Prime Masking vs. Faults - Exponential Security Amplification against Selected Classes of Attacks

Thorben Moos[1], Sayandeep Saha[1,2] and François-Xavier Standaert[1]

[1] Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium
firstname.lastname@uclouvain.be
[2] Indian Institute of Technology Bombay, Bombay, India
sayandeepsaha@cse.iitb.ac.in

Abstract. Fault injection attacks are a serious concern for cryptographic hardware. Adversaries may extract sensitive information from the faulty output that is produced by a cryptographic circuit after actively disturbing its computation. Alternatively, the information whether an output would have been faulty, even if it is withheld from being released, may be exploited. The former class of attacks, which requires the collection of faulty outputs, such as Differential Fault Analysis (DFA), then either exploits some knowledge about the position of the injected fault or about its value. The latter class of attacks, which can be applied without ever obtaining faulty outputs, such as Statistical Ineffective Fault Attacks (SIFA), then either exploits a dependency between the effectiveness of the fault injection and the value to be faulted (e.g., an LSB stuck-at-0 only affecting odd numbers), denoted as SIFA-1, or a conditional propagation of a faulted value based on a sensitive intermediate (e.g., multiplication of a faulted value by 0 prevents propagation), denoted as SIFA-2. The aptitude of additive masking schemes, which were designed to prevent side-channel analysis, to also thwart fault attacks is typically assumed to be limited. Common fault models, such as toggle/bit-flip, stuck-at-0 or stuck-at-1 survive the recombination of Boolean shares well enough for generic attacks to succeed. More precisely, injecting a fault into one or multiple Boolean shares often results in the same, or at least a predictable, error appearing in the sensitive variable after recombination. In this work, we show that additive masking in prime-order fields breaks such relationships, causing frequently exploited biases to decrease exponentially in the number of shares. As a result, prime masking offers surprisingly strong protection against generic statistical attacks, which require a dependency between the effectiveness of an injected fault and the secret variable that is manipulated, such as SIFA-1. Operation-dependent statistical attacks, such as SIFA-2 and Fault Template Attacks (FTA), may still be performed against certain prime-field structures, even if they are masked with many shares. Yet, we analyze the corresponding cases and are able to provide specific guidelines on how to avoid vulnerabilities either at the cipher design or implementation level by making informed decisions about the primes, non-linear mappings and masked gadgets used. Since prime-field masking appears to be one of the rare instances of affordable countermeasures that naturally provide sound protection against side-channel analysis and certain fault injection attacks, we believe there is a strong incentive for developing new ciphers to leverage these advantages.

Keywords: prime-field masking, fault injection attacks, SIFA, security amplification

1 Introduction

Implementations of cryptographic algorithms often face security threats beyond mathematical cryptanalysis. Passive adversaries may observe execution characteristics of a

computing device, such as their dissipation or emanation, in order to collect unintentionally leaked information. Active adversaries may even deliberately disturb cryptographic operations, changing the processed data or execution flow in order to force the revelation of sensitive values. The former class of attacks was first discovered in the late 1990s, initially exploiting the timing [Koc96] and instantaneous power consumption [KJJ99] of cryptographic implementations and is typically summarized by the term Side-Channel Analysis (SCA). Instances of the latter security threat are commonly referred to as Fault Injection Attacks (FIAs) and have been discovered almost simultaneously (around 1997), targeting public-key [BDL97] and secret-key [BS97] algorithms respectively. For the past three decades, studying vulnerabilities and protections with respect to SCA and FIA has blossomed into a major research field in cryptography. However, due to the irrefutable differences between the two adversarial strategies, the development of countermeasures against these threats, especially those striving for provable security in a formal model, is often conducted separately. Techniques based on secret sharing, frequently called *masking* in this context, were early on identified as a mathematically sound strategy to thwart SCA attacks [CJRR99, GP99, CG00]. In contrast, algorithmic protections against faulting adversaries mostly employ some form of *computational redundancy* (e.g., in space, time or information) to detect or correct maliciously inserted errors [BCN+06]. Since neither masking nor redundancy alone are able (or at least sufficient) to protect against both attack vectors, they are often seen as orthogonal to each other. Indeed, they often have to be combined in order to provide resistance against both threats collectively. Protection against attackers able to conduct passive *and* active methods *simultaneously* requires even more expensive dedicated solutions [FGM+23, BEF+23].

Nowadays, masking is the uncontested leader among algorithmic approaches to protect cryptographic implementations against side-channel adversaries. Different manifestations of the countermeasure have been explored in academic literature. Based on the selected encoding of the secrets and the operations performed on such encoded data, masking schemes can imply different overheads but also exhibit different security properties. The most common form of data encoding is based on an additive recombination function over binary extension fields, frequently referred to as *Boolean masking*. In Boolean masking, d individual uniformly distributed shares $s_i \in \mathbb{F}_{2^n}$ with $0 \leq i < d$ of a variable s are recombined as $s = \bigoplus_{i=0}^{d-1} s_i$. Due to the simple bitwise XOR ($\oplus$) operation used to perform addition in such fields, computation on the encoded data is remarkably efficient (especially for linear operations). Furthermore, the majority of symmetric cryptographic algorithms that are widely deployed today, including the AES, operate natively over binary field elements. Modern symmetric primitives, especially those envisioned for lightweight contexts, tend to consider the efficient application of masking even from the very beginning of the design process, influencing crucial decisions on the non-linear operations that may or may not be used. This is particularly evident in the recently concluded NIST Lightweight Cryptography (LWC) standardization process that culminated in the selection of the Ascon family in February 2023 [oSN17]. Ascon uses a masking-friendly low-degree S-box to specifically facilitate the realization of efficient masked implementations [DEMS21]. However, despite the strong efficiency of Boolean masking and its emergence as a design criterion in modern symmetric (lightweight) cryptography, this type of data encoding also comes with a number of disadvantages with respect to its physical security. First of all, it is well-known that Boolean masking requires a minimum noise level to be effective against SCA adversaries in the sense that it provides exponential security amplification in the number of shares (i.e., adversaries requiring an exponentially increasing number of observations to succeed) only when sufficient noise is present. Consequently, it has been demonstrated that a lack of noise can lead to devastating attacks in practice, even on implementations that are masked with many shares [BCPZ16, BS21]. The second observation that can be perceived as a disadvantage of the specific data encoding in

Boolean masking is caused by the same underlying characteristic, namely the fact that each individual bit of the secret encoded state can be restored from a single bit of each of its shares via a summation modulo two. This property not only causes the noise issue explained above but also leads to suboptimal protection against fault attacks, both differential and statistical ones [BH08, DEK+18]. For Differential Fault Attacks (DFA), this is obvious, as flipping a bit in a single share leads to the same bit being flipped in the encoded variable after recombination, requiring no change in the adversarial procedure when attacking such an implementation compared to an unprotected one. The situation is slightly more complex with respect to Statistical Ineffective Fault Attacks (SIFA). Any useful masking scheme is able to force such adversaries to bias all d shares simultaneously to perform a successful attack[1], which is often said to become more difficult with increasing d (although previous works have demonstrated the practical feasibility [SM20]). However, for Boolean masking, it is clear that adversaries who may reliably inject targeted faults into multiple shares (for simplicity, consider a stuck-at-0 in the Least Significant Bit (LSB) of each share) can completely bypass the countermeasure without any further performance penalty. This observation begs the question whether the problem is related to masking in general or whether other encoding schemes may provide more favorable features.

At TCC 2016, a number of interesting properties of masking in groups of prime order have been formally established, most notably the ability to amplify arbitrarily low noise levels with respect to SCA [DFS16]. In other words, prime masking is able to amplify any arbitrarily small adversarial uncertainty about the currently processed values exponentially in the number of shares. Therefore, encoded data in groups of prime order is not vulnerable to the same kind of low-noise side-channel attacks that Boolean encoded information is, and the complexity of attacks increases exponentially in the number of shares in all cases, except if the adversary already knows the entire values of shares with certainty (in that case no form of masking can provide security). While this is a very compelling result, it was unclear at the time whether such an encoding could be applied efficiently to relevant cryptographic operations or whether performance penalties would offset the security gains in the presence of realistic leakage functions. Two recent works from Eurocrypt 2023 and CHES 2023 attempted to answer these questions and found, based on information theoretic analyses and real-world experiments in software and in hardware, that *additive* masking over *prime-order fields* indeed has a great potential to improve the security vs. efficiency tradeoff of protected cryptographic implementations [MMMS23, CMM+23]. In additive prime-field masking, d individual uniformly distributed shares $s_i \in \mathbb{F}_p$ with $0 \leq i < d$ of a variable s are recombined as $s = \sum_{i=0}^{d-1} s_i \bmod p$. Our work falls in line with those prior investigations on the promising physical security properties that additive prime-field masking provides, in order to enable the design of future cryptographic algorithms inherently facilitating the application of effective implementation-level countermeasures. However, instead of focusing on side-channel attacks, we investigate its resistance to faulting adversaries here.

1.1 Our Contribution

In this work, we present the results of an in-depth study of the properties of additive prime-field masking with respect to fault attacks. We describe a number of positive traits that have not been reported in the literature yet and explain their mathematical background. We derive closed-form expressions to predict the likelihood of ineffective faults under multiple adversary models and estimate the number of outputs needed for key recovery. The mathematical analysis is substantiated by a large quantity of simulation results obtained through several tenths of thousands of CPU hours computation, performing

[1] Against most targets, this can be achieved with a single 1-bit fault in a single share exploiting non-bijectivity or structural features of masked non-linear operations (we give details in the paper).

key recovery attacks on toy ciphers and circuits instantiated with different properties, including variable-sized primes and hence fields $\mathbb{F}_p$. We have chosen to keep the majority of our investigations and results independent of any concrete cryptographic primitives in order to not limit the analysis to a specific setting but rather enable us to contrast certain basic constructions against each other. This allows us to provide guidelines on how to design future prime-field ciphers with inherent protection against faulting adversaries. However, to better illustrate the practical impact of our findings we also compare the complexity of certain simulated attacks on AES and AES-prime (an AES-like toy cipher proposed as a study object for prime-field masking research in [MMMS23]) at the end of this work.

Following the state-of-the-art understanding that redundancy is essentially inevitable to provide generic protection against fault attacks, we assume in our study that a simple detection-based countermeasure is already present (i.e., checking the unmasked ciphertexts from different redundant executions before releasing or discarding the result). This can be realized through simple time or area redundancy followed by an equality check (i.e., detection is needed only once at the very end of the computation). Without this type of baseline countermeasure, the susceptibility to structural fault attacks exploiting the specifics of the target cipher will typically be the dominating weakness. For example, the infamous Differential Fault Analysis (DFA) on the AES by Piret and Quisquater [PQ03] applies regardless of how the data is encoded as long as faulty outputs reach the adversary since the attack requires only random-value faults. In particular, the same attack applies equally to the AES-prime as it shares the same structural elements responsible for the weakness, regardless of any protection order.

Yet, ever since the introduction of SIFA at CHES 2018 [DEK+18], it is known that even perfect detection of all faults is insufficient to prevent key extraction. For these attacks to apply, the adversary only needs a dependency between the effectiveness of the fault and the value to be faulted (SIFA-1), for example, an LSB stuck-at-0 fault only affecting the binary representation of odd numbers. Alternatively, the adversary may exploit if the propagation of a faulted value through an operation is conditioned on a sensitive intermediate (SIFA-2, FTA). Consider, for example, a multiplication operation $a \cdot b$, which prevents propagation of faults injected in a if $b = 0$ and vice versa.

Our main result in this respect is that prime-field masking naturally provides strong protection against the former, operation-independent, type of SIFA directly performed on the encoded data. Indeed, the dependency between the secret masked variable and the effectiveness of any fault injected into its prime-field encoding decreases exponentially in the number of shares d. This holds true even in the presence of unrealistically strong adversaries who may inject almost arbitrary numbers of stuck-at-0, stuck-at-1 or toggle/bit-flip faults at the same time with arbitrary precision. The only mild limitation we have to place on the adversary to prevent trivial generic attacks independent of the masking scheme is that she cannot fault *all* the bits in the shares at the same time with perfect precision (this will be explained later in more detail). However, faulting all-but-one bit in each share with arbitrary precision is for example still within the model. To the best of our knowledge, this is the first report of a fault countermeasure providing exponential security amplification against a large and important class of fault attacks under such a strong adversary model.

Operation-dependent SIFA-2 or FTA may still be performed against certain structures and primitives independent of the masking order, including common non-linear operations used in prime-field masking, such as multiplications, squarings (non-linear in $\mathbb{F}_p$) and combinations thereof. Yet, we show that the complexity of such attacks (i.e., the total number of fault attempts performed) becomes prohibitive for larger primes p. Further, if instead of masking non-bijective field multiplications or squarings individually, the resulting bijective functions are turned into masked gadgets directly (e.g., $x^5 + 2 \bmod 2^7 - 1$ as

used in the AES-prime [MMMS23]), the described problems can be avoided to a large extent[2]. Finally, even attacks on simple duplication-based fault-detection countermeasures are naturally harder if prime-field masking is applied.

It is important to keep in mind that this collective level of protection is provided essentially for free, considering that effective masking is anyway needed to defend against side-channel adversaries, while detection-based countermeasures are required to prevent structural DFA, as detailed above. Taking into account the cost of alternative dedicated SIFA protections (see the related work in Section 5.1), this appears to be a great deal for designers of secure implementations. In summary, due to the strong protection that prime-field masking provides against wide classes of statistical fault attacks, we believe there is a strong incentive for developing new cryptographic primitives dedicated to leveraging these advantages most efficiently.

2 Background

In this section, we introduce the necessary background on different types of fault injection attacks, discuss formal security models as well as provable countermeasures, and finally introduce prime-field masking.

2.1 Differential Fault Analysis

The earliest discovered and still most common type of FIA on symmetric cryptographic algorithms is DFA [BS97]. Essentially, the attack relies on faulting data or an operation inside of a cryptographic implementation and then exploiting the differentials that can be observed between correct and faulty ciphertexts at the output. The most simple example is flipping a bit in the S-box input during the last round of a cipher computation and recording the faulty ciphertext. Together with a correct ciphertext for the same input, an adversary may now propagate the difference between faulty and correct ciphertext backwards to the injection point by guessing a part of the key. If the guessed key leads to a 1-bit difference at the location of the initially flipped bit, it is a candidate for the correct key, if not it can be discarded. A few pairs are usually sufficient to isolate the correct key candidate uniquely. Many different variants of DFA attacks tailored to specific target ciphers and implementations have been proposed in the literature. The most powerful ones exploit the structural properties of the targeted ciphers and require only very relaxed fault injection capabilities [PQ03]. This class of attacks can typically be prevented well by employing detection-based countermeasures, as the primary condition is to have access to faulty ciphertexts.

2.2 Statistical Ineffective Fault Attacks & Fault Template Attacks

Statistical Ineffective Fault Attacks (SIFA) and Fault Template Attacks (FTA) have been proposed as effective methods to overcome detection-based fault countermeasures at CHES 2018 and Eurocrypt 2020 respectively [DEK+18, SBR+20]. Both exploit the statistically biased distribution of the internal state of a cipher caused by a fault injection when conditioned on the ineffectiveness of the fault. In simple words, a fault is typically only effective for certain values of the internal state. Consider as an example the insertion of a stuck-at-0 fault in the LSB of a word, which is only effective if the numerical value expressed by the binary representation is odd, as for even numbers the LSB is already 0. This is called a value-dependent fault. If an adversary obtains only the results of executions that led to a fault-free computation (assuming that all faulty ones are filtered out by the

[2] Alternatively, known approaches to keep faults effective while passing through non-linear operations can be applied too [DDE+20, DOT24], yet those come at the cost of additional overheads.

detection-based countermeasure), it is clear that among these executions, the internal state at the faulted location collectively deviates significantly from a uniform distribution (since no odd values are part of the collection). The adversary may thus calculate backwards through the cipher to the fault location by guessing parts of the unknown key and checking the uniformity of the distribution. Since a statistical bias (i.e. deviation from the uniform distribution) will only be observed for the correct key guess, the adversary has obtained a distinguisher for the correct key. According to the classification introduced in [SJR+20] this constitutes a SIFA-1. Using a similar procedure, it is possible to exploit the propagation of faults through non-linear operations. Consider, as an example, a fault injected into one of the inputs to an AND gate or a larger field multiplication. This fault propagates through the gate and potentially to the output of the cipher only if the second input is non-zero. This is called a value-dependent fault *propagation* and can similarly be used to obtain a distinguisher for the correct key. Such an attack is commonly referred to as SIFA-2. FTA combines SIFA-1 and SIFA-2 attacks with the ability of the adversary to build templates and allow even attacks in the middle rounds of a cipher without any access to plain- or ciphertexts (correct or faulty). Adversaries only require the ability to trigger repeated encryptions of the same unknown plaintext several times [SBR+20].

2.3 Statistical Fault Analysis

It is worth mentioning that a predecessor of SIFA exists, which is called Statistical Fault Analysis (SFA) [FJLT13]. This attack utilizes a similar statistical bias but is based on the collection of faulty ciphertexts. Such attacks have one advantage over DFA – they do not require to obtain the correct ciphertexts corresponding to the faulty ciphertexts. SFAs can typically be prevented using the same detection-based countermeasures as DFA.

2.4 Formal Adversary Models for SCA and FIA

While early approaches in the development of effective countermeasures against both SCA and FIA focused on repairing the observed issues and vulnerabilities in an *ad hoc* manner, the research community slowly but surely realized that a more formal treatment of attacks and protections is required to comprehensively and confidently argue about the security of implementations. This includes precisely defining the capabilities of adversaries to be able to argue which defences are effective against them. Formal models for side-channel adversaries typically grant *read* access to a limited number of internal wires or values (called probes), either with or without noise affecting the accuracy of the observations. Such abstractions include the probing model [ISW03], the noisy leakage model [PR13], the random probing model [DDF14], the bounded moment model [BDF+17] and the robust probing model [FGP+18]. The latter considers physical defaults such as glitches or transitions in its abstraction as their negligence proved to be fatal for the mapping of secure implementations from theory to practice [NRR06, NRS08]. Formal models for fault injection adversaries typically grant *write* access to a limited number of internal wires or values (called faults), either with or without taking the inaccuracy of the insertions into account. Such abstractions include the standard threshold fault model [IPSW06], the consolidated fault adversary model of [RSG23] and the random fault model [DN23]. However, the formal aspects of fault adversary models are not as well-settled as for SCA yet. In fact, the concrete powers that should be given to hypothetical faulting adversaries are still hotly debated and, of course, highly depend on injection method and targeted implementation. Often, current models are perceived as too conservative with respect to the adversary's precision, as accurately faulting multiple individual bits in close proximity at the same time with high repeatability is hardly feasible in practice, but too restrictive on the number of faults that may be injected simultaneously, as affecting many bits at once without immaculate precision is typically straightforward in practice. For the latter

reason, we don't employ models in this work which limit the number of faults injected to a minor portion of the bits in an encoding, but instead allow adversaries to manipulate an almost arbitrary number of bits with perfect precision or an arbitrary number of bits with almost perfect precision (hence, being conservative in both aspects, the number and the precision of fault injections per computation).

