Subversion-Resilient Signature Schemes[*]

Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi

Department of Computer Science, Sapienza University of Rome

May 29, 2015

Abstract

We provide a formal treatment of security of digital signatures against *subversion attacks* (SAs). Our model of subversion generalizes previous work in several directions, and is inspired by the proliferation of software attacks (e.g., malware and buffer overflow attacks), and by the recent revelations of Edward Snowden about intelligence agencies trying to surreptitiously sabotage cryptographic algorithms. The main security requirement we put forward demands that a signature scheme should remain unforgeable even in the presence of an attacker applying SAs (within a certain class of allowed attacks) in a fully-adaptive and *continuous* fashion. Previous notions—e.g., the notion of security against algorithm-substitution attacks introduced by Bellare *et al.* (CRYPTO '14) for symmetric encryption—were non-adaptive and non-continuous.

In this vein, we show both positive and negative results for the goal of constructing subversion-resilient signature schemes.

- **Negative results.** As our main negative result, we show that a broad class of randomized signature schemes is unavoidably insecure against SAs, even if using just a single bit of randomness. This improves upon earlier work that was only able to attack schemes with larger randomness space. When designing our new attack we consider undetectability as an explicit adversarial goal, meaning that the end-users (even the ones knowing the signing key) should not be able to detect that the signature scheme was subverted.
- **Positive results.** We complement the above negative results by showing that signature schemes with *unique* signatures are subversion-resilient against all attacks that meet a basic undetectability requirement. A similar result was shown by Bellare *et al.* for symmetric encryption, who proved the necessity to rely on *stateful* schemes; in contrast unique signatures are *stateless*, and in fact they are among the fastest and most established digital signatures available.

  We finally show that it is possible to devise signature schemes secure against arbitrary tampering with the computation, by making use of an un-tamperable cryptographic reverse firewall (Mironov and Stephens-Davidowitz, EUROCRYPT '15), i.e., an algorithm that "sanitizes" any signature given as input (using only public information). The firewall we design allows to successfully protect so-called re-randomizable signature schemes (which include unique signatures as special case).

As an additional contribution, we extend our model to consider multiple users and show implications and separations among the various notions we introduced. While our study is mainly theoretical, due to its strong practical motivation, we believe that our results have important implications in practice and might influence the way digital signature schemes are selected or adopted in standards and protocols.

[*] Partially supported by the European Commission (Directorate-General Home Affairs) under the GAINS project HOME/2013/CIPS/AG/4000005057, and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644666.

Contents

1 Introduction

Balancing national security interests with the rights to privacy of lawful citizen is always a daunting task. It has been particularly so in the last couple of years after the revelations of Edward Snowden [PLS13, BBG13, Gre14] that have evidenced a massive collection of metadata and other information perpetrated by several intelligence agencies. It is now clear that intelligence operators were not just interested in collecting and mining information but they also actively deployed malware, exploited zero-day vulnerabilities, and carried out active attacks against standard protocols. In addition, it appears some cryptographic protocol specifications were modified to embed backdoors.

Whether this activity was effective or even allowed by the constitution is open to debate and it is indeed being furiously discussed among policy makers, the public, and the intelligence community. Ultimately, a balance between security and privacy must be found for a free and functioning society.

The ability of substituting a cryptographic algorithm with an altered version was first considered formally by Young and Yung (extending previous works of Simmons on subliminal channels [Sim83, Sim84]), who termed this field *kleptography* [YY96, YY97]. The idea is that the attacker surreptitiously modifies a cryptographic scheme with the intent of subverting its security. This research area has recently been revitalized by Bellare *et al.* [BPR14] who considered encryption algorithms with the possibility of mass surveillance under the algorithm-substitution attack. They analyzed the possibility of an intelligence agency substituting an encryption algorithm with the code of an alternative version that undetectably reveals the secret key or the plaintext. What they uncovered is that any randomized and stateless encryption scheme would fall to generic algorithm-substitution attacks. The only way to achieve a meaningful security guarantee (CPA-security) is to use a nonce-based encryption that must keep state. Unfortunately, only stateless schemes are deployable effectively with the current network technology and indeed all deployed encryption algorithms are in this class.

In this paper we analyze digital signature schemes under the so-called *subversion attacks* (SAs), that in particular include algorithm-substitution and kleptographic attacks as a special case, but additionally cover more general malware and virus attacks (see below). Unlike encryption, we show positive results and truly efficient schemes that provide the strongest security guarantee and can thus be deployed within real systems. We stress that our intention is not to propose schemes that can be abused by criminals to avoid monitoring. We are motivated by pure scientific curiosity and aspire to contribute to an active field of research.

1.1 Our Results and Techniques

We introduce a new and generic framework and definitions for subversions of digital signatures. In the standard black-box setting, a signature scheme should remain unforgeable even against an adversary able to obtain signatures on (polynomially many) chosen messages. Our security definitions empower the adversary with the ability of *continuously* subverting the signing algorithm within a class $\mathcal{A}$ of allowed SAs. For each chosen subversion in the class, the adversary can access an oracle that answers (polynomially many) signature queries using the subverted signature algorithm. Importantly, the different subversions can be chosen in a fully-adaptive manner possibly depending on the target verification key of the user.

We believe our model is very general and flexible, as it nicely generalizes previous models and definitions. First off, when the class $\mathcal{A}$ consists of a set of algorithms containing a secretly embedded backdoor, and in case the adversary is restricted to non-adaptively choose only a single subversion algorithm from this class, we obtain the setting of algorithm-substitution and kleptographic attacks as a special case. However, we note that the above definition is far more general as it covers (fully-adaptive and continuous) *tampering with the computation* performed by the signing algorithm (within the class $\mathcal{A}$). This models, for instance, a machine running a signature software infected by a malware (e.g., via a buffer overflow attack [One96, Fry00, PB04]); we also obtain memory and randomness tampering (see Section 1.3) as a special case. We refer the reader to Section 3.1 (where we introduce our model formally) for a more throughout discussion.

Clearly, without making any restriction on the class $\mathcal{A}$ (or without making additional assumptions) there is no hope for security: An arbitrary subverted signature algorithm could, for instance, just ignore all inputs and output the secret key. In this paper we investigate two approaches to tackle attacks of this sort and obtain positive results.

- **Limiting the adversarial power.** We consider a setting where the adversarial goal is to subvert the signature algorithm in a way that is *undetectable* to the end-user (or at least allows to maintain plausible deniability). For instance the simple attack above—where the subversion outputs the secret key—is easily detectable given only public information. As we show in Section 5, requiring that the class $\mathcal{A}$ satisfies a basic undetectability requirement already allows for interesting positive results.

- **Using a Reverse Firewall.** In Section 6 we show that security against *arbitrary* tampering with the computation can be achieved, by making the additional assumption of an un-tamperable cryptographic reverse firewall (RF) [MS15]. Roughly, a RF takes as input a message/signature pair and is allowed to "sanitize" the input signature using only *public information.*

A more detailed description of our techniques follows.

Negative results. We define what it means for a class $\mathcal{A}$ of SAs to be (efficiently) undetectable; roughly this means that a user, given polynomially many queries, cannot distinguish the output of the genuine signature algorithm from the output of the subverted algorithm. See Section 3.2 for a precise definition. Our definitions of undetectability are similar in spirit to the ones put forward by [BPR14] for the setting of symmetric encryption. Importantly we distinguish the case where the user (trying to detect the attack) knows only public or private information (i.e., it knows the secret key).[1]

Next, we explore the possibility of designing classes of SAs that are (even secretly) undetectable and yet allow for complete security breaches. This direction was already pursued by Bellare *et al.*, who showed that it is possible to stealthily bias the random coins of sufficiently randomized symmetric encryption schemes in a way that allows to extract the secret key after observing a sufficient number of (subverted) ciphertexts. As a first negative result, we explain how to adapt the "biased randomness attack" of [BPR14] to the case of signature schemes.

The above generic attack requires that the signature scheme uses a minimal amount of randomness (say, 7 bits). This leaves the interesting possibility that less randomized schemes (such as the Katz-Wang signature scheme [KW03], using only one bit of randomness) might be secure. In Section 4, we present a new attack showing that this possibility is vacuous: Our attack allows to stealthily bias the randomness in a way that later allows to extract the signing key—regardless of the number of random bits required by the scheme—assuming that the targeted signature scheme is *coin-extractable.* The latter roughly means that the random coins used for generating signatures can be extracted efficiently from the signature itself; as we discuss in more detail in Section 4.2 many real schemes (including Katz-Wang) are coin-extractable.

Positive results. We complement the above negative results by showing that fully deterministic schemes with *unique*[2] signatures are subversion-resilient against the class of SAs that satisfies the so-called verifiability condition; this essentially means that—*for all values in the message space*—signatures produced by the subverted signature algorithm should (almost always) verify correctly under the target verification key. Note that both attacks mentioned above fall into this category.

Clearly, the assumption that the verifiability condition should hold for all messages is quite a strong one. In fact, as shown very recently by Degabriele *et al.* [DFP15] for the case of symmetric encryption, it is not hard to show that such limitation is inherent: No (even deterministic) scheme can be subversion-resilient against the class of SAs that meets the verifiability condition for all but a negligible fraction of the messages. (See Section 1.3 for more details.) As our main positive result, we provide an approach how to bypass the above limitation and achieve the ambitious goal of protecting signature schemes against *arbitrary* SAs, relying on a cryptographic reverse firewall. The latter primitive was recently introduced in [MS15] to model security of arbitrary two-party protocols run on machines possibly corrupted by a virus. On a high level a RF for a signature scheme is a trusted piece of software taking as input a message/signature pair (m, σ) and some *public* state, and outputting a "patched" signature (m, σ'); the initial state of the firewall is typically a function of the verification key vk. A good RF should maintain functionality, meaning that whenever the input is a valid message/signature pair the patched signature (almost always) verifies correctly under the target verification key. Moreover, we would like the firewall to preserve unforgeability; this means that patched signatures (corresponding to signatures generated via the subverted signing algorithm) should not help an adversary to forge on a fresh message.

We prove that every signature scheme that is re-randomizable (as defined in [Wat05]) admits

[1] As we show, secret and public undetectability are *not* equivalent, in that there exist natural classes of SAs that are publicly undetectable but secretly detectable.

[2] A signature scheme is unique if for a honestly generated verification key it is hard to find two *distinct* valid signatures of a given message.

a RF that preserves unforgeability against arbitrary SAs. Re-randomizable signatures admit an efficient algorithm ReRand that takes as input a tuple (m, σ, vk) and outputs a signature σ' that is distributed uniformly over the set of all valid signatures on message m (under vk); unique signatures, for instance, are re-randomizable. Upon input a pair (m, σ) our firewall uses the public state to verify (m, σ) is valid under vk, and, in case the test passes, it runs ReRand on (m, σ) and outputs the result. Otherwise the firewall simply returns an invalid symbol $\perp$ and *self-destructs*, i.e., it stops processing any further query.[3] The latter is a requirement that we prove to be unavoidable: No RF can at the same time maintain functionality and preserve unforgeability of a signature scheme without the self-destruct capability.

We remark that our results and techniques for the setting of RFs are incomparable to the ones in [MS15]. The main result of Mironov and Stephens-Davidowitz is a compiler that takes as input an arbitrary two-party protocol and outputs a functionally equivalent (but different) protocol that admits a RF preserving both functionality and security (whatever security property the original protocol has). Instead, we model directly security of RFs for signatures schemes in the game-based setting; while our goal is more restricted (in that we only design RFs for signatures), our approach results in much more efficient and practical solutions.

Multi-user setting. Our discussion so far considered a single user. In Appendix A we discuss how our models and results can be extended to the important (and practically relevant) multi-user scenario. In particular, similarly to [BPR14], we generalize our undetectability and security notions to a setting with $u > 1$ users, where each user has a different signing/verification key.

As we argue, security in the single-user setting already implies security in the multi-user setting (by a standard hybrid argument). This does not hold for undetectability, as there exists classes of SAs that are undetectable by a single user but can be efficiently detected by more than one user. However, as we show in Appendix B, the concrete attacks analysed in Section 4 can be modified to remain undetectable even with multiple users.

1.2 Impact

Our study has strong implications in practice and might influence the way digital signature schemes are selected or adopted in standards and protocols. A subverted signature scheme is arguably even more deceitful and dangerous in practice than subverted encryption. Indeed, it is well-known that authenticated encryption must involve digital certificates that are signed by Certification Authorities (CAs). If a CA is using a subverted signature scheme, it is reasonable to expect the signing key will eventually be exposed. With knowledge of the signing key, it is possible to impersonate any user and carry out elementary man-in-the-middle attacks. This renders the use of any type of encryption utterly pointless and underlines the important role played by signatures in the context of secure communications.

Unfortunately, signature schemes currently employed to sign digital certificates, or used in protocols such as OTR, TLS/SSL, SSH, etc., are all susceptible to a subversion attack and their use should possibly be discontinued. The positive news however is that there already exist signature schemes that are subversion-resilient and they are efficient and well-established. This is in contrast with encryption where *good* schemes are not deployable in all contexts since they require retention of state information (see [BPR14]).

[3] This can be implemented, for instance, by having the public state include a single one-time writable bit used to signal a self-destruct took place.

1.3 Related Work

Sabotage of cryptographic primitives before and during their deployment has been the focus of extensive research over the past years. We briefly review the main results below.

Subliminal channels and backdoored implementations. After their introduction, the potential of subliminal channels has been explored in several works (e.g., [Des88a, Des88b, BDI$^+$99]); this line of research lead for instance to the concept of divertible protocols, that are intimately related to reverse firewalls.

The setting of backdoored implementations has also been the focus of extensive research. This includes, in particular, the realm of kleptography and SETUP attacks (see [YY04] for a survey). In recent work, Dodis *et al.* [DGG$^+$15] provide a formal treatment of trapdoored pseudorandom generators (building on previous work of Vazirani and Vazirani [VV83]); this setting is of particular importance, given the potential sabotage of the NIST Dual EC PRG [NIS07].

We refer the reader to [SFKR15] for a complete taxonomy of these (and more) types of attacks.

Input-triggered subversions. In a very recent paper, Degabriele, Farshim and Poettering (DFP) [DFP15] pointed out some shortcomings of the Bellare-Patterson-Rogaway (BPR) [BPR14] security model for subversion resilience of symmetric encryption schemes. Consider the class of SAs that upon input a secret (trapdoor) message m^* outputs the secret key, but otherwise behaves like the genuine encryption algorithm. Clearly this class of SAs allows for a complete breach of security, yet it will be undetectable by the users as without knowing the trapdoor there is only a negligible chance to query the secret message m^* and check if the signature algorithm was subverted (at least if the message space is large enough). As a consequence, it is impossible to prove subversion resistance against such "input-triggered" subversions in the BPR model.

