Journal of Cryptographic Engineering (2021) 11:147–160
https://doi.org/10.1007/s13389-020-00229-4

REGULAR PAPER

IPM-RED: combining higher-order masking with robust error detection

Osnat Keren[1] · **Ilia Polian**[2]

Received: 15 November 2019 / Accepted: 28 April 2020 / Published online: 19 June 2020

Abstract

Cryptographic hardware becomes increasingly vulnerable to physical attacks—both passive side-channel analysis and active fault injections—performed by skillful and well-equipped adversaries. In this paper, we introduce a technique that provides very high security against both types of attacks. It combines inner product masking (IPM), which offers higher-order side-channel attack resistance on word level and on bit level, with nonlinear security-oriented error-detection codes that provide robustness, i.e., strong detection guarantees for arbitrary faults. We prove that our scheme has the same security against side-channel attacks that an earlier, non-robust IPM-based solution has and in addition preserves robustness during addition and multiplication (and therefore arbitrary computations). Moreover, we prove that the information leakage from the checker is small and that the attack will be detected far before the attacker will gain significant information.

1 Introduction

Confidentiality, integrity and authenticity of sensitive information processed by integrated circuits are protected by cryptographic primitives, which in turn are endangered by various physical attacks [2]. Security-critical systems must withstand different attacks vectors, most notably, side-channel analysis and fault-injection attacks. Countermeasures that are highly effective against one class of attacks have been known for a long time, and there is a good understanding of the trade-off between a specific countermeasure's effectiveness in protecting a system against an attacker with certain capabilities and the cost associated with that countermeasure. It turns out, however, that countermeasures are not easily composable and adding a countermeasure, against, e.g., fault attacks, can worsen the system's resistance against side-channel threats [3,4].

In this article, we propose the scheme IPM-RED, which combines two sophisticated protections against fault and side-channel attacks, and prove formally that this scheme retains security guarantees related to each of the classes. "IPM" stands for "inner product masking" [5], a side-channel countermeasure that supports user-specified security orders both at the word and at the bit level. "RED" points to "robust error detection," an approach toward detecting (and correcting) maliciously injected faults with a guaranteed minimal probability of detection [6]. IPM-RED extends an earlier scheme IPM-FD [7], which was also based on enhancing inner product masking with fault detection but used a non-robust error-detection approach (a repetition code). Non-robustness implies the existence of errors with 0% detection probability, and an adversary with very precise fault-injection equipment could systematically mount fault attacks without ever being detected. The same is true for other combined side-channel/fault-attack countermeasures based on non-robust codes, e.g., parity codes [4]. Note that IPM-RED does not use information theoretic MAC tags which require fresh (random) MAC keys for each operation [8].

Our specific contribution is as follows:

- We combine, for the first time (to the best of our knowledge) an advanced masking scheme with robust error detection in a way that neither reduces side-channel resistance nor error-detection guarantees.

Parts of this paper were presented at the Proceedings of 8th International Workshop on Security Proofs for Embedded Systems, PROOFS 2019 [1]. This research was supported by the ISRAEL SCIENCE FOUNDATION (Grant No. 923/16) and by the DFG (German Research Foundation) Project Po 1220/7-2 "Algebraic Fault Attacks" .

✉ Ilia Polian
ilia.polian@informatik.uni-stuttgart.de

Osnat Keren
osnat.keren@biu.ac.il

[1] Faculty of Engineering, Bar-Ilan University, Ramat Gan, Israel

[2] Institute of Computer Engineering and Computer Architecture, University of Stuttgart, Stuttgart, Germany

- We provide formal proofs that all security orders and bounds are preserved during addition and multiplication in the IPM-RED representation. Since arbitrary computations can be reduced to additions and multiplications [7], IPM-RED enables universal computations in the masked domain. We analyze the computational complexity of the operations and find them tolerable, despite the fact that sophisticated nonlinear codes are used.
- We formalize the problem of quantifying information leakage from the checker (assuming an adversary can observe its output) by means of mutual information. We prove that this leakage is of no practical concern: the probability that an attack will go undetected is exponentially smaller than the entropy loss due to information leak from the checker.

The remainder of the paper is organized as follows: The next section starts with introducing notation, and then physical attacks and masking strategies are reviewed; the earlier work [7] is recapitulated, and the various security metrics that IPM-RED aims to preserve are summarized. Section 3 introduces IPM-RED, discusses its construction and proves its security. Section 4 contains proof that security is preserved during operations in the masked domain. In Sect. 5, information leakage from IPM-RED (and IPM-FD) checkers is discussed and its extent is proven[1]. Section 6 concludes the paper.

2 Preliminaries

2.1 Notations

We denote by $\mathbb{F}(q) = GF(q)$ a finite field of characteristic 2 and size $q = 2^s$. For example, $q = 2^4$ for PRESENT and $q = 2^8$ for AES. Scalars and vectors over this field are denoted by lowercase letters (e.g., $x \in \mathbb{F}_q$, $y \in \mathbb{F}_q^k$), and matrices are denoted by capital letters (e.g., $A_{k_1 \times k_2} \in \mathbb{F}_q^{k_1 \times k_2}$). An all zero (one) vector of length k is denoted by $\mathbf{0}_k$ ($\mathbf{1}_k$), and an identity matrix of dimension k is denoted by $I_{k \times k}$. When the dimensions of a matrix are clear from the context we omit them.

Calligraphic letters denote sets (e.g., $\mathcal{C} \subseteq \mathbb{F}_q^k$). The fields $\mathbb{F}_q$ and the vector space $\mathbb{F}_2^s$ are isomorphic; thus, a q-ary vector of length k can be referred to as a binary vector of length $k_b = s \cdot k$ bits and vice versa. For example, for a PRESENT cipher, the plaintext is a binary vector of length $k_b = 64$; however, in an inner product masking scheme [5], it is referred to as a q-ary vector of length 16 over $\mathbb{F}_{16}$.

We use parentheses to denote concatenation of vectors; that is, for $x \in \mathbb{F}_q^{k_1}$ and $y \in \mathbb{F}_q^{k_2}$ the vector (x, y) is a q-ary vector of length $k_1 + k_2$. Finally, the inner product of two vectors $x, y \in \mathbb{F}_q^k$ is denoted by $< x, y >$; that is, $< x, y >= \sum_{i=1}^{k} x_i y_i$ where the computation is performed in the $\mathbb{F}_q$.

2.2 Fault-injection and side-channel attacks

It has been noticed in [9] that faults that occur during a cryptographic operation can compromise its security. Fault-injection attacks utilize this fact by deliberately inducing a physical disturbance during the execution of a cryptographic function. A plethora of fault-injection techniques have been demonstrated, including inducing glitches on the circuit's inputs or clock lines; overheating; underpowering; applying electromagnetic pulses; and illuminating the circuit with a laser [10,11]. The best known attacks, which need a single fault injection to recover the complete secret key of a cipher [12,13], require very high temporal and spatial resolution, i.e., a fault injection in a precisely known clock cycle at a precisely known portion of the circuit. Even the most sophisticated fault-injection techniques [14] currently do not provide such resolution, and as a consequence, an attack must be attempted several times. Robust codes used in this paper make a conservative assumption of the attacker: namely that the attacker can precisely select an arbitrary number of specific bits that will be flipped as a consequence of the fault injection. This assumption is captured by defining the error as a binary vector which is added (XORed) to the circuit's outputs and errors which the attacker can choose.

Once a fault has been injected, the attacker can analyze the circuit's outputs (ciphertexts in the case of encryption) obtained in the presence and in the absence of the disturbance. Applying differential cryptanalysis techniques, secret key bits can be inferred. It is also possible to collect a larger number of the circuit's responses for faults of different intensities and to infer the key from a statistical analysis [15,16]. In this paper, we make no assumption about which type of fault analysis the attacker will perform. Any injected fault that goes undetected by the robust code is considered critical and the codes aim at increasing the probability of detection. The question considered is whether knowing the check bits gives the attacker additional information about the secret key and how much such information leakage can occur.

Another relevant class of physical attacks is side-channel analysis [17], where an adversary observes a circuit's behavior without actively manipulating it. For example, observed delays or power consumption can allow the attacker to infer bits of a secret key, or at least restrict the number of potential

[1] This journal article extends the comment [1] which appeared at the PROOFS workshop 2019. The material from the workshop paper is largely restricted to Sect. 5. Note that the workshop paper [1] focused on information leakage from an unmasked circuit and did not consider secure composition of protections against side-channel and fault attacks.

key candidates to a value for which brute-first search becomes feasible. Side-channel analysis is typically done using statistical techniques such as differential [18] or correlation power analysis [19]. Vulnerability of an implementation to side-channel analysis can be captured by information-theoretical concepts such as mutual information [20].

