a loss in generality, that at least one component of the error (say e_1) is nonzero.

1. when $e_1 + x_1 < 2^t, e_2 + x_2 < 2^t$, for a fixed $x_i, i > 2$ the error is masked iff

$$|e_1 x_2 + e_2 x_1|_p = |E|_p \qquad (5)$$

where E is a constant that is a function of e_w, e_i, x_i where $i > 2$. For any constant E, the number of values of x_1 and x_2 in this range that will satisfy the above error masking equation for a fixed E is at most $\min(2^t - e_1, 2^t - e_2)$. Since the constant E can be generated in $2^{t(s-2)}$ ways by selecting all possible values of $x_i, i > 2$ the total number of messages that can mask an error in this range is at most

$$R_1 \leq \min(2^t - e_1, 2^t - e2)(2^{t(s-2)}). \quad (6)$$

2. likewise when $e_1 + x_1 < 2^t, e_2 + x_2 \geq 2^t$, for a fixed $x_i, i > 2$ the error is masked iff

$$|e_{x1}x_2 + (e_2 - 2^k)x_1|_p = |E|_p \quad (7)$$

For any constant E, the number values of x_1 and x_2 in this range that will satisfy the above error masking equation is at most $\min(2^t - e_1, e_2)$. The total number of messages that can mask an error in this range is at most

$$R_2 \leq \min(2^t - e_1, e2)(2^{t(s-2)}) \quad (8)$$

3. when $e_1 + x_1 \geq 2^t, e_2 + x_2 + 1 < 2^t$, for a fixed $x_i, i > 2$ the error is masked iff

$$|(e_1 - 2^k)x_2 + (e_2 - 1)x_1|_p = |E|_p \quad (9)$$

and thus

$$R_3 \leq \min(e_1, 2^t - 1 - e2) * (2^{t(s-2)}) \quad (10)$$

4. when $e_1 + x_1 \geq 2^t, e_2 + x_2 + 1 \geq 2^t$, for a fixed $x_i, i > 2$ the error is masked iff

$$|(e_1 - 2^k)x_2 + (e_2 - 1 - 2^k)x_1|_p = |E|_p \quad (11)$$

and thus

$$R_4 \leq \min(e_{x1}, ex2 + 1) * (2^{t(s-2)}) \quad (12)$$

Summing the four different cases, the number of messages that can mask the error e is therefore

$$\sum_{i=1}^{4} R_i \leq 2^{t(s-1)+1} \quad (13)$$

The same analysis can be performed when any component of e_x is assumed nonzero. When e_w is the only nonzero component the error is always detected.

Example 3.3 *For the code from Theorem 3.3 where $k = 64$ and $t = 16, s = 4$, all arithmetic errors are detectable and $Q(e) \leq 2^{-15}$ for all nonzero e.*

3.2 Partially Robust and Minimum Distance Partially Robust Codes

Theorem 3.4 *Let $f : GF(q^r) \rightarrow GF(q^r)$ have nonlinearity P_f and let $l : GF(q^k) \rightarrow GF(q^r), r \leq k$ be a linear onto function. The set of words in the form $(x, f(l(x)))$ form a partially robust code with q^{k-r} undetectable errors.*

Proof By Theorem 3.1 the set of words in the form $(l(x), f(l(x)))$ forms a robust code of length $2r$. For this code the only undetectable error is the zero error. Since l is a linear onto function it maps q^{k-r} words from $GF(q^k)$ to the *r*-bit zero vector. Hence the code with words in the form $(x, f(l(x)))$ has a total of q^{k-r} undetectable errors.

Example 3.4 ***(Partially Robust Hamming)*** *The code $C = \{(x, (Hx)^3)\}$ where $x \in GF(2^{32})$, H is a 32 by 6 encoding matrix of a shortened $(38, 32)$ Hamming code, and the cubing operation is over $GF(2^6)$ is a binary partially robust code with the number of undetectable errors $|K| = 2^{26}$ and $\max_{e \notin K} Q(e) = 2^{-5}$.*

The construction of Theorem 3.4 can be extended and adapted for arithmetic devices where $k = 1$ and errors are additive. In such devices the modular reduction (mod p, where p is prime) operation can be used as the linear function of Theorem 3.4. Similarly to where $k > 1$, the partially robust codes over arithmetic errors can be shown to reduce the number of undetectable and bad errors over linear codes with the same parameters by a factor of p.