The solution proposed by DFP is to simply modify the definition of undetectability so that the adversary (and not the user) specifies the input messages to the (potentially subverted) encryption algorithm, whereas the goal of the user is to detect the attack given access to the transcript of all queries made by the adversary (and answers to these queries). Hence, a scheme is said to be subversion-resilient if there exists a fixed polynomial-time test algorithm such that either a subversion attack cannot be detected efficiently but it does not leak any useful information, or it is possible to efficiently detect that the system was subverted.[4]

It is possible to make a similar change as in [DFP15] and adapt the DFP model to signature schemes. The end result would share some similarities with our approach using cryptographic RFs;[5] however, our framework provides notable advantages. First, note that the DFP model does not provide any guarantee against SAs that are efficiently detectable, whereas our RF model explicitly accounts for the actions to be taken after an attack is detected; this is particularly relevant for signature schemes where our generic attack uncovered the necessity of a self-destruct capability. Second, the polynomial-time detection test in DFP is performed directly by the user since it requires knowledge of the secret key. This is problematic in practice since often the user's machine is completely compromised; instead, in our framework, a cryptographic RF for a signature scheme relies only on public information and could easily be located on a (trusted) external proxy.

[4] For instance, in case of the attack outlined above, the polynomial-time test could simply decrypt the ciphertext and check the outcome matches the input message.

[5] On a high level, one can interpret the polynomial-time test as playing the role of the reverse firewall.

Tampering attacks. A related line of research analyses the security of cryptosystems against tampering attacks. Most of these works are restricted to the simpler setting of memory tampering (sometimes known as related-key security), where only the secret key of a targeted cryptoscheme is subject to modification. By now we know several concrete primitives that remain secure against different classes of memory-tampering attacks, including pseudorandom functions and permutations [BK03, Luc04, BC10, AFPW11, BCM11], pseudorandom generators and hard-core bits [GL10], hash functions [GOR11], public-key encryption [AHI11, Wee12], identification and digital signature schemes [KKS11, DFMV13]. Elegant generic compilers are also available, relying on so-called tamper-resilient encodings and non-malleable codes (see, among others, [GLM$^+$04, DPW10, LL12, FMNV14, FMVW14, ADL14, JW15, DLSZ15, AGM$^+$15, FMNV15, DFMV15]).

The setting of randomness tampering, where the random coins of a cryptographic algorithm are subject to tampering, has also been considered. For instance Austrin *et al.* [ACM$^+$14] consider so-called p-tampering attacks, that can efficiently tamper with each bit of the random tape with probability p. In this setting they show that some cryptographic tasks (including commitment schemes and zero-knowledge protocols) are impossible to achieve, while other tasks (in particular signature and identification schemes) can be securely realized.

Yet another related setting is that of tampering attacks against gates and wires in the computation of a cryptographic circuit, and the design of tamper-proof circuit compilers [IPSW06, FPV11, DK12, KT13, DK14, GIP$^+$14].

2 Preliminaries

2.1 Notation

For a string x, we denote its length by $|x|$; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the number of elements in $\mathcal{X}$. When x is chosen randomly in $\mathcal{X}$, we write $x \leftarrow\!\!\$\, \mathcal{X}$. When A is an algorithm, we write $y \leftarrow \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is randomized, then y is a random variable and $\mathsf{A}(x; r)$ denotes a run of A on input x and randomness r. An algorithm A is *probabilistic polynomial-time* (PPT) if A is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in at most $\mathit{poly}(|x|)$ steps.

We denote with $\kappa \in \mathbb{N}$ the security parameter. A function $\mathit{negl} : \mathbb{N} \to \mathbb{R}$ is negligible in the security parameter (or simply negligible) if it vanishes faster than the inverse of any polynomial in κ, i.e. $\mathit{negl}(\kappa) = \kappa^{-\omega(1)}$.

The statistical distance between two random variables $\mathbf{A}$ and $\mathbf{B}$ defined over the same domain $\mathcal{D}$ is defined as $\mathbb{SD}(\mathbf{A}; \mathbf{B}) = \frac{1}{2} \sum_{x \in \mathcal{D}} |\mathbb{P}[\mathbf{A} = x] - \mathbb{P}[\mathbf{B} = x]|$. We rely on the following lemma:

Lemma 1. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be a pair of random variables, and* E *be an event defined over the probability space of* $\mathbf{A}$ *and* $\mathbf{B}$*. Then,*

$$\mathbb{SD}(\mathbf{A}; \mathbf{B}) \leq \mathbb{SD}(\mathbf{A}; \mathbf{B} | \neg E) + \mathbb{P}[E].$$

2.2 Signature Schemes

A signature scheme is a triple of algorithms $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ specified as follows: (i) KGen takes as input the security parameter κ and outputs a verification/signing key pair $(vk, sk) \in \mathcal{VK} \times \mathcal{SK}$, where $\mathcal{VK} := \mathcal{VK}_\kappa$ and $\mathcal{SK} := \mathcal{SK}_\kappa$ denote the sets of all verification and secret keys produced by $\mathsf{KGen}(1^\kappa)$; (ii) Sign takes as input the signing key $sk \in \mathcal{SK}$, a message $m \in \mathcal{M}$ and random coins $r \in \mathcal{R}$, and outputs a signature $\sigma \in \Sigma$; (iii) Vrfy takes as input the verification key $vk \in \mathcal{VK}$ and a pair (m, σ), and outputs a decision bit that equals 1 iff σ is a valid signature for message m under key vk.

Correctness of a signature scheme says that verifying honestly generated signatures always works (with overwhelming probability over the randomness of all involved algorithms).

Definition 1 (Correctness)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that $\mathcal{SS}$ satisfies ν_c-correctness if for all $m \in \mathcal{M}$

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \mathsf{Sign}(sk, m))) = 1 : (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)\right] \geq 1 - \nu_c,$$

where the probability is taken over the randomness of KGen, Sign, and Vrfy.

The standard notion of security for a signature scheme demands that no PPT adversary given access to a signing oracle returning signatures for arbitrary messages, can forge a signature on a "fresh" message (not asked to the signing oracle).

Definition 2 (Existential Unforgeability)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that $\mathcal{SS}$ is (t, q, ε)-existentially unforgeable under chosen-message attacks ((t, q, ε)-ufcma in short) if for all PPT adversaries A running in time t it holds:

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m^*, \sigma^*)) = 1 \wedge m^* \notin \mathcal{Q} : \ (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa); (m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{Sign}(sk, \cdot)}(vk)\right] \leq \varepsilon,$$

where $\mathcal{Q} = \{m_1, \ldots, m_q\}$ denotes the set of queries to the signing oracle. Whenever $\varepsilon(\kappa) = negl(\kappa)$ and $q = poly(\kappa)$, we simply say that $\mathcal{SS}$ is ufcma.

Unique signatures. For our positive results we rely on so called *unique* signatures, that we define next. Informally a signature scheme is unique if for any message there is a single signature that verifies w.r.t. a honestly generated verification key.

Definition 3 (Uniqueness)**.** Let $\mathcal{SS}$ be a signature scheme. We say that $\mathcal{SS}$ satisfies ν_u-uniqueness if $\forall m \in \mathcal{M}$ and $\forall \sigma_1, \sigma_2$ s.t. $\sigma_1 \neq \sigma_2$

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \sigma_1)) = \mathsf{Vrfy}(vk, (m, \sigma_2)) = 1 : (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)\right] \leq \nu_u,$$

where the probability is taken over the randomness of the verification and key generation algorithms.

Full Domain Hash signatures with trapdoor permutations, for instance RSA-FDH [BR96], are unique. Sometimes unique signatures are also known under the name of *verifiable unpredictable functions* (VUFs).[6] Known constructions of VUFs exist based on strong RSA [MRV99], and on several variants of the Diffie-Hellman assumption in bilinear groups [Lys02, Dod03, DY05, ACF14, Jag15].

2.3 Pseudorandom Functions

Let $F : \{0,1\}^\kappa \times \mathcal{X} \to \mathcal{Y}$ be an efficient keyed function, where $\mathcal{X}$ and $\mathcal{Y}$ denote the domain and the range of F. Denote by $\mathcal{F}$ the set of all functions mapping $\mathcal{X}$ into $\mathcal{Y}$.

Definition 4 (Pseudorandom function)**.** A function $F : \{0,1\}^\kappa \times \mathcal{X} \to \mathcal{Y}$ is a (t, q, ε)-secure pseudorandom function (PRF), if for all adversaries D running in time at most t we have

$$\left| \underset{s \leftarrow\$ \{0,1\}^\kappa}{\mathbb{P}} \left[\mathsf{D}^{F_s(\cdot)}(1^\kappa) = 1\right] - \underset{f \leftarrow\$ \mathcal{F}}{\mathbb{P}} \left[\mathsf{D}^{f(\cdot)}(1^\kappa) = 1\right] \right| \leq \varepsilon,$$

where D asks at most q queries to its oracle.

[6] Strictly speaking, VUFs satisfy a stronger requirement—namely the uniqueness property holds even for maliciously generated verification keys; the weak variant above is sufficient for the results of this paper.

3 Subverting Signatures

We proceed to define what it means for an adversary B to subvert a signature scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$. We model subversion as the ability of the adversary to replace the genuine signing algorithm with a different algorithm within a certain class $\mathcal{A}$ of Subversion Attacks (SAs). A subversion of $\mathcal{SS}$ is an algorithm $\widetilde{\mathsf{A}} \in \mathcal{A}$, specified as follows.

- Algorithm $\widetilde{\mathsf{A}}(\cdot, \cdot; \cdot)$ takes as input a message $m \in \mathcal{M}$, a signing key $sk \in \mathcal{SK}$, random coins $r \in \mathcal{R}$, and outputs a subverted signature $\widetilde{\sigma} \in \Sigma$, where $\widetilde{\sigma} := \widetilde{\mathsf{A}}(sk, m; r)$. Notice that algorithm $\widetilde{\mathsf{A}}$ is completely arbitrary, with the only restriction that it maintains the same input-output interfaces as the original signing algorithm.

In particular, algorithm $\widetilde{\mathsf{A}}$ can hard-wire arbitrary auxiliary information chosen by the adversary, which we denote by a string $\alpha \in \{0,1\}^*$. In general we also allow algorithm $\widetilde{\mathsf{A}}$ to be *stateful*, even in case the original signing algorithm is not, and we denote the corresponding state by $\tau \in \{0,1\}^*$; the state is only used internally by the subverted algorithm and never outputted to the outside.

In Section 3.1 we define what it means for a signature scheme to be secure against a certain class of SAs. In Section 3.2 we define what it means for a class of SAs to be *undetectable* by a user. Some of our definitions are similar in spirit to the ones put forward in [BPR14], except that our modelling of subversion is more general (see below for a more detailed comparison).

3.1 Impersonation

We consider two security definitions, corresponding to different adversarial goals. In the first definition, it is required that an adversary B having access to polinomially many subversion oracles chosen adaptively (possibly depending on the user's verification key), cannot distinguish signatures produced via the standard signing algorithm from subverted signatures.

Definition 5 (Indistinguishability against SAs)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{SS}$ is (t, n, q, ε)-indistinguishable w.r.t *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B running in time t, we have $\left|\mathbb{P}\left[\mathsf{B} \text{ wins}\right] - \frac{1}{2}\right| \leq \varepsilon(\kappa)$ in the following game:

1. The challenger runs $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, samples $b \leftarrow\!\!\$\ \{0,1\}$, and gives vk to B.
2. The adversary B can ask the following two types of queries; the queries can be specified adaptively and in an arbitrary order:
   - Choose an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, for $j \in [n]$, and give it to the challenger.
   - Forward a pair $(j, m_{i,j})$ to the challenger, where $i \in [q]$ and $j \in [n]$. The answer to each query depends on the value of the secret bit b. In particular, if $b = 1$, the output is $\sigma_{i,j} \leftarrow \mathsf{Sign}(sk, m_{i,j})$; if $b = 0$, the output is $\widetilde{\sigma}_{i,j} \leftarrow \widetilde{\mathsf{A}}_j(sk, m_{i,j})$.
3. Finally, B outputs a value $b' \in \{0,1\}$; we say that B wins iff $b' = b$.

Whenever $\varepsilon(\kappa) = negl(\kappa), q = poly(\kappa)$, and $n = poly(\kappa)$ we simply say that $\mathcal{SS}$ is indistinguishable against continuous $\mathcal{A}$-SAs.

We also consider an alternative (strictly weaker—cf. Appendix A.3) definition, where the goal of the adversary is now to forge a signature on a "fresh" message (not asked to any of the oracles).

Definition 6 (Impersonation against SAs)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{SS}$ is (t, n, q, ε)-hard to impersonate w.r.t. *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B running in time t, we have $\mathbb{P}[\mathsf{B} \text{ wins}] \leq \varepsilon(\kappa)$ in the following game:

1. The challenger runs $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, and gives vk to B.
2. The adversary B is given oracle access to $\mathsf{Sign}(sk, \cdot)$. Upon input the i-th query m_i, this oracle returns $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$; let $\mathcal{Q} = \{m_1, \ldots, m_q\}$ be the set of all queried messages.
3. For each $j \in [n]$, the adversary B can adaptively choose an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$. For each algorithm, B is given oracle access to $\widetilde{\mathsf{A}}_j(sk, \cdot)$. Upon input a message $\widetilde{m}_{i,j}$, the oracle returns $\widetilde{\sigma}_{i,j} \leftarrow \widetilde{\mathsf{A}}_j(sk, \widetilde{m}_{i,j})$; let $\widetilde{\mathcal{Q}}_j = \{\widetilde{m}_{1,j}, \ldots, \widetilde{m}_{q,j}\}$ be the set of all queried messages to the oracle $\widetilde{\mathsf{A}}_j$.
4. Finally, B outputs a pair (m^*, σ^*); we say that B wins iff $\mathsf{Vrfy}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \widetilde{\mathcal{Q}}$, where $\widetilde{\mathcal{Q}} := \bigcup_{j=1}^n \widetilde{\mathcal{Q}}_j$.

Whenever $\varepsilon(\kappa) = \mathit{negl}(\kappa)$, $q = \mathit{poly}(\kappa)$, and $n = \mathit{poly}(\kappa)$ we simply say that $\mathcal{SS}$ is hard to impersonate against continuous $\mathcal{A}$-SAs.

Some remarks on the above definitions are in order.

