Security of Hedged Fiat–Shamir Signatures Under Fault Attacks

Diego F. Aranha[1](✉), Claudio Orlandi[2](✉), Akira Takahashi[2], and Greg Zaverucha[3]

[1] Department of Engineering, DIGIT, Aarhus University, Aarhus, Denmark
dfaranha@eng.au.dk
[2] Department of Computer Science, DIGIT, Aarhus University, Aarhus, Denmark
{orlandi,takahashi}@cs.au.dk
[3] Microsoft Research, Redmond, USA
gregz@microsoft.com

Abstract. Deterministic generation of per-signature randomness has been a widely accepted solution to mitigate the catastrophic risk of randomness failure in Fiat–Shamir type signature schemes. However, recent studies have practically demonstrated that such de-randomized schemes, including EdDSA, are vulnerable to differential fault attacks, which enable adversaries to recover the entire secret signing key, by artificially provoking randomness reuse or corrupting computation in other ways. In order to balance concerns of both randomness failures and the threat of fault injection, some signature designs are advocating a "hedged" derivation of the per-signature randomness, by hashing the secret key, message, and a nonce. Despite the growing popularity of the hedged paradigm in practical signature schemes, to the best of our knowledge, there has been no attempt to formally analyze the fault resilience of hedged signatures.

We perform a formal security analysis of the fault resilience of signature schemes constructed via the Fiat–Shamir transform. We propose a model to characterize bit-tampering fault attacks, and investigate their impact across different steps of the signing operation. We prove that, for some types of faults, attacks are mitigated by the hedged paradigm, while attacks remain possible for others. As concrete case studies, we then apply our results to XEdDSA, a hedged version of EdDSA used in the Signal messaging protocol, and to Picnic2, a hedged Fiat–Shamir signature scheme in Round 2 of the NIST Post-Quantum standardization process.

1 Introduction

Deterministic Signatures and Fault Attacks. Some signature schemes require a fresh, secret random value per-signature, sometimes called a nonce. Nonce misuse is a devastating security threat intrinsic to these schemes, since the signing key can be computed after as few as two different messages are signed using the same value. The vulnerability can result from either programming mistakes attempting to implement non-trivial cryptographic standards, or faulty pseudo-random number generators. After multiple real-world implementations were found to

© International Association for Cryptologic Research 2020
A. Canteaut and Y. Ishai (Eds.): EUROCRYPT 2020, LNCS 12105, pp. 644–674, 2020.
https://doi.org/10.1007/978-3-030-45721-1_23

be surprisingly vulnerable to this attack [22,36] researchers and practitioners proposed deterministic signature schemes, such as EdDSA [16], as a countermeasure, in which per-signature randomness is derived from the message and secret key as a defense-in-depth mechanism. However, it has been shown that simple low-cost fault attacks during the computation of the derandomized signing operation can leak the secret key by artificially provoking nonce reuse or by corrupting computation in other ways [3,7,9,68]. Recent papers have experimentally demonstrated the feasibility of these attacks [62,66,67]. Moreover, [23] and [64] extended such fault attacks to exploit deterministic lattice-based signature schemes among round two candidates of the NIST Post-Quantum Cryptography Standardization Process [2], where resistance to side-channel attacks is a design goal. Despite these attacks, deterministic signature generation is still likely a positive outcome in improving security, since fault attacks are harder to mount.

Fault Resilience of Hedged Signatures. In order to balance concerns of both nonce reuse and the threat of fault injection, some signature designs are advocating deriving the per-signature randomness from the secret key sk, message m, and a nonce n. The intention is to re-introduce some randomness as a countermeasure to fault injection attacks, and gracefully handle the case of poor quality randomness, to achieve a middle-ground between fully-deterministic and fully-probabilistic schemes. We call constructions following this paradigm *hedged signatures*. Despite the growing popularity of the *hedged* paradigm in practical signature schemes (such as in XEdDSA, VXEdDSA [61], qTESLA [17], and Picnic2 [72]), to the best of our knowledge, there has been no attempt to formally analyze the fault resilience of hedged signatures in the literature. While the hedged construction intuitively mitigates some fault attacks that exploit the deterministic signatures, it does add a step where faults can be injected, and it has not been shown if faults to the hedging operation allow further attacks, potentially negating the benefit. Therefore, we set out to study the following question within the provable security methodology:

To what extent are hedged signatures secure against fault attacks?

Concretely, we study fault attacks in the context of signature schemes constructed from identification schemes using the Fiat–Shamir transform [40]. We propose a formal model to capture the internal functioning of signature schemes constructed in the hedged paradigm, and characterize faults to investigate their impact across different steps of the signature computation.

We prove that for some types of faults, attacks are mitigated by the hedged paradigm, while for others, attacks remain possible. This provides important information when designing fault-tolerant implementations. We then apply our results to hedged EdDSA (called XEdDSA) and the Picnic2 post-quantum signature scheme [72], both designed using the hedged construction. The XEdDSA scheme is used in the Signal protocol [27] which is in turn used by instant messaging services such as WhatsApp, Facebook Messenger and Skype.

Threat Model. We consider a weaker variant of the standard adversary assumed in the fault analysis literature [50], who is typically capable of injecting a fault into an arbitrary number of values. Our adversary is capable of injecting a single-bit fault each time a signature is computed. We further restrict the faults to be

injected at the interfaces between the typical *commit, challenge, and response* phases of Fiat–Shamir signatures, i.e., only those function inputs and outputs can be faulted. This models transient faults injected into registers or memory cells, but does not fully capture persisting faults that permanently modify values in key storage, voltage glitches to skip instructions or micro-architectural attacks to modify executed instructions (such as RowHammer and variants [56]).

We argue that, even if our model does not capture *all* possible fault attacks, it provides a meaningful abstraction of a large class of fault attacks, and thus our analysis provides an important first step towards understanding the security of hedged signatures in the presence of faults. This way, designers and implementers can focus on protecting the portions of the attack surface that are detected as *most relevant* in practice. We observe that the effects of fault attacks found in the literature targeting deterministic signatures can be essentially characterized as simple bit-tampering faults on function input/output, even though some of actual experiments cause faults during computation [23]. Moreover, an abstract model is needed to prove general results, and the general functions common to all Fiat–Shamir signatures are a natural candidate for abstraction.

We consider two single-bit tampering functions to set or flip individual bits, respectively: $\texttt{flip_bit}_i(x)$ to perform a logical negation of the i-th bit of x, and $\texttt{set_bit}_{i,b}(x)$ to set the i-th bit of x to b. This captures both stuck-at and bit-flip fault injection attacks [51], introduced as data flows through the implementation. Such attacks are practically targeted at various components of the device, e.g., memory cells, processor registers, or data buses.

1.1 Our Contributions

A New Security Model for Analyzing Fault Attacks. We establish a formal security model tailored to Fiat–Shamir type signatures (hedged, deterministic or fully probabilistic). We survey the literature on fault attacks, showing that our model captures many practical attacks. As a first step, we abstract real-world hedged signature schemes, basing our formalization on Bellare and Tackmann's nonce-based signatures [15] and Bellare, Poettering and Stebila's de-randomized signatures [14]. We call this security notion *unforgeability under chosen message and nonce attacks* UF-CMNA. In this security experiment, when submitting a message to the signing oracle, the adversary may also choose the random input to the *hedged extractor*, a function that derives the per-signature randomness from a nonce, the secret key, and the message.

Then we extend UF-CMNA to include resilience to fault attacks. In this security experiment the adversary plays a game similar to the UF-CMNA game, but the signing oracle also allows the attacker to specify a fault to be applied to a specific part of the signing algorithm. We identify eleven different fault types that the adversary can apply to the signing algorithm, and we denote them by $f_0, \ldots, f_{10}$. For example, fault type f_1 applies `set_bit` or `flip_bit` to the secret key input to the hedged extractor. This notion is called *unforgeability under faults, chosen message and nonce attacks*, and is denoted F-UF-fCMNA where F is a set of fault types.

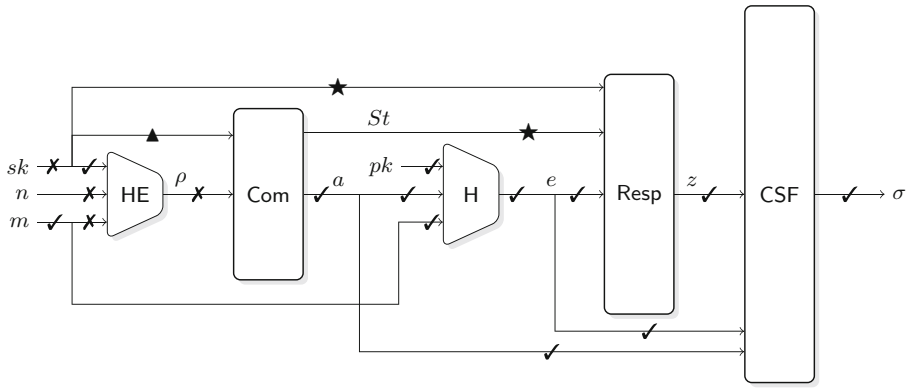

Fig. 1. Overview of our results for hedged Fiat–Shamir type signature schemes. ✓ indicates security against 1-bit fault on the corresponding wire value, and ✗ indicates an attack or counterexample. A ★ (resp. ▲) indicates that security only holds for the schemes derived from subset-revealing ID (resp. input-delayed ID) protocols. The function components HE, Com, H, Resp, and CSF stand for hedged extractor, commitment, hash function, response, and canonical serialization function, respectively (see Sects. 2 and 3 for the formal definitions).

Fault Resilience of Hedged Fiat–Shamir Signatures. We then prove that hedged Fiat–Shamir signature schemes are secure against attacks using certain fault types. Of the eleven fault types in our model, we found that the generic hedged Fiat–Shamir signature scheme is resilient to six of them (summarized in Fig. 1). As our model gives the attacker nearly full control of the RNG by default, our main results indicate that the hedged scheme can resist additional faults even in this (usually dire) scenario. The only constraint is that message-nonce pairs do not repeat as otherwise the scheme degenerates to a pure deterministic construction and attacks become trivial. When the underlying ID scheme has an additional property that we call *subset revealing*, the corresponding hedged signature scheme is secure against attacks that use eight of the eleven fault types. Overall, our results give a full characterization of which fault attacks are mitigated as intended by the hedged construction, and which fault attacks remain. Our conclusion is that hedging is never worse than the deterministic construction with respect to faults, plus it has the additional benefit of hedging against poor randomness.

Fault Resilience of XEdDSA and Picnic2. We use the Schnorr signature scheme throughout the paper as an example. As an application of our results, we show that hedged Schnorr resists attacks for six of the eleven fault types in our model. One implication is that the hedged scheme XEdDSA does provide better resistance to fault attacks than (deterministic) EdDSA. In particular, XEdDSA resists all fault injection attacks against EdDSA described in the literature that rely on nonce reuse without skipping nonce generation entirely [3,9,62,66,67]. We also show to what extent the Picnic2 signature scheme is secure against the

fault attacks in our model. Because it is subset-revealing, resistance to eight of the eleven fault types is immediately established by our results for generic ID schemes. For the remaining three, we prove security for one (using specific details of Picnic2), and show attacks for the other two.

