Strong Security from Probabilistic Signature Schemes

Sven Schäge*

University College London
s.schage@ucl.ac.uk

Abstract. We introduce a new and very weak security notion for signature schemes called target randomness security. In contrast to previous security definitions we focus on signature schemes with (public coin) probabilistic signature generation where the randomness used during signature generation is exposed as part of the signature. To prove practical usefulness of our notion we present a new signature transformation for mapping target randomness secure signature schemes to weakly secure signature schemes. It is well-known that, using chameleon hash functions, the resulting weakly secure scheme can then be turned into a fully secure one. Our transformation outputs signature schemes that in general produce signatures with l elements, where l is the bit length of the input randomness. We present an instantiation of a target randomness secure signature scheme based on the RSA assumption and show that after applying our new signature transformation to this scheme, we can *accumulate* the l signature elements into a single element. This results in a new efficient RSA-based signature scheme. In contrast to traditional security definitions, all signature schemes obtained with our transformation enjoy *strong* security, i.e. they remain secure even if the adversary outputs a new signature on a previously queried message. In our proofs, we rely on the prefix-based technique introduced by Hohenberger and Waters at Crypto'09. However, using a precise analysis we are able decrease the security loss in proofs relying on the prefix-based technique. This result may be of independent interest.

Keywords: digital signature, signature scheme, RSA, accumulation, target randomness, transformation, prefix, tightness

1 Introduction

Signature transformations that map signature schemes with weak security guarantees to schemes which fulfill the standard notion of security have proven very useful in the past. This is because it is much more easy to design a signature scheme which only fulfills a very weak notion of security than a fully secure scheme. Many of the existing signature schemes like [25, 5, 21, 6] have been developed in this spirit by first specifying a signature scheme with weak guarantees and then applying a corresponding signature transformation to construct a

* Supported by EPSRC grant number EP/G013829/1.

scheme with stronger properties. This is also true for the recent signature scheme by Hohenberger and Waters (HW) from Crypto'09. In their work, Hohenberger and Waters present the first RSA-based hash-and-sign signature scheme in the standard model and so solved a long-standing open problem [21]. To this end, they tackle the problem that the challenger can predict the message $\overline{M}$ that the adversary will generate a signature on (in the following called target message) only with negligible probability. Their solution is to force the adversary to not only process $\overline{M}$ but also, *independently*, all prefixes of $\overline{M}$. (In the following, we refer to this approach as 'prefix-based technique'.) This greatly reduces the complexity for the simulator, because now it can guess with non-negligible probability one of the prefixes of $\overline{M}$. The remaining task of the challenger is to embed the RSA challenge such that it can extract a solution from the attacker's forgery if its guess was correct.

As Hohenberger-Waters and Brakerski and Tauman Kalai (BTK) [6] pointed out, the HW scheme gives rise to a new transformation[1] that step-wisely transforms signature schemes which only guarantee a very weak form of security – security against universal (message) forgeries under generic chosen (message) attacks (`UMUF-GMA`)[2], to weakly secure schemes, i.e. schemes secure against existential (message) forgeries under generic chosen message attacks (`EMUF-GMA`). (For formal definitions we refer to Section 2.1.) The `UMUF-GMA` security game is equivalent to the definition of `EMUF-GMA` security, except that the attacker *is given* the message it has to produce a forgery on – subsequently called 'target message' – in the first move of the security experiment. The transformation essentially grasps the ideas behind the prefix-based technique by HW. The key idea is to use a `UMUF-GMA` secure scheme to sign the l prefixes of the message M as in the HW scheme. The final signature consists of all the signatures on these prefixes. However, apparently the HW scheme cannot be obtained as a direct application of this transformation; HW signatures have constant size and do not grow with the message length as one might expect given the BTK transformation description. There must be some additional structure that the HW scheme exploits.

In this work we analyze public coin probabilistic signature schemes, where the randomness used in the signature generation is also sent to the verifier (as part of the signature). We show how public coin signatures schemes that only fulfill very weak notions of security can be used to construct efficient fully *and strongly* secure signature schemes.

Contribution. We extend the existing work on signature schemes in the standard model. In particular we

- define a new and very weak security notion called *target randomness security* that defines 'existential message universal randomness unforgeability against generic chosen message and randomness attacks' (`EMURUF-GMRA`) for signa-

[1] This transformation was implicitly given in [21] and made explicit in [6].
[2] [6] use the term *a-priori-message unforgeability* to refer to `UMUF-GMA` security.

tures with public coin probabilistic signature generation. Signature schemes that are EMURUF-GMRA secure are always secure in the strong sense.
- present a new *general* transformation from EMURUF-GMRA secure signature schemes to weakly secure signature schemes. Signatures of the resulting scheme consist of l elements and are strongly secure as well.
- present a new and efficient target randomness secure RSA-based signature scheme with a probabilistic signing algorithm.
- show that when applying our transformation to our RSA-based scheme the signature elements of the resulting weakly secure signature scheme can be accumulated into a single group element. This results in a new and efficient RSA-based signature scheme with constant-size signatures. Slightly modified our technique makes obvious why the size of the HW signatures does not grow with the message size.
- improve the loss of tightness in prefix-based security reductions. This improvement transfers to all proofs that rely on the prefix-based techniques like HW [21], BTK [6], and the recent signature scheme by Hofheinz, Jager and Kiltz (HJK) [18].

To obtain signature schemes secure under the standard definition – security against existential message unforgeability under adaptive chosen message attacks (EMUF-AMA) [17] or full security – we can apply the well-known Shamir-Tauman transformation to generate EMUF-AMA secure schemes from EMUF-GMA secure schemes [31]. In Section 2.2 we extend this result by showing that if the chameleon hash function also guarantees a certain form of strong security, the resulting signature scheme is strongly secure too. We stress that most chameleon hash function are also secure in this strong sense. This is advantageous in scenarios where strong security is required, as we do not have to apply an additional transformation like the Bellare-Shoup [3] or Huang *et al.* [22] transformation to turn EMUF-AMA secure schemes into strongly secure ones. These transformations increase the signature size and make the signing and verification algorithms less efficient.

Related Work. Our work is related to the existing standard model signature schemes. Most of the factoring-based hash-and-sign signature schemes were proven secure under the Strong RSA (SRSA) assumption [16, 13, 25, 34, 35, 7, 15, 19, 30]. Until 2009, it was an open problem to design an efficient signature scheme that is secure solely under the RSA assumption. Hohenberger and Waters stepwisely solved this problem by first presenting a stateful RSA-based signature scheme at EUROCRYPT'09 [20] and, in the same year, a stateless RSA-based signature scheme at CRYPTO'09 [21]. Recently, Hofheinz, Jager, and Kiltz (HJK) presented a new signature scheme that is secure under the sole RSA assumption [18]. It essentially relies on programmable hash functions as introduced in [19] and results in very small signature sizes while having a relatively large number of public key elements. Technically, the authors also rely on the prefix-based proof technique and similar to our result, they also use all the prefixes of the *randomness* to sign a message. Our RSA-based scheme differs from their scheme in the following way. 1) Signatures are longer as we need two

random values for signature generation. However, in 2006 Mironov showed how to re-use the second random value as a key to a target collision resistant (TCR) hash function [24]. Target collision resistant hash functions can be used as an alternative to collision-resistant (CR) hash functions for domain extension of the message space. In contrast to CRs, TCRs do only rely on the existence of one-way functions. At the same time they are much more efficient than provably secure CRs in groups of hidden (composite) order [9]. Thus our instantiation allows to efficiently sign long messages without relying on additional security assumptions. Similar arguments hold if we compare our scheme with the HW scheme, which also only uses a single random value, see Section 6.1. 2) Our public key is much smaller and comparable to that of the HW scheme. We stress that besides these issues our focus is much more general. Our main result consist in a new security definition together with an appropriate transformation to weakly secure schemes. We aim at showing that probabilistic signature schemes, even with very weak security properties, provide interesting starting points for the construction of strongly secure signature schemes. At the same time our tightness improvements hold for prefix-based security proofs *in general* and independent of the underlying security assumption.