Theorem 3.5 *The code*

$$C = \{(x, w) | x \in \mathbb{Z}_{2^k}, w = f(x)\} \quad (14)$$

where $r = \lceil log_2 p \rceil$, p is a prime, and $f(x) = |x^2|_p$ is a function with perfect nonlinearity is a partially robust code with respect to additive errors $e = (e_x, e_w)$, $e_x \in \mathbb{Z}_{2^k}$, $e_w \in \mathbb{Z}_p$ where $c + e = (|x + e_x|_{2^k}, |w + e_w|_p), c \in C$. The nonzero error masking probability $Q(e)$, for an error e is

$$\begin{array}{ll} \leq 2^{-k}(\lceil 2^k/p \rceil + 1) & , e_w \neq 0 \\ 2^{-k}((2^k - e_x) + (e_x/p)) & , |e_x|_p = 0, e_w = 0 \\ 2^{-k}(e_x + (2^k - e_x)/p) & , |e_x|_p = 2^k, e_w = 0 \end{array} \quad (15)$$

where there are

$$\begin{array}{l} 2^k p - 2\lfloor 2^k/p \rfloor \text{ errors of the first type} \\ \lfloor 2^k/p \rfloor \text{ errors of the second type} \\ \lfloor 2^k/p \rfloor \text{ errors of the third type} \end{array} \quad (16)$$

Proof Under additive arithmetic errors. An error $e = (e_x, e_w), e_x \in \mathbb{Z}_{2^k}, e_w \in \mathbb{Z}_{2^r}$ distorts a message $m =$

(x, w) into an erroneous message $\tilde{m} = m + e = (|x + e_x|_{2^k}, |w + e_w|_p)$.

The properties of the codes can be derived by analyzing different cases depending on the value of $x + e_x$

1. For $x + e_x < 2^k$ we have $|x + e_x|_{2^k} = x + e_x$ and $f(|x + e_x|_{2^k}) = f(x + e_x) = |x^2 + 2xe_x + e_x^2|_p$. In this case $e = (e_x, e_w)$ is missed for message x iff

$$|e_x^2 + 2xe_x - e_w|_p = 0. \quad (17)$$

   (a) If $e_x = 0 \pmod p$, then (e_x, e_w) is missed iff $e_w = 0$. Each error where $|e_x|_p = e_w = 0$ is masked for $2^k - e_x$ messages in this range.

   (b) If $|e_x|_p \neq 0$, then (e_x, e_w) is missed iff $|x|_p = |\frac{e_w - e_x^2}{2e_x}|_p, 0 \leq x < 2^k - e_x$. There are at most $\lceil (2^k - e_x)/p \rceil$ such messages for each (e_x, e_w).

2. For $x + e_x \geq 2^k$ we have $|x + e_x|_{2^k} = x + e_x - 2^k$ and $f(|x + e_x|_{2^k}) = f(x + e_x - 2^k) = |x^2 + e_x^2 + 2^{2k} + 2xe_x - 2x2^k - 2e_x2^k|_p$. In this case e is missed iff

$$|e_x^2 + 2^{2k} + 2xe_x - 2x2^k - 2e_x2^k - e_w|_p = 0. \quad (18)$$

   (a) If $|e_x|_p = |2^k|_p$, then (e_x, e_w) is missed iff $e_w = 0$. The error where $|e_x|_p = |2^k|_p, e_w = 0$ is masked for e_x messages in this range.

   (b) If $|e_x|_p \neq |2^k|_p$, then (e_x, e_w) is missed iff $|x|_p = |\frac{e_w - (e_x - 2^k)^2}{2(e_x - 2^k)}|_p, 2^k - e_x \leq x < 2^k$. There are at most $\lceil e_x/p \rceil$ such messages for each (e_x, e_w) in this range.

   There are thus $2\lfloor 2^k/p \rfloor$ errors of class 1(a) and 2(a). These errors are considered "bad" since their error masking probability may be close to one. The number of these errors is reduced by factor of p over a linear code.

Many classical constructions of nonlinear codes are partially robust minimum distance codes. They have a minimum distance but have much fewer undetectable errors than linear codes. Such codes can even be perfect with respect to the classical Hamming bound.