1.2 Related Work

To the best of our knowledge, ours is the first work considering fault attacks on hedged constructions. However, the modeling and construction of secure cryptographic schemes in the presence of faults or tampering attacks has received plenty of attention in recent years. We survey some of this work below. Related work on fault attacks to deterministic signature schemes is given in Sect. 2.3.

De-randomized and Hedged Constructions. Bellare and Tackmann [15] studied cryptography that is hedged against randomness failures. They also describe the "folklore construction", where the signing key and message to be signed are used to derive the per-signature randomness, and additional randomness may or may not be included in the derivation. Schnorr signatures with this construction have been analyzed by M'Raihi et al. [59]. A generic version of the folklore derandomization construction was proven UF-CMA secure by Bellare, Pottering and Stebila [14]. Other works on hedged cryptography include [65] and [11,12,19,47] when considering hedged public-key encryption in particular.

Fault Attacks and Tamper-Resilient Signatures. Tamper-resilient cryptography has received plenty of attention, both in the context of theoretical and practical cryptographic research, dating back at least to the early paper of Boneh, Demillo and Lipton [20] considering fault attacks on RSA signatures (here it is noted that some attacks fail when a random padding is used, since it ensures that the same message is never signed twice). Later Coron and Mandal [28] proved that RSA-PSS is protected against random faults, and Barthe et al. [10] extends this to non-random faults as well. All of the above works contain examples of how randomization improves the security of signature schemes against fault attacks (in a provable way).

Other early work includes Gennaro et al. [43] that provides an early framework for proving tamper resilience, and Ishai et al. [49] which proposes generic transformation for tamper-resilient circuits. In a later work by Faust et al. [39] a different and incomparable model was considered, which in particular guarantees security against tampering with *arbitrary* number of wires. We note that our model is similar to theirs since it also considers adversaries that are allowed to flip or reset each bit in the circuit. Similar ideas are also used in practice when considering fault resilient masking (e.g., [32]).

In our model the adversary is only allowed to tamper with part of the computation. Similar limitations have been considered before in the literature to circumvent impossibility results, in particular in the so called *split-state model* [35]. Several constructions have been proposed in this model including: non-malleable codes (Dziembowski, Pietrzak and Wichs [35]), signature schemes (Faonio et al. [37]), and more (Liu and Lysyanskaya [57]).

Other related work on tamper resilient signature schemes includes [6,31,38,42]. Most of this previous work has focused on *constructing* novel tamper resilient signature schemes, or understanding the limits of tamper resilience, in theory. Instead, we focus on analyzing the tamper resilience of a popular transformation used in practice.

Related key attacks (RKA) can be seen as a special case of tampering. Bellare and Kohno [13] initiated the formal study of related-key attacks. Morita et al. [58] analyzed RKA security of Schnorr signatures.

Ineffective Fault Attacks (IFA) and Countermeasures. In this paper we consider not only `flip_bit` fault attacks, but also `set_bit` faults for the following reason. Clavier [26] proposed *ineffective fault attacks (IFA)*, in which the adversary forces a certain intermediate bit value to be stuck at 0 or 1, and tries to recover the secret internal state by observing whether the correct output is obtained (i.e., the injected fault was ineffective). IFA is very powerful, and works even if the target algorithm contains typical countermeasures against fault attacks, such as a correctness check after redundant operations [8] and the infective countermeasure [71]. IFA has been recently superseded by *statistical* ineffective fault attacks (SIFA) [33,34], that use statistical analysis to enable mounting IFA with low-precision bit-fixing, random or bit-flip faults. Daemen et al. [29] provided several practical countermeasures against SIFA, and their abstract adversarial model is close to ours in the sense that the adversaries are allowed to flip or set a single bit wire value in the circuit per query, though their security argument does not follow the provable security methodology.

Concurrent Work. An independent work by Fischlin and Günther [41] proposes a memory fault model for digital signatures and authenticated encryption. Their main result about a generic hedged signature scheme is two-fold: it is provably secure when the nonce is fully faulted, or when the message, nonce, and hedged extractor output are all differentially faulted in each signing query. The former essentially coincides with our Lemma 3, but with a different proof technique. For the latter, the outcome diverges because the adversarial power in our model is different in the following ways: 1. the adversary can locally inject a fault into sk as a hedged extractor input, 2. the adversary can inject a bit-fixing fault, not only a bit-flip (i.e., differential) fault, 3. the adversary has nearly full control over the nonce, instead of assuming nonces are randomly generated and subject to bit flips later on, but 4. the adversary cannot inject multi-bit faults into multiple variables in a query. We additionally consider fault attacks on other various intermediate values inside the signing operation. Our treatment is then more fine-grained and successfully captures typical existing attacks on deployed deterministic schemes (like attacks that fault the challenge hash), while [41] does not. The upside of the generic approach in [41] is that the result applies to more signature schemes.

$\mathsf{Gen}(1^\lambda)$	$\mathsf{Sign}(sk, m; \rho)$	$\mathsf{Verify}(pk, m, \sigma)$
$(pk, sk) \leftarrow \mathsf{IGen}(1^\lambda)$	$(a, St) \leftarrow \mathsf{Com}(sk; \rho)$	$(a, e, z) \leftarrow \mathsf{CDF}(\sigma, pk)$
return (pk, sk)	$e \leftarrow \mathsf{H}(a, m, pk)$	**return** $\mathsf{V}(a, e, z, pk) \stackrel{?}{=} 1$
	$z \leftarrow \mathsf{Resp}(sk, e, St)$	$\wedge\, \mathsf{H}(a, m, pk) \stackrel{?}{=} e$
$\mathsf{H}(x)$	$\sigma \leftarrow \mathsf{CSF}(a, e, z)$	
If $\mathrm{HT}[x] = \perp$:	**return** σ	
$\mathrm{HT}[x] \leftarrow\!\!\$\, D_H$		
return $\mathrm{HT}[x]$		

Fig. 2. The Fiat–Shamir transform applied to canonical ID with serialization CSF, to construct the signature scheme $\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$. The function $\mathsf{H} : \{0,1\}^* \to D_H$ is constructed with a cryptographic hash function which we model as a random oracle.

2 Preliminaries

Notation. The notation $|\cdot|$ denotes two quantities depending on the context: $|S|$ denotes the cardinality of a set S, and $|s|$ denotes the length of a bit string s. The notation $x \leftarrow\!\!\$\, X$ means that an element x is sampled from the set X uniformly at random. We often use the notation $[n]$ as a short hand for a set $\{1, \ldots, n\}$ where $n \in \mathbb{N}$. When we explicitly mention that an algorithm A is randomized, we use the notation $\mathsf{A}(x; \rho)$ meaning that it is executed on input x with random tape ρ. We also remark that if the lemmas/theorems are marked with "(informal)", then it means that asymptotic bounds are omitted. The full version [5] includes more rigorous statements for all of them.

Fiat–Shamir type Signature Schemes. This paper studies the robustness of Fiat–Shamir type signature schemes against fault attacks. The details of these algorithms appear in the full version. The Schnorr signature scheme [69] is one of the most well-known signature schemes using the Fiat–Shamir transform, and EdDSA and XEdDSA are essentially deterministic and hedged variants of Schnorr. The Picnic2 signature scheme [72] is constructed by applying the Fiat–Shamir transform to a three-round zero-knowledge proof system by Katz et al. [52], which follows so-called "MPC-in-the-head" paradigm [48]. The hedging strategy we study in this paper is recommended in its specification.

2.1 Definitions

In this subsection we recall several basic definitions related to digital signatures constructed from the identification protocols. Since this paper deals with Fiat–Shamir signatures, we always assume that the signing algorithm of digital signature schemes takes some randomness as input.

We now define a three-round public-coin identification protocol, the basis of Fiat–Shamir-type signatures. The definition below essentially follows the formalization of [54] unless explicitly stated.

Definition 1 (Canonical Identification Protocol). *A* canonical identification protocol*, denoted by a tuple of algorithms* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{Com}, \mathsf{Resp}, \mathsf{V})$*, is a three-round protocol defined as follows:*

- $\mathsf{IGen}(1^\lambda)$*, where* λ *is a security parameter, outputs a key pair* (sk, pk)*. In the context of identification protocols,* pk *and* sk *are sometimes called* statement *and* witness*. We assume that* IGen *defines a* hard-relation*, and that* pk *defines the parameters of the scheme including:* randomness space D_ρ*,* commitment space A*,* challenge space D_H *and* response space Z*.*
- Prover *invokes a committing algorithm* Com *on a secret key* sk *and randomness* $\rho \in D_\rho$ *as input, and outputs a* commitment $a \in A$ *and* state St*.*
- $\mathsf{Verifier}$ *samples a* challenge e *from the challenge space* $D_H \subseteq \{0,1\}^*$*.*
- Prover *executes a response algorithm* Resp *on* (sk, e, St) *to compute a* response $z \in Z \cup \{\perp\}$*, where* $\perp \notin Z$ *is a special symbol indicating failure. On top of this standard formalization, we further require that* Resp *returns* $\perp$ *whenever it receives a malformed challenge* $\tilde{e} \notin D_H$*, as such a simple sanity check is performed in most practical implementations.*
- $\mathsf{Verifier}$ *executes a verification algorithm* V *on* (a, e, z, pk) *as input, to output 1 (i.e., accept) or 0 (i.e., reject).*

We call a triple $(a, e, z) \in A \times D_H \times Z \cup \{\perp, \perp, \perp\}$ *a* transcript*, and it is said to be* valid *with respect to* pk *if* $\mathsf{V}(a, e, z, pk) = 1$*. We say that* ID *is* correct *if for every pair* (pk, sk) *output by* IGen*, for every* $\rho \in D_\rho$*, and for every transcript* (a, e, z) *from an honest execution of the protocol between* $\mathsf{Prover}(sk; \rho)$ *and* $\mathsf{Verifier}(pk)$*,* $\Pr[\mathsf{V}(a, e, z, pk) = 1] = 1$*.*

Remark. The response algorithm in the above definition does not explicitly take a commitment a as input. We decided to do so since a is generally not required to compute z, such as in the Schnorr identification scheme and, if needed, we assume that St contains a copy of a.

The following definition is adapted from [46, Chapter 6]. We explicitly differentiate three flavors of the special HVZK property depending on a level of indistinguishability, following the approach found in [44, Chapter 4]. Note that ϵ_{HVZK} below is equal to 0 for special *perfect* HVZK.

Definition 2 (Special c/s/p-HVZK). *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{Com}, \mathsf{Resp}, \mathsf{V})$ *be a canonical identification protocol.* ID *is said to be* special computational/statistical/perfect honest-verifier zero knowledge (special c/s/p-HVZK) *if there exists a probabilistic polynomial-time simulator* $\mathcal{M}$*, which on input* pk *and* e *outputs a transcript of the form* (a, e, z) *that is computationally/statistically/perfectly indistinguishable from a real transcript between an honest prover and verifier on common input* pk*. We also denote by* ϵ_{HVZK} *the upper bound on the advantage of all probabilistic polynomial-time distinguishing algorithms.*

In our security analysis of specific hedged-signature schemes in the presence of faults we will provide a concrete bound on the min-entropy of the associated ID scheme. But here we present a useful lemma stating that the commitment message a of any secure identification scheme must have high min-entropy. The lemma might be folklore but we were unable to find a reference to it, so we include it for completeness in the full version.