By now there exist several security definitions and corresponding transformations for signature schemes. In 1989 Even, Goldreich and Micali showed 1) how to construct fully secure signature schemes from schemes that are secure under known message attacks and 2) how to construct *practical* fully secure signature schemes from schemes that are chosen message secure [14]. Cramer, Damgård, and Pedersen presented an alternative construction for 1) that features a much smaller signature size ($O(\kappa)$ instead of $O(\kappa^2)$ in the Even *et. al.* transformation where κ is the security parameter) [12]. The signature scheme by Naccache *et al.* can be interpreted as an application of this transformation to an SRSA-based known message secure signature scheme [25]. In 2001, Shamir and Tauman presented an improved transformation for 2) that maps weakly secure signature schemes to fully secure schemes using chameleon hash functions [31]. Due to the efficiency of the resulting signature schemes this transformation is very popular and an essential ingredient in several signature schemes like [5, 21]. In 2007, Bellare and Shoup [3] and independently Huang *et. al.* [22] presented a generic transformation to construct strongly secure signature schemes. In contrast to the standard security notion, in the attack game of strongly secure signature schemes the adversary is also allowed to output *a new signature on a previously queried message*. Figure 1 gives an overview of the existing (and new) notions and transformations of chosen-message security. We have ignored selective security, where the adversary may choose the target message/randomness, as universally secure schemes can trivially be transformed into selectively secure schemes under generic chosen message attacks (see HW [21] and BTK [6]). A random element X is simply added to the public key and in the first step of the signature generation the message is XORed with X. This technique is implicit in HW and the signature transformation of Section 3.

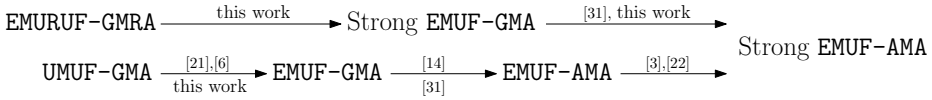

Fig. 1. Security Notions and Transformations of Chosen-Message Secure Signature Schemes. Arrows from A to B indicate that an efficient transformation exists that constructs a secure instantiation of B from a secure instantiation of A. References below arrows point to improved transformations.

With respect to tightness improvements, our work follows the line of work initiated by Bellare and Rogaway who showed tight security of PSS [1]. We concentrate on tightness improvements for existing signature schemes (*without* introducing additional modifications). In 2000, Coron provided tighter proofs for Full-Domain Hash [10, 11]. Eight years later Bernstein [4] presented the first tight proofs for the Rabin-William's signature schemes. All these result hold in the random oracle model. In 2008, Hofheinz and Kiltz [19] presented asymptotical tightness improvements for the Computational Diffie-Hellman based signature scheme by Waters [33] that is secure without random oracles. Recently, Schäge presented tight proofs for (new and) existing SRSA and Strong Diffie-Hellman based signature schemes in the standard model like the Cramer-Shoup [13], Fischlin [15], Zhu [34, 35], and Camenisch-Lysyanskaya [8] scheme.

2 Preliminaries and Notation

The security parameter is denoted as $\kappa \in \mathbb{N}$. We write 1^κ to describe the string that consist of κ ones and let $l = l(\kappa)$ and $q = q(\kappa)$ be polynomials. For a set S, we use $x \xleftarrow{\$} S$ to denote that x is drawn from S uniformly at random and $|S|$ to denote the cardinality of S. If s is a string, we write $|s|$ to denote its bit-length.We let $\perp$ denote the empty string. If $M \in \{0,1\}^l$ we let $M = m_1 m_2 \dots m_l$ with $m_j \in \{0,1\}$ for $j \in \{1, \dots, l\}$ be the binary representation of M. We use M^i to denote the prefix of M which consist of the first $i \in [1; l]$ bits: $M^i = m_1 m_2 \dots m_i$. For an algorithm $\mathcal{A}$ we write $\mathcal{A}(i_1, i_2, \dots)$ to denote that $\mathcal{A}$ has input parameters $i_1, i_2, \dots$. Similarly, we denote with $y \leftarrow \mathcal{A}(i_1, i_2, \dots)$ that $\mathcal{A}$ outputs y when running on inputs $i_1, i_2, \dots$. We write PPT (probabilistic polynomial time) to refer to randomized algorithms that run in polynomial time. As usual $\gcd(a, b)$ with $a, b \in \mathbb{Z}$ denotes the greatest common divisor of a and b. Our new signature scheme will be secure under the well-known RSA assumption [27].

Definition 1 (RSA assumption (RSA)). *Given an RSA modulus $n = pq$, where p, q are sufficiently large primes, a prime $\alpha < \phi(n)$ with $\gcd(\alpha, \phi(n)) = 1$, and an element $u \in \mathbb{Z}_n^*$, we say that the $(t_{RSA}, \epsilon_{RSA})$-RSA assumption holds if for all t_{RSA}-time adversaries $\mathcal{A}$*

$$Pr[(x) \leftarrow \mathcal{A}(n, u, \alpha),\ x \in \mathbb{Z}_n^*,\ x^\alpha = u \bmod n] \leq \epsilon_{RSA}.$$

The probability is over the random choices of u, n, α and the random coins of $\mathcal{A}$.

2.1 Signature Scheme

In a digital signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ the PPT algorithm KeyGen generates the key material: secret key SK and public key PK. The algorithm $\mathsf{Sign}(SK, M)$ uses SK and a message M from the message space $\{0,1\}^l$ to output a signature σ. If the signing algorithm is probabilistic we use $R \xleftarrow{\$} \{0,1\}^l$ to denote the randomness used for the signature generation. In this case we also write $\mathsf{Sign}(SK, M, R)$. The verification algorithm $\mathsf{Verify}(PK, M, \sigma)$ processes PK, a message M, and a purported signature σ on M and outputs 1 if σ is a legitimate signature on M and 0 otherwise. If Sign is probabilistic $\mathsf{Verify}(PK, M, \sigma, R)$ additionally processes randomness R that was used for the signature generation. Usually one may regard this randomness as a part of the signature. For clarity we deviate from this convention and make R explicit.

We restrict ourself to signature schemes where the randomness R is generated in the signing phase and directly given to the verifier (as part of the signature). Most of the existing signature schemes have this property, examples are [16, 13, 25, 34, 35, 7, 15, 20, 21, 5, 8, 19, 29]. In the following two security definitions we consider the most general case of forgeries – existential forgeries. We use the terminology of Goldwasser, Micali and Rivest [17].