- First, note that it is impossible to prove that a signature scheme $\mathcal{SS}$ satisfies Definition 5 (and consequently Definition 6) for an *arbitrary* class $\mathcal{A}$, without making further assumptions.[7] To see this, consider the simple algorithm that ignores all inputs and outputs the secret key.[8]
- We observe that continuous $\mathcal{A}$-SAs security, implies security against continuous tampering attacks with the secret key. This can be seen by considering a class of algorithms $\mathcal{A}_{\mathsf{key}} = \{\widetilde{\mathsf{A}}_f\}_{f \in \mathcal{F}}$, where $\mathcal{F}$ is a class of functions such that each $f \in \mathcal{F}$ has a type $f : \mathcal{SK} \to \mathcal{SK}$, and for all $f \in \mathcal{F}$, $m \in \mathcal{M}$ and $r \in \mathcal{R}$ we have that $\widetilde{\mathsf{A}}_f(\cdot, m; r) := \mathsf{Sign}(f(\cdot), m; r)$.[9]
- It is useful to compare Definition 5 to the security definition against algorithm-substitution attacks given in [BPR14] (for the case of symmetric encryption). In the language of Bellare *et al.* [BPR14], a subversion of a signature scheme would be a triple of algorithms $\widetilde{\mathcal{SS}} = (\widetilde{\mathsf{KGen}}, \widetilde{\mathsf{Sign}}, \widetilde{\mathsf{Vrfy}})$, where in the security game $\widetilde{\mathsf{KGen}}$ is run by the challenger in order to obtain a trapdoor $\alpha \in \{0,1\}^*$ and some initial state $\tau \in \{0,1\}^*$ which are both hard-wired in the algorithm $\widetilde{\mathsf{Sign}} := \widetilde{\mathsf{Sign}}_{\alpha,\tau}$ (and given to B).[10]

  The above setting can be cast in our framework by considering the class of SAs $\mathcal{A}_{\mathsf{BRP14}} := \{\widetilde{\mathsf{A}}_{\alpha,\tau} : (\alpha, \tau) \leftarrow \widetilde{\mathsf{KGen}}(1^\kappa)\}$, and by setting $n = 1$ in Definition 5. Our definition is more general, as it accounts for arbitrary classes of SAs and moreover allows B to subvert a user's algorithm continuously and in a fully-adaptive fashion (possibly depending on the target verification key).

[7] Looking ahead, one of our positive results achieves security w.r.t. arbitrary SAs assuming the existence of a cryptographic reverse firewall. See Section 6.

[8] In case the secret key is too long, one can make the algorithm stateful so that it outputs a different chunk of the key at each invocation. Alternatively, consider the class of algorithms $\{\widetilde{\mathsf{A}}_{\bar{m}}\}_{\bar{m} \in \mathcal{M}}$ that always outputs a signature $\bar{\sigma}$ on $\bar{m}$; obviously this subversion allows to forge on $\bar{m}$ without explicitly querying the message to any of the oracles.

[9] It is worth noting that already for $n = 1$ Definition 6 implies *non-adaptive* key tampering, as the subverted algorithm can hard-wire (the description of) polynomially many pre-set tampering functions.

[10] The algorithm $\widetilde{\mathsf{Vrfy}}$ is not explicitly part of the definitions in [BPR14]—in fact, a secure scheme implicitly excludes that any $\widetilde{\mathsf{Vrfy}}$ algorithm exists—and can be considered as part of the adversary itself.

Multi-user setting. For simplicity Definition 5 and 6 consider a single user. We provide an extension to the more general setting with $u \geq 2$ users, together with a complete picture of the relationships between different notions, in Appendix A.

3.2 Public/Secret Undetectability

We say that $\mathcal{A}$ meets the *verifiability condition* relative to $\mathcal{SS}$ if for all $\widetilde{\mathsf{A}} \in \mathcal{A}$ and for all $m \in \mathcal{M}$ the signatures produced using the subverted signing algorithm $\widetilde{\mathsf{A}}$ (almost) always verify under the corresponding verification key vk. Such a verifiability condition is a very basic form of (public) undetectability.

Definition 7 (Verifiability)**.** Let $\mathcal{A}$ be some class of SAs for a signature scheme $\mathcal{SS}$. We say that $\mathcal{A}$ satisfies ν_v-verifiability if for all $\widetilde{\mathsf{A}} \in \mathcal{A}$ and for all $m \in \mathcal{M}$

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \widetilde{\mathsf{A}}(sk, m))) = 1 : (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)\right] \geq 1 - \nu_v,$$

where the probability is taken over the randomness of all involved algorithms.

By undetectability, we mean the inability of ordinary users to tell whether signatures are computed using the subverted or the genuine signing algorithm. We will distinguish between the case where a subversion is *publicly* or *secretly* undetectable. Roughly speaking, public undetectability means that no user can detect subversions using the verification key vk only (i.e., without knowing the signing key sk); secret undetectability means that no user, even with knowledge of the signing key sk, can detect subversions.

A formal definition follows.

Definition 8 (Public/Secret Undetectability)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{A}$ is *secretly* (t, q, ε)-undetectable w.r.t. $\mathcal{SS}$ if for all PPT adversaries U running in time t, there exists an efficient challenger such that $\left|\mathbb{P}\left[\mathsf{U} \text{ wins}\right] - \frac{1}{2}\right| \leq \varepsilon(\kappa)$ in the following game:

1. The challenger runs $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, chooses an algorithm $\widetilde{\mathsf{A}} \in \mathcal{A}$ (possibly depending on vk), samples $b \leftarrow\!\!\$\; \{0, 1\}$ and gives (vk, sk) to U.
2. The user U can ask queries $m_i \in \mathcal{M}$, for all $i \in [q]$. The answer to each query depends on the secret bit b. In particular, if $b = 1$, the challenger returns $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$; if $b = 0$, the challenger returns $\widetilde{\sigma}_i \leftarrow \widetilde{\mathsf{A}}(sk, m_i)$.
3. Finally, U outputs a value $b' \in \{0, 1\}$; we say that U wins iff $b' = b$.

We say that $\mathcal{A}$ is *publicly* undetectable w.r.t. $\mathcal{SS}$ if in step 1. of the above game, U is only given the verification key. Moreover, whenever $\varepsilon(\kappa) = negl(\kappa)$ and $q = poly(\kappa)$ we simply say that $\mathcal{A}$ is secretly/publicly undetectable w.r.t. $\mathcal{SS}$.

Our definition of undetectability is similar to the corresponding definition considered by Bellare *et al.* [BPR14] for the case of symmetric encryption. One key difference is that, in the definition above, the challenger is allowed to choose the subversion algorithm possibly depending on the verification key of the user.[11] While one could in principle define even stronger forms of undetectability, e.g. by requiring that continuous and fully-adaptive SAs remain undetectable, we do not pursue this direction here. The reason for this is that the attacks we analyse in Section 4 are non-adaptive and only require to use a single subversion.

[11] Looking ahead, our new attack (cf. Section 4.2) will rely on this feature in the multi-user setting.

Secret vs. public undetectability. While secret undetectability clearly implies public undetectability, the converse is not true. In particular, in Appendix A.4 we show that there exists a signature scheme $\mathcal{SS}$ and a set of subversions $\mathcal{A}$ of it such that $\mathcal{A}$ is publicly undetectable w.r.t. $\mathcal{SS}$ but it is secretly detectable w.r.t. $\mathcal{SS}$.

Public undetectability vs. verifiability. One might think that verifiability is a special case of public undetectability. However, this is not true and in fact Definition 7 and 8 are incomparable. To see this, consider the class of SAs $\mathcal{A}_{\mathsf{msg}} = \{\widetilde{\mathsf{A}}_{\bar{m}}\}_{\bar{m}\in\mathcal{M}}$ that behaves identically to the original signing algorithm, except that upon input $\bar{m} \in \mathcal{M}$ it outputs an invalid signature. Clearly, $\mathcal{A}_{\mathsf{msg}}$ satisfies public undetectability as a user has only a negligible chance of hitting the value $\bar{m}$; yet $\mathcal{A}_{\mathsf{msg}}$ does not meet the verifiability condition as the latter is a property that holds for *all* messages.

On the other hand, consider the class of SAs $\mathcal{A}_{\mathsf{det}}$ that is identical to the original signing algorithm, except that it behaves deterministically on repeated inputs. Clearly, $\mathcal{A}_{\mathsf{det}}$ meets the verifiability condition relative to any (even randomized) signature scheme $\mathcal{SS}$; yet $\mathcal{A}_{\mathsf{det}}$ does not satisfy public undetectability for any randomized signature scheme $\mathcal{SS}$, as a user can simply query the same message twice in order to guess the value of the hidden bit b with overwhelming probability.

Multi-user setting. For simplicity Definition 8 considers a single user. We provide an extension to the more general setting with $u \geq 2$ users, together with a complete picture of the relationships between different notions, in Appendix A.

4 Mounting Subversion Attacks

In Section 4.1 we show that the biased-randomness attack of [BPR14] (adapted to the case of signatures), satisfies secret undetectability as per Definition 8 while allowing to recover the user's signing key with overwhelming probability. This attack allows to break all signature schemes using a sufficient amount of randomness; in Section 4.2 we present a new attack allowing to surreptitiously subvert even signature schemes using only little randomness (say 1 bit), provided that the targeted scheme satisfies an additional property.

4.1 Attacking Coin-Injective Schemes

We start by recalling an information-theoretic lemma from [BPR14]. Suppose $g : \mathcal{R} \to \mathcal{R}'$ where $\mathcal{R}, \mathcal{R}' \subseteq \{0,1\}^*$, $f : \{0,1\}^* \to \{0,1\}$, and $\rho = |\mathcal{R}|$. For $b \in \{0,1\}$ consider the following *biased* distribution:

$$\widetilde{\mathcal{R}}^{f,g}(b,\mathcal{R}) = \{r \in \mathcal{R} : f(g(r)) = b\}. \tag{1}$$

The lemma below roughly says that if a value r is chosen at random from the real distribution $\mathcal{R}$, the probability that r is also in the biased distribution $\widetilde{\mathcal{R}}$ is high if $|\mathcal{R}|$ is large enough.

Lemma 2 (Lemma 1 of [BPR14]). *Let f, g, b, $\mathcal{R}$, and $\widetilde{\mathcal{R}} = \widetilde{\mathcal{R}}^{f,g}(b,\mathcal{R})$ be as defined above. Then, if g is injective, for all $r \in \mathcal{R}$ we have*

$$\mathop{\mathbb{P}}_{\widetilde{r} \leftarrow\!\!\$\, \widetilde{\mathcal{R}}}[r = \widetilde{r}] = (1 - 2^{-\rho})/\rho.$$

The following attack is based on the biased-randomness attack from [BPR14]. Roughly, what it does is to embed a trapdoor—a key for a pseudorandom function—in the subverted signing algorithm and to "bias" the randomness in a way that it becomes possible to any party that knows the trapdoor to leak one bit of the signing key for each signed messaged under that signing key. Hence, if the adversary can obtain at least $|sk|$ signed messages then it can later extract the entire signing key in full.

SA class $\mathcal{A}^{F}_{\mathsf{bias}}$

Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a randomized signature scheme with randomness space $\mathcal{R}$, and $F : \{0,1\}^{\kappa} \times \{0,1\}^{*} \to \{0,1\}$ be a pseudorandom function. The class $\mathcal{A}^{F}_{\mathsf{bias}}$ consists of a set of algorithms $\{\widetilde{\mathsf{A}}_{s,\tau}\}_{s \in \{0,1\}^{\kappa}, \tau = 1}$, where each algorithm in the class behaves as follows:

$\underline{\widetilde{\mathsf{A}}_{s,\tau}(sk, m):}$

- For $|sk| = \ell$, let $i := \tau \mod \ell$.
- Define the function $g(\cdot) := \mathsf{Sign}(sk, m; \cdot)||\tau$ and sample a random element $\widetilde{r}$ from the distribution
$$\widetilde{\mathcal{R}}^{F(s,\cdot),g(\cdot)}(sk[i], \mathcal{R}) := \{r \in \mathcal{R} : F(s, g(r)) = sk[i]\}. \qquad (2)$$
- Return the signature $\sigma := \mathsf{Sign}(sk, m; \widetilde{r})$, and update the state $\tau \leftarrow \tau + 1$.

Extracting the signing key. Given as input a vector of signatures $\vec{\sigma} = (\sigma_1, ..., \sigma_\ell)$, for each signature $\sigma_i \in \vec{\sigma}$ try to extract the i-th bit of the signing key by defining $sk'[i] := F(s, \sigma_i||i)$. Return the signing key $sk' := (sk'[1], \dots, sk'[\ell])$.

Figure 1: Attacking coin-injective schemes

For the analysis, which relies on Lemma 2, we will need to assume the signing function is injective w.r.t. its random coins—a notion which we define below.

Definition 9 (Coin-injective)**.** We say that $\mathcal{SS}$ is *coin-injective* if for all $m \in \mathcal{M}$, and for all $(vk, sk) \leftarrow \mathsf{KGen}(1^{\kappa})$, we have that $\mathsf{Sign}(sk, m; \cdot)$ is injective.

Theorem 1. *Let $F : \{0,1\}^{\kappa} \times \{0,1\}^{*} \to \{0,1\}$ be a $(t_{\mathsf{prf}}, q_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}})$-secure PRF. For a randomized, coin-injective signature scheme $\mathcal{SS}$ with randomness space of size $\rho = |\mathcal{R}|$, consider the class of SAs $\mathcal{A}^{F}_{\mathsf{bias}}$ described in Fig. 1. Then,*

(i) $\mathcal{A}^{F}_{\mathsf{bias}}$ is secretly (t, q, ε)-undetectable for $t \approx t_{\mathsf{prf}}$, $q \approx q_{\mathsf{prf}}$ and $\varepsilon \leq q \cdot 2^{-(\rho+1)} + \varepsilon_{\mathsf{prf}}$.

(ii) Each $\widetilde{\mathsf{A}} \in \mathcal{A}^{F}_{\mathsf{bias}}$ recovers the signing key of the user with probability $(1 - 2^{-\rho})^{\ell}$ where ℓ is the size of the signing key.

Proof. (i) Let $\mathbf{G}$ be the game described in Definition 8, where the challenger picks $\widetilde{\mathsf{A}} \leftarrow\!\!\$\ \mathcal{A}^{F}_{\mathsf{bias}}$ (independently of the user's verification key). Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$, an identical copy of game $\mathbf{G}$ when $b = 1$. For the first part of the proof the objective is to show that $\mathbf{G}_0 \approx \mathbf{G}_1$.

Now consider game $\mathbf{G}'_0$ an identical copy of game $\mathbf{G}_0$ except that $\mathbf{G}'_0$ utilizes the distribution from Eq. (1) instead of the distribution from Eq. (2).

Claim 1. $|\mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}_0] - \mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}'_0]| \leq \varepsilon_{\mathsf{prf}}$.

Proof. We assume that there exists an adversary U that distinguishes between games $\mathbf{G}_0$ and $\mathbf{G}'_0$, and we build a distinguisher D (using U) that breaks the pseudorandomness of the PRF F. Distinguisher D is described below below.

Distinguisher D:

- Run $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, and return (vk, sk) to U.
- For each query $m_i \in \mathcal{M}$ asked by U, do:
  1. Pick a random $r \leftarrow\!\!\$\ \mathcal{R}$ and compute $x_i = \mathsf{Sign}(sk, m_i; r)||\tau$.
  2. Forward x_i to the target oracle, which answers with $y_i = f(x_i)$ if $b = 0$ or with $y_i = F(s, x_i)$ if $b = 1$ (for a hidden bit b).
  3. If $y_i = sk[i]$, then forward $\sigma_i = \mathsf{Sign}(sk, m_i; r)$ as an answer to the query of U, otherwise return to step (1).[12]
- Output whatever U outputs.

Notice that the probability that D aborts in step (3) of the reduction is the same probability that in game $\mathbf{G}_0$ and $\mathbf{G}_0'$ the subverted signing algorithm fails to sample from the set $\widetilde{\mathcal{R}}$. It follows that in case $b = 0$ distinguisher D perfectly emulates the distribution of $\mathbf{G}_0$, whereas in case $b = 1$ it perfectly emulates the distribution of $\mathbf{G}_0'$. The claim follows. □

Claim 2. $|\mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}_0'] - \mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}_1]| \leq q \cdot 2^{-(\rho+1)}$.

