# Strongly-Optimal Structure Preserving Signatures from Type II Pairings: Synthesis and Lower Bounds[⋆]

Gilles Barthe[1], Edvard Fagerholm[1,2], Dario Fiore[1],
Andre Scedrov[2], Benedikt Schmidt[1], and Mehdi Tibouchi[3]

[1] IMDEA Software Institute, Madrid, Spain
{gilles.barthe, dario.fiore, benedikt.schmidt}@imdea.org
[2] University of Pennsylvania
{edvardf,scedrov}@math.upenn.edu
[3] NTT Secure Platform Laboratories
tibouchi.mehdi@lab.ntt.co.jp

**Abstract.** Recent work on structure-preserving signatures studies optimality of these schemes in terms of the number of group elements needed in the verification key and the signature, and the number of pairing-product equations in the verification algorithm. While the size of keys and signatures is crucial for many applications, another important aspect to consider for performance is the time it takes to verify a given signature. By far, the most expensive operation during verification is the computation of pairings. However, the concrete number of pairings that one needs to compute is not captured by the number of pairing-product equations considered in earlier work.
To fill this gap, we consider the question of what is the minimal number of pairings that one needs to compute in the verification of structure-preserving signatures. First, we prove lower bounds for schemes in the Type II setting that are secure under chosen message attacks in the generic group model, and we show that three pairings are necessary and that at most one of these pairings can be precomputed. We also extend our lower bound proof to schemes secure under random message attacks and show that in this case two pairings are still necessary.
Second, we build an automated tool to search for schemes matching our lower bounds. The tool can generate automatically and exhaustively all valid structure-preserving signatures within a user-specified search space, and analyze their (bounded) security in the generic group model. Interestingly, using this tool, we find a new randomizable structure-preserving signature scheme in the Type II setting that is optimal with respect to the lower bound on the number of pairings, and also minimal with respect to the number of group operations that have to be computed during verification.

## 1 Introduction

Structure-preserving signatures [3] (SPS) are signature schemes defined over groups with a bilinear map in which messages, public keys and signatures are all group elements, and the verification algorithm consists of evaluating so-called "pairing-product equations" (i.e., products of pairings of the aforementioned group elements). One of the main motivations of considering such specific signature schemes is that they are remarkably useful in the modular design of several cryptographic protocols, notably in combination with non-interactive zero-knowledge (NIZK) proofs of knowledge about group elements, and more specifically with the celebrated Groth-Sahai proof system [24]. In a nutshell, Groth-Sahai proofs allow one to prove knowledge of a set of group elements satisfying a certain pairing-product equation. For instance, by using SPS with Groth-Sahai proofs one can create a NIZK proof showing knowledge of a valid signature on some message (perhaps satisfying certain properties) without disclosing the message, the signature or both. This is only a basic example, though. Indeed, the combination of SPS with Groth-Sahai proofs has been shown to be a powerful tool for the modular design of several cryptographic protocols, such as blind signatures [3,20], group signatures [3,20,27], homomorphic signatures [10,26], oblivious transfer [22,13], tightly-secure encryption [25,1], and more.

[⋆] © IACR 2015. This article is the full version of a paper that appears in the proceedings of PKC 2015.

Realization of SPS has been considered over the three possible bilinear groups settings introduced in the classification of Galbraith, Paterson and Smart [21]; the type of a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ depends on whether the two source groups are the same, i.e., $\mathbb{G}_1 = \mathbb{G}_2$ (*Type I*), or there is a one-way, efficiently computable homomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ (*Type II*), or there is no known efficiently computable homomorphism in either direction between $\mathbb{G}_2$ and $\mathbb{G}_1$ (*Type III*). However, more recent work has focused on proving lower bounds on the complexity of SPS, and exhibiting optimal constructions that match lower bounds. The common measures of complexity adopted in all these works are the number of group elements in the public key, the number of group elements in the signature, and the number of pairing-product equations in the verification algorithm. Considering these measures, it has been shown in [4,8] that in both the Type I and the Type III settings SPS require at least 3 group elements in the signatures and 2 verification equations. However, the Type II setting has been shown to (surprisingly) deviate from these bounds: SPS in the Type II setting require at least 2 group elements in the signatures and admit a *single* verification equation [7]. Moreover, for SPS in the Type II setting, it has been shown that the lower bound for the number of group elements in the verification key is 2. Together with showing such lower bounds, these works [4,8,7] have proposed SPS schemes matching these (optimal) measures.

## 1.1 Our Contribution

We continue the study of the efficiency of SPS schemes by focusing on another important measure that, to the best of our knowledge, has not been considered in any previous work: the number of pairing computations that need to be performed by the verifier. Previous work [4,8,7] considers verifier efficiency only in terms of the number of pairing-product equations. Such a number, however, does not tell much about the number of pairings that the verifier needs to compute, and thus about the concrete verification running time. So, considering that pairings are definitely the most expensive operation in this process, here we refine this question and ask what is the *minimal* number of pairings necessary in the verification of SPS, and in particular of schemes with optimal bandwidth (i.e., 2 elements in the signatures and 2 elements in the verification key). Indeed, even though having fewer elements in the public key and in the signature intuitively leads to fewer pairings, in practice it is unclear what is the minimal number of pairings that is needed.

In this paper we initiate this study focusing on the Type II setting, and our contribution is mainly twofold. First, we show lower bounds on the number of pairings necessary in the pairing-product verification equation. Second, we build a synthesis tool that automates the generation and security analysis of SPS schemes, and we leverage our tool in order to find new SPS schemes that match our new lower bounds and improve over previous work. In the following paragraphs, we discuss our contribution in more detail.

NEW OPTIMALITY MEASURES AND LOWER BOUNDS. First, we show lower bounds for the number of pairings in the pairing-product verification equation of SPS in the Type II setting. We prove that, when considering schemes that are already optimal with respect to previously considered measures (i.e., two group elements in the verification key, two group elements in the signature, and a single verification equation), *three* pairings are necessary for achieving security against chosen-message attacks, whereas *two* pairings are necessary for achieving security against random-message attacks.

More specifically, we refine our analysis and distinguish between, what we call, offline and online pairings. Informally speaking, *offline pairings* are pairings that involve only group elements in the public key or in the public parameters, whereas *online pairings* involve the message and/or elements of the signature. In other words, offline pairings are computed in every signature verification (when using the same verification key) and thus can be precomputed "offline" and be re-used in an arbitrary number of verifications. In contrast, online pairings involve elements, such as the message and the signature, that inherently change every time, and thus must be computed "online". So, given this notion of online and offline pairings, we ask how many of the three necessary pairings can be computed offline. Such question is indeed quite relevant for practical purposes since online pairings are those that really matter (e.g., think of the case in which one verifies several signatures with the same verification key). We answer this question by proving that, for schemes secure against chosen-message attacks, among the three pairings, *at most one* can be precomputed, i.e., two online pairings are necessary. For schemes that are secure against random-message attacks, instead two online pairings are always necessary. We call schemes matching these bounds *strongly-optimal*.

Once established these bounds, we address the question of constructing strongly-optimal SPS schemes. First, we consider schemes secure against chosen-message attacks: we look at the previous work (in the Type II setting) and observe that there already exists a strongly-unforgeable SPS matching our lower bounds [7]. Yet there is no known SPS scheme that is re-randomizable and allows for only two online pairings in verification. As discussed in [7], re-randomizable schemes are useful because one of the group elements in the signature is uniformly random. This property is convenient in some applications, e.g., anonymization protocols, as one of the signature elements can be revealed in the clear without leaking information on what was the original signature. So, as an additional contribution, in this paper we show a new re-randomizable SPS scheme that is strongly-optimal and improves over the re-randomizable scheme proposed in [7] by requiring one less online pairing. Then we take into consideration schemes secure against random-message attacks (RMA) for which there is no strongly-optimal candidate in the previous work. We fill this gap by showing a simple, strongly-optimal, RMA-secure SPS. We note that although random-message security is a weak notion, it has been shown useful in applications such as constructing adaptive oblivious transfer [22] and in a transformation for obtaining chosen-message secure SPS [1]. By using our strongly-optimal RMA-secure SPS scheme, all these applications can benefit of its improved efficiency.

AUTOMATED SYNTHESIS OF SPS. As emerges from the previous discussion, optimality results (at least in the single-dimensional form in which they have been developed so far) are insufficient in rich settings such as structure-preserving signatures where many meaningful measures of efficiency can be considered (e.g., verification time, key or signature size). Therefore, an attractive approach for achieving a broad range of optimality results is to perform an exhaustive search of valid SPS within user-defined parameters. In the second part of our work, we develop a synthesis tool that takes as input a user-defined budget, consisting of the number of pairings, group elements, etc. that can be used by the construction, and generates all possible expressions within this budget. Broadly speaking, our tool then uses an extension of the Generic Group Analyzer reported in [12], to generate, whenever it exists, a verification equation for the signature algorithm. Finally, our tool proves or disproves security of candidate schemes in the generic group model for the case when the adversary makes a bounded number of signing queries. Through this approach, we generate an exhaustive database, by exploring more than 2000 candidate SPS schemes, that can then be mined for different efficiency criteria. For instance, our database contains our new scheme with optimal number of online/offline pairings as well as the SPS schemes in the Type II setting that were previously proposed in [7]. Beyond its intrinsic interest, the database can also be used to validate or refute empirically new conjectures on SPS. For example, it is interesting to mention that our work on proving the new lower bounds was motivated by observing that among the schemes generated by our tool none of the ones secure against chosen-message attacks can be verified with only two pairings. More generally, our tool suggests the feasibility and interest to develop synthesis methods for structure-preserving cryptography. We believe that our methods can be extended to the Type I and Type III settings (however exhaustive search will be more difficult to attain because secure schemes must use two verification equations, which results in an exponential growth in the search space), and to other forms of structure-preserving cryptography, such as structure-preserving commitments [3] and encryption [14].

## 1.2 Other Related Work

STRUCTURE-PRESERVING SIGNATURES. While the notion of structure-preserving signature was first given by Abe et al. in [3], the first construction was proposed earlier by Groth [23], though its efficiency is far from being truly practical (it consists of hundreds of group elements). Green and Hohenberger [22] proposed SPS that are proved secure only against random-message attacks. Cathalo, Libert and Yung [15] constructed a scheme that is structure-preserving in a relaxed sense since it has a verification key which includes elements of the target group.

The study of lower bounds for SPS was put forward by Abe et al. [4] who showed that SPS in the Type III setting require at least three group elements in the signature and two pairing-product equations, and also proposed schemes matching these bounds that are only proven secure in the generic bilinear group model. Next, Abe et al. [5] refined the result in [4] considering schemes whose security can be proved under a non-interactive assumption using a black-box reduction. For this case they show that any scheme with only 3 elements in the signature cannot be proved secure under a non-interactive assumption. Optimal schemes in the symmetric (Type I) setting have been explored more recently by Abe et al. [8] who show that Type I SPS schemes require 3 elements in the signature and 2 verification equations (i.e., the same bounds as in Type III). Furthermore, the same work [8] proposes a general scheme that works in all three bilinear settings, and thus shows a Type II scheme with 3 elements in the signature and 2 verification equations. Finally, the recent work of Abe et al. [7] focused on the Type II setting and showed that in this setting the lower bounds are (surprisingly) different. Namely, Type II SPS schemes require 2 elements in the signature, a single verification equation, and 2 elements in the verification key (the latter being the first lower bound for the size of the verification key).

