A preliminary version of this paper is published in the proceedings of the 22nd ACM Conference on Computer and Communications Security—CCS 2015. This is the full version.

Subversion-Resilient Signatures: Definitions, Constructions and Applications

Giuseppe Ateniese[1], Bernardo Magri[2], and Daniele Venturi[3]

[1] *Stevens Institute of Technology, USA*
[2] *Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany*
[3] *Sapienza University of Rome, Italy*

November 3, 2018

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.** We show that a broad class of randomized signature schemes is insecure against *stateful* SAs, even if using just a single bit of randomness. On the other hand, we establish that signature schemes with enough min-entropy are insecure against *stateless* SAs. The attacks we design are undetectable to the end-users (even if they know the signing key).
- **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. As our second positive result, we show how to construct subversion-resilient identification schemes from subversion-resilient signature schemes. 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 us to successfully protect so-called re-randomizable signature schemes (which include unique signatures as a 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.

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 few 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 is open to debate, and it is indeed being furiously discussed among policy makers, the public, and the intelligence community.

The ability of an adversary to replace 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. Clearly, stateless schemes are preferable as they are easier to deploy effectively, and indeed, most of the 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 subversion 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.

The above definition is very general, as it covers adaptive and continuous *tampering with the computation* performed by the signing algorithm (within the class $\mathcal{A}$). In particular, it includes algorithm-substitution and kleptographic attacks as a special case, but additionally models, e.g., a machine running a signature software infected by multiple viruses and malware (e.g., due to buffer overflow attacks [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 comprehensive discussion.

Clearly, without making any restriction on the class $\mathcal{A}$ (or without making additional assumptions) there is no hope for security: An arbitrarily 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, DMS16]. 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 *undetectable*. Roughly, this means that there exists a single efficient sampling strategy yielding an algorithm $\widetilde{\mathsf{A}}$ from the class $\mathcal{A}$ such that the following holds: (i) no user can distinguish black-box access to the genuine signature algorithm from black-box access to the subverted signature algorithm $\widetilde{\mathsf{A}}$; (ii) an adversary given a polynomial number of samples from $\widetilde{\mathsf{A}}$, no matter how the samples are chosen, can recover the signing key with high probability. See Section 3.2 for a precise definition. Our definitions of undetectability are similar in spirit to the ones put forward by [BPR14, BJK15] 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, or even is allowed to choose, 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.* [BPR14], 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 [BJK15] to the case of signature schemes; similar to [BJK15], our attack is completly stateless (i.e., the class of SAs we consider does not need to maintain a state across invocations), which makes it undetectable even in the presence of state resets.

The above generic attack requires that the signature scheme uses a minimal amount of randomness. 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.

Our second attack is stateful, in that a counter has to be maintained by the subverted algorithm in order to leak different parts of the secret key. We leave it as an open problem to design a stateless attack that works for (coin-extractable) signature schemes with small randomness.

Positive results. As a first positive result, we show that fully deterministic schemes with *unique*[2] signatures are existentially unforgeable under chosen-message attacks against the class of SAs that satisfies the so-called verifiability condition.[3] This 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).

The above fact is reminiscent of the main positive result in [BPR14], who also showed that *stateful* symmetric encryption schemes with unique ciphertexts remain CPA-secure in the presence of one-time SAs that satisfy a similar "decryptability condition." Unique ciphertexts here means that, for any given key, message, associated data and state, there exists at most one ciphertext that the receiver will decrypt to the message in question. The main difference between the two results is that, for the case of signatures, we achieve security against adaptive and continuous attacks, without the need of keeping any state.

Clearly, the assumption that the verifiability condition should hold for all messages is quite a strong one. Hence, we also relax the verifiability condition to hold for all but a negligible fraction of the messages. However, we are not able to prove that unique signatures achieve existential unforgeability under chosen-message attacks against the class of SAs that satisfies

[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 an honestly generated verification key there is a single valid signature for each message.

[3] One might ask whether a similar result holds for all deterministic schemes where signatures are not unique; the answer to this question is negative as our attacks also apply to certain types of deterministic schemes (e.g., de-randomized schemes—see the proof of Theorem 9 in Section 7.2).

relaxed verifiability.[4] Instead, as our second positive result, we show that unique signatures are existentially unforgeable under random-message attacks (where the adversary can only see potentially subverted signatures of random messages) against the class of SAs that satisfies relaxed verifiability. Interestingly, this weaker security flavor is still useful for applications, e.g. to construct subversion-resilient identification schemes.

As our third positive result, we provide a way to achieve the ambitious goal of protecting signature schemes against *arbitrary* SAs, relying on a cryptographic reverse firewall. The latter primitive was introduced in [MS15] (see also [DMS16]) 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 an algorithm taking as input a message/signature pair (m, σ), 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 [HJK12]) admits 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 identically to a freshly generated signature on m under signing key sk (corresponding to 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.[5] 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 (different) protocol that admits a RF preserving functionality and preventing leakage to an eavesdropper. 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 only a single user. In Section 7, 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 $u \geq 1$ 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), and the same holds for *secret* undetectablity in case of stateless subversion.[6] This does not hold for *public* undetectability though, as there exist

[4] In fact, as shown 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 achieve security under chosen-message attacks against the class of SAs that meets relaxed verifiability. See Section 1.3 for more details.

[5] 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.

[6] A previous version of this paper [AMV15] considered a weaker flavour of secret undetectability that does not immediately generalize to the multi-user setting.

classes of SAs that are publicly undetectable by a single user but can be efficiently publicly detected by more than one user, as shown in Theorem 9 (iii).

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, public-key cryptography typically involves 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 subversion attacks, and as such they should be used with caution. The positive news, however, is that there already exist signature schemes that are subversion-resilient, and they are very efficient and well established.

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

Subliminal channels. Remarkably, digital signatures were the first cryptographic primitive used to create a subliminal channel in order to solve Simmons' "Prisoners' Problem" [Sim83]: By agreeing on a partition of the secret keys into two sets (one set for "0" and the other set for "1"), two prisoners can communicate confidentially over an insecure channel, without being detected by the warden which knows all keys and reads the entire communication. Later work [Sim84, Sim85, Sim93, Sim94, AVPN96, Sim98] showed that several digital signature schemes, and even natural ones, such as the Digital Signature Algorithm [Sim93, AVPN96], admit subliminal channels.

After their introduction, the potential of subliminal channels has been explored in other settings beyond digital signatures (e.g., [Des88a, Des88b, BDI$^+$99]); this line of research led, for instance, to the concept of divertible protocols, that are intimately related to reverse firewalls.

Backdoored implementations. The setting of backdoored implementations includes, in particular, the realm of kleptography and SETUP attacks (see [YY04] for a survey). Dodis *et al.* [DGG$^+$15] provide a formal treatment of backdoored pseudorandom generators (building on previous work of Vazirani and Vazirani [VV83]); their work has been extended in [DPSW16] to the setting of robust pseudorandom generators with inputs. Subversion of pseudorandom generators is of particular importance, given the potential sabotage of the NIST Dual EC PRG [NIS07]. The problem of parameters subversion has also been considered in the context of zero-knowledge proofs [BFS16], and public-key encryption [ABK18].

Bellare and Hoang [BH15] tackle the question of SAs in the setting of public-key encryption (PKE). In particular, they give a standard model construction of an IND-secure[7] deterministic

[7] In the context of deterministic PKE, IND security captures the best possible privacy in terms of semantic security in the presence of unpredictable messages (that do not depend on the public key).

tic PKE [BFOR08]—although leveraging strong tools such as universal computational extractors [BHK13] and lossy trapdoor functions [PW11]—and show a generic transformation from any such PKE to a unique-ciphertext PKE, which in turn achieves IND-security against the class of one-time SAs meeting the decryptability condition.

Russell *et al.* [RTYZ16, RTYZ17, RTYZ18] consider the setting of *complete* subversion, where all algorithms (including, for instance, the key generation algorithm) are subject to kleptographic attacks, and show how to build (trapdoor) one-way permutations, pseudorandom generators, digital signatures, and chosen-plaintext secure encryption in this model, by relying on the random oracle methodology [BR96].

Input-triggered subversions. 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 $\bar{m}$ outputs the secret key, but otherwise behaves like the genuine signature algorithm. Clearly this class of SAs will be undetectable by the users, as without knowing the trapdoor, there is only a negligible chance to query the secret message $\bar{m}$ and detect that the signature algorithm was subverted (at least if the message space is large enough). Yet, an adversary mounting a chosen-message attack can recover the signing key by asking a signature for message $\bar{m}$.

As a consequence, it is impossible to prove existential unforgeability under chosen-message attacks against such "input-triggered" subversions (in the BPR model). Note, however, that for the case of signatures, one can still prove a positive result by restricting the adversary to only see signatures of random messages (i.e., in case of a random-message attack). Indeed, input-triggered subversions meet our notion of relaxed verifiability (see Section 1.1), and thus our positive results for unique signatures apply to such case.

The solution proposed by DFP is to 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.[8]

It is possible to make a similar change as in [DFP15] and adapt the DFP model to signature schemes in order to achieve security under chosen-message attacks. The end result would share some similarities with our approach using cryptographic RFs; 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, especially in cases where the user's machine is compromised; instead, in our framework, a cryptographic RF for a signature scheme relies only on public information and could easily be located on an (untrusted) external proxy.

Tampering attacks. A related line of research analyzes 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 cryp-

[8] 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.

toscheme 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 predicates [GL10], hash functions [GOR11], public-key encryption [AHI11, Wee12], and digital signatures and identification protocols [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].

Conference version. An abridged version of this paper appeared as [AMV15]. The current version contains significantly revised proofs and new material, including the stateless undetectable attack against signature schemes with non-trivial randomness, the definition and construction of subversion-resilient identification schemes, and a complete treatment of the multi-user setting.

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 $poly(|x|)$ steps.

We denote with $\kappa \in \mathbb{N}$ the security parameter. A function $\varepsilon : \mathbb{N} \to [0,1]$ is negligible in the security parameter (or simply negligible) if it vanishes faster than the inverse of any polynomial in κ, i.e. $\varepsilon(\kappa) = \kappa^{-\omega(1)}$. We sometimes write $negl(\kappa)$ to denote an unspecified negligible function in the security parameter.

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

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

Proof. The lemma follows by the definition of statistical distance and an application of the triangle inequality:

$$\begin{aligned}\mathbb{SD}(\mathbf{A};\mathbf{B}) &= \frac{1}{2}\sum_{x\in\mathcal{D}}\left|\underset{\mathbf{A}}{\mathbb{P}}[x] - \underset{\mathbf{B}}{\mathbb{P}}[x]\right| \\ &= \frac{1}{2}\sum_{x\in\mathcal{D}}\left|\underset{\mathbf{A}}{\mathbb{P}}[x|E]\cdot\underset{\mathbf{A}}{\mathbb{P}}[E] - \underset{\mathbf{B}}{\mathbb{P}}[x|E]\cdot\underset{\mathbf{B}}{\mathbb{P}}[E] + \underset{\mathbf{A}}{\mathbb{P}}[x|\neg E]\cdot\underset{\mathbf{A}}{\mathbb{P}}[\neg E] - \underset{\mathbf{B}}{\mathbb{P}}[x|\neg E]\cdot\underset{\mathbf{B}}{\mathbb{P}}[\neg E]\right| \\ &\le \mathbb{P}[E]\cdot\mathbb{SD}(\mathbf{A}|E;\mathbf{B}|E) + \mathbb{P}[\neg E]\cdot\mathbb{SD}(\mathbf{A}|\neg E;\mathbf{B}|\neg E) \\ &\le \mathbb{SD}(\mathbf{A}|\neg E;\mathbf{B}|\neg E) + \mathbb{P}[E].\end{aligned}$$

□

2.2 Signature Schemes

A signature scheme is a triple of polynomial 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 $\mathsf{KGen}(1^\kappa)$; associated to each $vk \in \mathcal{VK}$ are a message space $\mathcal{M} := \mathcal{M}_{vk}$, a randomness space $\mathcal{R} := \mathcal{R}_{vk}$, and a signature space $\Sigma := \Sigma_{vk}$. (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 $(m,\sigma) \in (\{0,1\}^*)^2$, 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 informally says that verifying honestly generated signatures always[9] works.

Definition 1 (Correctness of signatures)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. We say that $\mathcal{SS}$ satisfies (perfect) correctness if for all $\kappa \in \mathbb{N}$, for all (vk, sk) output by $\mathsf{KGen}(1^\kappa)$, and all $m \in \mathcal{M}$,

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \mathsf{Sign}(sk, m))) = 1\right] = 1,$$

where the probability is taken over the randomness of the signing algorithm.

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)-EUF-CMA in short) if for all adversaries A running in time t it holds:

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m^*, \sigma^*)) = 1 \wedge m^* \not\in \mathcal{Q} :\ (vk, sk) \leftarrow_{\$} \mathsf{KGen}(1^\kappa); (m^*, \sigma^*) \leftarrow_{\$} \mathsf{A}^{\mathsf{Sign}(sk,\cdot)}(vk)\right] \le \varepsilon,$$

where $\mathcal{Q} = \{m_1, \ldots, m_q\}$ denotes the set of queries to the signing oracle. If for all $t, q = poly(\kappa)$ there exists $\varepsilon(\kappa) = negl(\kappa)$ such that $\mathcal{SS}$ is (t, q, ε)-EUF-CMA, then we simply say $\mathcal{SS}$ is EUF-CMA.

[9] All our results can be extended to the case where there is a negligible correctness error. Note, however, that assuming perfect correctness is wlog. if we allow the signing algorithm to run in expected polynomial time.

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 only a single signature that verifies w.r.t. an honestly generated verification key.

Definition 3 (Uniqueness of signatures)**.** Let $\mathcal{SS}$ be a signature scheme. We say that $\mathcal{SS}$ satisfies uniqueness if for all $\kappa \in \mathbb{N}$, for all vk output by $\mathsf{KGen}(1^\kappa)$, and all $m \in \mathcal{M}$, there exists a single value $\sigma \in \Sigma$ such that $\mathsf{Vrfy}(vk, (m, \sigma)) = 1$.

Full Domain Hash signatures using trapdoor permutations, for instance RSA-FDH [BR96], are unique. Sometimes unique signatures are also known under the name of *verifiable unpredictable functions* (VUFs).[10] 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 Generators

We say that $G : \{0,1\}^\kappa \to \{0,1\}^\kappa \times \{0,1\}^d$ is a stateful pseudorandom generator (PRG) if G is polynomial-time computable and $d \geq 1$. A run of the PRG yields $G(s_{i-1}) = (s_i, v_i)$ where $s_0 \in \{0,1\}^\kappa$ is the initial seed, and (s_i, v_i) are, respectively, the seed and the output at the i-th iteration. Security of a PRG demands that its outputs are computationally indistinguishable from a uniform bitstring, for all efficient distinguishers.

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

$$\left| \underset{s_0 \leftarrow\$ \{0,1\}^\kappa}{\mathbb{P}} \left[\mathsf{D}(v_1, \ldots, v_q) = 1 : \ \forall i \in [q], G(s_{i-1}) = (s_i, v_i)\right] - \underset{v_1,\ldots,v_q \leftarrow\$ \mathbf{U}_d}{\mathbb{P}} \left[\mathsf{D}(v_1, \ldots, v_q) = 1\right] \right| \leq \varepsilon,$$

where $\mathbf{U}_d$ is the uniform distribution over d-bit strings.

2.4 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 $\mathcal{Y}$. Intuitively, a pseudorandom function (PRF) is a function that is computationally indistinguishable from a truly random function, for all efficient distinguishers.

Definition 5 (Pseudorandom function)**.** A function $F : \{0,1\}^\kappa \times \mathcal{X} \to \mathcal{Y}$ is a (t, q, ε)-secure pseudorandom function, 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.

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

[10] 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.

(SAs). A subversion of $\mathcal{SS}$ is a randomized algorithm $\widetilde{\mathsf{A}} \in \mathcal{A}$ taking as input a signing key $sk \in \mathcal{SK}$ and a message $m \in \mathcal{M}$, and outputting a subverted signature $\widetilde{\sigma} \in \{0,1\}^*$, where $\widetilde{\sigma} := \widetilde{\mathsf{A}}(sk, m; r)$ for $r \leftarrow\!\!\$\; \{0,1\}^*$.