2.5 Provably Secure Countermeasures against SCA

Countermeasures that deliver provable security against passive side-channel attacks considering the above-mentioned adversary models are commonly based on secret sharing techniques [Sha79a], also called masking schemes in this context [CJRR99, GP99, CG00]. Their main principle is based on splitting all sensitive variables inside a cryptographic algorithm into multiple so-called shares in such a way that an adversary needs to acquire information about all of the shares individually in order to recombine that knowledge into information about the underlying variable. Usually, the entire cryptographic algorithm is then executed on a randomly shared representation of the data, freshly and uniformly chosen for each execution, to protect all intermediate values against adversarial access. The security of the countermeasure is intuitively based on the assumption that passive non-invasive adversaries may only make noisy or imprecise observations of the data processed inside of an integrated circuit, for example, by observing its power consumption or electromagnetic radiation. This is modeled formally by either limiting the number of available probes or by directly considering an inaccurate probing process (depending on the chosen formal model, see above). Combining noisy or imprecise observations of all shares into knowledge about the underlying secrets is a process that yields exponentially decreasing quality of results in the number of shares. This is called security amplification and allows the use of the number of shares as an effective security parameter to scale an implementation's resistance. In general, masking is said to deliver resistance that grows exponentially in the security parameter while requiring quadratic resources, leading to an appealing cost effectiveness [DFS19].

2.6 Provably Secure Countermeasures against FIA

Protection against active attackers, which try to disturb the computation of cryptographic algorithms to learn information from the faulty responses or the resulting device behavior, is commonly achieved through computational redundancy in space, time or information [BCN+06]. Employing redundancy and comparing the output of multiple iterations or copies typically allows the detection of errors, as well as the correction of a bounded number of maliciously introduced faults. Existing solutions based on code-based detection [AMR+20], correction [SRM20] or hybrid [RSM21] schemes have attempted to reduce the cost of such countermeasures by decreasing the size of the redundancy while still covering the most relevant faults. Yet, it has been shown at CHES 2022 that using laser fault injection, it can become trivial in practice to escape these limitations by simply injecting more errors than covered by the underlying adversary model [BBM+22]. Hence, the authors suggest to rely on full redundancy instead and refused the notion that injecting more faults without requiring utmost precision is also more difficult. Another direction in the research of countermeasures is based on infection [GST12, PCM17, MAN+19]. Such techniques, upon detection of a fault, randomize the computation and generate a faulty but non-informative ciphertext. While the response to a detected fault is different for these countermeasures than that of simple detection techniques, we note that some form of redundancy is still required to identify the presence of faults. Therefore, the issues associated with redundancy still apply. Moreover, the construction of the infection mechanism is often complex, cipher-specific, and not amenable to any parametrization which may quantify the security. As already mentioned above, detection of faults alone is conceptually

insufficient, even if infection is used, since adversaries may even exploit ineffective faults for key recovery using statistical analyses [DEK+18, SBR+20]. Thus, generally, it is still unclear how to achieve a cost-effectiveness in countermeasure design against FIA that is similar to the situation of masking against SCA. In particular, current solutions lack a scalable parameter that can be used to increase the complexity of attacks much faster than the cost of the implementation.

2.7 Prime-Field Masking

Most common masking schemes provide their strong security guarantees only under two assumptions. First, the leakages about individual shares are sufficiently independent. This initially caused problems on certain implementation platforms due to physical defaults such as glitches and transitions re-combining multiple shares, but is now well-understood and considered in formalizations such as the robust probing model [FGP+18]. Secondly, they assume that the leakages obtained by an adversary are sufficiently noisy. Whether a concrete noise level is sufficient for security amplification or not depends in part on the concrete encoding scheme that is used [DFS16]. Luckily, it has been proven that masking schemes defined over prime-order fields, in contrast to most traditional methods based on Boolean or arithmetic masking in binary fields, may provide security amplification even for arbitrarily low noise levels [DFS16]. This holds true for any non-injective leakage function (for injective ones, any masking is ineffective). It has been shown that, if combined with applicable cryptographic algorithms, additive prime-field masking can provide (sometimes vastly) superior security levels in practice against low-noise SCA attacks while still leading to efficient hard- and software implementations [MMMS23, CMM+23]. The relaxed condition on the noise also means that assuming its level or amount surpasses a certain threshold is not required anymore, thus, relying on fewer and safer assumptions (presence of arbitrarily small adversarial uncertainty). While field-agnostic composable (hardware) gadgets for masked multiplication can be transferred one-to-one from binary schemes, e.g., ISW [FGP+18], DOM [GMK16], HPC1 [CGLS21], HPC3 [KM22], new gadgets had to be constructed for the squaring operation as it is non-linear in prime fields [CMM+23].

2.7.1 Ciphers and Schemes working over Prime Fields

The application of prime masking to cryptographic primitives is primarily useful when the underlying schemes already natively operate over prime-field elements. Applying prime masking to binary primitives would incur large performance overheads due to the necessity of complex field conversions. Interestingly, however, there is a surprisingly large number of asymmetric and symmetric cryptographic algorithms fulfilling the former requirements. Elliptic-Curve Cryptography (ECC) implementations evidently benefit from masking in prime-order fields [BR23]. The recently standardized lattice-based post-quantum schemes CRYSTALS-Kyber [BDK+18] and CRYSTALS-Dilithium [DKL+18] spend a significant part of their computational effort on multiplications in a polynomial ring defined over primes $q = 3319$ and $q = 8380417 = 2^{23} - 2^{13} + 1$, respectively. Furthermore, so-called *arithmetization-oriented* primitives have been extensively studied in the past years to reduce the cost of recurring operations in advanced cryptographic applications such as Multi-Party Computation (MPC), Fully/Hybrid Homomorphic Encryption (FHE/HHE), and Zero-Knowledge proofs (ZK) [AAB+20]. Examples include (in alphabetical order) Anemoi [BBC+23], CIMINION [DGGK21], GMiMC [AGP+19], Griffin [GHR+23], HADESMiMC [GLR+20], HERA [CHK+21], Hydra [GØSW23], MiMC [AGR+16], Neptune [GOPS22], PASTA [DGH+23], Poseidon [GKR+21], Poseidon2 [GKS23], Reinforced Concrete [GKL+22] and Rescue-Prime [AAB+20, SAD20]. Obviously, the majority of these schemes are not a great fit for resource-constrained applications where side-channel and fault attacks are typically considered as important attack vectors (although some of

the schemes are customizable and can be instantiated with arbitrary primes). For this reason, the authors of [MMMS23] have developed a toy primitive for study purposes called the AES-prime, which is an AES-like cipher instantiated with an efficient S-box and MDS matrix over the chosen prime field $\mathbb{F}_{2^7-1}$. However, the general idea is that more efficient lightweight prime-field primitives are needed to fully leverage the advantages of prime masking. Our work aims at facilitating the development of such primitives by providing useful insights and hints on how to obtain designs with inherently favorable properties for physically secure implementation.

3 Boolean and Prime-Field Encoding vs. SIFA-1

In this section, we analyze and compare the resistance that Boolean masking and prime-field masking provide against *generic* statistical ineffective fault attacks. Generic means that these attacks do not exploit the existence or characteristics of any specific operations or constructions but rather make use of faults that are directly injected into the encoding of the data. Such attacks are universally applicable to any scheme using the analyzed encoding and typically categorized as SIFA-1 [SJR+20]. The goal of this section is to explore whether there is any notable gap in the security against SIFA-1 between additive masking in binary fields and additive masking in prime fields in the presence of strong fault adversaries. From hereon, we assume that a detection-based countermeasure is applied to all implementations we analyze and successfully withholds any faulty ciphertext from being released to adversaries. Without such protection, straightforward DFA attacks could be applicable, and any analysis of the SIFA protection would be meaningless. We start this section by discussing the adversary model considered in this work and provide some initial observations and examples to illustrate the differences between Boolean and prime-field masking with respect to SIFA-1. Then, we detail the toy cipher circuit that is used as a general target in the simulation experiments. Afterwards, we provide a detailed study of the resistance of both encoding schemes, which is supported by extensive simulation results and mathematical explanations for the observed behavior. We divide the analysis into two separate parts, one for stuck-at faults and another for toggle/bit-flip faults. Finally, we shortly discuss the impact of Boolean and prime masking on the security of simple redundancy countermeasures for detection.

3.1 The Heisenberg Adversary Model

Since we are interested in a worst-case analysis of the considered schemes, we assume a very strong adversary model by default and then, where applicable, discuss or demonstrate the impact certain restrictions on its capabilities can have on the attack success. In particular, we allow the adversary to insert an *almost* arbitrary number of stuck-at-0, stuck-at-1 or toggle/bit-flip faults into the encoded state with perfect accuracy and precision at the same time. Alternatively, the adversary may insert an arbitrary number of said faults, but only with *almost* perfect accuracy and precision. The keyword *almost* is crucial in both sentences, as we need to prevent the trivial attack of simply faulting all bits in all shares simultaneously with perfect precision (or all-but-x for x a number independent of the masking order), which can lead to a generic attack independent of the encoding or masking order. Consider for example the injection of stuck-at-0 faults in all bits of all shares of an encoded variable, succeeding with probability 1. For any typical masking scheme, such an attack would lead to an ineffective fault whenever the original secret variable was zero, leaking information independent of the masking order. We believe this attack has little relevance in practice, as it requires the injection of a large and growing (in d) number of perfectly placed faults at the same time to obtain an ineffective fault with probability of only $\frac{1}{\text{field size}}$. Yet, we exclude it explicitly here, as it prevents formal

security amplification in the number of shares. Hence, we impose the mild limitation on the adversary that if the success probability of each injection is 1, she cannot fault all bits in the shares at once. However, faulting all-but-one bit in each share with perfect accuracy, or all bits in all shares with an injection probability < 1 is well within the model. We call this the *Heisenberg* adversary model in reference to the infamous Heisenberg uncertainty principle which states that certain pairs of physical quantities cannot be known with arbitrary precision simultaneously.

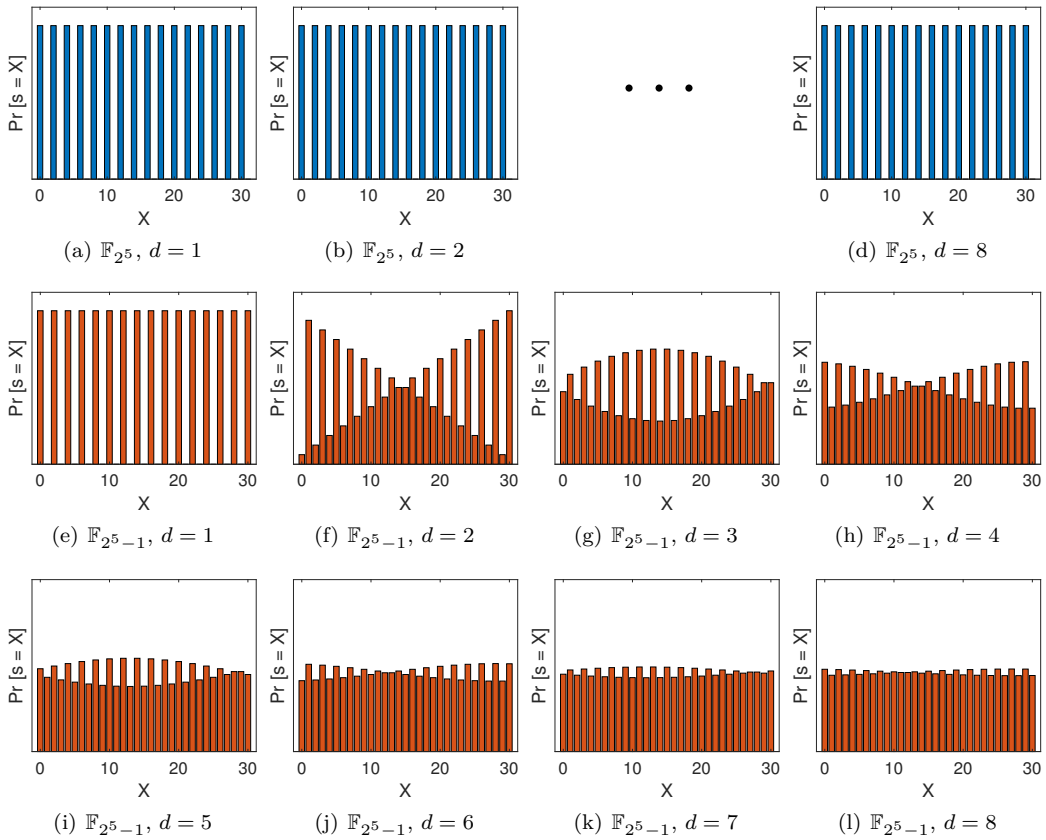
We assume that the adversary targets digital computing hardware and thus manipulates binary digital signals, or in other words, bits. Accordingly, the adversary is explicitly *not* able to inject any faults δ that follow a certain arithmetic structure like, for example, $s'_i = s_i + \delta \bmod p$, or $s'_i = s_i - \delta \bmod p$. This does not appear to be a relevant limitation in practice, as we are not aware of any work claiming such faults to be feasible. Instead, we consider regular stuck-at ($s'_i = s_i \wedge \delta$; $s'_i = s_i \vee \delta$) and toggle faults ($s'_i = s_i \oplus \delta$) which are the type of bit-level manipulations most commonly considered in fault analysis literature. Stuck-at-0 and stuck-at-1 faults are sometimes also denoted as *set* and *reset* faults, respectively. To reiterate, we assume that any effective fault is caught by the detection-based countermeasure and causes an abort instead of releasing the faulty output.

3.2 Model Justification

Why do we consider such a strong adversary capable of injecting multi-bit faults in every share with arbitrary precision at once? This is motivated by a disparity between prior works who claim that masking, in general, is a strong SIFA protection due to the difficulty to fault all shares of an encoded variable at once and practical results which have demonstrated that biasing all shares simultaneously is relatively easy considering hardware implementations with parallel processing of the shares using fault injections methods like clock glitching [SM20, SSM22]. In particular, it was experimentally shown in [SM20] that using clock glitches, it is indeed possible to perform practical SIFA attacks against hardware implementations which are protected by both masking and error correction. This possibility has been confirmed later in [SSM22] by simulations of a post-layout netlist of a three-share PRESENT implementation derived via commercial off-the-shelf design tools. The latter work also provides two potential explanations for such behavior, namely 1) that the same bit of all shares is processed or stored by gates in very close proximity to each other, and 2) that the same bit of all shares can even be stored in gates that are clocked by the same clock buffer. Given these results, it is at least risky to assume that faulting multiple shares of a masked circuit simultaneously becomes infeasible for commonly considered numbers of shares d. In reality, d is certainly expected to have an effect on such attacks, but quantifying how much the success rate is impacted remains an open research problem. We first take the conservative approach, assuming a strong adversary that faces no difficulty when attempting to fault as many shares as desired, and then show how the attack success changes if limiting these capabilities.

3.3 Initial Observation and Example

To get a first intuition for the behavior of Boolean and prime-field masking under attack, let us consider a situation where the adversary injects stuck-at-0 faults in the Least Significant Bit (LSB) of each share s_i of secret value s. Figure 1 shows the resulting probability distributions of s when such a fault is ineffective. Results from $d = 1$ up to $d = 8$ are shown for two different mathematical structures, a binary field $\mathbb{F}_{2^5}$ and a prime field $\mathbb{F}_{2^5-1}$ (since $2^5 - 1 = 31$ is a Mersenne prime), both considering addition as the recombination operation ($\oplus$ and $+$ respectively). Clearly, for Boolean masking, the statistical bias of the distribution is not affected by the increase in order d. Regardless of the number of shares, only odd values of s can lead to an ineffective fault under the described injection pattern.



(a) $\mathbb{F}_{2^5}$, $d = 1$ (b) $\mathbb{F}_{2^5}$, $d = 2$ (d) $\mathbb{F}_{2^5}$, $d = 8$

(e) $\mathbb{F}_{2^5-1}$, $d = 1$ (f) $\mathbb{F}_{2^5-1}$, $d = 2$ (g) $\mathbb{F}_{2^5-1}$, $d = 3$ (h) $\mathbb{F}_{2^5-1}$, $d = 4$

(i) $\mathbb{F}_{2^5-1}$, $d = 5$ (j) $\mathbb{F}_{2^5-1}$, $d = 6$ (k) $\mathbb{F}_{2^5-1}$, $d = 7$ (l) $\mathbb{F}_{2^5-1}$, $d = 8$

Figure 1: Probability distribution of s in fields $\mathbb{F}_{2^5}$ (top row) and $\mathbb{F}_{2^5-1}$ (bottom two rows) when given knowledge of outputs obtained for ineffective fault injections after biasing the LSB of each share of its encoding with a stuck-at-0 fault.

For prime-field masking, however, the increase in masking order lets the distributions progressively approach a uniform distribution. This means that the dependency between the ineffectiveness of the fault and the value of s progressively decreases.

Let us first provide an intuitive explanation for Boolean masking. As illustrated in Figure 2, an ineffective fault in a Boolean-masked bit can only occur after injecting a stuck-at-0 fault in each share if the actual (unmasked) bit is 0. This is true for any encoding of 0 among d shares. A stuck-at-0 fault in all shares converts any possible encoding to a single encoding $\{0\}^d$, which is still a valid representation of 0 and thus may lead to an ineffective fault. In summary, all possible encodings of 0 lead to ineffective faults, while no encoding of 1 can lead to an ineffective fault, i.e., $Pr[s = 0|NF] = 1$ and $Pr[s = 1|NF] = 0$ hold irrespective of d. Here NF denotes the event of an ineffective fault, and F denotes the event of an effective fault. The probability of ineffective faults is given by $Pr[NF] = \frac{2^{d-1}}{2^d} = \frac{1}{2}$, which is also a constant irrespective of the masking order.