All the optimal schemes in [4,8,7] are proved secure directly in the generic bilinear group model. Another line of work investigated efficient SPS that can be proved secure under standard assumptions. Hofheinz and Jager [25] and Abe et al. [1,2] proposed schemes based on the decision linear assumption. The efficiency of these schemes, however, does not meet that of the schemes secure in the generic group model.

Finally, Chatterjee and Menezes [17] re-considered the result of Abe et al. [7] for Type II SPS in light of the current state-of-the-art implementations of Type II vs. Type III pairings. They start from the observation that implementations of Type III pairings are currently more efficient than the ones of Type II pairings. Then they note that Type II SPS, albeit optimal in terms of the number of signature elements and number of verification equations, are not as efficient as their Type III counterparts (i.e., the SPS schemes that can be obtained through a semi-generic transformation from Type II to Type III [16]).[4] Although this is a valid point when considering concrete efficiency, we believe that the exploration of Type II SPS is still quite interesting. For example, they have a simpler structure that leads to a smaller search space when looking for new schemes. Yet, given a Type II scheme, one can always translate it to the Type III setting if concrete efficiency is a concern, e.g., using the approach from [16].

COMPUTER-AIDED CRYPTOGRAPHY. In contrast to computer-aided tools for verifying cryptographic proofs, which have existed for some time, computer-aided tools for synthesizing new constructions are very recent. Barthe et al. [11] develop an automated tool, called ZooCrypt, for synthesizing padding-based encryption schemes; their tool uses a dedicated logic with an efficient proof search procedure to prove chosen-plaintext or chosen-ciphertext security, and an efficient method for finding attacks on insecure schemes. Because the search space for reasonably-sized constructions is small (about $10^6$ well-typed schemes), simple trimming techniques are sufficient to cover the full search space efficiently. Malozemoff, Katz, and Green [28] develop

[4] One main issue that leads to such difference of performance here is that testing group membership in $\mathbb{G}_2$ (which is required in the signature verification) is significantly more expensive in Type II than in Type III groups.

an automated tool for proving security of modes of operations; the security of candidate schemes is proved using a type system, but no attack is exhibited for insecure schemes.

Akinyele, Green, and Hohenberger [9] develop two synthesis tools for pairing-based cryptography. Their tool AutoGroup converts schemes in the Type I setting into schemes in the Type III setting, whereas their tool AutoStrong transforms an existentially unforgeable signature into a strongly unforgeable one, using SMT solvers to check whether the original signature satisfies a criterion allowing an efficient transformation. The idea of automatically transforming constructions from the symmetric to the asymmetric setting was further considered by Abe, Groth, Ohkubo and Tango [6], who develop an automated transformation of Type I protocols into Type III protocols.

Automating proofs in the generic group model has been recently considered by Barthe et al. [12] who propose a tool that enables to automatically analyze the validity of cryptographic assumptions in the generic (multilinear) group model. The tool developed in this work actually builds on the techniques of [12] in order to perform the security analysis of SPS in the generic bilinear group model.

Finally, in a concurrent and independent work, De Ruiter [18] recently proposed a tool for analyzing the security of structure-preserving signatures. The proposed tool provides a security analysis of SPS similar to the one we provide, even though the security arguments in [18] do not have a full formalization in the generic group model. Additionally, we note that our tool is not limited to the security analysis of SPS, but also includes the novel synthesis component which allows to automatically generate SPS schemes.

## 2 Preliminaries

### 2.1 Bilinear groups

A *bilinear group description* is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H)$ where $p$ is a prime number, $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cylic groups of order $p$, $G, H$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, $\psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ is the homomorphism sending $H$ to $G$ (so that $\psi(H^x) = G^x$ for all $x \in \mathbb{Z}$), and $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a nondegenerate bilinear pairing, meaning that $e(G, H)$ generates $\mathbb{G}_T$ and $e(G^x, H^y) = e(G, H)^{xy}$ for all $x, y \in \mathbb{Z}$.

A *bilinear group generator* $\mathcal{G}$ is an efficient algorithm which, on input of a security parameter $1^\lambda$, returns a bilinear group description $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H)$ with $p = 2^{\Omega(\lambda)}$ and efficient algorithms for computing group operations and the bilinear map $e$, and deciding equality of group elements and membership in the groups.

Furthermore, following Galbraith, Paterson and Smart [21], we say that the result is a Type I bilinear group if the homomorphism $\psi$ is efficiently computable and efficiently invertible (in which case we can simplify identify $\mathbb{G}_1$ and $\mathbb{G}_2$), a Type II bilinear group if $\psi$ is efficiently computable but not efficiently invertible (i.e. it is a one-way function) and a Type III bilinear group if $\psi$ is neither efficiently computable nor invertible. This paper mainly focuses on the Type II setting.

**Generic algorithms.** In a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H)$ generated by $\mathcal{G}$ we refer to deciding group membership, computing group operations in $\mathbb{G}_1$, $\mathbb{G}_2$ or $\mathbb{G}_T$, comparing group elements and evaluating the homomorphism or the bilinear map as the generic bilinear group operations. The signature schemes we construct only use generic bilinear group operations.

As is customary in the literature, we denote group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ by uppercase letters such as $M, R, S, V, W, \ldots$, and their discrete logarithms with respect to base $G$ or $H$ using the corresponding lowercase letters $m, r, s, v, w, \ldots$ In particular, for an element $X \in \mathbb{G}_2$, we have $\psi(X) = G^x$, and for

$(Y, Z) \in \mathbb{G}_1 \times \mathbb{G}_2$, we have $e(Y, Z) = e(G, H)^{yz}$. Furthermore, we will often express pairings equations, such as $e(X, Y)^{a_1} = e(W, Z)^{a_2} \cdot e(T, H)^{a_3}$, as a quadratic polynomial involving the corresponding exponents, i.e., $a_1 xy = a_2 wz + a_3 t$.

## 2.2 Structure-preserving signature schemes

We study structure-preserving signature schemes (SPS) [3] on bilinear groups generated by group generator $\mathcal{G}$. We refer to Appendix C.1 for basic definitions about signature schemes. In a structure preserving signature scheme the verification key, the messages and the signatures consist only of group elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ and the verification algorithm evaluates the signature by deciding group membership of elements in the signature, using the homomorphism $\psi$ and by evaluating pairing product equations, which are equations of the form:

$$\prod_i \prod_j e(X_i, Y_j)^{a_{ij}} = 1,$$

where $X_1, X_2, \ldots \in \mathbb{G}_1, Y_1, Y_2, \ldots \in \mathbb{G}_2$ are group elements appearing in *PP*, *VK*, $M$ and $\Sigma$ (where in the Type II setting it may hold $Y_i = \Psi(X_j)$ for some $i, j$) and the elements $a_{ij} \in \mathbb{Z}_p$ are constants stored in *PP*. More precisely:

**Definition 1 (Structure-preserving signatures).** *A signature scheme* (Setup, KeyGen, Sign, Verify) *is said to be structure-preserving with respect to some bilinear group generator* $\mathcal{G}$ *if*

- *PP consists of a bilinear group* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H)$ *generated by* $\mathcal{G}$ *and constants in* $\mathbb{Z}_p$*,*
- *the verification key consists of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
- *the messages consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
- *the signatures consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
- *the signing algorithm only uses generic group operations,*[5] *and*
- *the verification algorithm only needs to decide membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, use the homomorphism* $\psi$*, and evaluate pairing product equations.*

## 2.3 Known lower bounds on Type II SPS

A number of lower bounds on signature size, verification key size and the number of verification equations have been established for secure structure-preserving signature schemes and one-time signature schemes. In particular, Abe et al. [7] establish many such bounds in the Type II setting. As some of our results rely heavily on those bounds, we recall them below. Note that membership tests are not counted as "verification equations", although some of them may require an amortizable (aka offline) pairing computation in practical instantiations.

First, just as Type I and Type III SPS, Type II SPS for messages in $\mathbb{G}_1$ require two verification equations:

**Lemma 1 ([7, Theorem 3]).** *A structure-preserving signature scheme for messages in* $\mathbb{G}_1$ *must have at least two verification equations. This holds even for one-time signatures with security against random message attack.*

[5] Technically, this condition was not required in the original definition of Abe et al. [3], but all known constructions satisfy this property and it is required for the proofs of most lower bounds to go through. Since an SPS scheme with a non-generic signing algorithm would be very unnatural and surprising, it seems appropriate to include genericity of the signer in the definition (see also the discussion in [7, §2.3]).

Since this paper will mostly focus on schemes with a single verification equation, we will therefore consider signatures on messages in $\mathbb{G}_2$, which can have a single verification equation. In that case, Abe et al. obtain a lower bound on verification key size, and show that all signature elements must be in $\mathbb{G}_2$.

**Lemma 2 ([7, Theorem 4 and Lemma 1]).** *A structure-preserving signature scheme with a single verification equation must have at least two group elements in the verification key, and can have no non-redundant signature element in $\mathbb{G}_1$. This holds even for one-time signatures secure under random message attack.*

Finally, signatures in a secure Type II SPS scheme must consist of at least two elements (although that property does not hold for one-time signatures), and three elements for messages in $\mathbb{G}_1$ (idem).

**Lemma 3 ([7, Theorem 5]).** *An EUF-RMA-secure structure-preserving signature scheme must have at least $2$ group elements for messages in $\mathbb{G}_2$ and at least $3$ group elements for messages in $\mathbb{G}_1$.*

## 3 Lower Bounds on the Number of Pairings in the Type II Setting

In this section we show lower bounds for the number of pairings in the pairing-product verification equations of SPS in the Type II setting. In particular, in our analysis we consider SPS schemes that already match the lower bounds shown in [7], i.e., they have 2 group elements in the verification key, 2 group elements in the signature and the verification consists of a single pairing-product equation (as well as possible group membership tests).

To have a more refined and practically interesting analysis, we distinguish between pairings according to whether they can be precomputed from the public key or not. In the former case we call a pairing *offline* while in the latter case *online*.

### 3.1 Main result

Having defined the notion of online and offline pairings, we are now ready to state our main result. It shows that any optimal-size SPS scheme requires at least three pairings for verification, and two of these pairings must be online ones.

**Theorem 1 (Main result).** *Any EUF-CMA-secure structure-preserving signature scheme in the Type II setting with $1$ verification pairing-product equation, $2$ group elements in the verification key and $2$ elements in the signature requires at least $3$ pairings in the pairing-product equation, and at least $2$ of them must be online pairings.*

To prove the theorem, we distinguish between three cases according to which groups the two elements $V, W$ of the verification key belong to, i.e., (i) $V, W \in \mathbb{G}_2$, (ii) $V, W \in \mathbb{G}_1$, and (iii) $V \in \mathbb{G}_1, W \in \mathbb{G}_2$.

The first case is rather simple and is addressed in the following lemma which shows that there exists no such structure-preserving signature scheme.

**Lemma 4.** *There is no secure structure-preserving signature scheme in the Type II setting with a single verification equation and a verification key consisting entirely of elements of $\mathbb{G}_2$.*

*Proof.* We know by Lemma 2 and Lemma 3 that the signatures and messages all have to be in $\mathbb{G}_2$. Since all inputs are in $\mathbb{G}_2$ the scheme would also be secure in the Type I setting and must therefore be insecure since Type I SPS require two pairing product equations for security [8, Theorem 4]. $\square$

The second case is somewhat more involved, and mainly addressed by the following lemma, proved in Section 3.3 below.

**Lemma 5.** *An* EUF-CMA*-secure structure-preserving signature scheme in the Type II setting with* $1$ *verification pairing-product equation,* $2$ *group elements* $V, W \in \mathbb{G}_1$ *in the verification key and* $2$ *group elements in the signature requires at least* $3$ *pairings in the pairing-product equation.*

That result establishes that 3 pairings are needed, so all that remains to show is that 2 of them must be online. This follows immediately from the following observation.