Proof. Abusing notation, let us write $\mathbf{G}_0'$ and $\mathbf{G}_1$ for the distribution of the random variables corresponding to U's view in games $\mathbf{G}_0'$ and $\mathbf{G}_1$ respectively. For an index $i \in [0, q]$ consider the hybrid game $\mathbf{H}_i$ that answers the first i signature queries as in game $\mathbf{G}_0'$ while all the subsequent queries are answered as in $\mathbf{G}_1$. We note that $\mathbf{H}_0 = \mathbf{G}_1$ and $\mathbf{H}_q = \mathbf{G}_0'$.

We claim that for all $i \in [q]$, we have $\mathbb{SD}(\mathbf{H}_{i-1}, \mathbf{H}_i) \leq 2^{-(\rho+1)}$. To see this, fix some $i \in [q]$ and denote with $\mathbf{R}$ (resp. $\widetilde{\mathbf{R}}$) the random variable defined by sampling an element from $\mathcal{R}$ (resp. $\widetilde{\mathcal{R}}$) uniformly at random. Clearly,

$$\begin{aligned}\mathbb{SD}(\mathbf{H}_{i-1}, \mathbf{H}_i) \leq \mathbb{SD}(\mathbf{R}, \widetilde{\mathbf{R}}) &= \frac{1}{2} \cdot \sum_{r \in \mathcal{R}} \left| \mathbb{P}[\mathbf{R} = r] - \mathbb{P}[\widetilde{\mathbf{R}} = r] \right| \\ &= \frac{1}{2} \cdot \sum_{r \in \mathcal{R}} \left| \frac{1}{\rho} - \frac{1 - 2^{-\rho}}{\rho} \right| \qquad (3) \\ &= \frac{1}{2} \cdot 2^{-\rho} = 2^{-(\rho+1)},\end{aligned}$$

where Eq. (3) follows by Lemma 2.

The claim now follows by the triangle inequality, as

$$\mathbb{SD}(\mathbf{G}_1, \mathbf{G}_0') \leq \sum_{i=1}^{q} \mathbb{SD}(\mathbf{H}_{i-1}, \mathbf{H}_i) \leq q \cdot 2^{-(\rho+1)}.$$

□

The two claims above finish the proof of statement (i).

(ii) For the second part of the proof we show that the attack of Fig. 1 fails to recover the secret key with probability at most $e_1 + e_2 + \ldots + e_\ell$, where $e_j := \mathbb{P}[sk'[j] \neq sk[j]]$. In the analysis, we replace w.l.o.g. the function F with a truly random function f; note that all applications of f are independent because we append the value τ to each query.

[12] In case $|\mathcal{R}|$ is exponential D simply aborts after polynomially many trials.

Now if g is injective and f is a random function that outputs one bit, then for each element $r \in \mathcal{R}$ we have $\mathbb{P}\left[f(g(r)) = sk[j]\right] = 1/2$. Extending to the entire set $\mathcal{R}$ of size ρ we have that

$$e_j := \mathbb{P}\left[\widetilde{\mathcal{R}}^{f,g}(sk[j], \mathcal{R}) = \emptyset\right] = 2^{-\rho},$$

is the error probability for each bit of the secret key. Therefore the probability of recovering the key is at least $(1 - 2^{-\rho})^{\ell}$.

□

Notice that for the attack to be undetectable with high probability, the underlying signature scheme needs to rely on a *minimal* amount of randomness, say $\rho \geq 2^7$.

4.2 Attacking Coin-Extractable Schemes

SA class $\mathcal{A}_{\mathsf{cext}}$

Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a coin-extractable, randomized signature scheme with randomness space $\mathcal{R}$ of size $\rho = 2^d$. For simplicity assume that $d|\ell$, where ℓ is the size of the signing key (a generalization is straightforward). The class $\mathcal{A}_{\mathsf{cext}}$ consists of a set of algorithms $\{\widetilde{\mathsf{A}}_{s,\tau}\}_{s \in \{0,1\}^{\ell}, \tau=0}$, where each algorithm in the class behaves as follows:

$\underline{\widetilde{\mathsf{A}}_{s,\tau}(sk, m)\text{:}}$

- If $\tau \geq \ell$ output a honestly generated signature $\sigma := \mathsf{Sign}(sk, m; r)$.
- Else,
    - for each value $j \in [d]$ compute the biased random bit $\widetilde{r}[j] := s[\tau + j] \oplus sk[\tau + j]$;
    - return the signature $\sigma := \mathsf{Sign}(sk, m; \widetilde{r})$, and update the state $\tau \leftarrow \tau + d$.

Extracting the signing key. Given as input a vector of signatures $\vec{\sigma} = (\sigma_1, \ldots, \sigma_{\ell/d})$, parse the trapdoor s as ℓ/d chunks of d bits $s = \{s_1, \ldots, s_{\ell/d}\}$. For each signature $\sigma_i \in \vec{\sigma}$ try to extract the d-bit chunk sk'_i of the signing key as follows.

- Extract the randomness from the i-th signature $\widetilde{r} \leftarrow \mathsf{CExt}(vk, m_i, \sigma_i)$.
- For each value $j \in [d]$ compute the secret key bit $sk'_i[j] := \widetilde{r}[j] \oplus s_i[j]$.

Return the signing key $sk' := (sk'_i, \ldots, sk'_{\ell/d})$.

Figure 2: Attacking coin-extractable schemes

The attack on Section 4.1 allows to break all sufficiently randomized schemes. This leaves the interesting possibility to show a positive result for schemes using less randomness, e.g. the Katz-Wang signature scheme [KW03] that uses a single bit of randomness. In this section we present a simple attack (cf. Fig. 2) ruling out the above possibility for all signature schemes that are *coin-extractable*, a notion which we define next.

Definition 10 (Coin-extractable)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that $\mathcal{SS}$ is ν_{ext}-coin-extractable if there exists a PPT algorithm CExt such that for all $m \in \mathcal{M}$

$$\mathbb{P}\left[\sigma = \mathsf{Sign}(sk, m; r) : (vk, sk) \leftarrow \mathsf{KGen}(1^{\kappa}); \sigma = \mathsf{Sign}(sk, m); r \leftarrow \mathsf{CExt}(vk, m, \sigma)\right] \geq 1 - \nu_{ext}.$$

We point that many existing signature schemes are coin-extractable:

- All *public-coin* signature schemes, where the random coins used to generate a signature are included as part of the signature. Concretely, the schemes in [BR97, GHR99, CS00, NPS01, CL02, Fis03, CL04, BB08, HW09a, HW09b, HK12], the Unstructured Rabin-Williams scheme [Ber08], and signature schemes derived by Sigma-protocols via the Fiat-Shamir transform [FS86] are all public-coin.
- The Katz-Wang scheme [KW03], where the signature on a message m is computed as $\sigma = f^{-1}(H(m||r))$ such that f is a trapdoor permutation, H is a hash function, and r is random bit. Given a pair (m, σ) the extractor simply sets $r = 1$ iff $f(\sigma) = H(m||1)$.

Theorem 2. *For a randomized, ν_{ext}-coin-extractable, signature scheme $\mathcal{SS}$ with randomness space $\mathcal{R}$ of size $\rho = 2^d$, consider the class of SAs $\mathcal{A}_{\mathsf{cext}}$ described in Fig. 2. Then,*

(i) $\mathcal{A}_{\mathsf{cext}}$ is secretly $(t, q, 0)$-undetectable for $t, q \in \mathbb{N}$.

(ii) Each $\widetilde{\mathsf{A}} \in \mathcal{A}_{\mathsf{cext}}$ recovers the signing key of the user with probability at least $(1 - \nu_{ext})^{\ell/d}$, where ℓ is the size of the key.

Proof. (i) Let $\mathbf{G}$ be the game described in Definition 8, where the challenger picks $\widetilde{\mathsf{A}} \leftarrow\!\!\$\ \mathcal{A}_{\mathsf{cext}}$ uniformly at random (and independently of the user's verification key). Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$, an identical copy of game $\mathbf{G}$ when $b = 1$. For the first part of the proof the objective is to show that $\mathbf{G}_0 \approx \mathbf{G}_1$.

Claim 3. $|\mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}_0] - \mathbb{P}[\mathsf{U} \text{ wins in } \mathbf{G}_1]| = 0$.

Proof. Abusing notation, let us write $\mathbf{G}_0$ and $\mathbf{G}_1$ for the distribution of the random variables corresponding to U's view in games $\mathbf{G}_0$ and $\mathbf{G}_1$ respectively. For an index $i \in [0, q]$ consider the hybrid game $\mathbf{H}_i$ that answers the first i signature queries as in game $\mathbf{G}_0$ while all the subsequent queries are answered as in $\mathbf{G}_1$. We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_q \equiv \mathbf{G}_0$.

We claim that for all $i \in [q]$, we have $\mathbf{H}_{i-1} \equiv \mathbf{H}_i$. To see this, fix some $i \in [q]$ and denote with $\mathbf{R}$ (resp. $\widetilde{\mathbf{R}}$) the random variable defined by sampling an element from $\mathcal{R}$ (resp. $\widetilde{\mathcal{R}}$) uniformly at random. It is easy to see that $\mathbf{R}$ and $\widetilde{\mathbf{R}}$ are identically distributed, as the biased distribution consists of a one-time pad encryption of (part of) the signing key with a uniform key. The claim follows. □

(ii) For the second part of the proof we note that the attack of Fig. 2 successfully recovers the biased randomness $\widetilde{r}$ of each $\sigma_i \in \{\sigma_1, \ldots, \sigma_{\ell/d}\}$ and computes the chunk sk_i of the signing key with probability at least $1 - \nu_{ext}$. This gives a total probability of recovering the entire signing key of at least $(1 - \nu_{ext})^{\ell/d}$.

□

5 Security of Unique Signatures

The theorem below shows that unique signature schemes (cf. Definition 3) are secure against the class of all SAs that meet the verifiability condition (cf. Definition 7).

Theorem 3. *Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with ν_c-correctness and ν_u-uniqueness, and denote by $\mathcal{A}^{\nu_v}_{\mathsf{ver}}$ the class of all algorithms that satisfy ν_v-verifiability relative to $\mathcal{SS}$. Then $\mathcal{SS}$ is (t, n, q, ε)-indistinguishable against continuous $\mathcal{A}^{\nu_v}_{\mathsf{ver}}$-SAs, for all $n, q \in \mathbb{N}$ and for $\varepsilon \leq qn \cdot (\nu_c + \nu_v + \nu_u)$.*

Proof. Let $\mathbf{G}$ be the game described in Definition 5. Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$, an identical copy of game $\mathbf{G}$ when $b = 1$. The objective here is to show that $\mathbf{G}_0 \approx \mathbf{G}_1$.

For an index $k \in [0, n]$, consider the hybrid game $\mathbf{H}_k$ that answers each query $(j, m_{i,j})$ such that $j \leq k$ as in game $\mathbf{G}_0$ (i.e., by running $\mathsf{Sign}(sk, m_{i,j})$), while all queries $(j, m_{i,j})$ such that $j > k$ are answered as in $\mathbf{G}_1$ (i.e., by running $\widetilde{\mathsf{A}}_j(sk, m_{i,j})$). We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_n \equiv \mathbf{G}_0$. Abusing notation, let us write $\mathbf{G}_k$ for the distribution of the random variable corresponding to B's view in games $\mathbf{G}_k$.

Fix a particular $k \in [0, n]$, and for an index $l \in [0, q]$ consider the hybrid game $\mathbf{H}_{k,l}$ that is identical to $\mathbf{H}_k$ except that queries $(k, m_{i,j})$ with $i \leq l$ are treated as in game $\mathbf{G}_0$, while queries $(k, m_{i,j})$ with $i > l$ are treated as in $\mathbf{G}_1$. Observe that $\mathbf{H}_{k,0} \equiv \mathbf{H}_{k-1}$, and $\mathbf{H}_{k,q} \equiv \mathbf{H}_k$.

Claim 4. *Fix some $k \in [0, n]$. For each $l \in [0, q]$, we have $\mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \leq \nu_c + \nu_v + \nu_u$.*

Proof. Notice that the only difference between $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$ is how the two games answer the query $(k, m_{l,k})$: Game $\mathbf{H}_{k,l-1}$ returns $\sigma_{l,k} \leftarrow \mathsf{Sign}(sk, m_{l,k})$, whereas game $\mathbf{H}_{k,l}$ returns $\widetilde{\sigma}_{l,k} \leftarrow \widetilde{\mathsf{A}}_k(sk, m_{l,k})$. Now let $E_{l,k}$ be the event that $\sigma_{l,k} \neq \widetilde{\sigma}_{l,k}$. We can write

$$\begin{aligned}\mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) &\leq \mathbb{SD}(\mathbf{H}_{k,l-1}; \mathbf{H}_{k,l} | \neg E_{l,k}) + \mathbb{P}[E_{l,k}] &(4)\\ &\leq \nu_c + \nu_u + \nu_v. &(5)\end{aligned}$$

Eq. (4) follows by Lemma 1 and Eq. (5) follows by the fact that $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$ are identically distributed conditioned on $E_{l,k}$ not happening, and moreover $\mathbb{P}[E_{l,k}] \leq \nu_c + \nu_u + \nu_v$. The latter can also be seen as follows. By the correctness condition of $\mathcal{SS}$ we have that $\sigma_{l,k}$ is valid for $m_{l,k}$ under vk except with probability at most ν_c. By the assumption that $\widetilde{\mathsf{A}}_k \in \mathcal{A}_{\mathsf{ver}}^{\nu_v}$ we have that $\widetilde{\sigma}_{l,k}$ is also valid for $m_{l,k}$ under vk except with probability at most ν_v. Finally, by the uniqueness property of $\mathcal{SS}$ we have that $\sigma_{l,k}$ and $\widetilde{\sigma}_{l,k}$ must be equal except with probability at most ν_u. It follows that $\mathbb{P}[E_{l,k}] \leq \nu_c + \nu_u + \nu_v$, as desired. □

The statement now follows by the above claim and by the triangle inequality, as

$$\mathbb{SD}(\mathbf{G}_0, \mathbf{G}_1) \leq \sum_{k=1}^{n} \mathbb{SD}(\mathbf{H}_{k-1}, \mathbf{H}_k) \leq \sum_{k=1}^{n} \sum_{l=1}^{q} \mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \leq qn \cdot (\nu_c + \nu_u + \nu_v).$$

□

6 Reverse Firewalls for Signatures

In Section 5 we have shown that unique signatures are secure against a restricted class of SAs, namely all SAs that meet the so-called verifiability condition. As discussed in Section 3, by removing the latter requirement (i.e., allowing for arbitrary classes of SAs in Definition 5 and 6) would require that a signature scheme $\mathcal{SS}$ remains unforgeable even against an adversary allowed *arbitrary tampering with the computation* performed by the signing algorithm. This is impossible without making further assumptions.