Lemma 1. *Let* ID *be a canonical identification protocol as in Definition 1, satisfying special-soundness and HVZK (as in Definition 2). Then, the min-entropy* α *of the commitment message* a *(given the public key) is at least* $\alpha = \omega(\log(\lambda))$

Definition 3 (Subset Revealing Identification Protocol). *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{Com}, \mathsf{Resp}, \mathsf{V})$ *be a canonical identification protocol. We say that* ID *is* subset revealing *if* ID *satisfies the following. (1)* St *is a set of* c *states* $\{St_1, \ldots, St_c\}$*, (2)* Resp *first derives an index set* $I \subset [c]$ *using only* e *as input, and outputs* St_i *for* $i \in I$ *as* z*, and (3)* $|St|$ *and* $|D_H|$ *are both polynomial in* λ*.*

Remark. Similar definitions were previously given by Kilian et al. [53] and Chailloux [24], where they make zero-knowledge or identification protocols simply reveal a subset of committed strings. Our definition generalizes their notion so that it can cover some protocols that reveal arbitrary values other than committed strings. Also notice that the Resp function of subset revealing ID schemes does not use sk at all. The above definition includes the $\mathsf{Picnic2}$ identification protocol (discussed in more detail in Sect. 6), and many classic three-round public-coin zero-knowledge proof protocols, such as the ones for graph isomorphism, Hamilton graphs, and 3-colorable graphs [45]. We also emphasize that $|St|$ and $|D_H|$ need to be restricted for efficiency reasons – otherwise any identification protocol (including Schnorr) could be made subset revealing by simply precomputing (exponentially many) responses for every possible challenge and storing them in the state.

Serialization of Transcripts. For efficiency purposes, most Fiat-Shamir based signature schemes do not include the entire transcript of the identification protocol as part of the signature. Instead, redundant parts are omitted and recomputed during the verification phase. Different signature schemes omit different parts of the transcript: in some cases a is omitted and in others e is omitted. To capture this in our framework without loss of generality we introduce a *serialization* function that turns the transcript of an identification protocol into a signature.

Definition 4 (Canonical Serialization Function). *Let* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{Com}, \mathsf{Resp}, \mathsf{V})$ *be a canonical identification protocol, and let* pk *be a public key output by* IGen*. We call a function* $\mathsf{CSF} : \{0,1\}^* \to \{0,1\}^*$ *a* canonical serialization function *if* CSF *is efficiently computable and deterministic, and satisfies the following basic properties: (1) it is* valid*, meaning that there exists a corresponding de-serialization function* CDF *which satisfies the following: for any transcript* $(a, e, z) \in A \times D_H \times Z \cup \{\perp, \perp, \perp\}$ *such that* $\mathsf{V}(a, e, z, pk) = 1$*, it holds that* $\mathsf{CDF}(\mathsf{CSF}(a, e, z), pk) = (a, e, z)$*, and (2) it is* sound with respect to invalid responses*, meaning that it returns* $\perp$ *upon receiving* $z = \perp$ *as input.*

Definition 5 (Fiat–Shamir Transform). The Fiat–Shamir transform, *denoted by* **FS**, *takes a canonical identification protocol* ID *and canonical serialization function* CSF *as input, and outputs a signature scheme* $\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *defined in Fig. 2. For convenience, this paper refers to such schemes as* Fiat–Shamir type signature schemes.

Remarks. By construction, it holds that if ID is correct, then $\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}]$ is a correct signature scheme. We assume ID is correct throughout the paper. In Fig. 2, the verification condition may appear redundant. However, the above definition allows us to capture several variations of the Fiat–Shamir transform. For instance, a type of Fiat–Shamir transform found in some papers e.g., Ohta–Okamoto [60] and Abdalla et al. [1] can be obtained by letting $\mathsf{CSF}(a, e, z)$ output $\sigma := (a, z)$ and letting $\mathsf{CDF}(\sigma, pk)$ call $e \leftarrow \mathsf{H}(a, m, pk)$ inside to reconstruct the whole transcript. In contrast, if ID is commitment-recoverable [54], one can instantiate its serialization as follows: $\mathsf{CSF}(a, e, z)$ outputs $\sigma := (e, z)$ and $\mathsf{CDF}(\sigma, pk)$ calls $a \leftarrow \mathsf{Recover}(pk, e, z)$ inside to reconstruct the transcript.

2.2 Relation Between UF-KOA Security and UF-CMA Security

The security notion *unforgeability against key-only attacks* (UF-KOA), is the same as UF-CMA, but with the restriction that the adversary is only given the public key, and no Sign oracle. The following result is a mild generalization of [55, Lemma 3.8]: the original lemma only covers perfect HVZK and does not include the serialization function which we use in this work. The proof is very similar to the original one and is provided in the full version. In Sect. 4, we extend this result, showing that for some signature schemes security against key-only attacks implies security against certain fault attacks.

Lemma 2 (UF-KOA $\rightarrow$ UF-CMA **(informal)).** *Let* ID *be a correct canonical identification protocol and* CSF *be a canonical serialization function for* ID. *Suppose* ID *is special c/s/p-HVZK and has* α*-bit min-entropy. If* $\mathsf{FS} := \mathbf{FS}[\mathsf{ID}, \mathsf{CSF}]$ *is* UF-KOA *secure, then* FS *is* UF-CMA *secure in the random oracle model.*

2.3 Fault Attacks on Deterministic Fiat–Shamir Signatures

In recent years, several papers [3,9,62,66,67] presented differential fault attacks against deterministic Fiat–Shamir-type schemes. We present the conceptual overview of those previous attacks. A more detailed survey is given in the full version [5].

Special Soundness Attack (SSND). This type of attack exploits the *special soundness* property of the underlying canonical identification protocol. That is, there exists an efficient algorithm that extracts the witness sk corresponding to the statement pk, given two accepting transcripts (a, e, z) and (a, e', z'), where $e \neq e'$ [30]. Note in fact that it is easier to extract the secret key for an attacker than for a knowledge extractor in a proof of security, since the attacker can

assume that the prover honestly follows the protocol while the special soundness property considers possibly cheating provers. SSND can be cheaply achieved by injecting a fault into commitment output, or hash input/output.

Large Randomness Bias Attack (LRB). This attack slightly modifies the randomness ρ to $\rho' = \rho + \Delta$ using, e.g., `flip_bit` fault. The attack highly relies on the deterministic property because the adversary knows that all signatures on the same message m use the same ρ, and if ρ is slightly perturbed by some sufficiently small Δ, he can find Δ with an exhaustive search. Then the adversary can recover the secret key by querying two deterministic signatures on the same message, which were computed using correlated randomness ρ and $\rho + \Delta$. LRB can be cheaply achieved by injecting a fault into the deterministic randomness derivation phase, or the randomness as response input.

3 Formal Treatment of Hedged Signatures

In this section, we give formal definitions for a hedged signature scheme and its security notion, based on Bellare–Tackmann's *nonce-based signatures* [15, § 5] and Bellare–Poettering–Stebila's *de-randomized signatures* [14, § 5.1]. Then we define our new security notion for hedged Fiat–Shamir signature schemes, which guarantees resilience against 1-bit faults on function inputs/outputs.

$\mathsf{HSign}(sk, m, n)$	$\mathsf{Exp}^{\mathsf{UF\text{-}CMNA}}_{\mathsf{HSIG,HE}}(\mathcal{A})$	$\mathsf{OHSign}(m, n)$
$\rho \leftarrow \mathsf{HE}(sk, (m, n))$ $\sigma \leftarrow \mathsf{Sign}(sk, m; \rho)$ **return** σ	$M \leftarrow \emptyset; \mathrm{HET} \leftarrow \emptyset$ $(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$ $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OHSign,HE}}(pk)$ $v \leftarrow \mathsf{Verify}(m^*, \sigma^*)$ **return** $(v = 1) \wedge m^* \notin M$	$\sigma \leftarrow \mathsf{HSign}(sk, m, n)$ $M \leftarrow M \cup \{m\}$ **return** σ $\mathsf{HE}(sk', (m', n'))$ If $\mathrm{HET}[sk', m', n'] = \perp$: $\mathrm{HET}[sk', m', n'] \leftarrow_\$ D_\rho$ **return** $\mathrm{HET}[sk', m', n']$

Fig. 3. Hedged signature scheme $\mathsf{HSIG} = \mathbf{R2H}[\mathsf{SIG}, \mathsf{HE}] = (\mathsf{Gen}, \mathsf{HSign}, \mathsf{Verify})$ and UF-CMNA experiment. Key generation and verification are unchanged.

3.1 Security of Hedged Signature Schemes

We now consider a simple transformation **R2H**, which converts a randomized signature scheme to a so-called "hedged" one, and its security notion UF-CMNA (unforgeability against chosen message and nonce attacks). See Fig. 3 for the full details. Parts of the transformation appear in the literature independently, but by combining them, we can model the concrete hedged signature schemes of

interest. We now describe the differences and similarities between **R2H** and the transformations that appeared in previous works.

On one hand, a hedged signing algorithm HSign takes a *nonce* n along with a message m, and derives the randomness $\rho \in D_\rho$ (of length ℓ_ρ bits) with a *hedged extractor* HE with $(sk, (m, n))$ as input. We do not specify how the nonces are generated here, but in practice they are the output of a pseudorandom number generator. As we will see soon, low entropy nonces do not really degrade the security of hedged signatures as long as the underlying randomized signature scheme is secure. The hedged construction we presented is essentially based on the approach taken in [15]. Note that HE is in practice a cryptographic hash function, that we will model as a random oracle.

On the other hand, we use the signing key sk as the key for the hedged extractor, whereas Bellare and Tackmann used a separately generated key (which they called the "seed"), that must be stored with sk. We chose to do so in order to model concrete hedged Fiat–Shamir type schemes, such as XEdDSA and Picnic2. In fact, the security of the deterministic construction that hashes sk and m to derive ρ (with no nonce) was formally treated by Bellare–Poettering–Stebila [14], and our security proof in the next section extends their result. Moreover, the signing oracle OHSign in our $\mathsf{UF\text{-}CMNA}$ experiment takes m and n as input adaptively chosen by the adversary $\mathcal{A}$. This can be regarded as the strongest instantiation of the oracle provided in [15], where nonces are derived via what they call a nonce generator (NG). Indeed, one of their results for nonce-based signatures (Theorem 5.1) does not impose any restrictions on NG, and it implicitly allows adversaries to fully control how the nonces are chosen in the signing oracle.

Now we formally define a security notion for hedged signature schemes, as a natural extension of the standard $\mathsf{UF\text{-}CMA}$ security definition. We also give a tweaked version of Theorem 4 in [14], where they only consider the signing oracle that doesn't take adversarially chosen nonces. Note that Lemma 3 applies to *any* secure signature schemes and hence it may be of independent interest. We present a proof in the full version [5] for completeness.