**Lemma 6.** *In a Type II structure-preserving signature scheme where all verification key elements are in* $\mathbb{G}_1$*, it is possible to compute all the offline pairings in a verification pairing-product equation using a single pairing evaluation. More generally,* $\ell + 1$ *pairing evaluations are sufficient if the verification key contains* $\ell$ *elements in* $\mathbb{G}_2$*.*

*Proof.* Indeed, if the verification key is $(V_1, \dots, V_k, U_1, \dots, U_\ell) \in \mathbb{G}_1^k \times \mathbb{G}_2^\ell$, then any product of offline pairings can be expanded into an expression of the form $\prod_{i,j} e(X_i, Y_j)^{c_{ij}}$ where $Y_j$ runs through $H, U_1, \dots, U_\ell$ (since these are the only elements of $\mathbb{G}_2$ in the verification key and the public parameters) and $X_i$ runs through $G, V_1, \dots, V_k, \psi(U_1), \dots, \psi(U_\ell)$. By rewriting the product as:

$$\prod_j e\Big(\prod_i X_i^{c_{ij}}, Y_j\Big)$$

we can compute it with at most $\ell + 1$ pairing evaluations as required. □

Finally, to complete the proof of Theorem 1, we only need to prove it when the verification key consists of one element of $\mathbb{G}_1$ and one element of $\mathbb{G}_2$. This case, which is somewhat less interesting as such a scheme is less space efficient than when all key elements are in $\mathbb{G}_1$, but turns out to be more technically challenging, is dealt with in details in Appendix A.

## 3.2 Gaps in Bounds Between EUF-RMA and EUF-CMA-Security

Following Lemma 5, in the setting when $(V, W) \in \mathbb{G}_1^2$ the bound of Theorem 1 holds only for EUF-CMA-secure SPS schemes. This however is not the case in the setting when the verification key is of the form $(V, W) \in \mathbb{G}_1 \times \mathbb{G}_2$: as discussed in Appendix A, the bound of Theorem 1 holds even for EUF-RMA-secure SPS schemes. In what follows we establish a slightly weaker general lower bound on the number of pairings in the single pairing-product equation of minimal EUF-RMA-secure SPS schemes in the setting when $(V, W) \in \mathbb{G}_1^2$.

**Theorem 2.** *Any* EUF-RMA*-secure structure-preserving signature scheme in the Type II setting with* $1$ *verification pairing-product equation requires at least* $2$ *pairings in the pairing-product equation, and both of them must be online pairings.*

*Proof.* It suffices to show that a Type II SPS scheme with a single verification equation (and which we can assume without loss of generality signs one-element messages) cannot be EUF-RMA-secure if the pairing-product equation consists of only one online pairing (and any number of offline pairings).

To see this, denote by $(S_1, \dots, S_k)$ the signature vector (which is in $\mathbb{G}_2^k$ without loss of generality by Lemma 2), and observe that the pairing product equation must be of the form:

$$\prod_{i,j} e(X_i, Y_j) = e\Big(\psi(M)^{a_0} \cdot \prod_{i=1}^{k} \psi(S_i)^{a_i} \cdot Z, M^{b_0} \cdot \prod_{j=1}^{k} S_j^{b_j} \cdot T\Big)$$

where the pairings on the left-hand side are offline (and hence the $X_i$'s and $Y_j$'s do not depend on the message or the signature), and $Z, T$ are elements which also do not depend on the message or the signature. But then we can do the change of variables:

$$(R', S') = \left(\psi(M)^{a_0} \cdot \prod_{i=1}^{k} \psi(S_i)^{a_i}, M^{b_0} \cdot \prod_{j=1}^{k} S_j^{b_j}\right)$$

and then $(R', S')$ provides a two-element EUF-RMA-secure signature scheme whose verification equation is just:

$$\prod_{i,j} e(X_i, Y_j) = e(R' \cdot Z, S' \cdot T),$$

and in particular does not depend on the message: this is a contradiction. □

We see that the bounds given by Theorem 1 and Theorem 2 show a gap. Namely, there could exist an EUF-RMA-secure scheme with precisely two online pairings and no offline pairing. We confirm that both lower bounds are indeed tight, by providing an EUF-RMA-secure SPS with precisely two online pairings in Section 5.2.

## 3.3 Proof of Lemma 5

*Proof.* The proof proceeds by contradiction showing that having a scheme with only 2 pairings in the single pairing-product equation is impossible. Let us first recall that in this setting we have a message $M \in \mathbb{G}_2$, verification keys $V, W \in \mathbb{G}_1$ and signature elements $R, S \in \mathbb{G}_2$. As usual, we denote their discrete logarithms by the corresponding lower case letters. We may write the general verification equation in terms of the discrete logarithms of $M, R, S, V, W$ as follows:

$$\begin{aligned}(c_1m + c_2r + c_3s + c_4v + v_5w + c_6)(d_1m + d_2r + d_3s + d_4) = \\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_3s + f_4),\end{aligned} \tag{1}$$

where the products represent a pairing, the left factor in each product represent the element in $\mathbb{G}_1$ and the right factor the element in $\mathbb{G}_2$ of each pairing.

Now if we define the vectors $X_1 = (c_1, \ldots, c_6), X_2 = (e_1, \ldots, e_6), Y_1 = (d_1, \ldots, d_4), Y_2 = (f_1, \ldots, f_4)$ over $\mathbb{Z}_p$ and the matrix:

$$E = X_1^t Y_1 - X_2^t Y_2,$$

then the verification equation (1) can be rewritten as

$$(m, r, s, v, w, 1) \cdot E \cdot (m, r, s, 1)^t = 0.$$

A simple observation shows that if $\operatorname{Ker} E$ contains a vector $(x_1, \ldots, x_4)$, where $x_4 \neq 0$, then we may scale to $x_4 = 1$ and then

$$m = x_1,\ r = x_2,\ s = x_3$$

is a valid key-only attack forgery (since the kernel of $E$ can be computed entirely from the public parameters). It follows that if the scheme is secure, then $\operatorname{Ker} E \subset \{(x_1, \ldots, x_4) \mid x_4 = 0\}$. However, this implies that the following system

$$\begin{cases} d_1m + d_2r + d_3s = -d_4 \\ f_1m + f_2r + f_3s = -f_4 \end{cases}$$

lacks a solution. Up to exchanging the roles of $Y_1$ and $Y_2$ without loss of generality, this implies that $Y_1 = cY_2 + (0,0,0,\lambda)$ for some constants $c, \lambda$, and hence:

$$E = X_1^t Y_1 - X_2^t Y_2 = X_1^t\big(cY_2 + (0,0,0,\lambda)\big) - X_2^t Y_2 = (\lambda X_1)^t \cdot (0,0,0,1) - (X_2 - cX_1)^t Y_2.$$

Therefore, after relabeling the coefficients, we may assume that $Y_1 = (0,0,0,1)$ and the verification equation (1) can be rewritten as

$$\begin{gathered} c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6 = \\ (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_1 m + f_2 r + f_3 s + f_4), \end{gathered} \tag{2}$$

Now if $(c_4, c_5) = \lambda(e_4, e_5)$ or $(e_4, e_5) = \lambda(c_4, c_5)$, then we may replace the verification key by a single element $t = e_4 v + e_5 w$ or $t = c_4 v + c_5 w$, which is insecure by Lemma 2. It follows that

$$\det \begin{pmatrix} c_4 & c_5 \\ e_4 & e_5 \end{pmatrix} \neq 0$$

and we may do a linear change of variables

$$\begin{pmatrix} v' \\ w' \end{pmatrix} = \begin{pmatrix} c_4 & c_5 \\ e_4 & e_5 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} + \begin{pmatrix} c_6 \\ e_6 \end{pmatrix},$$

so that the verification equation (2) becomes, after renaming coefficients,

$$c_1 m + c_2 r + c_3 s + v = (e_1 m + e_2 r + e_3 s + w)(f_1 m + f_2 r + f_3 s + f_4). \tag{3}$$

Note that the vectors $(c_2, c_3), (e_2, e_3), (f_2, f_3)$ cannot all be collinear, because otherwise we may again compress the signature into one group element, which we already know is impossible.

Next, we look at the matrix

$$N = \begin{pmatrix} e_2 & e_3 \\ f_2 & f_3 \end{pmatrix}$$

and distinguish two cases depending on the determinant.

On one hand, if $\det N \neq 0$, then as before, a change of variables let us write the verification equation (3) in the form

$$c_1 m + c_2 r + c_3 s + v = (r + w)s.$$

Since $m$ must be used in the verification equation, we know that $c_1 \neq 0$. An easy calculation then shows that if $(m, r, s)$ is a triple satisfying the verification equation for the keys $v, w$, then so does $(m - (c_2 - s)/c_1, r + 1, s)$. This gives us a forgery unless $c_2 = s$ for a non-negligible set of signatures. However, if this happens, then $s$ would be a redundant signature element. From Lemma 3 we know that the scheme must be insecure.

On the other hand, if $\det N = 0$, we have the two cases $(e_2, e_3) = \lambda(f_2, f_3)$ or $(f_2, f_3) = 0$. If $(f_2, f_3) = 0$, then

$$\det \begin{pmatrix} c_2 & c_3 \\ e_2 & e_3 \end{pmatrix} \neq 0$$

or otherwise $(c_2, c_3), (e_2, e_3), (f_2, f_3)$ would be collinear. It follows that the verification equation (3) reduces to

$$r + v = (s + w)(f_1 m + f_4).$$

map $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. iso $\mathbb{G}_2 \to \mathbb{G}_1$.

input $[V]$ in $\mathbb{G}_1$. input $[W]$ in $\mathbb{G}_2$.

oracle $o(M : \mathbb{G}_2) =$ sample $R$; return $[R, (1 + W^2 + M * V) * R^{-1}]$ in $\mathbb{G}_2$.

win $(M' : \mathbb{G}_2, R' : \mathbb{G}_2, S' : \mathbb{G}_2) = (S' * R' = 1 + W^2 + V * M' \wedge \forall i : M' \neq M_i)$.

**Fig. 1.** Example of input for analyzing EUF-CMA security of an SPS scheme using our extended version of the GGA tool.

and since $f_1 \neq 0$, as the message must be used, we may query $m_1 = -f_1^{-1} f_4$ getting back a signature $(r_1, s_1)$, where $r_1 = -v$. Now make another query with $m_2 = f_1^{-1}(1 - f_4)$ to get back a signature $(r_2, s_2)$. Then

$$r_2 + v = s_2 + w \Rightarrow w = r_2 + v - s_2 = r_2 - r_1 - s_2,$$

so with two *chosen-message* queries the attacker can transfer $V, W$ to $\mathbb{G}_2$ and then we know the scheme cannot be secure (concretely, $(R_1, R_1 R_2^{-1} S_2) = (V^{-1}, W^{-1})$ is a valid signature on any message). Therefore, we must have that $(e_2, e_3) = \lambda(f_2, f_3)$. Again using the fact that $(c_2, c_3), (e_2, e_3), (f_2, f_3)$ are not collinear, we must have

$$\det \begin{pmatrix} c_2 & c_3 \\ f_2 & f_3 \end{pmatrix} \neq 0.$$

It follows that we may do a change of variables $s' = f_1 m + f_2 r + f_3 s + f_4$ and $r' = c_1 m + c_2 r + c_3 s$ and by the collinearity of $(e_2, e_3)$ and $(f_2, f_3)$ the verification equation (3) becomes of the form

$$r + v = (e_1 m + e_3 s + w + e_6) s$$

and now if $(m, r, s)$ is a valid signature, then so is $(m + 1/e_1, r + s, s)$, which is a valid forgery, since $m$ must be used in the verification equation and hence $e_1 \neq 0$. Also, note that in the latter case, the attack can be performed in the random-message security game. □

## 4 Synthesis of Schemes

Our tool[6] for the synthesis of SPS schemes consists of two components. The first component takes the description of a search space and generates all included SPS schemes. The second component classifies a given scheme by performing a proof and attack search.