In this section we explore to what extent one can model signature schemes secure against arbitrary tampering with the computation, by making the extra assumption of an un-tamperable cryptographic reverse firewall (RF) [MS15]. Roughly, a RF for a signature scheme is a (possibly stateful) algorithm that takes as input a message/signature pair and outputs an updated signature; importantly the firewall has to do so using only public information (in particular, without knowing the signing key). A formal definition follows.

Definition 11 (RF for signatures)**.** Let $\mathcal{SS}$ be a signature scheme. A RF for $\mathcal{SS}$ is a pair of algorithms $\mathcal{FW} = (\mathsf{Setup}, \mathsf{Patch})$ specified as follows: (i) Setup takes as input the security parameter and a verification key $vk \in \mathcal{VK}$, and outputs some initial (public) state $\delta \in \{0,1\}^*$; (ii) Patch takes as input the current (public) state δ, and a message/signature pair (m, σ) and outputs a possibly modified signature or a special symbol $\perp$ and an updated (public) state δ'. We write this as $\sigma' \leftarrow \mathsf{Patch}_\delta(m, \sigma)$ (and omit to denote the updated state δ' as an explicit output).

We will typically assume that the current state δ_{cur} of the RF, can be computed efficiently given just the verification key vk, the initial state δ and the entire history of all inputs to the RF.

6.1 Properties

Below, we discuss the correctness and security requirements of cryptographic RF $\mathcal{FW}$ for a signature scheme $\mathcal{SS}$.

Maintaining functionality. The first basic property of a RF is that it should preserve the functionality of the underlying signature scheme, i.e. if a signature σ on a message m is computed using signing key sk, and the firewall is initialized with the corresponding verification key vk, the patched signatures σ' should (almost always) be a valid signatures for m under vk. More precisely, we say that $\mathcal{FW}$ is *functionality maintaining* for $\mathcal{SS}$, if for any polynomial $p(\kappa)$ and any vector of inputs $(m_1, \ldots, m_p) \in \mathcal{M}$, there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\mathbb{P}\left[\exists i \in [p] \text{ s.t. } \mathsf{Vrfy}(vk, (m_i, \sigma'_i)) = 0 : \begin{array}{c} (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa), \delta \leftarrow \mathsf{Setup}(vk, 1^\kappa) \\ \sigma_1 \leftarrow \mathsf{Sign}(sk, m_1), \ldots, \sigma_p \leftarrow \mathsf{Sign}(sk, m_p) \\ \sigma'_1 \leftarrow \mathsf{Patch}_\delta(m_1, \sigma_1), \ldots, \sigma'_p \leftarrow \mathsf{Patch}_\delta(m_p, \sigma_p) \end{array}\right] \leq \nu(\kappa),$$

where the probability is taken over the coin tosses of all involved algorithms. Recall that each invocation of algorithm Patch updates the (public) state δ of the RF.

Preserving Unforgeability. The second property of a RF is a security requirement. Note that a firewall can never "create" security (as it does not know the signing key). Below we define what it means for a RF to *preserve* unforgeability of a signature scheme against *arbitrary* tampering attacks.

Definition 12 (Unforgeability preserving RF)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with RF $\mathcal{FW} = (\mathsf{Setup}, \mathsf{Patch})$. We say that $\mathcal{FW}$ (t, n, q, ε)-preserves unforgeability for $\mathcal{SS}$ against continuous SAs if for all adversaries B running in time t we have that $\mathbb{P}[\mathsf{B} \text{ wins}] \leq \varepsilon$ in the following game:

1. The challenger runs $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, $\delta \leftarrow \mathsf{Setup}(vk, 1^\kappa)$, and gives (vk, δ) to B.
2. The adversary B is given oracle access to $\mathsf{Sign}(sk, \cdot)$. Upon input the i-th query m_i, this oracle returns $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$. Let $\mathcal{Q} = \{m_1, \ldots, m_q\}$ be the set of all signature queries.
3. The adversary B can adaptively choose an arbitrary algorithm $\widetilde{\mathsf{A}}_j$, and correspondingly obtain oracle access to $\mathsf{Patch}_\delta(\cdot, \widetilde{\mathsf{A}}_j(sk, \cdot))$:
   - Upon input the i-th query $\widetilde{m}_{i,j}$, for $i \in [q]$ and $j \in [n]$, the oracle returns $\widetilde{\sigma}_{i,j} \leftarrow \mathsf{Patch}_\delta(\widetilde{m}_{i,j}, \widetilde{\mathsf{A}}_j(sk, \widetilde{m}_{i,j}))$ and updates the public state δ;

- Whenever $\widetilde{\sigma}_{i,j} = \bot$ the oracle enters a special self-destructs mode, in which the answer to all future queries is by default set to $\bot$.

Let $\widetilde{\mathcal{Q}}_j = \{\widetilde{m}_{1,j}, \ldots, \widetilde{m}_{q,j}\}$ be the set of all queries for each $\widetilde{\mathsf{A}}_j$.

4. Finally, B outputs a pair (m^*, σ^*); we say that B wins iff $\mathsf{Vrfy}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \widetilde{\mathcal{Q}}$, where $\widetilde{\mathcal{Q}} := \bigcup_{j=1}^{n} \widetilde{\mathcal{Q}}_j$.

Whenever $t = poly(\kappa)$, $q = poly(\kappa)$ and $\varepsilon = negl(\kappa)$ we simply say that $\mathcal{FW}$ preserves unforgeability for $\mathcal{SS}$. Furthermore, in case A specifies all of its queries $\{\widetilde{\mathsf{A}}_j, \widetilde{m}_{i,j}\}_{j\in[n],i\in[q]}$ at the same time we say that $\mathcal{FW}$ *non-adaptively* preserves unforgeability.

We observe that Definition 12 is very similar to Definition 6, except for a few crucial differences. First, note that the above definition considers arbitrary classes of SAs instead of SAs within a given class $\mathcal{A}$; this is possible because the output of each invocation of the subverted signing algorithm is patched using the firewall (which is assumed to be un-tamperable).

Second, observe that the above definition relies on the so-called self-destruct capability: Whenever the firewall returns $\bot$, all further queries to any of the oracles results in $\bot$; as we show in Section 6.2 this is necessary as without such a capability there exists simple generic attacks that allow for complete security breaches. We stress, however, that the assumption of the self-destruct capability does not make the problem of designing an unforgeability preserving reverse firewall trivial. In fact, the biased-randomness attacks of Section 4 allow to break all randomized scheme *without* ever provoking a self-destruct. On the positive side, in Section 6.3, we show how to design an unforgeability preserving RF for any *re-randomizable* signature scheme.

Exfiltration resistance. More in general, one might require a stronger security property from a RF. Namely, we could ask that patched signatures are indistinguishable from real signatures to the eyes of an attacker. This property, which is called exfiltration resistance in [MS15], would be similar in spirit to our definition of indistinguishability w.r.t. continuous SAs (see Definition 5).

It is not hard to see that exfilatration resistance against arbitrary SAs is impossible to achieve in the case of signature schemes; this is because the attacker could simply set the subverted signing algorithm to always output the all-zero string, in which case the RF has no way to patch its input to a valid signature (and thus the adversary can easily distinguish subverted patched signatures from real signatures).[13]

6.2 Necessity of Self-Destruct

We show that no RF can preserve both functionality and unforgeability, without assuming the self-destruct capability. This is achieved by a generic (non-adaptive) attack that allows to extract the secret key in case the RF does not self-destruct. The attack itself is a generalization of a similar attack by Gennaro *et al.* [GLM+04] in the context of memory tampering.

Theorem 4. *Let $\mathcal{SS}$ be a ufcma signature scheme. No RF $\mathcal{FW}$ can at the same time be functionality maintaining and non-adaptively $(poly(\kappa), 1, poly(\kappa), negl(\kappa))$-preserve unforgeability for $\mathcal{SS}$, without assuming the self-destruct capability.*

Proof sketch. Consider the following adversary B playing the game of Definition 12 (omitting the self-destruct capability).

[13] We note, however, that our techniques from Section 5 can be extended to design a RF that is *weakly* exfiltration resistant, namely it is exfiltration resistant against restricted SAs that satisfy the verifiability condition.

Alice		♜ **Alice's Firewall** ♜
$m \in \mathcal{M}$		
$\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}(sk, m)$		
	$\xrightarrow{m,\widetilde{\sigma}}$	
		Read $\delta = (vk, \beta)$
		If $\beta = 1$ set $\widetilde{\sigma}' = \bot$
		Else if $\mathsf{Vrfy}(vk, (m, \widetilde{\sigma})) = 1$
		$\quad \widetilde{\sigma}' \leftarrow \mathsf{ReRand}(vk, m, \widetilde{\sigma})$
		Else set $\widetilde{\sigma}' = \bot$ and $\beta = 1$
		Forward $(m, \widetilde{\sigma}')$

Figure 3: A cryptographic reverse firewall preserving unforgeability of any re-randomizable signature scheme against arbitrary SAs.

- Upon input the verification key vk, and the initial state δ, initialize $\tau := 1$.
- Forward $\widetilde{\mathsf{A}}_\tau$ to the challenger, where algorithm $\widetilde{\mathsf{A}}_\tau$ is defined as follows: Upon input a message $\widetilde{m}_i$, set $j = \tau \bmod \ell$ (where $\ell := |sk|$) and
  - If $sk[j] = 1$, output $\widetilde{\sigma}_i \leftarrow \mathsf{Sign}(sk, \widetilde{m}_i)$.
  - Else, output $0^{|\sigma|}$.

  Update $\tau \leftarrow \tau + 1$.
- Let $(\bar{m}, \widetilde{\sigma}'_1), \ldots, (\bar{m}, \widetilde{\sigma}'_\ell)$ be the set of tampered signature queries (and answers to these queries) asked by B, where $\widetilde{\sigma}'_i \leftarrow \mathsf{Patch}_\delta(\bar{m}, \widetilde{\mathsf{A}}_\tau(sk, \bar{m}))$. Define $sk'[i] = \mathsf{Vrfy}(vk, (\bar{m}, \widetilde{\sigma}'_i))$ and return $sk' := (sk'[1], \ldots, sk'[\ell])$.

Notice that B specifies its queries non-adaptively, and moreover it only uses one subversion which is queried upon a fixed message $\bar{m} \in \mathcal{M}$. We will show that the extracted key sk' is equal to the original secret key sk with overwhelming probability, which clearly implies the statement. The proof is by induction; assume that the statement is true up to some index $i \geq 1$. We claim that $sk'[i+1] = sk[i+1]$ with all but negligible probability. To see this, define the event E_{i+1} that $sk[i+1] = 0$ and $\mathsf{Vrfy}(vk, (\bar{m}, \widetilde{\sigma}'_{i+1})) = 1$ *or* $sk[i+1] = 1$ and $\mathsf{Vrfy}(vk, (\bar{m}, \widetilde{\sigma}'_{i+1})) = 0$. By the assumption that the RF does not self-destruct and is functionality maintaining, we get that the latter sub-case happens only with negligible probability. On the other hand, if the former sub-case happens we get that the RF forged a signature on $\bar{m}$, which contradicts ufcma security of $\mathcal{SS}$. By a union bound, we get that $\mathbb{P}[E_{i+1}]$ is negligible as desired. □

6.3 Patching Re-Randomizable Signatures

We design a RF preserving unforgeability of so-called *re-randomizable* signature schemes (that include unique signatures as a special case).

Definition 13 (Re-randomizable signatures)**.** A signature scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is efficiently ν_r-re-randomizable, if there exists a PPT algorithm ReRand such that for all messages $m \in \mathcal{M}$ and for all $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and $\sigma \leftarrow \mathsf{Sign}(sk, m)$, we have that $\mathbb{SD}(\mathsf{ReRand}(vk, m, \sigma); \mathsf{Sign}(sk, m)) \leq \nu_r$.

Note that unique signatures are efficiently re-randomizable, for $\mathsf{ReRand}(vk, m, \sigma) = \sigma$ and $\nu_r = 0$; Waters' signature scheme [Wat05], and its variant by Hofheinz *et al.* [HJK12], are also efficiently re-randomizable.

Our firewall, which is formally described in Fig. 3, first checks if σ is a valid signature on message m under key vk (provided that a self-destruct was not provoked yet). If not, it self-destructs and returns $\perp$; otherwise it re-randomizes σ and outputs the result. The self-destruct capability is implemented using a one-time writable bit β (which is included in the public state).

Theorem 5. *Let $\mathcal{SS}$ be a $(t, (q+1)n, \varepsilon)$-ufcma signature scheme that is efficiently ν_r-re-randomizable and that satisfies ν_c-correctness. Then, the RF of Fig. 3 maintains functionality and (t', q, ε')-preserves unforgeability for $\mathcal{SS}$, where $t' \approx t$ and $\varepsilon' \leq qn \cdot (\nu_c + \nu_r + \varepsilon)$.*

Proof. The fact that the firewall maintains functionality follows directly by ν_c-correctness of $\mathcal{SS}$.

We now show the firewall preserves unforgeability. Let $\mathbf{G}$ be the game of Definition 12; we write $(i^*, j^*) \in [q] \times [n]$ for the pair of indexes in which the firewall self-destructs (if any). Consider the modified game $\mathbf{H}$ that is identical to $\mathbf{G}$ except that tampered signature queries are answered as described below:

- For all $j < j^*$, upon input $(j, \widetilde{m}_{i,j})$ return $\sigma_{i,j} \leftarrow \mathsf{Sign}(sk, \widetilde{m}_{i,j})$ for all $i \in [q]$.
- For $j = j^*$, upon input $(j, \widetilde{m}_{i,j})$ if $i < i^*$ return $\sigma_{i,j} \leftarrow \mathsf{Sign}(sk, \widetilde{m}_{i,j})$; else return $\perp$.
- For all $j > j^*$, upon input message $\widetilde{m}_{i,j}$ return $\perp$ for all $i \in [q]$.

Claim 5. $|\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{G}] - \mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}]| \leq qn \cdot (\nu_c + \nu_r)$.

Proof. For an index $k \in [0, n]$, consider the hybrid game $\mathbf{H}_k$ that answers each query $(j, \widetilde{m}_{i,j})$ such that $j \leq k$ as in game $\mathbf{G}$, while all queries $(j, \widetilde{m}_{i,j})$ such that $j > k$ are answered as in $\mathbf{H}$. We note that $\mathbf{H}_0 \equiv \mathbf{H}$ and $\mathbf{H}_n \equiv \mathbf{G}$. Abusing notation, let us write $\mathbf{H}_k$ for the distribution of the random variable corresponding to B's view in game $\mathbf{H}_k$.

We will show that $\mathbb{SD}(\mathbf{H}_{k-1}, \mathbf{H}_k) \leq q \cdot (\nu_c + \nu_r)$ for all k. Fix a particular $k \in [0, n]$, and for an index $l \in [0, q]$ consider the hybrid game $\mathbf{H}_{k,l}$ that is identical to $\mathbf{H}_k$ except that it answers queries $(k, \widetilde{m}_{i,j})$ with $i \leq l$ as in game $\mathbf{G}$, while all queries $(k, \widetilde{m}_{i,j})$ with $i > l$ are treated as in $\mathbf{H}$. Observe that $\mathbf{H}_{k,0} \equiv \mathbf{H}_{k-1}$, and $\mathbf{H}_{k,q} \equiv \mathbf{H}_k$.