Notice that algorithm $\widetilde{\mathsf{A}}$ is completely arbitrary; in particular, its randomness space does not need to be equal to the randomness space $\mathcal{R}$, and the value $\widetilde{\sigma}$ could be outside the range Σ of the original signing algorithm. Moreover, algorithm $\widetilde{\mathsf{A}}$ can hard-wire arbitrary auxiliary information chosen by the adversary, which we sometimes denote by a string $\alpha \in \{0,1\}^*$. Finally, 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 revealed 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. Apart from being undetectable, a SA class should successfully extract the secret key of the underlying signature scheme, a goal that we formalize in Section 3.3. Intuitively, the last two adversarial goals (i.e., undetectability and key recovery) must be satisfied at the same time by a single, and efficient, sampling strategy that selects one subversion algorithm for a given SA class; this yields a successful class of SAs, as defined in Section 3.4.

3.1 Security

We consider two security definitions, corresponding to different adversarial goals.

Indistinguishability. In the first definition, it is required that an adversary B, with the ability to subvert the original signing algorithm polynomially many time (possibly depending on the user's verification key), cannot distinguish signatures produced via the genuine signing algorithm from subverted signatures.

Definition 6 (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 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 specify an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, for a total of at most $n \in \mathbb{N}$ queries. Each such algorithm implicitly defines an oracle that can be queried adaptively up to $q \in \mathbb{N}$ times.
    - Upon input a query of the form $(j, m_{i,j})$, where $j \in [n]$ and $i \in [q]$, the answer from the j-th oracle depends on the value of the secret bit b: If $b = 1$, the output is $\sigma_{i,j} \leftarrow\!\!\$\; \mathsf{Sign}(sk, m_{i,j})$; if $b = 0$, the output is $\widetilde{\sigma}_{i,j} \leftarrow\!\!\$\; \widetilde{\mathsf{A}}_j(sk, m_{i,j})$. In case $\widetilde{\mathsf{A}}_j$ is undefined, the oracle returns $\perp$.
    - Note that B does not need to ask all q queries before choosing the next algorithm, i.e. the queries to each oracle $\widetilde{\mathsf{A}}_j$ can be interleaved in an arbitrary manner.
3. Finally, B outputs a value $b' \in \{0,1\}$; we say that B wins iff $b' = b$.

If for all $t, q, n = poly(\kappa)$ there exists $\varepsilon(\kappa) = negl(\kappa)$ such that $\mathcal{SS}$ is (t, n, q, ε)-indistinguishable w.r.t *continuous* $\mathcal{A}$-SAs, we simply say that $\mathcal{SS}$ is indistinguishable against continuous $\mathcal{A}$-SAs.

Impersonation under chosen-message attacks. We also consider an alternative definition, where the goal of the adversary is to forge a signature on a "fresh" message (not asked to any of the oracles). As we show in Theorem 8, this definition is strictly weaker than Definition 6 for all signature schemes that are unforgeable (i.e., UF-CMA as per Definition 2).

Definition 7 (EUF-CMA 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)-EUF-CMA w.r.t. *continuous* $\mathcal{A}$-SAs if for all adversaries B running in time t, we have $\mathbb{P}[\mathsf{B} \text{ wins}] \le \varepsilon(\kappa)$ 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 $\sigma_i \leftarrow\$ \mathsf{Sign}(sk, m_i)$ for a total of at most $q \in \mathbb{N}$ queries.
3. The adversary B can specify an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, for a total of at most $n \in \mathbb{N}$ queries. Each such algorithm implicitly defines an oracle that can be queried adaptively up to $q \in \mathbb{N}$ times.
    - Upon input a query of the form $(j, m_{i,j})$, where $j \in [n]$ and $i \in [q]$, the j-th oracle outputs $\widetilde{\sigma}_{i,j} \leftarrow\$ \widetilde{\mathsf{A}}_j(sk, m_{i,j})$. In case $\widetilde{\mathsf{A}}_j$ is undefined, the oracle returns $\perp$.
    - Note that B does not need to ask all q queries before choosing the next algorithm, i.e. the queries to each oracle $\widetilde{\mathsf{A}}_j$ and to oracle Sign can be interleaved in an arbitrary manner.
4. Let $\mathcal{Q} = \{m_1, \dots, m_q\}$ be the set of messages queried to oracle $\mathsf{Sign}(sk, \cdot)$; similarly, for each $j \in [n]$, let $\widetilde{\mathcal{Q}}_j = \{m_{1,j}, \dots, m_{q,j}\}$ be the set of messages queried to oracle $\widetilde{\mathsf{A}}_j$.
5. 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 $\widetilde{\mathcal{Q}} := \bigcup_{j=1}^{n} \widetilde{\mathcal{Q}}_j$.

If for all $t, q, n = poly(\kappa)$ there exists $\varepsilon(\kappa) = negl(\kappa)$ such that $\mathcal{SS}$ is (t, n, q, ε)-EUF-CMA against continuous $\mathcal{A}$-SAs, we simply say that $\mathcal{SS}$ is EUF-CMA against continuous $\mathcal{A}$-SAs.

Remarks. 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 6 (and consequently Definition 7) for an *arbitrary* class $\mathcal{A}$, without making further assumptions.[11] To see this, consider the simple algorithm that ignores all inputs and outputs the secret key.[12]
- Consider the 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 define $\widetilde{\mathsf{A}}_f(\cdot, m; r) := \mathsf{Sign}(f(\cdot), m; r)$. Note that continuous security against $\mathcal{A}_{\mathsf{key}}$-SAs implies security in the presence of related-key attacks within the family $\mathcal{F}$. It is also worth noting that, already for $n = 1$, Definition 7 implies EUF-CMA security under *non-adaptive* key tampering, as a single subverted algorithm can hard-wire (the description of) polynomially many pre-set tampering functions.

[11] 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.

[12] 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 single subversion algorithm $\widetilde{\mathsf{A}}_{\bar{m}}$ that always outputs a signature $\bar{\sigma}$ on some hard-wired message $\bar{m} \in \mathcal{M}$; obviously this subversion allows to forge on $\bar{m}$ without explicitly querying the message $\bar{m}$ to any of the oracles.

- We note that each algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$ keeps its own state, say $\tau_j \in \{0,1\}^*$, which gets updated at each invocation. However, this state is not shared among different algorithms within the class $\mathcal{A}$. This models, e.g., a machine infected by multiple (but different) viruses, causing the execution of malicious code.

Multi-user setting. For simplicity, Definition 6 and 7 consider a single user. We provide an extension to the more general setting with $u \geq 1$ users, together with a complete picture of the relationships between different notions, in Section 7.

3.2 Public/Secret Undetectability

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. While reading it, bear in mind that the challenger plays the role of the "bad guy", trying to sabotage the signature scheme without being detected. The definition is parameterized by a distribution over the class of SAs, corresponding to an efficient strategy by which the saboteur can (efficiently) select a specific algorithm from the SA class in such a way that undetectability holds (with high probability).

Definition 8 (Public/secret undetectability)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$, and $\mathbf{D}_{\mathcal{A}}$ be an efficiently samplable distribution over $\mathcal{A}$. We say that $\mathcal{A}$ is *secretly* $(t, q, \varepsilon, \mathbf{D}_{\mathcal{A}})$-undetectable w.r.t. $\mathcal{SS}$ if for all users U running in time t, we have that $\left|\mathbb{P}[\mathsf{U} \text{ wins}] - \frac{1}{2}\right| \leq \varepsilon(\kappa)$ in the following game:

1. The challenger samples an algorithm $\widetilde{\mathsf{A}} \leftarrow\$ \mathbf{D}_{\mathcal{A}}$, and picks $b \leftarrow\$ \{0,1\}$.
2. The user U can ask queries (sk_i, m_i), where $sk_i \in \mathcal{SK}$ and $m_i \in \mathcal{M}$, for all $i \in [q]$. The answer to each query depends on the secret bit b. If $b = 1$, the challenger returns $\sigma_i \leftarrow\$ \mathsf{Sign}(sk_i, m_i)$; if $b = 0$, the challenger returns $\widetilde{\sigma}_i \leftarrow\$ \widetilde{\mathsf{A}}(sk_i, m_i)$.
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 the challenger picks $(vk, sk) \leftarrow\$ \mathsf{KGen}(1^\kappa)$ and gives vk to U, and the queries specified by U in step 2 only consist of values $m_i \in \mathcal{M}$ upon which the challenger replies with either $\sigma_i \leftarrow\$ \mathsf{Sign}(sk, m_i)$ or with $\widetilde{\sigma}_i \leftarrow\$ \widetilde{\mathsf{A}}(sk, m_i)$ (depending on the value of the hidden bit b). Moreover, if for all $t, q = poly(\kappa)$ there exists $\varepsilon(\kappa) = negl(\kappa)$ such that $\mathcal{A}$ is $(t, q, \varepsilon, \mathbf{D}_{\mathcal{A}})$-secretly/publicly undetectable w.r.t. $\mathcal{SS}$, we simply say that $\mathcal{SS}$ is secretly/publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$.

Our definition of undetectability is similar to the corresponding definition considered by Bellare *et al.* [BPR14, BJK15] for the case of symmetric encryption. One key difference is that, in the definition above, the challenger is required to choose the subversion algorithm $\widetilde{\mathsf{A}} \in \mathcal{A}$ following some polynomial-time sampling strategy $\mathbf{D}_{\mathcal{A}}$ that is a parameter in the definition. We stress that achieving Definition 8 in isolation might be trivial;[13] however, the saboteur's goal is to design an attack class $\mathcal{A}$ that breaks security (e.g., by exposing the signing key—see Section 3.3) while at the same time being (publicly or secretly) undetectable for the same sampling distribution $\mathbf{D}_{\mathcal{A}}$. See Section 3.4 for further discussion on this point.

[13] It suffices, e.g., to artificially include the original signing algorithm in any class of SAs.

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 analyze in Section 4 are non-adaptive, and only require to use a single subversion.

Secret vs. public undetectability. While secret undetectability clearly implies public undetectability, the converse is not true. In particular, in Section 7.2, 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. $\mathcal{SS}$.

Strong undetectability. The notion of *strong* undetectability, introduced in [BJK15], basically limits the class of undetectable subversions to be the class of stateless subversions, which is enforced by the challenger returning the state to the user in the definition of undetectability. Since we will construct both stateless and stateful attacks, we will equivalently say that a class of SAs $\mathcal{A}$ is strongly (publicly or secretly) undetectable, if it satisfies Definition 8, and additionally $\mathcal{A}$ is a family of stateless algorithms.

Multi-user setting. For simplicity, Definition 8 considers a single user. We provide an extension to the more general setting with $u \geq 1$ users, together with a complete picture of the relationships between different notions, in Section 7.

3.3 Signing Key Recovery

Apart from being (publicly or secretly) $\mathbf{D}_{\mathcal{A}}$-undetectable, a class $\mathcal{A}$ of SAs w.r.t. a signature scheme $\mathcal{SS}$ should yield some advantage from the point of view of the saboteur. For instance, one could imagine the saboteur being able to distinguish subverted signatures from real ones, or to produce signature forgeries on chosen messages. Below, we formalize an even more ambitious goal which is the ability of fully recovering the secret signing key.

Definition 9 (Signing key recovery)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$, and $\mathbf{D}_{\mathcal{A}}$ be an efficiently samplable distribution over $\mathcal{A}$. We say that adversary B $(t, q, \varepsilon, \mathbf{D}_{\mathcal{A}}, \mathsf{M})$-recovers the signing key of $\mathcal{SS}$ if B runs in time t, and moreover $\mathbb{P}[\mathsf{B} \text{ wins}] \geq \varepsilon(\kappa)$ in the following game:

1. The challenger runs $(vk, sk) \leftarrow\!\!\$\, \mathsf{KGen}(1^\kappa)$, samples $\widetilde{\mathsf{A}} \leftarrow\!\!\$\, \mathbf{D}_{\mathcal{A}}$, and gives vk to B.
2. The adversary B is given access to an oracle that can be queried at most q times: Upon an empty input, the oracle picks $m_i \leftarrow\!\!\$\, \mathsf{M}$, computes $\widetilde{\sigma}_i \leftarrow\!\!\$\, \widetilde{\mathsf{A}}(sk, m_i)$, and sends $(m_i, \widetilde{\sigma}_i)$ to B.
3. Finally, B outputs a key sk'; we say that B wins iff $sk' = sk$.

Whenever $\varepsilon(\kappa)$ is non-negligible, for some $t, q = poly(\kappa)$, we simply say that the attacker $(\mathbf{D}_{\mathcal{A}}, \mathsf{M})$-recovers the signing key of $\mathcal{SS}$.

We note that in the above definition, M is a message sampler algorithm that chooses the message to be signed according to some pre-defined strategy. From the perspective of an adversary, an attack is stronger the less it assumes about M; in fact, the strongest attack is one that works for any M, i.e., regardless of which messages the signer chooses to sign.

3.4 Successful SAs

Note that designing a class of SAs that achieves either (public/secret) undetectability or key recovery in isolation (w.r.t. some efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$) might be trivial. For instance, consider the class of SAs $\mathcal{A}_{\mathsf{bad}}$ that contains only two algorithms: the original signing algorithm, and the constant function whose output is always equal to the signing key. Clearly, such class satisfies undetectability w.r.t. the distribution that always returns the first algorithm in the class, whereas the same class satisfies key recovery (for any message sampler algorithm) w.r.t. the distribution that always returns the second algorithm in the class.

However the challenge, from the perspective of an attacker, is to design a class of SAs that admits a single sampling strategy achieving at the same time undetectability and key recovery. This motivates the following definition.

Definition 10 (Successful SA)**.** Let $\mathcal{SS}$ be a signature scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$. We call $\mathcal{A}$ $((t,t'),(q,q'),(\varepsilon,\varepsilon'))$-successful w.r.t. $\mathcal{SS}$ if there exists an adversary B and a distribution $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, such that the following conditions are met:

(i) $\mathcal{A}$ is secretly/publicly $(t',q',\varepsilon',\mathbf{D}_{\mathcal{A}})$-undetectable w.r.t. $\mathcal{SS}$;

(ii) For any message sampler algorithm M, attacker B $(t,q,\varepsilon,\mathbf{D}_{\mathcal{A}},\mathsf{M})$-recovers the signing key of $\mathcal{SS}$.

We observe that it is still trivial to design classes of SAs that are successful for some uninteresting range of the parameters. For instance, let $\mathcal{SS}$ be EUF-CMA and consider the uniform distribution over the above defined class of SAs $\mathcal{A}_{\mathsf{bad}}$. The latter yields large key-recovery probability $\varepsilon \geq 1/2$, but also large detection advantage $\varepsilon' \geq 1/4$. Looking ahead, our attacks provide classes of SAs that are successful for tiny ε', large ε, and reasonably small t, t', q, q'. (In the asymptotic setting, negligible ε', overwhelming ε, and polynomial t, t', q, q'.)

4 Mounting Subversion Attacks

In Section 4.1, we show that the biased-randomness attack of [BJK15] (adapted to the case of signatures) yields a successful class of SAs, as per Definition 10, against all signature schemes using a sufficient amount of randomness. This attack is completely *stateless.* In Section 4.2, we present a *stateful* attack that is successful even for signatures using only little randomness (in fact, even 1 bit), provided that the targeted scheme satisfies an additional property.

4.1 Non-Trivial Randomness Attack

In this section, we describe a strongly undetectable[14] attack against all probabilistic signature schemes with a non-trivial amount of randomness. We measure the randomness of a signature scheme via a notion of min-entropy, which is adapted from [BJK15].

Definition 11 (Signatures min-entropy)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. For all $\kappa \in \mathbb{N}$, we say that $\mathcal{SS}$ has min-entropy $\eta(\kappa) \in \mathbb{N}$ if the following holds:

$$2^{-\eta} = \max_{sk\in\mathcal{SK},m\in\mathcal{M},\sigma\in\Sigma} \mathbb{P}\left[\mathsf{Sign}(sk,m;r) := \sigma\right], \tag{1}$$

where $\mathcal{M}$, $\mathcal{R}$, $\mathcal{SK}$, Σ denote, respectively, the message/randomness/secret-key/signature space of $\mathcal{SS}$, and where the probability is taken over the choice of the random coins $r \in \mathcal{R}$.