### 4.1 Generation of Schemes

For our generation algorithm, we consider SPS schemes with generic *KeyGen* and *Sign* algorithms and assume all random values are sampled uniformly.

Our definition of an SPS scheme consists of

- the employed group type and the supported message space $\mathbb{G}_1^k \times \mathbb{G}_2^l$,
- the randomly sampled values $u_i \in \mathbb{Z}_p$ used in *KeyGen*,
- the verification keys $V_i = G^{f_i(\boldsymbol{u})} \in \mathbb{G}_1$ and $W_i = H^{g_i(\boldsymbol{u})} \in \mathbb{G}_2$,
- the randomly sampled values $r_i \in \mathbb{Z}_p$ used in *Sign*,

[6] available at `https://www.easycrypt.info/GGA`

– the signature elements $S_i = G^{s_i(\boldsymbol{u},\boldsymbol{r},\boldsymbol{m})} \in \mathbb{G}_1$ and $T_i = H^{t_i(\boldsymbol{u},\boldsymbol{r},\boldsymbol{m})} \in \mathbb{G}_2$, and
– the pairing-product equations used by *Verify*.

Here, $f_i$ and $g_i$ are arbitrary rational functions in the random variables $\boldsymbol{u}$. Similarly, $s_i$ and $t_i$ are rational functions in the random variables $\boldsymbol{u}, \boldsymbol{r}$ and the discrete logarithms $\boldsymbol{m}$ of the messages such that there exists a corresponding generic signing algorithm, i.e., $S_i$ and $T_i$ can be computed without knowing the discrete logarithms of the messages.

A search space description characterizes a finite set of SPS schemes and consists of (1) the group type, (2) the number of messages, verification key elements, and signature elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, (3) the number of random values sampled in *KeyGen* and *Sign*, and (4) a description of the rational expressions that can be used for $f_i$, $g_i$, $s_i$, and $t_i$.

There are two ways to characterize the allowed rational expressions. First, the tool can take a set of Laurent polynomials with placeholders and allowed values for these placeholders, and generate all instances. Second, the tool accepts a set of constraints that specify bounds on the number of additions, the size of coefficients, and the degree of monomials. Then, it generates all Laurent polynomials that satisfy these constraints.

Given a search space description and concrete polynomials for the verification keys and the signature elements, the tool can compute the (strongest) verification equation as follows. Using distinct variables $Z_1, Z_2, \ldots$ for all group elements in the verification keys, signature elements, and messages, enumerate all products over these variables that can be computed by applying the homomorphisms and the bilinear map. This yields a sequence of monomials $M_1, M_2, \ldots$ over the the variables $Z_i$ denoting products in $\mathbb{G}_T$ that can be computed from the input of the verification algorithm using $\Psi$ and $e$. To characterize the linear relations between the elements in $\mathbb{G}_T$ corresponding to the monomials $M_i$, we associate a rational expression $F_i$ over $\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{m}$ to $M_i$ by evaluating the monomial for $Z_i := h_i(\boldsymbol{u}, \boldsymbol{r}, \boldsymbol{m})$ where $h_i$ is the exponent of the group element associated with $Z_i$. Finally, we use linear algebra to compute a basis of the linear relations between the $F_i$ and map them back to verification equations using $M_i$.

## 4.2 Proof and Attack Search

We classify generated schemes using a proof and attack search based on an extension of the generic group analyzer developed by Barthe et al. [12]. The generic group analyzer (GGA) is a tool that automatically analyzes cryptographic assumptions in generic group models. To analyze SPS schemes, we use GGA's support for the generic bilinear group model. Here, the adversary is given blackbox access (using handles) to elements in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ and provided with oracles for performing the group operations and applying the bilinear map and the efficiently computable homomorphisms. The GGA tool also supports a restricted class of interactive assumptions that enable the analysis of signature schemes that sign messages in $\mathbb{Z}_p$, but does not support oracles that take group elements. To analyze such interactive assumptions, the GGA tool exploits that the signing oracle queries are essentially non-adaptive. More concretely, since the signing oracle takes elements in $\mathbb{Z}_p$ and returns handles to group elements, the adversary can only use these returned handles to compute the forgery, but the arguments to signing oracle queries cannot depend on the results of earlier oracle queries. This allows the GGA tool to treat oracle return values like initially known values by using parameters to model the oracle arguments in $\mathbb{Z}_p$. Another reason why the GGA cannot be directly applied to most SPS schemes is that there is no support for Laurent polynomials, which are required to model signing algorithms that invert elements of $\mathbb{Z}_p$.

To overcome these limitations, we have extended the GGA tool with support for both features and our extension can now analyze assumptions, such as the one shown in Figure 1, that were out of scope of the original version. To support signing oracles that take handles to group elements, the adversary knowledge

| Search Space | | Schemes | | Results (for equiv. classes) | | |
|---|---|---|---|---|---|---|
| Verification equation | First signature element | total | equivalence classes | Noverif | Attack | Proof |
| $s = f(t, v, w, m)$ | $T = H^r$ for random $r$ | 212 | 57 | 0 | 55 | 1 |
| $s\,t = f(t, v, w, m)$ | $T = H^r$ for random $r$ | 224 | 67 | 0 | 55 | 12 |
| $s\,(t - w) = f(t, v, w, m)$ | $T = H^{r+w}$ for random $r$ | 1344 | 774 | 651 | 103 | 14 |
| $s\,w = f(t, v, w, m)$ | $T = H^r$ for random $r$ | 224 | 126 | 0 | 120 | 3 |
| | | 2004 | 1024 | 651 | 333 | 30 |

**Table 1.** Synthesis results for Type II with keys $V, W \in \mathbb{G}_1$, message $M \in \mathbb{G}_2$, and signature $T, S \in \mathbb{G}_2$. The value $r$ in the exponent of $T$ is always chosen as a random element in $\mathbb{Z}_p$.

can contain polynomials with parameters that are used to model oracle arguments in $\mathbb{Z}_p$ and in the groups $\mathbb{G}_*$. The parameters introduced to model known group elements correspond to coefficients of linear combinations, i.e., we exploit that every known group element is a linear combination of initially known group elements, group elements returned by earlier oracle queries, or the result of applying a pairing or an isomorphism to such group elements. To compute a basis for all known group elements after $q$ oracle queries, we recursively extend the knowledge after $i$ queries with the results of an additional query, starting with the initial knowledge.

The definition in Figure 1 specifies the EUF-CMA security experiment for an SPS scheme in the Type II setting. The verification keys are specified in the second line, the signing algorithm is given in the third line, and the winning condition (including the verification equation) is given in the last line. Here, group elements are specified by giving their exponent polynomials and the variables $V$ and $W$ are assumed to be randomly sampled. For such an input and a bound on the number $q$ of performed oracle queries, the tool either returns an attack or a proof that the scheme is $q$-EUF-CMA secure in the generic group model. We note that our tool can be invoked with any specified value of $q$, though in this specific setting of SPS it runs efficiently only with small values.

## 5 Synthesized Schemes

In this section we will present and discuss the SPS schemes that we obtained by searching using our tool described in Section 4.

### 5.1 A Summary of Our Search

We have performed an exhaustive search for Type II schemes with keys $V, W \in \mathbb{G}_1$, message $M \in \mathbb{G}_2$, and signature $T, S \in \mathbb{G}_2$ such that: $V$ and $W$ are random; $T = H^r \cdot U$ where $r$ is random and $U$ does not involve $r$; the exponent polynomials of $S$, i.e., $s(r, v, w, m)$, have coefficients in $\{0, 1\}$. The results of our search are presented in Table 1. We use "Proof" to denote that our tool could prove at least 2-EUF-CMA security. Among the SPS schemes that are found, we identify equivalent schemes according to the following notion: we say two schemes $\Sigma$ and $\Sigma'$ are in the same equivalence class if $\Sigma$ can be obtained from $\Sigma'$ by applying invertible affine transformations to the verification keys, the messages, and the signature elements. This implies the existence of reductions from the security of $\Sigma$ to the security of $\Sigma'$ and vice-versa. As a simple example, consider a scheme that is obtained from another scheme by first multiplying the message $M$ with $G$ and then applying the original signing algorithm. Overall, except for 10 of the 1024 analyzed schemes, our tool either finds an attack, proves at least 2-EUF-CMA security, or proves that there is no verification equation. For the 10 schemes, the tool either returned unknown or timed out[7].

## 5.2 New SPS Schemes

Among the schemes that we found using our tool, we highlight two of them that are of particular interest, as well as a counterpart of the first one in the Type III setting.

**A Strongly-Optimal Randomizable SPS.** The first scheme is an SPS which is randomizable and matches the lower bound of Theorem 1, i.e., it can be verified using one offline and two online pairings. This scheme improves over the previously known randomizable schemes (in particular over the one recently proposed in [7]) as the latter requires three online pairings. This new scheme is presented in Fig. 2, and its security, stated as follows, is proved in Appendix B.1.

**Theorem 3.** *The signature scheme in Fig. 2 is EUF-CMA-secure in the generic bilinear group model.*

This scheme can be translated to Type III groups using the transformation in [16], which essentially consists in "duplicating" $R$, i.e., to give $R = G^r \in \mathbb{G}_1$ and $T = H^r \in \mathbb{G}_2$, and adding a pairing-product equation to check that $R, T$ have the same discrete logarithm, i.e., $e(R, H) = e(G, T)$. Such transformed scheme however requires one offline and four online pairings in the pairing-product equations. In what follows we propose a slightly different way to transform our scheme in the Type III setting which yields a solution requiring only three online pairings. The basic idea is that in the previous transformation $T$ is not used in the first pairing-product equation, and its utility is to force the adversary to show that it knows the discrete log of $R$ (or obtained $(R, T)$ by applying a linear operation on a pair received by the challenger). We obtain the same functionality by letting the signer compute $T = H^{1/r}$. This allows us to test "equality of $r$ between $R$ and $T$" by checking $e(R, T) = e(G, H)$. The last pairing, however, involves only the generators and can thus be computed offline. A precise description of the resulting scheme is provided in Fig. 3, and the security result, stated below, is proved in Appendix B.2.

**Theorem 4.** *The signature scheme in Fig. 3 is EUF-CMA-secure in the generic bilinear group model.*

**A Strongly-Optimal RMA-Secure SPS.** Our second new SPS scheme, presented in Fig. 4, is secure against random-message attacks, and achieves the lower bound of only two pairings in the pairing-product equation (both necessarily online) for EUF-RMA-secure schemes, as stated in Theorem 2. In particular, it *beats* the lower bound of Theorem 1 that holds for EUF-CMA-secure schemes.

This scheme is also perfectly randomizable, with the simple randomization algorithm that sends a signature $(R, S)$ on $M$ to $(R \cdot H^t, S \cdot M^t)$ for some uniformly random $t$.

As an interesting note, we observe that the verification equation of this scheme is exactly the only possible one, according to our impossibility proof. Indeed, while our Lemma 5 holds for SPS schemes that are EUF-CMA-secure, the actual proof relies on EUF-RMA-security in all cases but one. For that particular case, in which we show a chosen-message attack, the verification equation is of the form $s + w = (r + v)(f_1 m + f_4)$ for some constants $f_1, f_4$, up to invertible linear transformations on the verification key and signature elements.