The first nonlinear perfect code was constructed by Vasil'ev in [16] and was generalized by Mollard in [12]. We first review the basic construction of Vasil'ev code.

Theorem 3.6 ***(Vasil'ev [16])*** *For $x \in GF(2^n)$, let $p(x) = wt(x) \bmod 2$. Let V be a perfect not necessarily linear Hamming code of length $n = 2^r - 1$ with $k = n - r$ information bits. Let $f : V \to \{0, 1\}$ be an arbitrary mapping such that $f(0) = 0$ and $f(v) + f(v') \neq f(v + v')$ for some $v, v' \in V$. The code C defined by*

$$C = \{(x, x + v, p(x) + f(v)) | x \in GF(2^n), v \in V\} \quad (19)$$

(where $+$ is over $GF(2)$) is perfect.

Theorem 3.7 *Vasil'ev code is a $(2n + 1, 2n - r, 3)$ ia partially robust code with $|K| = 2^n$ and $\max_{e \notin K} Q(e) = P_f$ where P_f is the nonlinearity of f, and K is the kernel of the code, and $n = 2^r - 1$.*

Proof Let H be the check matrix of V. An error $e = (e_1, e_2, e_3)$ where $e_1, e_2 \in GF(2^n)$ and $e_3 \in GF(2)$ is missed if and only if $H(e_1 + e_2) = 0$ and $f(v + e_1 + e_2) + f(v) + p(e_1) + e_3 = 0$. The errors can be divided into four classes as follows.

1. $e_1 = e_2$ and $p(e_1) = e_3$, the error will always be missed. The number of errors in this class is 2^n;
2. $e_1 = e_2$ but $p(e_1) \neq e_3$, the error will always be detected. There are 2^n errors belonging to this class.
3. $H(e_1 + e_2) = 0$ but $e_1 \neq e_2$, the error masking probability depends on the nonlinear function f. In the worst case, a specific error will be masked by $P_f \cdot |V|$ codewords, which can be easily derived from the property of nonlinear functions. The number of errors in this class is $2^{n+1} \cdot (2^{n-r} - 1)$
4. $H(e_1 + e_2) \neq 0$. The error in this class will always be detected. The number of errors is $2^{n+1}(2^n - 2^k)$.

Theorem 3.8 *For $x \in GF(2^a)$, let $p(x) = wt(x) \bmod 2$. Let V be a $[n, k]$ not necessarily linear Hamming code with $r = n - k$ redundant bits. Without lost of generality, assume that the first k bits in any codeword are information bits. Denote by $v = (y, z), y \in GF(2^k), z \in GF(2^r)$ the codewords of V. Let $f : GF(2^k) \to \{0, 1\}$ be an arbitrary mapping such that $f(0) = 0$ and $f(y) + f(y') \neq f(y + y')$ for some $y, y' \in GF(2^k)$. The code C defined by*

$$C = \{(x, (0, x) + v, p(x) + f(y))\} \quad (20)$$

where $x \in GF(2^a), 0 \in GF(2^{n-a}), 0 \leq a < n, v \in V$ is a $(a + n + 1, a + k, 3)$ code with $|K| = 2^a$, and $max_{e \notin K} Q(e) = P_f$.

Other constructions of perfect nonlinear codes with have small kernels can be found in [13][14][15][3][5]. Using the switching constructions, perfect nonlinear codes can be constructed with a kernel dimension of one. The maximum $Q(e)$ for these codes, however, is close to one.

Example 3.5 *Let $a = 6$ and V be a $[31, 26, 3]$ perfect Hamming code. Select f to be the same nonrepetitive quadratic function as described in Example 3.1. The generalized Vasil'ev code from Theorem 20 is a $(38, 32, 3)$ partially robust code with $|K| = 2^6$ and $max_{e \notin K} Q(e) = 0.5$.*

4 Robust Protection of AES

For comparison of error detecting characteristics and hardware overheads for robust architectures for private key cryptosystems we use a sub-circuit of the Advanced Encryption Standard [1] which is one of the most used symmetric key algorithms and has been the target of numerous fault injection based attack campaigns. As in most private key algorithms, AES involves bitwise operations over small fields and the algebraic error model is most often observed and used in analysis. We use a sub-circuit of a typical round of encryption and compare architectures based on codes and constructions defined in Section 3.