We now argue that for each $l \in [q]$, one has that $\mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \leq \nu_c + \nu_r$. Observe that, since for $k > j^*$ both games always return $\perp$, we can assume without loss of generality that $k \leq j^*$. Note that the only difference between $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$ is how the two games answer the query $(k, \widetilde{m}_{l,k})$: $\mathbf{H}_{k,l-1}$ returns $\sigma_{l,k} \leftarrow \mathsf{Sign}(sk, \widetilde{m}_{l,k})$ whereas $\mathbf{H}_{k,l}$ returns $\widetilde{\sigma}'_{l,k} \leftarrow \mathsf{Patch}_\delta(\widetilde{m}_{l,k}, \widetilde{\sigma}_{l,k})$ where $\widetilde{\sigma}_{l,k} \leftarrow \widetilde{\mathsf{A}}_k(sk, \widetilde{m}_{l,k})$. Let $E_{l,k}$ be the event that $\mathsf{Vrfy}(vk, (\widetilde{m}_{l,k}, \sigma_{l,k})) = 0$. We have

$$\begin{aligned} \mathbb{SD}(\mathbf{H}_{k,l-1}; \mathbf{H}_{k,l}) &\leq \mathbb{SD}(\mathbf{H}_{k,l-1}; \mathbf{H}_{k,l} | \neg E_{l,k}) + \mathbb{P}[E_{l,k}] &(6)\\ &\leq \nu_r + \nu_c. &(7) \end{aligned}$$

Eq. (6) follows by Lemma 1 and Eq. (7) by the fact that $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$ are statistically close (up to distance ν_r) conditioned on $E_{l,k}$ not happening, and moreover $\mathbb{P}[E_{l,k}] \leq \nu_c$. The former is because signatures are re-randomizable, and thus (as long as the firewall did not self-destruct) the output of ReRand is statistically close (up to distance ν_r) to the output of the original signing algorithm; the latter follows by ν_c-correctness of the signature scheme.

The statement now follows by the above argument and by the triangle inequality, as

$$\begin{aligned}\mathbb{SD}(\mathbf{G},\mathbf{H}) &\le \sum_{k=1}^{n}\mathbb{SD}(\mathbf{H}_{k-1},\mathbf{H}_k)\\ &\le \sum_{k=1}^{n}\sum_{l=1}^{q}\mathbb{SD}(\mathbf{H}_{k,l-1},\mathbf{H}_{k,l})\\ &\le qn\cdot(\nu_c+\nu_r).\end{aligned}$$

□

Claim 6. $\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}] \le qn\cdot\varepsilon$.

Proof. Towards a contradiction, assume B wins in game $\mathbf{H}$ with probability larger than $qn\cdot\varepsilon$. Wlog. we assume that B always outputs its forgery after provoking a self-destruct.[14] We build an adversary B' (using B) that breaks ufcma of $\mathcal{SS}$. Adversary B' is described below.

Adversary B':

- Receive the verification key vk from the challenger, sample a random pair $(j^*, i^*) \leftarrow\!\!\$\ [n]\times[q]$, and return vk to B.
- Upon input the i-th signature query m_i, forward this value to the signing oracle receiving back a signature $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$. Return σ_i to B.
- Upon input a query of the form $(j, \widetilde{m}_{i,j})$ answer as follows:
  - In case $j < j^*$, forward $\widetilde{m}_{i,j}$ to the signing oracle, obtaining $\widetilde{\sigma}_{i,j} \leftarrow \mathsf{Sign}(sk, \widetilde{m}_i)$, and return $\widetilde{\sigma}_{i,j}$ to B.
  - In case $j = j^*$, if $i < i^*$ forward $\widetilde{m}_{i,j}$ to the signing oracle, obtaining $\widetilde{\sigma}_{i,j} \leftarrow \mathsf{Sign}(sk, \widetilde{m}_i)$, and return $\widetilde{\sigma}_{i,j}$ to B. Else, return $\perp$.
  - In case $j > j^*$ answer with $\perp$.
- Whenever B outputs (m^*, σ^*), output (m^*, σ^*).

For the analysis, note that B' runs in time similar to that of B and asks a total of at most $q + qn$ signing queries. Moreover, define the event E that B' guesses correctly the query (j^*, i^*) where B provokes a self-destruct. Clearly, in case E happens we have that B' perfectly simulates the distribution of game $\mathbf{H}$. Hence $\mathbb{P}[\mathsf{B}' \text{ wins}] \ge (qn\cdot\varepsilon)/(qn) = \varepsilon$, a contradiction. □

The proof follows by combining the above two claims. □

Acknowledgements

We are grateful to Abhishek Jain for his insightful comments, suggestions, and contributions during the early stages of this work.

[14] If not we can always modify B in such a way that it asks one additional query provoking a self-destruct; this clearly does not decrease B's advantage.

References

[ACF14] Michel Abdalla, Dario Catalano, and Dario Fiore. Verifiable random functions: Relations to identity-based key encapsulation and new constructions. *J. Cryptology*, 27(3):544–593, 2014.

[ACM$^+$14] Per Austrin, Kai-Min Chung, Mohammad Mahmoody, Rafael Pass, and Karn Seth. On the impossibility of cryptography with tamperable randomness. In *CRYPTO*, pages 462–479, 2014.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *STOC*, pages 774–783, 2014.

[AFPW11] Martin R. Albrecht, Pooya Farshim, Kenneth G. Paterson, and Gaven J. Watson. On cipher-dependent related-key attacks in the ideal-cipher model. In *FSE*, pages 128–145, 2011.

[AGM$^+$15] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In *TCC*, pages 375–397, 2015.

[AHI11] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *Innovations in Computer Science*, pages 45–60, 2011.

[BB08] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[BBG13] James Ball, Julian Borger, and Glenn Greenwald. Revealed: how US and UK spy agencies defeat internet privacy and security. *Guardian Weekly*, September 2013.

[BC10] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, pages 666–684, 2010.

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, pages 486–503, 2011.

[BDI$^+$99] Mike Burmester, Yvo Desmedt, Toshiya Itoh, Kouichi Sakurai, and Hiroki Shizuya. Divertible and subliminal-free zero-knowledge proofs for languages. *J. Cryptology*, 12(3):197–223, 1999.

[Ber08] Daniel J. Bernstein. Proving tight security for Rabin-Williams signatures. In *EUROCRYPT*, pages 70–87, 2008.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT*, pages 491–506, 2003.

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In *CRYPTO*, pages 1–19, 2014.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures—How to sign with RSA and Rabin. In *EUROCRYPT*, pages 399–416, 1996.

[BR97] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *CRYPTO*, pages 470–484, 1997.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *SCN*, pages 268–289, 2002.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, pages 56–72, 2004.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.

[Des88a] Yvo Desmedt. Abuses in cryptography and how to fight them. In *CRYPTO*, pages 375–389, 1988.

[Des88b] Yvo Desmedt. Subliminal-free authentication and signature (extended abstract). In *EUROCRYPT*, pages 23–33, 1988.

[DFMV13] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT*, pages 140–160, 2013.

[DFMV15] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. The chaining lemma and its application. In *ICITS*, pages 181–196, 2015.

[DFP15] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. A more cautious approach to security against mass surveillance. In *FSE*, 2015. To appear.

[DGG$^+$15] Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. A formal treatment of backdoored pseudorandom generators. In *EUROCRYPT*, pages 101–126, 2015.

[DK12] Dana Dachman-Soled and Yael Tauman Kalai. Securing circuits against constant-rate tampering. In *CRYPTO*, pages 533–551, 2012.

[DK14] Dana Dachman-Soled and Yael Tauman Kalai. Securing circuits and protocols against $1/poly(k)$ tampering rate. In *TCC*, pages 540–565, 2014.

[DLSZ15] Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. In *TCC*, pages 427–450, 2015.

[Dod03] Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In *PKC*, pages 1–17, 2003.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *Innovations in Computer Science*, pages 434–452, 2010.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *PKC*, pages 416–431, 2005.

[Fis03] Marc Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *PKC*, pages 116–129, 2003.

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, pages 465–488, 2014.

[FMNV15] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. A tamper and leakage resilient von Neumann architecture. In *PKC*, pages 579–603, 2015.

[FMVW14] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *EUROCRYPT*, pages 111–128, 2014.

[FPV11] Sebastian Faust, Krzysztof Pietrzak, and Daniele Venturi. Tamper-proof circuits: How to trade leakage for tamper-resilience. In *ICALP*, pages 391–402, 2011.

[Fry00] Niklas Frykholm. Countermeasures against buffer overflow attacks. Technical report, RSA Data Security, Inc., November 2000.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[GHR99] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT*, pages 123–139, 1999.

[GIP$^+$14] Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In *STOC*, pages 495–504, 2014.

[GL10] David Goldenberg and Moses Liskov. On related-secret pseudorandomness. In *TCC*, pages 255–272, 2010.

[GLM$^+$04] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In *TCC*, pages 182–200, 2011.

[Gre14] Glenn Greenwald. No place to hide: Edward Snowden, the NSA, and the U.S. surveillance state. *Metropolitan Books*, May 2014.

[HJK12] Dennis Hofheinz, Tibor Jager, and Edward Knapp. Waters signatures with optimal security reduction. In *PKC*, pages 66–83, 2012.

[HK12] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. *J. Cryptology*, 25(3):484–527, 2012.

[HW09a] Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In *EUROCRYPT*, pages 333–350, 2009.

[HW09b] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In *CRYPTO*, pages 654–670, 2009.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: keeping secrets in tamperable circuits. In *EUROCRYPT*, pages 308–327, 2006.

[Jag15] Tibor Jager. Verifiable random functions from weaker assumptions. In *TCC*, pages 121–143, 2015.

[JW15] Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In *TCC*, pages 451–480, 2015.

[KKS11] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.

[KT13] Aggelos Kiayias and Yiannis Tselekounis. Tamper resilient circuits: The adversary at the gates. In *ASIACRYPT*, pages 161–180, 2013.

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In *ACM CCS*, pages 155–164, 2003.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *CRYPTO*, pages 517–532, 2012.

[Luc04] Stefan Lucks. Ciphers secure against related-key attacks. In *FSE*, pages 359–370, 2004.

[Lys02] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In *CRYPTO*, pages 597–612, 2002.

[MRV99] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS*, pages 120–130, 1999.

[MS15] Ilya Mironov and Noah Stephens-Davidowitz. Cryptographic reverse firewalls. In *EUROCRYPT*, pages 657–686, 2015.

[NIS07] NIST (National Institute of Standards and Technology). Special Publication 800-90: Recommendation for random number generation using deterministic random bit generators, March 2007.

[NPS01] David Naccache, David Pointcheval, and Jacques Stern. Twin signatures: an alternative to the hash-and-sign paradigm. In *ACM CCS*, pages 20–27, 2001.

[One96] Aleph One. Smashing the stack for fun and profit. *Phrack Magazine*, 7(49):File 14, 1996.

[PB04] Jonathan D. Pincus and Brandon Baker. Beyond stack smashing: Recent advances in exploiting buffer overruns. *IEEE Security & Privacy*, 2(4):20–27, 2004.

[PLS13] Nicole Perlroth, Jeff Larson, and Scott Shane. N.S.A. able to foil basic safeguards of privacy on web. *The New York Times*, September 2013.

[Rey11] Leo Reyzin. Lecture notes: Extractors and the leftover hash lemma, March 2011.

[SFKR15] Bruce Schneier, Matthew Fredrikson, Tadayoshi Kohno, and Thomas Ristenpart. Surreptitiously weakening cryptographic systems. *IACR Cryptology ePrint Archive*, 2015:97, 2015.

[Sim83] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983.

[Sim84] Gustavus J. Simmons. The subliminal channel and digital signature. In *EUROCRYPT*, pages 364–378, 1984.

[VV83] Umesh V. Vazirani and Vijay V. Vazirani. Trapdoor pseudo-random number generators, with applications to protocol design. In *FOCS*, pages 23–30, 1983.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[Wee12] Hoeteck Wee. Public key encryption against related key attacks. In *PKC*, pages 262–279, 2012.

[YY96] Adam L. Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust Capstone? In *CRYPTO*, pages 89–103, 1996.

[YY97] Adam L. Young and Moti Yung. Kleptography: Using cryptography against cryptography. In *EUROCRYPT*, pages 62–74, 1997.

[YY04] Adam L. Young and Moti Yung. *Malicious Cryptography: Exposing Cryptovirology.* John Wiley & Sons, Inc., first edition, 2004.

A The Multi-User Setting

In this section we consider the multi-user setting for all definitions that appear in this paper. We also provide a complete picture of relationships between all definitions, as shown in Fig. 4 and Fig. 5.

A.1 Impersonation (Multi-User Version)

Analogous to the single-user setting, we consider two security definitions corresponding to different adversarial goals.

In the indistinguishability definition for the multi-user setting adversary B now receives $u \geq 1$ key pairs from the challenger and can continuously subvert each user independently. A formal definition follows.

Definition 14 (Indistinguishability against SAs—Multi-User)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{SS}$ is u-users indistinguishable w.r.t *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B there exists a negligible function $\nu : \mathbb{N} \to [0, 1]$, such that $\left|\mathbb{P}[\mathsf{B}\text{ wins}] - \frac{1}{2}\right| \leq \nu(\kappa)$ in the following game:

1. The challenger samples $b \leftarrow\!\!\$\ \{0, 1\}$, generates $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$ for $i \in [u]$ and gives $vk_1, \ldots, vk_u$ to B.
2. The adversary B can specify polynomially many queries (adaptively and in an arbitrary order) of the form $(i, \widetilde{\mathsf{A}})$ for $i \in [u]$.
   (a) For each such query, B is given access to an oracle that can be queried polynomially many times on inputs $m \in \mathcal{M}$.
   (b) The answer to each query m depends on the value of the secret bit b. In particular, if $b = 1$, the output is $\sigma \leftarrow \mathsf{Sign}(sk_i, m)$; if $b = 0$, the output is $\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}(sk_i, m)$.
3. Finally, B outputs a value $b' \in \{0, 1\}$; we say that B wins iff $b' = b$.

In the impersonation definition for the multi-user setting adversary B now receives $u \geq 1$ key pairs from the challenger and can continuously subvert each user independently; adversary B is successful if it can impersonate *any* of the users. A formal definition follows.

Definition 15 (Impersonation against SAs—Multi-User)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{SS}$ is u-users hard to impersonate w.r.t. *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B there exists a negligible function $\nu : \mathbb{N} \to [0,1]$, such that $\mathbb{P}[\mathsf{B} \text{ wins}] \leq \nu(\kappa)$ in the following game:

1. The challenger generates $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$ for $i \in [u]$ and gives $vk_1, \ldots, vk_u$ to B.
2. Adversary B is given oracle access to $\mathsf{Sign}(sk_i, \cdot)$. Upon input query $m \in \mathcal{M}$, this oracle returns $\sigma \leftarrow \mathsf{Sign}(sk_i, m)$; let $\mathcal{Q}$ be the set of all messages queried to this oracle.
3. The adversary B can specify polynomially many queries (adaptively and in an arbitrary order) of the form $(i, \widetilde{\mathsf{A}})$ for $i \in [u]$.
   (a) For each such query, B is given access to an oracle that can be queried polynomially many times upon inputs $m \in \mathcal{M}$.
   (b) The answer to each query m is $\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}(sk_i, m)$; let $\widetilde{\mathcal{Q}}$ be the set containing all queried messages to all oracles.
4. Finally, B outputs a tuple (m^*, σ^*, i^*); we say that B wins iff $\mathsf{Vrfy}(vk_{i^*}, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \widetilde{\mathcal{Q}}$.