Full Security: Existential Message Unforgeability under Adaptive Chosen Message Attacks (`EMUF-AMA`). The standard notion of security for signature schemes is called existential message unforgeability under adaptive chosen message attacks [17]. Here the adversary is given access to a signing oracle $\mathcal{O}_{SK}(\cdot)$ to adaptively query signatures.

Setup. In the setup phase, $\mathsf{KeyGen}(1^\kappa)$ is run and the public key PK is given to the adversary.

Signature queries. The adversary adaptively queries the signing oracle $\mathcal{O}_{SK}(\cdot)$ with q messages $M_1, \ldots, M_q \in \{0,1\}^l$ of his choice and obtains q signatures $\sigma_1, \ldots, \sigma_q$ with $\mathsf{Verify}(PK, M_i, \sigma_i) = 1$ for $i \in \{1, \ldots, q\}$.

Output. The attacker outputs $(\overline{M}, \overline{\sigma})$ such that $\overline{M} \notin \{M_1, \ldots, M_q\}$ and at the same time $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}) = 1$.

Weak Security: Existential Unforgeability under Generic Chosen Message Attacks (`EMUF-GMA`). In this attack model, the attacker specifies all signature queries *before it receives the public key.*

Signature queries. At first the adversary outputs a list of q signature queries $M_1, \ldots, M_q \in \{0,1\}^l$.

Public Key Generation and Signature Output. In the next phase, the public key PK is given to the adversary together with q signatures $\sigma_1, \ldots, \sigma_q$ such that $\mathsf{Verify}(PK, M_i, \sigma_i) = 1$ for $i \in \{1, \ldots, q\}$.

Output. The attacker outputs $(\overline{M}, \overline{\sigma})$ such that $\overline{M} \notin \{M_1, \ldots, M_q\}$ and at the same time $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}) = 1$.

Both of the above security notions are well-known. As sketched above, the only difference between the definition of `UMUF-GMA`-security and weak security is that

the attacker is given the 'target message' $\overline{M}$ in the first step of the security game and only has to output $\overline{\sigma}$. Let us now present our new security definition for signature schemes with a probabilistic signing algorithm.

Target Randomness Security: Existential Message Universal Randomness Unforgeability under Generic Chosen Message & Randomness Attacks (`EMURUF-GMRA`). In contrast to the previous security definitions we exploit the randomness R used in probabilistic public coin signing algorithms; both the challenger and the attacker can now also specify the randomness used for the signature generation. The adversary is given the target randomness in the first step of the security experiment. Informally we refer to this notion as target randomness security.

Target Randomness. At first the attacker is given the (target) randomness $\overline{R}$.

Signature queries. The adversary outputs q pairs of message/randomness as $(M_1, R_1) \ldots, (M_q, R_q)$ with $M_i, R_i \in \{0,1\}^l$. For at most one pair (M_i, R_i) it may hold that $R_i = \overline{R}$.

Public Key Generation and Signature Output. Next, the public key PK is given to the adversary together with q signatures $\sigma_1, \ldots, \sigma_q$ such that $\mathsf{Verify}(PK, M_i, \sigma_i, R_i) = 1$.

Output. The attacker outputs $\overline{M}, \overline{\sigma}$ such that $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}, \overline{R}) = 1$ and $\overline{M}, \overline{R}$ is not among the signature queries.

We denote the success probability of an adversary $\mathcal{A}$ (taken over the random coins of the challenger and the adversary) to win the i-security game as $Adv_{\mathcal{S},\mathcal{A},i}$ where $i \in \{$`EMUF-AMA`, `EMUF-GMA`, `EMURUF-GMRA`$\}$.

Definition 2 (Secure signature scheme). *An adversary $\mathcal{A}$ is said to (q, t, ϵ)-break the i-security ($i \in \{$`EMUF-AMA`, `EMUF-GMA`, `EMURUF-GMRA`$\}$) of a signature scheme $\mathcal{S}$ if $\mathcal{A}$ has success probability $Adv_{\mathcal{S},\mathcal{A},i} = \epsilon$ after generating at most q queries and running in time t. $\mathcal{S}$ is said to be (q, ϵ, t)-secure if there exists no PPT adversary that (q, ϵ, t)-breaks the existential unforgeability of $\mathcal{S}$. A signature scheme is called* strongly secure *if in the above games $\mathcal{A}$ may also output a forgery with $\overline{M} \in \{M_1, \ldots, M_q\}$ but $(\overline{M}, \overline{\sigma}) \notin \{(M_1, \sigma_1), \ldots, (M_q, \sigma_q)\}$ (or $(\overline{M}, \overline{\sigma}, \overline{R}) \notin \{(M_1, \sigma_1, R_1), \ldots, (M_q, \sigma_q, R_q)\}$ in case of probabilistic signatures).*

Discussion. In our new security definition the adversary is still allowed to output messages $\overline{M}$ of his choice (as in the first two security definitions), i.e. existential message forgeries (in contrast to `UMUF-GMA` security). Observe that previously queried messages may be re-used in the forgery what makes our definition inherently strongly secure. In the new security game, the adversary can now explicitly specify the random values R_i. To the best of our knowledge no existing security definition for signature schemes allows similar attack capabilities. It does not only give the adversary control over the messages to be signed but also *specifies* the randomness used for signature generation. In most signature schemes it is essential for security that the randomness is not controlled by the adversary. In our case, the only restriction is that the forgery must verify under randomness $\overline{R}$,

while $\overline{R}$ has been queried at most once before. However, due to this freedom it is technically more difficult to construct a target randomness secure scheme than a UMUF-GMA secure scheme. Informally, when constructing a UMUF-GMA secure scheme one might directly use an 'all-but-one assumption' where the simulator can easily invert a function f for all but a single output value. This is possible because we give the adversary the target forgery $\overline{M}$ that *cannot be sent to the signing oracle* as the security definition requires $\overline{M} \notin \{M_1, \ldots, M_q\}$. The signature must simply be set up such that the target message $\overline{M}$ exactly corresponds to the one value that the simulator cannot invert. We cannot transfer this technique to target randomness secure schemes because here the adversary might *re-use the target randomness* $\overline{R}$ in his signature queries. On the one hand, the simulator *must be able* to answer all signature queries even the one with $R_i = \overline{R}$. On the other hand it *must not be able* to construct the forgery by itself as it wants to extract a solution to an underlying problem from it. We must setup the parameters such that the simulator is not required to invert this function for the signature query (M_i, R_i) with $R_i = \overline{R}$. The idea is to make the inversion also depend on the message M such that only for M_i the simulator can produce a signature without actually inverting. For all other $M \neq M_i$ this must not be possible.

2.2 From Weakly to Fully Secure Schemes

There exists a well-known transformation by Shamir and Tauman [31] for constructing fully secure schemes from weakly secure signature schemes to using chameleon hash functions [23].[3] It will be applied to our final weakly secure signature scheme to yield a fully secure one. The basic idea is to first use the message as input to a (randomized) chameleon hash function and then sign the output using the weakly secure signature scheme. Signatures consist of the so produced signature and the randomness used for the computation of the chameleon hash. There exist chameleon hash functions that are secure under the RSA [21] or the factoring assumption [31]. Since the factoring assumption is weaker than the RSA assumption we can utilize this transformation without making additional complexity assumptions. In both cases, the overhead amounts to an additional element in the secret key and the public key. We can easily extend the Shamir-Tauman result by showing that if 1) the chameleon hash function has slightly stronger security guarantees than required by the standard definition and 2) the weakly secure signature scheme is strongly secure then the resulting fully secure scheme is strongly secure too. We again stress that most of the existing chameleon hash functions are secure in this strong sense. The proof of the following theorem is straight-forward and, for space reasons, appears in the full version.