2.3 Masking against side-channel attacks

Masking is a well-known countermeasures for block ciphers against side-channel attacks. It is based on data randomization where any key-dependent q-ary variable is split into ν random and secret shares. The shares are processed in a way that preserves the correctness of computations and ensures that intermediate values remain independent of the sensitive variables. In general, the security order of a masking scheme is defined as the largest number above which it is possible to exploit secret information from the protected implementation, e.g., by probing wires of the protected circuit. In Sect. 2.5, we describe five security metrics that can be used to evaluate the security order of a masking scheme against side-channel and fault injection attacks.

In order to provide end-to-end security, all the cipher operations are performed in the mask domain and they must be secure. That is, any cryptographic algorithm has to start with plaintext and key in ν share representations, carry out addition and multiplication operations to implement the algorithm, and end up with a ciphertext that still has share representation. The most challenging part of cipher implementations is the design of a provable secure nonlinear S-box , which involves finite field arithmetic.

There are many types of masking. The best known examples are Boolean masking [21], masked computations over finite fields [22] and polynomial masking based on Shamir's secret-sharing and secure multi-party computation techniques [23,24], inner product masking (IPM) [5]. Boolean masking, which involves computations over $\mathbb{F}_2$, is widely used because of its simplicity, ease of implementation and comparatively low-performance overhead. Masking techniques with higher algebraic complexity, i.e., defined over fields of size $q = 2^s$ with $s = 4, 8$, provide more security than Boolean masking at the cost of higher overheads; for example, the software implementation of the inner product masking in [5] is said to be four times slower than Boolean masking.

The occurrence of glitches and maliciously injected faults can lead to side-channel information leakage [25]. A circuit-level approach to secret sharing, threshold cryptography and multi-party computation protocols based on masking schemes that resist side-channel attacks in the presence of glitches is presented in [26]. In [4], a countermeasure for cryptographic hardware implementations, dubbed ParTI, that combines a threshold implementation (TI) with additional parity bits against fault injection was introduced. As shown in [3], because the parity bits can leak information, a threshold implementation of the redundant bits is required to prevent side-channel information leakage from these bits. The codes in [4] are linear with a minimum distance of $d > 1$; thus, any number of bit-flips smaller than $d - 1$ is always detected. However, due to the linearity of the code, many error patterns are never detected. In [8], the masked linear redundant bits were replaced by masked MAC in order to detect an arbitrary number of errors.

2.4 Linear code-based approach for detecting faults in IPM schemes

Inner product masking (IPM) is a special case of direct sum masking (DSM). In DSM, the secret is a symbol x in $\mathbb{F}_q$. The secret is represented as sharing, i.e., a tuple of ν field elements,

$$z = (x, m)L = xG + mH \in \mathbb{F}_q^{\nu}$$

where $m = (m_2, m_3, \ldots, m_{\nu}) \in \mathbb{F}_q^{\nu}$ is the random mask drawn uniformly, the matrix $L = (l_1^T, l_2^T, \ldots, l_{\nu}^T)$ is a $\nu \times \nu$ matrix where $l_i = (l_{1,i}, l_{2,i}, \ldots, l_{\nu,i}) \in \mathbb{F}_q^{\nu}$ and L is of the form

$$L = (l_1^T, l_2^T, \ldots, l_{\nu}^T) = \left(\begin{array}{c|ccc} 1 & 0 & 0 & \ldots & 0 \\ \hline l_{2,1} & 1 & 0 & \ldots & 0 \\ l_{3,1} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \vdots & \\ l_{\nu,1} & 0 & 0 & \ldots & 1 \end{array}\right). \quad (1)$$

The subspace spanned by G is referred to as the code $\mathcal{C}$. The matrix H is a $(\nu - 1) \times \nu$ matrix. The subspace spanned by the rows of H is referred to as the code $\mathcal{D} \subset \mathbb{F}_q^{\nu}$. The dual code of $\mathcal{D}$ is denoted by $\mathcal{D}^{\perp}$.

The value of x is demasked (recovered) by computing $x =< z, l_1 >$. In orthogonal DSM, $GH^T = 0$; hence, it is possible to recover the masking vector m knowing z by projecting it onto $\mathcal{D}$ [27]. This serves to check that m has not been tampered with since it was last refreshed.

Recently, a new masking scheme dubbed IPM-FD which is built on IPM and allows for fault detection by using linear codes was presented [7]. IPM-FD is secure both at the word-level and the bit-level probing models and allows for end-to-end fault detection against fault-injection attacks. In the IPM-FD masking scheme, the secret information is protected by a repetition code $\mathcal{C}_0$ of length μ. That is, IPM-FD defines a set $\mathcal{Z}$ of legal words (codewords)

$$z = xG + mH = (x, x \ldots, x, m)L \in \mathcal{Z},$$

where $x \in \mathbb{F}_q$, $m \in \mathbb{F}_q^{\nu-\mu}$ and

$$G = (\mathbf{1}_\mu \mid \mathbf{0}_{\nu-\mu}) \in \mathbb{F}_q^{\nu},$$
$$H = (\hat{L}_{(\nu-nur)\times\mu} \mid I_{(\nu-\mu)\times(\nu-\mu)}) \in \mathbb{F}_q^{(\nu-\mu)\times\nu}.$$

The corresponding matrix L is of the form

$$L = (l_1^T, l_2^T, \ldots, l_\nu^T) = \left(\begin{array}{ccc|c} 1 & 0 & \ldots\ 0 & 0\ 0\ \ldots\ 0 \\ 0 & 1 & \ldots\ 0 & 0\ 0\ \ldots\ 0 \\ & \vdots & & \vdots \\ 0 & 0 & \ldots\ 1 & 0\ 0\ \ldots\ 0 \\ \hline l_{\mu+1,1} & l_{\mu+1,2} & \ldots\ l_{\mu+1,\mu} & 1\ 0\ \ldots\ 0 \\ l_{\mu+2,1} & l_{\mu+2,2} & \ldots\ l_{\mu+2,\mu} & 0\ 1\ \ldots\ 0 \\ & \vdots & & \vdots \\ l_{\nu,1} & l_{\nu,2} & \ldots\ l_{\nu,\mu} & 0\ 0\ \ldots\ 1 \end{array}\right)$$
$$= \left(\begin{array}{c|c} I_{\mu\times\mu} & \mathbf{0}_{(\nu-\mu)\times(\nu-\mu)} \\ \hline \hat{L} & I_{(\nu-\mu)\times(\nu-\mu)} \end{array}\right). \quad (2)$$

Note that the set of legal sharings $\mathcal{Z}$ is a linear q-ary code of length ν and dimension $\nu - \mu + 1$.

The demasking operation maps a vector $v \in \mathbb{F}_q^\nu$ to a vector $y \in \mathbb{F}_q^\mu$,

$$y = (y_1, \ldots, y_\mu) =< v, (l_1, \ldots, l_\mu) > .$$

Then, a checker verifies that y is a legal codeword of the repetition code $\mathcal{C}_0$, i.e., $y_1 = y_2 \cdots = y_\mu$. In a fault-free scenario, the demasking extracts the correct x from z.

The IPM-FD masking scheme requires secure computation in the mask domain that can be achieved by Lagrange interpolation using addition and multiplication operations [7]. The authors in [7] defined a secure addition and a secure multiplication of IPM-FD sharings for which verification can be carried out at the very end by comparing the μ copies of the ciphertext. Figure 1 depicts a schematic PRESENT architecture protected by inner product masking. As shown in the figure, 16 mask-nibble blocks transform each input nibble into 4ν-bit sharing and then encryption is performed in the masking domain; that is, the output of each computation step is a codeword in $\mathcal{Z}$. When the encryption ends, 16 demask-nibble blocks extract a ciphertext from the 4ν-bit sharings. An implementation of a PRESENT cipher protected by IPM-FD requires checker blocks. These checkers should be placed at the output of the demask-nibble blocks.

2.5 Security metrics

The effectiveness of masking against a side-channel attack or a fault injection is evaluated via the following security orders [28]:

- Word-level (nibble/byte) security order d_w against a word-level probing attack. d_w equals to $d_D^\perp - 1$, where $d_D^\perp$ is the minimum distance of the q-ary code $\mathcal{D}^\perp$ (equivalently the dual distance of $\mathcal{D}$).
- Bit-level security order d_b against a probing attack on single or/and several bits that can be probed simultaneously. The bit-level security order d_b equals $d_{D_2}^\perp - 1$ where $d_{D_2}^\perp$ is the dual distance of the binary expansion code $\mathcal{D}_2$ derived from $\mathcal{D}$ by replacing each q-ary element in $\mathcal{D}$ by its sub-field representation as a (4×4) (or (8×8)) binary matrix.
- Word-level error detection order d_e against random (benign) and malicious fault injections. d_e is the maximal number of erroneous words (nibbles/bytes) that can be detected by every codeword in $\mathcal{C}$. Thus, $d_e = d_C - 1$, where the minimum d_C is the distance of the q-ary code $\mathcal{C}$. For example, the G matrix in Eq. 2 defines the code $\mathcal{C}$ which is the repetition code $\mathcal{C}_0$ padded with $(\nu - \mu)$ zeros. Hence, its minimal distance is $d_e = \mu - 1$.
- Bit-level error detection order d_f against random(benign) and malicious fault injections. d_f is the maximal number of bit flips that can always be detected by every codeword in $\mathcal{C}$. For example, the IPM-FD scheme has $d_f = d_{c_2} - 1$, where d_{c_2} is the minimum distance of the expansion code $\mathcal{C}_2$ derived from $\mathcal{C}$. Thus, $d_e = \mu - 1$.