Definition 6 ($\mathsf{UF\text{-}CMNA}$)**.** *A hedged signature scheme* $\mathsf{HSIG} = (\mathsf{Gen}, \mathsf{HSign}, \mathsf{Verify})$ *is said to be* $\mathsf{UF\text{-}CMNA}$ *secure in the random oracle model, if for any probabilistic polynomial time adversary* $\mathcal{A}$*, its advantage*

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMNA}}_{\mathsf{HSIG},\mathsf{HE}}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\mathsf{UF\text{-}CMNA}}_{\mathsf{HSIG},\mathsf{HE}}(\mathcal{A}) = 1\right]$$

is negligible in security parameter λ*, where* $\mathsf{Exp}^{\mathsf{UF\text{-}CMNA}}_{\mathsf{HSIG},\mathsf{HE}}(\mathcal{A})$ *is described in Fig. 3.*

Lemma 3 ($\mathsf{UF\text{-}CMA} \rightarrow \mathsf{UF\text{-}CMNA}$ **(informal))**. *Let* $\mathsf{SIG} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *be a randomized digital signature scheme, and let* $\mathsf{HSIG} := \mathbf{R2H}[\mathsf{SIG}, \mathsf{HE}] = (\mathsf{Gen}, \mathsf{HSign}, \mathsf{Verify})$ *be the corresponding hedged signature scheme with* HE *modeled as a random oracle. If* SIG *is* $\mathsf{UF\text{-}CMA}$ *secure, then* HSIG *is* $\mathsf{UF\text{-}CMNA}$ *secure.*

$\mathsf{Exp}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{A})$	$\mathsf{Exp}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS,HE}}(\mathcal{A})$	$\mathsf{OFaultHSign}(m, n, j, \phi)$
$M \leftarrow \emptyset; \mathrm{HT} \leftarrow \emptyset;$	$\boxed{\mathrm{HET} \leftarrow \emptyset}$	$f_j := \phi; f_k := Id$ for $k \neq j$
$(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$		$\rho \leftarrow f_2(\mathsf{HE}(f_1(sk), f_0(m, n)))$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OFaultSign,H}}(pk)$		$(a, St) \leftarrow f_4(\mathsf{Com}(f_3(sk; \rho)))$
		$\hat{a}, \hat{m}, \hat{pk} \leftarrow f_5(a, m, pk)$
	$\boxed{(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OFaultHSign,H,HE}}(pk)}$	$e \leftarrow f_6(\mathsf{H}(\hat{a}, \hat{m}, \hat{pk}))$
$v \leftarrow \mathsf{Verify}(m^*, \sigma^*)$		$z \leftarrow f_8(\mathsf{Resp}(f_7(sk, e, St)))$
return $(v = 1) \wedge m^* \notin M$		$\sigma \leftarrow f_{10}(\mathsf{CSF}(f_9(a, e, z)))$
		$M \leftarrow M \cup \{\hat{m}\}$; **return** σ

Fig. 4. UF-fCMNA and UF-fCMA security experiments and faulty signing oracles for both hedged (HFS) and plain (FS) Fiat–Shamir signature schemes. Id stands for the identity function. The function H and HE (not shown), are the same as in Figs. 2 and 3, respectively. The procedure $\mathsf{OFaultSign}(m, j, \phi)$ (omitted) is the same as OFaultHSign, but the line assigning to ρ is replaced with $\rho \leftarrow\$ D_\rho; \rho \leftarrow f_2(\rho)$.

3.2 Security of Hedged FS Type Signature Schemes Against Fault Adversaries

1-bit Transient Fault on Function Input/Output. To model transient fault attackers on data flow, recall that we consider the following 1-bit tampering functions: (1) $\texttt{flip_bit}_i(x)$, which does a logical negation of the i-th bit of x, and $\texttt{set_bit}_{i,b}(x)$, which sets the i-th bit of x to b. Using $\texttt{flip_bit}_i(x)$ (for instance, with a random position i), we can model a typical bit-flip induced from fault injection to the memory cells, CPU register values, or data buses of the target device. Beyond faults, we also wish to capture the case in which the randomness has a 1-bit bias, which has been shown to be a serious threat for some Fiat–Shamir type signatures [4]. We can model this using $\texttt{set_bit}_{i,b}$: when this function is applied to ρ, we can ensure that the first bit of ρ is "stuck" at zero by setting $i = 0$ and $b = 0$ to model 1-bit bias. Moreover, `set_bit` is a typical way to achieve so-called ineffective fault attacks [26,34]. Our formalization covers many fault attacks found in the surveyed literature (in the full version), as they rely only on low precision faults like random bit flips of the function input or output.

As a notable difference between our fault adversary model and actual attacks, some surveyed papers caused faults on several bits/bytes of function input or output when performing fault attack experiments. This is *not* to take advantage of multiple-bit faults, but rather because reliably causing a fault on a specific target memory cell is difficult in practical experiments. In fact, the attacks we classified as `SSND` and `LRB` can be achieved with uncontrolled 1-bit flip faults, and hence our model at least seems to capture the essence of previous attacks exploiting the deterministic nature of signing. A natural generalization is to

allow set_bit to work on multiple bits, for example to model word faults, or word zeroing faults. We can also model stronger attacks that are uncommon in the literature, such as setting words to arbitrary values. However, we focus on 1-bit faults in this paper as a first attempt to perform the formal analyses. We leave the security analysis against multi-bit faults for future work. In the full version, we describe some more fault attacks that are not covered by our model, to illustrate the limitations of our analysis. Each of these issues makes an interesting direction for future work.

Equipping UF-CMNA *Adversaries with Faults.* Now we are ready to define security against fault adversaries using the above tampering functions. In Fig. 4, we give the modified hedged signing oracle OFaultHSign, which additionally takes a tampering function $\phi \in \{\texttt{set_bit}_{i,b}, \texttt{flip_bit}_i, Id\}$ and $j \in [0, 10]$ as input, where Id is the identity function. This way, the adversary can specify for each query the tampering function (ϕ) as well as the target input/output position (j) within the signing operation to be faulted. For example, when $j = 6$, ϕ is applied to the output of the hash function H, and when $j = 5$ it is applied to the input to H. The other positions are not faulted. Notice that we also allow the adversary to set $\phi := Id$ in arbitrary signing queries, so OFaultHSign includes the behavior of the non-faulty oracle OHSign as a special case. A generalization we considered but decided against, is allowing faults on multiple wire values per sign query. The combinatorial complexity of security analysis in this setting is daunting, and we did not find this to be relevant in practice, based on our survey of practical attacks.

Definition 7 (UF-fCMNA)**.** *A hedged Fiat–Shamir signature scheme*

$$\mathsf{HFS} := \mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}] = (\mathsf{Gen}, \mathsf{HSign}, \mathsf{Verify})$$

*is said to be F-*UF-fCMNA *secure, if for any probabilistic polynomial time adversary $\mathcal{A}$ who makes queries to* OFaultHSign *with a fault function $f_j \in F \subseteq \{f_0, \ldots, f_{10}\}$ for each query (called F-adversary), its advantage*

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS},\mathsf{HE}}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS},\mathsf{HE}}(\mathcal{A}) = 1\right]$$

is negligible in security parameter λ, where $\mathsf{Exp}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS},\mathsf{HE}}(\mathcal{A})$ is described in Fig. 4.

In the next section, we also use the following intermediate security notion, which essentially guarantees the security of plain randomized Fiat–Shamir signature scheme against fault adversaries.

Definition 8 (UF-fCMA)**.** *A Fiat–Shamir signature scheme*

$$\mathsf{FS} := \mathbf{FS}[\mathsf{ID}, \mathsf{CSF}] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$$

*is said to be F-*UF-fCMA *secure, if for any probabilistic polynomial time adversary $\mathcal{A}$ who makes queries to* OFaultSign *with a fault function $f_j \in F \subseteq \{f_2, \ldots, f_{10}\}$ per each query (called F-adversary), its advantage*

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{A}) = 1\right]$$

is negligible in security parameter λ, where $\mathsf{Exp}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{A})$ is described in Fig. 4.

Trivial Faults on the Root Input Wire Values. We remark the existence of two faults on the left most input wires in Fig. 1, which we do not explicitly consider in our model, but its (in)security can be proven trivially. First, faulting message m before it is loaded by the signing oracle can be regarded as a situation where the adversary queries a faulty message $\hat{m}$ to begin with, since the oracle stores $\hat{m}$ in M. Hence we can just treat such a query as one to non-faulty signing oracle (OSign). Second, the adversary could easily recover the entire secret key after roughly $|sk|$ signing queries by injecting $\texttt{set_bit}$ faults to sk before it is loaded by the signing oracle, and the faulty secret key $\tilde{sk}$ is globally used throughout the signing operation: for example, if the most significant bit of sk is set to 0 at the very beginning of signing and its output still passed the verification, then the adversary can conclude that sk has 0 in the most significant bit with high probability. In doing so, the adversary iteratively recovers sk bit-by-bit if the fault is transient. The attack above is essentially a well-known impossibility result by Gennaro et al. [43] and such an attack can be practically achieved with ineffective faults. To overcome this issue, one would require an additional strict assumption on the upper-bound of faulty signing queries [31], or the signing algorithm needs to have some sophisticated features like self-destruct or key-updating mechanisms, which, however, are not yet widely implemented in real-world systems and are beyond the scope of this paper.

Winning Condition of Fault Adversaries. As described in Fig. 4, the $\mathsf{UF\text{-}fCMNA}$ experiment keeps track of possibly faulty messages $\hat{m}$ instead of queried messages m, and it does not regard σ^* as valid forgery if it verifies with $\hat{m}$ that $\mathcal{A}$ caused in prior queries. This may appear artificial, but we introduced this condition to rule out a trivial forgery "attack": if the experiment only keeps track of queried message m_i in i-th query, and adversaries target f_5 at m_i as hash input, they obtain a valid signature $\hat{\sigma}_i$ on message $\hat{m}_i$, yet $\hat{m}_i$ is not stored in a set of queried messages M. Hence the adversary can trivially win $\mathsf{UF\text{-}fCMNA}$ game by just submitting $(\hat{\sigma}_i, \hat{m}_i)$, which of course verifies. This is not an actual attack, since what $\mathcal{A}$ does there is essentially asking for a signature on $\hat{m}_i$ from the signing oracle, and hence outputting such a signature as forgery should not be considered as a meaningful threat.

Note that the $\mathsf{OFaultHSign}$ oracle in Fig. 4 stores all queried messages in the same set M, whether the adversary $\mathcal{A}$ decides to inject a fault (i.e., $\phi \in \{\texttt{set_bit}_{i,b}, \texttt{flip_bit}_i\}$) or not (i.e., $\phi := Id$), and so a forgery (m^*, σ^*) output by $\mathcal{A}$ is *not* considered valid even if m^* was only queried to $\mathsf{OFaultHSign}$ to obtain a faulty invalid signature. For some signature algorithms and fault types this is required; for example with Fiat–Shamir type signatures (derived from a commitment recoverable identification [54]), one can query $\mathsf{OFaultHSign}$ to get a signature (e, z) with a single bit-flip in z, and create a valid forgery by unflipping the bit.

Validity of Oracle Output. The signature output by $\mathsf{OFaultHSign}$ does not need to verify, but it may need to be well-formed in some way. Typically we show with a hybrid argument that $\mathsf{OFaultHSign}$ can be simulated without use of the private key, in a similar way to OHSign. In order for simulated outputs of $\mathsf{OFaultHSign}$ to

be indistinguishable from real outputs, simulated signatures must be correctly distributed. In [10,28], the security proof shows that the faulty signature is statistically close to a value drawn from the uniform distribution, so OFaultHSign can output a random value. For the Fiat–Shamir type signature schemes we study this is not the case, for some fault types the real output of OFaultHSign verifies with an appropriately faulted hash function, and our proofs must take care to maintain these properties when simulating OFaultHSign.