Next, we consider the case of prime-field masking. Figure 3 illustrates a simple example based on prime $p = 7$ and $d = 2$ shares.[3] As detailed, the faults are injected at the LSBs of all the shares, and the fault model is stuck-at-0. For each value of the secret variable,

[3]The choices of primes we make for these examples are based on the ease of illustration. The actual impact of the prime size will be discussed in the following sections.

$d = 2$

$s = 0$			$s = 1$		
s_1	s_2		s_1	s_2	
0	0	NF	0	1	F
1	1	NF	1	0	F

$d = 3$

$s = 0$				$s = 1$			
s_1	s_2	s_3		s_1	s_2	s_3	
0	0	0	NF	0	0	1	F
1	1	0	NF	1	0	0	F
1	0	1	NF	1	1	1	F
0	1	1	NF	0	1	0	F

Figure 2: Boolean masking under stuck-at-0 faults on all shares. The red colour indicates bit values for which a change happens due to faults, and the blue colour indicates the opposite. The label NF indicates no fault on the unmasked state (i.e., $\bigoplus_{i=0}^{d-1} s_i$), and the label F indicates a faulty unmasked state.

$d = 2,\ p = 7$

$s = 0$			$s = 1$			$s = 2$			$s = 3$			$s = 4$			$s = 5$			$s = 6$		
s_1	s_2		s_1	s_2		s_1	s_2		s_1	s_2		s_1	s_2		s_1	s_2		s_1	s_2	
0	0	NF	0	1	F	0	2	NF	0	3	F	0	4	NF	0	5	F	0	6	NF
1	6	F	1	0	F	1	1	F	1	2	F	1	3	F	1	4	F	1	5	F
2	5	F	2	6	NF	2	0	NF	2	1	F	2	2	NF	2	3	F	2	4	NF
3	4	F	3	5	F	3	6	F	3	0	F	3	1	F	3	2	F	3	3	F
4	3	F	4	4	NF	4	5	F	4	6	NF	4	0	NF	4	1	F	4	2	NF
5	2	F	5	3	F	5	4	F	5	5	F	5	6	F	5	0	F	5	1	F
6	1	F	6	2	NF	6	3	F	6	4	NF	6	5	F	6	6	NF	6	0	NF

Figure 3: Prime-field masking under stuck-at-0 faults on the LSBs of all shares. The red colour indicates values for which a change happens due to faults, and the blue colour indicates the opposite. The label NF indicates no fault on the unmasked state (i.e., $(\sum_{i=0}^{d-1} s_i) \mod p$), and the label F indicates a faulty unmasked state.

there is at least one possible encoding for which the injected fault pattern is ineffective (NF) and multiple encodings for which it is effective (F). Contrary to the situation in Boolean masking, the adversary may thus not exclude any secret value when obtaining an ineffective fault. However, since the distribution is obviously not uniform ($Pr[NF|s]$ is distinct for the unmasked values), the adversary still learns information by repeating this process in order to perform a statistical fault attack. Yet, with increasing d, the amount of information that may possibly be extracted decreases drastically.

3.4 A Toy Circuit.

In this work, we mostly target different toy ciphers and circuits to keep the main analysis simple and independent of any concrete primitives. For illustration purposes and to focus only on the important aspects, it is usually helpful to analyze a minimal working example. We have derived the generic toy circuit depicted in Figure 4 as such a minimal working example for SIFA-1. Here, p_0 and p_1 are two plaintext words, k_0 and k_1 are two key words and c_0 and c_1 are two ciphertext words. All words are of size n bits. M is a linear operation and is assumed to be realized as multiplication by the matrix $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$,[4] while S is assumed to be a non-linear operation, namely a cryptographically strong S-box. For comparison purposes, we consider two different instantiations of this circuit, one for analyzing binary-field masking and one for prime-field masking.

[4] $\begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix}$ in $\mathbb{F}_2$-polynomial notation for binary field operation.

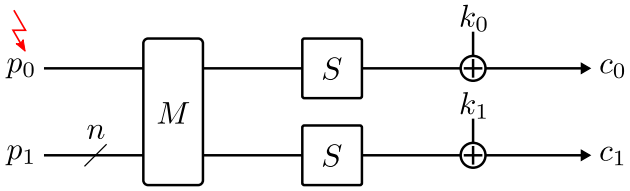


Figure 4: Toy cipher circuit for SIFA-1 analysis. Fault location is denoted by a red lightning symbol.

- For the circuit simulations based on Boolean masking, we choose the field $\mathbb{F}_{2^n}$ with $n = 8$, resulting in $M^{-1} = \begin{pmatrix} 82 & 164 \\ 164 & 82 \end{pmatrix}$.[5] S is chosen to be the AES S-box, based on multiplicative inversion in the field under irreducible polynomial $x^8 + x^4 + x^3 + x + 1$.
- For the circuit simulations based on prime masking, we choose the field $\mathbb{F}_{2^n-1}$ with $n = 7$, resulting in $M^{-1} = \begin{pmatrix} 42 & 43 \\ 43 & 42 \end{pmatrix}$. S is chosen to be the AES-prime S-box, defined as $S(x) = x^5 + 2 \bmod 2^7 - 1$.

While M^{-1} is directly determined by the chosen field, we stress that the choices for S have been made without loss of generality. We have verified that other choices of cryptographically strong S-boxes lead to equivalent results. While the word sizes n have a disparity of 1 bit between Boolean and prime circuit, this is only owed to the facts that $2^8 - 1$ is not a prime number and that no 7-bit binary S-boxes are widely used in symmetric cryptography. Furthermore, these two fields and S-boxes have been compared in previous works on prime-field masking already [MMMS23, CMM+23]. Since we also analyze primes of different sizes later on, we believe this small disparity will not affect our general conclusions drawn from this comparison. In the following, we consider attacks that repeatedly inject faults in the encoding of p_0 in order to obtain a distinguisher for both k_0 and k_1 from the collection of ineffective faults.

3.5 Simulated Attacks and Analysis for Stuck-at Faults

After the introductory example and the description of the target, we continue by simulating concrete attacks for different encodings and numbers of shares and then provide an analytical description of the observed behavior. We consider exclusively stuck-at faults in this part of the section.

Simulation Results. Following the intuitive example discussed earlier, we first assume that the adversary injects stuck-at-0 faults in the LSBs of all Boolean and prime-field shares, respectively. The attack works by performing a certain number of fault attempts on p_0 for randomly chosen input data and collecting all outputs obtained for ineffective fault injections. Although not explicitly illustrated in Figure 4, the shared representations of c_0 and c_1, respectively, are unmasked before being released to the output. For each output, the adversary calculates backwards through the circuit up the injection point by guessing k_0 and k_1 and assessing the uniformity of the distribution for each key guess by calculating the Squared Euclidean Imbalance (SEI) [DEK+18]. If the attack is successful, the maximum SEI among all key candidates corresponds to the correct key.

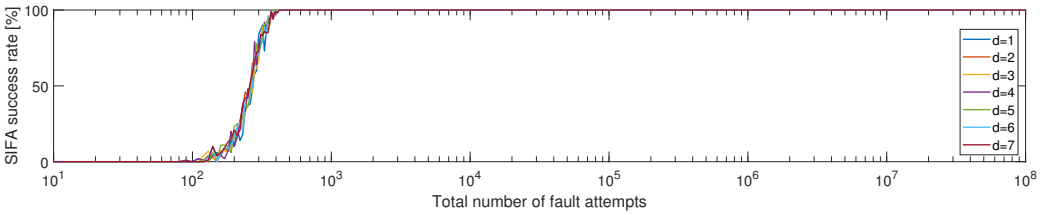
Figure 5 shows the results of such SIFA-1 attacks on the toy circuit from Figure 4. The top half of the figure corresponds to the Boolean masked circuit, while the bottom half

[5] $\begin{pmatrix} x^6 + x^4 + x & x^7 + x^5 + x^2 \\ x^7 + x^5 + x^2 & x^6 + x^4 + x \end{pmatrix}$ in $\mathbb{F}_2$-polynomial notation.

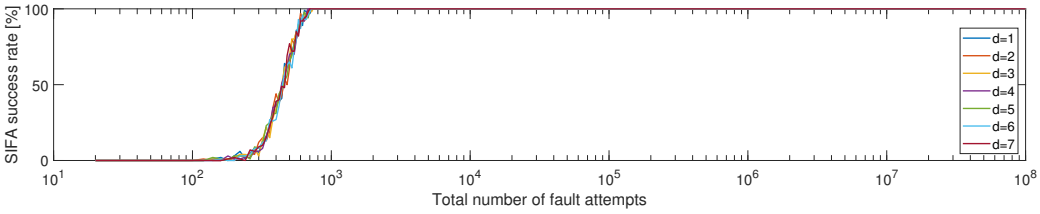
corresponds to prime-field masking. Each half contains three graphs for different levels of biasing success (100 %, 80 %, 60 %), which is described in the following paragraph in more detail. The success rate of the attacks is plotted over the number of fault injection attempts conducted. As denoted by the legends, the curves in different colors refer to the results for different numbers of shares d from $d = 1$ (unmasked) to $d = 7$ (6th-order SCA security). For each point of the curves, we performed 100 independent simulations for uniformly random inputs p_0 and p_1 (freshly chosen for each iteration within each attack simulation) as well as uniformly random keys k_0 and k_1 (freshly chosen for each attack simulation) making use of the total number of fault attempts denoted on the x-axis. The values on the y-axis show the portion of simulations among the 100 independent attacks where *both* keys k_0 and k_1 are correctly determined in the attack by resulting in the maximum SEI among all key candidates. The probability of guessing the correct key in this process by chance, without the attack actually succeeding, is $\frac{1}{(2^8)^2} = 0.0015$ % for the binary case and $\frac{1}{(2^7-1)^2} = 0.0062$ % for the prime case, hence it is not expected to affect the graphs visibly.

As already observed in the introductory example, no dependency on the number of shares can be observed in the case of Boolean masking. However, in the case of prime-field masking, the number of fault attempts needed to perform an attack with a certain success rate increases exponentially in the number of shares (the x-axis is plotted in logarithmic scale). We have performed the attack for three different values of *biasing success*, namely 100 %, 80 % and 60 %. By biasing success, we denote the percentage of cases in which the adversary succeeds in biasing all d shares as intended, thereby causing an informative fault injection (either effective or ineffective). In the cases where the adversary fails to introduce the bias, for example, because only a part of the shares are affected by the injection, the probability distribution $Pr[s|NF]$ would be uniform and hence any ineffective fault related to it non-informative. This is an attempt to simulate a slightly weaker adversary without yet making its success dependent on the number of shares. It can be observed that lower biasing success consistently leads to an offset on the x-axis, but also that in the case of prime masking, the distance between the curves for different numbers of shares is consistently wider, indicating an additional exponential amplification of the imprecision that the adversary faces. Hence, both in the presence of a perfect adversary and in the presence of a slightly imperfect (but still very strong) adversary, prime masking exhibits security advantages of multiple orders of magnitude against SIFA-1 compared to Boolean masking. For completeness, we have also performed the same simulations under the assumption that the adversary has a certain independent probability to succeed in each individual single-bit fault injection per share and, therefore, needs to stack at least d successes in order to bias the entire encoding. This corresponds to the random fault model introduced in [DN23]. The results are depicted in Figure 6. While under this assumption, any encoding scheme can provide exponential security amplification in the number of shares against imperfect adversaries (injection probability < 1), including Boolean masking, it is still evident that adversaries require orders of magnitude more fault attempts to perform the same attacks against higher-order prime-field masking. Yet, as discussed before, the underlying assumption that each injection is an independent event with fixed probability to occur, has been demonstrated to not always hold in practice [SM20, SSM22]. Complexity estimations for more realistic assumptions, or even data collected from real-world experiments, would probably fall somewhere in between Figure 5 and Figure 6.

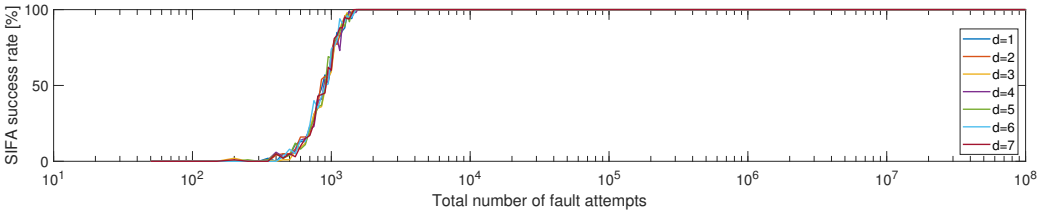
Another important detail regarding the prime-field simulations is, that two separate effects are causing the exponential security amplification. In the case of the LSB stuck-at-0 faults in all shares, we can observe that 1) the number of correct outputs that need to be collected for ineffective faults to accumulate enough information about the distribution to perform a successful SIFA increases exponentially in the number of shares, and 2) the



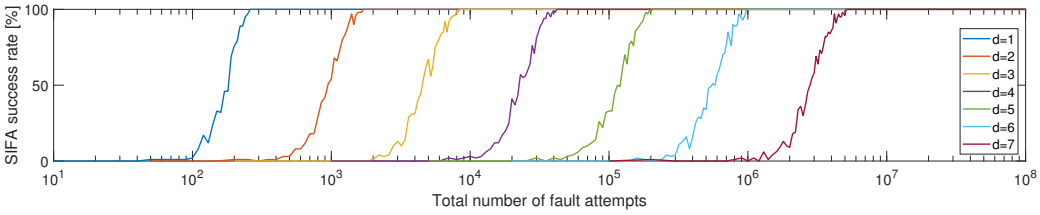
(a) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 100 % biasing success.



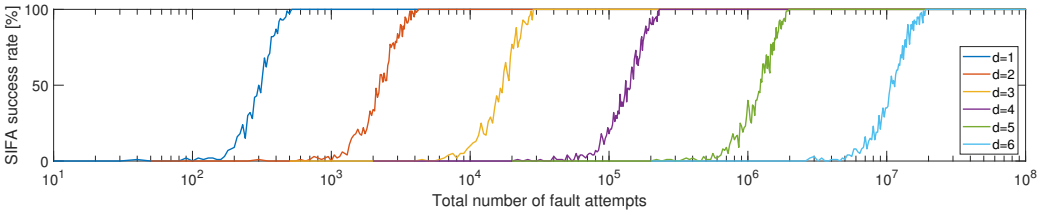
(b) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 80 % biasing success.



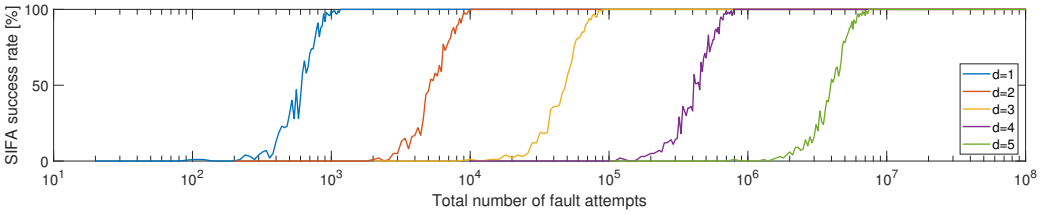
(c) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 60 % biasing success.



(d) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 100 % biasing success.

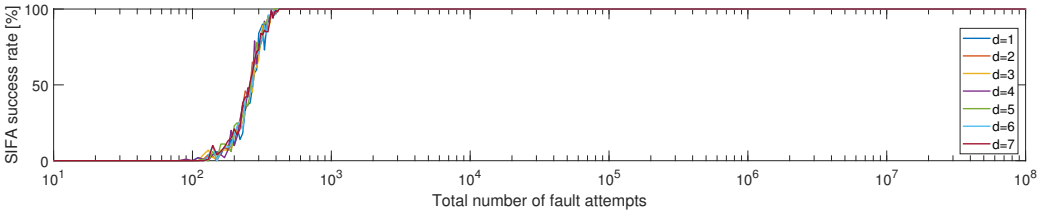


(e) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 80 % biasing success.

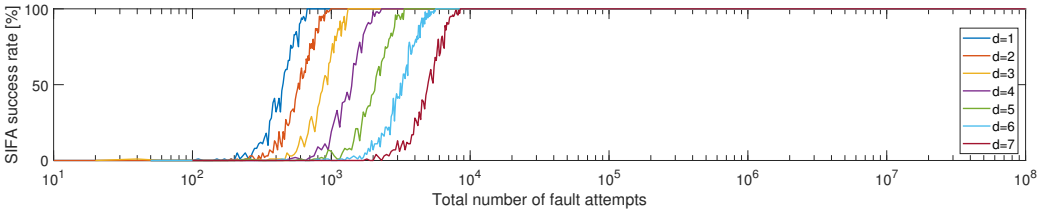


(f) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 60 % biasing success.

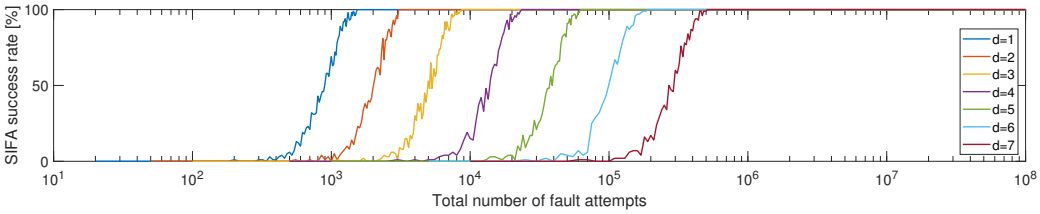
Figure 5: SIFA-1 attack results based on LSB stuck-at-0 faults in all shares of p_0 of masked toy circuit (Figure 4) with different numbers of shares d and for different levels of adversarial biasing success.