In this paper, we suggest a fifth security metric that relates to the robustness of the masking scheme against faults that cause an arbitrary (unlimited) number of bit flips.

- Arbitrary-error detection rate d_a. The value of d_a indicates how robust the code is against a sophisticated attacker that can apply any error pattern it chooses. That is, $d_a = 1 - \bar{Q}$, where

  $$\bar{Q} = \max_{e \in \mathbb{F}_q^\nu \setminus \mathcal{D}} |\{z\ :\ z + e \in \mathcal{Z}\}|/|\mathcal{Z}|$$

  is the probability that an injected error pattern will pass unnoticed.

Note that in the definition of the arbitrary-error detection rate we exclude error patterns (vectors) which are codewords of $\mathcal{D}$ because adding an error $e \in \mathcal{D}$ to a legal codeword of $\mathcal{Z}$ is equivalent to a mask-refresh operation. In other words, an error $e \in \mathcal{D}$ can be represented as $e = (\mathbf{0}_\mu, w)L = (w\hat{L}, w)$ where $w \in \mathbb{F}_q^{\nu-\mu}$, hence

$$< e, (l_1, \ldots l_\mu) >= (w\hat{L}, w)(I, \hat{L})^T = \mathbf{0}_\mu.$$

Therefore, e does not change the secret, i.e.,

$$< c + e, (l_1, \ldots l_\mu) >=< c, (l_1, \ldots l_\mu) >,$$

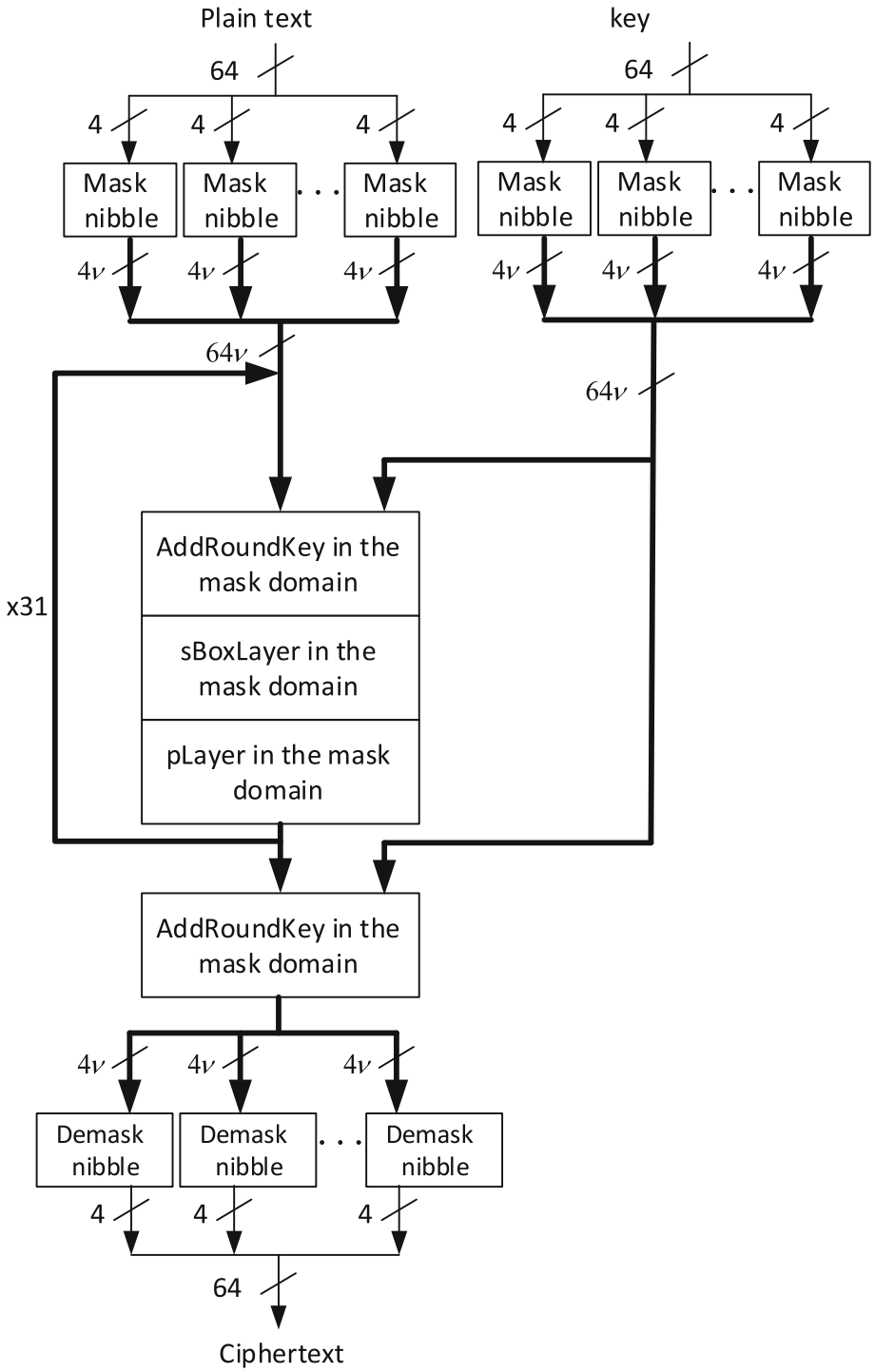

Fig. 1 Inner product masking implementation of PRESENT

it only changes the mask elements. Since a mask refresh is a legal operation, $e \in \mathcal{D}$ cannot be considered by the code as "malicious". It is assumed that a system-level mechanism controls the refresh procedure and has ways to detect an unauthorized refresh.

Note that the IPM-FD masking scheme in [7] is based on linear codes and thus it does not provide robustness ($d_a = 0$). In fact, any fault-injection attack for which $e \in \mathcal{Z}$ will go undetected.

3 Inner product masking with robust error detection (IPM-RED)

In this section, we present a fault detecting IPM scheme based on robust systematic codes of rate one-half. In general, for systematic codes, the undetected error probability of the code is determined by the nonlinearity of the functions used to compute the redundant portion. There are several ways to define nonlinearity; in this paper, we define nonlinearity with respect to the derivative of the function:

Definition 1 [29] Let F be a function $\mathbb{F}_q \mapsto \mathbb{F}_q$. For any $a \in \mathbb{F}_q$, the derivative of F with respect to a is the function D_aF from $\mathbb{F}_q$ to itself defined by

$$D_aF(x) = F(x+a) \ominus F(x), \forall x \in \mathbb{F}_q.$$

The nonlinearity of function $F : \mathbb{F}_q \mapsto \mathbb{F}_q$ can be measured by using Def. 1. Let

$$\Delta(a,b) = |\{x \in \mathbb{F}_q | D_aF(x) = b\}|.$$

The nonlinearity of function F is

$$\Delta(F) = \max_{0 \neq a, b \in \mathbb{F}_q} \Delta(a,b).$$

The smaller the value of $\Delta(F)$, the higher the corresponding nonlinearity of F.

Definition 2 [30] A function $F : \mathbb{F}_q \mapsto \mathbb{F}_q$ has perfect nonlinearity if $\Delta(F) = 1$.

Perfect nonlinear functions do not exist for all sets of parameters. In the case of binary fields, for example, there are no perfect nonlinear functions from $\mathbb{F}_q$ to itself [30]. In this case, functions with optimum nonlinearity are almost perfect nonlinear functions, that is,

Definition 3 [29] Let q be a power of two, and let F be a function $\mathbb{F}_q \mapsto \mathbb{F}_q$. We have

$$\Delta(F) \geq 2$$

and the functions for which equality holds are said to be almost perfect nonlinear (APN).

Equivalently, if for any $a, b \in \mathbb{F}_q$, $a \neq 0$, an equation of the form

$$F(x) + F(x+a) = b \tag{3}$$

has no more than two solutions in $\mathbb{F}_q$, then F is APN. In terms of realization cost, the cubic APN function $F(x) = x^3$ is the simplest to implement.

Consider the following inner product masking scheme with robust error detection (IPM-RED).