4 Security of Hedged Signatures Against Fault Attacks

In this section we establish the (in)security of the class of hedged Fiat–Shamir signatures schemes. We give here a short overview of the main intuition behind the results in Table 1: f_0 faults (on the (message, nonce) pair which is input to the hedged-extractor) cannot be tolerated since they allow the adversary to get two signatures with the same randomness. On the other hand f_1 faults (on the secret key input to the hedged-extractor) can be tolerated since they do not significantly change the distribution input to the hedged-extractor. If the adversary faults the output of the hedged extractor (using f_2), we cannot prove security in general (and we can list concrete attacks e.g., against the Schnorr signature schemes), but we can prove security for the specific case of Picnic2, since the output of the hedged-extractor is not used directly, but is given as input to a PRG – thus the small bias is "absorbed" by PRG security. We remark that, while present, this attack is much less devastating than the large randomness bias LRB attack on deterministic schemes (described in Sect. 2.3). With the LRB attack, the adversary only needs two signatures to recover the full key, while the attack we will show on Schnorr signature requires a significant amount of faulty biased signatures as input in practice. This indicates that hedged constructions do, to some extent, mitigate the effect of faults on the synthetic randomness.

The hedged approach does not help when the adversary faults the input to the commitment function (via f_3), since in this case the adversary can attempt to set the bits of the secret key one at the time and check if the output signature is valid or not. Note that in some kinds of ID schemes like Schnorr (known as *input-delayed* protocols [25]) the secret key is not used in the commitment function. Faulting the input of the commitment function can still lead to insecurity, e.g., in Schnorr the adversary can bias the randomness, which in turns leads to a total break of the signature scheme. Next, the adversary can fault the output of the commitment function (via f_4): this leads to insecurity in general, e.g., in Schnorr this also leads to randomness bias. However, for a large class of ID schemes (which we call *subset-revealing*), including Picnic2, this fault does not lead to insecurity: intuitively either the adversary faults something that will be output as part of the response (which can easily be simulated by learning a non-faulty signature and then applying the fault on the result), or it is not part of the output and therefore irrelevant. Attacking the input or the output of the random oracle used to derive the challenge (f_5 and f_6) does not lead to insecurity, since the distribution of the random oracle does not change due to the fault (note that

Table 1. Summary of results for UF-fCMNA security of the hedged Fiat–Shamir type construction, for all fault types. ✓ indicates a proof of UF-fCMNA security, and ✗ indicates an attack or counterexample.

Fault type	ID is subset-revealing	ID not subset-revealing	XEdDSA	Picnic2
f_0	✗ Lemma 11		✗	✗
f_1	✓ Lemma 4		✓ Corollary 1	✓ Corollary 3
f_2	✗ Lemma 13		✗	✓ Lemma 19
f_3	✗ Lemma 12		✗	✗ Sect. 6
f_4	✓ Lemma 10	✗ Lemma 15	✗	✓ Corollary 3
f_5	✓ Lemma 7		✓ Corollary 1	
f_6	✓ Lemma 8			
f_7	✓ Lemma 9	✗ Lemma 14		
f_8, f_9, f_{10}	✓ Lemma 6			

this would not be the case for deterministic signatures, where this kind of fault would be fatal). Faults against the input of the response function (via f_7) can break non-subset revealing signatures (once again, we can show that this fault can be used to break Schnorr signatures), but do not help the adversary in the case of a subset-revealing signature like Picnic2: similar to the case of f_4 faults, we use the fact that if the response function only outputs subsets of its input, faulting part of the input either has no effect or can be efficiently simulated given a non-faulty signature. Similarly, faults against the output of the response function or the input/output of the serialization function (fault types f_8, f_9, f_{10}) can also be easily simulated from a non-faulty signature.

We expand this high-level intuition into full proofs by carefully measuring the concrete security loss in the reductions which is introduced by the different kind of faults. More precisely, we present a concrete reduction from UF-KOA to $\{f_1, f_4, \ldots, f_{10}\}$-UF-fCMNA security for schemes derived from subset-revealing ID schemes, and to $\{f_1, f_5, f_6, f_8, f_9, f_{10}\}$-UF-fCMNA when ID is non-subset-revealing. Our theorems generalize and adapt results from [14] and [55] without introducing significant additional concrete security loss. Then in Sect. 4.7, we describe attacks for the remaining fault types (f_0, f_2 and f_3), completely characterizing the security of generic $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}]$ signature schemes for fault types $f_0, \ldots, f_{10}$.

4.1 Main Positive Result

Theorem 1 (UF-KOA → UF-fCMNA)**.** *Let* ID *be a canonical identification protocol and* CSF *be a canonical serialization function for* ID*. Suppose* ID *satisfies the same properties as in Lemma 2 and it is subset revealing, and moreover, let us assume that* $\mathcal{A}$ *does not query the same* (m, n) *pair to* OFaultHSign *more than once. Then if* $\mathsf{FS} := \mathbf{FS}[\mathsf{ID}, \mathsf{CSF}]$ *is* UF-KOA *secure,* $\mathsf{HFS} := \mathbf{R2H}[\mathsf{FS}, \mathsf{HE}]$ *is* $\{f_1, f_4, \ldots, f_{10}\}$-UF-fCMNA *secure in the random oracle model. Concretely, given* $\{f_1, f_4, \ldots, f_{10}\}$*-adversary* $\mathcal{A}$ *against* HFS *running in time* t*, and making*

at most Q_s queries to OFaultHSign*, Q_h queries to* H *and Q_{he} queries to* HE*, one can construct another adversary $\mathcal{B}$ against* FS *such that*

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS,HE}}(\mathcal{A}) \leq 2 \cdot \left(\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{FS}}(\mathcal{B}) + \frac{(Q_s + Q_h)Q_s}{2^{\alpha-1}} + Q_s \cdot \epsilon_{HVZK} \right),$$

where $\mathcal{B}$ makes at most Q_h queries to its hash oracle, and has running time t plus $Q_{he} \cdot |sk|$ invocations of Sign *and* Verify *of* FS*. Moreover, if we do not assume the subset-revealing property of* ID *and assume all the other conditions above, then we have that* HFS *is $\{f_1, f_5, f_6, f_8, f_9, f_{10}\}$-*UF-fCMNA *secure.*

Proof. The proof is two-fold. See Lemmas 4 and 5.

For the rest of this section we will assume that ID satisfies the properties in Lemma 2. As a first step, we give a reduction from UF-fCMA to UF-fCMNA security, and then we later give a reduction from UF-KOA to UF-fCMA. We observe that the UF-CMA-to-UF-CMNA reduction in Lemma 3 is mostly preserved, even in the presence of 1-bit faults on sk as a hedged extractor key. However, our proof shows that such a fault does affect the running time of the adversary because the reduction algorithm needs to go through all secret key candidates queried to random oracle *and* their faulty bit-flipped variants. We present a proof in the full version.

Lemma 4 **(F-UF-fCMA $\rightarrow F \cup \{f_1\}$-UF-fCMNA).** *Suppose the fault adversary $\mathcal{A}$ does not query the same (m, n) pair to* OFaultHSign *more than once. If* FS $:=$ **FS**[ID, CSF] *is F-*UF-fCMA *secure, then* HFS $:=$ **R2H**[FS, HE] *is F'-*UF-fCMNA *secure in the random oracle model, where $F' = F \cup \{f_1\}$. Concretely, given an F'-adversary $\mathcal{A}$ against* HFS *running in time t, and making at most Q_s queries to* OFaultHSign*, Q_h queries to* H *and Q_{he} queries to* HE*, one can construct F-adversary $\mathcal{B}$ against* FS *such that*

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS,HE}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{B}),$$

where $\mathcal{B}$ makes at most Q_s queries to its signing oracle OFaultSign *and Q_h queries to its hash oracle, and has running time $t' \approx t + Q_{he} \cdot |sk|$.*

Remarks. Our reduction above crucially relies upon the assumption that adversaries are not allowed to query the same (m, n) pair. Without this condition, OFaultHSign must return a faulty signature derived from the same randomness ρ if the same (m, n) is queried twice, and thus one could not simulate it using OFaultSign as an oracle, since OFaultSign uses the fresh randomness even if queried with the same message m. In fact, by allowing the same (m, n) query the hedged construction HFS degenerates to a deterministic scheme and thus the `SSND` or `LRB` type fault attacks would become possible as we saw in Sect. 2.3. For the same reason, once we allow the adversaries to mount a fault f_0 on (m, n) right before HE is invoked during the signing query, the security is completely compromised. We will revisit this issue as a negative result in Lemma 11.

Lemma 5 (UF-KOA → UF-fCMA). *Suppose* ID *is subset revealing. If* FS := **FS**[ID, CSF] *is* UF-KOA *secure, then* FS *is* $\{f_4, \ldots, f_{10}\}$-UF-fCMA *secure in the random oracle model. Concretely, given* $\{f_4, \ldots, f_{10}\}$*-adversary* $\mathcal{A}$ *against* FS *running in time* t*, and making at most* Q_s *queries to* OFaultSign*,* Q_h *queries to* H*, one can construct another adversary* $\mathcal{B}$ *against* FS *such that*

$$\mathbf{Adv}_{\mathsf{FS}}^{\mathsf{UF\text{-}fCMA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathsf{FS}}^{\mathsf{UF\text{-}KOA}}(\mathcal{B}) + \frac{(Q_s + Q_h)Q_s}{2^{\alpha-1}} + Q_s \cdot \epsilon_{HVZK},$$

where $\mathcal{B}$ *makes at most* Q_h *queries to its hash oracle, and has running time* t*. If we do not assume the subset-revealing property of* ID *and assume all the other conditions above, then we have that* FS *is* $\{f_5, f_6, f_8, f_9, f_{10}\}$-UF-fCMA *secure.*

Proof. We obtain the results by putting together Lemmas 6 to 10 for FS derived from subset-revealing ID, and Lemmas 6 to 8 for FS derived from non-subset-revealing ID. The proofs for these lemmas appear in the full version.

Our proof extends the UF-KOA-to-UF-CMA reduction in [55]. We show that UF-KOA security of a randomized Fiat–Shamir signature scheme FS can be broken by a successful UF-fCMA adversary $\mathcal{A}$ by constructing an adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine and simulates OFaultSign without using sk. We denote the random oracle and hash table in UF-fCMA experiment (resp. UF-KOA experiment) by H and HT (resp. H′ and HT′).

Preparation of Public Key. Upon receiving pk in the UF-KOA game, $\mathcal{B}$ forwards pk to $\mathcal{A}$.

Simulation of Random Oracle Queries. Upon receiving a random oracle query $\mathsf{H}(a, m, pk)$ from $\mathcal{A}$, $\mathcal{B}$ forwards the input (a, m, pk) to its own random oracle (H′ from the UF-KOA game) and provides $\mathcal{A}$ with the return value.