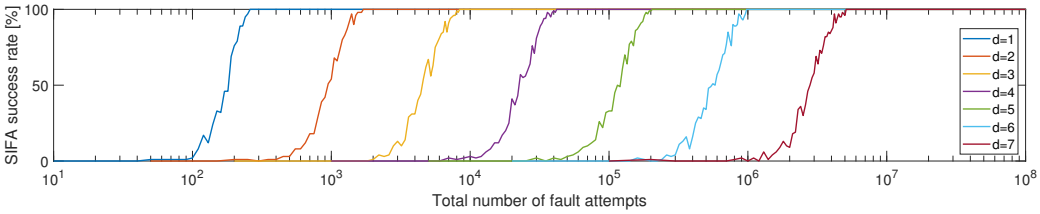
(a) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 100 % injection probability per share.



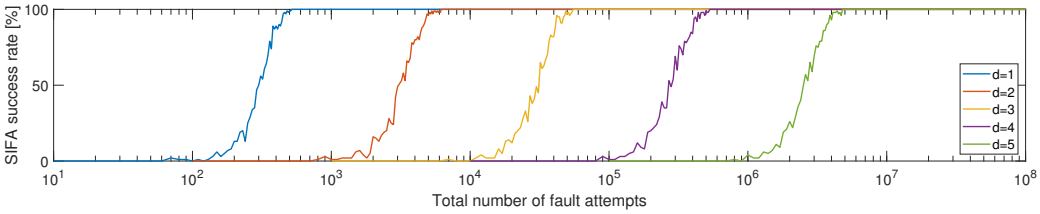
(b) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 80 % injection probability per share.



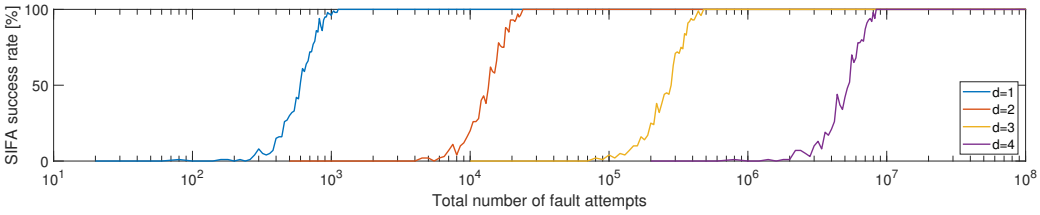
(c) **Boolean** masking in $\mathbb{F}_{2^8}$. Attack simulation with 60 % injection probability per share.



(d) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 100 % injection probability per share.



(e) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 80 % injection probability per share.



(f) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 60 % injection probability per share.

Figure 6: SIFA-1 attack results based on LSB stuck-at-0 faults in all shares of p_0 of masked toy circuit (Figure 4) with different numbers of shares d and for different levels of adversarial per-share injection probability.

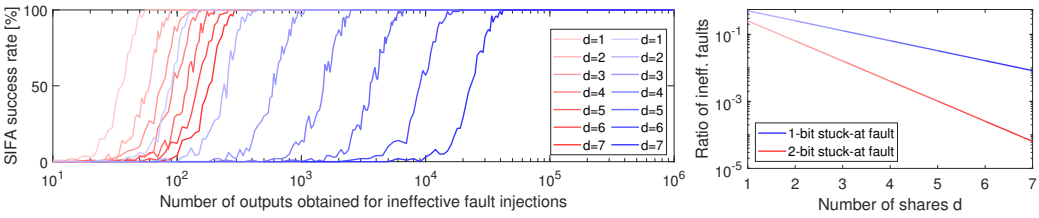


Figure 7: Separate depiction of the number of ineffective faults needed to perform a successful SIFA-1 attack and the probability to obtain an ineffective fault for different numbers of shares, depending on the number of bits with stuck-at-0 faults per share.

number of ineffective faults among a fixed number of attempts decreases exponentially in the number of shares. We have depicted these effects separately in Figure 7 for 1-bit (blue) and 2-bit (red) stuck-at-0 faults per share, respectively. While the outputs obtained for ineffective faults become much more informative when faulting multiple bits per share (hence needing to collect less of them for a successful attack), they also occur much less frequently. For the chosen circuit and prime, this results in a net loss in attack performance, considering the total number of fault attempts that an adversary requires to extract information (this is specific and does not hold for arbitrary choices). We have verified exhaustively for our prime-field toy circuit instantiation that no stronger attack than the presented ones can be achieved by faulting different bits than the LSB (regardless of fixed or varying location per share), by faulting more than one bit per share (using any possible combination of locations) or by injecting stuck-at-1 instead of stuck-at-0 faults. Hence, the presented fault injection pattern, namely LSB stuck-at-0 in each share, constitutes the strongest possible attack any adversary fitting our model might perform on this exemplary target. Please note that our simulation-based study focuses on *Mersenne* primes, making some of our observations specific to this choice. The following analysis, however, is independent of the choice of p.

Analytical Results. Let us consider a prime-field encoding with d shares as $\{s_0, s_1, \ldots, s_{d-1}\}$. The unmasked state s is given as:

$$s = \sum_{i=0}^{d-1} s_i \bmod p \tag{1}$$

where $s_i \in \mathbb{F}_p$. Every possible encoding of a value s satisfies Equation (1). Depending on the value of s_i, injecting a stuck-at-0 fault in its LSB can have two different effects:

- If the LSB of s_i is 0, the value remains unchanged.
- If the LSB of s_i is 1, the faulted value is $s_i - 1$.

Therefore, a fault in this model becomes ineffective only when $\forall i, s_i = 2m$ with m a positive integer. Also, $\forall i, 0 \leq s_i \leq (p-1)$. For all other cases, the faulty value $s' < s$. With these constraints on the value of each s_i, we may count the number of possible solutions to Equation 1 for each value of s. The number of possible solutions for an s can be counted as the sum of coefficients of terms $z^{(kp+s)}$ in the following polynomial:

$$(1 + z^2 + z^4 + \cdots + z^{(p-1)})^d.$$

Here $k \geq 0$ and $(kp+s) \leq d(p-1)$. Let us denote such a sum of coefficients as $C_{\mathrm{LSB}-st0}(s)$. The expression for $C_{\mathrm{LSB}-st0}(s)$ is given as:

$$C_{\mathrm{LSB}-st0}(s) = \sum_{\substack{\{(s+kp)\} \\ k\in\mathbb{N} \\ (s+kp)\leq d(p-1)}} \sum_{\substack{j=0 \\ (s-j(p+1)) \bmod 2=0 \\ \wedge(s-j(p+1))\geq 0}}^{d} \binom{d}{j}\binom{d+\frac{s-j(p+1)}{2}-1}{d-1}$$

Given $C_{\mathrm{LSB}-st0}(s)$, we calculate:

$$Pr[NF|s] = \frac{C_{\mathrm{LSB}-st0}(s)}{p^{d-1}}$$

and

$$Pr[NF] = \left(\frac{p+1}{2p}\right)^d.$$

Finally, we have

$$Pr[s|NF] = \frac{2^d \cdot C_{LSB-st0}(s)}{(p+1)^d}$$

which is the same quantity depicted in Figure 1. It is quite evident from the expressions that the probability of obtaining an ineffective fault decreases exponentially in the cardinality of the prime field $\mathbb{F}_p$ (ref. Figure 8(a)). The impact is that, with an increase in d, the number of required fault injections also increases exponentially. For Boolean masking, this factor is constant irrespective of d.

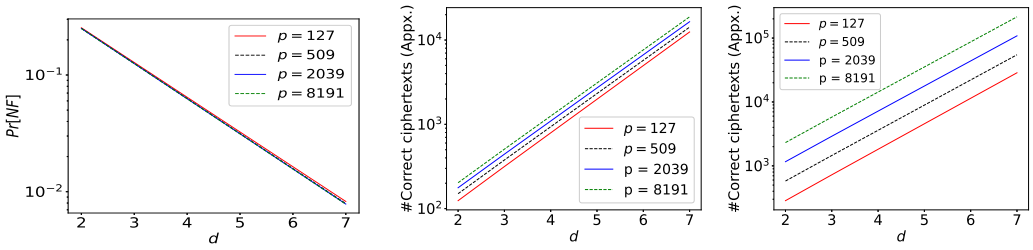
The probability of ineffective faults is not the only factor which determines the attack complexity. The other determining factor is the distribution $Pr[s|NF]$. In order to understand the impact of the masking order on this distribution, we first recall that SIFA attacks exploit the deviation of this distribution $f(x) := Pr[x = s|NF]$ for $s \in \mathcal{X}$ from a uniform random distribution $\theta(x)$ with the same support. This deviation is measured by calculating a statistic for each key guess and assigning a score to each key. The key with the highest score assigned is assumed to be the correct key. Usually, the SEI statistic (proportional to Pearson's χ^2 statistic, if $\theta(x)$ is uniform) is used for SIFA as $f(x)$ is unknown in a real scenario and can only be estimated. However, since we have the exact distribution for our analytical model, we can also use the optimal statistic called the Log-Likelihood-Ratio (LLR). For any of these statistics, let us define Δ_a as the difference between the scores of the correct key and the key with rank $2^{\kappa-a}$. Here, 2^κ is the number of key candidates. For a prime field, 2^κ will be a number closest to $p-1$. The success probability of the attack is given as $Pr[\Delta_a > 0]$. Based on this formulation, we can estimate the required number of correct ciphertexts corresponding to the LLR and Pearson's χ^2 statistic (CHI) as follows [DEG+18]:

$$N_{LLR} = \frac{2\left(\Phi_{0,1}^{-1}(Pr[\Delta_a > 0]) + \Phi_{0,1}^{-1}(\alpha)\right)^2}{\mathcal{C}(f,\theta)} \qquad N_{CHI} = \frac{t_1 + \sqrt{{t_1}^2 - t_2}}{\mathcal{C}(f,\theta)}$$

where $t_1 = \sqrt{2h}\Phi_{0,1}^{-1}(\alpha) + 2\Phi_{0,1}^{-2}(Pr[\Delta_a > 0])$ and $t_2 = 2h(\Phi_{0,1}^{-2}(\alpha) - \Phi_{0,1}^{-2}(Pr[\Delta_a > 0]))$. Here, $\alpha = 1 - 2^{-a}$, $h = |\mathbb{F}_p| - 1$. Further, $\mathcal{C}$, also called the *capacity*, is calculated as follows:

$$\mathcal{C}(f,\theta) = \sum_{x\in\mathcal{X}} \frac{(f(x)-\theta(x))^2}{\theta(x)}$$

with $\mathcal{X} := \mathbb{F}_p$. We note that any such estimation is only meaningful if $f(x)$ is very close to uniform. Figures 8(b)–(c) plot N_{LLR} and N_{CHI} for different masking order and prime sizes, for $a = \kappa - 1$, and $Pr[\Delta_a > 0] = 0.99$. We have chosen the primes for illustration



(a) (b) (c)

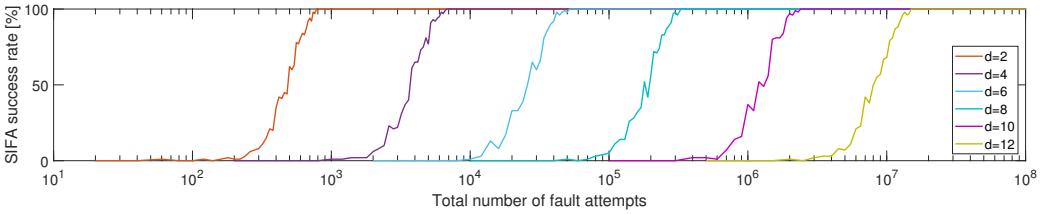
Figure 8: Analytical estimations for LSB-stuck-at-0 fault model: (a) Probability of ineffective faults with respect to d for different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes); b) Estimated count of correct ciphertexts with LLR statistic with respect to d and different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes); c) Estimated count of correct ciphertexts with CHI statistic with respect to d and different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes).

purposes, with a 2 bit difference in binary length between neighbors. As it can be observed, the required number of correct ciphertexts grows exponentially with the masking order d. Also, there is a constant increase in ciphertext count for an increase in prime size (we shall explore this point further in Section 3.7). Overall, our analytical estimations match the simulation results – increasing the masking order makes the attacks exponentially difficult. We recall that for Boolean masking, $f(x)$ does not change with the masking order. Therefore, the statistics (resp. the count of correct ciphertexts) will remain constant for Boolean masking irrespective of d.

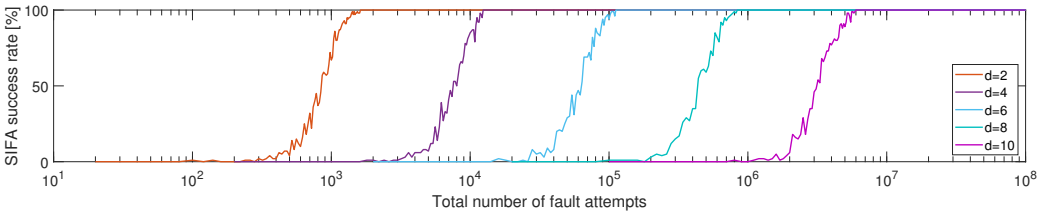
3.6 Simulated Attacks and Analysis for Toggle/Bit-Flip Faults

As a next step we consider a similar scenario, but the adversary attempts to inject toggle/bit-flip faults instead. We begin with a quick analysis of LSB bit-flip faults on Boolean encoded data. Such a fault pattern cannot leak any information. To understand why, let us consider the Boolean sharing $\{s_0, s_1, \cdots s_{d-1}\}$ such that $s = \bigoplus_{i=0}^{d-1} s_i$. With LSB bit-flip faults in all the shares, the faulty state becomes $s' = \bigoplus_{i=0}^{d-1}(s_i \oplus 1) = \bigoplus_{i=0}^{d-1} s_i \oplus \bigoplus_{i=0}^{d-1} 1 = s \oplus \bigoplus_{i=0}^{d-1} 1$. If d is even, the unmasked state remains fault-free, whereas the LSB of the unmasked state is always flipped if d is odd. For none of the cases, the ineffectiveness depends on the actual state value. Therefore, (unbiased) bit-flip faults are not useful to perform SIFA on Boolean masking.

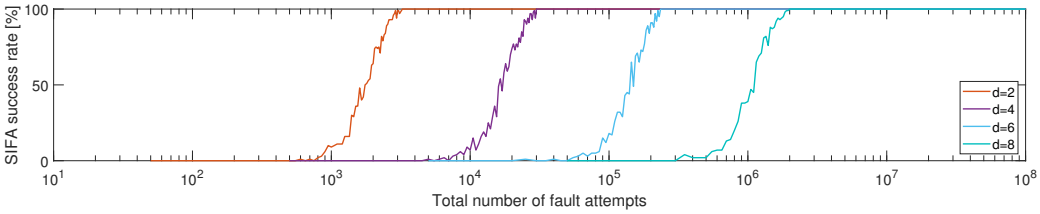
Prime-field masking, on the other hand, may indeed leak information for SIFA-1 attacks under certain circumstances when injecting bit-flip faults in all shares of an encoding. For example, when all LSBs are flipped, and *the number of shares is even*, ineffective faults can occur and can leak information about the unshared variable. While this appears to be a negative result at first, as prime masking is vulnerable to a specific attack that may not work on Boolean masking, we stress that 1) this observation has little relevance in practice and 2) prime-field masking still provides exponential security amplification under that attack. The fact that Boolean encodings are resilient to SIFA-1 using bit-flip fault injections is strictly limited to perfect unbiased bit-flips with a 100 % probability to cause $0 \mapsto 1$ transitions and a 100 % probability to cause $1 \mapsto 0$ transitions (depending on the initial value of the bit). It is clear from practical experience, and has also been argued in the original SIFA proposal already, that such perfect fault injections are not typically observed in practice [DEK+18]. Once the probability of causing $0 \mapsto 1$ transitions differs from that of causing $1 \mapsto 0$ transitions (biased toggle), the resilience of Boolean masking



(a) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 100 % biasing success.



(b) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 80 % biasing success.



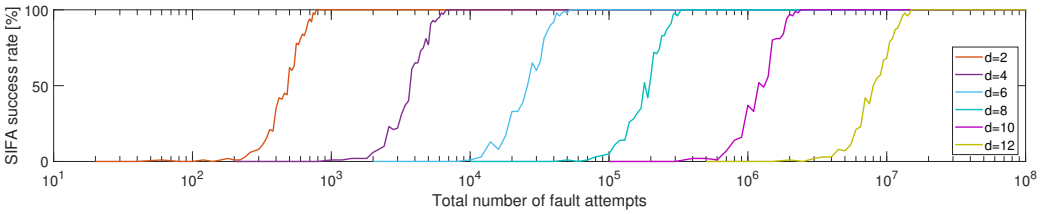
(c) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 60 % biasing success.

Figure 9: SIFA-1 attack results based on LSB toggle/bit-flip faults in all shares of p_0 of masked toy circuit (Figure 4) with different numbers of shares d and for different levels of adversarial biasing success. The attacks using an LSB toggle in each share only leads to ineffective faults if the number of shares is even.

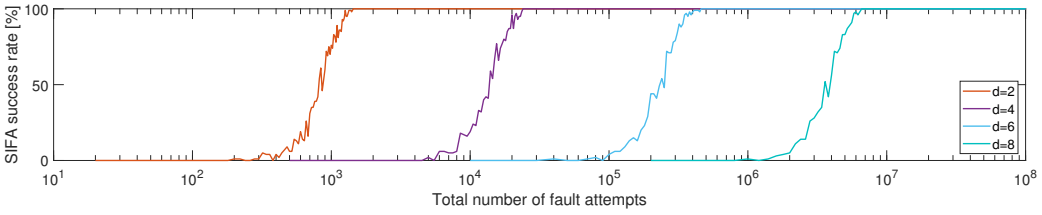
(and prime-field masking with odd numbers of shares) collapses. Hence, while studying the impact of perfect unbiased bit-flip faults is an interesting theoretical exercise, and will therefore be presented in the following, its practical impact is likely to be limited.

Simulation Results. We now assume that the adversary injects toggle/bit-flip faults in the LSBs of all prime-field shares. The attack procedure is then the same as in the previous part of this section, which focused on stuck-at faults. The fault is injected in the encoding of p_0 in Figure 4 in order to create a biased distribution depending on the unmasked state that can be used to build a distinguisher for k_0 and k_1. The simulated attack results are depicted in Figure 9 for even numbers of shares only. When the number of shares is odd, this fault injection pattern never leads to an ineffective fault, hence no attack may be performed. Once again, we provide three graphs for different levels of biasing success (100 %, 80 %, 60 %) in order to simulate slightly different adversaries (as before, these probabilities are related to the success of the adversary in biasing the entire encoding using perfect unbiased fault injections). While we also achieve exponential security amplification in the number of shares here, the absolute numbers of fault attempts required are lower for the same masking order compared to Figure 5. The impact of the adversary's biasing success on the attack is similar to the stuck-at fault scenario discussed before. When the individual fault injections in each share are assumed to be independent events with a fixed probability following the random fault model [DN23] (as detailed before, this is not always a safe assumption to make), we obtain the simulation results depicted in Figure 10.