Construction 1 (IPM-RED with ν shares) *Let $x \in \mathbb{F}_q$ be the secret symbol, and let $m = (m_1, \ldots, m_{\nu-2}) \in \mathbb{F}_q^{\nu-2}$ be random mask drawn uniformly. Define,*

$$z = (x, x^3, m)L \in \mathcal{Z},$$

where L is the corresponding $(\nu \times \nu)$ q-ary matrix

$$L = (l_1^T, l_2^T, \ldots, l_\nu^T) = \left(\begin{array}{cc|c} 1 & 0 & 0\ 0 \ldots 0 \\ 0 & 1 & 0\ 0 \ldots 0 \\ \hline l_{3,1} & l_{3,2} & 1\ 0 \ldots 0 \\ l_{4,1} & l_{4,2} & 0\ 1 \ldots 0 \\ \vdots & & \vdots \\ l_{\nu,1} & l_{\nu,2} & 0\ 0 \ldots 1 \end{array}\right). \tag{4}$$

The set of legal words forms a nonlinear q-ary code $\mathcal{Z}$ of length ν, dimension $\nu - 1$ and its minimum distance is at most two. Thus, the code has no correction capability, and the non-bijective APN function $F(x) = x^3$ is good enough for this construction.

The sharing z can be represented as

$$z = (x, x^3, \mathbf{0}_{\nu-2}) + mH \in \mathcal{Z},$$

where

$$H = (L_{(\nu-2)\times 2} \mid I_{(\nu-2)\times(\nu-2)}) \in \mathbb{F}_q^{(\nu-2)\times\nu}, \tag{5}$$

is the generator matrix of the code $\mathcal{D}$, the pairs (x, x^3) form a nonlinear code $\mathcal{C}_0$ of cardinality q and thus the code $\mathcal{C}$ is the code $\mathcal{C}_0$ padded with $(\nu - \mu)$ zeros. That is, it consists of the codewords $(c, \mathbf{0}_{\nu-2})$ where $c \in \mathcal{C}_0$,

The demasking extracts the pair

$$(y_1, y_2) =< z, (l_1, l_2) > \in \mathbb{F}_q^2.$$

A checker raises a decoding error flag if $(y_1, y_2) \notin \mathcal{C}_0$; that is, if $y_2 \neq y_1^3$.

Security orders of IPM-RED.

The word-level security order d_w and the bit-level security order d_b are determined by the structure of H so that in this sense this masking scheme is equivalent to IPM-FD with $\mu = 2$.

The word-level and bit-level error detection orders (d_e and d_f, respectively) depend on the properties of the code $\mathcal{C}$; or equivalently, on the properties of the APN function. Note that for bijective functions we have $d_e = d_f = 1$. The function x^3 is a bijective function only when q is an odd power of two (i.e., s is odd). When q is an even power of two, using x^3 results in $d_e = d_f = 0$. Note, however, that there are bijective APN functions of the form $x^3 + ax^2 + bx$ for which $d_e = 0$ and $d_f = 1$; see for example [31].

In terms or robustness, any attack that aims to change the demasked information symbol x is detected with nonzero probability. The following example clarifies this statement.

Example 1 Let $q = 2^8$. Consider an inner product masking scheme with $v = 4$ shares. Let $\alpha \in \mathbb{F}_q$ be the root of the primitive polynomial $\pi(x) = x^8 + x^4 + x^3 + x + 1$. The following L matrix is taken from [7],

$$L = \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline \alpha^8 & \alpha^{27} & 1 & 0 \\ \alpha^{20} & \alpha^7 & 0 & 1 \end{array}\right). \tag{6}$$

We start with the IPM-FD scheme. The security orders of this scheme are $d_b = 6$, $d_w = 2$ and $d_e = d_f = 1$. However, this scheme is not robust; consider, for example, the error vector

$$e = (\alpha^{157}, \alpha^{179}, \alpha^5, \alpha^{34}) = (\alpha, \alpha, \alpha^5, \alpha^{34})L \in \mathbb{F}_q^v \setminus \mathcal{D}.$$

For any sharing $z = (x, x, m)L \in \mathcal{Z}$ we have $(y_1, y_2) =< z + e, (l_1, l_2) >= (x + \alpha, x + \alpha)$; since $y_1 = y_2$, the error is never detected. Therefore, we have, $d_{a,\text{IPM}-\text{FD}} = 0$.

The security orders d_b and d_w are determined from the properties of the L matrix; thus, the IPM-RED and the IPM-FD provide the same immunity against probing attacks. The function x^3 is not bijective over $\mathbb{F}_2^8$; thus, we have $d_e = d_f = 0$.

As we show next, the IPM-RED masking scheme is robust; that is, $d_{a,\text{IPM}-\text{RED}} > 0$. For example, the error e is detected by some sharings (e.g., $z = (\alpha^{64}, \alpha^{105}, \alpha^{44}, 1)$ and undetected by others (e.g., $z = (\alpha^{44}, \alpha^{78}, \alpha^{44}, 1)$). □

Formally,

Lemma 1 *The arbitrary-error detection rate of the code $\mathcal{Z}$ as defined by the IPM-RED scheme is $d_{a,\mathcal{Z}} \geq 1 - 2/q$.*

Proof First, recall that we do not consider error vectors $e \in \mathcal{D}$ because they do not tamper with the secret x. That is, $< e, (l_1, l_2) >= (0, 0)$, and thus $< z + e, (l_1, l_2) >=< z, (l_1, l_2) >$.

Next, let $e \in \mathbb{F}_q^v \setminus \mathcal{D}$ be an error vector. That is, $(\beta_1, \beta_2) =< e, (l_1, l_2) >$ is a nonzero pair in $\mathbb{F}_q^2$. Let x be the original information symbol and z its corresponding tuple. Then, $z + e \in \mathcal{Z}$ if and only if the pair $(y_1, y_2) =< z + e, (l_1, l_2) >$ satisfies $y_2 = y_1^3$. Equivalently, if

$$x^3 + (x + \beta_1)^3 = \beta_2.$$

The function x^3 is an APN function; hence, the number of codewords in $\mathcal{Z}$ that mask this error is $2|\mathcal{D}|$. Consequently, the minimal arbitrary-error detection rate is $1 - 2/q$. □

Since any error in $\mathbb{F}_q^v \setminus \mathcal{D}$ is detected and the errors in $\mathcal{D}$ are never detected, $\mathcal{D}$ forms the detection kernel of $\mathcal{Z}$, i.e., $K_d = \mathcal{D}$.

4 Computing with representations of IPM-RED

In the following sections, we define two basic operations, multiplication and addition and prove that all the security orders are preserved. We follow the ideas presented in [7], i.e., we treat z as two IPM sharings of length $v - 1$, that is

$$z(1) = (z_1, z_3, z_4, \ldots, z_v)$$
$$z(2) = (z_2, z_3, z_4, \ldots, z_v).$$

We perform the computation with the two IPM sharings and then merge the outcome into one vector using the secure homogenization algorithm in Algorithm 3, lines 4–6 [7].

Let $z, z' \in \mathcal{Z}$ be two sharings such that

$$(x, x^3) =< z, (l_1, l_2) > \text{ and } (x', (x')^3) =< z', (l_1, l_2) >;$$

each share may have different mask elements (m and m'). A secure multiplication algorithm in the IPM-RED domain that computes the product sharing $p = \texttt{IPMREDMult}(z, z') \in \mathcal{Z}$, and a secure addition algorithm that computes the sharing $p = \texttt{IPMREDAdd}(z, z') \in \mathcal{Z}$ are given in Algorithm 1 and Algorithm 2, respectively. Table 1 compares the computational complexity of the IPM-RED scheme to the IPM-DF scheme.

Both algorithms involve concatenations of secure addition and multiplication operations in the (single information symbol) IPM domain as defined in [5]. Therefore, both preserve the word-level (nibble/byte) security order d_w and the bit-level security order d_b. Similarly, since the outcome of the algorithm is a legal codeword in $\mathcal{Z}$, the word-level and the bit-level error detection orders (d_e and d_f) are preserved as well. As shown next, the fifth security metric—the arbitrary-error detection rate —is preserved as well. This property allows end-to-end robustness against fault-injection attacks.