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

Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a probabilistic signature scheme with signature space Σ, randomness space $\mathcal{R}$, and signing key space $\mathcal{SK} = \{0,1\}^\ell$. Also, let $F : \{0,1\}^\kappa \times \Sigma \to \{0,1\} \times [\ell]$ be a PRF. The class $\mathcal{A}_{\mathsf{bias}}$ consists of a set of algorithms $\{\widetilde{\mathsf{A}}_{s,\tau}\}_{s \in \{0,1\}^\kappa}$, for some $\tau \in \mathbb{N}$, where each algorithm in the class behaves as follows.

```
$\widetilde{\mathsf{A}}_{s,\tau}(sk, m)$:
  • Initialize counter $j := 0$.
  • Repeat the following instructions, until $(sk[i] = v \vee j = \tau)$:
      – Increment the counter $j := j + 1$;
      – Sample $r \leftarrow\!\!\$\, \mathcal{R}$;
      – Compute a fresh signature $\sigma = \mathsf{Sign}(sk, m; r)$;
      – Evaluate $(v, i) = F_s(\sigma)$.
  • Return $\sigma$ as the signature of $m$.
```

Figure 1: Strongly undetectable attack against probabilistic signature schemes with non-trivial randomness.

The intuition behind the attack is as follows. Let F be a PRF (cf. Section 2.4) with range $\{0,1\} \times [\ell]$, where ℓ is the length of the secret key in bits. We consider the class of SAs which outputs a real signature, subject to the constraint that the output of the PRF (for a key which is embedded in the subverted implementation and known only by the adversary) evaluated on the computed signature represents a correct guess for one of the bits of the secret key. As the analysis shows, this allows to recover the entire key with high probability.

Theorem 1. *Let $\mathcal{SS}$ be a randomized signature scheme with signature space Σ, secret key space $\mathcal{SK} = \{0,1\}^\ell$, and min-entropy $\eta \in \mathbb{N}$. Consider the class of SAs $\mathcal{A}_{\mathsf{bias}}$ described in Fig. 1, and let $F : \{0,1\}^\kappa \times \Sigma \to \{0,1\} \times [\ell]$ be a $(t_{\mathsf{prf}}, q_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}})$-secure PRF, $\mathbf{U}_{\mathsf{bias}}$ be the uniform distribution over $\mathcal{A}_{\mathsf{bias}}$, and M be an arbitrary message sampling algorithm. Then:*

(i) There exists an adversary B (and we describe it in the proof of this theorem) that $(t, q, \varepsilon, \mathbf{U}_{\mathsf{bias}}, \mathsf{M})$-recovers the signing key of $\mathcal{SS}$, for any $\tau, q \in \mathbb{N}$ and with

$$\varepsilon \geq 1 - (\varepsilon_{\mathsf{prf}} + \ell \cdot e^{-q/\ell} + q^2\tau^2 \cdot 2^{-\eta-1} + q \cdot 2^{-\tau}),$$

as long as $q_{\mathsf{prf}} \geq \tau \cdot q$, and t_{prf} is at least $\tau \cdot q$ times the sum of the running time of algorithm Sign plus the running time required to sample an element from M. The running time t of B is at most q times the time to compute F, whereas the maximum running time of each $\widetilde{\mathsf{A}}_{s,\tau} \in \mathcal{A}_{\mathsf{bias}}$ is roughly τ times the sum of the running time of algorithm Sign plus the time to compute F.

(ii) For any $\tau, q \in \mathbb{N}$, the class $\mathcal{A}_{\mathsf{bias}}$ is strongly secretly $(t_{\mathsf{prf}}, q, \varepsilon_{\mathsf{prf}} + q^2\tau^2 \cdot 2^{-\eta-1}, \mathbf{U}_{\mathsf{bias}})$-undetectable.

Proof. (i) Consider the following adversary B playing the game described in Definition 9, for the SA class $\mathcal{A}_{\mathsf{bias}}$ described in Fig. 1.

[14] This terminology is inherited from [BJK15], and it simply means that the subversion is undetectable and *stateless*.

Adversary B:

- Initialize $sk' := 0^\ell$.
- The challenger chooses a subversion algorithm $\widetilde{\mathsf{A}}_{s,\tau}$ uniformly at random from the SA class $\mathcal{A}_{\mathsf{bias}}$. (This is equivalent to sampling the algorithm from the distribution $\mathbf{U}_{\mathsf{bias}}$, and implicitly corresponds to picking a random key $s \in \{0,1\}^\kappa$ for the PRF.)
- For $j \in [q]$ do:
  - Make a query to the signature oracle, receiving a pair (m_j, σ_j) such that $\sigma_j \leftarrow \widetilde{\mathsf{A}}_{s,\tau}(sk, m_j)$ for $m_j \leftarrow \mathsf{M}$;
  - Compute $(v, i) = F_s(\sigma_j)$.
  - Set $sk'[i] := v$.
- Return sk'.

Notice that B specifies a single subversion algorithm, and runs in time roughly equal to q times the time to evaluate F. Consider the following game.

Game $\mathbf{G}_0(\kappa, q, \tau, \mathsf{M})$:

- Sample $(vk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$ and $s \leftarrow\!\!\$\ \{0,1\}^\kappa$.
- For q times:
  - Initialize $j := 0$;
  - Repeat the following instructions, until $(sk[i] = v \vee j = \tau)$:
    - ∗ Increment the counter $j := j + 1$;
    - ∗ Sample $r \leftarrow\!\!\$\ \mathcal{R}$ and $m \leftarrow\!\!\$\ \mathsf{M}$;
    - ∗ Compute a fresh signature $\sigma = \mathsf{Sign}(sk, m; r)$;
    - ∗ Evaluate $(v, i) = F_s(\sigma)$.

Define the event $E := E' \vee E''$, where both events E' and E'' are defined in the probability space of $\mathbf{G}_0$ and are specified as follows:

- Event E': The event becomes true whenever, for at least one of the q repetitions, the index j reaches the value $\tau + 1$ (i.e., for no pair $(v, i) \in \{0,1\} \times [\ell]$ we have that $sk[i] = v$).

- Event E'': The event becomes true whenever, at the end of the q repetitions, the values v do not cover the entire set $[\ell]$ (i.e., there exists an index $\hat{\imath} \in [\ell]$ such that for all pairs $(v, i) \in \{0,1\} \times [\ell]$ we have $i \neq \hat{\imath}$).

Observe that the distribution of the pairs (v, i) in game $\mathbf{G}_0$ is identical to that induced during a run of the game from Definition 9. Moreover, whenever the event E does not happen, adversary B recovers the signing key with probability one,[15] and thus $\mathbb{P}[\mathsf{B}\text{ wins}] \geq 1 - \mathbb{P}[E]$. It remains to upper bound the probability of event E.

Towards this goal, consider a mental experiment in which instead of sampling the pair (v, i) by running the PRF on input the signature σ, we now pick $(v, i) \leftarrow\!\!\$\ \{0,1\} \times [\ell]$ uniformly at random, unless the value σ was already generated as part of a previous query, in which case we return the previously sampled pair (v, i). More formally:

Game $\mathbf{G}_1(\kappa, q, \tau, \mathsf{M})$:

[15] This is because in each of the q repetitions we always have that $sk[i] = v$ for some pair (v, i), and furthermore the latter happens for all $i \in [\ell]$.

- Sample $(vk, sk) \leftarrow_{\$} \mathsf{KGen}(1^\kappa)$ and initialize an empty array $\mathcal{S}$.
- For q times:
  - Initialize $j := 0$;
  - Repeat the following instructions, until $(sk[i] = v \vee j = \tau)$:
    - Increment the counter $j := j + 1$;
    - Sample $r \leftarrow_{\$} \mathcal{R}$ and $m \leftarrow_{\$} \mathsf{M}$;
    - Compute a fresh signature $\sigma = \mathsf{Sign}(sk, m; r)$;
    - If $\mathcal{S}[\sigma]$ is undefined, sample $(v, i) \leftarrow_{\$} \{0,1\} \times [\ell]$ and let $\mathcal{S}[\sigma] := (v, i)$.

Claim 1.1. *For all message samplers* M*, we have that* $|\mathbb{P}_{\mathbf{G}_0}[E] - \mathbb{P}_{\mathbf{G}_1}[E]| \leq \varepsilon_{\mathsf{prf}}$.

Proof. Fix an arbitrary message sampler algorithm M with sampling running time t_{M}. Consider the following distinguisher $\mathsf{D}_{\mathsf{prf}}$ attempting to break the PRF security of the function F.

Distinguisher $\mathsf{D}_{\mathsf{prf}}$:

- Run $(vk, sk) \leftarrow_{\$} \mathsf{KGen}(1^\kappa)$.
- For q times:
  - Initialize $j := 0$;
  - Repeat the following instructions, until $(sk[i] = v \vee j = \tau)$:
    - Increment the counter $j := j + 1$;
    - Sample $r \leftarrow_{\$} \mathcal{R}$ and $m \leftarrow_{\$} \mathsf{M}$;
    - Compute a fresh signature $\sigma = \mathsf{Sign}(sk, m; r)$;
    - Query σ to the target oracle (i.e., either the PRF or a random function) and obtain a pair (v, i).
- Use all the collected pairs to check whether the event E has happened or not. In case the event happens output 1, and otherwise output 0.

For the analysis, note that $\mathsf{D}_{\mathsf{prf}}$ asks at most $\tau \cdot q \leq q_{\mathsf{prf}}$ oracle queries and runs in time at most equal to $\tau \cdot q$ times the sum of the running time of algorithm Sign plus the time required to sample an element from M. Moreover, in case $\mathsf{D}_{\mathsf{prf}}$'s target oracle is $F_s(\cdot)$, for a randomly chosen key, the probability that $\mathsf{D}_{\mathsf{prf}}$ outputs 1 is identical to the probability that event E happens in $\mathbf{G}_0$. Similarly, in case $\mathsf{D}_{\mathsf{prf}}$'s target oracle is a random function $f(\cdot)$, the probability that $\mathsf{D}_{\mathsf{prf}}$ outputs 1 is identical to the probability that event E happens in $\mathbf{G}_1$. Hence, by security of the PRF:

$$\left| \mathop{\mathbb{P}}_{\mathbf{G}_0}[E] - \mathop{\mathbb{P}}_{\mathbf{G}_1}[E] \right| = \left| \mathop{\mathbb{P}}_{s \leftarrow_{\$} \{0,1\}^\kappa} \left[\mathsf{D}_{\mathsf{prf}}^{F_s(\cdot)}(1^\kappa) = 1 \right] - \mathop{\mathbb{P}}_{f \leftarrow_{\$} \mathcal{F}} \left[\mathsf{D}_{\mathsf{prf}}^{f(\cdot)}(1^\kappa) = 1 \right] \right| \leq \varepsilon_{\mathsf{prf}},$$

concluding the proof. □

Next, we modify the previous game by requiring that a different pair (v, i) is sampled uniformly at random in each of the queries (independently of collisions on signatures). More formally:

Game $\mathbf{G}_2(\kappa, q, \tau, \mathsf{M})$:

- Sample $(vk, sk) \leftarrow_{\$} \mathsf{KGen}(1^\kappa)$.
- For q times:

```
– Initialize $j := 0$;
– Repeat the following instructions, until $(sk[i] = v \lor j = \tau)$:
    ∗ Increment the counter $j := j + 1$;
    ∗ Sample $r \leftarrow_{\$} \mathcal{R}$ and $m \leftarrow_{\$} \mathsf{M}$;
    ∗ Compute a fresh signature $\sigma = \mathsf{Sign}(sk, m; r)$;
    ∗ Sample $(v, i) \leftarrow_{\$} \{0, 1\} \times [\ell]$.
```

Claim 1.2. *For all message samplers* M*, and for all* $q \in \mathbb{N}$*, we have that* $|\mathbb{P}_{\mathbf{G}_1}[E] - \mathbb{P}_{\mathbf{G}_2}[E]| \leq q^2\tau^2 \cdot 2^{-\eta-1}$.

Proof. Let W be the event that for some of the messages sampled via the message sampler algorithm M there is a collision in the computation of a signature. Clearly, if W does not happen, the values (v, i) are sampled uniformly and independently for each query, and thus $\mathbf{G}_1$ and $\mathbf{G}_2$ are identical. Since $\mathbb{P}_{\mathbf{G}_1}[W] = \mathbb{P}_{\mathbf{G}_2}[W] = \mathbb{P}[W]$, by Lemma 1, $\mathbb{SD}(\mathbf{G}_1; \mathbf{G}_2) \leq \mathbb{P}[W]$.

It remains to bound the probability of event W. By definition of min-entropy, the worst-case probability (over the choice of the randomness) that a particular value σ is hit is at most $2^{-\eta}$ (regardless of the distribution by which we sample the message). Since at most τq signatures are generated, a standard union bound gives

$$\mathbb{P}[W] \leq \binom{\tau q}{2} \cdot 2^{-\eta} \leq \frac{q^2\tau^2}{2} \cdot 2^{-\eta},$$

as desired. □

Finally, we upper bound the probabilty of event E in game $\mathbf{G}_2$.

Claim 1.3. *For all message samplers* M*, and for all* $q \in \mathbb{N}$*, we have that* $\mathbb{P}_{\mathbf{G}_2}[E] \leq \ell \cdot e^{-q/\ell} + q \cdot 2^{-\tau}$.

Proof. We simply analyze the probability of event E happening in $\mathbf{G}_2$, by looking at the subevents E', E''.

- Event E': Since in each of the τ trials, the value v hits $sk[i]$ with probability $1/2$, and since such a process is repeated at most q times, we have that $\mathbb{P}_{\mathbf{G}_2}[E'_2] \leq q \cdot 2^{-\tau}$.
- Event E'': Since each index $i \in [\ell]$ is hit with probability $1/\ell$, the probability that a particular index is not hit after q trials is at most $(1 - \frac{1}{\ell})^q \leq e^{-q/\ell}$. Hence, by applying the union bound, we have that $\mathbb{P}_{\mathbf{G}_2}[E''] \leq \ell \cdot e^{-q/\ell}$.

The claim follows by a union bound and by the definition of event $E := E' \lor E''$. □

The above claims imply $\mathbb{P}[\mathsf{B} \text{ wins}] \geq 1 - \varepsilon_{\mathsf{prf}} - \frac{q^2\tau^2}{2} \cdot 2^{-\eta} - (\ell \cdot e^{-q/\ell} + q \cdot 2^{-\tau})$, concluding the proof of part (i).

(ii) Let $\mathbf{G}$ be the game described in Definition 8. Consider the game $\mathbf{G}_0$ to be an identical copy of $\mathbf{G}$ when $b = 0$, and consider the game $\mathbf{G}_1$ to be an identical copy of $\mathbf{G}$ when $b = 1$. We need to prove that $\mathbf{G}_0$ and $\mathbf{G}_1$ are computationally indistinguishable. To this end, we consider a sequence of games defined below.

Game $\mathbf{H}_0$: Identical to $\mathbf{G}_1$.

Game $\mathbf{H}_1$: We change the way the subversion algorithm $\widetilde{\mathsf{A}}_{s,\tau}$ works. Namely, instead of computing the pair (v, i) by running the PRF on input the signature σ, we now pick $(v, i) \leftarrow_{\$} \{0, 1\} \times [\ell]$ uniformly at random unless the value σ was already generated as part of a previous query, in which case we return the previously sampled pair (v, i).

Game $\mathbf{H}_2$: Identical to the previous game, but now a different pair (v, i) is sampled uniformly at random in each of the queries (independently of collisions on signatures).

Notice that in game $\mathbf{H}_2$ the choice of (v, i), for each signature query, is independent of the choice of σ, and thus it does not affect the distribution of the latter; hence, $\mathbf{H}_2 \equiv \mathbf{G}_0$. The next two claims, whose proof is analogous to the proof of Claim 1.1 and Claim 1.2, imply the statement.

Claim 1.4. *For all distinguishers* D *running in time* t *and asking* q *oracle queries, we have that* $|\mathbb{P}[\mathsf{D}(\mathbf{H}_0) = 1] - \mathbb{P}[\mathsf{D}(\mathbf{H}_1) = 1]| \leq \varepsilon_{\mathsf{prf}}$.

Claim 1.5. *For all (even unbounded) distinguishers* D *asking* q *oracle queries, we have that* $|\mathbb{P}[\mathsf{D}(\mathbf{H}_1) = 1] - \mathbb{P}[\mathsf{D}(\mathbf{H}_2) = 1]| \leq q^2\tau^2 \cdot 2^{-\eta-1}$.

The fact that the class $\mathcal{A}_{\mathsf{bias}}$ is *strongly* undetectable follows directly by the fact that each algorithm $\widetilde{\mathsf{A}}_{s,\tau}(sk, m)$, as described in Fig. 1, is stateless. □

Concrete parameters. Observe that the size of the signing key determines the efficiency of the attack. If we consider the case of RSA-PSS signatures [BR96], with a signing key of size 1536 bits, we can achieve a key-recovery probability of ≈ 0.73 by setting $q = 1536 \cdot 10 = 15360$ and $\tau = 18$ as long as $\eta \geq 38$.

As a comparison, if we consider the case of ECDSA signatures, with signing key of size 256 bits, we can achieve a key-recovery probability of ≈ 0.85 if we take the parameters to be $q = 256 \cdot 8 = 2048$ and $\tau = 15$ as long as $\eta \geq 40$.

4.2 Coin-Extractable Attack

The attack in 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 12 (Coin-extractable signatures)**.** 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 $\kappa \in \mathbb{N}$, for all (vk, sk) output by $\mathsf{KGen}(1^\kappa)$, and all $m \in \mathcal{M}$,

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

We point out that many existing signature schemes are ν_{ext}-coin-extractable, for small ν_{ext}:

- All *public-coin* signature schemes [Sch12], where the random coins used to generate a signature are included as part of the signature. Concretely, the schemes in [GHR99, CS00, NPS01, CL02, Fis03, CL04, BB08, HW09a, HW09b, HK12], and the Unstructured Rabin-Williams scheme [Ber08], 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 (modeled as a random oracle), and r is random bit. Given a pair (m, σ) the extractor simply sets $r = 1$ iff $f(\sigma) = H(m||1)$.

- The PSS signature scheme [BR96, Cor02].

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

Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with randomness space $\mathcal{R} := \{0,1\}^d$, and let $G : \{0,1\}^\kappa \to \{0,1\}^\kappa \times \{0,1\}^d$ be a stateful pseudorandom generator. For simplicity, assume that $d|\ell$, where ℓ is the size of the signing key in bits (a generalization is straightforward). The class $\mathcal{A}_{\mathsf{cext}}$ consists of a set of algorithms $\{\widetilde{\mathsf{A}}_{s,\tau=0}\}_{s\in\{0,1\}^\kappa}$, where each algorithm in the class behaves as follows.

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

- If $\tau \geq \ell$, then $\tau := 0$;
- Run $(s', v) = G(s)$ and re-define $s := s'$;
- Compute the biased randomness $\widetilde{r} := v \oplus sk[\tau+1, \tau+d]$;
- Update the state $\tau \leftarrow \tau + d$;
- Return $\sigma := \mathsf{Sign}(sk, m; \widetilde{r})$.

Figure 2: Attacking coin-extractable schemes

In Fig. 2 we describe a secretly undetectable attack that allows to break all coin-extractable schemes (even if the scheme relies on a single bit of randomness). The intuition behind it is as follows.[16] Let G be a *stateful* PRG (cf. Section 2.3) with $d \geq 1$ bits of stretch, and assume that the randomness space of the signature scheme consists of all d-bit strings. We consider the class of SAs which outputs a real signature, except that the randomness used to compute the signature is derived by first running the PRG, and later xor-ing the obtained pseudorandom bits with the next d bits of the secret key. Since the scheme is coin-extractable, an adversary can recover the coins used to generate each signature and remove the pseudorandom pad by re-iterating the PRG, which allows to fully recover the secret key.

Theorem 2. *Let $\ell, d \geq 1$, with $d|\ell$, and let $\mathcal{SS}$ be a randomized, ν_{ext}-coin-extractable, signature scheme, with randomness space $\mathcal{R} = \{0,1\}^d$ and signing key space $\mathcal{SK} = \{0,1\}^\ell$. Consider the class of SAs $\mathcal{A}_{\mathsf{cext}}$ described in Fig. 2, and let $G : \{0,1\}^\kappa \to \{0,1\}^\kappa \times \{0,1\}^d$ be a stateful $(t_{\mathsf{prg}}, q_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-secure PRG, $\mathbf{U}_{\mathsf{cext}}$ be the uniform distribution over $\mathcal{A}_{\mathsf{cext}}$, and M be an arbitrary message sampler algorithm. Then:*

(i) There exists an adversary B (and we describe it in the proof of this theorem) that $(t, \ell/d, \varepsilon, \mathbf{U}_{\mathsf{cext}}, \mathsf{M})$-recovers the signing key of $\mathcal{SS}$, with

$$\varepsilon \geq 1 - \nu_{ext} \cdot \ell/d,$$

The running time t of B is roughly equal to ℓ/d times the sum of the running time of algorithm CExt plus the time to evaluate a single iteration of the function G.

(ii) The class $\mathcal{A}_{\mathsf{cext}}$ is secretly $(t_{\mathsf{prg}}, q_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}}, \mathbf{U}_{\mathsf{cext}})$-undetectable, as long as $q_{\mathsf{prg}} \geq \ell/d$ and t_{prg} is at least ℓ/d times the running time of algorithm Sign.

Proof. (i) Consider the following adversary B playing the game $\mathbf{G}$ described in Definition 9, for the SA class $\mathcal{A}_{\mathsf{cext}}$ described in Fig. 2, and let $q := \ell/d$.

$\underline{\text{Adversary } \mathsf{B}\text{:}}$

[16] A previous version of this paper [AMV15] contained a similar attack that even achieves secret undetectability unconditionally; however that attack does not immediately work in the multi-user setting.

```
• Initialize $sk' := 0^\ell$.
• The challenger chooses a subversion algorithm $\widetilde{\mathsf{A}}_{s,\tau}$ uniformly at random from
  the SA class $\mathcal{A}_{\mathsf{cext}}$. (This is equivalent to sampling the algorithm from the
  distribution $\mathbf{U}_{\mathsf{cext}}$, and implicitly corresponds to picking an initial random seed
  $s_0 \in \{0,1\}^\kappa$ for the PRG.)
• For $j \in [q]$ do:
    – Make a query to the signature oracle, receiving a pair $(m_j, \sigma_j)$ such that
      $\sigma_j \leftarrow \widetilde{\mathsf{A}}_{s,\tau}(sk, m_j)$ for $m_j \leftarrow \mathsf{M}$;
    – Run $(s_j, v_j) = G(s_{j-1})$;
    – Extract the randomness from the $j$-th signature $\widetilde{r}_j \leftarrow \mathsf{CExt}(vk, m_j, \sigma_j)$.
    – Set $sk'[(j-1)d+1, jd] := \widetilde{r}_j \oplus v_j$.
• Return $sk'$.
```

Notice that B specifies a single subversion algorithm, asks exactly $q := \ell/d$ oracle queries, and runs in time roughly equal to q times the sum of the running time of algorithm CExt plus the time to evaluate a single iteration of the function G. The claim below concludes the proof.

Claim 2.1. *The following holds for the adversary* B *defined above:* $\mathbb{P}[\mathsf{B} \text{ wins } \mathbf{G}] \geq 1 - q \cdot \nu_{ext}$.

Proof. We note that adversary B fails to extract the biased randomness bits $\widetilde{r}_j \in \{0,1\}^d$ for each σ_j with negligible probability, more precisely with probability at most ν_{ext}. By the union bound, we have that the probability of B failing to recover all ℓ bits of the signing key sk is at most $q \cdot \nu_{ext}$, and thus B's advantage is lower bounded by $\varepsilon \geq 1 - q \cdot \nu_{ext}$. □

(ii) 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. 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 $b = 1$. We need to show that $\mathbf{G}_0$ and $\mathbf{G}_1$ are computationally indistinguishable.

Define the hybrid game $\mathbf{H}$ that is identical to $\mathbf{G}_0$, except that the subversion algorithm $\widetilde{\mathsf{A}}_{s,\tau}$ now uses, at each run, a random $v \leftarrow\!\!\$\, \mathbf{U}_d$, where $\mathbf{U}_d$ is the uniform distribution over d-bit strings. The two claims below conclude the proof.

Claim 2.2. *For all distinguishers* D *running in time* t_{prg} *and asking* q_{prg} *oracle queries, we have that* $|\mathbb{P}[\mathsf{D}(\mathbf{G}_0) = 1] - \mathbb{P}[\mathsf{D}(\mathbf{H}) = 1]| \leq \varepsilon_{\mathsf{prg}}$.

Proof. Let D be an algorithm that distinguishes $\mathbf{G}_0$ and $\mathbf{H}$ with probability at least $\varepsilon_{\mathsf{prg}}$, running in time t and asking q oracle queries. We build a distinguisher $\mathsf{D}_{\mathsf{prg}}$ that, by using distinguisher D, breaks the PRG security of the function G. A description of $\mathsf{D}_{\mathsf{prg}}$ follows.

```
Distinguisher $\mathsf{D}_{\mathsf{prg}}(v_1, \ldots, v_{q_{\mathsf{prg}}})$:

• Initialize $\tau := 0$.
• For each query $(sk_i \in \mathcal{SK}, m_i \in \mathcal{M})$ asked by $\mathsf{D}$:
    – If $\tau \geq \ell$, then $\tau := 0$;
    – Let $r_i := v_i \oplus sk_i[\tau + 1, \tau + d]$;
    – Compute a fresh signature $\sigma = \mathsf{Sign}(sk_i, m_i; r_i)$;
    – Update $\tau \leftarrow \tau + d$;
    – Return $\sigma$ to $\mathsf{D}$.
• Output whatever $\mathsf{D}$ outputs.
```

For the analysis, note that $\mathsf{D}_{\mathsf{prg}}$ needs $q = \ell/d \leq q_{\mathsf{prg}}$ samples, and runs in time t_{prg} roughly equal to q times the running time of algorithm Sign. Moreover, the simulation done by $\mathsf{D}_{\mathsf{prg}}$ is perfect, in the sense that the view of D when run as a sub-routine by $\mathsf{D}_{\mathsf{prg}}$ is either identical to that of $\mathbf{G}_0$ (in case the values v_i are computed using the PRG), or to that of game $\mathbf{H}$ (in case the values v_i are uniformly random). Hence, $\mathsf{D}_{\mathsf{prg}}$ retains the same advantage of D, concluding the proof. □

Claim 2.3. $\mathbf{H} \equiv \mathbf{G}_1$.

Proof. The only difference between $\mathbf{H}$ and $\mathbf{G}_1$ is that, upon input a query $sk_j \in \mathcal{SK}$ from the distinguisher, both games sample $m_j \leftarrow \mathsf{M}$ and return $\sigma_j = \mathsf{Sign}(sk_j, m_j; r_j)$ where the randomness r_j is either sampled uniformly at random from $\{0,1\}^d$ (in $\mathbf{G}_1$) or computed as $v_j \oplus sk_j[(j-1)d+1, jd]$ for a random $v_j \in \{0,1\}^d$. Hence, the two games retain the same distribution. □

□

On removing the state. While the above attack works even against signature schemes using a very little amount of randomness, in fact as small as a single bit, it has the drawback that it requires to keep and update a state within each invocation. As observed by [BJK15], this might lead to detection in some settings, e.g., because of a state reset due to reboot or cloning to create a virtual machine.

We leave it as an open problem to design a *stateless* (secretly) undetectable attack that recovers the signing key for signature schemes with arbitrary min-entropy, and that works for *arbitrary* message samplers.

5 Security of Unique Signatures

In this section we prove that signature schemes with unique signatures are subversion-resilient against SAs that meet a basic undetectability requirement, which we call the verifiability condition.

5.1 The Verifiability Condition

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 $\widetilde{\mathsf{A}}$ upon input an honestly generated signing key sk, (almost) always verify under the corresponding verification key vk (for any value of the algorithm's internal state).

Definition 13 (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 $\kappa \in \mathbb{N}$, for all (vk, sk) output by $\mathsf{KGen}(1^\kappa)$, for all states $\tau \in \{0,1\}^*$, for all algorithms $\widetilde{\mathsf{A}}_\tau \in \mathcal{A}$, and all $m \in \mathcal{M}$,

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \widetilde{\mathsf{A}}_\tau(sk, m))) = 1\right] \geq 1 - \nu_v,$$

where the probability is taken over the randomness of algorithm $\widetilde{\mathsf{A}}_\tau$.

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 13 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.[17] Clearly, whenever the message space is large enough, $\mathcal{A}_{\mathsf{msg}}$ satisfies public undetectability w.r.t. the uniform distribution over $\mathcal{A}_{\mathsf{msg}}$, 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, let $\mathcal{SS}$ be any UF-CMA signature scheme with large randomness space (say, $\mathcal{R} = \{0,1\}^\kappa$). Consider the stateful 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 $\mathcal{SS}$; yet, $\mathcal{A}_{\mathsf{det}}$ does not satisfy public undetectability (for any distribution over $\mathcal{A}_{\mathsf{det}}$), as a user can simply query the same message twice in order to guess the value of the hidden bit b with overwhelming probability.

Relaxed verifiability. The assumption that the verifiability condition should hold for all values $m \in \mathcal{M}$ is quite a strong one. A natural relaxation is to require that the probability in Definition 13 is taken also over the choice of the message.

Definition 14 (Relaxed verifiability)**.** Let $\mathcal{A}$ be some class of SAs for a signature scheme $\mathcal{SS}$. We say that $\mathcal{A}$ satisfies *relaxed* ν_v-verifiability if for all $\kappa \in \mathbb{N}$, for all (vk, sk) output by $\mathsf{KGen}(1^\kappa)$, for all states $\tau \in \{0,1\}^*$, and for all algorithms $\widetilde{\mathsf{A}}_\tau \in \mathcal{A}$,

$$\mathbb{P}\left[\mathsf{Vrfy}(vk, (m, \widetilde{\mathsf{A}}_\tau(sk, m))) = 1 : \; m \leftarrow\!\!\$\, \mathcal{M}\right] \geq 1 - \nu_v,$$

where the probability is taken over the (uniform) choice of the message and over the randomness of algorithm $\widetilde{\mathsf{A}}_\tau$.

Unfortunately, public undetectability does *not* imply even *relaxed* verifiability in general. This is because relaxed verifiability still has to hold for all algorithms $\widetilde{\mathsf{A}} \in \mathcal{A}$, while public undetectability only holds w.r.t. some (efficiently samplable) distribution over $\mathcal{A}$. (See also the discussion after Definition 8.)

5.2 Chosen-Message Attacks

The theorem below shows that unique signature schemes (cf. Definition 3) achieve indistinguishability (and thus EUF-CMA) against the class of all SAs that meet the verifiability condition (cf. Definition 13).

Theorem 3. *Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with perfect correctness and uniqueness, and denote by $\mathcal{A}_{\mathsf{ver}}^{\nu_v}$ 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 $\mathcal{A}_{\mathsf{ver}}^{\nu_v}$-SAs, for all $t, n, q \in \mathbb{N}$ and for all $\varepsilon \leq qn \cdot \nu_v$.*

Proof. Let $\mathbf{G}$ be the game described in Definition 6. 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 $\mathbf{G}_0 \approx \mathbf{G}_1$.

[17] A similar class of attacks—under the name of input-triggered subversion—has been considered in [DFP15] for the case of symmetric encryption.

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 $\widetilde{\mathsf{A}}_j(sk, m_{i,j})$),[18] while all queries $(j, m_{i,j})$ such that $j > k$ are answered as in $\mathbf{G}_1$ (i.e., by running $\mathsf{Sign}(sk, m_{i,j})$). We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_n \equiv \mathbf{G}_0$. Abusing notation, let us write $\mathbf{H}_k$ for the distribution of the random variable corresponding to B's view in game $\mathbf{H}_k$.

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 $(k, m_{i,k})$ with $i \leq l$ are treated as in game $\mathbf{G}_0$, while queries $(k, m_{i,k})$ 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 3.1. *Fix some $k \in [0,n]$. For each $l \in [0,q]$, we have $\mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \leq \nu_v$.*

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 $\widetilde{\sigma}_{l,k} \leftarrow\!\!\$\ \widetilde{\mathsf{A}}_k(sk, m_{l,k})$, whereas game $\mathbf{H}_{k,l}$ returns $\sigma_{l,k} \leftarrow\!\!\$\ \mathsf{Sign}(sk, m_{l,k})$. Now let $E_{l,k}$ be the event that $\sigma_{l,k} \neq \widetilde{\sigma}_{l,k}$; observe that w.l.o.g. we may assume that each experiment $\mathbf{H}_{k,l}$ computes both sequences of signatures $(\sigma_{1,k}, \ldots, \sigma_{q,k})$ and $(\widetilde{\sigma}_{1,k}, \ldots, \widetilde{\sigma}_{q,k})$, so that the event $E_{l,k}$ is well defined on both $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$. We can write

$$\begin{aligned} \mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) &\leq \mathbb{SD}(\mathbf{H}_{k,l-1} | \neg E_{l,k}; \mathbf{H}_{k,l} | \neg E_{l,k}) + \mathbb{P}[E_{l,k}] && (2) \\ &\leq \nu_v. && (3) \end{aligned}$$

Eq. (2) follows by Lemma 1, since $\mathbb{P}_{\mathbf{H}_{k,l-1}}[E_{l,k}] = \mathbb{P}_{\mathbf{H}_{k,l}}[E_{l,k}] = \mathbb{P}[E_{l,k}]$. Eq. (3) 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 $\mathbb{P}[E_{l,k}] \leq \nu_v$. 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. 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 (for every possible value of the implicit state information $\tau_k \in \{0,1\}^*$). Finally, by the uniqueness property of $\mathcal{SS}$ we have that $\sigma_{l,k}$ and $\widetilde{\sigma}_{l,k}$ must be equal. It follows that $\mathbb{P}[E_{l,k}] \leq \nu_v$, as desired. □

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

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

□

Unfortunately, unique signatures do not satisfy EUF-CMA against the class of all SAs that satisfy *relaxed* verifiability (cf. Definition 14). In fact, it is not hard to show that no signature scheme with large enough message space (no matter if randomized or deterministic) can achieve EUF-CMA against such class of SAs. This can be seen by looking again at 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 the secret key. Clearly, $\mathcal{A}_{\mathsf{msg}}$ satisfies relaxed verifiability, as a randomly chosen message will be different from $\bar{m}$ with high probability; yet, $\mathcal{A}_{\mathsf{msg}}$ allows to break EUF-CMA for an adversary knowing $\bar{m}$.

[18] Recall that each subversion can be stateful, and thus algorithm $\widetilde{\mathsf{A}}_j$ additionally takes as input state information $\tau_j \in \{0,1\}^*$ that gets updated at each invocation. To simplify the notation we omit to write the dependency on τ_j explicitly.

5.3 Random-Message Attacks

We show that if we restrict to the case of random-message attacks (RMA), i.e. the adversary can only see signatures of randomly chosen messages, unique signatures achieve unforgeability against the class of SAs that meets relaxed verifiability (cf. Definition 14).

Definition 15 (EUF-RMA 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)-EUF-RMA w.r.t. *continuous* $\mathcal{A}$-SAs if for all adversaries B running in time t, we have $\mathbb{P}[\mathsf{B} \text{ wins}] \leq \varepsilon(\kappa)$ in the game of Definition 7, with the adaptation that the messages in the sets $\mathcal{Q}, \widetilde{\mathcal{Q}}_1, \ldots, \widetilde{\mathcal{Q}}_n$ are drawn uniformly at random from the message space $\mathcal{M}$.

While the above definition might seem a weak guarantee, it is still useful for applications. In particular, in Section 5.4, we show how to use any signature scheme that is EUF-RMA against a given class of SAs, to construct an identification scheme that is subversion-resilient against the same class of SAs.

Theorem 4. *Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a $(t, (q+1) \cdot n, \varepsilon)$-EUF-RMA signature scheme with perfect correctness and uniqueness, and denote by $\mathcal{A}^{\nu_v}_{\mathsf{r\_ver}}$ the class of all algorithms that satisfy* relaxed *ν_v-verifiability relative to $\mathcal{SS}$. Then, $\mathcal{SS}$ is (t', n, q, ε')-EUF-RMA against continuous $\mathcal{A}^{\nu_v}_{\mathsf{r\_ver}}$-SAs, for $t' \approx t$, for all $n, q \in \mathbb{N}$, and for $\varepsilon' \leq \varepsilon + qn \cdot \nu_v$.*

Proof. Let $\mathbf{G}$ be the game of Definition 15. Consider the modified game $\mathbf{H}$ that is identical to $\mathbf{G}$ except that queries to the subverted signing algorithms are answered as described below:

- For all $i \in [q]$, $j \in [n]$, sample $m_{i,j} \leftarrow\!\!\$\, \mathcal{M}$ and return $\sigma_{i,j} \leftarrow \mathsf{Sign}(sk, m_{i,j})$.

Claim 4.1. *$|\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{G}] - \mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}]| \leq qn \cdot \nu_v$.*

Proof. For an index $k \in [0, n]$, consider the hybrid game $\mathbf{H}_k$ that answers each query to the j-th subversion oracle, such that $j \leq k$, as in game $\mathbf{G}$, while all queries with $j > k$ 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 $\mathbb{SD}(\mathbf{H}_{k-1}, \mathbf{H}_k) \leq q \cdot \nu_v$ 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, i) with $i \leq l$ as in game $\mathbf{G}$, while all queries (k, i) 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 $\mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \leq \nu_v$. 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, l): Game $\mathbf{H}_{k,l-1}$ returns $\sigma_{l,k} \leftarrow\!\!\$\, \mathsf{Sign}(sk, m_{l,k})$, whereas game $\mathbf{H}_{k,l}$ returns $\widetilde{\sigma}_{l,k} \leftarrow\!\!\$\, \widetilde{\mathsf{A}}_k(sk, m_{l,k})$ (where $m_{l,k} \leftarrow\!\!\$\, \mathcal{M}$).[19] Now let $E_{l,k}$ be the event that $\sigma_{l,k} \neq \widetilde{\sigma}_{l,k}$; observe that w.l.o.g. we may assume that each experiment $\mathbf{H}_{k,l}$ computes both sequences of signatures $(\sigma_{1,k}, \ldots, \sigma_{q,k})$ and $(\widetilde{\sigma}_{1,k}, \ldots, \widetilde{\sigma}_{q,k})$, so that the event $E_{l,k}$ is well defined on both $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$. We can write

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

Eq. (4) follows by Lemma 1, since $\mathbb{P}_{\mathbf{H}_{k,l-1}}[E_{l,k}] = \mathbb{P}_{\mathbf{H}_{k,l}}[E_{l,k}] = \mathbb{P}[E_{l,k}]$. 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 $\mathbb{P}[E_{l,k}] \leq \nu_v$. 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. By the assumption that $\widetilde{\mathsf{A}}_k \in \mathcal{A}^{\nu_v}_{\mathsf{r\_ver}}$ we have

[19] Recall that each algorithm $\widetilde{\mathsf{A}}_k$ is stateful, but for simplicity we omit to write the state information explicitly.

that $\widetilde{\sigma}_{l,k}$ is also valid for $m_{l,k}$ under vk except with probability at most ν_v (this is because $m_{l,k}$ is chosen at random, and moreover the verifiability condition holds for every possible value of the implicit state information $\tau_k \in \{0,1\}^*$). Finally, by the uniqueness property of $\mathcal{SS}$ we have that $\sigma_{l,k}$ and $\widetilde{\sigma}_{l,k}$ must be equal. It follows that $\mathbb{P}[E_{l,k}] \le \nu_v$, as desired.

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

$$\mathbb{SD}(\mathbf{G}, \mathbf{H}) \le \sum_{k=1}^{n} \mathbb{SD}(\mathbf{H}_{k-1}, \mathbf{H}_k) \le \sum_{k=1}^{n}\sum_{l=1}^{q} \mathbb{SD}(\mathbf{H}_{k,l-1}, \mathbf{H}_{k,l}) \le qn \cdot \nu_v.$$

□

Claim 4.2. $\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}] \le \varepsilon$.

Proof. Towards a contradiction, assume B wins in game **H** with probability larger than ε. The claim follows directly by observing that in game **H** the answers to B's queries have exactly the same distribution as the answer to signature queries in game EUF-RMA, and that moreover B asks a total of $q + qn$ queries (i.e., q queries to the original signing algorithm plus q queries for each of the n subversions). □

The proof follows by combining the above two claims. □

5.4 Subversion-Resilient Identification Schemes

We show how to apply EUF-RMA against SAs to the setting of subversion-resilient identification (ID) schemes. Similar applications already appeared in the literature for leakage and tamper resistance [ADW09, FHN+12, DFMV13, NVZ14, FNV15].

In a public-key ID scheme, a prover with secret key sk attempts to prove its identity to a verifier holding the corresponding verification key vk. More formally, an ID scheme $\mathcal{ID} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{P}, \mathsf{V})$ consists of four polynomial-time algorithms described as follows: (1) The parameters generation algorithm takes as input the security parameter and outputs public parameters $\mathsf{params} \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$, shared by all users.[20] (2) The key generation algorithm takes as input the security parameter and outputs a verification key/secret key pair $(vk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$. (3) P and V are probabilistic Turing machines interacting in a protocol; at the end of the execution V outputs a decision bit $d \in \{0,1\}$, where $d = 1$ means that the identification was successful. We write $\langle \mathsf{P}(sk), \mathsf{V}(vk)\rangle$ for the random variable corresponding to the verifier's verdict, and $\mathsf{P}(sk) \rightleftarrows \mathsf{V}(vk)$ for the random variable corresponding to transcripts of honest protocol executions.

We now define a variant of passive security, where in a first phase the adversary is allowed to subvert the prover algorithm; in a second phase, the adversary has to impersonate the prover. Similarly to the case of signature schemes, subversion is modelled by considering a class $\mathcal{A}$ of SAs, where each $\widetilde{\mathsf{A}} \in \mathcal{A}$ is an algorithm replacing the prover algorithm P within the ID scheme $\mathcal{ID}$.

Definition 16 (Subversion-resilient identification)**.** Let $\mathcal{ID} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be an ID scheme, and $\mathcal{A}$ be some class of SAs for $\mathcal{ID}$. We say that $\mathcal{ID}$ is (t, n, q, ε)-secure w.r.t. *continuous* $\mathcal{A}$-SAs if for all adversaries B running in time t, we have $\mathbb{P}[\mathsf{B} \text{ wins}] \le \varepsilon(\kappa)$ in the following game:

1. The challenger runs $\mathsf{params} \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$, $(vk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$, and forwards (params, vk) to B.

[20] In what follows, all algorithms take as input params, but we omit it here for ease of notation.

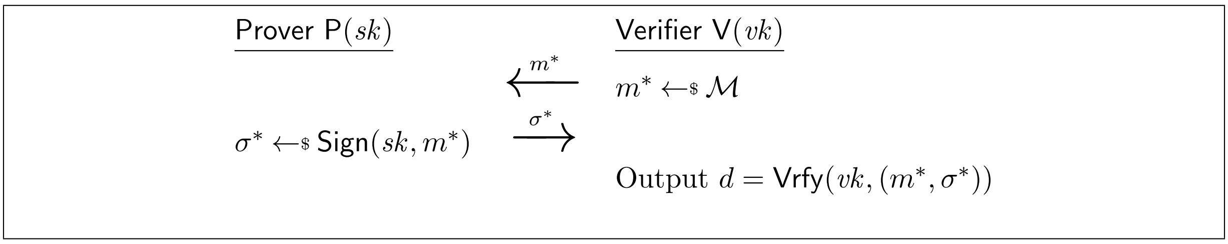
2. The adversary B can observe up to $q \in \mathbb{N}$ transcripts $\mathsf{P}(sk) \rightleftarrows \mathsf{V}(vk)$ corresponding to honest protocol executions between the prover and the verifier.

3. The adversary B can specify an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, for a total of at most $n \in \mathbb{N}$ queries. Each such query implicitly defines an oracle that can be queried up to $q \in \mathbb{N}$ times.
   - Upon an empty input, the j-th oracle outputs a transcript $\widetilde{\mathsf{A}}_j(sk) \rightleftarrows \mathsf{V}(vk)$ corresponding to a protocol execution between the subverted prover and the verifier. In case $\widetilde{\mathsf{A}}_j$ is undefined, the oracle returns $\perp$.
   - Note that B does not need to ask all q queries before choosing the next algorithm, i.e. the queries in step 2 and step 3 can be interleaved in an arbitrary manner.

4. The adversary B loses access to all oracles and plays the role of the prover in an execution with an honest verifier, obtaining $d \xleftarrow{\$} \langle \mathsf{B}(vk), \mathsf{V}(vk)\rangle$; we say that B wins if and only if $d = 1$.

Consider the following standard construction (see, e.g., [BFGM01]) of an identification scheme $\mathcal{ID}$ from a signature scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$.

- *Parameters generation.* Algorithm Setup samples the public parameters params for the signature schemes (if any).
- *Key Generation.* Algorithm KGen runs the key generation algorithm of the signature scheme, obtaining $(vk, sk) \xleftarrow{\$} \mathsf{KGen}(1^\kappa)$.
- *Identification protocol.* The interaction $\mathsf{P}(sk) \rightleftarrows \mathsf{V}(vk)$ is depicted in Figure 3.



Prover $\mathsf{P}(sk)$

Verifier $\mathsf{V}(vk)$

$\xleftarrow{m^*}$ $m^* \xleftarrow{\$} \mathcal{M}$

$\sigma^* \xleftarrow{\$} \mathsf{Sign}(sk, m^*)$ $\xrightarrow{\sigma^*}$

Output $d = \mathsf{Vrfy}(vk, (m^*, \sigma^*))$

Figure 3: Two-round identification using a signature scheme $\mathcal{SS}$ with message space $\mathcal{M}$

The theorem below states that the above protocol achieve subversion resilience w.r.t. a given class $\mathcal{A}$ of SAs, provided that the underlying signature scheme is EUF-RMA w.r.t. the same class $\mathcal{A}$.

Theorem 5. *Let $\mathcal{SS}$ be a signature scheme with message space $\mathcal{M}$, and let $\mathcal{A}$ be a class of SAs for $\mathcal{SS}$. Assume that $\mathcal{SS}$ is (t, n, q, ε)-EUF-RMA w.r.t. continuous $\mathcal{A}$-SAs. Then the ID scheme $\mathcal{ID}$ from Figure 3 is (t', n, q, ε')-secure w.r.t. continuous $\mathcal{A}$-SAs where $t' \approx t$ and $\varepsilon' \leq \varepsilon + \frac{(n+1)q}{|\mathcal{M}|}$.*

Proof. For the sake of contradiction, assume that there exists an adversary B breaking security of the identification scheme with probability larger than $\varepsilon + \frac{(n+1)q}{|\mathcal{M}|}$. We construct an adversary B' breaking EUF-RMA of $\mathcal{SS}$ with probability larger than ε (a contradiction). Adversary B' runs the game of Definition 15 and is described below. The main observation is that the prover's algorithm P is completely specified by algorithm Sign, and thus subverting the ID scheme is equivalent to subverting the signature scheme.

Adversary B':

1. Receive the public parameters params and the verification key vk for $\mathcal{SS}$, and forward (params, vk) to B.
2. Whenever B wants to observe an honest transcript $\mathsf{P}(sk) \rightleftarrows \mathsf{V}(vk)$, query the signing oracle obtaining a pair (m_i, σ_i) such that $\sigma_i \leftarrow\!\!\$\ \mathsf{Sign}(sk, m_i)$ and $m_i \leftarrow\!\!\$\ \mathcal{M}$. Forward (m_i, σ_i) to B.
3. Whenever B specifies an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, forward $\widetilde{\mathsf{A}}_j$ to the challenger. For each query of B to its own j-th oracle, query the target j-th oracle obtaining a pair $(m_{i,j}, \widetilde{\sigma}_{i,j})$ such that $\widetilde{\sigma}_{i,j} \leftarrow\!\!\$\ \widetilde{\mathsf{A}}_j(sk, m_{i,j})$ and $m_{i,j} \leftarrow\!\!\$\ \mathcal{M}$. Forward $(m_{i,j}, \widetilde{\sigma}_{i,j})$ to B.
4. Finally, when B is ready to start the impersonation phase, sample a random message $m^* \leftarrow\!\!\$\ \mathcal{M}$ and send it to B. Upon receiving a value σ^* from B, output (m^*, σ^*) as forgery.

It is easy to see that B''s simulation of B's queries is perfect; moreover, since the message m^* in the impersonation stage is chosen at random from $\mathcal{M}$, also the simulation of this phase has the right distribution, and in particular, the forgery (m^*, σ^*) will be valid with probability ε.

It remains to compute the probability that B' is successful. Observe that B' is successful whenever (m^*, σ^*) is valid and $m^* \not\in \mathcal{Q} \cup \widetilde{\mathcal{Q}}$. Also, note that m^* is independent from $\widetilde{\mathcal{Q}}$, so in particular

$$\mathbb{P}\left[m^* \in \mathcal{Q} \cup \widetilde{\mathcal{Q}}\right] \leq \frac{|\mathcal{Q}| + |\widetilde{\mathcal{Q}}|}{|\mathcal{M}|} = \frac{(n+1)q}{|\mathcal{M}|}.$$

Let E be the event that $m^* \not\in \mathcal{Q} \cup \widetilde{\mathcal{Q}}$. We have,

$$\begin{aligned}\mathbb{P}\left[\mathsf{B}' \text{ wins}\right] \geq \mathbb{P}\left[\mathsf{B} \text{ wins} \wedge E\right] &\geq \mathbb{P}\left[\mathsf{B} \text{ wins}\right] - \mathbb{P}\left[\neg E\right] \\ &\geq \mathbb{P}\left[\mathsf{B} \text{ wins}\right] - \frac{(n+1)q}{|\mathcal{M}|} \\ &> \varepsilon,\end{aligned}$$

where the last inequality follows by our initial assumption on B's advantage. This concludes the proof. □

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 6 and 7) 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). RFs were introduced by Mironov and Stephens-Davidowitz [MS15] as a means of sanitizing the transcript of arbitrary two-party protocols that are run on possibly corrupted machines; below, we specialize their definition to the case of signature schemes. 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 17 (RF for signatures)**.** Let $\mathcal{SS}$ be a signature scheme with verification-key space $\mathcal{VK}$. 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 $(m, \sigma) \in (\{0,1\}^*)^2$, and outputs a possibly modified signature or a special symbol $\perp$ and an updated (public) state δ'. We write this as $(\delta', \sigma') \leftarrow\!\!\$\ \mathsf{Patch}(\delta, (m, \sigma))$.

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/outputs ever processed by the RF.

6.1 Properties

Below, we discuss the correctness and security requirements of a 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 signature σ' should (almost always) be a valid signature for m under vk. More precisely, we say that $\mathcal{FW}$ is *functionality maintaining* for $\mathcal{SS}$ if there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that, for any polynomial $p(\kappa)$ and any vector of inputs $(m_1, \ldots, m_p) \in \mathcal{M}$,

$$\mathbb{P}\left[\exists i \in [p] \text{ s.t. } \mathsf{Vrfy}(vk, (m_i, \sigma'_i)) = 0 : \begin{array}{r} (vk, sk) \leftarrow \mathsf{KGen}(1^\kappa), \delta_0 \leftarrow\!\!\$\ \mathsf{Setup}(vk, 1^\kappa) \\ \sigma_1 \leftarrow \mathsf{Sign}(sk, m_1), \ldots, \sigma_p \leftarrow \mathsf{Sign}(sk, m_p) \\ \forall i \in [p], (\delta_i, \sigma'_i) \leftarrow\!\!\$\ \mathsf{Patch}(\delta_{i-1}, (m_i, \sigma_i)) \end{array}\right] \leq \nu(\kappa),$$

where the probability is taken over the coin tosses of all involved algorithms.

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 18 (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 EUF-CMA 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 $\sigma_i \leftarrow\!\!\$\ \mathsf{Sign}(sk, m_i)$ for a total of at most $q \in \mathbb{N}$ queries.
3. The adversary B can specify an algorithm $\widetilde{\mathsf{A}}_j \in \mathcal{A}$, for a total of at most $n \in \mathbb{N}$ queries. Each such algorithm implicitly defines an oracle that can be queried adaptively up to $q \in \mathbb{N}$ times.
    - Upon input a query of the form $(j, m_{i,j})$, where $j \in [n]$ and $i \in [q]$, the j-th oracle outputs $(\delta', \widetilde{\sigma}_{i,j}) \leftarrow\!\!\$\ \mathsf{Patch}(\delta_{\mathsf{cur}}, (m_{i,j}, \widetilde{\mathsf{A}}_j(sk, m_{i,j})))$, where δ_{cur} is the current state of the firewall, and updates the public state to $\delta_{\mathsf{cur}} := \delta'$. In case $\widetilde{\mathsf{A}}_j$ is undefined, the oracle returns $\perp$.

- Note that B does not need to ask all q queries before choosing the next algorithm, i.e. the queries to each oracle $\widetilde{\mathsf{A}}_j$ and to oracle Sign can be interleaved in an arbitrary manner.

4. Let $\mathcal{Q} = \{m_1, \ldots, m_q\}$ be the set of messages queried to oracle $\mathsf{Sign}(sk, \cdot)$; similarly, for each $j \in [n]$, let $\widetilde{\mathcal{Q}}_j = \{m_{1,j}, \ldots, m_{q,j}\}$ be the set of messages queried to oracle $\mathsf{Patch}(\delta, (\cdot, \widetilde{\mathsf{A}}_j(sk, \cdot)))$.

5. 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 $\widetilde{\mathcal{Q}} := \bigcup_{j=1}^{n} \widetilde{\mathcal{Q}}_j$.

If for all $t, n, q = poly(\kappa)$ there exists $\varepsilon = negl(\kappa)$ such that $\mathcal{FW}$ (t, n, q, ε)-preserves EUF-CMA for $\mathcal{SS}$, we simply say that $\mathcal{FW}$ preserves EUF-CMA for $\mathcal{SS}$. Furthermore, in case B specifies all of its queries $\{\widetilde{\mathsf{A}}_j, m_{i,j}\}_{j \in [n], i \in [q]}$ at the same time we say that $\mathcal{FW}$ *non-adaptively* preserves EUF-CMA.

We observe that Definition 18 is very similar to Definition 7, except 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).

Looking ahead, no RF can preserve unforgeability (even non adaptively) without keeping state. In fact, as we show in Section 6.2, there exist simple generic attacks that allow for complete security breaches in case of a stateless RF. On the positive side, in Section 6.3, we show how to design an unforgeability preserving RF for any *re-randomizable* signature scheme by using a *single* bit of *public* state that is used to implement the so-called self-destruct capability: Whenever the firewall returns $\perp$, all further queries will result in $\perp$. Let us stress, however, that assuming self-destruct does not make the problem of designing an unforgeability preserving reverse firewall trivial: The biased-randomness attacks of Section 4 allow to break all randomized schemes *without* ever provoking a self-destruct.

On exfiltration resistance. More generally, 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 6).

It is not hard to see that exfiltration 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).[21]

6.2 Necessity of Keeping State

We show that no RF can preserve both functionality and unforgeability, without maintaining state. This is achieved via 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.

[21] 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.

Theorem 6. *Let $\mathcal{SS}$ be an EUF-CMA signature scheme with perfect correctness. No* stateless *RF $\mathcal{FW}$ can, at the same time, be functionality maintaining and non-adaptively $(poly(\kappa), 1, poly(\kappa), negl(\kappa))$-preserve EUF-CMA for $\mathcal{SS}$.*

Proof. Let $\mathcal{FW} = (\mathsf{Setup}, \mathsf{Patch})$ be a stateless RF; formally, this means that the state δ is fixed by algorithm Setup and never updated by algorithm Patch. Consider the following adversary B playing the game of Definition 18.

- Upon input the verification key vk, and the initial state δ, initialize $\tau := 1$.
- Let $\widetilde{\mathsf{A}}_\tau$ be the following algorithm. Upon input a message m_i, set $j = \tau \bmod \ell$ (where $\ell := |sk|$) and
  - If $sk[j] = 1$, output $\widetilde{\sigma}_i \leftarrow \mathsf{Sign}(sk, m_i)$.
  - Else, output $0^{|\sigma|}$ (where $|\sigma|$ is the length of the signatures produced by the signing algorithm).

  Update $\tau \leftarrow \tau + 1$.
- Forward $(\widetilde{\mathsf{A}}_\tau, m_1, \ldots, m_\ell)$ to the challenger, where $m_1, \ldots, m_\ell \in \mathcal{M}$. Let $(\widetilde{\sigma}'_1, \ldots, \widetilde{\sigma}'_\ell)$ be the answers computed by the challenger, where $\widetilde{\sigma}'_i \leftarrow \mathsf{Patch}(\delta, (m_i, \widetilde{\mathsf{A}}_\tau(sk, m_i)))$.
- Define $sk'[i] = \mathsf{Vrfy}(vk, (m_i, \widetilde{\sigma}'_i))$ 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 the sequence of messages $m_1, \ldots, m_\ell \in \mathcal{M}$. We will show that the extracted key sk' is equal to the original secret key sk with overwhelming probability, which clearly implies the statement. Define the following events, parametrized by an index $i \in [\ell]$: (i) Event E'_i becomes true if $sk[i] = 1$ and $\mathsf{Vrfy}(vk, (m_i, \widetilde{\sigma}'_i)) = 0$; (ii) Event E''_i becomes true if $sk[i] = 0$ and $\mathsf{Vrfy}(vk, (m_i, \widetilde{\sigma}'_i)) = 1$. Let $E_i := E'_i \vee E''_i$.

Claim 6.1. *There exists a negligible function $\nu' : \mathbb{N} \to [0, 1]$ such that $\mathbb{P}[E'_i] \leq \nu'(\kappa)$, for all $i \in [\ell]$.*

Proof. Follows directly by the fact that $\mathcal{SS}$ satisfies (perfect) correctness, and moreover $\mathcal{FW}$ is stateless and functionality maintaining. □

Claim 6.2. *There exists a negligible function $\nu'' : \mathbb{N} \to [0, 1]$ such that $\mathbb{P}[E''_i] \leq \nu''(\kappa)$, for all $i \in [\ell]$.*

Proof. Intuitively, whenever E''_i happens, the RF forged a signature on message m_i by patching the all-zero string using only public information, which contradicts EUF-CMA security of $\mathcal{SS}$. More formally, assume there exists an index $i \in [\ell]$, a polynomial $p(\cdot)$, and an adversary B provoking event E''_i with probability at least $1/p(\kappa)$ for infinitely many values of $\kappa \in \mathbb{N}$. Consider the following adversary A' attacking EUF-CMA of $\mathcal{SS}$:

Adversary A':

1. Receive the target vk from the challenger.
2. Run $\delta \leftarrow\$\ \mathsf{Setup}(vk, 1^\kappa)$, and forward (vk, δ) to B.
3. Whenever B outputs $(\widetilde{\mathsf{A}}_\tau, m_1, \ldots, m_\ell)$, pick a random $j \leftarrow\$\ [\ell]$, compute $\widetilde{\sigma}'_j \leftarrow\$\ \mathsf{Patch}(\delta, (m_j, 0^{|\sigma|}))$, and return $(m_j, \widetilde{\sigma}'_j)$ as the forgery.

```
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 $\mathsf{Vrfy}(vk, (m, \widetilde{\sigma})) = 1$
                                            $\widetilde{\sigma}' \leftarrow\!\!\$\ \mathsf{ReRand}(vk, m, \widetilde{\sigma})$
                                        Else set $\widetilde{\sigma}' = \bot$ and $\beta = 1$
                                        Forward $(m, \widetilde{\sigma}')$
```

Figure 4: A cryptographic reverse firewall preserving unforgeability of any re-randomizable signature scheme against arbitrary SAs.

By assumption, with probability at least $1/p(\kappa)$, we have that $\mathsf{Vrfy}(vk, (m_i, \widetilde{\sigma}'_i)) = 1$, and moreover A' guesses the index i (i.e., $j = i$) with probability $1/\ell$. Thus $\mathbb{P}[\mathsf{A}' \text{ wins}] \geq 1/\ell \cdot 1/p(\kappa)$, which is non-negligible. This concludes the proof. □

Putting together the above two claims, we conclude that

$$\begin{aligned}\mathbb{P}\left[sk' = sk\right] = 1 - \mathbb{P}\left[sk' \neq sk\right] = 1 - \mathbb{P}\left[\exists i \in [\ell] :\ E'_i \vee E''_i\right] \\ \geq 1 - \sum_{i=1}^{\ell} \mathbb{P}\left[E'_i \vee E''_i\right] \geq 1 - \ell \cdot (\nu'(\kappa) + \nu''(\kappa)) \geq 1 - \nu(\kappa),\end{aligned}$$

which implies the theorem. □

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 19 (Re-randomizable signatures [HJK12])**.** A signature scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is efficiently re-randomizable if there exists a PPT algorithm ReRand such that, for all $\kappa \in \mathbb{N}$, for all (vk, sk) output by $\mathsf{KGen}(1^\kappa)$, for all messages $m \in \mathcal{M}$, and for all strings σ such that $\mathsf{Vrfy}(vk, (m, \sigma)) = 1$, we have that the output distribution of $\mathsf{ReRand}(vk, m, \sigma)$ is identical to the output distribution of $\mathsf{Sign}(sk, m)$.

The original formulation of re-randomizability (as defined in [HJK12]) is slightly different from the one defined above, in that the output distribution of ReRand is required to be identical to the uniform distribution over the set $\Sigma(vk, m)$ of all strings σ that verify for message m under the verification key vk, even if vk was maliciously generated. However, the variant considered above suffices for our purpose. Note that unique signatures are efficiently re-randomizable, for $\mathsf{ReRand}(vk, m, \sigma) = \sigma$; Waters' signature scheme [Wat05], and its variant by Hofheinz *et al.* [HJK12], are also efficiently re-randomizable as per the definition above.

Our firewall, which is formally described in Fig. 4, 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 $\bot$; 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 7. *Let $\mathcal{SS}$ be a $(t,(q+1)n,\varepsilon)$-EUF-CMA signature scheme that is efficiently re-randomizable and that satisfies perfect correctness. Then, the RF of Fig. 4 maintains functionality and (t',q,ε')-preserves EUF-CMA for $\mathcal{SS}$, where $t' \approx t$ and $\varepsilon' \leq qn \cdot \varepsilon$.*

Proof. The fact that the firewall maintains functionality follows directly by correctness of $\mathcal{SS}$. We now proceed to show that the firewall preserves unforgeability. Let $\mathbf{G}$ be the game of Definition 18. Consider the modified game $\mathbf{H}$ that is identical to $\mathbf{G}$, except that tampered signature queries $(j,m_{i,j})$ are answered as follows: If such query is specified before a self-destruct happens in $\mathbf{G}$ (if any), return $\sigma_{i,j} \leftarrow \mathsf{Sign}(sk,m_{i,j})$ and $\delta' = (vk,0)$, else return $\perp$ and $\delta' = (vk,1)$. In what follows, we write $(i^*,j^*) \in [q]\times[n]$ for the pair of indexes in which the firewall self-destructs; notice that such a pair depends on the randomness of the game, but is the same for both $\mathbf{G}$ and $\mathbf{H}$.

Claim 7.1. $\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{G}] = \mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}]$.

Proof. 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 $\mathbf{G}$, while all queries $(j,m_{i,j})$ 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 $\mathbb{SD}(\mathbf{H}_{k-1},\mathbf{H}_k) = 0$ 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 $(k,m_{i,k})$ with $i \leq l$ as in game $\mathbf{G}$, while all queries $(k,m_{i,k})$ 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 $\mathbb{SD}(\mathbf{H}_{k,l-1},\mathbf{H}_{k,l}) = 0$. 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 $(k,m_{l,k})$: $\mathbf{H}_{k,l-1}$ returns $\sigma_{l,k} \leftarrow\!\!\$\ \mathsf{Sign}(sk,m_{l,k})$, whereas $\mathbf{H}_{k,l}$ returns $\widetilde{\sigma}'_{l,k} \leftarrow\!\!\$\ \mathsf{Patch}_\delta(m_{l,k},\widetilde{\sigma}_{l,k})$, for $\widetilde{\sigma}_{l,k} \leftarrow\!\!\$\ \widetilde{\mathsf{A}}_k(sk,m_{l,k})$. Now, since $\widetilde{\sigma}_{l,k}$ is valid, the fact that signatures are re-randomizable directly implies that the $\mathbf{H}_{k,l-1}$ and $\mathbf{H}_{k,l}$ are identical. The statement follows. □

Claim 7.2. $\mathbb{P}[\mathsf{B} \text{ wins in } \mathbf{H}] \leq qn \cdot \varepsilon$.

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.[22] We build an adversary B' (using B) that breaks EUF-CMA 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 $(j,m_{i,j})$ answer as follows:
  - In case $j < j^*$, forward $m_{i,j}$ to the signing oracle, obtaining $\widetilde{\sigma}_{i,j} \leftarrow\!\!\$\ \mathsf{Sign}(sk,m_i)$, and return $\widetilde{\sigma}_{i,j}$ and $\delta' = (vk,0)$ to B.
  - In case $j = j^*$, if $i < i^*$ forward $m_{i,j}$ to the signing oracle, obtaining $\widetilde{\sigma}_{i,j} \leftarrow\!\!\$\ \mathsf{Sign}(sk,m_i)$, and return $\widetilde{\sigma}_{i,j}$ and $\delta' = (vk,0)$ to B. Else, return $\perp$ and $\delta' = (vk,1)$.

[22] 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.

  - In case $j > j^*$ answer with $(\bot, \delta' = (vk, 1))$.
- 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 $\mathbb{P}[\mathsf{B}' \text{ wins}] \geq (qn \cdot \varepsilon)/(qn) = \varepsilon$, a contradiction. □

The proof follows by combining the above two claims. □

7 The Multi-User Setting

In this section, we consider the multi-user setting for all the definitions of Section 3. We also provide a complete picture of relationships between the different definitions, as shown in Fig. 5 and Fig. 6.

7.1 Multi-User Security

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

Indistinguishability. In the indistinguishability definition for the multi-user setting, adversary B now receives $u \geq 1$ verification keys from the challenger and can continuously subvert each user independently. Similarly to Definition 6, multiple subversions of the same user are not allowed to share state. A formal definition follows.

Definition 20 (Indistinguishability against SAs—Multi-user setting)**.** 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 $u(\kappa)$-users indistinguishable w.r.t. *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B there exists a negligible function $\varepsilon : \mathbb{N} \to [0, 1]$, such that $\left|\mathbb{P}[\mathsf{B} \text{ wins}] - \frac{1}{2}\right| \leq \varepsilon(\kappa)$ in the following game:

1. The challenger samples $b \leftarrow\!\!\$\ \{0, 1\}$, generates $(vk_\ell, sk_\ell) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$ for $\ell \in [u]$, and gives $vk_1, \ldots, vk_u$ to B.
2. For each user $\ell \in [u]$, the adversary B can specify polynomially many algorithms $\widetilde{\mathsf{A}}_{\ell,j} \in \mathcal{A}$. Each such algorithm implicitly defines an oracle that can be queried adaptively polynomially many times.
   - Upon input a query of the form (ℓ, j, m), where $\ell \in [u]$, the answer depends on the value of the secret bit b: If $b = 1$, the output is $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk_\ell, m)$; if $b = 0$, the output is $\widetilde{\sigma} \leftarrow\!\!\$\ \widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, m)$. In case the algorithm $\widetilde{\mathsf{A}}_{\ell,j}$ is undefined the oracle returns $\bot$.
   - Note that B can interleave queries between different oracles in an arbitrary way.
3. Finally, B outputs a value $b' \in \{0, 1\}$; we say that B wins iff $b' = b$.

Impersonation. In the impersonation definition for the multi-user setting, adversary B now receives $u \geq 1$ verification keys from the challenger and can continuously subvert each user independently; adversary B is successful if it can impersonate *any* of the users. Similarly to Definition 7, multiple subversions of the same user are not allowed to share state. A formal definition follows.

Definition 21 (EUF-CMA against SAs—Multi-user setting)**.** 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 $u(\kappa)$-users EUF-CMA w.r.t. *continuous* $\mathcal{A}$-SAs if for all PPT adversaries B there exists a negligible function $\varepsilon : \mathbb{N} \to [0, 1]$, such that $\mathbb{P}[\mathsf{B} \text{ wins}] \leq \varepsilon(\kappa)$ in the following game:

1. The challenger generates $(vk_\ell, sk_\ell) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$ for $\ell \in [u]$, and gives $vk_1, \ldots, vk_u$ to B.
2. The adversary B can submit polynomially many queries of the form $(\ell \in [u], m)$ to the challenger that returns $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk_\ell, m)$.
3. For each user $\ell \in [u]$, the adversary B can specify polynomially many algorithms $\widetilde{\mathsf{A}}_{\ell,j} \in \mathcal{A}$. Each such algorithm implicitly defines an oracle that can be queried adaptively polynomially many times.
   - Upon input a query of the form (ℓ, j, m), where $\ell \in [u]$, the output is $\widetilde{\sigma} \leftarrow\!\!\$\ \widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, m)$. In case the algorithm $\widetilde{\mathsf{A}}_{\ell,j}$ is undefined the oracle returns $\perp$.
   - Note that B can interleave queries between different oracles in an arbitrary way, and that the queries in step 2 and step 3 can also be interleaved arbitrarily.
4. For each $\ell \in [u]$, let $\mathcal{Q}_\ell$ be the set of all messages queried to oracle $\mathsf{Sign}(sk_\ell, \cdot)$ and $\widetilde{\mathcal{Q}}_{\ell,j}$ be the set of all messages queried to oracle $\widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, \cdot)$.
5. Finally, B outputs a tuple (m^*, σ^*, ℓ^*); we say that B wins iff $\mathsf{Vrfy}(vk_{\ell^*}, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q}_{\ell^*} \cup \widetilde{\mathcal{Q}}_{\ell^*}$, where $\widetilde{\mathcal{Q}}_{\ell^*} := \bigcup_j \widetilde{\mathcal{Q}}_{\ell^*,j}$.

Security relations. Theorem 8 below formalizes the relations between the notions of impersonation/indistinguishability in the presence of SAs, which are depicted in Fig. 5. Note that for $u = 1$ Definition 20 and Definition 21 collapse, respectively, to Definition 6 and Definition 7.

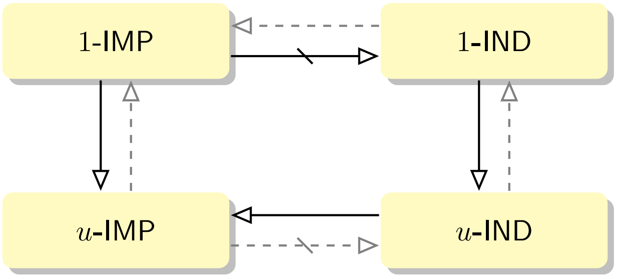


Figure 5: Diagram of the relationships between the subversion notions considered in this paper. $X \to Y$ means that X implies Y (for all SA classes $\mathcal{A}$); $X \nrightarrow Y$ indicates a separation between X and Y (for some specific SA class $\mathcal{A}$). The lighter arrows indicates trivial implications (or implications that follow from Theorem 8). Indistinguishability (cf. Definition 6) is represented by 1-IND and Impersonation (cf. Definition 7) is represented by 1-IMP. Multi-user Indistinguishability (cf. Definition 20) is represented by u-IND and multi-user Impersonation (cf. Definition 21) is represented by u-IMP.

Theorem 8. *The following relations hold.*

(i) (1-IND $\to$ u-IND) *For all signature schemes $\mathcal{SS}$ and all SA classes $\mathcal{A}$ against $\mathcal{SS}$, if $\mathcal{SS}$ is* 1*-user indistinguishable w.r.t. continuous $\mathcal{A}$-SAs, then, for any $u \in poly(\kappa)$, it is also u-users indistinguishable w.r.t. continuous $\mathcal{A}$-SAs.*

(ii) (1-$\mathsf{IMP} \not\rightarrow$ 1-IND) *Assuming the existence of EUF-CMA signature schemes, there exist a signature scheme $\mathcal{SS}$ and a SA class $\mathcal{A}$ against $\mathcal{SS}$ such that $\mathcal{SS}$ is* 1*-user EUF-CMA w.r.t. continuous $\mathcal{A}$-SAs, but it is not* 1*-user indistinguishable w.r.t. continuous $\mathcal{A}$-SAs.*

(iii) (u-$\mathsf{IND} \rightarrow u$-$\mathsf{IMP}$) *Let $\mathcal{SS}$ be a EUF-CMA signature scheme. For all SA classes $\mathcal{A}$ against $\mathcal{SS}$, and for any $u \in poly(\kappa)$, if $\mathcal{SS}$ is u-users indistinguishable w.r.t. continuous $\mathcal{A}$-SAs, then it is also u-users EUF-CMA w.r.t. continuous $\mathcal{A}$-SAs.*

(iv) (1-$\mathsf{IMP} \rightarrow u$-$\mathsf{IMP}$) *For all signature schemes $\mathcal{SS}$ and all SA classes $\mathcal{A}$ against $\mathcal{SS}$, if $\mathcal{SS}$ is* 1*-user EUF-CMA w.r.t. continuous $\mathcal{A}$-SAs then, for any $u \in poly(\kappa)$, it is also u-users EUF-CMA w.r.t. continuous $\mathcal{A}$-SAs.*

Proof. (i) Towards contradiction, consider an adversary B that wins the game described in Definition 20. We build an adversary B' that (using B) wins the game described in Definition 6. Let $\mathbf{G}$ be the game described in Definition 20. 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 $\ell^* \in [0, u]$, consider the hybrid game $\mathbf{H}_{\ell^*}$ where each oracle corresponding to query $(\ell, j, \cdot)$ such that $\ell \leq \ell^*$ behaves as $\widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, \cdot)$ (i.e., as in game $\mathbf{G}_0$), while all oracles corresponding to queries $(\ell, j, \cdot)$ such that $\ell > \ell^*$ behave as $\mathsf{Sign}(sk_\ell, \cdot)$ (i.e., as in game $\mathbf{G}_1$). We note that $\mathbf{H}_0 \equiv \mathbf{G}_1$ and $\mathbf{H}_u \equiv \mathbf{G}_0$. We can construct B' as follows.

<u>Adversary B':</u>

1. Sample a random $\ell^* \leftarrow\!\!\$\, [u]$.
2. Receive vk^* from the challenger and sample $(vk_\ell, sk_\ell) \leftarrow\!\!\$\, \mathsf{KGen}(1^\kappa)$ for all $\ell \in [u] \setminus \{\ell^*\}$. Define $vk_{\ell^*} = vk^*$ and forward $(vk_1, \ldots, vk_u)$ to adversary B.
3. Whenever B outputs a subversion $\widetilde{\mathsf{A}}_{\ell,j}$, if $\ell = \ell^*$ forward it to the challenger.
4. Upon input a query (ℓ, j, m) from B, behave as follows.
   - If $\ell \leq \ell^* - 1$ answer with $\widetilde{\sigma} \leftarrow\!\!\$\, \widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, m)$.
   - If $\ell = \ell^*$ forward m to the challenger and send the reply to B.
   - If $\ell \geq \ell^* + 1$ answer with $\sigma \leftarrow\!\!\$\, \mathsf{Sign}(sk_\ell, m)$.
5. Output whatever B outputs.

By assumption, we know that B can distinguish between the extreme hybrid games $\mathbf{H}_0$ and $\mathbf{H}_u$, so there must exist an index $\ell^*(\kappa) \in [0, u(\kappa)]$ such that B can distinguish $\mathbf{H}_{\ell^*(\kappa)}$ and $\mathbf{H}_{\ell^*(\kappa)-1}$ with a non-negligible advantage. Therefore,

$$\begin{aligned}\left|\mathbb{P}\left[\mathsf{B}'(\mathbf{G}_1) = 1\right] - \mathbb{P}\left[\mathsf{B}'(\mathbf{G}_0) = 1\right]\right| &= \frac{1}{u} \cdot \left|\sum_{\ell^*=0}^{u-1} \mathbb{P}\left[\mathsf{B}(\mathbf{H}_{\ell^*+1}) = 1\right] - \mathbb{P}\left[\mathsf{B}(\mathbf{H}_{\ell^*}) = 1\right]\right| \\ &= \frac{1}{u} \cdot \left|\mathbb{P}\left[\mathsf{B}(\mathbf{H}_u) = 1\right] - \mathbb{P}\left[\mathsf{B}(\mathbf{H}_0) = 1\right]\right| \\ &\geq 1/poly(\kappa),\end{aligned}$$

which is a contradiction, finishing the proof.

(ii) Consider $\mathcal{SS}$ to be an EUF-CMA signature scheme with signature size ℓ bits, and let $\mathcal{A}$ be the class of SAs for $\mathcal{SS}$ that always outputs 0^ℓ as the signature of any message $m \in \mathcal{M}$. By $\mathcal{SS}$ being EUF-CMA, adversary B has only a negligible probability of winning at the game described in Definition 7. Consider the adversary B against the game described in Definition 6.

<u>Adversary B:</u>

1. The challenger samples $b \leftarrow\!\!\mathrm{s}\ \{0,1\}$, runs $(vk, sk) \leftarrow\!\!\mathrm{s}\ \mathsf{KGen}(1^\kappa)$ and forwards vk to B.
2. B queries the oracle for an arbitrary message m and receives σ as a reply.
3. If $\sigma = 0^\ell$ then output 0, otherwise output 1.

Adversary B clearly has a non-negligible probability of distinguishing the real signing oracle from the subversion oracle in the game of Definition 6.

(iii) Let $\mathbf{G}$ be the game described in Definition 21; consider the hybrid game $\mathbf{H}$ that behaves exactly like $\mathbf{G}$, except that queries of type (ℓ, j), for $\ell \in [u]$, are answered using $\mathsf{Sign}(sk_\ell, \cdot)$ instead of $\widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, \cdot)$. We claim that for all PPT adversaries B, there exists a negligible function $\varepsilon' : \mathbb{N} \to [0,1]$ such that

$$|\mathbb{P}[\mathsf{B} \text{ wins } \mathbf{G}] - \mathbb{P}[\mathsf{B} \text{ wins } \mathbf{H}]| \leq \varepsilon'(\kappa). \tag{6}$$

In fact, $\mathbf{G}$ and $\mathbf{H}$ are computationally indistinguishable, as any distinguisher between the two games directly yields an efficient adversary against the u-users indistinguishability w.r.t. continuous $\mathcal{A}$-SAs of $\mathcal{SS}$.

On the other hand, note that in game $\mathbf{H}$ all queries are answered using the real signing algorithm. Thus, a straightforward reduction to the EUF-CMA security of $\mathcal{SS}$ implies that there exists a negligible function $\varepsilon'' : \mathbb{N} \to [0,1]$ such that, for all PPT adversaries B,

$$\mathbb{P}[\mathsf{B} \text{ wins } \mathbf{H}] \leq \varepsilon''(\kappa). \tag{7}$$

Putting Eq. (6) and Eq. (7) together, we conclude that for all PPT adversaries B there exists a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that

$$\mathbb{P}[\mathsf{B} \text{ wins } \mathbf{G}] \leq \varepsilon'(\kappa) + \varepsilon''(\kappa) \leq \varepsilon(\kappa),$$

concluding the proof.

(iv) Consider an adversary B that wins the game described in Definition 21. We build an adversary B' that (using B) wins the game described in Definition 7.

Adversary B':

1. Receive vk^* from the challenger, sample $\ell^* \leftarrow\!\!\mathrm{s}\ [u]$ and $(vk_\ell, sk_\ell) \leftarrow\!\!\mathrm{s}\ \mathsf{KGen}(1^\kappa)$ for all $\ell \in [u] \setminus \{\ell^*\}$. Set $vk_{\ell^*} := vk^*$ and forward $(vk_1, \ldots, vk_u)$ to B.
2. Upon each query (ℓ, m), for $\ell \in [u]$: If $\ell \neq \ell^*$ reply with $\sigma \leftarrow\!\!\mathrm{s}\ \mathsf{Sign}(sk_\ell, m)$, else forward the query to the challenger.
3. Whenever B outputs a subversion $\widetilde{\mathsf{A}}_{\ell,j}$, if $\ell = \ell^*$ forward it to the challenger.
4. Upon each query (ℓ, j, m), with $\ell \in [u]$, behave as follows.
    - If $\ell \neq \ell^*$, answer with $\widetilde{\sigma} \leftarrow\!\!\mathrm{s}\ \widetilde{\mathsf{A}}_{\ell,j}(sk_\ell, m)$, else forward the query to the challenger.
5. Eventually B outputs a forgery (ℓ', m', σ'); adversary B' outputs (m', σ') as its own forgery.

Adversary B' is successful if adversary B outputs a valid forgery for user ℓ^*. Define E to be the event that B' guesses correctly the index $\ell' = \ell^*$; note that $\mathbb{P}[E] = 1/u$. Therefore, since u is a polynomial in the security parameter, adversary B' has a non-negligible probability of winning at the game described in Definition 7. □

7.2 Multi-User Public/Secret Undetectability

In the undetectability definition for the multi-user setting, user 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 according to an efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$ that is a parameter in the definition. A formal definition follows.

Definition 22 (Public/secret undetectability—Multi-user)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$, and $\mathbf{D}_{\mathcal{A}}$ be an efficiently samplable distribution over $\mathcal{A}$. We say that $\mathcal{A}$ is $u(\kappa)$-users *secretly* $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ if for all PPT users U, there exists a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that $\left|\mathbb{P}[\mathsf{U} \text{ wins}] - \frac{1}{2}\right| \leq \varepsilon(\kappa)$ in the following game:

1. The challenger samples $b \leftarrow_{\$} \{0,1\}$, generates $(vk_\ell, sk_\ell) \leftarrow_{\$} \mathsf{KGen}(1^\kappa)$ for $\ell \in [u]$, samples $\widetilde{\mathsf{A}} \leftarrow_{\$} \mathbf{D}_{\mathcal{A}}$, and gives $((vk_1, sk_1), \dots, (vk_u, sk_u))$ to U. Let $\widetilde{\mathsf{A}}_1, \dots, \widetilde{\mathsf{A}}_u$ be u identical copies of $\widetilde{\mathsf{A}}$.

2. The user U can ask polynomially many queries of the form (ℓ, m), where $\ell \in [u]$. The answer to each query depends on the secret bit b. If $b = 1$, the challenger returns $\sigma \leftarrow_{\$} \mathsf{Sign}(sk_\ell, m)$; if $b = 0$, the challenger returns $\widetilde{\sigma} \leftarrow_{\$} \widetilde{\mathsf{A}}_\ell(sk_\ell, m)$.

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.

Undetectability relations. Theorem 9, below, formalizes the relations between the notions of public/secret undetectability in the presence of SAs, which are depicted in Fig. 6.

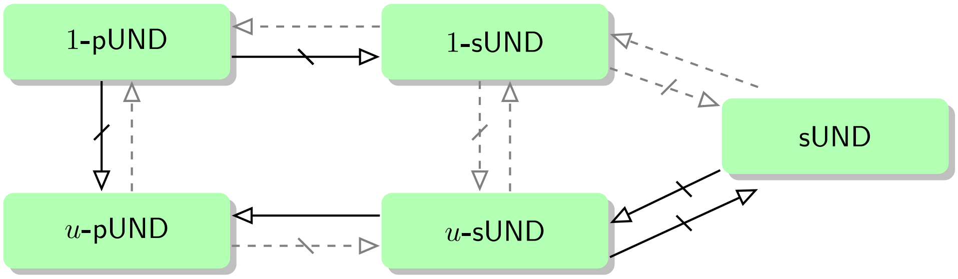


Figure 6: Diagram of the relationships between the undetectability notions considered in this paper. $X \to Y$ means that X implies Y (for all SA classes $\mathcal{A}$); $X \not\to Y$ indicates a separation between X and Y (for some specific SA class $\mathcal{A}$). The lighter arrows indicates trivial implications (or implications that follow from Theorem 9). For $u \geq 2$, public undetectability is represented by u-pUND and secret undetectability is represented by u-sUND (cf. Definition 22). Secret undetectability (cf. Definition 8) is represented by sUND, whereas public undetectability (cf. Definition 8) is equivalent to 1-pUND.

Theorem 9. *The following relations hold.*

(i) (u-sUND $\to$ u-pUND) *For any signature scheme $\mathcal{SS}$, all SA classes $\mathcal{A}$ against $\mathcal{SS}$ and all efficiently samplable distributions $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, if $\mathcal{A}$ is u-users secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$, then it is also u-users publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$.*

(ii) (1-pUND $\nrightarrow$ 1-sUND) *There exist a signature scheme $\mathcal{SS}$, a SA class $\mathcal{A}$ against $\mathcal{SS}$ and an efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, such that $\mathcal{A}$ is 1-user publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ but it is* not *1-user secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$.*

(iii) (1-pUND $\nrightarrow$ u-pUND) *There exist a signature scheme $\mathcal{SS}$, a SA class $\mathcal{A}$ against $\mathcal{SS}$ and an efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, such that $\mathcal{A}$ is 1-user publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ but it is* not *u-users publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ (for any $u \geq 2$).*

(iv) (sUND $\nrightarrow$ u-sUND) *Assuming that PRGs exist, there exist a signature scheme $\mathcal{SS}$, a SA class $\mathcal{A}$ against $\mathcal{SS}$ and an efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, such that $\mathcal{A}$ is 1-user secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ but it is* not *u-users secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ (for any $u \geq 2$).*

(v) (u-sUND $\nrightarrow$ sUND) *There exist a signature scheme $\mathcal{SS}$, a SA class $\mathcal{A}$ against $\mathcal{SS}$ and an efficiently samplable distribution $\mathbf{D}_{\mathcal{A}}$ over $\mathcal{A}$, such that $\mathcal{A}$ is u-users secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ but it is* not *1-user secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$ (for any $u \geq 1$).*

Proof. (i) Towards contradiction, consider a user U that wins the u-users public undetectability game described in Definition 22. We build a user U′ (using U) that wins the u-users secret undetectability game described in Definition 22.

User U′:

1. Receive $(vk_\ell, sk_\ell) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$, from the challenger, for $\ell \in [u]$, and forward $(vk_1, \ldots, vk_u)$ to user U.
2. User U asks polynomially many queries of the type (ℓ, m) which are forwarded to the challenger.
3. Output whatever U outputs.

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

(ii) Consider $\mathcal{SS}$ to be a randomized signature (with only two valid signatures for each message $m \in \mathcal{M}$), and let $\mathcal{SS}'$ be its derandomized implementation s.t. $sk' := (sk, b)$, $vk' := vk$, and $\sigma' := \mathsf{Sign}(sk, m; b)$, for a random $b \in \{0, 1\}$. Let $\mathcal{A} = \{\widetilde{\mathsf{A}}\}$ be the class of SAs for $\mathcal{SS}'$ which consists of a single algorithm described next.

$\widetilde{\mathsf{A}}((sk, b), m)$:

1. Let $b' := 1 - b$ (i.e., b' is the complement bit of b).
2. Output $\sigma := \mathsf{Sign}(sk, m; b')$.

Let $\mathbf{D}_{\mathcal{A}}$ be the distribution over $\mathcal{A}$ that always returns $\widetilde{\mathsf{A}}$. We note that the class of SAs $\mathcal{A}$ is 1-user publicly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}'$, i.e. any PPT user U has only a negligible probability of winning at the public undetectability game described in Definition 8. On the other hand, a user U playing the secret undetectability game (with knowledge of $sk' = (sk, b)$) can easily detect the subversion by simply comparing the output of the target oracle with that of $\mathsf{Sign}(sk, m; b')$.

(iii) and (iv) Consider $\mathcal{SS}$ to be a public-coin signature scheme, so that the signature of a message $m \in \mathcal{M}$ is $\sigma' := \sigma || r := \mathsf{Sign}(sk, m; r)$, where $r \in \mathcal{R}$ are the coins sampled to generate σ. Let $\mathcal{A} = \{\widetilde{\mathsf{A}}_{\bar{r}, \tau=0}\}_{\bar{r} \in \mathcal{R}}$ to be class of SAs for $\mathcal{SS}$ described next.

```
$\widetilde{\mathsf{A}}_{\bar{r},\tau}(sk, m)$:
    1. If $\tau = 0$ then set $r := \bar{r}$, else set $r \leftarrow\!\!\$\, \mathcal{R}$;
    2. Set $\tau := \tau + 1$;
    3. Output $\sigma || r := \mathsf{Sign}(sk, m; r)$.
```

Let $\mathbf{D}_{\mathcal{A}}$ to be the uniform distribution over $\mathcal{A}$. Clearly, the class $\mathcal{A}$ is 1-user publicly/secretly $\mathbf{D}_{\mathcal{A}}$-undetectable w.r.t. $\mathcal{SS}$, because the output of the subverted signature algorithm is indistinguishable 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/secretly $\mathbf{D}_{\mathcal{A}}$-*detectable* w.r.t. $\mathcal{SS}$, since it suffices to ask one query for each signing key (i.e., each signing oracle) and compare the last part of the signatures (i.e., the random coins).

(v) Consider $\mathcal{SS}$ to be any signature scheme. Let $\mathcal{A} = \{\widetilde{\mathsf{A}}_{\tau=0^\kappa}\}$ be the class of SAs for $\mathcal{SS}$ which consists of the algorithm described next.

```
$\widetilde{\mathsf{A}}_{\tau}(sk, m)$:
    1. If $\tau \neq sk$ and $\tau \neq 0^\kappa$ then $\sigma := \bot$;
    2. Else, $\sigma \leftarrow\!\!\$\, \mathsf{Sign}(sk, m)$;
    3. $\tau := sk$;
    4. Output $\sigma$.
```

Let $\mathbf{D}_{\mathcal{A}}$ be the distribution that always returns $\widetilde{\mathsf{A}}_{0^\kappa}$. A user U playing the secret undetectability game of Definition 8 can easily win the game by making two queries with different signing keys; if the answer to the second query is $\bot$, then U detects the subversion. On the other hand, all the queries made by a user U' playing the u-users secret undetectability game of Definition 22 will only produce a real signature, since a single signing key is used for each copy of the subversion oracle. □

Comparison between undetectability and multi-user undetectability. The above theorem shows that the *secret* undetectability notion of Definition 8 is actually incomparable to the multi-user *secret* undetectability notion of Definition 22. (For *public* undetectability, instead, Definition 8 is equivalent to Definition 22 with $u = 1$.) The latter is due to the fact that in the game of Definition 8 the challenger queries a single oracle, whereas in the game of Definition 22 the challenger queries different copies of the same oracle (one for each signing key).

This difference creates a gap between the two notions in the case of stateful SAs. Nevertheless, it is possible to slightly modify our attack from Section 4.2 in order to get secret undetectability in the multi-user setting.[23] On the other hand, for stateless SAs secret undetectability as per Definition 8 *does* imply u-users secret undetectability as per Definition 22 (for any $u \in poly(\kappa)$); in particular, the attack described in Fig. 1 is already secretly undetectable in the multi-user setting.

7.3 Multi-User Signing Key Recovery

In this section, we extend the key recovery definition (Definition 9) for the multi-user setting.

[23] Briefly, we generate the biased randomness via a PRF (instead of a PRG); the PRF has a fixed key, hard-wired in the subverted algorithm, and takes as input the signing key of each user and a counter that is incremented on every call.

Definition 23 (Key recovery—Multi-user setting)**.** Let $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, $\mathcal{A}$ be some class of SAs for $\mathcal{SS}$, and $\mathbf{D}_{\mathcal{A}}$ be an efficiently samplable distribution over $\mathcal{A}$. We say that adversary B recovers the signing key of all $u(\kappa)$-users of $\mathcal{SS}$ w.r.t. message sampler M and distribution $\mathbf{D}_{\mathcal{A}}$ if there exists a non-neglible function $\varepsilon : \mathbb{N} \to [0,1]$, such that $\mathbb{P}[\mathsf{B} \text{ wins}] \geq \varepsilon(\kappa)$ in the following game:

1. The challenger runs $(vk_\ell, sk_\ell) \leftarrow\$ \mathsf{KGen}(1^\kappa)$ and samples $\widetilde{\mathsf{A}}_\ell \leftarrow\$ \mathbf{D}_{\mathcal{A}}$, for each $\ell \in [u]$, and gives $(vk_1, \cdots, vk_u)$ to B.

2. For each $\ell \in [u]$, adversary B is given access to an oracle that can be queried polynomially many times: Upon an empty input, the oracle samples $m \leftarrow\$ \mathsf{M}$, computes $\widetilde{\sigma} \leftarrow \widetilde{\mathsf{A}}_\ell(sk_\ell, m)$, and sends $(m, \widetilde{\sigma})$ to B.

3. Finally, B outputs the keys $(sk'_1, \cdots, sk'_u)$; we say that B wins iff $sk'_\ell = sk_\ell$ for all $\ell \in [u]$.

As in Definition 9, we note that M is a message sampler algorithm, that chooses the message to be signed according to some pre-defined strategy.

Both attacks from Section 4 satisfy Definition 23, meaning that they can recover the signing keys of *all* users of the scheme. More precisely, for the attack of Fig. 1, the key recovery probability of adversary B, described in Theorem 1, is $\varepsilon \geq (1 - \varepsilon_{\mathsf{prf}} - \ell e^{-q/\ell} - q^2\tau^2 \cdot 2^{-\eta-1} - q \cdot 2^{-\tau})^u$. For the attack of Fig. 2, the key recovery probability of adversary B, defined in Theorem 2, is $\varepsilon \geq (1 - \nu_{ext} \cdot \ell/d)^u$.

Acknowledgements

This work was supported by the European Commission H2020 program under the SUNFISH project, grant N.644666, and by the European Commission Directorate General Home Affairs, under the GAINS project, HOME/2013/CIPS/AG/4000005057.

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

References

[ABK18] Benedikt Auerbach, Mihir Bellare, and Eike Kiltz. Public-key encryption resistant to parameter subversion and its realization from efficiently-embeddable groups. In *PKC*, pages 348–377, 2018.

[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.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[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.

[AMV15] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signature schemes. In *CCS*, pages 364–375, 2015.

[AVPN96] Ross J. Anderson, Serge Vaudenay, Bart Preneel, and Kaisa Nyberg. The newton channel. In *Information Hiding*, pages 151–156, 1996.

[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.

[BFGM01] Mihir Bellare, Marc Fischlin, Shafi Goldwasser, and Silvio Micali. Identification protocols secure against reset attacks. In *EUROCRYPT*, pages 495–511, 2001.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *CRYPTO*, pages 360–378, 2008.

[BFS16] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In *ASIACRYPT*, pages 777–804, 2016.

[BH15] Mihir Bellare and Viet Tung Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In *EUROCRYPT*, pages 627–656, 2015.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via uces. In *CRYPTO*, pages 398–415, 2013.

[BJK15] Mihir Bellare, Joseph Jaeger, and Daniel Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In *CCS*, pages 1431–1440, 2015.

[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.

[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.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In *EUROCRYPT*, pages 272–287, 2002.

[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*, pages 579–598, 2015.

[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.

[DMS16] Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. Message transmission with reverse firewalls - Secure communication on corrupted machines. In *CRYPTO*, pages 341–372, 2016.

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

[DPSW16] Jean Paul Degabriele, Kenneth G. Paterson, Jacob C. N. Schuldt, and Joanne Woodage. Backdoors in pseudorandom number generators: Possibility and impossibility results. In *CRYPTO*, pages 403–432, 2016.

[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.

[FHN$^+$12] Sebastian Faust, Carmit Hazay, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Angela Zottarel. Signature schemes secure against hard-to-invert leakage. In *ASIACRYPT*, pages 98–115, 2012.

[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.

[FNV15] Antonio Faonio, Jesper Buus Nielsen, and Daniele Venturi. Mind your coins: Fully leakage-resilient signatures with graceful degradation. In *ICALP*, pages 456–468, 2015.

[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.

[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.

[NVZ14] Jesper Buus Nielsen, Daniele Venturi, and Angela Zottarel. Leakage-resilient signatures with graceful degradation. In *PKC*, pages 362–379, 2014.

[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.

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM J. Comput.*, 40(6):1803–1844, 2011.

[RTYZ16] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Cliptography: Clipping the power of kleptographic attacks. In *ASIACRYPT*, pages 34–64, 2016.

[RTYZ17] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Generic semantic security against a kleptographic adversary. In *ACM CCS*, pages 907–922, 2017.

[RTYZ18] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Correcting subverted random oracles. In *CRYPTO*, pages 241–271, 2018.

[Sch12] Sven Schäge. Strong security from probabilistic signature schemes. In *PKC*, pages 84–101, 2012.

[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.

[Sim85] Gustavus J. Simmons. A secure subliminal channel (?). In *CRYPTO*, pages 33–41, 1985.

[Sim93] Gustavus J. Simmons. Subliminal communication is easy using the DSA. In *EUROCRYPT*, pages 218–232, 1993.

[Sim94] Gustavus J. Simmons. Subliminal channels; past and present. *European Transactions on Telecommunications*, 5(4):459–474, 1994.

[Sim98] Gustavus J. Simmons. The history of subliminal channels. *IEEE Journal on Selected Areas in Communications*, 16(4):452–462, 1998.

[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.