4.1 Hardware Architecture for Robust AES

The datapath of a typical round of AES-128 consists of four main transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey [1]. The SubBytes transformation involves two operations, inversion in $GF(2^8)$ followed by a linear affine transform. All of the transformations are defined for at most 32-bit operands and the 128-bit datapath of a round of AES can be divided into four independent and identical 32-bit data streams. For the test circuits we used the linear transformations of a 32-bit wide portion of a typical round of AES. The circuit consists of one MixColumns transformation and four affine transformations. It is completely linear and can be implemented with 217 XOR gates.

We compare six different protection methods : linear parity, robust parity (Example 3.1), linear+robust parity(Example 3.2), partially robust $(x, (Hx)^3)$ code from Example 3.4, Hamming code and a partially minimum distance robust code based on Vasil'ev code (Example 3.5). Each code protects one 32-bit linear block.

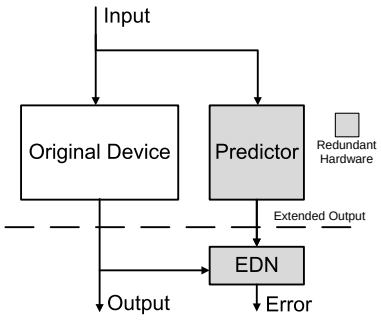

Figure 2. General Architecture

The general architecture utilizing error detection codes to protect devices against fault analysis attacks is shown in Figure 2. In addition to the original device, two extra blocks, the predictor and the error detection network (EDN) are needed. The extended outputs are codewords of the error detection code. The predictor predicts the redundant outputs from the inputs of the devices. EDN is used to verify the integrity of the output data. By selecting an appropriate error detection code and implementing the corresponding predictor and EDN, the desired level of error detection capability can be achieved.

Table 2. Hardware Overhead of Architectures

	predictor	EDN	overhead(%)
linear parity	31	32	30%
robust parity (Example 3.1)	185	32	100%
min. dist. robust (Example 3.2)	196	64	120%
Hamming	253	80	153%
gen. Vasil'ev (Example 3.5)	292	116	188%
$(x, (Hx)^3)$ (Example 3.4)	432	266	322%

The overheads of each of the implementations are summarized in Table 1. The table lists the number of two-input gates required for each implementation and the overhead compared to the unprotected implementation. The linear parity requires very little overhead due to the parity preserving nature of the linear operations. For this sub-circuit of AES, the parity of the inputs is equal to the parity of the outputs which results in a compact parity predictor and a 30% gate count overhead. The robust parity implementation requires the prediction of a nonlinear function of the output and results in a larger overhead. Despite the larger overhead, the error detecting network is the same size as for parity. The implementation which combines the the linear and robust parity into one implementation requires slightly more hardware in the predictor and 32 more gates in the EDN compared to robust parity. The implementations based on Hamming codes, generalized Vasil'ev codes and $(x, (Hx)^3)$ codes require much larger overhead due to the fact that more redundant bits need to be predicted. The overhead of the scheme utilizing generalized Vasil'ev code is slightly higher than Hamming implementation because it needs to compute one nonlinear redundant bit. Finally, the scheme based on $(x, (Hx)^3)$ code requires more than 300% hardware overhead because the predictor and EDN needs to implement a cube operation in $GF(2^6)$ in addition to the matrix multiplication in $GF(2)$.

4.2 Error Detection Analysis

To illustrate the error detection characteristics of robust codes and their variants we first show results of exhaustive

simulations comparing the error detection ability of codes with smaller dimensions.

Figure 3 shows the percentage of errors e that are masked as a function of the the error masking probability $Q(e)$ where the number of information bits is $k = 8$. Linear parity code has the largest portion of undetectable errors (50 %). For robust parity code, all errors are detectable with a probability of at least 0.5. The code with both linear and robust parity bits can detect 50% errors with probability 1 and all the others with probability 0.5. Generalized Vasil'ev code, $(x, (Px)^3)$ code and Hamming code have Hamming distance 3 and can detect all single and double errors which are most probable in practice. The first two have much smaller portion of undetectable errors than Hamming code due to their robustness. For generalized Vasil'ev code nearly 90% of errors are always detectable. $(x, (Px)^3)$ code can detect only 50% errors with probability one, but it can detect another 45% errors with probability 0.875.