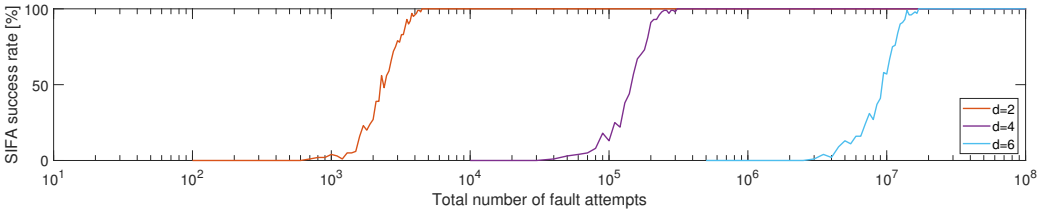
We now take a look at the two responsible effects separately again in Figure 11, namely



(a) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 100 % injection probability per share.



(b) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 80 % injection probability per share.



(c) **Prime** masking in $\mathbb{F}_{2^7-1}$. Attack simulation with 60 % injection probability per share.

Figure 10: SIFA-1 attack results based on LSB toggle/bit-flip faults in all shares of p_0 of masked toy circuit (Figure 4) with different numbers of shares d and for different levels of adversarial per-share injection probability. The attacks using an LSB toggle in each share only leads to ineffective faults if the number of shares is even.

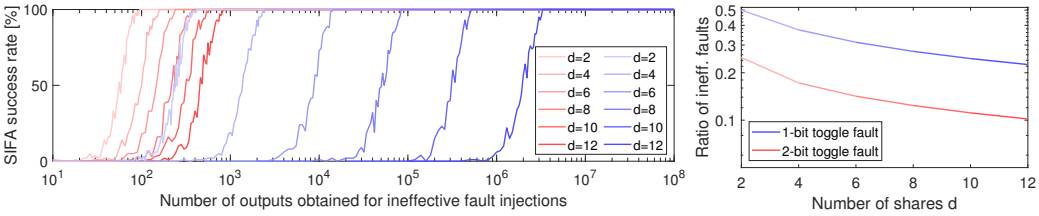


Figure 11: Separate depiction of the number of ineffective faults needed to perform a successful SIFA-1 attack and the probability to obtain an ineffective fault for different numbers of shares, depending on the number of bits toggled per share.

the number of ineffective faults needed for an attack and the likelihood of obtaining an ineffective fault. While the informativeness of ineffective faults appears very similar compared to those obtained in the stuck-at model, the ratio of ineffective faults does not decrease exponentially in the number of shares here. Furthermore, the slope of the decrease in the right graph of Figure 11 is the same for 1-bit and 2-bit faults, showing that the multi-bit toggle faults in each share actually help to decrease the total number of fault attempts needed. Hence, very strong adversaries, able to reliably flip many bits in each of many shares, can theoretically achieve attacks with a very low number of fault attempts (yet, still exponential in d). With multi-bit toggle faults that vary between the different shares (at least one fault-tuple has to be different than the rest), also attacks on odd numbers of shares are feasible using bit-flip fault injections. Yet, the complexity of such attacks is higher relative to the order of masking, which means that even numbers of shares are generally more vulnerable under bit-flip faults. In all these cases, the number of attempts still increases exponentially in the number of shares as well as the adversary's imprecision and is thus generally much larger than for comparable SIFA-1 on Boolean masked circuits.

There exists one peculiar corner case when using multiple optimal bit-flip faults on *Mersenne*-prime encodings. Namely, in Mersenne-prime fields, when flipping *all* the bits of the binary representation of an entire share s_i, then regardless of the concrete value of s_i, one obtains the share's additive inverse $s_i' = -s_i$. This opens up the possibility of removing such share from the equation when comparing s and s'. However, the concrete impact of this is unclear, and the fault injection is highly impractical. Furthermore, the adversary model only permits to perform such an injection with limited precision and accuracy. We have not observed any similar cases for non-Mersenne prime numbers, as they do not exhibit the same kind of regularity with respect to additive inverses.

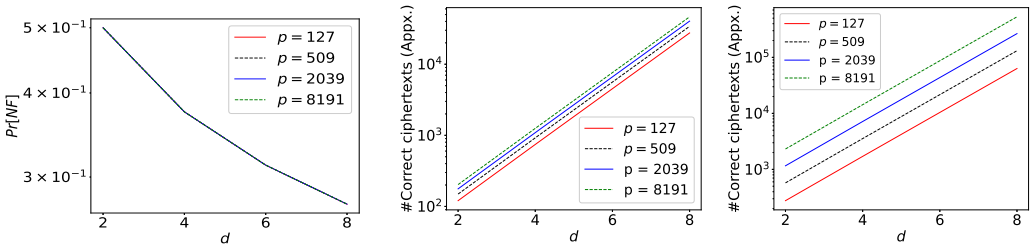
Analytical Results. Let us now consider the prime-field encoding with d shares as $\{s_0, s_1, \ldots, s_{d-1}\}$ again. The unmasked state s is given as:

$$s = \sum_{i=0}^{d-1} s_i \bmod p \tag{2}$$

where $s_i \in \mathbb{F}_p$. Every possible encoding of a value s satisfies Equation (2). Depending on the value of s_i, injecting a bit-flip fault in its LSB can have two different effects:

- If the LSB of s_i is 0, the faulted value is $(s_i + 1)$.
- If the LSB of s_i is 1, the faulted value is $(s_i - 1)$

Let us first consider the case where d is odd. In this case, the unmasked value will always be faulty under LSB flips in each share. However, for a sharing with even order,



(a) (b) (c)

Figure 12: Analytical estimations for LSB-toggle/bit-flip fault model: (a) Probability of ineffective faults with respect to d for different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes); b) Estimated count of correct ciphertexts with LLR statistic with respect to d and different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes); c) Estimated count of correct ciphertexts with CHI statistic with respect to d and different prime sizes (7-bit, 9-bit, 11-bit, and 13-bit primes).

there can be ineffective faults. More precisely, if there are equal numbers of even-valued and odd-valued shares in an encoding instance, then the fault will become ineffective. It can be deduced from Figure 3 (by considering LSB-bit-flip faults) that the number of such encodings varies for different values of s. Therefore, LSB bit-flip faults can be utilized for an attack if d is even.

Next, we quantify the amount of the aforementioned leakage. The number of distinct encodings for a value s, in which there are equal numbers of even and odd shares, can be counted as the sum of coefficients of the following polynomial:

$$(1 + z^2 + z^4 + \cdots + z^{p-1})^{\frac{d}{2}} (z + z^3 + \cdots + z^{p-2})^{\frac{d}{2}}$$

Additionally, for each such distinct encoding choice, there are a total $\binom{d}{\frac{d}{2}}$ permutations of the shares. Consequently, we can compute the quantity $C_{\text{LSB}-bf}$ similar to the stuck-at-0 faults. Although deriving a closed-form expression for $C_{\text{LSB}-bf}$ is complex in this case, it is possible to compute explicitly for each even value of d using efficient polynomial multiplication routines available in standard programming languages (at least up to certain values of p and d). The $Pr[NF]$ can be directly computed from the polynomial as $Pr[NF] = \binom{d}{\frac{d}{2}} \frac{(p^2-1)^{\frac{d}{2}}}{(2p)^d}$. Finally, we have

$$Pr[S|NF] = \frac{2^d \cdot C_{\text{LSB}-bf}}{\binom{d}{\frac{d}{2}}(p^2-1)^{\frac{d}{2}}}.$$

The probability of ineffective faults, as well as the estimated count of correct ciphertexts, are depicted in Figure 12(a) and Figures 12(b)–(c), respectively. Once again, we confirm that the security amplifies exponentially in the masking order. Furthermore, the positive impact of prime size is retained for $Pr[s|NF]$. However, as expected, the impact of increasing d on $Pr[NF]$ is not as rapid as the case of stuck-at-0 faults. Also, we observe that the estimations with LLR statistic show lesser counts for the ciphertexts than that with the CHI statistic (same for the stuck-at-0 case). However, this is mainly attributed to the fact that the estimations with LLR have some ideal assumptions regarding the knowledge of the distribution $f(x)$. The CHI statistic is closer to the reality, and, in the next subsection, we shall use values corresponding to this statistic to compare our estimations with the simulation results.

3.7 Impact of the Field Size

One interesting question about the security of prime-field masking is, of course, the impact that the size of the prime (and thus the cardinality of the field) has on the number of fault attempts needed for a successful SIFA-1. As it can be observed from the analytical estimations (ref. Figures 8 and 12), the growth of the size of the prime increases the attack complexity. We also performed a simulation-based analysis to visualize these results in Figure 13.[6] The three graphs at the top show the impact of the prime size on the success of SIFA-1 on 2-share, 3-share and 4-share prime-field masking under LSB stuck-at-0 faults. The bottom three graphs show the equivalent for 2-share, 4-share and 6-share prime-field masking under LSB toggle/bit-flip faults. All results are obtained for Mersenne primes only, including $p = 2^5 - 1 = 31$, $p = 2^7 - 1 = 127$, $p = 2^{13} - 1 = 8191$, $p = 2^{17} - 1 = 131071$, $p = 2^{19} - 1 = 524287$ which can lead to efficient implementations either in hardware or software according to [MMMS23]. The security gain from increasing the size of the prime appears to be constant on the log scale, regardless of d, which is also consistent with the trend observed from the analytical estimates. We also present the analytical estimates for the total ciphertext counts with the CHI statistic in Figure 14. This quantity is calculated by multiplying the estimated count of correct ciphertexts with the inverse of the probabilities of ineffective faults ($Pr[NF]$). These plots and the simulation results can be directly compared for the 7-bit and the 13-bit primes, which shows that the ciphertext counts for the estimation and the simulation are of the same order. We note that the estimates with CHI statistics are always closer to reality than those of the LLR statistics. One may also observe from the results that the difference between the smallest and largest primes is about two orders of magnitude, which can make a huge difference in the feasibility of attacks in practice. In general, it appears, and we will come to similar conclusions in Section 4 on structural attacks, that larger primes are generally superior for the security of such masking against statistical ineffective fault attacks.

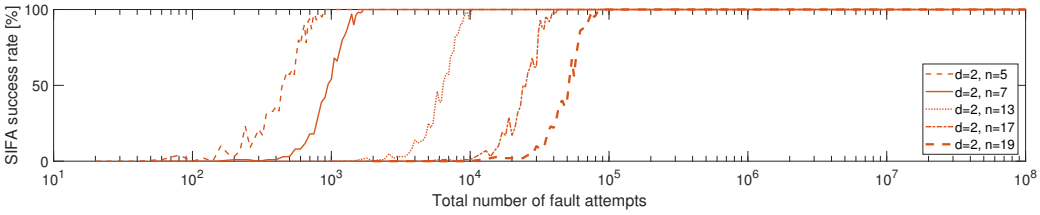
3.8 Exemplary SIFA-1 Results on AES and AES-prime

To demonstrate the resistance of concrete prime-field masked cipher implementations we have analyzed the performance of simulated SIFA-1 attacks on both AES and AES-prime [MMMS23] implementations with ($d > 1$) and without masking applied ($d = 1$). The corresponding results are depicted in Table 3.8. The simulations are based on perfect LSB stuck-at-0 faults injected before the last MixColumns operation and the table entries denote the minimum number of total fault attempts that were needed to isolate the correct 4 key words corresponding to the targeted column of the state by resulting in the maximum SEI value with at least 80 % success rate. There is no implicit advantage of unmasked prime-field computations against SIFA-1 attacks, as reflected in the $d = 1$ column. However, once masking is applied and the security order increases, the complexity of the simulated attacks grows exponentially for AES-prime implementations, while it fluctuates only slightly and stays more or less constant for the respective AES implementations. This observation matches the SIFA-1 simulation results on toy circuits under the same attack shown in Figure 6.

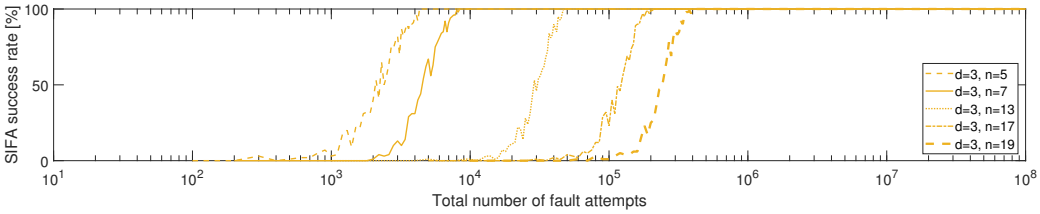
3.9 Additional Remarks on the Detection

In general, we consider attacks on the detection-based countermeasure, whose presence we assume in this work, as out of scope. This threat has been discussed in many previous

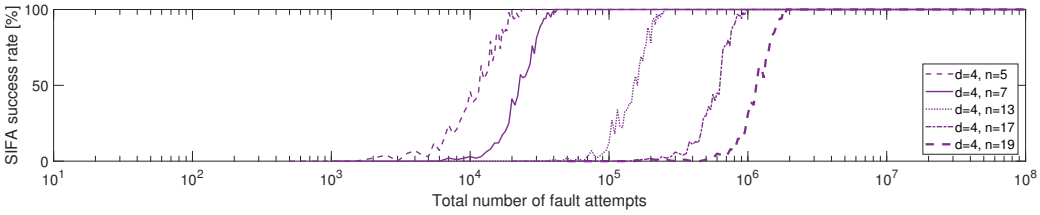
[6] Note that the analytical figures get difficult to calculate beyond 13-bit primes due to the extensive computation time required to multiply the polynomials. Also, for small primes, the estimations are not very meaningful due to the high statistical bias of the distributions. However, for attack simulations, we can work with larger primes, as well as smaller primes.



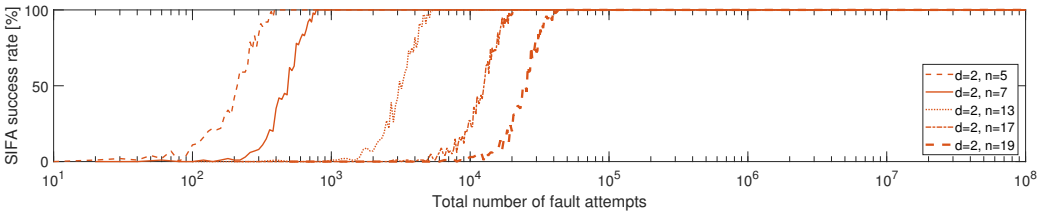
(a) LSB stuck-at-0 fault in 2 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.



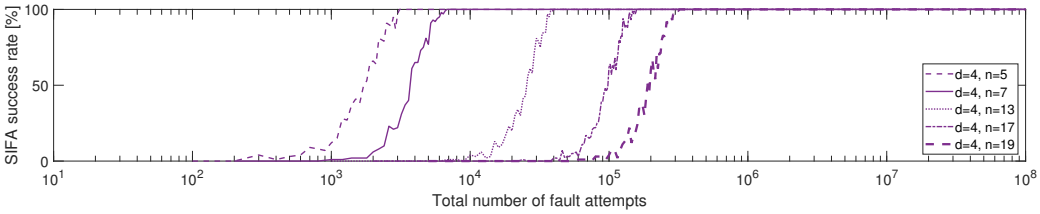
(b) LSB stuck-at-0 fault in 3 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.



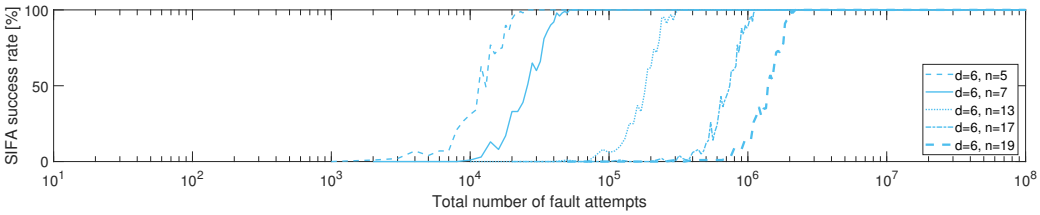
(c) LSB stuck-at-0 fault in 4 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.



(d) LSB toggle/bit-flip fault in 2 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.



(e) LSB toggle/bit-flip fault in 4 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.



(f) LSB toggle/bit-flip fault in 6 shares of prime-field encoding in $\mathbb{F}_{2^n-1}$.

Figure 13: Dependency of the number of fault attempts needed on the size of the Mersenne prime $p = 2^n - 1$.

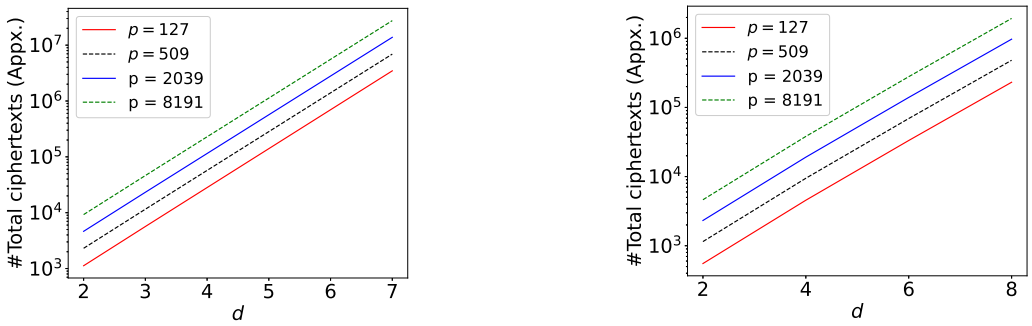


Figure 14: Analytical estimations for total ciphertext counts with CHI statistic: a) LSB-stuck-at-0 fault model; b) LSB-toggle/bit-flip fault model.