The security of this scheme, stated below, is proved in Appendix B.3.

**Theorem 5.** *The signature scheme in Fig. 4 is EUF-RMA-secure in the generic bilinear group model.*

[7] we used a timeout of 30 seconds

***Setup***$(1^k)$**:** return $PP = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H) \leftarrow \mathcal{G}(1^k)$.
***KeyGen***$(PP)$**:** choose random $v, w \leftarrow \mathbb{Z}_p$ and return $VK = (V, W)$, $SK = (v, w)$ where $V = G^v$ and $W = G^w$.
***Sign***$(PP, SK, M)$**:** given $M \in \mathbb{G}_2$, choose a random $r \leftarrow \mathbb{Z}_p^*$ and return $(R, S)$ where $R = H^r$ and $S = (M^v H^w)^{1/r}$.
***Rerand***$(PP, VK, M, (R, S))$**:** pick a random $\alpha \leftarrow \mathbb{Z}_p^*$ and compute a randomized signature $(R', S')$ as $R' = R^\alpha$ and $S' = S^{1/\alpha}$.
***Verify***$(PP, VK, M, (R, S))$**:** accept if and only if $M, R, S \in \mathbb{G}_2$ and

$$e(\psi(R), S) = e(V, M) \cdot e(W, H).$$

**Fig. 2.** Our strongly-optimal re-randomizable SPS.

***Setup***$(1^k)$**:** return $PP = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^k)$.
***KeyGen***$(PP)$**:** choose random $v, w \leftarrow \mathbb{Z}_p$ and return $VK = (V, W)$, $SK = (v, w)$ where $V = G^v$ and $W = G^w$.
***Sign***$(PP, SK, M)$**:** given $M \in \mathbb{G}_2$, choose a random $r \leftarrow \mathbb{Z}_p^*$ and return $(R, T, S) \in \mathbb{G}_1 \times \mathbb{G}_2^2$ where $R = G^r$, $T = H^{1/r}$ and $S = (M^v H^w)^{1/r}$.
***Rerand***$(PP, VK, M, (R, T, S))$**:** pick a random $\alpha \leftarrow \mathbb{Z}_p^*$ and compute a randomized signature $(R', T', S')$ as $R' = R^\alpha$, $T' = T^{1/\alpha}$ and $S' = S^{1/\alpha}$.
***Verify***$(PP, VK, M, (R, T, S))$**:** accept if and only if $R \in \mathbb{G}_1$, $M, T, S \in \mathbb{G}_2$ and

$$e(R, S) = e(V, M) \cdot e(W, H) \quad \text{and} \quad e(R, T) = e(G, H).$$

**Fig. 3.** A re-randomizable SPS in Type III groups.

***Setup***$(1^k)$**:** return $PP = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \psi, G, H) \leftarrow \mathcal{G}(1^k)$.
***KeyGen***$(PP)$**:** choose random $v, w \leftarrow \mathbb{Z}_p$ and return $VK = (V, W)$, $SK = (v, w)$ where $V = G^v$ and $W = G^w$.
***Sign***$(PP, SK, M)$**:** given $M \in \mathbb{G}_2$, choose a random $r \leftarrow \mathbb{Z}_p$ and return $(R, S)$ where $R = H^r$ and $S = M^{r+v} H^{-w}$.
***Verify***$(PP, VK, M, (R, S))$**:** accept if and only if $M, R, S \in \mathbb{G}_2$ and

$$e(\psi(S) \cdot W, H) = e(R \cdot V, M).$$

**Fig. 4.** Our RMA-secure SPS with two pairings.

## 6 Conclusion

In this work, we considered a new measure for the efficiency of SPS, that is, the number of pairings required in the verification equation. With respect to this measure, we proved lower bounds and proposed new schemes matching these bounds in the Type II setting. In order to find schemes, we built a synthesis tool that automates the generation and security analysis of SPS. Currently, our methods only support security proofs with respect to a bounded number of signing queries. Developing new methods that support security proofs with respect to an unbounded number of queries for SPS is an interesting open problem that requires new techniques. Another direction left open by our work is to obtain similar results for the Type I and Type III setting. In contrast to the Type II setting, the Type I and Type III settings are more complex since they need more than one verification equation [4,8]. For synthesis, this causes a significant blowup of the search space, while for the minimality result, our proofs exploit that there is exactly one equation.

**Acknowledgements.** This work is supported in part by ONR grant N00014-12-1-0914, Madrid regional project S2009TIC-1465 PROMETIDOS, and Spanish projects TIN2009-14599 DESAFIOS 10 and TIN2012-39391-C04-01 Strongsoft. Additional support for Scedrov and Fagerholm is from the AFOSR MURI "Science of Cyber Security: Modeling, Composition, and Measurement" and from NSF Grant CNS-0830949. The research of Fiore and Schmidt has received funds from the European Commissions Seventh Framework Programme Marie Curie Cofund Action AMAROUT II (grant no. 291803).

## References

1. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In X. Wang and K. Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 4–24. Springer, Dec. 2012.
2. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013: 16th International Workshop on Theory and Practice in Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 312–331. Springer, Feb. / Mar. 2013.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236. Springer, Aug. 2010.
4. M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 649–666. Springer, Aug. 2011.
5. M. Abe, J. Groth, and M. Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In D. H. Lee and X. Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 628–646. Springer, Dec. 2011.
6. M. Abe, J. Groth, M. Ohkubo, and T. Tango. Converting cryptographic schemes from symmetric to asymmetric bilinear groups. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 241–260. Springer, Aug. 2014.
7. M. Abe, J. Groth, M. Ohkubo, and M. Tibouchi. Structure-preserving signatures from type II pairings. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 390–407. Springer, Aug. 2014.
8. M. Abe, J. Groth, M. Ohkubo, and M. Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 688–712. Springer, Feb. 2014.
9. J. A. Akinyele, M. Green, and S. Hohenberger. Using SMT solvers to automate design tasks for encryption and signature schemes. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 399–410. ACM Press, Nov. 2013.
10. N. Attrapadung, B. Libert, and T. Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013: 16th International Workshop on Theory and Practice in Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 386–404. Springer, Feb. / Mar. 2013.

11. G. Barthe, J. M. Crespo, B. Grégoire, C. Kunz, Y. Lakhnech, B. Schmidt, and S. Zanella Béguelin. Fully automated analysis of padding-based encryption in the computational model. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 1247–1260. ACM Press, Nov. 2013.
12. G. Barthe, E. Fagerholm, D. Fiore, J. C. Mitchell, A. Scedrov, and B. Schmidt. Automated analysis of cryptographic assumptions in generic group models. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 95–112. Springer, Aug. 2014.
13. J. Camenisch, M. Dubovitskaya, R. R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. In I. Visconti and R. D. Prisco, editors, *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 559–579. Springer, Sept. 2012.
14. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, and V. Naessens. Structure preserving CCA secure encryption and applications. In D. H. Lee and X. Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 89–106. Springer, Dec. 2011.
15. J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 179–196. Springer, Dec. 2009.
16. S. Chatterjee and A. Menezes. On cryptographic protocols employing asymmetric pairings - the role of $\Psi$ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.
17. S. Chatterjee and A. Menezes. Type 2 structure-preserving signature schemes revisited. Cryptology ePrint Archive, Report 2014/635, 2014. `http://eprint.iacr.org/2014/635`.
18. J. de Ruiter. Automated algebraic analysis of structure-preserving signature schemes. Cryptology ePrint Archive, Report 2014/590, 2014. `http://eprint.iacr.org/2014/590`.
19. R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.
20. G. Fuchsbauer and D. Vergnaud. Fair blind signatures without random oracles. In D. J. Bernstein and T. Lange, editors, *AFRICACRYPT 10: 3rd International Conference on Cryptology in Africa*, volume 6055 of *Lecture Notes in Computer Science*, pages 16–33. Springer, May 2010.
21. S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16):3113–3121, Sept. 2008.
22. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In J. Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 179–197. Springer, Dec. 2008.
23. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, Dec. 2006.
24. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, Apr. 2008.
25. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 590–607. Springer, Aug. 2012.
26. B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 289–307. Springer, Aug. 2013.
27. B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 571–589. Springer, Aug. 2012.
28. A. J. Malozemoff, J. Katz, and M. D. Green. Automated analysis and synthesis of block-cipher modes of operation. In *CSF 2014*, 2014.
29. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.
30. R. Zippel. Probabilistic algorithms for sparse polynomials. In E. W. Ng, editor, *EUROSM '79*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.

## A SPS Schemes With Verification Key $(V, W) \in \mathbb{G}_1 \times \mathbb{G}_2$

This appendix is devoted to the analysis of minimal Type II SPS schemes whose verification key is of the form $(V, W) \in \mathbb{G}_1 \times \mathbb{G}_2$. As mentioned in Section 3.2, Theorem 1 holds in this case even for EUF-RMA-secure schemes. This follows directly from Theorem 2 together with the following lemma.

**Lemma 7.** *An EUF-RMA-secure structure-preserving signature scheme in the Type II setting with* $1$ *verification pairing-product equation,* $2$ *group elements* $V \in \mathbb{G}_1$ *and* $W \in \mathbb{G}_2$ *in the verification key and* $2$ *group elements in the signature requires at least* $3$ *pairings in the pairing-product equation.*

*Proof.* We can write the verification equation in the form:

$$\begin{aligned}(c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6)(d_1 m + d_2 r + d_3 s + d_4 w + d_5) = \\ (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_1 m + f_2 r + f_3 s + f_4 w + f_5).\end{aligned} \tag{4}$$

Our proof is a large case distinction based on the values of the parameters. The following Figure 5 illustrates the case distinction tree. We label the cases according to the tree and assume the reader follows the case distinctions by looking at the tree nodes.

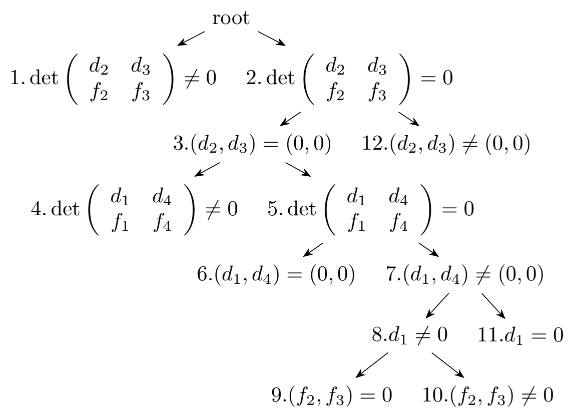
$$\begin{tikzcd}
& \text{root} \arrow[dl] \arrow[dr] & & & \\
1.\det\begin{pmatrix} d_2 & d_3 \\ f_2 & f_3 \end{pmatrix} \neq 0 & & 2.\det\begin{pmatrix} d_2 & d_3 \\ f_2 & f_3 \end{pmatrix} = 0 \arrow[dl] \arrow[dr] & & \\
& 3.(d_2, d_3) = (0,0) \arrow[dl] \arrow[dr] & & 12.(d_2, d_3) \neq (0,0) & \\
4.\det\begin{pmatrix} d_1 & d_4 \\ f_1 & f_4 \end{pmatrix} \neq 0 & & 5.\det\begin{pmatrix} d_1 & d_4 \\ f_1 & f_4 \end{pmatrix} = 0 \arrow[dl] \arrow[dr] & & \\
& 6.(d_1, d_4) = (0,0) & & 7.(d_1, d_4) \neq (0,0) \arrow[dl] \arrow[dr] & \\
& & 8.d_1 \neq 0 \arrow[dl] \arrow[dr] & & 11.d_1 = 0 \\
& 9.(f_2, f_3) = 0 & & 10.(f_2, f_3) \neq 0 &
\end{tikzcd}$$

**Fig. 5.** Case distinction tree for the proof of Lemma 7.

**Case 1.** Assume first that $\det(d_2\ d_3; f_2\ f_3) \neq 0$. Then we can do a linear change of variables

$$\begin{pmatrix} r' \\ s' \end{pmatrix} = \begin{pmatrix} d_2 & d_3 \\ f_2 & f_3 \end{pmatrix} \begin{pmatrix} r \\ s \end{pmatrix} + \begin{pmatrix} d_1 m + d_4 w + d_5 \\ f_1 m + f_4 w + f_5 \end{pmatrix}.$$

After relabeling the constants and dropping the primes, this reduces the equation into the form

$$(c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6) r = (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6) s$$

We can now choose $r = s = 0$, which is a valid signature for any $m$. Therefore, this case is impossible.