In what follows, we analyze the arbitrary-error detection rates $d_{a,\text{Mult},1}$ and $d_{a,\text{Add},1}$ at the output of `IPMREDMult` and `IPMREDAdd` in the presence of a single erroneous sharing (operand) and the arbitrary-error detection rates $d_{a,\text{Mult},2}$ and $d_{a,\text{Add},2}$ in the presence of two erroneous operands. Based on the bounds developed below, we have

Table 1 Complexity of the IPM-FD and IPM-RED schemes in terms of the number of elementary masking operations

Algorithm	`IPAdd`	`IPMult`	`Homogenization`
IPM-FD Add [7]	1	–	–
IPM-FD Mult. [7]	–	μ	$\mu - 1$
`IPMREDAdd`	4	4	1
`IPMREDMult`	–	2	1

Theorem 1 *The arbitrary-error detection rate of the overall IPM-RED scheme with respect to errors that are not in the kernel of $\mathcal{Z}$ is*

$$d_{a,\text{IPM-RED}} = \min\left(d_{a,\mathcal{Z}}, d_{a,\text{Mult},1}, d_{a,\text{Add},1}, d_{a,\text{Mult},2}, d_{a,\text{Add},2}\right) > 1 - 4/q.$$

4.1 Secure multiplication

Lemma 2 (*Arbitrary-error detection rate* of secure IPM-RED multiplication with a single erroneous sharing) *The arbitrary-error detection rate at the output of Algorithm 1 when one operand is erroneous is $d_{a,\text{Mult},1} \geq 1 - 2/q$.*

Proof First note that the homogenization algorithm in [7] involves linear operations between the sharing symbols. For two sharings, say $a, b \in \mathbb{F}_q^{v-1}$, it finds an equivalent third sharing c such that a and c share all coordinates except the first one and $< b, l_2 >=< c, l_2 >$. This makes it possible to merge the two shares a and c into a single share p of length v such that

$$\begin{aligned} p(1) &= a \\ p(2) &= c \\ < p, (l_1, l_2) > &= (< a, l_1 >, < b, l_2 >). \end{aligned}$$

Let $x, x' \in \mathbb{F}_q$ be two secrets represented by the sharings $z, z' \in \mathbb{F}_q^v$ such that

$$(x, x^3) =< z, (l_1, l_2) > \text{ and } (x', (x')^3) =< z', (l_1, l_2) > .$$

Without loss of generality, assume that an attacker tampers with the sharing z, that is $v = z + \hat{e}$ where $\hat{e}$ is not in the kernel of $\mathcal{Z}$. (Otherwise, it is not considered as a harmful error.) The injected error can be represented as $\hat{e} = e + \epsilon$ where $e = (e_1, e_2, \mathbf{0}_{v-2})$ is a nonzero vector and $\epsilon \in \mathcal{D}$. The sharing v satisfies

$$< v, (l_1, l_2) >= (x + e_1, x^3 + e_3) =< z + e, (l_1, l_2) > .$$

That is, all the errors in the coset $e + \mathcal{D}$ are detected with the same probability.

Denote by $\texttt{IPMREDMult}(v, z')$ the output of the multiplication algorithm. Algorithm 1 receives v and z' and computes $t = \texttt{IPMult}(v(1), z'(1))$ and $u = \texttt{IPMult}(v(2), z'(2))$ that satisfy

$$\begin{aligned} < t, l_1 > &= (x + e_1)x' \\ < u, l_2 > &= (x^3 + e_2)(x')^3, \end{aligned}$$

respectively. Therefore, after the homogenization step we have

$$< p, (l_1, l_2) >= ((x + e_1)x', (x^3 + e_2)(x')^3).$$

If $x' = 0$, then $< p, (l_1, l_2) >= (0, 0)$ and the multiplication algorithm has properly corrected the error since $\texttt{IPMREDMult}(v, z') = \texttt{IPMREDMult}(z, z')$. On the other hand, if $x' \neq 0$ an erroneous result may be produced, that is,

$$\texttt{IPMREDMult}(v, z') = \texttt{IPMREDMult}(z, z') + \tilde{e} \in \mathcal{Z}$$

where $\tilde{e} \in \mathbb{F}_q^v \setminus \mathcal{D}$. This happens if and only if $(x + e_1)^3 = x^3 + e_2$. In this case, $\tilde{e}$ will not be detected by the checker with probability $Q(\tilde{e}) = Q(e) \leq 2/q$. Consequently, the robustness of the IPM-RED scheme is preserved, $d_{a,\text{Mult},1} \geq 1 - 2/q$. □

Lemma 3 (*Arbitrary-error detection rate* of secure IPM-RED multiplication with two erroneous sharings) *The arbitrary-error detection rate at the output of Algorithm 1 when two operands are erroneous is $d_{a,\text{Mult},2} \geq 1 - 4/q$.*

Proof Let $x, x' \in \mathbb{F}_q$ be two secrets represented by the sharings $z, z' \in \mathbb{F}_q^v$ such that

$$(x, x^3) =< z, (l_1, l_2) > \text{ and } (x', (x')^3) =< z', (l_1, l_2) > .$$

Let $\hat{e}, \hat{e}' \in \mathbb{F}_q^v \setminus \mathcal{D}$ be two error vectors not in the kernel of $\mathcal{Z}$. That is, $\hat{e} = e + \epsilon$, $\hat{e}' = e' + \epsilon'$ where $e = (e_1, e_2, \mathbf{0}_{v-2})$, $e' = (e'_1, e'_2, \mathbf{0}_{v-2})$ are nonzero vectors and $\epsilon, \epsilon' \in \mathcal{D}$.

Define $v = z + \hat{e}$ and $v' = z' + \hat{e}'$ and denote by $\texttt{IPMREDMult}(v, v')$ the output of the multiplication algorithm. Algorithm 1 receives v and v' and computes $t = \texttt{IPMult}(v(1), v'(1))$ and $u = \texttt{IPMult}(v(2), v'(2))$ that satisfy

$$\begin{aligned} < t, l_1 > &= (x + e_1)(x' + e'_1) \\ < u, l_2 > &= (x^3 + e_2)((x')^3 + e'_2), \end{aligned}$$

respectively. Therefore, after the homogenization step we have

$$< p, (l_1, l_2) >= ((x + e_1)(x' + e'_1), (x^3 + e_2)((x')^3 + e'_2).$$

An error is undetected when

$$(x + e_1)^3(x' + e'_1)^3 = (x^3 + e_2)((x')^3 + e'_2). \tag{7}$$

In general, there are q^2 pairs of (x, x') which are equally likely to occur. For a given pair of error vectors e and e', some of the (x, x') pairs will detect its presence and some will mask it.

The number of pairs for which an erroneous product will not be detected depends on the relation between the first two components of the error vectors. That is, if $(e'_1)^3 = e'_2$, then

- For $x' = e'_1$, all the q pairs of the form $(x, x' = e'_1)$ cannot detect the error because the de-masking will produce the pair $(y_1, y_2) = (0, 0)$ which is a legal codeword in $\mathcal{C}_0$.
- For $x' = 0$ at most two pairs of the form $(x, x' = 0)$ are solutions to Eq. 7, and hence cannot detect the error.
- For any $x' \in \mathbb{F}_q \setminus \{e'_1, 0\}$, there are at most three x's that solve Eq. 7.

Overall, for $(e'_1)^3 = e'_2$ there are at most $(q + 2 + 3(q - 2))$ pairs of (x, x')s (and hence $4(q-1)|\mathcal{D}|^2$ pairs of sharings) for which the erroneous output of the secure multiplication algorithm will not be detected.

However, if $(e'_1)^3 \neq e'_2$, then $Q_{Mult,2}(e, e') \leq 3/q$; because, if the equation $(x' + e'_1)^3 = (x')^3 + e'_2$ has no solutions, then there are at most $3q$ pairs that satisfy Eq. 7. But if there exists an x' that satisfies $(x' + e'_1)^3 = (x')^3 + e'_2$, then Eq. 7 has $(2 \cdot 2 + 3 \cdot (q - 2))$ solutions.

Therefore, we have $d_{a,\text{Mult},2} > 1 - 4/q$. □

4.2 Secure addition

Lemma 4 (*Arbitrary-error detection rate* of secure IPM-RED addition with a single erroneous sharing) *The arbitrary-error detection rate at the output of Algorithm 2 when one operand is erroneous is* $d_{a,\text{Add},1} = 1 - 2/q$.

Proof The proof is similar to the proof of Lemma 3. Let $x, x' \in \mathbb{F}_q$ be two secrets represented by the sharings $z, z' \in \mathbb{F}_q^v$ such that $(x, x^3) =< z, (l_1, l_2) >$ and $(x', (x')^3) =< z', (l_1, l_2) >$. Assume that an attacker tampers with the sharing z, that is, $v = z + \hat{e}$ where $\hat{e} = e + \epsilon$, $e = (e_1, e_2, \mathbf{0}_{v-2})$ is a nonzero vector and $\epsilon \in \mathcal{D}$.

Algorithm 2 receives v and z' and computes $p1 = \texttt{IPAdd}(v(1), z'(1))$ and $p2 = \texttt{IPAdd}(v(2), z'(2))$ that satisfy

$$< p1, l_1 >= (x + e_1) + x' \text{ and } < p2, l_2 >= (x^3 + e_2) + (x')^3,$$

respectively. The value of t and u after lines 6 and 9 satisfy

$$< t, l_2 >= (x + e_1)^2 x' \text{ and } < u, l_2 >= (x + e_1)(x')^2,$$

respectively. Therefore, after the addition in line 10 we have,

$$< p2, l_2 >= ((x^3 + e_2) + (x')^3) + (x + e_1)^2 x' + (x + e_1)(x')^2$$

and after the homogenization we get

$$< p, (l_1, l_2 >= ((x + x' + e_1), \quad (x + x' + e_1)^3 + (x + e_1)^3 + (x^3 + e_2).$$

Consequently, the error $\hat{e}$ is not detected by the checker if and only if $(x + e_1)^3 + (x^3 + e_2) = 0$ and this occurs with probability $Q(\hat{e}) \leq 2/q$. Therefore, the arbitrary-error detection rate of an IPM-RED addition when a single sharing is erroneous is $d_{a,\text{Add},1} = 1 - 2/q$. □

Lemma 5 (*Arbitrary-error detection rate* of secure IPM-RED addition with two erroneous sharings)

The arbitrary-error detection rate at the output of Algorithm 2 is $d_{a,\text{Add},2} = 1 - 2/q$.