A.2 Public/Secret Undetectability (Multi-User Version)

In the undetectability definition for the multi-user setting adversary U now receives $u \geq 1$ key pairs from the challenger (only the verification keys for public undetectability) and is allowed to make polynomially many signature queries for all users (key pairs). The answer to these queries are either computed using the real signature algorithm or a subverted algorithm previously chosen by the challenger possibly depending on the verification keys of the users. A formal definition follows.

Definition 16 (Public/Secret Undetectability—Multi-User)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We say that $\mathcal{A}$ is u-users *secretly* undetectable w.r.t. $\mathcal{SS}$ if for all PPT adversaries U, there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ and an efficient challenger such that $\left|\mathbb{P}[\mathsf{U} \text{ wins}] - \frac{1}{2}\right| \leq \nu(\kappa)$ in the following game:

1. The challenger samples $b \leftarrow\!\!\$ \{0,1\}$, generates $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$ for $i \in [u]$, chooses $\widetilde{\mathsf{A}} \leftarrow \mathcal{A}$ (possibly depending on $vk_1, \ldots, vk_u$), and gives $(vk_1, sk_1, \ldots, vk_u, sk_u)$ to B.
2. The adversary U can ask polynomially many queries of the form (i, m), where $i \in [u]$ and $m \in \mathcal{M}$. The answer to each query depends on the secret bit b. In particular, if $b = 1$, the challenger returns $\sigma \leftarrow \mathsf{Sign}(sk_i, m)$; if $b = 0$, the challenger returns $\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}(sk_i, m)$.
3. Finally, U outputs a value $b' \in \{0,1\}$; we say that U wins iff $b' = b$.

We say that $\mathcal{A}$ is u-users *publicly* undetectable w.r.t. $\mathcal{SS}$ if in step 1. of the above game, U is only given the verification keys of the u-users.

A.3 Impersonation Relations

Theorem 6 formalizes the relations depicted in Fig. 4.

Theorem 6. *The following relations hold: (i)* 1-IND $\Rightarrow$ u-IND*, (ii)* 1-IMP $\not\Rightarrow$ 1-IND*, (iii)* u-IND $\Rightarrow$ u-IMP *for any ufcma secure signature scheme, and (iv)* 1-IMP $\Rightarrow$ u-IMP*.*

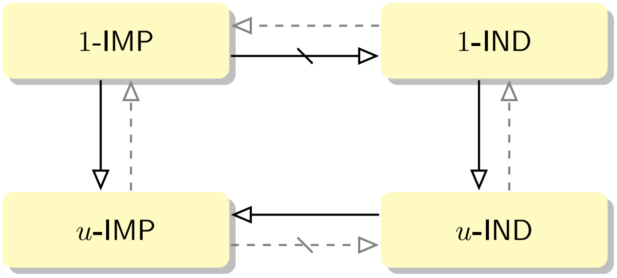

Figure 4: Diagram of the relationships between the subversion notions considered in this paper. $X \rightarrow Y$ means that X implies Y; $X \nrightarrow Y$ indicates a separation between X and Y. The lighter arrows indicates trivial implications (or implications that follow from Theorem 6). Definition 14 is represented by u-IND and Definition 15 is represented by u-IMP.

Proof. (i) 1-$\mathsf{IND} \Rightarrow u$-$\mathsf{IND}$. Towards contradiction, consider an adversary B that wins the game described in Definition 14. We build an adversary B' that (using B) wins the game described in Definition 5.

Let $\mathbf{G}$ be the game described in Definition 14. Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$ an identical copy of game $\mathbf{G}$ when $b = 1$.

For an index $k \in [0, u]$, consider the hybrid game $\mathbf{H}_k$ where each oracle corresponding to query $(i, \widetilde{\mathsf{A}})$ such that $i \leq k$ behaves as $\widetilde{\mathsf{A}}(sk_i, \cdot)$ (i.e., as in game $\mathbf{G}_0$), while all oracles corresponding to queries $(i, \widetilde{\mathsf{A}})$ such that $i > k$ behave as $\mathsf{Sign}(sk_i, \cdot)$ (i.e., as in game $\mathbf{G}_1$). We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_u \equiv \mathbf{G}_0$. By assumption, we know that B can distinguish between the extreme hybrid games $\mathbf{H}_0$ and $\mathbf{H}_u$. So there must exist a pair of hybrids $\mathbf{H}_i$, $\mathbf{H}_{i-1}$ that B can distinguish with a non-negligible advantage. We can construct B' as follows.

Adversary B':

1. Receive vk^* from the challenger and sample $(vk_j, sk_j) \leftarrow \mathsf{KGen}(1^\kappa)$ for all $j \in [u] \setminus \{i\}$. Define $vk_i = vk^*$ and forward $(vk_1, \ldots, vk_u)$ to adversary B.
2. Upon input a query $(j, \widetilde{\mathsf{A}})$ from B, behave as follows.
   - If $j \leq i - 1$ answer all queries $m \in \mathcal{M}$ as $\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}(sk_j, m)$;
   - if $j = i$ forward all queries $m \in \mathcal{M}$ to the challenger;
   - if $j \geq i$ answer all queries $m \in \mathcal{M}$ as $\sigma \leftarrow \mathsf{Sign}(sk_j, m)$.
3. Output whatever B outputs.

We observe that adversary B' simulates perfectly the distribution of the games $\mathbf{H}_{i-1}$ (when $b = 0$) and $\mathbf{H}_i$ (when $b = 1$). Since adversary B can distinguish this pair of hybrids with non-negligible probability it follows that adversary B' wins in the single-user game with the same probability.

(ii) 1-$\mathsf{IMP} \nRightarrow 1$-$\mathsf{IND}$. We sketch a separation for the definitions. Consider $\mathcal{SS}$ to be a ufcma secure signature scheme with signature size ℓ bits, and let $\mathcal{A}$ be the class of SAs for $\mathcal{SS}$ such that for all $\widetilde{\mathsf{A}} \in \mathcal{A}$ the output of $\widetilde{\mathsf{A}}$ is 0^ℓ. By $\mathcal{SS}$ being ufcma secure, adversary B has only a negligible probability of winning at the game described in Definition 15. However adversary B can clearly win the game described in Definition 14, because it is easy for B to distinguish between real signatures and subverted signatures.

(iii) u-$\mathsf{IND} \Rightarrow u$-$\mathsf{IMP}$. Consider $\mathcal{SS}$ to be a ufcma secure signature scheme and let $\mathcal{A}$ be a class of SAs for $\mathcal{SS}$. The objective here is to show that if $\mathcal{A}$ is u-users indistinguishable w.r.t continuous SAs (Definition 14) then $\mathcal{A}$ is also u-users hard to impersonate w.r.t continuous SA (Definition 15). We sketch a proof by considering a modified game for Definition 15, where all oracles behave like the real signing oracle (one oracle for each signing key). Since the class $\mathcal{A}$ is u-users indistinguishable we get that the advantage of any adversary in winning the game of Definition 15 is negligibly close to the advantage of winning in the modified game. However, by ufcma security of $\mathcal{SS}$ no PPT adversary can win the modified game with non-negligible advantage, and so $\mathcal{SS}$ satisfies u-IMP.

(iv) 1-IMP $\Rightarrow$ u-IMP. Towards contradiction, consider an adversary B that wins the game described in Definition 15. We build an adversary B' that (using B) wins the game described in Definition 6.

Adversary B':

1. Receive vk^* from the challenger, sample $i^* \leftarrow\$ \{1,\ldots,u\}$ and $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$ for all $i \in [u] \setminus \{i^*\}$. Set $vk_{i^*} := vk^*$ and forward $(vk_1,\ldots,vk_u)$ to adversary B.
2. Upon each query (i, m), for $i \in [u]$ and $m \in \mathcal{M}$: If $i \neq i^*$ reply with $\sigma = \mathsf{Sign}(sk_i, m)$, else forward the query to the challenger.
3. Upon each query $(i, \widetilde{\mathsf{A}})$, with $i \in [u]$, behave as follows.
   - For each $m \in \mathcal{M}$ chosen by the adversary B, if $i \neq i^*$ answer with $\widetilde{\sigma} = \widetilde{\mathsf{A}}(sk_j, m)$, else forward the query to the challenger.
4. Eventually B outputs a forgery (i', m', σ'); adversary B' outputs (m', σ') as its own forgery.

Adversary B' is successful if adversary B outputs a forgery for user i^*. Define E, to be the event that B' guesses correctly the index $i' = i^*$; note that $\mathbb{P}[E] = 1/u$. Therefore adversary B' has a non-negligible probability of winning at the game described in Definition 6. □

A.4 Undetectability Relations

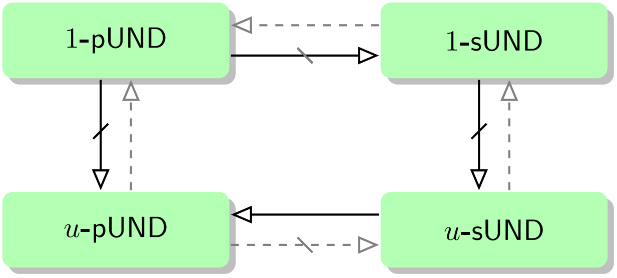

Figure 5: Diagram of the relationships between the undetectability notions considered in this paper. $X \rightarrow Y$ means that X implies Y; $X \nrightarrow Y$ indicates a separation between X and Y. The lighter arrows indicates trivial implications (or implications that follow from Theorem 7). Public undetectability (Definition 16) is represented by u-pUND and the secret undetectability (Definition 15) is represented by u-sUND.

The following theorem (7) formalizes the relations depicted in Fig. 5.

Theorem 7. *The following relations hold. (i)* u-sUND $\Rightarrow$ u-pUND*, (ii)* 1-pUND $\not\Rightarrow$ 1-sUND*, (iii)* 1-pUND $\not\Rightarrow$ u-pUND*, and (iv)* 1-sUND $\not\Rightarrow$ u-sUND*.*

Proof. (i) u-$\mathsf{sUND} \Rightarrow u$-$\mathsf{pUND}$. Fix any class $\mathcal{A}$ of SAs for $\mathcal{SS}$, and let C^* be the (efficient) challenger that exists by the assumption that $\mathcal{A}$ is secretly undetectable. We claim that $\mathcal{A}$ is also publicly undetectable for the same choice of the challenger C^*. Towards contradiction, consider an adversary U that wins the public undetectability game described in Definition 16 against C^*. We build an adversary U' (using U) that wins the secret undetectability game described in Definition 16 against C^*.

Adversary U':

1. Receive $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$, for $i \in [u]$, from the challenger C^* and forward it to adversary U.
2. Adversary U asks polynomially many queries of the type (i, m) which are forwarded to the challenger C^*.
3. Output whatever U outputs.

We note that the simulation performed by adversary U' is perfect, therefore U' wins the secret undetectability game with the same probability that adversary U wins the public undetectability game.

(ii) 1-$\mathsf{pUND} \not\Rightarrow$ 1-sUND. We sketch a separation between the definitions. Let $\mathcal{SS}$ be a randomized signature scheme, and let $\mathcal{SS}'$ be its derandomized implementation s.t. $sk' := (sk, s)$, $vk' := vk$, and $\sigma' := \mathsf{Sign}(sk, m; r')$ with $r' := F_s'(m)$ (for a PRF F). We note that the only difference between $\mathcal{SS}$ and $\mathcal{SS}'$ is how the randomness r is computed for the signing algorithm. Let $\mathcal{A}_{\mathsf{bias}}^F$ be the class of SAs described in Fig. 1. By security of the PRF, and by Theorem 1, an adversary U has only a negligible probability of winning at the public undetectability game described in Definition 8. However, an adversary U playing the secret undetectability game knows $sk' = (sk, s)$ and thus U can easily distinguish subverted and real signatures by simply re-computing r' and re-signing the input message; if both signatures match then with a high probability the target oracle is the real signining oracle. Notice that the last statement holds no matter how the subversion algorithm $\widetilde{\mathsf{A}} \in \mathcal{A}_{\mathsf{bias}}^F$ is selected by the challenger in the secret undetectability game.

(iii) 1-$\mathsf{pUND} \not\Rightarrow u$-$\mathsf{pUND}$. We sketch a separation for the definitions. Consider $\mathcal{SS}$ to be a contrived signature scheme such that the signature of a message $m \in \mathcal{M}$ is $\sigma = \mathsf{Sign}(sk, m; r)||r$, where $r \leftarrow\$ \{0,1\}^\kappa$. Let $\mathcal{A} = \{\widetilde{\mathsf{A}}_{\tau,\bar{r}}\}_{\tau=0,\bar{r}\in\{0,1\}^\kappa}$ to be class of SAs for $\mathcal{SS}$ described next.

$\widetilde{\mathsf{A}}_{\tau,\bar{r}}(sk, m)$:

1. If $\tau = 0$ then let $r := \bar{r}$, else let $r \leftarrow\$ \{0,1\}^\kappa$.
2. Output $\sigma \leftarrow \mathsf{Sign}(sk, m; r)||r$ and update $\tau = \tau + 1$.

Clearly, the class $\mathcal{A}$ is publicly undetectable for a single user because the output of the subverted signature algorithm is indistiguishable from that of the real signing algorithm, even for the first query (when $\tau = 0$). However, the class $\mathcal{A}$ is clearly 2-users publicly *detectable* since (no matter the strategy of the challenger) it suffices to ask one query for each user and compare the last κ bits of the signatures to distinguish between real and subverted signatures.

(iv) 1-$\mathsf{sUND} \not\Rightarrow u$-$\mathsf{sUND}$. We show a separation between the definitions. Consider $\mathcal{SS}$ to be a randomized, coin-extractable signature scheme, with randomness size of ℓ-bits, where $\ell = |sk|$, and $\mathcal{A}_{\mathsf{cext}}$ to be the class of SAs for $\mathcal{SS}$ described in Fig. 2. We already showed in Theorem 2 that (for the challenger C^* that chooses $\widetilde{\mathsf{A}}$ at random from $\mathcal{A}_{\mathsf{cext}}$) any PPT adversary U playing the secret undetectability game described in Definition 8 has a negligible advantage. Now consider the same adversary U playing the 2-users secret undetectability game described in Definition 16; adversary U now has 2 key pairs that can be used to detect the attack in the following way.