Theorem 1. *If the underlying weakly secure signature scheme is strongly secure and the chameleon hash function also guarantees that it is hard (given only the*

[3] In 1999, Gennaro-Halevi-Rabin also proposed a similar but less general solution for their signature scheme [16].

public parameters) to compute two distinct random values that make any message map to the same output value, then the Shamir-Tauman transformation produces fully and strongly *secure signature schemes.*

3 A New Transformation to Weakly Secure Signature Schemes

Let us now present our new transformation that maps `EMURUF-GMRA` secure signature schemes to `EMUF-GMA` secure schemes. In Section 4 we then present a new RSA-based `EMURUF-GMRA` secure signature scheme. First, we fix some additional notation.

Encoding Function. In the following we will regularly produce signatures on strings $S \in \{0,1\}^l$ and on their prefixes. These strings will naturally be interpreted as integers. Now, if $s_l = 0$, S^l and S^{l-1} obviously map to the same integer. However, our proof technique requires that these strings map to different values. To accomplish this we apply an injective and invertible encoding function $\mathsf{enc} : \{0,1\}^{\leq l} \to \{0,1\}^{l+1} \setminus \{0^{k+1}, 0^k 1\}$ that maps to fixed-size outputs. Given an input string $S \in \{0,1\}^{\leq l}$, enc first prepends a 1 and subsequently leading zeros until the result has length $l+1$: $\mathsf{enc}(S) = 0^{l-|S|}1s_1 \dots s_{|S|}$. In the following, we will denote with $R^{\underline{i}} \in \{0,1\}^{l+1}$, $i \in [1;l]$ the string $R^{\underline{i}} = \mathsf{enc}(R^i) = \mathsf{enc}(r_1 \dots r_i)$. It is easy to see that we now always have $R^{\underline{i}} \neq R^{\underline{j}}$ for $j \in [1;l]$ and $i \neq j$.

Target Randomness Secure Signature Scheme => Weakly Secure Signature Scheme. The final signature σ consist of l distinct signatures $\sigma = (\sigma_1, \dots, \sigma_l)$. Each of the single signatures is *on the same message* M. We use *the prefixes of the randomness* R to modify the signature generation, such that $\sigma_i = \mathsf{Sign}(SK, M, R^{\underline{i}} \oplus X)$ for all $i \in [1;l]$. Let us go into more detail.

Let $\mathcal{S}_{\mathrm{tRand}} = (\mathsf{KeyGen}_{\mathrm{tRand}}, \mathsf{Sign}_{\mathrm{tRand}}, \mathsf{Verify}_{\mathrm{tRand}})$ be a probabilistic and `EMURUF-GMRA` secure signature scheme. Then we can construct the `EMUF-GMA` secure signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ as follows.

- $\mathsf{KeyGen}(1^\kappa)$: run $\mathsf{KeyGen}_{\mathrm{tRand}}(1^\kappa)$ and obtain a key pair $(PK_{\mathrm{tRand}}, SK_{\mathrm{tRand}})$ for the signature scheme. Next, draw a random $X \in \{0,1\}^{l+1}$. The scheme's secret key is $SK = SK_{\mathrm{tRand}}$, its public key is $PK = (PK_{\mathrm{tRand}}, X)$.
- $\mathsf{Sign}(SK, M, R)$: to obtain a signature on message $M \in \{0,1\}^l$ using randomness $R \in \{0,1\}^l$, compute the l-tuple $\sigma = (\sigma_1, \dots, \sigma_l)$ s.t. for each $i \in [1;l]$ the i-th component is computed as $\sigma_i = \mathsf{Sign}_{\mathrm{tRand}}(SK_{\mathrm{tRand}}, M, R^{\underline{i}} \oplus X)$.
- $\mathsf{Verify}(PK, M, \sigma, R)$: parse σ as $\sigma_1, \dots, \sigma_l$. If it holds that $M, R \in \{0,1\}^l$ and $\bigwedge_{i=1}^{l} \mathsf{Verify}(PK, M, \sigma_i, R^{\underline{i}} \oplus X) = 1$ output 1, otherwise 0.

Theorem 2. *Let $\mathcal{S}_{tRand}$ be a (q', t', ϵ')-secure probabilistic signature scheme that is secure under* `EMURUF-GMRA` *attacks. Then the application of the above transformation produces a* `EMUF-GMA` *signature scheme $\mathcal{S}$ that is (q, t, ϵ)-secure under generic chosen message attacks provided that*

$$q = q',\ t \approx t',\ \epsilon \leq 2(q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1)\epsilon' + q^2/2^l$$

for $q \geq 1$ *and*

$$q = q',\ t \approx t',\ \epsilon \leq 2\epsilon'$$

for $q = 0$. *Moreover, if* $\mathcal{S}_{tRand}$ *is strongly secure so is* $\mathcal{S}$.

Theorem 2 already makes use of our improved security analysis. To prove Theorem 2 we first need to analyze the set of prefixes of q l-bit strings.

Definition 3 (Prefix-Closure). *Let* S *be an* l*-bit string. We use* $\mathsf{precl}(S) = \{S^j |\ j \in [0;l]\} \cup \perp$ *to denote the prefix-closure of* S*, i.e. the set of all prefixes of* S (including the empty string $\perp$). If R is a set of q l-bit strings, i.e. $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ for $q \in \mathbb{N}$ and $q > 0$, we call $\mathsf{precl}(R) = \bigcup_{i=1}^{q} \mathsf{precl}(R_i)$ the prefix-closure of R. It is the set of all the prefixes of all the R_i. In case $q = 0$, we define $\mathsf{precl}(R) = \{\perp\}$.

Definition 4 (Co-Path of Prefix-Closure). *Let* $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ *for* $q \in \mathbb{N}, q > 0$ *and* $\mathsf{precl}(R)$ *be the prefix-closure of* R*. Let* $\mathcal{Z}_{\mathsf{precl}(R)}$ *denote the set of all strings* $z = z_1 \ldots z_k$ *with* $k \in [1;l]$ *such that* $z_1 \ldots z_{k-1} \in \mathsf{precl}(R)$ *but* $z \notin \mathsf{precl}(R)$*. We say that* $\mathcal{Z}_{\mathsf{precl}(R)}$ *is the co-path of* $\mathsf{precl}(R)$*. For* $q = 0$*, we define* $\mathcal{Z}_{\mathsf{precl}(R)} = \{0,1\}$.

In prefix-based security proofs it is essential to bound the *maximal* size of the co-path of the prefix-closure. Roughly, in the proof the simulator chooses an element z of $\mathcal{Z}_{\mathsf{precl}(R)}$ uniformly at random. This element is used to embed the complexity challenge. With probability $\geq 1/|\mathcal{Z}_{\mathsf{precl}(R)}|$ z will be a prefix of the randomness in the forgery[4]. Thus the simulator's success probability is $\geq 1/|\mathcal{Z}_{\mathsf{precl}(R)}|$. This accounts for the security loss in prefix-based proofs. The existing results upper bound $|\mathcal{Z}_{\mathsf{precl}(R)}|$ simply as $|\mathcal{Z}_{\mathsf{precl}(R)}| \leq ql$. Subsequently, in Theorem 3, we present a more precise analysis. But first we need to analyze the worst-case size of the prefix-closure of R.