Simulation of Faulty Signing Queries. Suppose $\mathcal{A}$ chooses to use a fault function f_{j_i} in each faulty signing oracle query $i \in [Q_s]$. Then $\mathcal{B}$ answers i-th query by simulating the signature on m_i (or $\hat{m}_i$ if $\mathcal{A}$ chooses to apply f_5 to the message as hash input) using only pk as described in the lemma for f_{j_i}. Notice that the simulations are independent except they share the random oracle H and the set M storing (possibly faulty) queried messages. The hash input $(\hat{a}_i, \hat{m}_i, \hat{pk})$ in each signature simulation has at least $(\alpha - 1)$ bits of min-entropy (see the simulation in Lemma 7 in the full version). Because HT has at most $Q_h + Q_s$ existing entries, $\mathcal{B}$ fails to program the random oracle with probability at most $(Q_h + Q_s)/2^{\alpha-1}$ for each query. Moreover, $\mathcal{A}$ distinguishes the simulated signature from the one returned by the real signing oracle OFaultHSign with probability at most ϵ_{HVZK} for each query, since we use the special c/s/p-HVZK simulator $\mathcal{M}$ to derive a signature in every simulation.

Recalling that the number of signing queries is bounded by Q_s, and by a union bound, $\mathcal{A}$ overall distinguishes its simulated view from that in UF-fCMA game with probability at most

$$\frac{(Q_h + Q_s)Q_s}{2^{\alpha-1}} + Q_s \cdot \epsilon_{HVZK}.$$

Forgery. Suppose that at the end of the experiment $\mathcal{A}$ outputs its forgery (m^*, σ^*) that verifies and $m^* \notin M = \{\hat{m}_i : i \in [Q_s]\}$. (Recall from Fig. 4 that M stores possibly faulty messages $\hat{m}_i$ here instead of queried messages m_i, and thus $\mathcal{A}$ cannot win the game by simply submitting a signature on some faulty message that has been used for random oracle programming.) This means that the reconstructed transcript $(a^*, e^*, z^*) \leftarrow \mathsf{CDF}(\sigma^*, pk)$ satisfies

$$\mathsf{V}(a^*, e^*, z^*, pk) = 1 \quad \text{and} \quad \mathsf{H}(a^*, m^*, pk) = e^*.$$

Here we can guarantee that the $\mathrm{HT}[a^*, m^*, pk]$ has not been programmed by signing oracle simulation since m^* is fresh, i.e., $m^* \notin M$. Hence we ensure that $e^* = \mathrm{HT}[a^*, m^*, pk]$ has been directly set by $\mathcal{A}$, and $e^* = \mathrm{HT}'[a^*, m^*, pk]$ holds due to the hash query simulation. This implies (m^*, σ^*) is a valid forgery in the UF-KOA game as well.

4.2 Faulting Serialization Input/Output and Response Output

As a warm-up, we begin with the simplest analysis where faults do not have any meaningful impact on the signing oracle simulation. As we will show below, faulting with f_8, f_9 and f_{10} has no more security loss than the plain UF-KOA-to-UF-CMA reduction [55] does.

Lemma 6 (UF-KOA $\rightarrow$ $\{f_8, f_9, f_{10}\}$-UF-fCMA **(informal))**. *If* FS $:=$ **FS**[ID, CSF] *is* UF-KOA *secure, then* FS *is* $\{f_8, f_9, f_{10}\}$-UF-fCMA *secure in the random oracle model.*

Remark. As we briefly remarked after Definition 5, Lemma 6 holds for any instantiation of serialization as long as CSF and CDF are efficiently computable.

4.3 Faulting Challenge Hash Input

Recall that f_5 is the fault type that allows the attacker to fault the input (a, m, pk) to the hash function used to compute the challenge. Here we prove that randomized Fiat–Shamir signature schemes are secure against this type of fault attack, under the same conditions required for the plain UF-KOA-to-UF-CMA reduction [55]. Note that the proof of lemma below introduces a slight additional security loss compared to the plain UF-KOA-to-UF-CMA reduction because `set_bit` faults to the hash input increase the failure probability of random oracle programming.

Lemma 7 (UF-KOA $\rightarrow$ $\{f_5\}$-UF-fCMA **(informal))**. *If* FS $:=$ **FS**[ID, CSF] *is* UF-KOA *secure, then* FS *is* $\{f_5\}$-UF-fCMA *secure in the random oracle model.*

4.4 Faulting Challenge Hash Output

Recall that f_6 is the fault type that allows the attacker to fault the challenge hash function output, i.e., he can fault the bit string $e = \mathsf{H}(a, m, pk)$. We show that, unlike the fault with f_5, this type of fault does not introduce any additional loss in concrete security as long as the Resp function fails for invalid challenges outside the challenge space D_H.

Lemma 8 ($\mathsf{UF\text{-}KOA} \rightarrow \{f_6\}\text{-}\mathsf{UF\text{-}fCMA}$ **(informal)).** *If* $\mathsf{FS} := \mathbf{FS}[\mathsf{ID}, \mathsf{CSF}]$ *is* $\mathsf{UF\text{-}KOA}$ *secure, then* FS *is* $\{f_6\}\text{-}\mathsf{UF\text{-}fCMA}$ *secure in the random oracle model.*

Remarks. The above lemma relies on the fact that faulty $\tilde{e}_i$ is necessarily a "well-formed" challenge. For example, the challenge in some subset-revealing schemes has a specific structure (e.g., a list of pairs (c_i, p_i) where the c_i are distinct, as in $\mathsf{Picnic2}$). Computing Resp with a malformed challenge may cause σ to leak private information. This is why we required Definition 1 to have the condition that Resp validates $\tilde{e}_i \in D_h$ and otherwise returns $\perp$. This way, the signing algorithm does not leak information when a malformed challenge is input to the response phase, and eventually outputs $\perp$ as a signature because CSF is sound with respect to invalid response (see Definition 4).

Note that the proof can be generalized to the multi-bit fault setting. More specifically, the random oracle programming becomes unnecessary for output replacement faults (i.e., f_6 applies `set_bit` to every bit of e) because in that case the fault adversary would no longer be able to observe any relation between faulty $\tilde{e}_i$ and the original, unfaulty e.

4.5 Faulting Response Input

Next we prove the security against tampering function f_7, which lets an attacker fault the input (sk, e, St) to the Resp function. We only guarantee security assuming that the signature scheme is based on a subset revealing identification protocol (see Definition 3), and Resp and CSF make sure to rule out invalid challenge and response, respectively. As we will see in the next section, Picnic2 satisfies these additional properties.

Lemma 9 ($\mathsf{UF\text{-}KOA} \rightarrow \{f_7\}\text{-}\mathsf{UF\text{-}fCMA}$ **(informal)).** *Suppose* ID *is subset revealing. If* $\mathsf{FS} := \mathbf{FS}[\mathsf{ID}, \mathsf{CSF}]$ *is* $\mathsf{UF\text{-}KOA}$ *secure, then* FS *is* $\{f_7\}\text{-}\mathsf{UF\text{-}fCMA}$ *secure in the random oracle model.*

Remark. Intuitively, subset revealing ID schemes are secure against faults on St because the adversary only obtains what they could have computed by changing non-faulty signatures by themselves. On the other hand, the Schnorr signature scheme is not secure against tampering with f_7 and we describe concrete fault attacks in Lemma 14.

As we remarked after Definition 3, one can consider a highly inefficient version of Schnorr signature that enumerates all possible responses in St and opens one of them. In doing so, the Resp function avoids any algebraic operations involving

sk and ρ, and we can mitigate the risk of faulty response input attacks described above. This countermeasure is of course impractical since the challenge space is too large, but it illustrates a concrete case where subset revealing ID schemes are more robust against fault attacks, in our model.

4.6 Faulting Commitment Output

Recall that a fault of type f_4 allows the attacker to fault the output of $\mathsf{Com}(sk;\rho)$, the commitment function in the first step of the ID scheme. Here we prove that randomized Fiat–Shamir signature schemes are secure against this type of fault attack, under the same conditions as ones in Lemma 9.

Lemma 10 ($\mathsf{UF\text{-}KOA} \rightarrow \{f_4\}\text{-}\mathsf{UF\text{-}fCMA}$ **(informal)).** *Suppose* ID *is subset revealing. If* $\mathsf{FS} := \mathbf{FS}[\mathsf{ID},\mathsf{CSF}]$ *is* $\mathsf{UF\text{-}KOA}$ *secure, then* FS *is* $\{f_4\}\text{-}\mathsf{UF\text{-}fCMA}$ *secure in the random oracle model.*

4.7 Negative Results

Here we show that fault attacks of type f_0, f_2 and f_3 are not mitigated by the hedged construction for an ID scheme with the same properties as in Theorem 1.

Lemma 11. *There exist canonical* ID *schemes such that* $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID},\mathsf{CSF}],\mathsf{HE}]$ *is* $\mathsf{UF\text{-}CMNA}$*-secure, but not* $\{f_0\}\text{-}\mathsf{UF\text{-}fCMNA}$ *secure.*

Proof. We consider the Schnorr scheme that returns (e,z) as a signature for which $\mathbf{FS}[\mathsf{ID},\mathsf{CSF}]$ is known to be $\mathsf{UF\text{-}CMA}$ secure and therefore $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID},\mathsf{CSF}],\mathsf{HE}]$ is $\mathsf{UF\text{-}CMNA}$ secure due to Lemma 3. Our $\{f_0\}$-adversary's strategy is as follows. The adversary first calls $\mathsf{OFaultHSign}$ with some (m,n) without fault (i.e., $\phi = Id$) to obtain a legitimate signature (e,z). Next, the adversary calls $\mathsf{OFaultHSign}$ with $\phi = \texttt{flip_bit}_i$, $j = 0$ and (m',n), where m' is identical to m except at the i-th bit. This way, it can fault m' back to m before the invocation of HE and hence the signature is derived from the same ρ as in the previous query, while the challenge and response are different since $e' = \mathsf{H}(a,m',pk)$ and $z = \rho + e' \cdot sk \mod q$. Hence we can recover sk with the $\texttt{SSND}$ attack in Sect. 2.3 and break the scheme.

Lemma 12. *There exist canonical* ID *schemes such that* $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID},\mathsf{CSF}],\mathsf{HE}]$ *is* $\mathsf{UF\text{-}CMNA}$*-secure, but not* $\{f_3\}\text{-}\mathsf{UF\text{-}fCMNA}$ *secure.*

Proof. We describe a simple attack that works for the Picnic ID scheme. Recall that f_3 is applied to input of $\mathsf{Com}(sk;\rho)$. When querying $\mathsf{OFaultHSign}$, the attacker uses $\texttt{set_bit}$ to set the i-th bit of sk, denoted sk_i to 0, then observes whether the signature output is valid. If so, then the true value of sk_i is 0, and if not, then sk_i is one. By repeating this for each of the secret key bits, the entire key may be recovered. Some ID schemes may include internal checks and abort if some computations are detected to be incorrect relative to the public key, in this case the attacker checks whether $\mathsf{OFaultHSign}$ aborts.

Note that Lemma 12 only applies to ID schemes where sk is used by the Com function. For the Schnorr scheme and other so-called *input delayed protocols* [25], sk is only used by the Resp function. In this way subset-revealing ID schemes and input delayed ID schemes have the opposite behavior, since subset-revealing schemes do not use sk in the Resp function, but they must use it in the Com function.

The sensitivity of ephemeral randomness ρ in Schnorr-like schemes is well known, and once the attacker obtains sufficiently many biased signatures, the secret key can be recovered by solving the so-called *hidden number problem (HNP)* [21]. Previous works have shown that even a single-bit bias helps to recover sk by making use Bleichenbacher's solution to HNP [4,18]. However, the currently known algorithms for the HNP do not give an asymptotically efficient attack, they only reduce the concrete security of the scheme sufficiently to allow a practical attack on some parameter sets. For instance, with the current state-of-the-art algorithm based on Bleichenbacher's attack found in the literature [70, Theorem 2], one can practically break 1-bit biased signatures instantiated over 192-bit prime order groups, using $2^{29.6}$ signatures as input, and with $2^{29.6}$ space and $2^{59.2}$ time, which is tractable for computationally well-equipped adversaries as of today.