Proof Let $x, x' \in \mathbb{F}_q$ be two secrets represented by the sharings $z, z' \in \mathbb{F}_q^v$ such that $(x, x^3) =< z, (l_1, l_2) >$ and $(x', (x')^3) =< z', (l_1, l_2) >$. Let $\hat{e}, \hat{e}' \in \mathbb{F}_q^v \setminus \mathcal{D}$ be two error vectors not in the kernel of $\mathcal{Z}$. That is, $\hat{e} = e + \epsilon$, $\hat{e}' = e' + \epsilon'$ where $e = (e_1, e_2, \mathbf{0}_{v-2})$, $e' = (e'_1, e'_2, \mathbf{0}_{v-2})$ are nonzero vectors and $\epsilon, \epsilon' \in \mathcal{D}$.

Define $v = z + \hat{e}$ and $v' = z' + \hat{e}'$. Algorithm 2 receives v and v' and after the homogenization step it produces p that satisfies

$$< p, (l_1, l_2 >= ((x + e_1 + x' + e'_1), \quad (x^3 + e_2) + ((x')^3 + e'_2) + (x + e_1)^2(x' + e'_1) + (x + e_1)(x' + e'_1)^2.$$

Consequently, the errors $\hat{e}$ and $\hat{e}'$ are not detected if and only if the pair (x, x') solves the equation

$$(x + e_1)^3 + (x^3 + e_2) = (x' + e'_1)^3 + ((x')^3 + e'_2).$$

For a given value of x', this equation has at most two solutions; therefore, $d_{a,\text{Add},2} = 1 - 2q/q^2$. □

Algorithm 1 Secure Multiplication algorithm for IPM-RED (`IPMREDMult`)

1: Input: Two secrets $x, x' \in \mathbb{F}_q$ represented by $z, z' \in \mathbb{F}_q^v$ such that
$$(x, x^3) =< z, (l_1, l_2) >, \quad (x', (x')^3) =< z', (l_1, l_2) > .$$
2: Output: A sharing $p \in \mathbb{F}_q^v$ such that $(xx', (xx')^3) =< p, (l_1, l_2) >$.
3: t:= `IPMult(z(1),z'(1))` // `IPMult`algorithm from [5]
4: u:= `IPMult(z(2),z'(2))`
5: p := `Homogenization(t,u);` // Homogenization algorithm from [7]
6: **return** $p \in \mathcal{Z}$

Algorithm 2 Secure Addition algorithm for IPM-RED (`IPMREDAdd`)

1: Input: Two secrets $x, x' \in \mathbb{F}_q$ represented by $z, z' \in \mathbb{F}_q^v$ such that

$$(x, x^3) =< z, (l_1, l_2) >, \quad (x', (x')^3) =< z', (l_1, l_2) > .$$

2: Output: A sharing $p \in \mathbb{F}_q^v$ such that $(x + x', (x + x')^3) =< p, (l_1, l_2) >$.

3: p1:= `IPAdd`(z(1),z'(1)) // `IPAdd`algorithm from [5]
4: p2:= `IPAdd`(z(2),z(2)) // $< p(2), l_2 >= x^3 + (x')^3$
5: t:= `IPMult`(z(1),z(1)) // `IPMult`algorithm from [5]
6: t := `IPMult`(t ,z'(1))
7: p2:= `IPAdd`(p2,t) // $< p(2), l_2 >= x^3 + (x')^3 + x^2 \cdot x'$
8: u := `IPMult`(z'(1),z'(1)) // `IPMult`algorithm from [5]
9: u := `IPMult`(u ,z(1))
10: p2:= `IPAdd`(p2,u) // $< p(2), l_2 >= (x + x')^3$
11: p := `Homogenization`(p1,p2) // Homogenization algorithm from [7]
12: **return** $p \in \mathcal{Z}$

5 Information leakage from IPM-FD and IPM-RED checkers

In this section, we examine whether the use of error detecting codes with a deterministic encoder in IPM schemes can degrade security; that is, whether the checker's response can narrow the key's search space.

5.1 The checker's structure and operation

In principle, checkers for masking schemes that have error detecting capabilities can be implemented after each calculation in the masking domain. However, the hardware implementation of online checkers is costly in terms of area and power consumption. Therefore, a checker is typically placed after the state register, thus making it possible to detect faults in the combinational parts of the circuit as well as bit-flips in the register itself. In this paper, we assume that the state register is located at the output of the S-box layer (see Fig. 2). The input to the checker is a de-masked sharing word $\hat{c} =< z, (l_1, \ldots l_\mu) >$. In a fault-free circuit $\hat{c} = c \in \mathcal{C}_0$. For the IPM-FD scheme, $\mathcal{C}_0$ is a q-ary repetition code and for the IPM-RED scheme $\mathcal{C}_0$ is a nonlinear systematic robust code.

The checker examines the word it sees in its inputs. If it decides that the word is not a legal codeword or that it cannot be corrected into a legal one, it outputs the value $\perp$. Depending on the design, a $\perp$ may halt the processor or cause the system to replace the output by a random one. If the checker decides that the word is legal or that it is within the correction capability of the code, it decodes it into a legal codeword. The checker's output is denoted by v, $v \in \mathcal{C} \cup \{\perp\}$.

The checker is implemented in hardware on the same circuit as the logic it protects. Therefore, the checker itself can be subject to fault injection as well the cryptographic circuit. Usually, a checker is composed of two parts: an encoder and a comparator. An attack on the encoder logic is equivalent to an attack on the register and hence will be detected with the same probability as if an error was injected into the cryptographic circuit. Detection of an attack on the comparator requires a security aware implementation [32].

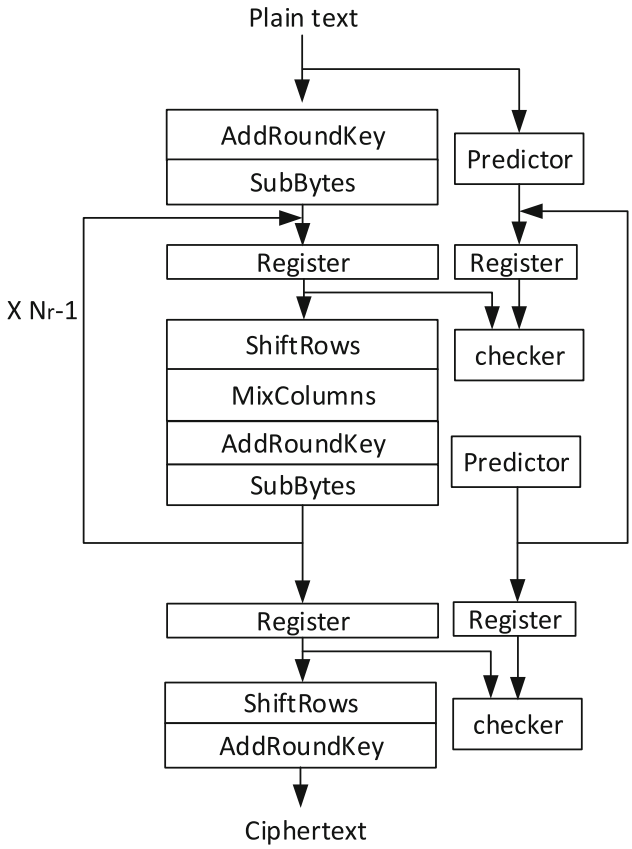

Fig. 2 Protected AES architecture. The registers are located after the SubBytes module to enable on-line prediction and checking

Note that the data processed by the last two blocks in Fig. 2 are not examined by a checker. In order to provide an end-to-end security, these outputs should be checked as well. It is worth noting that an effective attack on these blocks will aim to stick the bits and not flip them. Thus, codes detecting stuck-at faults would be more effective for this part of the circuit. This, however, is beyond of the scope of this paper. A detailed analysis of the security provided by unidirectional error detecting codes can be found in [33].

5.2 Attack model and illustrative example

To evaluate the *average* reduction in the size of the search space, we assume a hypothetical probing attack on all the checker's output bits. That is, we assume that the checker's output is *completely observable* to the attacker. In addition,

we assume that the attacker knows the code $\mathcal{C}_0$ and hence can know which errors will always be detected and which have a chance of passing unnoticed. Therefore, we consider the worst scenario, i.e., we assume that an error e of the latter type is injected.