Lemma 1. *Let* $q, l \in \mathbb{N}$ $q, l > 0$*. Let* $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ *be an arbitrary set of* q l*-bit strings. Then it holds that*

$$\max_R |\mathsf{precl}(R)| = q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1$$

Proof. We will show which properties R must fulfill to have a maximum size prefix-closure. For convenience we partition $\mathsf{precl}(R) = \mathsf{precl}_{\leq}(R) \cup \mathsf{precl}_{>}(R)$ depending on q. We consider the set $\mathsf{precl}_{\leq}(R)$ of prefixes with length smaller than or equal to $\lfloor \log(q) \rfloor$ and the set of prefixes $\mathsf{precl}_{>}(R)$ that are longer than $\lfloor \log(q) \rfloor$-bits separately. If we maximize both sets, $|\mathsf{precl}(R)| = |\mathsf{precl}_{\leq}(R)| + |\mathsf{precl}_{>}(R)|$ is maximal too.

For $q > 1$ it is clear that there are 'prefix-collisions', i.e. there are $R_j, R_{j'}$ with $R_j \neq R_{j'}$ such that $\mathsf{precl}(R_j) \cap \mathsf{precl}(R_{j'}) \neq \emptyset$. (In fact the pigeon-hole principle shows that for $q > 2^w$ and $w \geq 0$ there is always a pair of distinct indices $j, j' \in [1;q]$ with $|\mathsf{precl}(R_j) \cap \mathsf{precl}(R_{j'})| = w + 1$.) The size of $\mathsf{precl}_{\leq}(R)$ is maximal if i) despite of these collisions the prefixes of the R_i cover all prefixes lower

[4] In HW and BTK, z must be a prefix of the forgery's message.

than or equal to $\lfloor \log(q) \rfloor$, i.e. $\mathsf{precl}_{\leq}(R) = \{0,1\}^{\leq \lfloor \log(q) \rfloor}$. Thus, the maximal size of $\mathsf{precl}_{\leq}(R)$ is $|\mathsf{precl}_{\leq}(R)| \leq \sum_{i=0}^{\lfloor \log(q) \rfloor} = 2^{\lfloor \log(q) \rfloor + 1} - 1$. To analyze $\mathsf{precl}_{>}(R)$ observe that if $2^{\lfloor \log(q) \rfloor + 1} > q$ ii) there need not be any prefix-collisions among the prefixes of length greater than $\lfloor \log(q) \rfloor$. In this case every $\mathsf{precl}(R_i)$ adds the maximal number, $l - \lfloor \log(q) \rfloor$, of additional prefixes to $\mathsf{precl}(R)$. Thus $|\mathsf{precl}_{>}(R)| \leq q(l - \lfloor \log(q) \rfloor)$ is maximal and $|\mathsf{precl}(R)| \leq 2^{\lfloor \log(q) \rfloor + 1} - 1 + q(l - \lfloor \log(q) \rfloor)$.

Theorem 3. *Let $q, l \in \mathbb{N}$ $q, l > 0$. Let $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ be an arbitrary set of q l-bit strings. Then it holds that*

$$q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor} \leq \max_R \left| \mathcal{Z}_{\mathsf{precl}(R)} \right| \leq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1.$$

Proof. To prove the upper-bound observe that for $q \geq 1$ we always have that $|\mathsf{precl}(R)| \geq |\mathcal{Z}_{\mathsf{precl}(R)}|$. This directly follows from the definition of $\mathsf{precl}(R)$. Lemma 1 gives the maximal size of $\mathsf{precl}(R)$. To show the lower bound recall the construction of sets $R = \{R_1, \ldots, R_q\}$ with maximal sized prefix-closure. For $\mathsf{precl}(R)$ to have maximal cardinality we must have that all prefixes lower than $\lfloor \log(q) \rfloor$ are in $\mathsf{precl}(R)$, i.e. $\{0,1\}^{\leq \lfloor \log(q) \rfloor} \subset \mathsf{precl}(R)$. However, this means that for all prefixes with length i strictly lower than $\lfloor \log(q) \rfloor$ we cannot find a corresponding element in $\mathcal{Z}_{\mathsf{precl}(R)}$. This is because all prefixes of length $i+1$ are already in $\mathsf{precl}(R)$ and by definition cannot also be in $\mathcal{Z}_{\mathsf{precl}(R)}$. Thus, for a maximum size prefix-closure we have for the size of the corresponding co-path:

$$|\mathcal{Z}_{\mathsf{precl}(R)}| \geq \max_R \{|\mathsf{precl}(R)|\} - \sum_{j=0}^{\lfloor \log(q) \rfloor - 1} 2^j = q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor}.$$

Now, the maximum size of $|\mathcal{Z}_{\mathsf{precl}(R)}|$ over all R must be at least as large as $q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor}$. This concludes the proof of Theorem 3.

Discussion. Since $|\mathcal{Z}_{\mathsf{precl}(R)}| \leq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1 \leq q(l - \lfloor \log(q) \rfloor + 2)$ and both l and q are polynomials in the security parameter we obtain a clear improvement, in particular for large q. Still q is the dominant factor of the security loss. In their paper, HW also gave a new prefix-based security proof for the CDH-based Waters signature scheme [33]. Before that, Hofheinz and Kiltz (HK) have already proposed *asymptotically* better bounds for the security loss in the Waters scheme. They give a new security analyses of the Waters hash function showing that the security loss is in $O(q\sqrt{l})$ [19]. The original reduction by Waters accounts for a security loss of $8q(l+1)$. The HK improvement relies on random walks and essentially exploits that the message bits of the prefixes can be processed independently as variables of a linear function in the exponents of the hash function's group elements. However, there does not seem to be a general way to transfer their approach to signature scheme like the HW scheme where prefixes are first mapped to prime numbers in a non-linear way. Although our improvements are quantitatively smaller our result 1) provides a *non-asymptotic*, i.e. concrete, bound on the security loss and 2) works for prefix-based technique

in general and thus also applies to the (original) Waters, Hohenberger-Waters, and Hofheinz-Jager-Kiltz scheme and the general transformation presented by Brakerski and Tauman Kalai.

3.1 Proof of Theorem 2

We are now ready to prove the security of our transformation.

Proof. Assume $\mathcal{S}$ is not secure and let $\mathcal{A}$ be a successful attacker against $\mathcal{S}$. Then we can build a simulator $\mathcal{B}$ that uses $\mathcal{A}$ in a black box manner to break the security of $\mathcal{S}_{\text{tRand}}$. Let $\tilde{R}$ be the first message (the target randomness) received from $\mathcal{B}$'s challenger and let $M_1 \ldots, M_q$ be $\mathcal{A}$'s signature queries. In the first step $\mathcal{B}$ draws q random values $R_1, \ldots, R_q \in \{0,1\}^l$. With overwhelming probability these values are all distinct: a simple union bound shows that a collision occurs with probability $\leq q^2/2^l$.

In the next step, $\mathcal{B}$ draws a random coin $y \xleftarrow{\$} \{0,1\}$ indicating whether $\mathcal{A}$ will re-use any of the R_i as randomness in the forgery. According to y, $\mathcal{B}$ will setup the public parameters in two different ways.