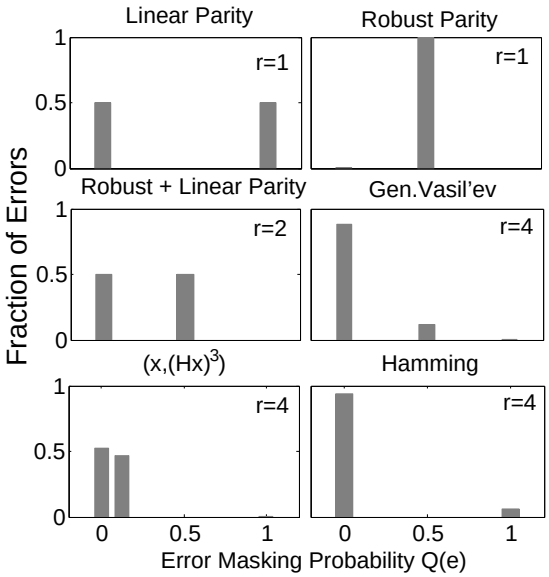

Figure 3. Error Distributions for codes with $k = 8$

The experimental results of fault simulations for the linear sub-circuit of AES protected with the above six different codes are shown in Figure 4 and Figure 5. Single stuck-at faults were injected into the original and predictor portions of the corresponding six designs. Due to the linear function of the AES sub-circuit the faults tend to manifest themselves as repeating errors at the outputs. It is shown in [11] that robust codes have better detection characteristics in channels where errors have a high laziness or probability of repeating themselves. Thereby, robust and minimum distance robust codes are expected to have better performance

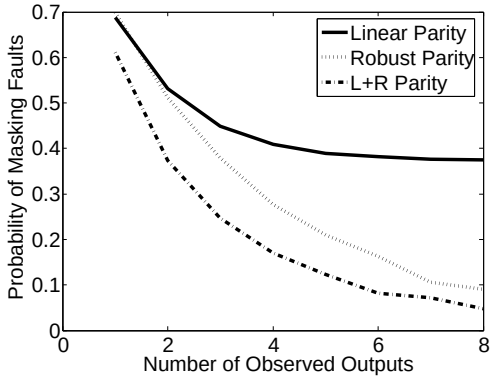

Figure 4. Probability of Missing Faults for different length of input sequences–Linear Parity, Robust Parity, L+R Parity ($k = 32, r = 6$)

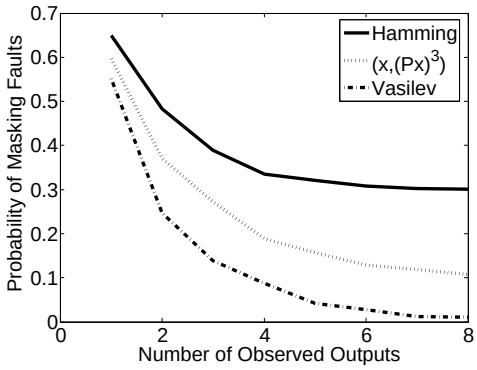

Figure 5. Probability of Missing Faults for different length of input sequences–Hamming, gen.Vasil'ev, $(x, (Px)^3)$ where $k = 32$

when faults stay for several consecutive operations.

For each of the architectures the probability of not detecting a fault at least once decreases as more outputs are observed. Due to the large kernels of the linear parity and Hamming codes and the structure of the circuit based on these codes, about 30% of single faults result in errors which are undetectable. As shown in the Figures, the probability of not detecting at least once a fault after eight messages approaches 30% for both architectures.

Robust codes have no undetectable errors and partially robust codes reduce the number of undetectable errors over linear codes. For the robust and partially robust codes the probability of not detecting a fault at least once is much smaller than linear codes.

We note that protection methods aiming at only increasing the Hamming distance of codes do not bring big improvement for the error detection probabilities as compared

with schemes based on codes with distance 1 or 2. The architecture based on Hamming code is only a little bit better than that based on linear parity code and is much worse than the one based on robust parity code in terms of fault masking probability when several consecutive outputs are observed. The reason is that most of single stuck-at faults will result in single errors or affect an odd number of output bits, which can be detected by linear parity code. If faults do manifest themselves as errors with high multiplicities, Hamming codes still do not have benefits due to the large number of undetectable errors and the disadvantage of detecting repeating errors compared with robust codes or partially robust codes. Thereby, we claim that to further increase the fault detection capability, robust codes and partially robust codes with minimum distances are better choices than linear codes with higher Hamming distances.