Adversary U:

1. Receive $(vk_i, sk_i) \leftarrow \mathsf{KGen}(1^\kappa)$, for $i = 1, 2$.
2. Fix a message $\bar{m} \in \mathcal{M}$ and query $(1, \bar{m})$ and $(2, \bar{m})$ to the challenger, that replies with σ_1 and σ_2.
3. Use CExt to extract the randomness from σ_1 and σ_2 to get $r_1 \leftarrow \mathsf{CExt}(vk_1, \bar{m}, \sigma_1)$ and $r_2 \leftarrow \mathsf{CExt}(vk_2, \bar{m}, \sigma_2)$.
4. Compute $sk_1 \oplus sk_2$ and return 0 iff the result equals $r_1 \oplus r_2$.

We note that the above detection strategy works regardless what strategy the challenger uses to select an algorithm from the class $\mathcal{A}_{\mathsf{cext}}$. We conclude that adversary U has an overwhelming probability of distinguishing between real and subverted signatures.

□

B Mounting Multi-User SAs

In this section we extend the attacks of Fig. 1 and Fig. 2 to the multi-user setting.

B.1 Attacking Coin-Injective Schemes (Multi-User Version)

The attack described in Fig. 1 can be extended to the multi-user setting with minor modifications. We create an SA class $\mathcal{A}_{\mathsf{bias}}^{F,u}$ from the class $\mathcal{A}_{\mathsf{bias}}^{F}$ of Fig. 1 by just appending the index j, that represents each user, to the function $g(\cdot) = \mathsf{Sign}(sk_j, m)||\tau||j$, so that each application of the random function $f(g(\cdot))$ remains independent.

The two lemmas below (Lemma 3 and Lemma 4) are needed for the proof of undetectability in the multi-user setting. The two lemmas combined roughly state that the statistical distance of a joint distribution of u random variables is *at most* u times the statistical distance of each pair of the random variables.

Lemma 3 ([Rey11])**.** *Let $\mathbf{X}$ and $\mathbf{Y}$ be two random variables over some finite domain, and let G be a randomized function. Then $\mathbb{SD}\left(G(\mathbf{X}), G(\mathbf{Y})\right) \leq \mathbb{SD}\left(\mathbf{X}, \mathbf{Y}\right)$.*

Lemma 4. *Let $\mathbf{X}$ and $\mathbf{Y}$ be two random variables over some finite domain, and let $(\mathbf{X}_1, \ldots, \mathbf{X}_u)$ and $(\mathbf{Y}_1, \ldots, \mathbf{Y}_u)$ be u independent copies of the random variables $\mathbf{X}$ and $\mathbf{Y}$. Then*

$$\mathbb{SD}\left((\mathbf{X}_1, \ldots, \mathbf{X}_u), (\mathbf{Y}_1, \ldots, \mathbf{Y}_u)\right) \leq u \cdot \mathbb{SD}\left(\mathbf{X}, \mathbf{Y}\right).$$

Proof. We prove this lemma by induction. First we consider the basis case where $i = 1$, which trivially holds as $\mathbb{SD}\left(\mathbf{X}_1, \mathbf{Y}_1\right) \leq \mathbb{SD}\left(\mathbf{X}, \mathbf{Y}\right)$.

For the induction step we define the random functions $G_1(\cdot) = (\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \cdot)$ and $G_2(\cdot) = (\cdot, \mathbf{Y}_i)$. We assume that the statement holds up to $i - 1$ random variables and then we proceed to show that it also holds for i random variables.

$$
\begin{aligned}
&\mathbb{SD}\left((\mathbf{X}_1,\ldots,\mathbf{X}_i),(\mathbf{Y}_1,\ldots,\mathbf{Y}_i)\right)\\
&\leq \mathbb{SD}\left((\mathbf{X}_1,\ldots,\mathbf{X}_i),(\mathbf{X}_1,\ldots,\mathbf{X}_{i-1},\mathbf{Y}_i)\right)+\mathbb{SD}\left((\mathbf{X}_1,\ldots,\mathbf{X}_{i-1},\mathbf{Y}_i),(\mathbf{Y}_1,\ldots,\mathbf{Y}_i)\right)\\
&= \mathbb{SD}\left(G_1(\mathbf{X}_i),G_1(\mathbf{Y}_i)\right)+\mathbb{SD}\left(G_2(\mathbf{X}_1,\ldots,\mathbf{X}_{i-1}),G_2(\mathbf{Y}_1,\ldots,\mathbf{Y}_{i-1})\right)\\
&\leq \mathbb{SD}\left(\mathbf{X},\mathbf{Y}\right)+\mathbb{SD}\left((\mathbf{X}_1,\ldots,\mathbf{X}_{i-1}),(\mathbf{Y},\ldots,\mathbf{Y}_{i-1})\right)\\
&\leq i\cdot\mathbb{SD}\left(\mathbf{X},\mathbf{Y}\right),
\end{aligned}
$$

where the first inequality follows by the triangle inequality, the second inequality follows by Lemma 3, and the third inequality follows by the induction hypothesis. □

The theorem below quantifies the effectiveness of the attack of Fig. 1 in the multi-user setting.

Theorem 8. *Let $F : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}$ be a secure PRF. For a randomized, coin-injective signature scheme $\mathcal{SS}$ with randomness space of size $\rho = |\mathcal{R}|$, consider the class of SAs $\mathcal{A}^{F,u}_{\mathsf{bias}}$ described above. Then,*

(i) $\mathcal{A}^{F,u}_{\mathsf{bias}}$ is u-users secretly undetectable.

(ii) Each $\widetilde{\mathsf{A}} \in \mathcal{A}^{F,u}_{\mathsf{bias}}$ recovers the signing key of any of the users with probability $(1-2^{-\rho})^\ell$, where ℓ is the size of the signing key.

Proof. (i) Let $\mathbf{G}$ be the game described in Definition 16. Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$, an identical copy of game $\mathbf{G}$ when $b = 1$. For the first part of the proof the objective is to show that $\mathbf{G}_0 \approx \mathbf{G}_1$.

Now consider game $\mathbf{G}'_0$ an identical copy of game $\mathbf{G}_0$ except that $\mathbf{G}'_0$ utilizes the distribution from the random function f (analogous to Eq. (1) in the single user attack) instead of the distribution from the PRF F (analogous to Eq. (2) in the single user attack).

Claim 7. $|\mathbb{P}[\mathsf{U}\text{ wins in }\mathbf{G}_0] - \mathbb{P}[\mathsf{U}\text{ wins in }\mathbf{G}'_0]| \leq negl(\kappa)$.

The above claim follows by a standard reduction argument to the hardness of the PRF F to distinguishing games $\mathbf{G}_0$ and $\mathbf{G}'_0$. The proof is similar to the one in Theorem 1 and is therefore ommited.

Claim 8. $|\mathbb{P}[\mathsf{U}\text{ wins in }\mathbf{G}'_0] - \mathbb{P}[\mathsf{U}\text{ wins in }\mathbf{G}_1]| \leq negl(\kappa)$.

Proof. Abusing notation, let us write $\mathbf{G}'_0$ and $\mathbf{G}_1$ for the distribution of the random variables corresponding to U's view in games $\mathbf{G}'_0$ and $\mathbf{G}_1$ respectively. For an index $i \in [0,q]$ consider the hybrid game $\mathbf{H}_i$ that answers the first i signature queries as in game $\mathbf{G}'_0$ while all the subsequent queries are answered as in $\mathbf{G}_1$. We note that $\mathbf{H}_0 = \mathbf{G}_1$ and $\mathbf{H}_q = \mathbf{G}'_0$.

We claim that for all $i \in [q]$, we have $\mathbb{SD}\left(\mathbf{H}_{i-1},\mathbf{H}_i\right) \leq 2^{-(\rho+1)}$. To see this, fix some $i \in [q]$ and denote with $\mathbf{R}_1,\ldots,\mathbf{R}_u$ (resp. $\widetilde{\mathbf{R}}_1,\ldots,\widetilde{\mathbf{R}}_u$) the random variables defined by sampling an element from $\mathcal{R}$ (resp. $\widetilde{\mathcal{R}}$) uniformly at random. Clearly,

$$
\begin{aligned}
\mathbb{SD}\left(\mathbf{H}_{i-1},\mathbf{H}_i\right) &\leq \mathbb{SD}((\mathbf{R}_1,\ldots,\mathbf{R}_u),(\widetilde{\mathbf{R}}_1,\ldots,\widetilde{\mathbf{R}}_u))\\
&\leq u\cdot\mathbb{SD}(\mathbf{R},\widetilde{\mathbf{R}}) \qquad (8)\\
&= u\cdot 2^{-(\rho+1)}, \qquad (9)
\end{aligned}
$$

where Eq. (8) follows by Lemma 4 and Eq. (9) follows by Eq. (3).

The claim now follows by the triangle inequality, as

$$\mathbb{SD}\left(\mathbf{G}_1, \mathbf{G}'_0\right) \le \sum_{i=1}^{q} \mathbb{SD}\left(\mathbf{H}_{i-1}, \mathbf{H}_i\right) \le qu \cdot 2^{-(\rho+1)}$$

and the last term becomes negligible for $u, q = poly(\kappa)$ and for ρ large enough. □

The two claims above finish the proof of statement (i).

(ii) For the second part of the proof we proceed as in Theorem 1. We note that the specified class of SAs $\mathcal{A}^{F,u}_{\mathsf{bias}}$ maintains each application of the random function f independent by appending the index j, that represents each user, to the function g, obtaining $g(\cdot) = \mathsf{Sign}(sk_j, m)||\tau||j$. The statement follows. □

B.2 Attacking Coin-Extractable Schemes (Multi-User Version)

The attack against coin-extractable schemes described in Fig. 2 becomes easily detectable in the presence of two or more users (see the proof of Theorem 7, item (iv)). An easy solution is to modify the SA class such that each algorithm in the class uses a different one-time pad key for each user it wishes to attack. We describe this class of SAs in Fig 6.

SA class $\mathcal{A}^u_{\mathsf{cext}}$

Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a coin-extractable randomized signature scheme with randomness space $\mathcal{R}$ of size $\rho = 2^d$. The class $\mathcal{A}^u_{\mathsf{cext}}$ consists of a set of algorithms $\{\widetilde{\mathsf{A}}_{\vec{s},\vec{vk},\vec{\tau}}\}_{\vec{s}\in\{0,1\}^{\ell\cdot u}, \vec{vk}\in\mathcal{VK}^u, \vec{\tau}=0^u}$, where $\ell = |sk|$, and where each algorithm in the class behaves as follows:

$\widetilde{\mathsf{A}}_{\vec{s},\vec{vk},\vec{\tau}}(sk_i, m)$:

- Parse $\vec{s}$ as $(s_1, \ldots, s_u)$, $\vec{vk}$ as $(vk_1, \ldots, vk_u)$, and $\vec{\tau}$ as $(\tau_1, \ldots, \tau_u)$.
- Find the index i such that $\mathsf{Vrfy}(vk_i, m, \mathsf{Sign}(sk_i, m)) = 1$.
- If $\tau_i \ge \ell$ output a real signature $\sigma \leftarrow \mathsf{Sign}(sk_i, m)$.
- Else,
  - For each value $j \in [d]$ compute the biased random bit $\widetilde{r}[j] = s_i[\tau_i + j] \oplus sk_i[\tau_i + j]$.
  - Return the signature $\sigma := \mathsf{Sign}(sk_i, m; \widetilde{r})$, and update the state $\tau_i \leftarrow \tau_i + d$.

Extracting the signing key. Given as input a vector of signatures $\vec{\sigma} = (\sigma_1, ..., \sigma_{\ell/d})$ of user i, represent the trapdoor s_i as ℓ/d chunks of d bits $s_i = \{s_{i,1}, \ldots, s_{i,\ell/d}\}$. For each signature $\sigma_k \in \vec{\sigma}$ try to extract the d-bit chunk $sk'_{i,k}$ of the signing key as follows.

- Extract the randomness from the k-th signature $\widetilde{r} \leftarrow \mathsf{CExt}(vk_i, m_k, \sigma_k)$.
- For each value $j \in [d]$ compute the secret key bit $sk'_{i,k}[j] = \widetilde{r}[j] \oplus s_{i,k}[j]$.

Return the signing key $sk'_i := (sk'_{i,k}, \ldots, sk'_{i,\ell/d})$.

Figure 6: Attacking coin-extractable schemes in the multi-user setting

Theorem 9. *For a randomized, ν_{ext}-coin-extractable, signature scheme $\mathcal{SS}$ with randomness space $\mathcal{R}$ of size $\rho = 2^d$, consider the class of SAs $\mathcal{A}^u_{\mathsf{cext}}$ described in Fig. 6. Then,*

(i) $\mathcal{A}^u_{\mathsf{cext}}$ *is u-users secretly undetectable.*

(ii) Each $\widetilde{\mathsf{A}} \in \mathcal{A}^u_{\mathsf{cext}}$ *recovers the signing key of any of the users with probability at least* $(1 - \nu_{ext})^{\ell/d}$.

Proof. (i) Let $\mathbf{G}$ be the game described in Definition 16, where the challenger first generates all key pairs (vk_i, sk_i) (for $i \in [u]$) and afterwards selects the algorithm $\widetilde{\mathsf{A}} \leftarrow \mathcal{A}^u_{\mathsf{cext}}$ such that $\vec{vk} := (vk_1, \ldots, vk_u)$. Consider the game $\mathbf{G}_0$, an identical copy of game $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$, an identical copy of game $\mathbf{G}$ when $b = 1$. For the first part of the proof the objective is to show that $\mathbf{G}_0 \approx \mathbf{G}_1$.

Claim 9. $\mathbf{G}_0 \equiv \mathbf{G}_1$.

Proof. Abusing notation, let us write $\mathbf{G}_0$ and $\mathbf{G}_1$ for the distribution of the random variables corresponding to U's view in games $\mathbf{G}_0$ and $\mathbf{G}_1$ respectively. For an index $i \in [0, q]$ consider the hybrid game $\mathbf{H}_i$ that answers the first i signature queries as in game $\mathbf{G}_0$ while all the subsequent queries are answered as in $\mathbf{G}_1$. We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_q \equiv \mathbf{G}_0$.

We claim that for all $i \in [q]$, we have $\mathbf{H}_{i-1} \equiv \mathbf{H}_i$. To see this, fix some $i \in [q]$ and denote with $\mathbf{R}_1, \ldots, \mathbf{R}_u$ the random variables defined by sampling an element from $\mathcal{R}$ uniformly at random and with $\widetilde{\mathbf{R}}_1, \ldots, \widetilde{\mathbf{R}}_u$ the random variables defined by sampling an element from the biased distribution $\widetilde{\mathcal{R}}$ also uniformly at random. It is easy to see that $\mathbf{R}_i$ and $\widetilde{\mathbf{R}}_i$, for $i \in [q]$, are identically distributed, as the biased distribution consists of a one-time pad encryption of (part of) the signing key with a uniform key (a different key for each user). The claim follows. □

(ii) For the second part of the proof we note that the attack of Fig. 6 successfully recovers the biased randomness $\widetilde{r}$ of each $\sigma_i \in \{\sigma_1, \ldots, \sigma_{\ell/d}\}$ and computes the chunk $sk_{j,i}$ of the signing key of a user j with probability at least $1 - \nu_{ext}$. This gives a total probability of recovering an entire signing key of at least $(1 - \nu_{ext})^{\ell/d}$. □