If $y = 0$, $\mathcal{B}$ assumes that $\mathcal{A}$ will not re-use any of the R_i in the forgery. Observe that $\overline{R} \notin \{R_1, \ldots, R_q\}$ implies that there must be at least one prefix $\overline{R}^j$ with $\overline{R}^j \notin \mathsf{precl}(R)$. By construction of $\mathcal{Z}_{\mathsf{precl}(R)}$ this means that there exists a prefix $\overline{R}^v$ with $\overline{R}^v \in \mathcal{Z}_{\mathsf{precl}(R)}$. In the first step, $\mathcal{B}$ guesses this $\overline{R}^v$ upfront by drawing a random string $z \xleftarrow{\$} \mathcal{Z}_{\mathsf{precl}(R)}$. Then $\mathcal{B}$ computes $X \in \{0,1\}^{l+1}$ as $X = \mathsf{enc}(z) \oplus \tilde{R}$. Next, $\mathcal{B}$ sends ql signature queries $\{(M_i, T_i^j)\}_{i\in[1;q],j\in[1;l]}$ to the challenger with $T_i^j = R_i^j \oplus X$ for all $i \in [1;q]$. The challenger answers with a public key PK_{tRand} and ql signatures $\{\sigma_{i,j}\}_{i\in[1;q],j\in[1;l]}$ on the given messages such that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, M_i, \sigma_{i,j}, T_i^j) = 1$ for all $i \in [1;q], j \in [1;l]$. In the next step $\mathcal{B}$ sends $PK_{\text{weak}} = (PK_{\text{tRand}}, X)$ and q randomness/signature pairs $(\Sigma_1, R_1), \ldots, (\Sigma_q, R_q)$ to $\mathcal{A}$. Each Σ_i consist of l signatures of the target randomness secure scheme: $\Sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,l})$ for all $i \in [1;q]$. By assumption $\mathcal{A}$ then outputs a forgery $(\overline{M}, \overline{\Sigma} = (\overline{\sigma_1}, \ldots, \overline{\sigma_l}), \overline{R})$ with $\overline{R} \notin \{R_1, \ldots, R_q\}$. Now with probability $\geq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1$ $\mathcal{B}$'s guess of z is right. In this case there exists a $v \in [1;l]$ with $\overline{R}^v = z$ and it holds that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, \overline{M}, \overline{\sigma_v}, \overline{R}^v \oplus X) = 1$. Since by definition we have that $\overline{R}^v \oplus X = \mathsf{enc}(z) \oplus X = \tilde{R}$, $(\overline{M}, \overline{\sigma_v})$ breaks the security of the target randomness secure signature scheme.

In case $y = 1$, $\mathcal{B}$ assumes that $\mathcal{A}$ will re-use any of the R_i in the forgery and guesses the signature index of the corresponding randomness upfront by drawing $w \xleftarrow{\$} [1;q]$. Next, $\mathcal{B}$ computes $X = R_w^l \oplus \tilde{R}$ and T_i^j as $T_i^j = R_i^j \oplus X$ for all $i \in [1;q], j \in [1;l]$. Then $\{(M_i, T_i^j)\}_{i\in[1;q],j\in[1;l]}$ is given to the challenger who in turn answers with the public key PK_{tRand} and ql signatures $\{\sigma_{i,j}\}_{i\in[1;q],j\in[1;l]}$. By assumption it must always be the case that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, M_i, \sigma_{i,j}, R_i^j \oplus X) = 1$ with $i \in [1;q], j \in [1;l]$. As before $\mathcal{B}$ now gives $PK_{\text{weak}} = (PK_{\text{tRand}}, X)$

and q signatures $\Sigma_1, \ldots, \Sigma_l$ with $\Sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,l}, R_i)$, $i \in [1; q]$ to $\mathcal{A}$. However, this time $\mathcal{A}$ outputs a forgery $(\overline{M}, \overline{\Sigma} = (\overline{\sigma_1}, \ldots, \overline{\sigma_l}), \overline{R})$ with $\overline{R} \in \{R_1, \ldots, R_q\}$. With probability $\geq q$, we have that $\overline{R} = R_w$. Then $(\overline{M}, \overline{\sigma_l})$ breaks the security of the underlying `EMURUF-GMRA` secure signature scheme since $\overline{R}^l \oplus X = R_w^l \oplus X = R$. Observe that only in case $y = 1$ we need that the R_i are all distinct to comply with the requirement of the `EMURUF-GMRA` security game, i.e. the target randomness is only queried at most once to the signature oracle.

With probability $\geq 1/2$, $\mathcal{B}$'s guess of y is correct. Observe that by construction $\mathcal{A}$ cannot tell apart the values produced by $\mathcal{B}$ from those of the original attack game. This concludes the proof of Theorem 2. Also note that at no point in the proof we rely on the fact that $\overline{M} \notin \{M_1, \ldots, M_q\}$. Thus the resulting signature scheme is strongly secure.

4 A Target Randomness Secure Signature Scheme Based on the RSA Assumption

Let us now present our new RSA-based `EMURUF-GMRA` secure signature scheme. For simplicity we focus on the practically most relevant case of balanced, safe RSA moduli with prime public exponent but we stress that the signature scheme can be instantiated in general RSA groups as well.

- $\mathsf{KeyGen}(1^\kappa)$: Choose two large balanced and safe primes $\hat{p} = 2\overline{p}+1, \hat{q} = 2\overline{q}+1$ with $|\overline{p}| = |\overline{q}|, \overline{p} > \overline{q}$ and $2^{l+1} < \overline{p} \cdot \overline{q}$. Set $n = \hat{p}\hat{q}$ and $SK = \hat{p}, \hat{q}$. Next choose a random $X' \xleftarrow{\$} \{0,1\}^{l+1}$. We will also need a target collision-resistant function $p : \{0,1\}^{l+1} \rightarrow \mathsf{Primes}_{l+1}$ that maps strings to the set of odd prime numbers $\mathsf{Primes}_{l+1} \subset [1; 2^{l+1} - 1]$. We use the following instantiation. Choose a key s' for a pseudo-random function $t : \{0,1\}^* \rightarrow \{0,1\}^{l+1}$. Whenever we want to evaluate p on input $R^i \in \{0,1\}^{l+1}$ we continually increment *the resolving index* $ind \in \{0,1\}^*$, which is initially set to $ind := 0$, until $t(ind||R^i) \oplus X$ is prime.
  For a proof that p indeed is collision-free for the first polynomial many input values we refer to [21] or [18].[5] Finally, choose two random generators a, b of a subgroup S of $\mathbb{Z}_n^*$ with $<a>=<b>= S \subset \mathbb{Z}_n^*$ and $|S| = \overline{p} \cdot \overline{q}$. Publish $PK = (a, b, n, s', X')$.
- $\mathsf{Sign}(SK, M, R)$: To sign a message $M \in \{0,1\}^{l+1}$, choose a random $R \in \{0,1\}^{l+1}$ and compute $s = (ab^M)^{1/p(R)} \bmod n$. Output σ and R.
- $\mathsf{Verify}(PK, M, \sigma, R)$: If $\sigma^{p(R)} = ab^M \bmod n$ output 1, else output 0.