**Case 2.** We know that $\det(d_2\ d_3; f_2\ f_3) = 0$, so this splits into two subcases.

**Case 3.** If $(d_2, d_3) = 0$, then the verification equation (4) becomes

$$\begin{aligned}(c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6)(d_1 m + d_4 w + d_5) = \\ (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_1 m + f_2 r + f_3 s + f_4 w + f_5).\end{aligned} \tag{5}$$

We want to simplify the right factor on the left-hand side. This gives us two case distinctions.

**Case 4.** If $\det(d_2\ d_3; f_2\ f_3) \neq 0$, then we can make a change of variables $m' = d_1 m + d_4 w + d_5$, $w' = f_1 m + f_4 w + f_5$, which turns (5) into the form

$$\begin{aligned} (c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6)m = \\ (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_2 r + f_3 s + w). \end{aligned} \tag{6}$$

If $f_2 \neq 0$, then $r = -f_2^{-1} w$, $s = 0$ is a valid signature for $m = 0$. If $f_3 \neq 0$, the same argument works with $r$ and $s$ swapped. Therefore, $f_2 = f_3 = 0$ and (6) becomes

$$(c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6)m = (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)w,$$

and in that equation, $(c_2, c_3), (e_2, e_3)$ cannot be collinear, as there would be only one independent signature element otherwise. As a result, a change of variables lets us rewrite it as:

$$(r + c_4 v)m = (s + e_4 v)w. \tag{7}$$

If $c_4 \neq 0$, then $r = s = 0$ gives a valid signature on $m = (e_4/c_4) \cdot w$. Therefore, we must have $c_4 = 0$, and for any valid message-signature pair $(m; r, s)$, the pair $(2m; r/2, s)$ is also valid, which breaks security against random-message attacks.

**Case 5.** $\det(d_2\ d_3; f_2\ f_3) = 0$, so either $(d_1, d_4) = 0$ or $(d_1, d_4) \neq 0$, i.e. $(f_1, f_4) = \lambda(d_1, d_4)$.

**Case 6.** If $(d_1, d_4) = 0$, then (5) becomes

$$(c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6)d_5 = (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_1 m + f_2 r + f_3 s + f_4 w + f_5).$$

If $d_5 = 0$, then verification equation requires just one pairing and this is easily shown to be impossible. Hence, $d_5 \neq 0$ and we may assume that $d_5 = 1$ by scaling. The verification equation thus becomes

$$\begin{aligned} c_1 m + c_2 r + c_3 s + c_4 v + c_5 w + c_6 = \\ (e_1 m + e_2 r + e_3 s + e_4 v + e_5 w + e_6)(f_1 m + f_2 r + f_3 s + f_4 w + f_5). \end{aligned} \tag{8}$$

Assume now that $(m; r, s)$ is a valid message-signature pair satisfying the equation. Then $(m^*; r^*, s^*)$ given by:

$$\begin{aligned} m^* &= m + (e_2 f_3 - e_3 f_2)(f_1 m + f_2 r + f_3 s + f_4 w + f_5) + (c_3 f_2 - c_2 f_3) \\ r^* &= r + (e_3 f_1 - e_1 f_3)(f_1 m + f_2 r + f_3 s + f_4 w + f_5) + (c_1 f_3 - c_3 f_1) \\ s^* &= s + (e_1 f_2 - e_2 f_1)(f_1 m + f_2 r + f_3 s + f_4 w + f_5) + (c_2 f_1 - c_1 f_2) \end{aligned} \tag{9}$$

is another valid signature (easily obtained by looking for forgeries of the form $(m + \mu; r + \rho, s + \sigma)$ fixing the $\mathbb{G}_2$ side of (8), and solving the resulting linear system in $(\mu, \rho, \sigma)$. This is a forgery unless $m^* = m$ with overwhelming probability, which implies that

$$(e_2 f_3 - e_3 f_2)(f_1 m + f_2 r + f_3 s + f_4 w + f_5) + (c_3 f_2 - c_2 f_3) = 0,$$

and unless $\det(e_2\ e_3; f_2\ f_3) = \det(f_2\ f_3; c_2\ c_3) = 0$, that relation makes verification equation (8) linear, and hence the scheme clearly insecure.

Thus, we must have $\det(e_2\ e_3; f_2\ f_3) = \det(f_2\ f_3; c_2\ c_3) = 0$, and since the vectors $(c_2, c_3), (f_2, f_3), (e_2, e_3)$ cannot be all collinear by independence of the two signature elements, it follows that $(f_2, f_3) = 0$ and $\det(c_2\ c_3; e_2\ e_3) \neq 0$. Therefore, after a change of variables, (8) becomes:

$$r + c_4 v = (s + e_4 v)(f_1 m + f_4 w + f_5),$$

where $f_1 \neq 0$, since $m$ must be used in the verification equation. Therefore, after $m' = f_1 m + f_4 w + f_5$, we get $r + c_4 v = (s + e_4 v)m$, which does not use $w$ and must hence be insecure.

**Case 7.** In this case we have assumed that $\det(d_1\ \ d_4; f_1\ \ f_4) = 0$ and $(d_1, d_4) \neq 0$. It follows that $(f_1, f_4) = \lambda(d_1, d_4)$ and the verification equation is as in (5), i.e., it has the form

$$\begin{aligned}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)(d_1m + d_4w + d_5) =\\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_3s + f_4w + f_5).\end{aligned} \tag{10}$$

We now distinguish between whether $d_1$ is zero or not.

**Case 8.** Assume that $d_1 \neq 0$, so that we may make a change of variables $m' = d_1m + d_4w + d_5$, so that by $(f_1, f_4) = \lambda(d_1, d_4)$ the verification equation (10) becomes

$$\begin{aligned}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)m =\\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_3s + f_5).\end{aligned} \tag{11}$$

namely we can drop the $w$ term from the $\mathbb{G}_2$ side on the right. We now make a case distinction on whether $(f_2, f_3) = 0$.

**Case 9.** Assume that $(f_2, f_3) = 0$. In this case the equation becomes

$$(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)m = (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_5).$$

Now $\det(c_2\ \ c_3; e_2\ \ e_3) \neq 0$ or else there is just one independent signature element. It follows that we may simplify the equation to the form $(r + c_4v)m = (s + e_4v)(f_1m + f_5)$. However, $w$ is not used here, so the scheme is trivially insecure.

**Case 10.** Assume that $(f_2, f_3) \neq 0$. Looking at equation (11), we see that $(c_2, c_3), (f_2, f_3), (e_2, e_3)$ cannot be all collinear or else there is just one independent signature element. It follows that one of $\det(c_2\ \ c_3; f_2\ \ f_3)$ and $\det(f_2\ \ f_3; e_2\ \ e_3)$ must be nonzero. In the first case, (11) becomes

$$(r + c_4v)m = (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)s.$$

so that $m = 0, s = 0$ gives a valid signature for any $r$. In the latter case, (11) becomes

$$(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)m = (r + e_4v)s,$$

which is insecure by the same argument.

**Case 11.** Since $d_1 = 0$, the equation (10) becomes

$$\begin{aligned}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)(d_4w + d_5) =\\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_3s + f_4w + f_5).\end{aligned}$$

Let $(m; r, s)$ be a valid message-signature pair, then so is $(m^*; r^*, s^*)$ given by:

$$\begin{aligned}m^* &= m + (e_3f_2 - e_2f_3)(f_2r + f_3s + f_4w + f_5) + (c_2f_3 - c_3f_2)(d_4w + d_5)\\ r^* &= r + e_1f_2f_3r + e_1f_3^2s + (-c_1d_4f_3 + e_1f_3f_4)w - c_1d_5f_3 + e_1f_3f_5\\ s^* &= s - e_1f_2^2r - e_1f_2f_3s + (c_1d_4f_2 - e_1f_2f_4)w + c_1d_5f_2 - e_1f_2f_5\end{aligned}$$

(obtained by the same approach as in Case 6). As in Case 6, this provides a valid forgery or lets us linearize the verification equation (concluding in both cases) unless the determinants $\det(e_2\ \ e_3; f_2\ \ f_3)$ and

$\det(c_2\ c_3; f_2\ f_3)$ are both zero. In that case, since we know that $(c_2, c_3), (f_2, f_3), (e_2, e_3)$ can't be collinear, we obtain $(f_2, f_3) = 0$ and $\det(c_2\ c_3; e_2\ e_3) \neq 0$. This implies that we may write the verification equation in the form

$$(r + c_4v)(d_4w + d_5) = (s + e_4v)(f_1m + f_4w + f_5).$$

The equation must depend on both $m$ and $w$, so $d_4$ and $f_1$ are non zero, and we may simplify that further to $(r + c_4v)w = (s + e_4v)m$, and this has been ruled out in Case 4 already.

**Case 12.** We now have the original verification equation

$$\begin{gathered}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)(d_1m + d_2r + d_3s + d_4w + d_5) = \\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_3s + f_4w + f_5),\end{gathered}$$

but we know that $(d_2, d_3) \neq 0$ and $(f_2, f_3) = \lambda(d_2, d_3)$. We note that the problem is completely symmetric with respect to $d_2$ and $d_3$, so we may assume that $d_2 \neq 0$. Set $r' = d_1m + d_2r + d_3s + d_4w + d_5$ and $s' = s$. Since $(f_2, f_3) = \lambda(d_2, d_3)$, the verification equation becomes

$$\begin{gathered}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)r = \\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_1m + f_2r + f_4w + f_5),\end{gathered}$$

If $f_1 \neq 0$, choose $r = 0$ and pick an $m$ such that $f_1m + f_4w + f_5 = 0$. This is a valid forgery for any value of $s$. It follows that $f_1 = 0$, so that the equation becomes

$$\begin{gathered}(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)r = \\ (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)(f_2r + f_4w + f_5).\end{gathered}$$

Assume now that $(m; r, s)$ is a valid message-signature pair. Then so is $(m^*; r^*, s^*)$ given by

$$\begin{aligned} m^* &= m + (c_3 - e_3f_2)r - e_3f_4w - e_3f_5 \\ r^* &= r \\ s^* &= s + (e_1f_2 - c_1)r + e_1f_4w + e_1f_5. \end{aligned}$$

By the argument of Case 6 again, we deduce that the scheme is insecure (by forgery or linearization of the verification equation) unless perhaps if $c_3 - e_3f_2 = 0$, $e_3f_4 = 0$ and $e_3f_5 = 0$. From $c_3 = e_3f_2$, we see that if $e_3 = 0$, then $c_3 = 0$ and $s$ would not be used in the verification equation, which is impossible. It follows that $e_3 \neq 0$ and $f_4 = f_5 = 0$. The verification equation is therefore

$$(c_1m + c_2r + c_3s + c_4v + c_5w + c_6)r = (e_1m + e_2r + e_3s + e_4v + e_5w + e_6)f_2r.$$

It follows that $r = 0$ gives a forgery for any choice of $m$ and $s$. Therefore, we have another contradiction and since we have exhausted all cases, the proof is complete. □

## B Security of the Synthesized Schemes

### B.1 Proof of Theorem 3

*Proof.* Since we are in the generic bilinear group model, the adversary $\mathcal{A}$ can only perform generic operations via the oracles. Thus, in $\mathbb{G}_1$ and $\mathbb{G}_2$ it can compute only linear combinations of elements that are in the public key, or elements that were returned by the signing oracle (i.e., signatures). Such linear combinations in $\mathbb{G}_2$ correspond to formal Laurent polynomials whose variables are the discrete logarithms of the corresponding group elements.