In what follows, the information that leaks from the checker is evaluated in terms of the mutual information (MI) between two random variables: the secret key (or sub-key) K and the checker's output V. (Random variables are denoted by capital letters and the values they take are written in lowercase letters.) The MI is analyzed as a function of the injected error; it represents the (average) number of bits that can be learned about the key from a single injection of a specific error vector.

As was shown in [1], the information leakage from the checker is maximal when the attack is mounted on the first round. In general, fault analysis attacks are performed on the middle/last rounds because of the tradeoff between the number of required correct-fault pairs and the time-memory complexity [34,35]. Nevertheless, when error detecting codes are embedded in hardware, an attacker who knows the coding scheme and observes the checker's output can derive information about the key even when the attack is performed on the very first cycle. The following example illustrates how the checker's output may be used to reduce the search space of the key:

Example 2 For simplicity, consider a hypothetical 12-bit cipher implemented without masking[2]. The inputs to the circuit implementing that cipher are a 12-bit plaintext s and a 12-bit key y. Assume that in the first cycle this cipher computes

$$x = (x_1, x_2, \ldots x_{12}) = f(s+y) = \begin{cases} (s+y)^{-1} & s+y \neq 0 \\ 0 & s+y = 0 \end{cases}$$

where the computation is performed in $\mathbb{F}_{2^{12}}$ with the primitive polynomial $D^{12}+D^6+D^4+D+1$. Consider a hardware implementation protected by the Quadratic Sum (QS) code [6]. The block we refer to as the "original component" calculates $x(s, y)$, and in parallel, the predictor generates two redundant bits $w_1(s, y)$ and $w_2(s, y)$. When the two binary vectors x and w are treated as vectors over $\mathbb{F}_4$, they form a codeword $c = (\xi_1, \xi_2, \ldots \xi_6, \eta)$ with $k = 6$ information symbols and a single redundant symbol. In this example, we use the following (intuitive) mapping $\xi_i = (x_{2i-1}, x_{2i})$ and $\eta = (w_1, w_2)$. The codeword fulfills the following property:

$$\eta = \xi_1\xi_2 + \xi_3\xi_4 + \xi_5\xi_6. \tag{8}$$

[2] Since the checker sees the de-masked vector, we simplified the example by considering a plain implementation (without masking).

The multiplication and additions in Eq. 8 are over $\mathbb{F}_4$ with the primitive polynomial $D^2 + D + 1$. ($\mathbb{F}_4$ is isomorphic to $\mathbb{F}_2^2$). Note that the predictor generates the two redundant bits directly from s and y without explicitly computing the variable x, even though η and ξ fulfill Eq. 8. The codeword stored in the register is then $c = (x, w)$. Our attack model assumes that the adversary can flip an arbitrary number of bits in arbitrary locations of both x and w.

Assume that s = (00 01 11 11 01 11), y = (00 00 11 10 10 10) and that the attacker injects the error e = (01 00 00 00 00 10 11). Then, the codeword written into the register is

$$c = (\underbrace{11\ 00\ 11\ 01\ 10\ 00}_{x}, \underbrace{11}_{w}) = (\underbrace{3\ 0\ 3\ 1\ 2\ 0}_{\xi}, \underbrace{3}_{\eta})$$

and the distorted word $\hat{c} = c + e = (\hat{x}, \hat{w}) = (\hat{\xi}, \hat{\eta})$ read from the register is

$$\hat{c} = c + e = (\mathbf{1}0\ 00\ 11\ 01\ 10\ \mathbf{1}0, \mathbf{00}) = (\underbrace{\mathbf{2}0\,3\,,1\,2\,\mathbf{2}}_{\hat{\xi}}, \underbrace{\mathbf{0}}_{\hat{\eta}})$$

(the erroneous bits and symbols appear in bold). The checker computes $\hat{\xi}_1\hat{\xi}_2 + \hat{\xi}_3\hat{\xi}_4 + \hat{\xi}_5\hat{\xi}_6 = 0$ and since it is equal to $\hat{\eta}$, it does not report a decoding error. The attacker who knows the code sees that the attack has been masked (undetected). Therefore, the attacker infers that the following error-masking equation must be fulfilled:

$$(\xi_1 + 1)\xi_2 + \xi_3\xi_4 + \xi_5(\xi_6 + 2) = \eta + 3.$$

There are exactly 2^{10} values of x that satisfy this equation; they are of the form

$$\xi = (\xi_1,\ \xi_2 = 3 - 2\xi_5,\ \xi_3,\ \xi_4,\ \xi_5,\ \xi_6).$$

Since $y = (x^{-1} - s)$, y can take 2^{10} values. Therefore, the search space for the key y has been narrowed from 12 to 10 bits.

Note that if the value of the key had been such that the error would have been detected, the attacker could observe this outcome and narrow the search space for y from 2^{12} down to $(2^{12}-2^{10})$ possible values. However, the system would then receive an alarm and prevent the attacker from continuing to reduce the search space by further fault injections. For example, the system could initiate re-keying, in which case all the information that the attacker had acquired would be useless. Note that the error is not detected for only 1/4 of the keys for this error vector due to the properties of the QS code used [6]. It is detected (leading to an alarm) for the remaining 3/4 of the keys. □

5.3 Information leakage from IPM-FD and IPM-RED checkers

The following theorem is taken from [1] with minor modifications. It distinguishes between standard checkers that output a $\perp$ symbol when the word in their input is illegal, and infective checkers that output a random codeword upon detecting such a problem.

Theorem 2 (Information leakage from the checker on the first round) *Let $\mathcal{C}$ be a systematic code of length n_b protecting a key (or a sub-key) of k_b bits, that is $|\mathcal{C}| = 2^{k_b}$. Denote by $Q_{\mathcal{C}}(e)$ the probability that an error e is undetected by the codewords of $\mathcal{C}$, $0 \leq Q_{\mathcal{C}}(e) \leq 1$. The mutual information (in bits) between the secret key y and a standard checker output $v \in \mathbb{F}_2^{n_b}$ for a given plaintext $s \in \mathbb{F}_2^{k_b}$ and an injected error $e \in \mathbb{F}_2^{n_b}$ is*

$$I_{\text{standard}}(Y; V|s, e) = k_b Q_{\mathcal{C}}(e) - (1 - Q_{\mathcal{C}}(e)) \cdot \log_2(1 - Q_{\mathcal{C}}(e)). \tag{9}$$

The mutual information between y and an infective checker output v for a given plaintext s and an injected error e is

$$I_{\text{infective}}(Y; V|s, e) = k_b Q_{\mathcal{C}}(e) - Q_{\mathcal{C}}(e)(2 - Q_{\mathcal{C}}(e)) \log_2(2 - Q_{\mathcal{C}}(e)) - (1 - Q_{\mathcal{C}}(e))^2 \log_2(1 - Q_{\mathcal{C}}(e)). \tag{10}$$

Denote by $\bar{I}_{\text{standard}}$ and $\bar{I}_{\text{infective}}$ the maximal mutual information $I(Y; V|s, e)$ over all the non-zero errors e, plaintexts s and over all the rounds. From [1] and Theorem 2 we have

$$\bar{I}_{\text{standard}} \leq k_b \bar{Q}_{\mathcal{C}_0} - (1 - \bar{Q}_{\mathcal{C}_0}) \cdot \log_2(1 - \bar{Q}_{\mathcal{C}_0}),$$
$$\bar{I}_{\text{infective}} \leq k_b \bar{Q}_{\mathcal{C}_0} - \bar{Q}_{\mathcal{C}_0}(2 - \bar{Q}_{\mathcal{C}_0}) \log_2(2 - \bar{Q}_{\mathcal{C}_0}) - (1 - \bar{Q}_{\mathcal{C}_0})^2 \log_2(1 - \bar{Q}_{\mathcal{C}_0}).$$

Because of the linearity of the repetition code used in a IPM-FD scheme, a single fault-injection attack causes the checker to leak all the k_b secret bits. That is, for IPM-FD scheme we have

$$\bar{I}_{\text{infective}} = \bar{I}_{\text{standard}} = \begin{cases} 4 \text{ bits PRESENT} \\ 8 \text{ bits AES} \end{cases}$$

In contrast, a single fault-injection attack on a IPM-RED scheme causes (on average over a uniformly distributed key space) a leak of

$$\bar{I}_{\text{infective}} \leq \bar{I}_{\text{standard}} = \begin{cases} 0.6686 \text{ bits PRESENT} \\ 0.0737 \text{ bits AES} \end{cases}$$

That is, when robust codes are used, the entropy loss due to the information leak from the checker in a IPM-RED scheme is small, whereas the probability that a single attack will not be detected decreases exponentially with s ($\bar{Q}_{\mathcal{C}_0} = q^{-s+1}$).