Theorem 4. *If the RSA assumption is secure then the above construction yields a target randomness secure probabilistic signature scheme. Moreover, the security reduction tightly reduces to the RSA assumption.*

[5] Basically, one first shows that with overwhelming probability one can always find a prime while $ind \leq (l+1)^2$ – otherwise the PRF could be distinguished from a truly random function. In the second step, one argues that the probability to find a collision only after $q(l+1)^2$ evaluations of the PRF must be negligible as otherwise the PRF construction could again easily be distinguished from a truly random function.

Proof. Assume the simulator $\mathcal{B}$ is given an RSA challenge (n, u, α). At first $\mathcal{B}$ draws a random key s' for the pseudo-random permutation and a random $R \in \{0,1\}^l$. Next it constructs X' such that $\alpha = t(ind||R^{\underline{l}}) \oplus X' = p(R)$. Then $\mathcal{B}$ sends R to the forger $\mathcal{A}$. As an answer $\mathcal{B}$ receives q signature queries $(M_1, R_1), \ldots, (M_q, R_q)$ from $\mathcal{A}$. With these values $\mathcal{B}$ can now set up the rest of the public key. To this end, $\mathcal{B}$ draws $h_a, h_b \in [1; (n-1)/4]$. In the following, $\mathcal{B}$ considers two cases.

- If there is no R_i with $R_i = R$ (Case 1), $\mathcal{B}$ computes $a = u^{h_a \prod_{i=1}^{q} p(R_i)}$ and $b = u^{h_b \prod_{i=1}^{q} p(R_i)}$.
- If there exists such a value (Case 2) so let $j \in [1; q]$ be the corresponding index with that $R_j = R$. Now, $\mathcal{B}$ computes $a = u^{h_a \prod_{i=1}^{q} p(R_i) + h_b M_j \prod_{i=1, i\neq j}^{q} p(R_i)}$ and $b = u^{-h_b \prod_{i=1, i\neq j}^{q} p(R_i)}$.

Observe that in both cases a and b are distributed almost like in the original attack game. This is because h_a, h_b are almost distributed uniformly in $[1; \phi(n)]$. The probability for a value $h \xleftarrow{\$} [1; (n-1)/4]$ not to be in $[1; \overline{p} \cdot \overline{q}]$ is

$$\Pr[h \xleftarrow{\$} [1; (n-1)/4], h \notin [1; \overline{p} \cdot \overline{q}]] \leq (\overline{p} + \overline{q})/(2\overline{pq} + \overline{p} + \overline{q}) < 1/(\overline{q} + 1) < 1/2^{|\overline{p}|-2}$$

and thus negligible.

$\mathcal{B}$ can now easily answer the signature queries by computing signatures σ_k on (M_k, R_k) for all $k \in [1; q]$ as follows.

- In Case 1, $\mathcal{B}$ computes $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i)} \cdot u^{M_k h_b \prod_{i=1, i\neq k}^{q} p(R_i)}$.
- If it holds that $R_k \neq R$ in Case 2, $\mathcal{B}$ computes the queried signature σ_k as follows: $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i) + M_j h_b \prod_{i=1, i\neq j,k}^{q} p(R_i)} \cdot u^{-M_k h_b \prod_{i=1, i\neq j,k}^{q} p(R_i)}$.
- If $R_k = R$ (or $k = j$) in Case 2, $\mathcal{B}$ computes $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i)}$.

Observe that these values perfectly simulate the original attack game.

The last case solves the problem that we are faced with when designing `EMURUF-GMRA` secure schemes (see the discussion in Section 2.1). If $M_k \neq M_j$ and $R_k = R$, $\mathcal{B}$ cannot compute a signature since for the exponent e it holds that $e = h_a \prod_{i=1}^{q} p(R_i) + h_b(M_j - M_k) \prod_{i=1, i\neq k}^{q} p(R_i)$ which implies $p(R) \not| e$ and thus $\mathcal{B}$ would need to compute $p(R)$ roots what, by the RSA assumption, is not feasible. However, if $M_j = M_k$ we get that $e = h_a \prod_{i=1}^{q} p(R_i)$ and clearly $p(R)|e$. This time $\mathcal{B}$ can generate a signature by just exponentiating.

Now when $\mathcal{A}$ outputs the forgery $(\overline{M}, \overline{\sigma}, R)$, $\mathcal{B}$ can extract a solution to the RSA challenge as follows.

In Case 1, the verification equation gives us $\overline{\sigma}^{p(R)} = u^{(h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)}$. Let us analyze the probability for the event $p(R)|(h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)$. Since p is target collision-resistant the $p(R_i)$ are all distinct from $p(R)$ and we only must analyze whether $p(R)|(h_a + \overline{M} h_b)$. Since $\mathcal{A}$ never gets to see u in the clear, h_a and h_b are perfectly hidden from her view. As 3 is the smallest prime number $p(R)$ can take on, the probability for $\mathcal{A}$ to output $\overline{M}$ with $p(R)|(h_a + \overline{M} h_b)$ is at most $1/3$. We so have with probability $\geq 2/3$ that $\gcd(p(R), (h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)) = 1$.

In this case, we can easily compute two integers w_1 and w_2 (using Euclidean algorithm) such that $w_1 p(R) + w_2(h_a + \overline{M}h_b)\prod_{i=1}^{q} p(R_i) = 1$. It then holds that $u = u^{w_1 p(R) + w_2(h_a+\overline{M}h_b)\prod_{i=1}^{q} p(R_i)} = u^{w_1 \cdot p(R)} \cdot \overline{\sigma}^{w_2 \cdot p(R)}$. Therefore the final solution to the RSA challenge is $u^{1/p(R)} = u^{1/\alpha} = u^{w_1} \cdot \overline{\sigma}^{w_2}$.

In Case 2, we can show with the same arguments as above that with probability at least 2/3 we have $p(R) \not| \left(h_a \prod_{i=1}^{q} p(R_i) + (M_j - \overline{M})h_b \prod_{i=1, i\neq j}^{q} p(R_i)\right)$. Using the same techniques as before $\mathcal{B}$ finds the corresponding values w_1 and w_2 such that $w_1 p(R) + w_2 \left(h_a \prod_{i=1}^{q} p(R_i) + (M_j - \overline{M})h_b \prod_{i=1, i\neq j}^{q} p(R_i)\right) = 1$. The final solution to the RSA challenge is $u^{1/\alpha} = u^{w_1} \cdot \overline{\sigma}^{w_2}$.

This concludes the proof of Theorem 4.

5 Accumulation of Signature Schemes

When applying our transformation from target randomness secure signature schemes to weakly secure schemes to the above signature scheme we get signatures that consist of l group elements.

We now show how to accumulate signatures of the type $\sigma' = (\sigma'_1, \ldots, \sigma'_l)$ where for each $i \in [1; q]$ $\sigma'_i = (ab^M)^{1/p(R^i)} \bmod n$ to a *single* element $\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} p(R^i)} \bmod n$. The accumulation technique used here is a direct application of the extended Euclidean algorithm. Observe that our accumulation technique does not require knowledge of the secret key.