From the public key, $\mathcal{A}$ sees in $\mathbb{G}_1$ the elements $G, V, W$ which correspond to $1, v, w$ respectively, while in $\mathbb{G}_2$ $\mathcal{A}$ only sees $H$, which corresponds to 1. For every signing query $M_i$ (with discrete log $m_i$), the adversary receives $R_i = H^{r_i}, S_i = M_i^{v/r_i} H^{w/r_i} \in \mathbb{G}_2$, namely $r_i$ and $s_i = m_i v/r_i + w/r_i$.

Let $(M, R, S) \in \mathbb{G}_2^3$ be the forgery returned by the adversary after making $q$ signing queries. From the assumption that $\mathcal{A}$ is generic, these three elements in $\mathbb{G}_2$ can be computed only as linear combinations of all the available material in $\mathbb{G}_2$. Namely, when looking at their discrete logarithms $(m, r, s)$, it must be

$$m = \mu + \sum_{i=1}^{q} \mu_{r_i} r_i + \sum_{i=1}^{q} \mu_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right)$$
$$r = \rho + \sum_{i=1}^{q} \rho_{r_i} r_i + \sum_{i=1}^{q} \rho_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right)$$
$$s = \sigma + \sum_{i=1}^{q} \sigma_{r_i} r_i + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right)$$

Moreover, in order for the adversary to succeed in the EUF-CMA security game, the verification equation

$$rs = mv + w$$

must be satisfied with significant probability over the random choices of $v, w$ and all the $r_i$ sampled in the security experiment. By the Schwartz-Zippel lemma [19,30,29] the above equation holds as an identity of polynomials.

Therefore, in our proof we show that for every forgery $(m, r, s)$ which verifies correctly it must be the case that $\mathcal{A}$ reuses one of the previously obtained messages, i.e., $m = m_j$ for some $j \in [q]$.

The verification equation is $rs = mv + w$ that, by expanding $r, s$ we can write as

$$\left( \rho + \sum_{i=1}^{q} \rho_{r_i} r_i + \sum_{i=1}^{q} \rho_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) \cdot \left( \sigma + \sum_{i=1}^{q} \sigma_{r_i} r_i + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = mv + w \quad (12)$$

First, we show that it must be $\sigma_{r_i} = 0$ for all $i = 1, \ldots, q$. Assume by contradiction that $\exists j \in [q] : \sigma_{r_j} \neq 0$. Then when looking at the coefficients of $r_j^2$, these must be zero, i.e., $\sigma_{r_j} \rho_{r_j} = 0$ which implies $\rho_{r_j} = 0$. Similarly, when analyzing the monomials $r_i r_j$ for all $i \neq j$, their coefficients $\rho_{r_i} \sigma_{r_j} + \rho_{r_j} \sigma_{r_i}$ must be zero. Since $\rho_{r_j} = 0$ and $\sigma_{r_j} \neq 0$ it must be $\rho_{r_i} = 0$ for all $i \neq j$. Hence, so far we have shown that it must be the case that $\rho_{r_i} = 0 \; \forall i \in [q]$.

If we now look at the coefficient of $r_j$ (on the left-hand side of equation (12)), i.e., $\rho \sigma_{r_j}$, this must be zero which implies $\rho = 0$. Essentially, equation (12) can be rewritten as

$$\left( \sum_{i=1}^{q} \rho_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) \cdot \left( \sigma + \sum_{i=1}^{q} \sigma_{r_i} r_i + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = mv + w \quad (13)$$

Now, let us analyze the terms $w/r_i$ whose coefficients must be zero as well, i.e., $\sigma \rho_{s_i} = 0 \; \forall i \in [q]$. This shows that $\sigma = 0$, otherwise (if $\sigma \neq 0$) it must be the case that $\rho_{s_i} = 0 \; \forall i \in [q]$, thus yielding a zero polynomial on the left-hand side of equation (13) which is impossible.

If we then look at the coefficients of terms $\frac{wr_j}{r_i}$ for all $i \neq j$, these must be zero, i.e., $\rho_{s_i}\sigma_{r_j} = 0$, which means $\rho_{s_i} = 0$ for all $i \neq j$ since $\sigma_{r_j} \neq 0$ by our assumption. Thus equation (13) can be rewritten as

$$\rho_{s_j}\left(\frac{m_j v}{r_j} + \frac{w}{r_j}\right) \cdot \left(\sum_{i=1}^{q} \sigma_{r_i} r_i + \sum_{i=1}^{q} \sigma_{s_i}\left(\frac{m_i v}{r_i} + \frac{w}{r_i}\right)\right) = mv + w \tag{14}$$

where $\rho_{s_j} \neq 0$ otherwise there would be a zero polynomial on the left-hand side of the equation.

Now, if we look at the terms $\frac{w^2}{r_i r_j}$, using a similar argument as above we obtain that $\sigma_{s_i} = 0$ for all $i \in [q]$. And if we look at the terms $\frac{wr_k}{r_j}$ for all $k \neq j$, it must be $\sigma_{r_k}\rho_{s_j} = 0$, which means $\sigma_{r_k} = 0$ for all $k \neq j$.

Hence, we are left with $\rho_{s_j}\left(\frac{m_j v}{r_j} + \frac{w}{r_j}\right)\sigma_{r_j} r_j = mv+w$ which simplifies to $\rho_{s_j}\sigma_{r_j}(m_j v+w) = mv+w$ that forces $\rho_{s_j}\sigma_{r_j} = 1$ and thus $m_j = m$, which is a contradiction since $(m, r, s)$ is a forgery only if $m \neq m_i$ for all $i = 1, \ldots, q$.

Therefore, it must be the case that $\sigma_{r_i} = 0$ for all $i = 1, \ldots, q$.

By using very similar arguments to the ones presented above, we can also argue that $\sigma_{r_i} = 0$ for all $i = 1, \ldots, q$ unless $m = m_j$.

Hence, since $\rho_{r_i} = \sigma_{r_i} = 0\ \forall i \in [q]$, equation (12) can be rewritten as

$$\left(\rho + \sum_{i=1}^{q} \rho_{s_i}\left(\frac{m_i v}{r_i} + \frac{w}{r_i}\right)\right) \cdot \left(\sigma + \sum_{i=1}^{q} \sigma_{s_i}\left(\frac{m_i v}{r_i} + \frac{w}{r_i}\right)\right) = mv + w \tag{15}$$

Now, if we expand $m$ in the right-hand side of the above equation, one can see that when considering the formal polynomials the equation cannot hold. To see this, assume that $\exists j : \rho_{s_j} \neq 0$. By looking at the coefficients of $\frac{w^2}{r_i r_j}$, it must hold $\sigma_{s_i} = 0$ for all $i \in [q]$. It is clear that $\sigma \neq 0$ otherwise one has a zero polynomial on the left-hand side of equation (15). At this point either $mv + w$ contains the constant term $\rho\sigma$, or (if $\rho = 0$) it must contain a term $w/r_j$ which is not the case either. This shows that $\rho_{s_i} = 0\ \forall i \in [q]$. Analogously, one can see that equation $\rho\left(\sigma + \sum_{i=1}^{q} \sigma_{s_i}\left(\frac{m_i v}{r_i} + \frac{w}{r_i}\right)\right) = mv + w$ cannot hold as well. $\square$

## B.2 Proof of Theorem 4

*Proof.* The proof proceeds similarly to the one of Theorem 3 except that now we have to take into considerations the two verification equations as well as the additional information which is received in the different groups.

From the public key, $\mathcal{A}$ sees in $\mathbb{G}_1$ the elements $G, V, W$ which correspond to discrete logs $1, v, w$ respectively, while in $\mathbb{G}_2$ $\mathcal{A}$ only sees $H$, which corresponds to 1. For every signing query $M_i$ (with discrete log $m_i$), the adversary receives $R_i = G^{r_i} \in \mathbb{G}_2, T_i = H^{1/r_i}, S_i = M_i^{v/r_i} H^{w/r_i} \in \mathbb{G}_2$, namely $r_i$, $1/r_i$ and $s_i = m_i v/r_i + w/r_i$.

Let $(M, R, T, S) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2^2$ be the forgery returned by the adversary after making $q$ signing queries. From the assumption that $\mathcal{A}$ is generic, the elements $(M, R, S)$ in $\mathbb{G}_2$ can be computed only as linear combinations of all the available material in $\mathbb{G}_2$, and similarly $T$ can be computed only as a linear combination of the available group elements in $\mathbb{G}_1$. Namely, when looking at their discrete logarithms

$(m, r, t, s)$, it must be

$$\begin{aligned} m &= \mu + \sum_{i=1}^{q} \mu_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \mu_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \\ r &= \rho + \sum_{i=1}^{q} \rho_{r_i} r_i + \rho_v v + \rho_w w \\ t &= \tau + \sum_{i=1}^{q} \tau_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \tau_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \\ s &= \sigma + \sum_{i=1}^{q} \sigma_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \end{aligned}$$

Moreover, in order for the adversary to succeed in the EUF-CMA security game, the two verification equations

$$rs = mv + w \wedge rt = 1$$

must be satisfied with significant probability over the random choices of $v, w$ and all the $r_i$ sampled in the security experiment. By the Schwartz-Zippel lemma [19,30,29] the above equations hold as identities of polynomials.

For our proof we first consider the second equation $rt = 1$ and we show that, when expanding $r, t$ as follows

$$\left( \rho + \sum_{i=1}^{q} \rho_{r_i} r_i + \rho_v v + \rho_w w \right) \cdot \left( \tau + \sum_{i=1}^{q} \tau_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \tau_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = 1 \qquad (16)$$

it must be the case that both $\rho_v$ and $\rho_w$ are 0.

Assume by contradiction that $\rho_w \neq 0$. Then by looking at the coefficients of $w^2/r_i$, these must be zero, which implies that $\tau_{s_i} = 0$ for all $i \in [q]$. Similarly, we also obtain that all coefficients of $w/r_i$ must be zero, and thus $\tau_{t_i} = 0$, $\forall i \in [q]$. So, we are left with

$$\left( \rho + \sum_{i=1}^{q} \rho_{r_i} r_i + \rho_v v + \rho_w w \right) \cdot \tau = 1 \qquad (17)$$

which, being $\rho_w \neq 0$ is however impossible. Hence, $\rho_w = 0$, and in an analogous way, one can prove that also $\rho_v = 0$.

In conclusion, from the fact that the forgery satisfies $rt = 1$ we obtain that it must be the case that $\rho_v = \rho_w = 0$. In the next part of the proof we will use this information to derive a contradiction on the fact that $m \neq m_i$ for all $i = 1$ to $q$.

Plugging this information in the first verification equation expanded for $r, s$ we obtain

$$\left( \rho + \sum_{i=1}^{q} \rho_{r_i} r_i \right) \cdot \left( \sigma + \sum_{i=1}^{q} \sigma_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = mv + w \qquad (18)$$

First, we show that $\sigma_{t_i} = 0$ for all $i = 1, \ldots, q$. Indeed, assume by contradiction that $\exists j : \sigma_{t_j} \neq 0$. Then, by looking at the coefficients of terms $r_i/r_j$ for all $i \neq j$ we obtain that $\rho_{r_i} = 0$ for all $i \neq j$. And similarly, since the term $1/r_j$ is not in $mv + w$ it must be $\rho\sigma_{t_j} = 0$ and thus $\rho = 0$.

Essentially, we can rewrite equation (18) as

$$\rho_{r_j} r_j \cdot \left( \sigma + \sum_{i=1}^{q} \sigma_{t_i} \frac{1}{r_i} + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = mv + w \tag{19}$$

where $\rho_{r_j} \neq 0$ otherwise we would get a zero polynomial on the left-hand side of the equation above. By analyzing the terms $r_j/r_i$ for all $i \neq j$ we obtain that $\sigma_{t_i} = 0$ for all $i \neq j$. Similarly, we also have $\sigma = 0$. When looking at the terms $wr_j/r_i$ for all $i \neq j$, we obtain that $\sigma_{s_i} = 0$ for all $i \neq j$. This allows us to rewrite equation (19) as

$$\rho_{r_j} r_j \cdot \left( \sigma_{t_j} \frac{1}{r_j} + \sigma_{s_j} \left( \frac{m_j v}{r_j} + \frac{w}{r_j} \right) \right) = mv + w$$

which simplifies to

$$\rho_{r_j} \sigma_{s_j} \left( m_j v + w \right) = mv + w \tag{20}$$

since $\sigma_{t_j}$ must be zero. However, equation (20) holds only if $m = m_j$ which is a contradiction. Therefore, $\sigma_{t_i} = 0$ for all $i = 1, \ldots, q$.

Using this information, we can rewrite equation (18) as

$$\left( \rho + \sum_{i=1}^{q} \rho_{r_i} r_i \right) \cdot \left( \sigma + \sum_{i=1}^{q} \sigma_{s_i} \left( \frac{m_i v}{r_i} + \frac{w}{r_i} \right) \right) = mv + w \tag{21}$$

Since the terms $r_j w/r_i$ must have coefficients zero, it must be $\rho_{r_j}\sigma_{s_i} = 0$ for all $i \neq j$. We claim that there exists exactly one $k \in [q]$ such that both $\rho_{r_k}$ and $\sigma_{s_k}$ are non-zero while $\rho_{r_i} = \sigma_{s_i} = 0$ for all $i \neq k$. First, assume by contradiction that there exists two distinct indices $k_1, k_2$ such that $\rho_{r_{k_1}} \neq 0$ and $\sigma_{s_{k_2}} \neq 0$. Then it would be the case the term $wr_{k_1}/r_{k_2}$ has a non-zero coefficient, which is impossible. Second, we show that it cannot be the case that only one of $\rho_{r_k}$ and $\sigma_{s_k}$ is non-zero. Indeed assume by contradiction that $\rho_{r_k} = 0$, then one can see that the equation (21) cannot be satisfied. Similarly, it holds $\sigma_{s_k} \neq 0$.

Therefore, it must be the case that $\rho_{r_k} \neq 0$ and $\sigma_{s_k} \neq 0$, and we can simplify the equation as

$$(\rho + \rho_{r_k} r_k) \cdot \left( \sigma + \sigma_{s_k} \left( \frac{m_k v}{r_k} + \frac{w}{r_k} \right) \right) = mv + w \tag{22}$$

However, it is not hard to see that it must be $\rho = \sigma = 0$, which means $\rho_{r_k}\sigma_{s_k}(m_k v + w) = mv + w$, i.e., $m = m_k$, which is a contradiction. □

### B.3 Proof of Theorem 5

*Proof.* In this case, the generic adversary receives the verification key $V, W \in \mathbb{G}_1$ as well as $q$ valid message-signature pairs $(M_i; R_i, S_i) \in \mathbb{G}_2 \times \mathbb{G}_2^2$, $i = 1, \ldots, q$, on random messages In particular, the discrete logarithms $(m_i; r_i, s_i)$ satisfy that $m_i, r_i$ are uniformly random in $\mathbb{Z}_p$ and $s_i$ is given by $s_i = (r_i + v)m_i - w$. We need to prove that this information does not enable the adversary to construct a valid signature $(R, S)$ on any message $M$ other the $M_i$'s themselves.

Suppose that he does construct a valid pair $(M; R, S)$, with discrete logarithms $(m; r, s)$. Since the adversary is generic and the only available elements in $\mathbb{G}_2$ are $H$ and the $M_i$'s, $R_i$'s and $S_i$'s, $m, r, s$ must be explicit linear combinations of $1, m_1, \ldots, m_q, r_1, \ldots, r_q, s_1, \ldots, s_q$. In other words, $m, r, s$ are of the form:

$$m = \mu + \sum_{i=1}^{q} \mu_{m_i} m_i + \mu_{r_i} r_i + \mu_{s_i}\big((r_i + v)m_i - w\big),$$
$$r = \rho + \sum_{i=1}^{q} \rho_{m_i} m_i + \rho_{r_i} r_i + \rho_{s_i}\big((r_i + v)m_i - w\big),$$
$$s = \sigma + \sum_{i=1}^{q} \sigma_{m_i} m_i + \sigma_{r_i} r_i + \sigma_{s_i}\big((r_i + v)m_i - w\big).$$

Moreover, for the adversary to succeed, the verification equation

$$rm + vm = s + w \tag{23}$$

should hold with significant probability on the choice of the private key $v, w$ and the random values $m_i, r_i$. As a result, the Schwartz-Zippel lemma [19,30,29] implies that relation (23) holds as an identity of polynomials in $\mathbb{Z}_p[v, w, m_1, \ldots, m_q, r_1, \ldots, r_q]$.

Now, the monomial $v$ appears only in the term $vm$, with coefficient $\mu$. Therefore, $\mu = 0$. Similarly, the monomials $vr_i$ and $v^2 m_i$ appear only in $vm$, with coefficient $\mu_{r_i} + \mu_{s_i}$ and $\mu_{s_i}$ respectively. Therefore, $\mu_{r_i} = \mu_{s_i} = 0$ for all $i$, and $m$ simply becomes:

$$m = \sum_{i=1}^{q} \mu_{m_i} m_i.$$

Clearly, the coefficients $\mu_{m_i}$ cannot all be zero: otherwise, we obtain $s + w = 0$, and inspection of the coefficients of the monomials $1$, $m_i$, $r_i$ and $vm_i$ yields that all the coefficients of $s$ vanish, and hence $w = 0$, which is a contradiction.

Therefore, we can fix an index $j$ such that $\mu_{m_j} \neq 0$. The monomial $m_j^2$ appears only in the term $rm$ with coefficient $\mu_{m_j}\rho_{m_j}$, hence $\rho_{m_j} = 0$, and $m_i m_j$, $i \neq j$ also appears only in $rm$ with coefficient $\mu_{m_i}\rho_{m_j} + \mu_{m_j}\rho_{m_i} = \mu_{m_j}\rho_{m_i}$, hence $\rho_{m_i} = 0$ for all $i$. Similarly, inspection of the coefficients of $vr_i m_j$, $r_i m_j$ yields $\rho_{s_i} = 0$ for all $i$ and $\rho_{r_i} = 0$ for all $i \neq j$ respectively. Hence:

$$r = \rho + \rho_{r_j} r_j.$$

Then, inspection of the coefficients of $vm_i$ and $r_i m_i$ gives $\sigma_{s_i} = \mu_{m_i} = \mu_{m_i}\rho_{r_i}$ for all $i$. In particular, we get $\sigma_{s_j} = \mu_{m_j}$, $\rho_{r_j} = 1$, and for all $i \neq j$, $\sigma_{s_i} = \mu_{m_i} = 0$. Moreover, the coefficient of $w$ is $0$ on the left-hand side and $1 - \sum_i \sigma_{s_i}$ on the right-hand side, hence $\sigma_{s_j} = \mu_{m_j} = 1$. This shows that $m = m_j$, which concludes the proof. □

## C Basic definitions

### C.1 Secure signature schemes

A digital signature scheme (with setup algorithm *Setup*) is a quadruple of efficient algorithms (*Setup*, *KeyGen*, *Sign*, *Verify*). The setup algorithm *Setup* takes the security parameter and outputs a public parameter *PP*. The key generation algorithm *KeyGen* takes *PP* as input and returns a public verification key

*VK* and a secret signing key *SK*. We will always assume that *SK* includes *VK*. The signing algorithm $\mathsf{Sign}$ takes the public parameters *PP*, the signing key *SK* and a message $M$ in the message space $\mathcal{M}$ defined by *PP* and returns a signature $\Sigma$. The verification algorithm $\mathsf{Verify}$ takes the public parameters *PP*, the verification key *VK*, a message $M$ and the signature $\Sigma$ and returns either 1 (accept) or 0 (reject). We say that the signature scheme $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is correct if for all correctly generated public parameters *PP* and key pairs $(SK, VK)$ and all messages $M$ in the corresponding message space $\mathcal{M}$, we have $\mathsf{Verify}\big(PP, VK, M, \mathsf{Sign}(PP, SK, M)\big) = 1$.

A signature scheme is said to be existentially unforgeable if no efficient adversary is able to produce a valid signature on a new message after having seen correctly generated signatures on arbitrarily many messages—either sampled randomly from the message space, in the case of random message attacks (RMA), or adaptively chosen by the adversary hismelf, in the case of chosen message attacks (CMA).

**Definition 2** **(EUF-CMA).** *A signature scheme* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is existentially unforgeable under adaptive chosen message attack (EUF-CMA) if for all non-uniform polynomial time algorithms* $\mathcal{A}$*, we have:*

$$\Pr\left[\begin{array}{ll} PP \leftarrow \mathsf{Setup}(1^\lambda) & \\ (VK, SK) \leftarrow \mathsf{KeyGen}(PP) & : M \notin Q \ \wedge \ \mathsf{Verify}(PP, VK, M, \Sigma) = 1 \\ (M, \Sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(PP, SK, \cdot)}(PP, VK) & \end{array}\right] = \mathrm{negl}(\lambda),$$

*where* $Q$ *is the set of queries made by* $\mathcal{A}$ *to the signing oracle.*

Sometimes it is also useful to prevent the adversary from issuing a new signature for a message that has already been signed. A signature scheme is strongly existentially unforgeable if it is hard to find a signature on a message that has not been signed before and also hard to find a new signature for a message that has already been signed. This notion, denoted by sEUF-CMA, is formally captured in the same way as the definition of EUF-CMA except for requiring $(M, \Sigma) \notin Q$ where $Q$ is the set of message-signature pairs from $\mathcal{A}$'s queries to the signing oracle.

We get the definition for existential unforgeability against random message attack (EUF-RMA) by modifying the signing oracle to picking $M \leftarrow \mathcal{M}$ at random, computing $\Sigma \leftarrow \mathsf{Sign}(PP, SK, M)$ and returning $(M, \Sigma)$ to the adversary whenever the signing oracle is queried.

Corresponding security notions for one-time (resp. $n$-time) signature schemes can be obtained by restricting the adversary to only calling the signing oracle once (resp. $n$ times) in the above definitions.