5. Robust Protection of Multipliers

The multiplier is a basic building block for many public key cryptosystems. Due to the arithmetical nature of the devices and operations the arithmetical error model is most often observed and used for such devices. We use the multiplier to demonstrate and analyze the error detection properties for arithmetic errors in architectures based on robust and partially robust codes (from Theorems 3.3 and 3.5).

5.1 Hardware Architecture for a Robust Multiplier

The general hardware architecture of a multiplier protected with the robust and partially robust encoding method are shown in Figure 6. Both of the architectures consists of the regular multiplier, a predictor circuit (shown inside a dotted box in the Figure), and an error detecting network (EDN) (shown in the dashed box in the Figure).

For the partially robust architecture based on the code from Theorem 3.5 the predictor performs multiplication $(\mathrm{mod}\ p)$ followed by a squaring operation modulo p as depicted in Figure 6. The reduced operands of the predictor can be assumed available from previous computations or easily recomputed in the predictor. The error detecting network consists of a modulo p reduction step of the output followed by a squaring operation, also modulo p. The squaring and redundant multipliers in the predictor and the squaring operation in the EDN are performed on $r = \lceil \log_2(p) \rceil$-bit operands.

The overhead and the protection level for the multiplier can be selected by the choice of the modulus p. For most applications a $r < k$ is used and the overhead of the architecture can be made less than 100%.

The architecture is comparable to the classical linear architecture with the addition of a squaring operations modulo

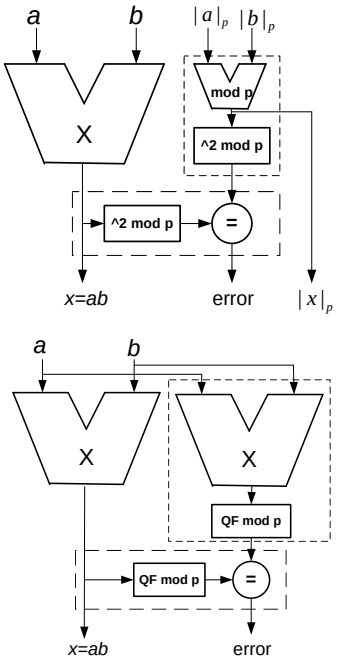

Figure 6. General architecture for a (a) partially robust multiplier and (b) robust multiplier

p in the predictor and the EDN. The hardware complexity of the squaring operation is of the order of $O(r^2)$, or comparable to a multiplier. The squaring operations thus add a significant additional overhead compared to a linear architecture. The extra overhead is acceptable in some applications since the robust architecture will have fewer "bad" errors than the linear architecture with any r where "bad" errors are considered those that are missed by more than 50% of messages. Increasing the redundancy or rather p in the linear architecture will only reduce percentage of bad errors but the overall number of them will not change.

For the robust architecture based on the code from Theorem 3.3 the predictor performs multiplication followed by the computation of the non-repetitive quadratic form (QF) modulo p. The computation of the quadratic form depends on the selection of t and s (see Theorem 3.3). The quadratic form requires $s/2$ multiplications modulo p as well as $s/2 - 1$ additions modulo p. The error detecting network computes the quadratic form of the output of the original multiplier and a comparison of the signature of the predictor.

A hardware overhead comparison of the linear, robust, and partially robust architectures is shown in Table 3 for a multiplier with $k = 64$ bit output and 32 bit operands. The table assumes a $O(m^2)$ (where m is the size of the operands) hardware complexity for a multiplier and a pes-

simistic $O(m^2)$ complexity for a the modular squaring operation. The table also shows the theoretical number of bad errors and probabilities of these bad errors for the codes with the parameters.

Table 3. Comparisons for a $k = 64$ bit multiplier

	Linear	Partially Robust	Robust
r	16	16	16
overhead	25 %	75%	200%
# of bad errors	$\approx 2^{65}$	$\approx 2^{49}$	0
prob. of bad error	$\approx 2^{-15}$	$\approx 2^{-31}$	0

As the table shows there is a large difference in hardware overheads depending on the robustness of the architecture. The more robust the architecture the more hardware overhead is necessary with the fully robust architecture requiring an overhead of more than 100%. It is important to point out however, that in terms of the number of bad errors the linear architecture cannot reach the protection of the robust architectures regardless of the number of redundant bits and hardware overhead added.

5.2 Error Detection Analysis for Nonuniformly Distributed Outputs

The proofs for the construction of arithmetic robust and partially robust codes assumed a uniform distribution of output messages. However, many arithmetic devices do not have uniformly distributed outputs even if inputs to the device can be considered uniformly distributed.

For a k-bit multiplier the output of the multiplier is not a uniform distribution on $\mathbb{Z}_{2^k}$. The range is limited to $(2^{k/2} - 1)^2$. Likewise some outputs, such as prime numbers greater than $2^{k/2}$ cannot be represented as a product of two $k/2$-bit inputs and hence also do not appear at the output, while other outputs, such as the zero output, appear multiple times. The masking probabilities of Theorem 3.5 can be rewritten to include the nonuniform distributions.

Let $\varphi_k(x)$ denote the number of ordered pairs (a, b) such that $x = ab, a, b \in \mathbb{Z}_{2^{k/2}}$. Assuming that the input operands a, b are equiprobable, the probability that x, the output of the multiplier, is $\varphi_k(x)2^{-k}$. Let $S(e_x, e_w)$ be the set of messages x that mask an error (e_x, e_w). The probability $Q(e_x, e_w)$ of missing an error can be written as

$$Q(e_x, e_w) = 2^{-k} \sum_{x \in S(e_x, e_w)} \varphi_k(x) \qquad (21)$$

An example error distribution comparing the error detection of $k = 8$-bit multiplier protected with linear arithmetic residue codes, partially robust codes, and the robust codes (where $p = 5$) is shown in Figure 7. The figures show histograms of the number of errors that are masked for a specific number of inputs to the multiplier. The figures clearly show most of the properties of robustness are preserved despite the nonuniform distribution of the ouputs.

The number of undetectable and bad errors is greatly reduced with the partially robust encoding versus the linear encoding. When the partially robust codes are used almost all of the errors are detected with a high probability. Only a few errors are masked for more than 100 messages. The linear residue code has p times more errors which have a high probability of being masked. It can be shown that even with non uniform message distributions the number of errors in each of the error classes of the partially robust codes remains the same, but the exact probability of masking depends on $\varphi_k(x)$. The number of errors which belong to the "bad" class and are masked with an average probability of $\gtrsim 0.5$ remains on the order of 2^{k+1}. The number of "bad" errors remains on the order of $2\lfloor 2^k/p \rfloor$.

The architecture based on fully robust codes based on quadratic forms showed further improvement in the reduction of errors that are masked with a high probability, almost all errors are have a probability of detection of at least 0.5. However, the non-uniform distribution increases the theoretical maximum error masking probability. For the code with $k = 8$ and $p = 5$ the worst case error masking probability should be at most 2^{-7} but several errors exist which are missed with a slightly higher multiplicity. The exact effect of the output distribution of the multiplier on the properties of the proposed robust code is under investigation.

6. Conclusions

The comparisons of protection methods shows that there is a large difference in hardware overheads depending on the robustness of the architecture. The more robust the architecture the more hardware overhead is necessary. While robust protection can be implemented efficiently for some highly nonlinear circuits such as SBoxes [7], fully robust architectures for more general circuits require an overhead of more than 100%. Partially robust codes that preserve some linear structures allow for a better minimization of the predictor and have an overhead of the order of 40-75%.

Robust architectures do offer an advantage against unpredictable fault attacker since the number of undetectable and bad errors however eliminated with robust codes and is greatly reduced with the partially robust encoding. It is important emphasize that in terms of the number of bad or undetectable errors, linear architecture cannot reach the protection of the robust architectures regardless of the number of redundant bits and hardware overhead added.