Table 1: Simulated SIFA-1 attack complexity using perfect LSB stuck-at-0 faults on unmasked ($d = 1$) and masked ($d > 1$) AES and AES-prime implementations measured in total number of fault attempts needed to achieve success rate 80%.

Target	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
AES	365	370	335	360	375
AES-prime	305	1 420	5 650	26 250	119 500

publications for many different types of detection-based schemes in detail [PM18]. As with most countermeasures, the storyline is that there is a tradeoff between the price a designer is willing to pay in terms of overhead and the maximum security that can be obtained for such a cost. In general, we share the belief that it is possible to instantiate such schemes in a manner which ensures that attacks on them will not be the weak point of the protected implementation in practice.

However, there is one side note we would like to mention, as it fits into the Boolean-vs-prime masking narrative. When the detection is realized as a simple duplication combined with a final comparator, then the most commonly suggested attack is injecting the same fault into each of the redundant copies or executions (for area or time redundancy, respectively) in order to fool the comparator into releasing a faulty output to the adversary. An attacker capable of deliberately flipping any bit at any location in the protected circuit with perfect accuracy may perform this attack with a 100 % success rate, even if each redundant copy or execution is Boolean masked with unique, freshly generated randomness. However, if additive prime-field masking is applied to each redundancy, also receiving independent freshly generated randomness unknown to the adversary, then the success rate is ≈ 50 % or less. This is due to the fact that any fault inserted into a single share always has at least 2 and at most $2^{\lfloor \log_2(p) \rfloor}$ different arithmetic impacts on the underlying variable[7]. For an LSB bit-flip for example, these impacts are $+1$ and -1. Generally, the number of possible impacts is determined by the number of flipped bits. In summary, the trivial attack on simple redundancy countermeasures suffers from an exponentially decreasing success rate in the number of redundancy domains, as the adversary has to hope that the same impact (e.g., $+1$ or -1) is caused in each of them by chance.

[7] Averaging for example 7.8, 17.2 and 194.6 different impacts when injecting random faults into data encoded in fields associated with Mersenne primes $p = 2^5 - 1$, $p = 2^7 - 1$ and $p = 2^{13} - 1$ respectively.

4 Boolean and Prime-Field Masked Operations vs. SIFA-2

It is evident from the analysis so far that masking in prime fields has certain advantages over Boolean masking against SIFA adversaries targeting the encodings. In this section, we analyze the security of these two masking paradigms when faced with attacks which exploit the structure and properties of the masked *operations* of cryptographic algorithms. Such attacks, also known as SIFA-2 [DEG+18, SJR+20], mostly utilize the fact that the propagation of faults through non-linear, non-bijective operations is conditional on the unmasked sensitive inputs to these operations[8]. Fault template attacks [SBR+20] also exploit such propagation properties. Since we can only distinguish between fault propagation or no fault propagation when the injection was at least temporarily effective, we will directly focus on bit-flip/toggle faults and random-value faults in this section. While bit-flips/toggles correspond to precise fault injections, random-value faults constitute a type of fault attack that imposes only very relaxed requirements on the fault injection capabilities of the adversary. Random-value faults replace the current value of the targeted word by an unknown random value and thus model an imprecise, potentially multi-bit, fault that is sometimes sufficient to perform structural fault attacks. Random-value faults are often called *random byte* faults in literature [BH22].

4.1 SIFA-2 on Boolean Masked Operations

It has been demonstrated on several occasions that operation-dependent fault propagation attacks can be fatal for Boolean masking. As shown in [DEG+18] (and also for FTA attacks [SBR+20]), it is sufficient to corrupt a single share of a masked implementation to make such structural attacks work. Unlike the case of encoding faults, bit-flip faults are also exploitable for operation-dependent attacks in the case of Boolean masking. Broadly speaking, such attacks can be performed in two ways.

The first one relies on corrupting the input of a non-bijective, non-linear masked operation such as an AND gadget or, more generally, a field multiplication gadget of arbitrary size. Binary ciphers for Boolean masking may employ Boolean operations at the bit-level, but also may use larger field operations at the word-level. Since binary fields of the type $\mathbb{F}_{2^n}$ generally have subfields down to $\mathbb{F}_2$, bit-level and word-level operations may not only be combined but even converted into and from each other. As an example, consider the heart of the AES S-box, namely inversion over $\mathbb{F}_{2^8}$. While this function may be computed by exponentiation of field elements, as $x^{-1} = x^{254}$, using multiplication chains over $\mathbb{F}_{2^8}$ [WM18], it can also be broken down into subfield operations over $\mathbb{F}_{2^2}$ and $\mathbb{F}_{2^4}$, such as done by the well-known Canright tower-field representation [Can05], or even down to bit-level operations such as done by the efficient Boyar-Peralta representation [BP12]. Accordingly, such a function may be masked in any of these fields, and the concrete choices made will affect the vulnerability to SIFA-2. The attack simply exploits the fact that a fault in one of the inputs of a multiplication (of any size) cannot propagate to the output of the gate if the other (unmasked) input is 0. Therefore, if one input to a multiplication operation is effectively faulted but still leads to an ineffective fault, i.e., a fault-free ciphertext, then this observation leaks the detail that the other (unmasked) multiplication input was 0. Since this property is inherent to multiplications as mathematical functions, it affects all correct implementations of a multiplier, regardless of the type of gadget that is chosen to realize it (e.g., low latency, low randomness, low area, ...), or the order of its masking.[9] However, the likelihood of one input being 0 obviously only depends on the size of the field ($\frac{1}{2^n}$ for $\mathbb{F}_{2^n}$, i.e., $\frac{1}{2}$ for an AND). Hence, the smaller the field, the more vulnerable to

[8] In fact, the value dependency of fault propagation is also true for bijective functions. However, they do not result in ineffective faults due to bijectivity. Such fault propagation can be exploited for a stronger adversary model combining SCA and FIA.

[9] Such an attack has been shown recently on a masked hardware implementation [HMA+24].

such attacks in general, highlighting the advantages of word-level over bit-level masking. This analysis obviously only applies for Boolean-masked circuits which do instantiate multiplications or similar standalone non-bijective functions and is not a general weakness of the encoding. Since the value of the injected fault is not of importance here, both bit-flip and random-value faults are theoretically applicable. However, in practice, a typical random-value / random byte fault might not lead to very effective attacks when the field size is so small that multiple masked multipliers are affected at the same time (consider for example a masked Boyar-Peralta representation of the AES Sbox). In that case, depending on the unknown random value injected, the attacker may not be able to determine which multiplication counterparts have been zero when the fault turns ineffective. This will increase the number of fault attempts required.

The second type of possible fault propagation exploitation is to fault inside a masked gadget. Note that faulting at the input of a bijective gadget does not result in a SIFA attack as, by the bijectivity, any change in the input must propagate to the output. However, the internals of masked bijective operations may not ensure such propagation guarantees. Therefore, depending on how the masked circuit has been constructed, one may inject faults inside the masked gadgets to create ineffective faults[10] [DEG+18]. Consequently, there exist design strategies to construct implementations, so that these attacks are prevented. One example of such a design strategy is presented in [DDE+20], where Toffoli gate-based structures have been utilized to ensure security against single fault injection. Another alternative strategy is to use error correction on each share and each masked non-linear operation [SJR+20]. With a $(2t+1)$-bit error-correcting code, such countermeasures ensure that every fault corrupting up to t bits gets corrected.

4.2 SIFA-2 on Prime-Field Masked Operations

In contrast to binary fields, prime fields cannot have any non-trivial subfields/subgroups. This means that non-linear functions over such fields also cannot be broken down into operations over smaller fields, which could then be masked individually. Instead, any non-linear function in a prime-field design is a polynomial over $\mathbb{F}_p$ and consequently can be computed by chains of non-linear squarings and multiplications combined with linear additions and subtractions in the field. Consider, for example, the AES-prime S-box $S(x) = x^5 + 2 \bmod p$, which is realized by two squarings and one multiplication in the constructions of [CMM+23]. Efficient masked squaring gadgets for prime fields have been developed in [CMM+23], while for multiplications many masked multipliers developed for binary fields can be reused as they are actually field agnostic (including ISW [FGP+18], DOM [GMK16], HPC1 [CGLS21], HPC3 [KM22]). In the following we evaluate the fault security of these masked operations. The goal is to understand the inherent resilience such gadgets provide against operation-dependent SIFA-2 due to their construction in the prime field. Such analysis also sheds some light on the feasibility of constructing SIFA secure gadgets over prime fields.

4.2.1 Squaring:

Given an input $x \in \mathbb{F}_p$, the squaring gadget performs the following operation $y = x^2 \bmod p$. First, we shall evaluate the resilience of the squaring operation against input faults. Next, the security will be evaluated with respect to concrete masked gadgets proposed in [CMM+23].

Gadget-independent Analysis: Being a non-bijective, non-linear operation, there always exist many-to-one mappings for squaring. In other words, y is identical for x and $-x$,

[10] Such attacks may also happen for any non-bijective gadgets depending on how the masking has been performed.

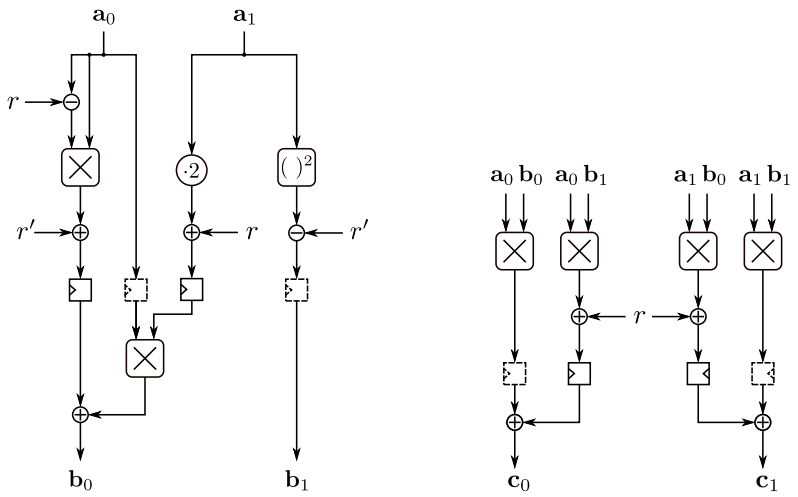
which are their respective mutual additive inverses. Let us now explain how this property leaks information. Consider a masked encoding of a variable s as $\{s_0, s_1, \cdots s_{d-1}\}$. Let us consider a fault fl which flips (resp. stuck-at) some specific bits in one s_i. The only condition on this fault is that it remains fixed (i.e., flips the same bits) in all executions (i.e., repeatability of the fault matters). Without loss of generality, let us assume that fl flips the LSB of s_i in all executions. Now we observe that, depending on the values assumed by s_i, fl will result in an additive error $\delta^j = \{+1, -1\}$ at the j-th execution, i.e., the faulty state $s' = s + \delta^j \bmod p$. Therefore, $\exists j, s' = s \bmod p$. For the LSB fault, this implies either $-s = (s-1) \bmod p$, or $-s = (s+1) \bmod p$. Finally, $s = \frac{(kP \pm 1)}{2} \bmod p$ for $k \geq 0$. Therefore, the fault becomes ineffective only for specific values of s and thus can be exploited to perform a SIFA-2. Similar to the Boolean scenario described above, the attack works even while a single bit in a single share is corrupted. In particular, for the special case of Mersenne primes $p = 2^n - 1$, it holds that $s = 2^{n-1}$ and $-s = 2^{n-1} - 1$ are the only mutual additive inverses (hence squaring to the same y, causing an ineffective fault) that can be reached with a single LSB flip in a single share. Yet, it is important to note that to observe this information leakage, fl must be maintained consistently with high probability. Hence, this attack is not applicable for random-value faults. Finally, even with a consistent fl, the probability of obtaining this information leakage (i.e., an ineffective fault) decreases in p. In the example of a single-bit single-share fault against a Mersenne-prime-field masked squaring operation, the probability is $\frac{1}{p}$. Accordingly, the choice of p in a prime-field scheme is an important factor that can make SIFA-2 attacks prohibitive in terms of the total number of fault attempts needed for successful key recovery. Mersenne primes up to $p = 2^{31} - 1$ have been suggested for implementation efficiency in [MMMS23], allowing a SIFA-2 attacker with perfect repeatability of LSB faults on the squaring input to obtain an informative fault every 2 147 483 647 successful fault injections. We believe this to be beyond practical feasibility (even when ignoring that the standard offline computational attack effort additionally requires the iteration over two 31-bit values simultaneously, which borders impracticality in terms of computational effort). Hence, a good rule of thumb is once again that larger primes are beneficial for fault security (as well as it can be for SCA security [MMMS23]).

Gadget-dependent Analysis: Next, we analyze the composable gadgets proposed in [CMM+23] with respect to a single faulting adversary corrupting only one wire or operation (addition, subtraction, or multiplication) inside the gadget computation. Our goal, in this case, is only to evaluate whether or not an adversary as powerful as considered in the aforementioned input fault attack can perform stronger attacks by faulting the internals of the gadgets. As we argue next, this is not possible for the 2-share gadget proposed in [CMM+23]. The same is true for the 3-share gadget proposed there (we omit this analysis here).

Given a fault injected at an internal wire/operation of a gadget, there can be two ways in which the fault may become exploitable:

1. The fault propagates to all the output shares making their combination statistically biased depending on the unmasked value. We note that fault propagation to all the shares may happen for a non-linear gadget as different share domains interact with each other.

2. Even though the fault does not propagate to all the output shares, the propagation of the fault may still be conditioned on all the input shares, thereby making the correctness of the gadget conditioned on the unshared input. This typically happens when a faulted variable is multiplied by all the shares of an input variable. Since multiplication does not propagate a fault if the fault-free input is 0, the ineffectiveness of a fault implies that all the shares of the input variable are 0. Therefore, it leaks the unshared value. Depending on the share compression operation in non-linear



(a) 2-share prime-field SQ [CMM+23] (b) 2-share DOM [GMK16]

Figure 15: Masked squaring (left) and multiplier (right) with two shares.

gadgets, it may also happen that even if some multiplications propagate the fault, it gets canceled during the addition/subtraction at the share compression step. Overall, whether or not the fault corrupts the unshared output depends on the correctness properties of the fault propagation path.

Based on the two conditions mentioned above, we first consider the 2-share gadget from [CMM+23] (ref. Figure 15(a)). It can be observed that the only fault that propagates to both output shares is an input fault at a_1. No other (e.g., gadget-specific) input fault propagates to both the output shares, except a fault in randomness r'. However, faulting the randomness does not have any effect on the correctness of the unmasked state. Therefore, it is not possible to create a biased distribution at the gadget output by corrupting all the shares. Next, we evaluate if the fault propagation to the gadget output can be made dependent on some input shares based on an internal fault. To evaluate this, we first observe that all operations before the register stage happen individually on each share. Therefore, a fault injection can (at most) extract the value of a single share at this stage. The interaction between shares happens after the register stage at the multiplication operation. Any fault at the input or inside of the multiplication operation can, at most, leak information about the share a_0. Note that no information about a_1 can leak here as this share is already blinded with randomness. The same is true for the final addition operation generating b_0, as both of its inputs are blinded. Overall, we conclude that the strongest attack, in this case, is still the gadget-independent input fault, and no stronger attacks are feasible with a similarly powerful adversary by faulting inside the gadget. Similar arguments can be derived for the 3-share gadget described in [CMM+23].

4.2.2 Multiplication:

Similar to the analysis of Boolean masked operations above, multiplication is also an important non-bijective non-linear operation in prime fields. For the AES-prime S-box constructions in [CMM+23], a gadget based on Domain-Oriented Masking (DOM) [GMK16] is used. Such a DOM multiplier (called DOM-indep) is depicted in Figure 15(b). We use this multiplication gadget for our analysis and note that commonly used variants needing two cycles and providing stronger composability properties, namely two-cycle ISW [FGP+18] (satisfying Strong Non-Interference [BBD+16]) and HPC1 [CGLS21] (satisfying Probe Isolating Non-Interference [CS20]) would lead to the same result.

Gadget-independent Analysis: As detailed above, any fault at one input of the multiplication gadget would not propagate if the other (unmasked) input is 0. Unlike the attack mentioned for the squaring gadget, however, the fault can be different for each execution and its value is not of importance, leading to very relaxed requirements on the fault injection capabilities of the adversary. Hence, both bit-flip and random-value faults are applicable in this scenario. As also mentioned before, the probability of obtaining an ineffective fault is $\frac{1}{p}$, and can be made arbitrarily low by choosing a large prime (at potential implementation overheads).

Gadget-dependent Analysis: Consider the first-order DOM gadget depicted in Figure 15(b). The partial multiplications occur at the beginning. Faulting any one of them only leaks one share of each variable. Also, such a fault can propagate to only one output share. The operations following the multiplication are linear. The only variable that can affect both share domains is the randomness. However, faulting the randomness has no effect on the correctness of the unmasked state and, therefore, such a fault cannot leak information by ineffectiveness. The final addition operations in each share domain can, at most, leak information about shares in the same domain and nothing beyond that. Therefore, faulting the internals of such a DOM gadget also does not result in a stronger attacker than the input faulting adversary. These arguments also apply to DOM gadgets with higher masking order.

4.2.3 Bijective Operations:

Instead of constructing bijective masked non-linear functions in prime fields by combining separate standalone non-bijective squarings and multiplications, it is also possible to directly share a bijective operation. Consider, for example, the function $S(x) = x^5 \bmod 2^7 - 1$. A gadget with two input shares s_0 and s_1 would need to compute $(s_0 + s_1)^5 = s_0^5 + 5 \cdot s_0^4 \cdot s_1 + 10 \cdot s_0^3 \cdot s_1^2 + 10 \cdot s_0^2 \cdot s_1^3 + 5 \cdot s_0 \cdot s_1^4 + s_1^5 \bmod p$. Due to the bijective nature of the function, no input fault could lead to an ineffective fault, preventing gadget-independent SIFA-2. However, the gadget structure itself would need to ensure that no internal fault results in a successful attack. As we have already shown for non-bijective gadgets, maintaining such a guarantee is feasible for prime-field masking, requiring a suitable combination of randomness distribution and compression function. Ensuring this for directly shared bijective operations is non-trivial, especially for larger numbers of shares, and left as an interesting direction for future research.