The same relationship holds for the dependency of the entropy loss and the probability of an undetected attack on the *number of fault injections*. In other words, we can expect that providing strong fault detection only has a limited and exponentially decreasing impact on information leakage and thus on security.

6 Conclusions and future work

We presented a solution that combines sophisticated and highly effective countermeasures against side-channel and fault-injection attacks in a provably composable manner. The resulting scheme IPM-RED offers protection against high-capability adversaries. We formalized relevant security properties by appropriate orders and proved that the protections against side-channel analysis do not compromise security against fault attacks, and vice versa. We introduced basic arithmetic operations in the new IPM-RED representation and studied their security and complexity.

A number of interesting theoretical and experimental questions are still open. The relationship between the security order of addition and multiplication algorithms and their realization (e.g., the order in which the operations are executed) could be better understood. Errors with maximal masking probability could be identified and studied. Also of interest is the strength of the IPM-RED scheme when the adversary only injects errors from a known subset, e.g., errors that are actually useful for cryptanalysis of a given cipher. A related question is whether the scheme can be further improved by incorporating more advanced security-oriented codes, such as the *compact protection code* [36].

Useful experimental knowledge would include the resistance of actual cryptographic primitives protected by IPM, IPM-FD and IPM-RED to side-channel attacks. Such investigations could focus on determining the success rate of actual attacks (e.g., correlation power attacks) or evaluating information leakage by means of mutual information [37]. An interesting question is the implementation cost; here, finding an efficient hardware realization of the basic operations in the masked domain is essential. This question is tightly bound to the practically achievable security order and also to robustness against fault attacks.

Acknowledgements Open Access funding provided by Projekt DEAL.

References

1. Keren, O., Polian, I.: A comment on information leakage from robust code-based checkers detecting fault attacks on cryptographic primitives. In: Heydemann, K., Kühne, U., Li, L. (eds.) Proceedings of 8th Workshop on Security Proofs for Embedded Systems, vol. 11, pp. 49–63. Kalpa Publications in Computing, EasyChair (2019)
2. Rostami, M., Koushanfar, F., Karri, R.: A primer on hardware security: models, methods, and metrics. Proc. IEEE **102**(8), 1283–1295 (2014)
3. Regazzoni, F., Breveglieri, L., Ienne, P., Koren, I.: Interaction between fault attack countermeasures and the resistance against power analysis attacks. In: Fault Analysis in Cryptography, (2012)
4. Schneider, T., Moradi, A., Güneysu, T.: ParTI - towards combined hardware countermeasures against side-channel and fault-injection attacks. In: Robshaw, M., Katz, J. (eds.), Advances in Cryptology—CRYPTO 2016. Annual International Cryptology Conference (CRYPTO-2016), August 14–18, Santa Barbara, CA, United States , Lecture Notes in Computer Science (LNCS), vol. 9815, pp. 302–332, Springer, (2016)
5. Balasch, J., Faust, S., Gierlichs, B.: Inner product masking revisited. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology–EUROCRYPT, pp. 486–510. Springer, Berlin (2015)
6. Karpovsky, M.G., Kulikowski, K.J., Wang, Z.: Robust error detection in communication and computational channels. In: Spectral Methods and Multirate Signal Processing. SMMSP'2007. 2007 International Workshop on, Citeseer, (2007)
7. Cheng, W., Carlet, C., Goli, K., Danger, J.-L., Guilley, S.: Detecting faults in inner-product masking scheme–IPM-FD: IPM with fault detection. In: Heydemann, K., Kühne, U., Li, L. (eds.) proceddings of 8th International Workshop on Security Proofs for Embedded Systems, vol. 11, pp. 17–32. Kalpa Publications in Computing, New York (2019)
8. De Meyer, L., Arribas, V., Nikova, S., Nikov, V., Rijmen, V.: M&M: Masks and macs against physical attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(1), 25–50 (2019)
9. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of eliminating errors in cryptographic computations. J. Cryptol. **14**(2), 101–119 (2001)
10. Barenghi, A., Breveglieri, L., Koren, I., Naccache, D.: Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. Proc. IEEE **100**(11), 3056–3076 (2012)
11. Polian, I., Regazzoni, F.: Counteracting malicious faults in cryptographic circuits. In: ETS, pp. 1–10, IEEE, (2017)
12. Tunstall, M., Mukhopadhyay, D., Ali, S.: Differential fault analysis of the advanced encryption standard using a single fault. In: WISTP, vol. 6633 of Lecture Notes in Computer Science, pp. 224–233, Springer, (2011)
13. Jovanovic, P., Kreuzer, M., Polian, I.: A fault attack on the LED block cipher. In: COSADE, vol. 7275 of Lecture Notes in Computer Science, pp. 120–134, Springer, (2012)
14. Tajik, S., Lohrke, H., Ganji, F., Seifert, J., Boit, C.: Laser fault attack on physically unclonable functions. In: 2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 85–96, (2015)
15. Li, Y., Sakiyama, K., Gomisawa, S., Fukunaga, T., Takahashi, J., Ohta, K.: Fault sensitivity analysis. In: CHES, vol. 6225 of Lecture Notes in Computer Science, pp. 320–334, Springer, (2010)
16. Ghalaty, N.F., Yuce, B., Taha, M.M.I., Schaumont, P.: Differential fault intensity analysis. In: FDTC, pp. 49–58, IEEE Computer Society, (2014)
17. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Springer, Berlin (2007)
18. Kocher, P., Jaffe, J., Jun, B., Rohatgi, P.: Introduction to differential power analysis. J. Cryptogr. Eng. **1**(1), 5–27 (2011)
19. Doget, J., Prouff, E., Rivain, M., Standaert, F.: Univariate side channel attacks and leakage modeling. J. Cryptogr. Eng. **1**(2), 123–144 (2011)
20. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F., Veyrat-Charvillon, N.: Mutual information analysis: a comprehensive study. J. Cryptol. **24**(2), 269–291 (2011)
21. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology–CRYPTO, pp. 463–481. Springer, Berlin (2003)
22. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010, pp. 413–427. Springer, Berlin (2010)
23. Goubin, L., Martinelli, A.: Protecting AES with shamir's secret sharing scheme. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems–CHES, pp. 79–94. Springer, Berlin (2011)
24. Roche, T., Prouff, E.: Higher-order glitch free implementation of the AES using secure multi-party computation protocols. J. Cryptogr. Eng. **2**, 111–127 (2012)
25. Mangard, S., Pramstaller, N., Oswald, E.: Successfully attacking masked AES hardware implementations. In: Rao, J.R., Sunar, B. (eds.) Cryptographic Hardware and Embedded Systems–CHES, pp. 157–171. Springer, Berlin (2005)
26. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: Ning, P., Qing, S., Li, N. (eds.) Information and Communications Security, pp. 529–545. Springer, Berlin (2006)
27. Ngo, X.T., Bhasin, S., Danger, J., Guilley, S., Najm, Z.: Linear complementary dual code improvement to strengthen encoded circuit against hardware trojan horses. In: IEEE International Symposium on Hardware Oriented Security and Trust, HOST, Washington, DC, 5–7 May, 2015, pp. 82–87, (2015)
28. Bringer, J., Carlet, C., Chabanne, H., Guilley, S., Maghrebi, H.: Orthogonal direct sum masking. In: Naccache, D., Sauveron, D. (eds.) Information Security Theory and Practice, pp. 40–56. Securing the Internet of Things, Springer, Berlin (2014)
29. Berger, T.P., Canteaut, A., Charpin, P., Laigle-Chapuy, Y.: On almost perfect nonlinear functions over $\mathbb{F}_2^n$. IEEE Trans. Inf. Theory **52**(9), 4160–4170 (2006)
30. Carlet, C., Ding, C.: Highly nonlinear mappings. J. Complex. **20**(2–3), 205–244 (2004)
31. Engelberg, S., Keren, O.: A comment on the Karpovsky-Taubin code. IEEE Trans. Inf. Theory **57**(12), 8007–8010 (2011)
32. Busaba, F., Lala, P.K., Walker, A.: On self-checking design of CMOS circuits for multiple faults. VLSI Des. **1998**(2), 151–161 (1998)
33. Burg, A., Keren, O.: On the efficiency of berger codes against error injection attacks on parallel asynchronous communication channels. Inf. Secur. J. A Glob. Perspect. **22**(5–6), 208–215 (2013)
34. Rivain, M.: Differential fault analysis on DES middle rounds. In: CHES, (2009)

35. Derbez, P., Fouque, P.-A., Leresteux, D.: Meet-in-the-middle and impossible differential fault analysis on AES. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 274–291, (2011)
36. Rabii, H., Neumeier, Y., Keren, O.: High rate robust codes with low implementation complexity. IEEE Transactions on Dependable and Secure Computing, https://doi.org/10.1109/TDSC.2018.2816638, (2018)
37. de Chérisey, E., Guilley, S., Rioul, O., Piantanida, P.: Best information is most successful mutual information and success rate in side-channel analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 49–79 (2019)