For attacks where faults of low multiplicities are known to be most likely robust codes and partially robust codes

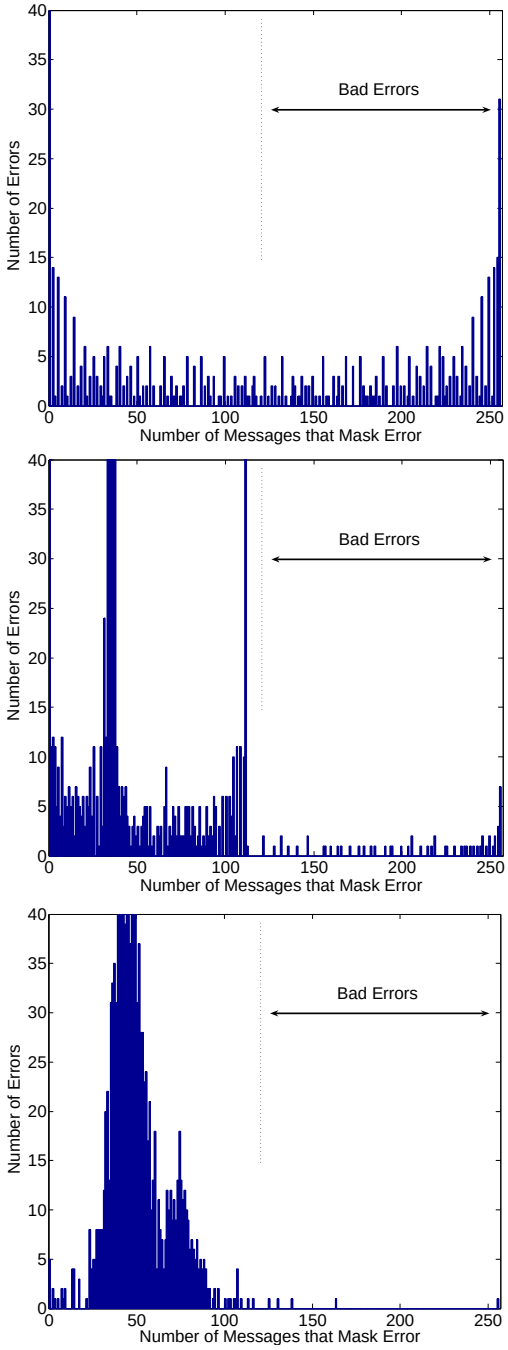

Figure 7. Multiplier error distributions ($k = 8, p = 5$) with (a) linear (b) partially robust and (c) robust code protection

with minimum distances resulted in better fault detection than linear codes with even higher Hamming distances.

The non-uniform distributions of outputs (such as those in multipliers) can have a negative impact on the robustness of a code and need to be included in design considerations. While the robust construction used for the protection of the multiplier showed important benefits, their worst case error masking probabilities were worse compared to theoretical estimates based for uniform message distributions.

References

[1] Fips pub 197: Advanced encryption standard. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. *In Proceedings of the IEEE*, 94(2):370–382, 2006.

[3] H. Baucer, B. Ganter, and F. Hergert. Algebraic techniques for nonlinear codes. In *Combinatoria*, volume 3, pages 21–33, 1983.

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

[5] T. Etzion and A. Vardy. Perfect binary codes: Constructions, properties, and enumeration. In *IEEE Trans. on Information Theory*, volume 40, pages 754–763, 1994.

[6] G. Gaubatz, B. Sunar, and M. G. Karpovsky. Non-linear residue codes for robust public-key arithmetic. In *(FDTC '06)*, 2006.

[7] 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.

[8] M. Karpovsky, K. Kulikowski, and A.Taubin. Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard. (DSN 2004), July 2004.

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

[10] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin. Robust codes and robust, fault tolerant architectures of the advanced encryption standard. *Journal of Systems Architecture*, 53:138–159, 2007.

[11] M.G.Karpovsky, K. Kulikowski, and Z. Wang. Robust error detection in communication and computation channels. Keynote paper, Int. Workshop on Spectral Techniques, 2007.

[12] M. Mollard. A generalized parity function and its use in the construction of perfect codes. In *SIAM J. Alg. Disc. Meth*, volume 7, pages 113–115, 1986.

[13] K. T. Phelps. A combinatorial construction of perfect codes. In *SIAM J. Alg. disc Meth.*, volume 4, pages 398–403, 1983.

[14] K. T. Phelps. A general product construction for error-correcting codes. In *SIAM J. Alg. disc Meth.*, volume 5, pages 224–228, 1984.

[15] F. I. Solov'eva. On binary nongroup codes. In *Methodi Diskr. Analiza*, volume 37, pages 65–76, 1981.

[16] J. L. Vasil'ev. On nongroup close-packed codes. In *Probl.Kibernet.*, volume 8, pages 375–378, 1962.