4.3 Gadget-independent SIFA-2 Simulation Results

We now simulate an exemplary gadget-independent SIFA-2 exploiting the non-bijectivity of squaring and multiplication for different prime sizes. A specific instance of the toy circuit using $S(x) = x^5 \bmod p$, implemented by two squarings and one multiplication, is shown in Figure 16. Regardless of Boolean or prime-field masking, any effective fault in the area (both bit-flip and random-value faults apply) marked in red leads to an ineffective fault with probability $\frac{1}{\text{field size}}$. Since the final multiplication result will be 0, the adversary directly observes $c_0 = k_0$ and no statistical analysis is needed. However, let us now assume that the fault is instead injected one round earlier, as denoted in Figure 17. Here, the attacker may exploit the non-bijectivity of both the squaring and the multiplication. Let the field be $\mathbb{F}_{2^7-1}$, then the following set of S-box inputs x can lead to ineffective faults when injecting an LSB bit-flip in a single share.

- $x = 0$, when the fault at the squaring input is effective. Probability is $\frac{1}{p} = \frac{1}{127}$.
- $x = 8$ with $x^2 \bmod p = 64$, when the LSB bit-flip has an effect of -1 on the sharing, since $64^2 \bmod p = 63^2 \bmod p$. Probability is $\frac{1}{2} \cdot \frac{1}{p} = \frac{1}{254}$.

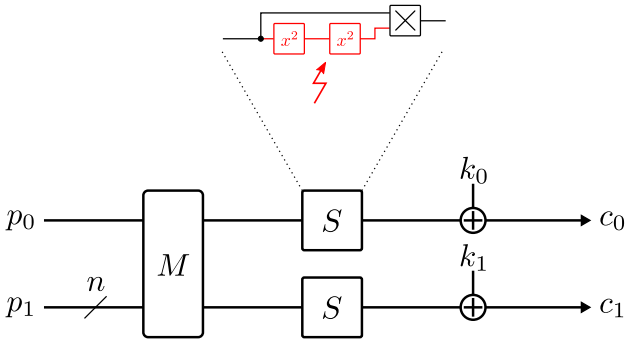


Figure 16: Toy circuit from Figure 4 with $S(x) = x^5$ implemented as two squarings and one multiplication. Fault location chosen to perform an ineffective fault attack on the multiplication (exploiting the fact that multiplication with 0 prevents fault propagation).

- $x = 119$ with $x^2 \bmod p = 64$, when the LSB bit-flip has an effect of -1 on the sharing, since $64^2 \bmod p = 63^2 \bmod p$. Probability is $\frac{1}{2} \cdot \frac{1}{p} = \frac{1}{254}$.

This example is, of course, limited to $\mathbb{F}_p$ with $p = 2^7 - 1$. To obtain the results depicted in Figure 18 we have performed equivalent attacks for different Mersenne primes, namely $p = 2^5 - 1 = 31$, $p = 2^7 - 1 = 127$, $p = 2^{13} - 1 = 8191$, $p = 2^{17} - 1 = 131071$, $p = 2^{19} - 1 = 524287$. All results are simulated for masked circuits with 2 shares, but the attack is not at all affected by increasing the number of shares d. As in previous sections, we have also considered adversaries which have a less-than-perfect injection probability (here, the same as the biasing success, as only 1 bit is faulted), namely 80 % and 60 %. It can be observed that for larger primes, the total number of fault attempts needed increases drastically, as expected. Between smallest and largest prime, the difference is more than 3 orders of magnitude for the perfect attacker. For the imprecise adversary, the complexities of attacks on the 17- and 19-bit primes even become too large to simulate in a reasonable time. As described, SIFA-2 on multiplications can be performed even with highly imprecise fault injections, while SIFA-2 on squarings requires more precision in practice (random-value faults do not apply). Hence, it could be interesting to abstain from using multiplication altogether, to only use squarings instead (preferably in large fields). This may not allow to compute bijective power maps in prime fields, but can be used in Feistel-like constructions to achieve multi-branch bijective operations as shown in Figure 19. While such a structure indeed forces adversaries to inject faults with high precision (i.e., bit-flips), any attacker who reliably and repeatedly toggles a bit at the marked location still requires a similar complexity to Figure 18 with the difference that 4 key parts need to be guessed at once.

We conclude that increasing the prime size is a very effective tool to make SIFA-2 attacks on prime-field masked circuits prohibitive. The same is true for increasing the field size of Boolean masked circuits when considering gadget-independent SIFA-2 on multiplications (hence, making word-level masking much more attractive than bit-level masking in this context), but not necessarily for gadget-dependent SIFA-2. The latter can be more difficult to achieve for binary field operation due to the existence of subfields, which could theoretically reduce the amplification in the field size to an amplification in a subfield size only. Prime masking can, by definition, not fall into this trap.

5 Discussion and Related Work

In this section, we discuss points which are relevant for positioning our observations on prime-field masking with respect to the state of the art. The background section already

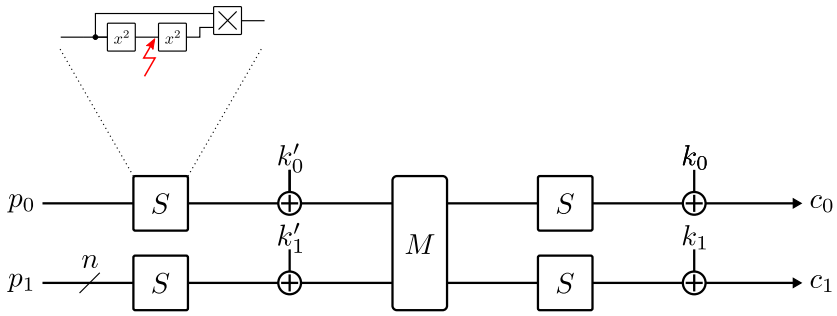
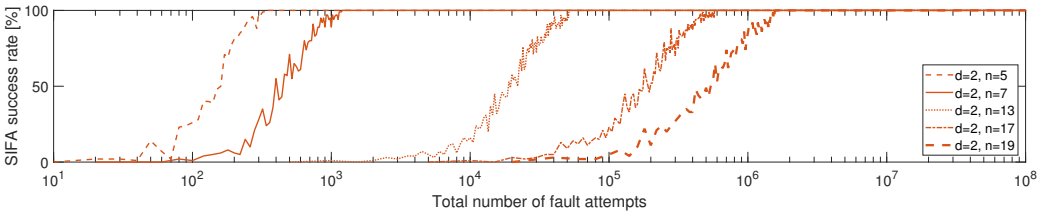
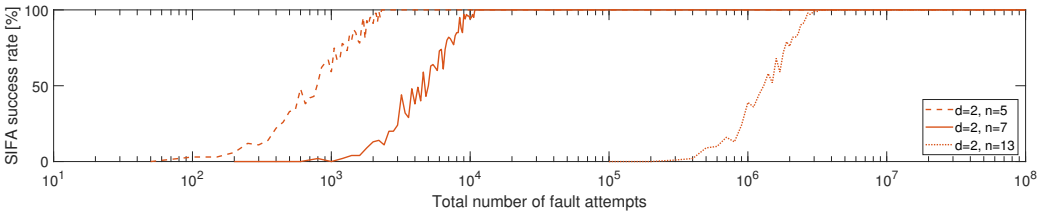


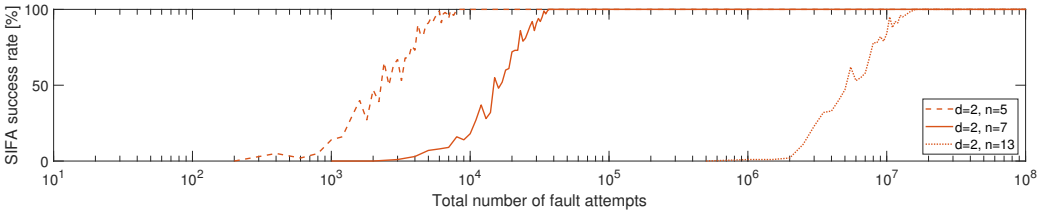
Figure 17: Two-round version of the toy circuit from Figure 4 with $S(x) = x^5$ implemented as two squarings and one multiplication. Fault location chosen to perform a SIFA-2 on the non-bijectivity of the squaring and multiplication.



(a) LSB toggle/bit-flip with 100 % injection probability in one share of the encoding.



(b) LSB toggle/bit-flip with 80 % injection probability in one share of the encoding.



(c) LSB toggle/bit-flip with 60 % injection probability in one share of the encoding.

Figure 18: Results of a SIFA-2 attack on the non-bijectivity of squaring and multiplication gadgets, for different sizes of Mersenne primes $p = 2^n - 1$.

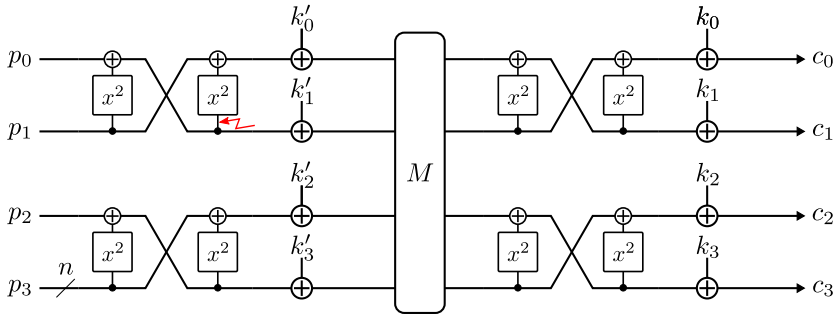


Figure 19: Two-round toy circuit for prime-field masking which uses only squaring as source of non-linearity. Fault location chosen to perform a SIFA-2 on the non-bijectivity of the squaring.

provides the bigger picture with respect to SCA and FIA. Here we shall specifically focus on works proposing SIFA countermeasures. Next, we briefly discuss the combined SCA-FIA attacks and how prime-field masking performs under this threat model. A summary of the already known security benefits of prime-field masking can be found in Appendix A.1.

5.1 Related Work

There have been several efforts to prevent ineffective fault attacks in the recent past. Most of these schemes employ some form of masking. Many of them have already been referenced in previous sections. However, here we provide a comprehensive summary. Primarily, there exist two approaches to be combined with masking: 1) Replicating each share and performing error correction [SJR+20]; the granularity of error correction is usually set at the level of each bijective function. 2) Performing the masking based on permutations as building blocks, such that every fault propagates to the output [DDE+20, DOT24]. Error detection on the ciphertexts is then sufficient for this approach. A relatively less explored third approach presented in [SRM20] proposes the use of error correction only (without masking), combined with a carefully crafted circuit design strategy against faults initially proposed under the name Impeccable Circuits [AMR+20]. Finally, there exist combined countermeasures, which also employ masking and provide security against simultaneous fault and side-channel attacks [IPSW06, RMB+18, DN20, FGM+23] (discussed in the next subsection).

Our proposal differs from the state of the art in the sense that most previous masking-based approaches tolerate only up to a small number of faulted bits per encoded variable. More precisely, they are only secure until all the shares are affected by faults. There is often no protection whatsoever against adversaries able to inject such fault patterns. Other approaches such as presented in [RMB+18, MAN+19] can indeed handle an unbounded number of faults, however their adversary models do not allow precise fault injections. We show that the discovered properties of prime-field masking can break both these barriers, and can still provide security increasing with the masking order even in the presence of multiple precisely faulted bits in each share. From a structural point of view, although our proposal also requires redundant computation, we do not generally need fine-grained error correction. Rather, for structural SIFA attacks (i.e., SIFA-2), we suggest exploiting the prime field size to increase the fault requirements significantly. However, our approach may indeed benefit further from employing ideas from the second major SIFA protection approach mentioned above.

Another line of work which uses masking against faults is based on code-based masking [CG18, WCG+22]. The rich forms of encoding strategies covered by code-based masking show conceptual possibilities to handle faults more efficiently. One instance of

such code-based masking for which the fault resilience has been examined is polynomial masking [SFES18, BEF+23]. In polynomial masking, the encoding is not additive but is based on Shamir's secret sharing [Sha79b]. The main idea here is to encode the secret as the coefficients of a polynomial to enable masking. In order to enable fault resilience, some extra coefficients (called error coefficients) are added to the polynomials. In the presence of a fault, the polynomials cannot be decoded successfully and result in randomized values. We note that the redundancy, in this case, is employed in the form of error coefficients and, therefore, does not need duplication of the same computation. However, while such approaches may have promising asymptotic complexities [RP12], the practical overheads for common levels of protection, especially in hardware, are tremendous for now.

In [SFES18, BEF+23], secure gadget constructions have also been shown for polynomial masking which withstand attacks like SIFA-2. However, in general, fault resilience in code-based masking has been explored for encoding only, and it is not known (except for the cases of polynomial masking mentioned above) how the gadget computations will survive faults. In fact, existing SCA-secure gadgets, like the one in [WMCS20], need to use Boolean masking for some intermediate computations. Therefore, they lose all good properties against SIFA-1. Overall, while including more structure in the encoding seems promising, the general security against faults is still largely unexplored for this class of masking. Compared to polynomial masking, prime masking is still additive and, hence, much more affordable than most solutions discussed in this section. Still, as shown in this paper, it provides significant practical gains against fault attacks.

5.2 Combined SCA and FIA Attacks

Combined attacks inherently come into context when we discuss the utility of masking for fault attacks. Such attacks have been known for years [AVFM07, CFGR10, IPSW06, Rea11], and very recently it has been shown that they can be fatal even for implementations employing masking-based SIFA countermeasures [SBJ+21, SRJB23]. The main idea behind breaking masking-based SIFA countermeasures is that the detection/correction logic may leak information (via side-channel leakage) resulting from the propagation of faults through non-linear gadgets. In other words, such attacks are typically operation-dependent/gadget-dependent. However, both bijective and non-bijective operations can be vulnerable. Also, faulting the intra-gadget randomnesses, which will not have any effect on correctness in most cases, can cause exploitable leakage in combined attacks [RFSG22].

Existing countermeasures for combined attacks are either based on Boolean masking [RMB+18, DN20, FGM+23], or polynomial masking [BEF+23]. For the former class, security is mainly achieved through careful, intra-gadget error correction [DN20, FGM+23], or by exploiting some costly extra computation [RMB+18]. The security is achieved from the structure of the gadgets, assuming a bounded fault model similar to the existing SIFA countermeasures. Therefore, the features we discover for prime masking in this paper would still benefit such gadget constructions. For example, the most recent combined-secure gadget due to [FGM+23] can be constructed for prime-field masking too, and is expected to benefit from its robust encoding. Polynomial masking, as already pointed out in Section 5.1, can be used to obtain combined security. Whether a combination with prime masking is worthwhile is currently unclear. As a standalone construction, prime masking is not expected to provide any improved security guarantees against combined attacks, at least in the existing adversary models, where it is assumed that the leakage is clearly observable just after the fault injection. In practice, an attack of course depends on how clearly such leakage is available. Prime masking may be advantageous in more practical situations, for example if there is some distance between injected fault and probed value, due to its natural resilience to even strong SCA and SIFA-1 adversaries. Whether such an adversarial limitation can be assumed without sacrificing security is an important topic of future research.

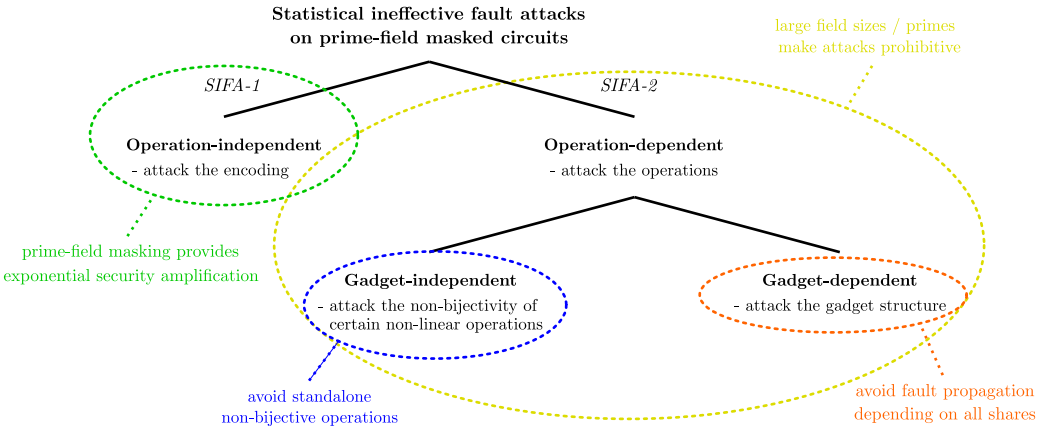


Figure 20: Schematic overview of our analysis results with respect to prime-field masking vs. SIFA.

6 Conclusions

Masking in fields of prime order has recently been suggested as a promising technique to protect cryptographic implementations against side-channel analysis adversaries. Its main advantage is that it may protect sensitive information even in presence of strong passive adversaries capable of making side-channel observations with low noise influence. In this work, we complete the picture and analyze the resistance that additive prime-field masking provides against strong active adversaries capable of injecting an almost arbitrary number of perfect precision faults, or alternatively, an arbitrary number of almost perfect precision faults. We discover a lot of interesting properties that have not been reported in literature before and conclude that prime masking is indeed a powerful ingredient to build physically secure implementations of cryptographic algorithms in the future.

While protection against differential fault analysis, especially structural attacks, still requires a detection-based countermeasure (to keep faulty outputs inaccessible), the advantages of prime-field masking start to emerge when shifting the focus to attacks that can remain feasible in such a setting, namely Statistical Ineffective Fault Attacks (SIFA) and their variants.

Our analysis results are schematically portrayed and summarized in Figure 20. Generally, SIFA on masked implementations can be categorized into generic attacks which target the encoded data directly (SIFA-1) and specific attacks which target the characteristics of certain operations present in the target algorithm (SIFA-2). The main result that we establish in this work is that SIFA-1 can generally be prevented by prime-field masking in the sense that the total number of fault attempts an attacker requires to break the scheme increases exponentially in the number of shares. This holds true even in the presence of unrealistically strong adversaries who may inject almost arbitrary numbers of stuck-at-0, stuck-at-1 or toggle/bit-flip faults with arbitrary precision. Boolean masking provides no security in such a model, as attacks are as trivial as on unprotected implementations. Even when restricting the precision of the adversary, a setting where masking is typically said to be beneficial against SIFA, prime-field encodings still provide orders of magnitude better security for the same number of shares than Boolean ones.