To attack Schnorr-like schemes with f_3, the adversary would instead target the randomness ρ to cause a single-bit bias in it, and this situation is essentially same as faulting with f_2. Such an attack would be also powerful enough to recover the entire signing key, which we describe below.

Lemma 13. *Relative to an oracle for the hidden number problem, there exist a non-subset revealing canonical* ID *scheme such that* $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}]$ *is* $\mathsf{UF\text{-}CMNA}$*-secure, but neither* $\{f_2\}$-$\mathsf{UF\text{-}fCMNA}$ *nor* $\{f_3\}$-$\mathsf{UF\text{-}fCMNA}$ *secure.*

Proof. We describe an attack that works for the Schnorr signature scheme. Recall that both f_2 and f_3 can tamper with ρ in Schnorr, as its St contains the randomness ρ. If f_2 or f_3 is `set_bit` and always targets at the most significant bit of ρ to fix its value, the attacker can introduce 1-bit bias in ρ.

Relative to an oracle for the HNP, the Schnorr scheme with unbiased ρ remains secure, however, the scheme with biased ρ is broken. We must assume here that the HNP oracle does not help an attacker break the Schnorr scheme with unbiased nonces (otherwise the Theorem is trivial). It is easy to see that the HNP with uniformly random nonces does not give a unique solution – the adversary is given a system of Q_s equations with $Q_s + 1$ unknowns, so a direct application of the HNP oracle does not help. However, there may be other ways to use the HNP oracle, so we must make the assumption.

For fault types f_7 and f_4, we have shown that $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}]$ is secure assuming ID is subset-revealing. The following two lemmas give counterexamples when ID is not subset revealing, showing that canonical ID schemes are not generically secure for faults f_7 and f_4.

Lemma 14. *There exist non-subset-revealing canonical* ID *schemes such that* $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}]$ *is* $\mathsf{UF\text{-}CMNA}$*-secure, but not* $\{f_7\}$-$\mathsf{UF\text{-}fCMNA}$ *secure.*

Proof. We describe two attacks that work for the Schnorr signature scheme.

- If f_7 is `set_bit` and targeted at sk, the adversary can use the strategy of Lemma 12 to learn each bit of sk by checking whether the faulty signatures pass verification.
- If f_7 is `flip_bit` and targeted at the most significant bit of $St = \rho$, the adversary obtains (e, z') such that $z' = e \cdot sk + f_7(\rho)$, and he can recover the "faulty" commitment $a' = [f_7(\rho)]G$. Recall that the non-faulty commitment $a = [\rho]G$ satisfies $\mathsf{H}(a, m, pk) = e$, so the adversary can learn 1-bit of ρ by checking whether $\mathsf{H}(a' + [2^{\ell_\rho - 1}]G, m, pk) = e$ or $\mathsf{H}(a' - [2^{\ell_\rho - 1}]G, m, pk) = e$ holds, where ℓ_ρ is the bit length of ρ. Since we now have the most significant bit of ρ, we use the same argument as in Lemma 13 to show the scheme is vulnerable to fault attacks.

Lemma 15. *There exist non-subset-revealing canonical* ID *schemes such that* $\mathbf{R2H}[\mathbf{FS}[\mathsf{ID}, \mathsf{CSF}], \mathsf{HE}]$ *is* UF-CMNA*-secure, but not* $\{f_4\}$-UF-fCMNA *secure.*

Proof. Recall that f_4 is applied to (a, St), the output of Com. In the Schnorr signature scheme, St contains the per-signature ephemeral value ρ, which is the output of the hedged extractor. Therefore, the same attack as described in Lemma 14 for f_7-faults can be mounted with an f_4-fault.

5 Analysis of XEdDSA

In this section we apply the results of Sect. 4 to the XEdDSA signature scheme. The scheme is presented in the full version [5]. The associated ID scheme is the Schnorr ID scheme (denoted ID-Schnorr). Then we define Schnorr $:=$ $\mathbf{FS}[\mathsf{ID\text{-}Schnorr}, \mathsf{CSF}]$ and XEdDSA $:=$ $\mathbf{R2H}[\mathsf{Schnorr}, \mathsf{HE}]$, where CSF returns (a, z). We start by establishing some well-known properties of ID-Schnorr. Proof is given in the full version [5]. As noted in Sect. 2 ID-Schnorr is not subset-revealing.

Lemma 16. ID-Schnorr *is perfect HVZK (therefore* $\epsilon_{HVZK} = 0$*) and has* 2λ *bits of min-entropy.*

UF-KOA *Security* Let $\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{Schnorr}}(\mathcal{A})$ be the (concrete) UF-KOA security of Schnorr against an adversary $\mathcal{A}$ running in time t. As non-hedged XEdDSA is identical to Schnorr in the UF-KOA setting, the concrete analysis for Schnorr of [55, Lemmas 3.5-3.7] and [63, Lemma 8] are applicable. We do not repeat those results here (as they are lengthy and don't add much to the present paper), but instead state our results in terms of $\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{Schnorr}}(\mathcal{A})$. We can now apply the results of Sect. 4.

Corollary 1. *XEdDSA is* $\{f_1, f_5, f_6, f_8, f_9, f_{10}\}$-UF-fCMNA *secure.*

Proof. We've shown above that ID-Schnorr is perfect HVZK (so $\epsilon_{HVZK} = 0$) and has $\alpha = 2\lambda$ bits of min-entropy. Then we can apply Theorem 1, to obtain

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{XEdDSA}}(\mathcal{A}) \leq 2\left(\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{Schnorr}}(\mathcal{B}) + \frac{(Q_s + Q_h)Q_s}{2^{2\lambda-1}}\right)$$

Remaining Fault Types. We now consider the faults of type f_0, f_2, f_3, f_4, and f_7 where we can't prove security. For each of these, we have given an attack elsewhere in the paper, for Schnorr signatures, but that also applies to XEdDSA. For type f_0 see Lemma 11, for types f_2 and f_3 see Lemma 13, for type f_4 see Lemma 15 and for type f_7 see Lemma 14.

6 Analysis of Picnic2

In this section we analyze the Picnic2 variant of the Picnic signature scheme using our formal model for fault attacks. Since Picnic is constructed from a subset-revealing ID scheme, more of the results from Sect. 4 apply, reducing our effort in this section. We use ID-Picnic2 to denote the ID scheme, and Picnic2 $:=$ **FS**[ID-Picnic2, CSF] and HS-Picnic2 $:=$ **R2H**[Picnic2, HE] to denote the randomized and hedged signature schemes. Proofs for this section, and details of the signature scheme are in the full version [5]. We begin with some general properties of Picnic2.

ID-Picnic2 *Is a Subset-Revealing ID Scheme.* Note that its St consists of $\{h_j, h'_j, \mathsf{seed}^*_j, \{\hat{z}_{j,\alpha}\}, \mathsf{state}_{j,i}, ,_{j,i}, \mathsf{msgs}_{j,i}\}_{j\in[M],i\in[n]}$ and Resp simply reveals a subset of it depending on a challenge $\mathcal{C}$ and $\mathcal{P}$.

The Picnic2 *Specification Is an Instance of* **R2H**. The specification recommends a hedging construction that is an instance of the **R2H** construction from Sect. 3. In this case, the salt and random seeds are derived deterministically from $sk\|m\|pk\|n$ where n is a 2λ-bit random value (acting as the nonce in the notation of Sect. 3). The function HE is instantiated with the SHA-3 based derivation function SHAKE. The security analysis in [72] applies to the randomized version of the signature scheme, so we must use Lemma 3 to establish UF-CMNA security of the hedged variant.

Lemma 17. *For security parameter λ,* ID-Picnic2 *has $\alpha \geq 2\lambda + 256$ bits of min-entropy.*

The next corollary shows that Picnic2 is secure against key-only attacks, and it follows from the unforgeability security proof of Picnic2 from [72].

Corollary 2. *The signature scheme* Picnic2 *is* UF-KOA *secure, when the hash functions H_0, H_1, H_2 and G are modeled as random oracles with 2λ-bit outputs, and key generation function* Gen *is (t, ϵ_{OW})-one-way.*

In particular, we have that

$$\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{Picnic2}}(\mathcal{A}) \leq \frac{3{Q_h}^2}{2^{2\lambda}} + 2\epsilon_{OW} + \frac{Q_h}{2^{\lambda}} \ .$$

Lemma 18. ID-Picnic2 *is a special c-HVZK proof, under the following assumptions: the hash functions H_0, H_1 and H_2 are modeled as random oracles, and*

the PRG is (t, ϵ_{PRG})-secure. Simulated transcripts are computationally indistinguishable from real transcripts, and all polynomial-time distinguishing algorithms succeed with probability at most

$$\epsilon_{HVZK} \leq (n+2)\tau \cdot \epsilon_{PRG} + \frac{q_0\tau + q_2 M}{2^\lambda} .$$

where q_0 and q_2 are the number of queries to H_0 and H_2, λ is the security parameter, and (M, n, τ) are parameters of the scheme.

We can now apply our results from Sect. 4.

Corollary 3. HS-Picnic2 *is* $\{f_1, f_4, \ldots, f_{10}\}$-UF-fCMNA *secure.*

Proof. Recall that by Corollary 2, Picnic2 is UF-KOA secure with

$$\mathbf{Adv}^{\mathsf{UF\text{-}KOA}}_{\mathsf{Picnic2}}(\mathcal{A}) \leq \frac{3{Q_h}^2}{2^{2\lambda}} + 2\epsilon_{OW} + \frac{Q_h}{2^\lambda}$$

and the min-entropy α is $2\lambda + 256$ as shown in Lemma 17.

We can apply Theorem 1, to obtain

$$\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HS\text{-}Picnic2}}(\mathcal{A}) \leq \frac{6{Q_h}^2}{2^{2\lambda}} + 4\epsilon_{OW} + \frac{2Q_h}{2^\lambda} + \frac{(Q_s + Q_h)Q_s}{2^{2\lambda+254}} + 2Q_s \cdot \epsilon_{HVZK},$$

where ϵ_{HVZK} is given in Theorem 18.

Fault type f_2 Recall that f_2 is a fault on ρ, the output of the hedged extractor. Intuitively, HS-Picnic2 is $\{f_2\}$-UF-fCMNA secure since ρ is not used directly, ρ is the list of seed^*_j values, which are used as input to a PRG when deriving the $\mathsf{seed}_{i,j}$ values. Applying a 1-bit fault to a seed^*_j value reduces the min-entropy by at most one bit, so only a small change to the security proof and analysis is required. Concretely we have:

Lemma 19. HS-Picnic2 *is* $\{f_2\}$-UF-fCMNA *secure.* $\mathbf{Adv}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HS\text{-}Picnic2}}(\mathcal{A})$ *is the same as given in Corollary 3, except that α is reduced by 1.*

Fault type f_3 Recall that f_3 faults are applied to $\mathsf{Com}(f_3(sk; \rho))$. By setting bits of sk, the attacker can recover sk with an IFA.