Lemma 2. *Let PK, M, R, and $\sigma'_1 = (ab^M)^{1/p(R^1)}, \ldots, \sigma'_l = (ab^M)^{1/p(R^l)} \in \mathbb{Z}_n^*$ with $l \in \mathbb{N}$ be given. Then we can easily compute $\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} p(R^i)} \bmod n$.*

Proof. Since for $i \neq j$ it holds that $R^i \neq R^j$ and because of the properties of $p(\cdot)$ we have that the $e_1 = p(R^1), \ldots, e_l = p(R^l)$ are distinct primes. Let $\bar{e}_i = \prod_{j=1, j\neq i}^{l} e_i$ and $e = \prod_{i=1}^{l} e_i$. By construction we have $\gcd(\bar{e}_1, \ldots, \bar{e}_l) = 1$. Next we can use extended Euclidean algorithm to find $a_1, \ldots, a_l \in \mathbb{Z}$ such that $\gcd(\bar{e}_1, \ldots, \bar{e}_l) = \sum_{i=1}^{l} a_i \bar{e}_i = 1$. We have $(ab^M) = (ab^M)^{\sum_{i=1}^{l} a_i \bar{e}_i} \bmod n$. As it holds for all e_i that $\gcd(e_i, \phi(n)) = 1$ we can finally find the e-th root of ab^M as

$$\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} e_i} = \prod_{i=1}^{l} (ab^M)_i^{a_i/e_i} = \prod_{i=1}^{l} \sigma'^{a_i} \bmod n.$$

Lemma 3. *Given PK, M, R, and $\Sigma' = (ab^M)^{1/(\prod_{i=1}^{l} p(R^i))} \in \mathbb{Z}_n^*$ we can easily compute $\sigma'_1 = (ab^M)^{1/p(R^1)}, \ldots, \sigma'_l = (ab^M)^{1/p(R^l)} \in \mathbb{Z}_n^*$.*

Proof. For all $j \in \{1, \ldots, l\}$, if we want to find the j-th component of the basic scheme we simply compute

$$\sigma'_j = \Sigma'^{\prod_{i=1, i\neq j}^{l} p(R^i)} \bmod n.$$

The previous two lemmas show that both signature descriptions, the l-element signature and the accumulated signature, are equivalent. The above technique can easily be adapted to accumulate a variant of the Gennaro-Halevi-Rabin [16] signature scheme by Brakerski-Tauman-Kalai [6]. The result is the Hohenberger-Waters signature. In the above we simply have to set $(ab^M) := u$ and $R := M$.

6 Final RSA Signature Scheme

Like the Hohenberger-Waters scheme, our final RSA-based `EMUF-GMA` secure signature scheme features a built-in accumulation process.

- $\mathsf{KeyGen}(1^\kappa)$: The key generation algorithm is exactly the same as in the scheme of Section 4.
- $\mathsf{Sign}(SK, M, R)$: To sign a message $M \in \{0,1\}^l$ choose a random $R \in \{0,1\}^l$ and compute $\sigma = (ab^M)^{1/(\prod_{i=1}^{l} p(R^i))} \bmod n$. Output (σ, R).
- $\mathsf{Verify}(PK, M, \sigma, R)$: If $\sigma^{\prod_{i=1}^{l} p(R^i)} = ab^M \bmod n$ output 1, else output 0.

Theorem 5. *Applying the Shamir-Tauman transformation to the above signature scheme as presented in Section 2.2 gives us a strongly and fully secure signature scheme under the RSA assumption.*

6.1 Comparison with the Hohenberger-Waters Scheme

Our RSA-based signature scheme presents an alternative to the Hohenberger-Waters signature scheme. The times for signature generation and verification are comparable. As a drawback, the size of our signatures is longer than the Hohenberger-Waters signature. Besides a group element in $\mathbb{Z}_n^*$, our scheme additionally contains a random string R where $|R|$ is ≈ 160. However, when signing long messages our scheme requires weaker security assumptions than the Hohenberger-Waters scheme. Let us explain this in more detail. To extend the input domain of a signature scheme, one usually applies a collision-resistant hash function and signs the hash value of the input message. Alternatively, one can also use a primitive called target collision resistant hash function (TCR) (or universal hash function) [26]. TCRs are fundamentally weaker primitives than collision-resistant hash functions, since on the one hand there exist efficient constructions of TCRs from one-way functions [28, 26, 17] but on the other hand collision resistant hash function cannot be constructed from one-way functions using black-box constructions [32]. There exists a standard transformation for signature schemes by Bellare and Rogaway that allows to exchange the collision-resistant hash function with a target collision-resistant function when signing long messages [2]. Usually this would require an additional random element to be embedded in the signature – the key of the TCR. However, following similar arguments as Mironov [24] we can re-use the message-independent randomness R of our signature scheme for this purpose. Therefore, our RSA-based signature scheme can use target collision resistant hash functions without any modification for domain extension.

References

1. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In *EUROCRYPT*, pages 399–416, 1996.
2. Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 1997.
3. Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 201–216. Springer, 2007.
4. Daniel J. Bernstein. Proving tight security for Rabin-Williams signatures. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2008.
5. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.
6. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086, February 2010. `http://eprint.iacr.org/`, version from 13th February 2010.
7. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002.
8. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.
9. Scott Contini, Arjen K. Lenstra, and Ron Steinfeld. VSH, an efficient and provable collision-resistant hash function. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 165–182. Springer, 2006.
10. Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2000.
11. Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287. Springer, 2002.
12. Ronald Cramer, Ivan Damgård, and Torben P. Pedersen. Efficient and provable security amplifications. In T. Mark A. Lomas, editor, *Security Protocols Workshop*, volume 1189 of *Lecture Notes in Computer Science*, pages 101–109. Springer, 1996.
13. Ronald Cramer and Victor Shoup. Signature schemes based on the Strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.
14. Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *J. Cryptology*, 9(1):35–67, 1996.
15. Marc Fischlin. The Cramer-Shoup Strong-RSA signature scheme revisited. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 116–129. Springer, 2003.
16. Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT*, pages 123–139, 1999.
17. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

18. Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 647–666. Springer, 2011.
19. Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2008.
20. Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 333–350. Springer, 2009.
21. Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 654–670. Springer, 2009.
22. Qiong Huang, Duncan S. Wong, Jin Li, and Yiming Zhao. Generic transformation from weakly to strongly unforgeable signatures. *J. Comput. Sci. Technol.*, 23(2):240–252, 2008.
23. Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS*. The Internet Society, 2000.
24. Ilya Mironov. Collision-resistant no more: Hash-and-sign paradigm revisited. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 140–156. Springer, 2006.
25. David Naccache, David Pointcheval, and Jacques Stern. Twin signatures: an alternative to the hash-and-sign paradigm. In *ACM Conference on Computer and Communications Security*, pages 20–27, 2001.
26. Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43. ACM, 1989.
27. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
28. John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394. ACM, 1990.
29. Sven Schäge. Twin signature schemes, revisited. In Josef Pieprzyk and Fangguo Zhang, editors, *ProvSec*, volume 5848 of *Lecture Notes in Computer Science*, pages 104–117. Springer, 2009.
30. Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 189–206. Springer, 2011.
31. Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 355–367. Springer, 2001.
32. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT*, pages 334–345, 1998.
33. Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.
34. Huafei Zhu. New digital signature scheme attaining immunity to adaptive-chosen message attack. *Chinese Journal of Electronics*, 10(4):484–486, October 2001.
35. Huafei Zhu. A formal proof of Zhu's signature scheme. Cryptology ePrint Archive, Report 2003/155, 2003. `http://eprint.iacr.org/`.