We also analyze the effectiveness of SIFA-2 on prime-field masked circuits. While general weaknesses inherent to the nature of mathematical operations, such as the non-bijectivity of multiplications and squarings, naturally persist regardless of the encoding, we demonstrate that the tendency of using larger fields in prime masking, in contrast to

the tendency of sharing bit-level non-linear operations (such as AND gates) in Boolean masking, can help to significantly frustrate attacks. The rule of thumb is similar to what has been discovered regarding SCA protection, namely the larger the prime, the better. Interestingly, and perhaps counterintuitively, those previous investigations of the SCA security showed that larger primes generally help, except for leakage models like noise-free LSB [MMMS23], while in this work we demonstrate that even in presence of perfect LSB faults, the security gains from increasing the prime size are significant.

An alternative avenue to thwart SIFA-2 without increasing the field size, is by avoiding any standalone non-bijective operations, and instead sharing bijective operations (such as appropriate power maps) directly. When attempting this, however, attention needs to be paid to such a gadget's internals in order to not provide any attack surface to adversaries that try to exploit fault propagation conditioned on secret values.

We have analyzed our discoveries mathematically and provide new formulas to estimate the success of attacks (SIFA-1). Furthermore, we have substantiated our theoretical analysis by a large amount of simulation results, originating from multiple tens of thousands of CPU hours computation time. We believe our results can assist and advance the development of new cryptographic algorithms specifically designed to lead to physically secure implementations in the future.

Acknowledgments

François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the ERC Advanced Grant 101096871 (acronym BRIDGE) and the Horizon Europe project 1010706275 (acronym REWIRE). Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symmetric Cryptol.*, 2020(3):1–45, 2020.

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for mpc, and more. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part II*, volume 11736 of *Lecture Notes in Computer Science*, pages 151–171. Springer, 2019.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, 2016.

[AMR+20] Anita Aghaie, Amir Moradi, Shahram Rasoolzadeh, Aein Rezaei Shahmirzadi, Falk Schellenberg, and Tobias Schneider. Impeccable circuits. *IEEE Trans. Computers*, 69(3):361–376, 2020.

[AVFM07] Frédéric Amiel, Karine Villegas, Benoit Feix, and Louis Marcel. Passive and active combined attacks: Combining fault attacks and side channel analysis. In *Fourth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2007, FDTC 2007: Vienna, Austria, 10 September 2007*, pages 92–102, 2007.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions: ttanemoi permutations and ttjive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539. Springer, 2023.

[BBD+16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 116–129. ACM, 2016.

[BBM+22] Timo Bartkewitz, Sven Bettendorf, Thorben Moos, Amir Moradi, and Falk Schellenberg. Beware of insufficient redundancy an experimental evaluation of code-based FI countermeasures. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(3):438–462, 2022.

[BCN+06] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The sorcerer's apprentice guide to fault attacks. *Proc. IEEE*, 94(2):370–382, 2006.

[BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2016.

[BDF+17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.

[BDK+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367. IEEE, 2018.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter

Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[BEF+23] Sebastian Berndt, Thomas Eisenbarth, Sebastian Faust, Marc Gourjon, Maximilian Orlt, and Okan Seker. Combined fault and leakage resilience: Composability, constructions and compiler. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 377–409. Springer, 2023.

[BH08] Arnaud Boscher and Helena Handschuh. Masking does not protect against differential fault attacks. In Luca Breveglieri, Shay Gueron, Israel Koren, David Naccache, and Jean-Pierre Seifert, editors, *Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008*, pages 35–40. IEEE Computer Society, 2008.

[BH22] Jakub Breier and Xiaolu Hou. How practical are fault injection attacks, really? *IEEE Access*, 10:113122–113130, 2022.

[BP12] Joan Boyar and René Peralta. A small depth-16 circuit for the AES s-box. In Dimitris Gritzalis, Steven Furnell, and Marianthi Theoharidou, editors, *Information Security and Privacy Research - 27th IFIP TC 11 Information Security and Privacy Conference, SEC 2012, Heraklion, Crete, Greece, June 4-6, 2012. Proceedings*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 287–298. Springer, 2012.

[BR23] Sonia Belaïd and Matthieu Rivain. High order side-channel security for elliptic-curve implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(1):238–276, 2023.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[BS21] Olivier Bronchain and François-Xavier Standaert. Breaking masked implementations with many shares on 32-bit software platforms or when the security order does not matter. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):202–234, 2021.

[Can05] David Canright. A very compact s-box for AES. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 441–455. Springer, 2005.

[CFGR10] Christophe Clavier, Benoit Feix, Georges Gagnerot, and Mylène Roussellet. Passive and active combined attacks on aes: Combining fault attacks and side channel analysis. In *2010 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2010, Santa Barbara, California, USA, 21 August 2010*, pages 10–19, 2010.

[CG00] Jean-Sébastien Coron and Louis Goubin. On boolean and arithmetic masking against differential power analysis. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 231–237. Springer, 2000.

[CG18] Claude Carlet and Sylvain Guilley. Statistical properties of side-channel and fault injection attacks using coding theory. *Cryptogr. Commun.*, 10(5):909–933, 2018.

[CGLS21] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware private circuits: From trivial composition to full verification. *IEEE Trans. Computers*, 70(10):1677–1690, 2021.

[CHK+21] Jihoon Cho, Jincheol Ha, Seongkwang Kim, ByeongHak Lee, Joohee Lee, Jooyoung Lee, Dukjae Moon, and Hyojin Yoon. Transciphering framework for approximate homomorphic encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III*, volume 13092 of *Lecture Notes in Computer Science*, pages 640–669. Springer, 2021.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[CMM+23] Gaëtan Cassiers, Loïc Masure, Charles Momin, Thorben Moos, and François-Xavier Standaert. Prime-field masking in hardware and its soundness against low-noise SCA attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):482–518, 2023.

[CS20] Gaëtan Cassiers and François-Xavier Standaert. Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Trans. Inf. Forensics Secur.*, 15:2542–2555, 2020.

[DDE+20] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. Protecting against statistical ineffective fault attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):508–543, 2020.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 423–440. Springer, 2014.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Stefan Mangard, Florian Mendel, and Robert Primas. Statistical ineffective fault attacks on masked AES with fault countermeasures. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information*

Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II, volume 11273 of *Lecture Notes in Computer Science*, pages 315–342. Springer, 2018.

[DEK+18] Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Stefan Mangard, Florian Mendel, and Robert Primas. SIFA: exploiting ineffective fault inductions on symmetric cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):547–572, 2018.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.*, 34(3):33, 2021.

[DFS16] Stefan Dziembowski, Sebastian Faust, and Maciej Skórski. Optimal amplification of noisy leakages. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 291–318. Springer, 2016.

[DFS19] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. *J. Cryptol.*, 32(4):1263–1297, 2019.

[DGGK21] Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion: Symmetric encryption based on toffoli-gates over large finite fields. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 3–34. Springer, 2021.

[DGH+23] Christoph Dobraunig, Lorenzo Grassi, Lukas Helminger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Pasta: A case for hybrid homomorphic encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):30–73, 2023.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[DN20] Siemen Dhooghe and Svetla Nikova. My gadget just cares for me - how NINA can prove security against combined attacks. In Stanislaw Jarecki, editor, *Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24-28, 2020, Proceedings*, volume 12006 of *Lecture Notes in Computer Science*, pages 35–55. Springer, 2020.

[DN23] Siemen Dhooghe and Svetla Nikova. The random fault model. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *Selected Areas in Cryptography - SAC 2023 - 30th International Conference, Fredericton, Canada, August 14-18, 2023, Revised Selected Papers*, volume 14201 of *Lecture Notes in Computer Science*, pages 191–212. Springer, 2023.

[DOT24] Siemen Dhooghe, Artemii Ovchinnikov, and Dilara Toprakhisar. Stati: Protecting against fault attacks using stable threshold implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(1):229–263, 2024.

[FGM+23] Jakob Feldtkeller, Tim Güneysu, Thorben Moos, Jan Richter-Brockmann, Sayandeep Saha, Pascal Sasdrich, and François-Xavier Standaert. Combined private circuits - combined security refurbished. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 990–1004. ACM, 2023.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[FJLT13] Thomas Fuhr, Éliane Jaulmes, Victor Lomné, and Adrian Thillard. Fault attacks on AES with faulty ciphertexts only. In Wieland Fischer and Jörn-Marc Schmidt, editors, *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, pages 108–118. IEEE Computer Society, 2013.

[GHR+23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst meets fluid-spn: Griffin for zero-knowledge applications. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 573–606. Springer, 2023.

[GKL+22] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Reinforced concrete: A fast hash function for verifiable computation. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1323–1335. ACM, 2022.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 519–535. USENIX Association, 2021.

[GKS23] Lorenzo Grassi, Dmitry Khovratovich, and Markus Schofnegger. Poseidon2: A faster version of the poseidon hash function. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*, volume 14064 of *Lecture Notes in Computer Science*, pages 177–203. Springer, 2023.

[GLR+20] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 674–704. Springer, 2020.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[GOPS22] Lorenzo Grassi, Silvia Onofri, Marco Pedicini, and Luca Sozzi. Invertible quadratic non-linear layers for mpc-/fhe-/zk-friendly schemes over fnp application to poseidon. *IACR Trans. Symmetric Cryptol.*, 2022(3):20–72, 2022.

[GØSW23] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From farfalle to megafono via ciminion: The PRF hydra for MPC applications. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 255–286. Springer, 2023.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.

[GST12] Benedikt Gierlichs, Jörn-Marc Schmidt, and Michael Tunstall. Infective computation and dummy rounds: Fault protection for block ciphers without check-before-output. In Alejandro Hevia and Gregory Neven, editors, *Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings*, volume 7533 of *Lecture Notes in Computer Science*, pages 305–321. Springer, 2012.

[HMA+24] Haruka Hirata, Daiki Miyahara, Victor Arribas, Yang Li, Noriyuki Miura, Svetla Nikova, and Kazuo Sakiyama. All you need is fault: Zero-value attacks on AES and a new λ-detection m&m. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(1):133–156, 2024.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David A. Wagner. Private circuits II: keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, 2006.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KM22] David Knichel and Amir Moradi. Low-latency hardware private circuits. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1799–1812. ACM, 2022.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[MAN+19] Lauren De Meyer, Victor Arribas, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. M&m: Masks and macs against physical attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(1):25–50, 2019.

[MMMS23] Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Effective and efficient masking with low noise using small-mersenne-prime ciphers. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2023.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[NRS08] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of non-linear functions in the presence of glitches. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, volume 5461 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2008.

[oSN17] National Institute of Standards and Technology (NIST). Lightweight cryptography. 2017.

[PCM17] Sikhar Patranabis, Abhishek Chakraborty, and Debdeep Mukhopadhyay. Fault tolerant infective countermeasure for AES. *J. Hardw. Syst. Secur.*, 1(1):3–17, 2017.

[PM18] Sikhar Patranabis and Debdeep Mukhopadhyay. *Fault tolerant architectures for cryptography and hardware security.* Springer, 2018.

[PQ03] Gilles Piret and Jean-Jacques Quisquater. A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2003.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.

[Rea11] Thomas Roche and et. al. Combined fault and side-channel attack on protected implementations of aes. In *CARDIS*, pages 65–83. Springer, 2011.

[RFSG22] Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, and Tim Güneysu. VERICA - verification of combined attacks automated formal verification of security against simultaneous information leakage and tampering. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(4):255–284, 2022.

[RMB+18] Oscar Reparaz, Lauren De Meyer, Begül Bilgin, Victor Arribas, Svetla Nikova, Ventzislav Nikov, and Nigel P. Smart. CAPA: the spirit of beaver against physical attacks. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2018.

[RP12] Thomas Roche and Emmanuel Prouff. Higher-order glitch free implementation of the AES using secure multi-party computation protocols - extended version. *J. Cryptogr. Eng.*, 2(2):111–127, 2012.

[RSG23] Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. Revisiting fault adversary models - hardware faults in theory and practice. *IEEE Trans. Computers*, 72(2):572–585, 2023.

[RSM21] Shahram Rasoolzadeh, Aein Rezaei Shahmirzadi, and Amir Moradi. Impeccable circuits III. In *IEEE International Test Conference, ITC 2021, Anaheim, CA, USA, October 10-15, 2021*, pages 163–169. IEEE, 2021.

[SAD20] Alan Szepieniec, Tomer Ashur, and Siemen Dhooghe. Rescue-prime: a standard specification (sok). *IACR Cryptol. ePrint Arch.*, page 1143, 2020.

[SBJ+21] Sayandeep Saha, Arnab Bag, Dirmanto Jap, Debdeep Mukhopadhyay, and Shivam Bhasin. Divided we stand, united we fall: Security analysis of some SCA+SIFA countermeasures against sca-enhanced fault template attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II*, volume 13091 of *Lecture Notes in Computer Science*, pages 62–94. Springer, 2021.

[SBR+20] Sayandeep Saha, Arnab Bag, Debapriya Basu Roy, Sikhar Patranabis, and Debdeep Mukhopadhyay. Fault template attacks on block ciphers exploiting fault propagation. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 612–643. Springer, 2020.

[SFES18] Okan Seker, Abraham Fernandez-Rubio, Thomas Eisenbarth, and Rainer Steinwandt. Extending glitch-free multiparty protocols to resist fault injection attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):394–430, 2018.

[Sha79a] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sha79b] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[SJR+20] Sayandeep Saha, Dirmanto Jap, Debapriya Basu Roy, Avik Chakraborty, Shivam Bhasin, and Debdeep Mukhopadhyay. A framework to counter statistical ineffective fault analysis of block ciphers using domain transformation and error correction. *IEEE Trans. Inf. Forensics Secur.*, 15:1905–1919, 2020.

[SM20] Aein Rezaei Shahmirzadi and Amir Moradi. Clock glitch versus SIFA. In Luigi Dilillo, Mihalis Psarakis, and Taniya Siddiqua, editors, *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT 2020, Frascati, Italy, October 19-21, 2020*, pages 1–6. IEEE, 2020.

[SRJB23] Sayandeep Saha, Prasanna Ravi, Dirmanto Jap, and Shivam Bhasin. Non-profiled side-channel assisted fault attack: A case study on DOMREP. In *Proceedings of 29th Design, Automation and Test in Europe (DATE) 2023*, pages 1–6, Antwerp, Belgium, April 2023. IEEE.

[SRM20] Aein Rezaei Shahmirzadi, Shahram Rasoolzadeh, and Amir Moradi. Impeccable circuits II. In *57th ACM/IEEE Design Automation Conference, DAC 2020, San Francisco, CA, USA, July 20-24, 2020*, pages 1–6. IEEE, 2020.

[SSM22] Rajat Sadhukhan, Sayandeep Saha, and Debdeep Mukhopadhyay. Antisifa-cad: A framework to thwart SIFA at the layout level. In Tulika Mitra, Evangeline F. Y. Young, and Jinjun Xiong, editors, *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2022, San Diego, California, USA, 30 October 2022 - 3 November 2022*, pages 63:1–63:9. ACM, 2022.

[WCG+22] Qianmei Wu, Wei Cheng, Sylvain Guilley, Fan Zhang, and Wei Fu. On efficient and secure code-based masking: A pragmatic evaluation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(3):192–222, 2022.

[WM18] Felix Wegener and Amir Moradi. Yet another size record for AES: A first-order SCA secure AES s-box based on \(\mathrm {GF}(2^8)\) multiplication. In Begül Bilgin and Jean-Bernard Fischer, editors, *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, volume 11389 of *Lecture Notes in Computer Science*, pages 111–124. Springer, 2018.

[WMCS20] Weijia Wang, Pierrick Méaux, Gaëtan Cassiers, and François-Xavier Standaert. Efficient and private computations with code-based masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):128–171, 2020.

A Appendix

A.1 Summary of the Known Security Benefits of Prime-Field Masking

Previous works on the physical security benefits of additive prime-field masking have mostly focused on its side-channel resistance when the noise level is low [DFS16, MMMS23, CMM+23] – a condition that is known to create issues for additive masking in binary fields $\mathbb{F}_{2^n}$ (called Boolean masking). In particular, it has been demonstrated that in presence of leakage functions which are based on a linear combination of bits, Boolean masking cannot provide exponential security amplification in the number of shares without a fair amount of noise [MMMS23]. This is the case for commonly considered leakage models such as Hamming Weight (HW), Hamming Distance (HD), Most Significant Bit (MSB), Least Significant Bit (LSB), and their generalizations. For these models, strong adversaries can extract a constant amount of information about secret intermediates irrespective of the number of shares and the associated protection order, completely nullifying the purpose of the countermeasure. Prime-field masking on the other hand has been shown to provide security amplification in presence of any leakage function, linear or non-linear, as long as it is non-injective [DFS16], otherwise no uncertainty is left and the adversary can read-out all intermediates directly. Intuitively, the security benefits of prime-field masking can be attributed to the algebraic incompatibility between leakage model and recombination function. In Boolean encodings each bit of a secret variable can be recovered from a single bit of each of its shares. Since digital hardware is operating on binary signals it is also leaking information in binary form, i.e., the leakage is compatible with the recombination function. In prime-field masking, information from all bits of all shares is needed to reconstruct a single bit of the secret with certainty. No weighted sum of bits (such as Hamming weight/distance, bit leakage and their generalizations) contains enough information about each share to do that. Hence leakage model and recombination function are algebraically less compatible.

Interestingly, those previous works also showed that a part of the stronger resistance of prime-field masking against SCA attacks under very low noise levels (i.e., strong adversaries) remains valid at higher noise levels. In fact, it has been demonstrated through information theoretic analyses and experimental case studies that prime-field masking can show improved resistance at any noise level [MMMS23, CMM+23].

Analogously, we have studied the resistance of prime-field masking against statistical fault attacks in this work. We equally found that strong adversaries may easily circumvent the security features of Boolean masking altogether (e.g., injecting precise LSB stuck-at-0 faults in the shares), while generic attacks require a much higher complexity for prime-field masking. Furthermore, a part of the benefits of prime-field masking against strong attackers also remains valid against weaker adversaries (e.g., injecting imprecise LSB stuck-at-0 faults in the shares). Additionally, as also indicated in [MMMS23], we find that the size of the prime plays an important role to achieve optimal resistance.

As mentioned above, the concrete advantages, if any, of prime masking in the combined SCA + FIA scenario are not yet clear and demand further investigation.