7 Concluding Remarks

This paper explored the effects of bit-tampering fault attacks on various internal values in hedged Fiat–Shamir signing operations, within the provable security methodology. Our security model is general enough to capture a large class of signatures, but also fine-grained enough to cover existing attacks surveyed in Sect. 2.3. We remark, however, that there are several more advanced, yet practically relevant fault types that are not covered by our model: (1) faulting

global parameters, (2) multiple bit and word faults, (3) faults within the Com and Resp functions, (4) multiple faults per signature query, and (5) persisting faults. A detailed discussion for each is given in the full version [5], to illustrate the limitations of our analysis. Each of these issues makes an interesting direction for future work.

Acknowledgments. This research was supported by: the Concordium Blockchain Research Center, Aarhus University, Denmark; the Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM); the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 803096 (SPEC); the Danish Independent Research Council under Grant-ID DFF-6108-00169 (FoCC). We thank anonymous reviewers for their valuable comments and suggestions.

References

1. Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From identification to signatures via the Fiat-Shamir transform: minimizing assumptions for security and forward-security. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 418–433. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_28
2. Alagic, G., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process (2019)
3. Ambrose, C., Bos, J.W., Fay, B., Joye, M., Lochter, M., Murray, B.: Differential attacks on deterministic signatures. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 339–353. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76953-0_18
4. Aranha, D.F., Fouque, P.-A., Gérard, B., Kammerer, J.-G., Tibouchi, M., Zapalowicz, J.-C.: GLV/GLS decomposition, power analysis, and attacks on ECDSA signatures with single-bit nonce bias. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 262–281. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_14
5. Aranha, D.F., Orlandi, C., Takahashi, A., Zaverucha, G.: Security of hedged Fiat-Shamir signatures under fault attacks. Cryptology ePrint Archive, Report 2019/956 (2019)
6. Austrin, P., Chung, K., Mahmoody, M., Pass, R., Seth, K.: On the impossibility of cryptography with tamperable randomness. Algorithmica **79**(4), 1052–1101 (2017)
7. Baert, M.: Ed25519 leaks private key if public key is incorrect #170. https://github.com/jedisct1/libsodium/issues/170
8. Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The sorcerer's apprentice guide to fault attacks. Proc. IEEE **94**(2), 370–382 (2006)
9. Barenghi, A., Pelosi, G.: A note on fault attacks against deterministic signature schemes. In: Ogawa, K., Yoshioka, K. (eds.) IWSEC 2016. LNCS, vol. 9836, pp. 182–192. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44524-3_11
10. Barthe, G., Dupressoir, F., Fouque, P.-A., Grégoire, B., Tibouchi, M., Zapalowicz, J.-C.: Making RSA–PSS provably secure against non-random faults. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 206–222. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44709-3_12

11. Bellare, M., et al.: Hedged public-key encryption: how to protect against bad randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_14
12. Bellare, M., Hoang, V.T.: Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_21
13. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_31
14. Bellare, M., Poettering, B., Stebila, D.: From identification to signatures, tightly: a framework and generic transforms. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 435–464. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_15
15. Bellare, M., Tackmann, B.: Nonce-based cryptography: retaining security when randomness fails. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 729–757. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_28
16. Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.Y.: High-speed high-security signatures. J. Cryptogr. Eng. **2**(2), 77–89 (2012). https://doi.org/10.1007/s13389-012-0027-1
17. Bindel, N., et al.: qTESLA. Technical report, National Institute of Standards and Technology. https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
18. Bleichenbacher, D.: On the generation of one-time keys in DL signature schemes. Presentation at IEEE P1363 Working Group Meeting (2000)
19. Boldyreva, A., Patton, C., Shrimpton, T.: Hedging public-key encryption in the real world. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 462–494. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_16
20. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_4
21. Boneh, D., Venkatesan, R.: Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 129–142. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_11
22. Brengel, M., Rossow, C.: Identifying key leakage of bitcoin users. In: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID 2018. LNCS, vol. 11050, pp. 623–643. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00470-5_29
23. Bruinderink, L.G., Pessl, P.: Differential fault attacks on deterministic lattice signatures. IACR TCHES **2018**(3), 21–43 (2018)
24. Chailloux, A.: Quantum security of the Fiat-Shamir transform of commit and open protocols. Cryptology ePrint Archive, Report 2019/699 (2019)
25. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved OR-composition of sigma-protocols. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016, Part II. LNCS, vol. 9563, pp. 112–141. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_5

26. Clavier, C.: Secret external encodings do not prevent transient fault analysis. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 181–194. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_13
27. Cohn-Gordon, K., Cremers, C.J.F., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the signal messaging protocol. In: EuroS&P, pp. 451–466. IEEE (2017)
28. Coron, J.-S., Mandal, A.: PSS is secure against random fault attacks. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 653–666. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_38
29. Daemen, J., Dobraunig, C., Eichlseder, M., Gross, H., Mendel, F., Primas, R.: Protecting against statistical ineffective fault attacks. Cryptology ePrint Archive, Report 2019/536
30. Damgård, I.: On Σ-protocols. http://www.cs.au.dk/~ivan/Sigma.pdf
31. Damgård, I., Faust, S., Mukherjee, P., Venturi, D.: Bounded tamper resilience: how to go beyond the algebraic barrier. J. Cryptol. **30**(1), 152–190 (2015). https://doi.org/10.1007/s00145-015-9218-0
32. De Meyer, L., Arribas, V., Nikova, S., Nikov, V., Rijmen, V.: M&M: masks and macs against physical attacks. IACR TCHES **1**, 25–50 (2019)
33. Dobraunig, C., Eichlseder, M., Gross, H., Mangard, S., Mendel, F., Primas, R.: Statistical ineffective fault attacks on masked AES with fault countermeasures. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 315–342. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_11
34. Dobraunig, C., Eichlseder, M., Korak, T., Mangard, S., Mendel, F., Primas, R.: SIFA: exploiting ineffective fault inductions on symmetric cryptography. IACR TCHES **3**, 547–572 (2018)
35. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. J. ACM **65**(4), 20:1–20:32 (2018)
36. fail0verflow: Console hacking 2010 - PS3 epic fail. 27th Chaos Communications Congress (2010)
37. Faonio, A., Nielsen, J.B., Simkin, M., Venturi, D.: Continuously non-malleable codes with split-state refresh. In: Preneel, B., Vercauteren, F. (eds.) ACNS 2018. LNCS, vol. 10892, pp. 121–139. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93387-0_7
38. Faonio, A., Venturi, D.: Efficient public-key cryptography with bounded leakage and tamper resilience. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 877–907. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_32
39. Faust, S., Pietrzak, K., Venturi, D.: Tamper-proof circuits: how to trade leakage for tamper-resilience. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 391–402. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22006-7_33
40. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
41. Fischlin, M., Günther, F.: Modeling memory faults in signature and authenticated encryption schemes. In: Jarecki, S. (ed.) CT-RSA 2020. LNCS, vol. 12006, pp. 56–84. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-40186-3_4
42. Fujisaki, E., Xagawa, K.: Public-key cryptosystems resilient to continuous tampering and leakage of arbitrary functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 908–938. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_33

43. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic Tamper-Proof (ATP) security: theoretical foundations for security against hardware tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_15
44. Goldreich, O.: Foundations of Cryptography, vol. 1. Cambridge University Press, New York (2000)
45. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In: 27th FOCS, pp. 174–187. IEEE Computer Society Press (1986)
46. Hazay, C., Lindell, Y.: Efficient Secure Two-Party Protocols. ISC. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14303-8
47. Huang, Z., Lai, J., Chen, W., Au, M.H., Peng, Z., Li, J.: Hedged nonce-based public-key encryption: adaptive security under randomness failures. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part I. LNCS, vol. 10769, pp. 253–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76578-5_9
48. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: 39th ACM STOC, pp. 21–30. ACM Press (2007)
49. Ishai, Y., Prabhakaran, M., Sahai, A., Wagner, D.: Private circuits II: keeping secrets in tamperable circuits. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 308–327. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_19
50. Joye, M., Tunstall, M.: Fault analysis in cryptography, Information Security and Cryptography, vol. 147. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29656-7
51. Karaklajic, D., Schmidt, J., Verbauwhede, I.: Hardware designer's guide to fault attacks. IEEE Trans. VLSI Syst. **21**(12), 2295–2306 (2013)
52. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: ACM CCS 2018, pp. 525–537. ACM Press (2018)
53. Kilian, J., Micali, S., Ostrovsky, R.: Minimum resource zero-knowledge proofs (extended abstract). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 545–546. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_47
54. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 552–586. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_18
55. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 33–61. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_2
56. Kim, Y., et al.: Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors. In: ISCA, pp. 361–372. IEEE Computer Society (2014)
57. Liu, F.-H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 517–532. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_30
58. Morita, H., Schuldt, J.C.N., Matsuda, T., Hanaoka, G., Iwata, T.: On the security of the schnorr signature scheme and dsa against related-key attacks. In: Kwon, S., Yun, A. (eds.) ICISC 2015. LNCS, vol. 9558, pp. 20–35. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30840-1_2

59. M'Raïhi, D., Naccache, D., Pointcheval, D., Vaudenay, S.: Computational alternatives to random number generators. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 72–80. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48892-8_6
60. Ohta, K., Okamoto, T.: On concrete security treatment of signatures derived from identification. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 354–369. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055741
61. Perrin, T.: The XEdDSA and VXEdDSA Signature Schemes. Signalrevision 1. https://signal.org/docs/specifications/xeddsa/
62. Poddebniak, D., Somorovsky, J., Schinzel, S., Lochter, M., Rosler, P.: Attacking deterministic signature schemes using fault attacks. In: Euro S&P 2018, pp. 338–352. IEEE (2018)
63. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000). https://doi.org/10.1007/s001450010003
64. Ravi, P., Jhanwar, M.P., Howe, J., Chattopadhyay, A., Bhasin, S.: Exploiting determinism in lattice-based signatures: practical fault attacks on pqm4 implementations of NIST candidates. In: Asia CCS 2019, pp. 427–440. ACM (2019)
65. Ristenpart, T., Yilek, S.: When good randomness goes bad: virtual machine reset vulnerabilities and hedging deployed cryptography. In: NDSS 2010. The Internet Society (2010)
66. Romailler, Y., Pelissier, S.: Practical fault attack against the Ed25519 and EdDSA signature schemes. In: FDTC 2017, pp. 17–24 (2017)
67. Samwel, N., Batina, L.: Practical fault injection on deterministic signatures: the case of EdDSA. In: Joux, A., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2018. LNCS, vol. 10831, pp. 306–321. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89339-6_17
68. Schmidt, B.: [curves] EdDSA specification. https://moderncrypto.org/mail-archive/curves/2016/000768.html
69. Schnorr, C.P.: Efficient signature generation by smart cards. J. Cryptol. **4**(3), 161–174 (1991). https://doi.org/10.1007/BF00196725
70. Takahashi, A., Tibouchi, M., Abe, M.: New bleichenbacher records: fault attacks on qDSA signatures. IACR TCHES **3**, 331–371 (2018)
71. Yen, S., Joye, M.: Checking before output may not be enough against fault-based cryptanalysis. IEEE Trans. Comput. **49**(9), 967–970 (2000)
72. Zaverucha, G., et al.: Picnic. Technical report, National Institute of Standards and Technology. https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions