Rerandomizable Signatures under Standard Assumption

Sanjit Chatterjee and R. Kabaleeshwaran

Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India
{sanjit, kabaleeshwar}@iisc.ac.in

Abstract. The Camenisch-Lysyanskaya rerandomizable signature (CL-RRS) scheme is an important tool in the construction of privacy preserving protocols. One of the limitations of CL-RRS is that the signature size is linear in the number of messages to be signed. In 2016, Pointcheval-Sanders introduced a variant of rerandomizable signature (PS-RRS) scheme which removes the above limitation. However, the security of PS-RRS scheme was proved under an interactive assumption. In 2018, Pointcheval-Sanders improved this to give a reduction under a parameterized assumption.
In 2012, Gerbush et al. introduced the dual-form signature technique to remove the dependency on interactive/parameterized assumption. They applied this technique on the CL-RRS scheme (for single message) and proved its unforgeability under static assumptions instead of the interactive assumption used in the original work but in the symmetric composite-order pairing setting.
In this work, we realize a fully rerandomizable signature scheme in the prime order setting without random oracle based on the SXDH assumption. The signature structure is derived from Ghadafi's structure-preserving signature. We first apply the dual-form signature technique to obtain a composite-order variant, called `RRSc`. A signature in `RRSc` consists of only two group elements and is thus independent of the message block length. The security of the proposed scheme is based on subgroup hiding assumptions. Then we use the dual pairing vector space framework to obtain a prime-order variant called `RRS` and prove its security under the SXDH assumption.

1 Introduction

In their seminal work, Camenisch and Lysyanskaya [CL04] introduced a rerandomizable signature (henceforth denoted as CL-RRS) scheme. The rerandomizability property says that, given a signature σ on some message $\boldsymbol{m}$ under the public key PK, anybody can compute another valid signature on the same message which is indistinguishable from the original signature. The rerandomizability property aids in replacing costly zero knowledge proof system in many privacy-preserving protocols. The CL-RRS scheme has an additional desirable property that a signature can be generated on multiple message blocks in a

single invocation of the signing algorithm. Due to these attractive properties, CL-RRS scheme has been used as a building block in many applications such as group signature [BCN+10], anonymous attestation [BFG+13], aggregate signature [LLY13] and E-cash [CPST15].

The main drawbacks of the CL-RRS scheme are (i)unforgeability is proved under the interactive LRSW assumption and (ii) the signature size depends on the length of the message block signed. In 2016, Pointcheval-Sanders [PS16] introduced a new rerandomizable signature (henceforth called PS-RRS) scheme in the Type-3 pairing setting: $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ [GPS08]. They considered the signature space to be $\mathbb{G}$ while the public key comes from $\mathbb{H}$. Separating the signature space from the public key space allows them to optimize the signature size. Using an ℓ-wise independent function in the signature structure, the PS-RRS construction can make the signature length constant. However, the unforgeability of PS-RRS is proved under a new interactive assumption (called as the PS assumption).

In 2018, their follow-up work [PS18] presented a weak unforgeability proof for the PS-RRS scheme under a parameterized (called as q-MSDH-1) assumption. They also modified the original PS-RRS (henceforth called mPS-RRS) scheme a bit and proved its unforgeability under the q-MSDH-1 assumption. However, they could achieve only weak rerandomizability for the mPS-RRS scheme, as one of the random exponent has to be provided explicitly as part of the signature. They also showed that mPS-RRS scheme can be modified to realize full rerandomizability, but only in the random oracle model. [PS18] further described a modified CL-RRS (henceforth called mCL-RRS) scheme under a new parameterized assumption (called as q-MSDH-2), instead of an interactive assumption. However, mCL-RRS achieves only weakly rerandomizability and random oracle is required to argue that it's fully rerandomizable.

A consequence of relying on some parameterized assumption is that one may need to increase the underlying group size to achieve the desired level of security. For example, Cheon [Che06] showed that the q-SDH problem can be solved using $O(\sqrt{p/q})$ group operations, where p is the underlying group order. To achieve the desired discrete-log level of security for q-SDH, one may thus have to increase the underlying group size. This certainly has a negative bearing on efficiency of the scheme which is further carried forward into the applications.

In 2012, Gerbush et al. [GLOW12] introduced an interesting technique called dual-form signature. They applied this technique in the composite order pairing setting to argue security of several signature schemes based on static assumptions. This way they were able to remove the dependency on interactive or parameterized (q-type) assumptions in some existing signature schemes [CL04, BGOY07, BB04]. In particular, they constructed a dual-form variant of Camenisch-Lysyanskaya signature (for the case of single message) in the symmetric composite-order setting. In a follow-up work, Yuen et al. [YCZY14] presented the dual-form Boneh-Boyen signature scheme in the prime-order setting under the SXDH assumption. In [CK18], Chatterjee and Kabaleeshwaran utilized the

dual-form signature technique to get a variant of Boneh-Waters group signature [BW07] under static assumptions instead of parameterized assumption.

While [GLOW12] did achieve a dual-form variant of CL-RRS scheme for single message block, the scheme is instantiated in the composite-order pairing setting. Due to the relative inefficiency [Fre10, Gui13] of the composite-order pairing, a construction in the prime-order setting is usually preferable. Since the PS-RRS scheme is instantiated in the prime-order and has constant size signature, the authors [PS16] demonstrated that it is a better alternative of the dual-form CL-RRS scheme in several privacy-preserving applications, such as group signatures [BCN+10], anonymous credentials [CL04, ASM06]. However, the unforgeability of PS-RRS (or, mPS-RRS) can be proved only under some interactive/parameterized assumption. So in this work we explore the applicability of the dual-form signature technique to realize an RRS scheme with constant size signature as in [PS16, PS18] based on some standard (static) assumptions.

The dual-form signature [GLOW12] consists of two signing algorithms, namely Sign_A and Sign_B that will respectively return two forms of signature both of which verify under the same public key. To argue security, the forgery space is partitioned into two types, namely Type-I and Type-II that respectively correspond to the signatures returned by Sign_A and Sign_B.

The approach that [GLOW12] had taken to argue security of the signature variants under static assumption, consists of two steps. The first step is to construct a dual-form of the signature variants and argue its security under some static assumptions. Next, they obtained the actual signature scheme by removing any one of the Sign_A or Sign_B algorithm. Finally, they argued that the security of the signature scheme is reducible from the security of the dual-form signature variants.

1.1 Our Contribution

We realize a fully rerandomizable signature scheme based on the SXDH assumption without random oracle in the prime order setting. Towards this goal, we first construct a rerandomizable signature scheme (denoted as `RRSc`), whose construction is inspired from [Gha17]'s structure-preserving signature scheme, in the composite-order setting with $N = p_1p_2$. We argue the unforgeability of `RRSc` under subgroup hiding assumptions. Then we convert the above `RRSc` scheme to the prime-order setting (denoted as `RRS`) which is instantiated using the dual pairing vector space (DPVS) framework [Lew12]. We argue the security of `RRS` under the SXDH assumption. We also describe a variant of `RRS` (denoted as `PS-RRS`) constructed from PS-RRS scheme [PS16]. Table 1 compares the proposed rerandomizable signature schemes with the existing ones.

Our approach is similar to the previous works that used the dual form signature technique [Wat09, LJYP14, LPY15, LMPY16]. Rather than first defining a dual-form variant of (rerandomizable) signature as in some of the previous works [GLOW12, YCZY14, CK18], we directly apply the dual-form signature techniques in the unforgeability proof. In other words, we use Sign_A in the ac-

Table 1. Comparing rerandomizable signatures in the standard model.

Scheme	Pairing Setting	Group Order	$\#\sigma$	Rand.	EUF-CMA
CL-RRS [CL04]	Symmetric	prime	$2\ell+1$	Full	LRSW (interactive)
mCL-RRS [PS18]			$2\ell+3$	Weak	q-MSDH-1
PS-RRS [PS16]	Asymmetric	prime	2	Full	PS (interactive)
mPS-RRS [PS18]			2	Weak	q-MSDH-2
DF-CL-RRS ‡ [GLOW12]	Symmetric	composite	3	Full	SGH, Static
`RRSc` §3	Asymmetric	composite	2	Full	SGH
`RRS` §4	Asymmetric	prime	1 †	Full	SXDH

† `RRS` scheme consists of a single signature component, but in the DPVS setting it requires four atomic group elements. ‡ Dual-form of CL-RRS scheme that signs a single message block.

tual scheme construction while Sign_B is used only in the unforgeability proof. Similar to previous results, security is argued using a hybrid argument.

Organization of the paper. In §2, we recall a few definitions that will be used in this paper. In §3 and §4, we present the rerandomizable signature scheme in the composite and prime order setting respectively. In §4.4, we present a variant of rerandomizable signature scheme and provide a comparative analysis in §4.5.

2 Preliminaries

2.1 Notation

For a prime p, $\mathbb{Z}_p^*$ denotes the set of all non-zero elements from $\mathbb{Z}_p$. We denote $a \xleftarrow{\$} A$ to be an element chosen uniformly at random from the non-empty set A. For $n > 1$, $\boldsymbol{b} \in \mathbb{Z}_p^n$ denotes the vector $(b_1, \ldots, b_n)$, where $b_j \in \mathbb{Z}_p$, for all $j \in [1, n]$. For any two vectors $\boldsymbol{b} = (b_1, \ldots, b_n), \boldsymbol{b}^* = (b_1^*, \ldots, b_n^*)$ from $\mathbb{Z}_p^n$, the 'dot' product is denoted as $\boldsymbol{b} \cdot \boldsymbol{b}^*$ which is same as $\boldsymbol{b}(\boldsymbol{b}^*)^\top$, since both are equals to $\sum_{i=1}^n b_i b_i^*$. We denote $GL(n, \mathbb{Z}_p)$ to be the set of all non-singular matrix of order n over $\mathbb{Z}_p$ and A^{-1} to be the inverse of the matrix $A \in GL(n, \mathbb{Z}_p)$. For any matrix M from $\mathbb{Z}_p^{m \times n}$, $M^\top$ denotes the transposition of the matrix M.

2.2 Digital Signature

We recall the definition of digital signature scheme from [CLL+12], which consists of three PPT algorithms.

KeyGen(1^λ) Given the security parameter λ, it returns the key pair (PK, SK).
Sign(SK, m) Given the message m and SK, it returns the signature σ on m.
Ver(PK, m, σ) Given the message and signature pair along with the public key PK, it returns 1 only if σ is a valid signature on the message m under PK.

The digital signature scheme is *correct*, if for all security parameter λ, all $(PK, SK) \leftarrow \mathsf{KeyGen}(1^\lambda)$, all messages m and $\sigma \leftarrow \mathsf{Sign}(SK, m)$, it holds that $\mathsf{Ver}(PK, m, \sigma)$=1.

The security of the digital signature scheme is captured using the existential unforgeability under chosen message attack (EUF-CMA) model [GMR88] which

is defined as follows. Informally, given the public key and polynomial many (in λ) access to the signing oracle, it is hard for an adversary to return a valid forgery (m, σ) such that m is not queried earlier to the signing oracle. Formally, it is defined using the following experiment between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup $\mathcal{C}$ runs the KeyGen to obtain (PK, SK). $\mathcal{A}$ is given with PK.
Queries $\mathcal{A}$ adaptively requests the signature on the message m_i, for $i \in [1, q]$. $\mathcal{C}$ answers each query by computing $\sigma_i = \mathsf{Sign}(SK, m_i)$.
Output Finally, $\mathcal{A}$ returns a message and signature pair (m^*, σ^*).

The advantage of $\mathcal{A}$ (denoted as $Adv_{\mathcal{A}}^{UF}$) is defined to be the probability that $\mathcal{A}$ wins in the above game, i.e., $\mathsf{Ver}(PK, m^*, \sigma^*)=1$ with $m^* \neq m_i$, for all $i \in [1, q]$. A signature scheme is said to be (t, q, ϵ)-secure against existential unforgeability under chosen message attack ((t, q, ϵ)-EUF-CMA secure), if for any t-time adversary $\mathcal{A}$ that makes at most q many signing oracle queries, $Adv_{\mathcal{A}}^{UF} \leq \epsilon$, where t and q are the polynomial functions of λ and ϵ is a negligible function in λ.

2.3 Bilinear Pairing Setting

We recall the definition of bilinear group generator from [Fre10].

Definition 1 *A bilinear group generator $\mathcal{G}$ is a probabilistic polynomial time (PPT) algorithm which takes the security parameter λ as input and outputs (N, G, H, G_T, e, μ), where N is either prime or composite, G, H and G_T are the groups such that $|G| = |H| = k_1 N$ and $|G_T| = k_2 N$ for $k_1, k_2 \in \mathbb{N}$, all the elements of G, H, G_T are of order at most N and $e : G \times H \longrightarrow G_T$ is a bilinear map which satisfies,*

(i) Bilinearity: For all $g, g' \in G$ and $h, h' \in H$, one has $e(g \cdot g', h \cdot h') = e(g, h) \cdot e(g, h') \cdot e(g', h) \cdot e(g', h')$,
(ii) Non degeneracy: If a fixed $g \in G$ satisfies $e(g, h) = 1$ for all $h \in H$, then $g = 1$ and similarly for a fixed element $h \in H$ and
(iii) Computability: The map e is efficiently computable.

The additional information μ is optional and defined as follows. Whenever G and H are prime-order cyclic groups, μ contains their respective generators g and h. Whenever the groups G and H are decomposed into its cyclic subgroups $G_1, \ldots, G_n$ and $H_1, \ldots, H_n$ respectively, μ contains the description of these subgroups and/or their generators.

The bilinear group generator $\mathcal{G}$ is said to be of composite-order (resp. prime-order), if N is composite (resp. prime). In this paper we use both prime-order and composite-order bilinear group settings. Hence for ease of readability, we use the following notation to differentiate between these two settings. In the prime-order setting, we denote $\mathcal{P} = \mathcal{G}$, $\mathbb{G} = G$, $\mathbb{H} = H$, $\mathbb{G}_T = G_T$ and we could obtain only trivial subgroups, hence μ contains the generators g and h of the respective groups $\mathbb{G}$ and $\mathbb{H}$. In the composite-order setting, we denote $\mathcal{G}_N = \mathcal{G}$

and we decompose the groups $G \cong G_1 \oplus \ldots \oplus G_n$ and $H \cong H_1 \oplus \ldots \oplus H_n$ for $N = p_1 \ldots p_n$ with μ containing required subgroup(s) information i.e., μ contains $\{g_i, h_i\}_{i=1}^n$, where g_i (resp. h_i) is the generator of the subgroup G_i (resp. H_i).

The Dual Pairing Vector Space (DPVS) was introduced by [OT08, OT10], though the following definition is taken from [CLL+12]. Here we consider the concrete case of $n = 4$, however, one can define the DPVS for any $n > 1$.

Definition 2 *Given the parameters $p, n = 4$, the dual orthogonal basis generator is denoted as $\mathsf{Dual}(\mathbb{Z}_p^4)$ and it returns two random bases $\mathbb{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_4)$ and $\mathbb{B}^* = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_4^*)$ which are defined from $\mathbb{Z}_p^4$ such that $\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = 0$ mod p, for $i \neq j$ and $\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* = \psi$ mod p, for all $i, j \in [1, 4]$ with $\psi \in \mathbb{Z}_p^*$.*

From the above definition, we denote $\mathcal{P}_{(\perp,4)}$ to be the bilinear group generator which takes the security parameter λ and an integer $n = 4$ as input and outputs $(p, G, H, G_T, e, \{G_i, H_i\}_{i=1}^4, g, h)$. Here $G = G_1 \oplus \ldots \oplus G_4 \approx \mathbb{G}^4$ and $H = H_1 \oplus \ldots \oplus H_4 \approx \mathbb{H}^4$. Also g (resp. h) be the generator of the group $\mathbb{G}$ (resp. $\mathbb{H}$). Let $g^{\boldsymbol{b}_i}$ (resp. $h^{\boldsymbol{b}_j^*}$) be the generator of the subgroup G_i (resp. H_j), for $i, j \in [1, 4]$ such that $e(g^{\boldsymbol{b}_i}, h^{\boldsymbol{b}_j^*}) = 1$, for $i \neq j$ and $e(g^{\boldsymbol{b}_i}, h^{\boldsymbol{b}_i^*}) = e(g, h)^{\psi}$, for $i, j \in [1, 4]$. Any element $\tilde{g} \in G$ can be written as $\tilde{g} = g^{\sum_{i=1}^4 \gamma_i \boldsymbol{b}_i}$, for some $\gamma_i \in \mathbb{Z}_p$. We say that γ_i is the coefficient of the term $\tilde{g}$ with respect to the basis $\boldsymbol{b}_i$, for any $i \in [1, 4]$.

2.4 Complexity Assumptions

Composite-Order Setting Recall that the subgroup hiding (SGH) assumptions in [LW10, CW14] are defined in the symmetric composite-order setting. Whereas, [CGKW18] defined the SGH assumption in the asymmetric composite-order setting with $N = p_1 p_2 p_3$. In this section we recast the SGH assumption of [CGKW18] in the asymmetric bilinear group of composite-order $N = p_1 p_2$. Let us denote $\Theta_N = (N = p_1 p_2, G, H, G_T, e, g, h)$, where g (resp. h) is the generator of G (resp. H) and the pairing is defined as $e : G \times H \to G_T$. Now we define the SGH assumptions as follows.

Assumption 1 $\mathbf{SGH}^H_{p_1 \to p_1 p_2}$ *Given $(\Theta_N, \mu = \{g_1, h_1, h_2\}, \hat{T})$, it is hard to decide whether $\hat{T} \in H_1$ or $\hat{T} \in H$.*

Assumption 2 $\mathbf{SGH}^G_{p_1 \to p_1 p_2}$ *Given $(\Theta_N, \mu = \{g_1, g_2, h_1\}, T)$, it is hard to decide whether $T \in G_1$ or $T \in G$.*

Assumption 3 $\mathbf{SGH}^H_{p_2 \to p_1 p_2}$ *Given $(\Theta_N, \mu = \{g_2, g_1^{\tau_1} g_2^{\tau_2}, h_1, h_2\}, \hat{T})$, it is hard to decide whether $\hat{T} \in H_2$ or $\hat{T} \in H$, for $\tau_1, \tau_2 \xleftarrow{\$} \mathbb{Z}_N$.*

Prime-Order Setting Now we define some variant of subspace assumptions similar to [CLL+12]. Here we consider the bilinear group generator $\mathcal{P}_{(\perp,4)}$ which outputs $(p, G, H, G_T, e, \{G_i, H_i\}_{i=1}^4)$. Let g (resp. h) be the generator of the group $\mathbb{G}$ (resp. $\mathbb{H}$), where $G \approx \mathbb{G}^4$ and $H \approx \mathbb{H}^4$. Let us denote $\Theta = (p, G, H, G_T, e, g, h)$ in the following definitions.

Assumption 4 *Given* Θ *and* $g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1^*}, h^{\boldsymbol{b}_2^*}, h^{\boldsymbol{b}_3^*}, h^{\boldsymbol{b}_4^*}, U_1 = g^{\mu_1 \boldsymbol{b}_1 + \mu_2 \boldsymbol{b}_3}$, $U_2 = g^{\mu_1 \boldsymbol{b}_2 + \mu_2 \boldsymbol{b}_4}, T_1 = h^{\tau_1 \boldsymbol{b}_1^* + \tau_2 \boldsymbol{b}_3^*}, T_2 = h^{\tau_1 \boldsymbol{b}_2^* + \tau_2 \boldsymbol{b}_4^*}$, *it is hard to decide whether* $\tau_2 = 0$ *mod* p *or not.*

Assumption 5 *Given* Θ *and* $g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, g^{\boldsymbol{b}_3}, g^{\boldsymbol{b}_4}, h^{\boldsymbol{b}_1^*}, h^{\boldsymbol{b}_2^*}, U_1 = h^{\mu_1 \boldsymbol{b}_1^* + \mu_2 \boldsymbol{b}_3^*}, U_2 = h^{\mu_1 \boldsymbol{b}_2^* + \mu_2 \boldsymbol{b}_4^*}, T_1 = g^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_3}, T_2 = g^{\tau_1 \boldsymbol{b}_2 + \tau_2 \boldsymbol{b}_4}$, *it is hard to decide whether* $\tau_2 = 0$ *mod* p *or not.*

We notice that Assumption 5 can be directly obtained from the subspace assumption in $\mathbb{G}$ [CLL+12, Definition 12] by removing the coefficient μ_2 and taking $N = 4$ and $k = 2$. Hence Assumption 5 is reducible to the subspace assumption in $\mathbb{G}$ [CLL+12, Definition 12]. Now from Lemma 2 of [CLL+12], $\mathrm{DDH}_{\mathbb{G}}$ is reducible to the subspace assumption in $\mathbb{G}$, and hence to Assumption 5. Similarly, Assumption 4 is also obtained from the subspace assumption in $\mathbb{H}$. Also, from Lemma 2 of [CLL+12], we infer that $\mathrm{DDH}_{\mathbb{H}}$ is reducible to the subspace assumption in $\mathbb{H}$ and hence to Assumption 4.

Now we recall the decisional Diffie-Hellman assumption (DDH) in $\mathbb{G}$ (denoted as $\mathrm{DDH}_{\mathbb{G}}$) as follows.

Assumption 6 *Given* $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h, g^a, g^b)$ *and* $T = g^{ab+\theta}$, *it is hard to decide whether* $\theta = 0$ *or not, for* $a, b \xleftarrow{\$} \mathbb{Z}_p$.

In the same way, we can define the DDH assumption in $\mathbb{H}$ (denoted as $\mathrm{DDH}_{\mathbb{H}}$). When $\mathcal{P}$ satisfies the DDH assumption in both $\mathbb{G}$ and $\mathbb{H}$, then we say that $\mathcal{P}$ satisfies the symmetric external Diffie-Hellman (SXDH) assumption.

3 RRS in the Composite-Order Setting

In this section, we present the rerandomizable signature scheme (denoted as `RRSc`) in the asymmetric composite-order setting. The structure of the signature is inspired from Ghadafi's [Gha17] structure-preserving signature (SPS) scheme. We prove unforgeability of the `RRSc` scheme under subgroup hiding assumptions. We recall the definition of signature scheme in § 2.2. Our main goal is to realize a rerandomizable signature scheme in the prime-order setting, which is described in §4. However we present the `RRSc` scheme in the composite-order setting as a stepping stone to explore the applicability of dual-form signature technique directly in the security argument, rather than explicitly constructing a dual-form variant of the signature scheme as in [GLOW12, YCZY14, CK18].

3.1 Construction

First we describe our `RRSc` construction idea in brief. The construction is instantiated in the bilinear group of composite-order $N = p_1 p_2$, in which the source groups are decomposed into two orthogonal subgroups i.e., $G \approx G_1 \oplus G_2$ and $H \approx H_1 \oplus H_2$ such that $e(g_i, h_j) = 1$, for $i \neq j$, where g_i (resp. h_j) is the generator of the subgroup G_i (resp. H_j). In the `RRSc` construction, we mimic the [Gha17]

SPS structure to obtain a signature on ℓ block of messages, for some $\ell \in \mathbb{N}$. In particular, we use the exponent of the form $(x + m_1 + \sum_{j=2}^{\ell} m_j y_j)/y_1$ suitably randomized in the subgroup G_1's component to obtain a rerandomizable signature. For verification, the variables x and y_j's are provided in the exponent of H_1 component of public key. To verify a signature, the randomness used in signing needs to be given in the subgroup G_1's exponent separately. Since the subgroup G_1 is of rank 1, we cannot retain the rerandomizable signature structure along with signing randomness in a single component. The subgroup G_2 is used to define another signing algorithm, which is used only in the unforgeability proof. The subgroup H_2 is used in the proof to determine the forgery type returned by the forger. In order to prove unforgeability under subgroup hiding assumptions, we adopt the Gerbush et al.'s [GLOW12] dual-form signature technique directly, rather than proceeding through the original dual-form signature construction.

Run the bilinear group generator $\mathcal{G}_N$ on λ which outputs $(\Theta_N, \mu = \{g_i, h_i\}_{i=1}^2)$, where $\Theta_N = (N = p_1 p_2, G, H, G_T, e)$ and g_i (resp. h_i) is a random element from the p_i-order subgroup G_i (resp. H_i) of G (resp. H), for $i \in [1, 2]$. The pairing is defined as $e : G \times H \to G_T$.

Table 2. `RRSc` scheme in the composite-order setting.

KeyGen(λ) Run $\mathcal{G}_N(\lambda) \to (\Theta_N, \mu = \{g_1, h_1\})$, where $\Theta_N = (N = p_1 p_2, G, H, G_T, e)$. Choose $x, \{y_j\}_{j=1}^{\ell} \xleftarrow{\$} \mathbb{Z}_N$ and set $PK := \{\Theta_N, h_1, \{Y_j := h_1^{y_j}\}_{j=1}^{\ell}, X := h_1^x\}$ and $SK := \{x, \{y_j\}_{j=1}^{\ell}, g_1\}$. Return (SK, PK).	**Sign**$(SK, \boldsymbol{m} = (m_1, \ldots, m_\ell))$ Choose $r \xleftarrow{\$} \mathbb{Z}_N$ and set $A := g_1^r$, $B := g_1^{\frac{r}{y_1}(x + m_1 + \sum_{j=2}^{\ell} m_j y_j)}$. Return $(\boldsymbol{m}, \sigma := (A, B))$. **Ver**$(PK, \boldsymbol{m}, \sigma)$ Accept if $e(A, h_1) \neq 1$ and $e(B, Y_1) = e(A, X h_1^{m_1} \prod_{j=2}^{\ell} Y_j^{m_j})$.

The `RRSc` scheme consists of three PPT algorithms, which are defined in Table 2. Notice that, we avoid the trivial forgery by checking $e(A, h_1) \neq 1$. Suppose we do not check the above condition, then anyone can output $\sigma = (1, 1)$ as a (trivial) forgery on any message $\boldsymbol{m} \in \mathbb{Z}_N^{\ell}$. The correctness of the scheme can be verified using the following equalities,

$$\begin{aligned} e(B, Y_1) &= e(g_1^{\frac{r}{y_1}(x + m_1 + \sum_{j=2}^{\ell} m_j y_j)}, h_1^{y_1}) \\ &= e(g_1^r, h_1^{x + m_1 + \sum_{j=2}^{\ell} m_j y_j}) = e(A, X h_1^{m_1} \prod_{j=2}^{\ell} Y_j^{m_j}). \end{aligned}$$

The second equality follows from the linearity of the pairing and the last equality follows from the definition of X and Y_j's.

3.2 Randomizability

An additional feature of a rerandomizable signature scheme is the so-called *rerandomizable property*. This feature has been utilized effectively in the construction of several other protocols, such as group signature [BCN+10] and anonymous credential scheme [CL04].

It is easy to see that the RRSc scheme satisfies rerandomizability property. Consider the signature $\sigma = (A, B)$ on the message $\boldsymbol{m}$ as defined in Table 2, which can be randomized by choosing a random $t \xleftarrow{\$} \mathbb{Z}_N$ and computing $\sigma' = (A^t, B^t)$. One can verify that σ' is a valid signature on $\boldsymbol{m}$ under the PK.

3.3 Unforgeability

As mentioned before, we use the Gerbush et al's [GLOW12] dual-form signature technique to prove unforgeability of the RRSc scheme under subgroup hiding assumptions. First we define the forgery classes as follows. Let $\mathcal{V}$ be the set of all message and signature pairs $(\boldsymbol{m}^*, \sigma^*)$ such that they verify under the public key PK, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*) \in \mathbb{Z}_N^\ell$ and $\sigma^* = (A^*, B^*) \in G^2$. Now we partition the forgery class $\mathcal{V}$ into two disjoint sets $\mathcal{V}_I$ and $\mathcal{V}_{II}$ which are defined as follows.

Type I: $\mathcal{V}_I = \{(\boldsymbol{m}^*, \sigma^*) \in \mathcal{V} : (A^*)^{p_2} = 1, (B^*)^{p_2} = 1\}$,
Type II: $\mathcal{V}_{II} = \{(\boldsymbol{m}^*, \sigma^*) \in \mathcal{V} : (A^*)^{p_2} \neq 1 \text{ or } (B^*)^{p_2} \neq 1\}$.

From the above definition, for any message and signature pair $(\boldsymbol{m}^*, \sigma^*)$ satisfying verification equation, the signature $\sigma^* = (A^*, B^*)$ can be written as $A^* = g_1^r g_2^{\delta_1}$ and $B^* = g_1^{\frac{r}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)} g_2^{\delta_2}$, for some r, δ_1, δ_2 from $\mathbb{Z}_N$. In order to prove unforgeability, we use the subgroup hiding assumptions defined in §2.4.

Theorem 1 *If the assumptions, $SGH^H_{p_1 \to p_1 p_2}$, $SGH^G_{p_1 \to p_1 p_2}$ and $SGH^H_{p_2 \to p_1 p_2}$ hold in $\mathcal{G}_N$, then the RRSc scheme is EUF-CMA secure.*

Proof. Let Sign_A be same as the Sign algorithm defined in Table 2. Next we define the following Sign_B algorithm, which is used by the simulator in the security argument. The Sign_B algorithm takes the secret key SK along with an element $g_2 \in G_2$ and the message $\boldsymbol{m} \in \mathbb{Z}_N^\ell$ and outputs a message-signature pair.

$\mathsf{Sign}_B(SK \cup \{g_2\}, \boldsymbol{m} = (m_1, \ldots, m_\ell))$:
Choose $r \xleftarrow{\$} \mathbb{Z}_N, \delta_1, \delta_2 \xleftarrow{\$} \mathbb{Z}_N$ and
set $A := g_1^r g_2^{\delta_1}, B := g_1^{\frac{r}{y_1}(x + m_1 + \sum_{j=2}^{\ell} m_j y_j)} g_2^{\delta_2}$.
Return $(\boldsymbol{m}, \sigma := (A, B))$.

Note that $e(g_2, h_1) = 1$ and hence the signature returned by Sign_B can be verified under PK. Now we use a hybrid argument to prove this theorem in terms of the following games.

Game_R. This is the original EUF-CMA game. Recall that, after receiving the PK from the challenger, the forger $\mathcal{A}$ makes q many signing oracle queries adaptively and then returns a forgery from $\mathcal{V}$.

Game_0. Same as Game_R except that $\mathcal{A}$ returns a forgery from $\mathcal{V}_I$. Let E be the event that $\mathcal{A}$ returns a forgery from $\mathcal{V}_{II}$ in Game_0. In Lemma 2, we prove that the event E happens with negligible probability under $\mathrm{SGH}^H_{p_1\to p_1p_2}$ assumption. Thus we deduce that Game_R and Game_0 are computationally indistinguishable under $\mathrm{SGH}^H_{p_1\to p_1p_2}$ assumption. In particular we have,

$$|Adv^{\mathsf{Game_R}}_{\mathcal{A}} - Adv^{\mathsf{Game_0}}_{\mathcal{A}}| \leq Pr[E] \leq Adv^{SGH^H_{p_1\to p_1p_2}}_{\mathcal{B}}.$$

Game_k. Same as Game_0 except that the first k signing queries are answered using Sign_B, for $k \in [1, q]$, whereas the last $q-k$ queries are answered using Sign_A. For $k \in [1, q]$, in Lemma 3, we prove that Game_{k-1} and Game_k are computationally indistinguishable under $\mathrm{SGH}^G_{p_1\to p_1p_2}$ assumption. In particular we have,

$$|Adv^{\mathsf{Game_{k-1}}}_{\mathcal{A}} - Adv^{\mathsf{Game_k}}_{\mathcal{A}}| \leq Adv^{SGH^G_{p_1\to p_1p_2}}_{\mathcal{B}}.$$

Finally in Lemma 4, we prove that $Adv^{\mathsf{Game_q}}_{\mathcal{A}}$ is negligible under $\mathrm{SGH}^H_{p_2\to p_1p_2}$ assumption. In particular we have,

$$Adv^{\mathsf{Game_q}}_{\mathcal{A}} \leq Adv^{SGH^H_{p_2\to p_1p_2}}_{\mathcal{B}}.$$

Hence by the hybrid argument and from Equations 1, 2 and 3, described below, we have,

$$\begin{aligned} Adv^{UF}_{\mathcal{A}} = Adv^{\mathsf{Game_R}}_{\mathcal{A}} &= |Adv^{\mathsf{Game_R}}_{\mathcal{A}} - Adv^{\mathsf{Game_0}}_{\mathcal{A}} + Adv^{\mathsf{Game_0}}_{\mathcal{A}} - Adv^{\mathsf{Game_1}}_{\mathcal{A}} + \\ &\qquad \ldots + Adv^{\mathsf{Game_{k-1}}}_{\mathcal{A}} - Adv^{\mathsf{Game_k}}_{\mathcal{A}} + \ldots + Adv^{\mathsf{Game_q}}_{\mathcal{A}}| \\ &\leq Pr[E] + q\, Adv^{SGH^G_{p_1\to p_1p_2}}_{\mathcal{B}} + Adv^{SGH^H_{p_2\to p_1p_2}}_{\mathcal{B}} \\ &\leq Adv^{SGH^H_{p_1\to p_1p_2}}_{\mathcal{B}} + q\, Adv^{SGH^G_{p_1\to p_1p_2}}_{\mathcal{B}} + Adv^{SGH^H_{p_2\to p_1p_2}}_{\mathcal{B}}. \end{aligned}$$

□

Lemma 2 *If* $\mathrm{SGH}^{H}_{p_1 \to p_1 p_2}$ *assumption holds in* $\mathcal{G}_N$*, then* $Pr[E]$ *is negligible.*

Proof. Assume that the event E happens with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the $\mathrm{SGH}^{H}_{p_1 \to p_1 p_2}$ problem as follows. $\mathcal{B}$ is given $\Theta_N, g_1, h_1, h_2, \hat{T}$ and his goal is to decide whether $\hat{T}$ is from H_1 or H. Now $\mathcal{B}$ chooses x, y_j from $\mathbb{Z}_N$, for $j \in [1, \ell]$ and defines the PK and SK as described in Table 2. Given the PK, $\mathcal{A}$ makes the signing oracle queries to $\mathcal{B}$. Since $\mathcal{B}$ knows the secret key SK, he computes σ_i using Sign_A algorithm and sends to $\mathcal{A}$. After q many queries, $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*)$ and $\sigma^* = (A^*, B^*)$. Then $\mathcal{B}$ checks (i) the forgery $(\boldsymbol{m}^*, \sigma^*)$ is valid and (ii) the message $\boldsymbol{m}^*$ is not queried earlier. If any of these checks fail to hold, $\mathcal{B}$ returns a random bit to his challenger. Otherwise, $\mathcal{B}$ checks whether $\mathcal{A}$ is returning a Type-II forgery. From the definition of forgery types, it is sufficient for $\mathcal{B}$ to check whether $e(A^*, h_2) \neq 1$ or $e(B^*, h_2) \neq 1$ holds.

As mentioned before, since the forgery returned by $\mathcal{A}$ is valid, $\mathcal{B}$ writes $A^* = g_1^r g_2^{\delta_1}$ and $B^* = g_1^{\frac{r}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)} g_2^{\delta_2}$, for some r, δ_1, δ_2 from $\mathbb{Z}_N$ unknown to $\mathcal{B}$. Now $\mathcal{B}$ defines the following backdoor verification test (BVT),

$$S := B^* (A^*)^{-\frac{1}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)} \stackrel{?}{=} 1.$$

Note that the correctness of the forgery ensures that $e(S, h_1) = 1$. Hence the above BVT can be simplified as $S = g_2^{\delta_2 - \frac{\delta_1}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)} \stackrel{?}{=} 1$. We argue that for a Type-II forgery, the event $S = 1$ happens with negligible probability. From the exponent of S, it is sufficient to prove that for a Type-II forgery, the event $\delta_2 - \frac{\delta_1}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j) = 0$ modulo p_2 happens with negligible probability.

Suppose $\delta_1 = 0$ modulo p_2, then the condition $\delta_2 - \frac{\delta_1}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j) = 0$ modulo p_2 ensures that δ_2 must be zero. This means the forgery cannot be Type-II, by definition. Hence assume that $\delta_1 \neq 0$ modulo p_2. We re-write the above condition as $(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)/y_1 = \delta_2/\delta_1$ modulo p_2. Since x and y_js are chosen uniformly at random from $\mathbb{Z}_N$, by Chinese Remainder Theorem (CRT), x (resp. y_j) modulo p_2 and x (resp. y_j) modulo p_1 are independent. Hence x and y_j modulo p_2 are information theoretically hidden to $\mathcal{A}$. Thus $\mathcal{A}$ have to guess the value of δ_2/δ_1. Hence the probability that $(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)/y_1 = \delta_2/\delta_1$ modulo p_2 is at most $1/N$, which is negligible.

Now $\mathcal{B}$ checks whether $S \stackrel{?}{=} 1$ or not. Suppose $S \neq 1$, then $\mathcal{B}$ checks whether $e(S, \hat{T}) \stackrel{?}{=} 1$ or not. If $e(S, \hat{T}) = 1$, then $\mathcal{B}$ returns 1 indicating $\hat{T} \in H_1$, else 0 indicating $\hat{T} \in H$. For the case of $S = 1$, $\mathcal{B}$ simply returns a random guess to his challenger. For a Type II forgery, the latter can happen only with a negligible probability. So we conclude,

$$Pr[E] \leq Adv_{\mathcal{B}}^{SGH^{H}_{p_1 \to p_1 p_2}}. \tag{1}$$

□

Lemma 3 *If* $\mathrm{SGH}^{G}_{p_1 \to p_1 p_2}$ *assumption holds in* $\mathcal{G}_N$*, then* $\mathsf{Game}_{k-1} \approx_c \mathsf{Game}_k$*, for* $k \in [1, q]$*.*

Proof. Suppose that, there exists a PPT adversary $\mathcal{A}$ who distinguishes between Game_{k-1} and Game_k with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the $\mathrm{SGH}^{G}_{p_1 \to p_1 p_2}$ problem as follows. $\mathcal{B}$ is given Θ_N and g_1, g_2, h_1, T and his goal is to decide whether $T \in G_1$ or $T \in G$. Now $\mathcal{B}$ chooses x, y_j and constructs the PK and SK as described in Table 2. After receiving PK, $\mathcal{A}$ makes signing queries on some message $\boldsymbol{m}_i = (m_{i1}, \ldots, m_{i\ell})$. For the first $k-1$ (resp. last $q-k$) request, $\mathcal{B}$ uses Sign_B (resp. Sign_A) algorithm to answer for signing queries, as he knows g_2 and all the secret key components. For the k-th request, $\mathcal{B}$ embeds the challenge term T and constructs and sends the signature $\sigma_k = (A_k, B_k)$ to $\mathcal{A}$, where $A_k = T, B_k = T^{(x + m_{k1} + \sum_{j=2}^{\ell} m_{kj} y_j)/y_1}$. Suppose $T \in G_1$, then the signature σ_k is distributed as an output of Sign_A. Thus $\mathcal{B}$ is simulating Game_{k-1}. Suppose $T \in G$, then from CRT, x (resp. y_j) modulo p_2 and x (resp. y_j) modulo p_1 are independent. Also from the definition of Sign_A and Sign_B, the values $x, y_j \bmod p_2$ are information theoretically hidden to $\mathcal{A}$. Hence the G_{p_2} part of σ_k is randomly distributed from the view of $\mathcal{A}$. Thus σ_k is distributed as an output of Sign_B and hence $\mathcal{B}$ is simulating Game_k.

Finally, $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*)$ and $\sigma^* = (A^*, B^*)$. Notice that σ_k is generated using the challenge term of the SGH assumption. Since $\mathcal{B}$ knows all the SK components in addition to the random element g_2 from G_2, he can generate the k-th signature of any type properly. However, $\mathcal{B}$ cannot on its own decide the type of the signatures generated using the problem instance as $\mathcal{B}$ is not given any element of H_2. In other words, $\mathcal{B}$ needs to rely on the advantage of $\mathcal{A}$.

As long as $\mathcal{A}$ distinguishes between Game_{k-1} and Game_k, $\mathcal{B}$ leverages $\mathcal{A}$ to solve the $\mathrm{SGH}^{G}_{p_1 \to p_1 p_2}$ assumption. Thus we have

$$|Adv_{\mathcal{A}}^{\mathsf{Game}_{k-1}} - Adv_{\mathcal{A}}^{\mathsf{Game}_k}| \leq Adv_{\mathcal{B}}^{SGH^{G}_{p_1 \to p_1 p_2}}. \tag{2}$$

□

Lemma 4 *If* $\mathrm{SGH}^{H}_{p_2 \to p_1 p_2}$ *assumption holds in* $\mathcal{G}_N$*, then* $Adv_{\mathcal{A}}^{\mathsf{Game}_q}$ *is negligible.*

Proof. Suppose that, there exists a PPT adversary $\mathcal{A}$ playing Game_q and winning with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the $\mathrm{SGH}^{H}_{p_2 \to p_1 p_2}$ problem as follows. $\mathcal{B}$ is given $\Theta_N, g_2, g_1^{\tau_1} g_2^{\tau_2}, h_1, h_2, \hat{T}$ and his goal is to decide whether $\hat{T} \in H_2$ or $\hat{T} \in H$. Next $\mathcal{B}$ chooses x, y_j uniformly at random from $\mathbb{Z}_N$ and defines the PK and SK as described in Table 2, while SK do not contain any random g_1 from G_1. Once PK is given to $\mathcal{A}$, he makes the signing queries on some message $\boldsymbol{m}_i = (m_{i1}, \ldots, m_{i\ell})$. Now $\mathcal{B}$ simulates the Sign_B algorithm and computes the signature as $\sigma_i = (A_i, B_i)$, where

$$A_i = (g_1^{\tau_1} g_2^{\tau_2})^{r'} g_2^{\delta_1'}, \quad B_i = (g_1^{\tau_1} g_2^{\tau_2})^{\frac{r'}{y_1}(x + m_{i1} + \sum_{j=2}^{\ell} m_{ij} y_j)} g_2^{\delta_2'},$$

for $r', \delta_1', \delta_2' \xleftarrow{\$} \mathbb{Z}_N$. It is easy to check that the above signature is properly distributed by substituting the randomness r, δ_1 and δ_2 by $\tau_1 r'$, $\delta_1' + \tau_2 r'$ and $\delta_2' + \tau_2 \frac{r'}{y_1}(x + m_{i1} + \sum_{j=2}^{\ell} m_{ij} y_j)$ respectively. After q many signing queries, $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*)$ and $\sigma^* = (A^*, B^*)$. As before, $\mathcal{B}$ checks (i) the forgery $(\boldsymbol{m}^*, \sigma^*)$ is valid and (ii) the message $\boldsymbol{m}^*$ is not queried earlier. If any of these checks fail to hold, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ proceeds as follows. As mentioned before, from the valid forgery, σ^* can be written as,

$$A^* = g_1^s, \quad B^* = g_1^{\frac{s}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)},$$

for some $s \in \mathbb{Z}_N$. By our initial assumption, $\mathcal{A}$ returns a Type-I forgery with some non-negligible probability. Now $\mathcal{B}$ checks $e(A^*, \hat{T}) \stackrel{?}{=} 1$. Notice that, $e(A^*, \hat{T}) = 1$ holds if and only if $\hat{T} \in H_2$, as A^* and $\hat{T}$ are non-trivial elements from G and H respectively. Thus we have,

$$Adv_{\mathcal{A}}^{\mathsf{Game_q}} \leq Adv_{\mathcal{B}}^{SGH^H_{p_2 \to p_1 p_2}}. \tag{3}$$

□

Remark 1 *In the above Lemma 4, we can use the following computational assumption. Given $\Theta_N, g_2, g_1^{\tau_1} g_2^{\tau_2}, h_1, h_2$, for $\tau_1, \tau_2 \in \mathbb{Z}_N$, it is hard to compute g_1^s, for some $s \in \mathbb{Z}_N$. Also, it is easy to see that the $SGH^H_{p_2 \to p_1 p_2}$ assumption (Assumption 3) implies this computational assumption.*

4 RRS in the Prime-Order Setting

Recall that Yuen et al. [YCZY14] presented the dual-form Boneh-Boyen signature scheme in the prime-order setting through the dual pairing vector space (DPVS) [OT08, OT10] framework. Following a similar approach, we use the DPVS framework to convert the `RRSc` scheme in the prime-order setting, which we call `RRS` scheme. We prove unforgeability of the scheme under the SXDH assumption.

4.1 Construction

In the DPVS setting, the underlying source groups are decomposed into four orthogonal subgroups i.e., $G \approx \oplus_{i=1}^4 G_i$ and $H \approx \oplus_{i=1}^4 H_i$ such that $e(g_i, h_j) = 1$, for $i \neq j$, where g_i (resp. h_i) is the generator of the subgroup G_i (resp. H_i). In the `RRS` construction, the subgroup $G_1 \oplus G_2$, which is of rank 2, is utilized to generate the signature. In particular, we use exponent of the form $\frac{1}{y_1}(x + m_1 + \sum_{j=2}^{\ell} m_j y_j)$ in the subgroup G_2 component while the corresponding randomness r is provided in the exponent of G_1 component. The rank 2 subgroup enables us to construct a signature having only one group element instead of two components in the `RRSc` construction. The associated public key structure is provided in the subgroup

Table 3. RRS in the prime order setting.

KeyGen(λ) Run $\mathcal{P}_{(\perp,4)} \to (\Theta, \{G_i, H_i\}_{i=1}^4)$, where $\Theta = (p, G, H, G_T, e, g, h)$. Choose $x, \{y_j\}_{j=1}^{\ell} \xleftarrow{\$} \mathbb{Z}_p$, $g^{\boldsymbol{d}_i} \xleftarrow{\$} G_i$ and $h^{\boldsymbol{d}_i^*} \xleftarrow{\$} H_i$, for $i \in [1,2]$. Set $PK := \{\Theta, h^{\boldsymbol{d}_1^*}, X := h^{x\boldsymbol{d}_1^*}, Y_1 := h^{y_1\boldsymbol{d}_2^*}, \{Y_j := h^{y_j\boldsymbol{d}_1^*}\}_{j=2}^{\ell}\}$, $SK := \{x, \{y_j\}_{j=1}^{\ell}, g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}\}$. Return (SK, PK).	**Sign**($SK, \boldsymbol{m} = (m_1, \ldots, m_\ell)$) Choose $r \xleftarrow{\$} \mathbb{Z}_p$ and set, $\sigma := g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x+m_1+\sum_{j=2}^{\ell} m_j y_j)\boldsymbol{d}_2}$. Return $(\boldsymbol{m}, \sigma)$. **Ver**($PK, \boldsymbol{m} = (m_1, \ldots, m_\ell), \sigma$) Accept if $e(\sigma, h^{\boldsymbol{d}_1^*}) \neq 1$ and $e(\sigma, Y_1 X (h^{\boldsymbol{d}_1^*})^{m_1} \prod_{j=2}^{\ell} Y_j^{m_j}) = 1$.

$H_1 \oplus H_2$. In particular, we retain the variables x, $\{y_j\}_{j\neq 1}$ in the subgroup H_1 while the variable y_1 is encoded in the subgroup H_2.

The RRS scheme consists of three PPT algorithms, which are defined in Table 3. Notice that we avoid the trivial forgery by checking $e(\sigma, h^{\boldsymbol{d}_1^*}) \neq 1$. As mentioned before, if this checking is removed, any one can produce $\sigma = 1$ as a (trivial) forgery on any message $\boldsymbol{m} \in \mathbb{Z}_p^{\ell}$. The correctness of the scheme can be verified using the following equation,

$$e(\sigma, Y_1 X (h^{\boldsymbol{d}_1^*})^{m_1} \prod_{j=2}^{\ell} Y_j^{m_j}) = e(g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x+m_1+\sum_{j=2}^{\ell} m_j y_j)\boldsymbol{d}_2}, h^{(x+m_1+\sum_{j=2}^{\ell} m_j y_j)\boldsymbol{d}_1^* + y_1\boldsymbol{d}_2^*})$$
$$= e(g,h)^{r(x+m_1+\sum_{j=2}^{\ell} m_j y_j)\psi - r(x+m_1+\sum_{j=2}^{\ell} m_j y_j)\psi} = 1.$$

The first equality follows from the definition of X and Y_j's. In the second equality, we use the fact that, $\boldsymbol{d}_i \cdot \boldsymbol{d}_i^* = \psi$ and $\boldsymbol{d}_i \cdot \boldsymbol{d}_j^* = 0$, for $i, j \in [1,4]$ and $i \neq j$.

4.2 Randomizability

It is easy to see that RRS scheme satisfies rerandomizability property. Consider the signature σ on the message $\boldsymbol{m} \in \mathbb{Z}_p^{\ell}$, which can be randomized by choosing a random $t \xleftarrow{\$} \mathbb{Z}_p$ and computing $\sigma' = \sigma^t$. One can verify that σ' is a valid signature on $\boldsymbol{m}$ under the PK.

4.3 Unforgeability

Recall that, in §3.3 we established the unforgeability proof for RRSc scheme using the orthogonality property and the CRT in the composite-order setting. However, in the prime-order setting, we will be using the parameter-hiding property [Lew12] instead of CRT. This necessitates instantiating the prime-order variant using DPVS framework, as it captures both orthogonality and parameter-hiding property.

Now we define the forgery classes. Let $\mathcal{V}$ be the set of all message and signature pairs such that they verify under the public key PK.

Type I: $\mathcal{V}_I = \{(\boldsymbol{m}^*, \sigma^*) \in \mathcal{V} : e(\sigma^*, h^{\boldsymbol{d}_3^*}) = 1, e(\sigma^*, h^{\boldsymbol{d}_4^*}) = 1\}$,
Type II: $\mathcal{V}_{II} = \{(\boldsymbol{m}^*, \sigma^*) \in \mathcal{V} : e(\sigma^*, h^{\boldsymbol{d}_3^*}) \neq 1 \text{ or } e(\sigma^*, h^{\boldsymbol{d}_4^*}) \neq 1\}$.

Consider the message and signature pair $(\boldsymbol{m}^*, \sigma^*)$ satisfying the verification equation, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*)$. Suppose the forgery is Type-I, then the signature σ^* can be written as

$$\sigma^* = g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2}, \tag{4}$$

for some $r \in \mathbb{Z}_p^*$. This is because, by definition, a Type-I forgery does not have any non-zero component of $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$. Suppose the forgery is Type-II, then the signature σ^* can be written as

$$\sigma^* = g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2 + \delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}, \tag{5}$$

for some $r, \delta_3, \delta_4 \in \mathbb{Z}_p$ where $r \neq 0$. This is because, a Type-II forgery contains some non-zero component of either $\boldsymbol{d}_3$ or $\boldsymbol{d}_4$. We will use the subspace assumptions (Assumption 4, 5) and $\text{DDH}_{\mathbb{H}}$ assumption defined in §2.4 to prove unforgeability.

Theorem 5 *The* RRS *scheme is EUF-CMA secure under the SXDH assumption.*

Proof. From §2.4, we know that $\text{DDH}_{\mathbb{H}}$ (resp. $\text{DDH}_{\mathbb{G}}$) is reducible to Assumption 4 (resp. Assumption 5). Let Sign_A be same as the Sign algorithm defined in Table 3. Next we define the following Sign_B algorithm, which is used by the simulator in the security argument.

$\mathsf{Sign}_B(SK \cup \{g^{\boldsymbol{d}_3}, g^{\boldsymbol{d}_4}\}, \boldsymbol{m} = (m_1, \ldots, m_\ell))$:
Choose $r, \delta_3, \delta_4 \xleftarrow{\$} \mathbb{Z}_p$ and set
$\sigma := g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2 + \delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}$,
Return $(\boldsymbol{m}, \sigma)$.

Note that the elements $g^{\boldsymbol{d}_3}$, $g^{\boldsymbol{d}_4}$ are orthogonal to the subgroup $H_1 \oplus H_2$, in which PK is defined. Hence the signature returned by Sign_B can be verified under PK. Now we use a hybrid argument to prove this theorem. First, we define the following games.

- Game_R. This is the original EUF-CMA game. Recall that, after receiving the PK from the challenger, the forger $\mathcal{A}$ makes q many signing oracle queries adaptively and then returns a forgery from $\mathcal{V}$.
- Game_0. Same as Game_R except that $\mathcal{A}$ returns a forgery from $\mathcal{V}_I$. Let E be the event that $\mathcal{A}$ returns a forgery from $\mathcal{V}_{II}$ in Game_0. In Lemma 6, we prove that the event E happens with negligible probability under Assumption 4. Thus we deduce that Game_R and Game_0 are computationally indistinguishable under Assumption 4. In particular we have,

$$|Adv_{\mathcal{A}}^{\mathsf{Game_R}} - Adv_{\mathcal{A}}^{\mathsf{Game_0}}| \leq Pr[E] \leq Adv_{\mathcal{B}}^{Ass\ 4}.$$

Game_k. Same as Game_0 except that the first k signing queries are answered using Sign_B, for $k \in [1,q]$, whereas the last $q-k$ queries are answered using Sign_A. For $k \in [1,q]$, in Lemma 7, we prove that Game_{k-1} and Game_k are computationally indistinguishable under Assumption 5. In particular we have,

$$|Adv_{\mathcal{A}}^{\mathsf{Game}_{k-1}} - Adv_{\mathcal{A}}^{\mathsf{Game}_k}| \leq Adv_{\mathcal{B}}^{Ass\ 5}.$$

Finally in Lemma 8, we prove that $Adv_{\mathcal{A}}^{\mathsf{Game}_q}$ is negligible under $\mathrm{DDH}_{\mathbb{H}}$ assumption. In particular we have,

$$Adv_{\mathcal{A}}^{\mathsf{Game}_q} \leq Adv_{\mathcal{B}}^{DDH_{\mathbb{H}}}.$$

Hence by the hybrid argument and from Equations 10, 11 and 12, described below, we have,

$$\begin{aligned} Adv_{\mathcal{A}}^{UF} = Adv_{\mathcal{A}}^{\mathsf{Game}_R} &= |Adv_{\mathcal{A}}^{\mathsf{Game}_R} - Adv_{\mathcal{A}}^{\mathsf{Game}_0} + Adv_{\mathcal{A}}^{\mathsf{Game}_0} - Adv_{\mathcal{A}}^{\mathsf{Game}_1} + \\ &\quad \ldots + Adv_{\mathcal{A}}^{\mathsf{Game}_{k-1}} - Adv_{\mathcal{A}}^{\mathsf{Game}_k} + \ldots + Adv_{\mathcal{A}}^{\mathsf{Game}_q}| \\ &\leq Adv_{\mathcal{B}}^{Ass\ 4} + q\, Adv_{\mathcal{B}}^{Ass\ 5} + Adv_{\mathcal{B}}^{DDH_{\mathbb{H}}} \\ &\leq (q+2)\, Adv_{\mathcal{B}}^{SXDH}. \end{aligned}$$

□

Lemma 6 *If Assumption 4 holds in $\mathcal{P}_{(\perp,4)}$, then $Pr[E]$ is negligible.*

Proof. Assume that the event E happens with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the Assumption 4 as follows. $\mathcal{B}$ is given $\Theta, \{g^{\boldsymbol{b}_i}\}_{i=1}^2, \{h^{\boldsymbol{b}_j^*}\}_{j=1}^4, U_1 = g^{\mu_1 \boldsymbol{b}_1 + \mu_2 \boldsymbol{b}_3}, U_2 = g^{\mu_1 \boldsymbol{b}_2 + \mu_2 \boldsymbol{b}_4}, T_1 = h^{\tau_1 \boldsymbol{b}_1^* + \tau_2 \boldsymbol{b}_3^*}, T_2 = h^{\tau_1 \boldsymbol{b}_2^* + \tau_2 \boldsymbol{b}_4^*}$ and his goal is to decide whether $\tau_2 = 0 \bmod p$ or not. First $\mathcal{B}$ chooses a matrix A uniformly at random from $GL(2, \mathbb{Z}_p)$ and defines the orthogonal basis as,

$$\boldsymbol{d}_i = \boldsymbol{b}_i,\ \ \boldsymbol{d}_i^* = \boldsymbol{b}_i^*,\ \ (\boldsymbol{d}_3, \boldsymbol{d}_4)^\top = A^{-\top}(\boldsymbol{b}_3, \boldsymbol{b}_4)^\top,\ \ (\boldsymbol{d}_3^*, \boldsymbol{d}_4^*)^\top = A(\boldsymbol{b}_3^*, \boldsymbol{b}_4^*)^\top,$$

for $i \in [1,2]$. Now $\mathcal{B}$ chooses x, y_j uniformly at random from $\mathbb{Z}_p$, for $j \in [1,\ell]$ and defines the PK and SK as described in Table 3. Recall that PK includes Θ which contains the description of G and H such that $G = \oplus_{i=1}^4 G_i$, $H = \oplus_{i=1}^4 H_i$. Notice that the information about the matrix A is given indirectly to the adversary $\mathcal{A}$ only through the description of the source groups G and H. However, from the parameter-hiding property [Lew12, Lemma 3], we can ensure that the matrix A is information theoretically hidden to the adversary $\mathcal{A}$.

Once PK is given to $\mathcal{A}$, he makes q many signing oracle queries to $\mathcal{B}$. Since he knows all the SK components, $\mathcal{B}$ can answer for the signing queries using Sign_A algorithm. Finally $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$. Then $\mathcal{B}$ checks (i) the forgery $(\boldsymbol{m}^*, \sigma^*)$ is valid and (ii) the message $\boldsymbol{m}^*$ is not queried earlier. If any of these checks fail to hold, $\mathcal{B}$ returns a random bit to his challenger. Otherwise, $\mathcal{B}$ checks whether $\mathcal{A}$ returns a Type-II forgery. From the definition of forgery types, it is sufficient for $\mathcal{B}$ to check whether $e(\sigma^*, h^{\boldsymbol{d}_3^*}) \neq 1$ or $e(\sigma^*, h^{\boldsymbol{d}_4^*}) \neq 1$ holds.

As $\mathcal{A}$ outputs a valid forgery $(\boldsymbol{m}^*, \sigma^*)$, from Equation 5, $\mathcal{B}$ can write the signature $\sigma^* := g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2 + \delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}$, for some r from $\mathbb{Z}_p^*$ and δ_3, δ_4 from $\mathbb{Z}_p$ unknown to $\mathcal{B}$. Now $\mathcal{B}$ defines the following backdoor verification test (BVT),

$$e(\sigma^*, \Delta) \stackrel{?}{=} 1, \tag{6}$$

where $\Delta := (h^{\boldsymbol{b}_1^*} h^{\boldsymbol{b}_3^*})^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} (h^{\boldsymbol{b}_2^*} h^{\boldsymbol{b}_4^*})^{y_1}$. Note that,

$$\begin{aligned} e(\sigma^*, \Delta) &= e(g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2}, (h^{\boldsymbol{d}_1^*})^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} (h^{\boldsymbol{d}_2^*})^{y_1}) \\ &\quad e(g^{\delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}, (h^{\boldsymbol{b}_3^*})^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} (h^{\boldsymbol{b}_4^*})^{y_1}) \\ &= e(g^{\delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}, (h^{\boldsymbol{b}_3^*})^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} (h^{\boldsymbol{b}_4^*})^{y_1}). \end{aligned} \tag{7}$$

The first equality follows from the orthogonality of the basis and the last equality follows from the correctness of the forgery.

Next we consider the coefficient of Δ with respect to the basis $(\boldsymbol{b}_3^*, \boldsymbol{b}_4^*)$, which is $((x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j), y_1)^\top$. However, from the change of basis techniques, we obtain the coefficient of Δ with respect to the basis $(\boldsymbol{d}_3^*, \boldsymbol{d}_4^*)$ as $A^{-1}((x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j), y_1)^\top$. Then we simplify the BVT Equation 6 as,

$$(\delta_3, \delta_4) \cdot A^{-1}((x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j), y_1)^\top \stackrel{?}{=} 0 \bmod p. \tag{8}$$

Now we argue that for a Type-II forgery, Equation 8 holds with only a negligible probability. Recall that, Lewko [Lew12, Lemma 3] ensures that the matrix A is information theoretically hidden to $\mathcal{A}$. Also, $\mathcal{A}$ is given Sign_A oracle access and PK contains the variables x, y_j in the exponent of $H_1 \oplus H_2$. Hence the value $A^{-1}((x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j), y_1)^\top$ is randomly distributed to $\mathcal{A}$. However, $\mathcal{A}$ has to produce a Type-II forgery which is having a non-zero coefficient of the basis $(\boldsymbol{d}_3, \boldsymbol{d}_4)$. The only possibility for $\mathcal{A}$ is to guess the values of δ_3 and δ_4 from $\mathbb{Z}_p$ such that Equation 8 holds. Thus $\mathcal{A}$ can create a Type-II forgery that satisfies BVT Equation 6 only with probability atmost $1/p^2$, which is negligible.

Now $\mathcal{B}$ checks whether the BVT Equation 6 holds or not. Suppose BVT does not hold, then $\mathcal{B}$ checks whether

$$e(\sigma^*, T_1^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} T_2^{y_1}) \stackrel{?}{=} 1 \tag{9}$$

holds or not. As similar to Equation 7, we can simplify Equation 9 as,

$$\begin{aligned} e(\sigma^*, T_1^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)} T_2^{y_1}) &= e(g^{r\boldsymbol{d}_1 - \frac{r}{y_1}(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2}, \\ &\qquad h^{\tau_1(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_1^* + \tau_1 y_1 \boldsymbol{d}_2^*}) \\ &\quad e(g^{\delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}, h^{\tau_2(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{b}_3^* + \tau_2 y_1 \boldsymbol{b}_4^*}) \\ &= e(g^{\delta_3\boldsymbol{d}_3 + \delta_4\boldsymbol{d}_4}, h^{(x+m_1^*+\sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{b}_3^* + y_1 \boldsymbol{b}_4^*})^{\tau_2}. \end{aligned}$$

The first equality follows from the orthogonality of the basis and the last equality follows from the correctness of the forgery. We already argued that Equation 8 holds only with a negligible probability and the same holds for the Equation 9, when $\tau_2 \neq 0$. Hence, $\mathcal{B}$ returns 1 if Equation 9 holds, indicating $\tau_2 = 0$, else $\mathcal{B}$ returns 0 to its challenger. If BVT Equation 6 holds, $\mathcal{B}$ simply returns a random guess to his challenger. We have already established that for a Type-II forgery, the latter can happen only with a negligible probability. So we conclude,

$$Pr[E] \leq Adv_{\mathcal{B}}^{Ass\ 4}. \tag{10}$$

□

Lemma 7 *If Assumption 5 holds in $\mathcal{P}_{(\perp,4)}$, then $\mathsf{Game}_{k-1} \approx_c \mathsf{Game}_k$, for $k \in [1, q]$.*

Proof. Suppose that, there exists a PPT adversary $\mathcal{A}$ who distinguishes between Game_{k-1} and Game_k with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the Assumption 5 as follows. $\mathcal{B}$ is given $\Theta, \{g^{\boldsymbol{b}_i}\}_{i=1}^4, \{h^{\boldsymbol{b}_j^*}\}_{j=1}^2$, $U_1 = h^{\mu_1 \boldsymbol{b}_1^* + \mu_2 \boldsymbol{b}_3^*}, U_2 = h^{\mu_1 \boldsymbol{b}_2^* + \mu_2 \boldsymbol{b}_4^*}, T_1 = g^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_3}, T_2 = g^{\tau_1 \boldsymbol{b}_2 + \tau_2 \boldsymbol{b}_4}$ and his goal is to decide whether $\tau_2 = 0 \bmod p$ or not. Now $\mathcal{B}$ chooses a matrix A uniformly at random from $GL(2, \mathbb{Z}_p)$ and defines the orthogonal basis as,

$$\boldsymbol{d}_i = \boldsymbol{b}_i,\ \ \boldsymbol{d}_i^* = \boldsymbol{b}_i^*,\ \ (\boldsymbol{d}_3, \boldsymbol{d}_4)^\top = A(\boldsymbol{b}_3, \boldsymbol{b}_4)^\top,\ \ (\boldsymbol{d}_3^*, \boldsymbol{d}_4^*)^\top = A^{-\top}(\boldsymbol{b}_3^*, \boldsymbol{b}_4^*)^\top,$$

for $i \in [1, 2]$. Next $\mathcal{B}$ chooses x, y_j uniformly at random from $\mathbb{Z}_p$, for $j \in [1, \ell]$ and defines the PK and SK as described in Table 3. Once $\mathcal{B}$ sends the PK, $\mathcal{A}$ makes q many signing queries on some message $\boldsymbol{m}_i = (m_{i1}, \ldots, m_{i\ell})$. $\mathcal{B}$ uses Sign_A algorithm to answer for the last $q - k$ signing queries, as he knows all the SK components. Also $\mathcal{B}$ uses Sign_B algorithm to answer for the first $k - 1$ signing queries, as he knows the elements $g^{\boldsymbol{d}_3}$ and $g^{\boldsymbol{d}_4}$ in addition to all the SK components. For the k-th signing request, $\mathcal{B}$ embeds the challenge terms T_1 and T_2 to compute $\sigma_k = T_1 T_2^{\frac{-1}{y_1}(x + m_{k1} + \sum_{j=2}^{\ell} m_{kj} y_j)}$. Then $\mathcal{B}$ sends σ_k to $\mathcal{A}$. Here $\mathcal{B}$ implicitly sets τ_1 as r modulo p. If $T_1 = g^{\tau_1 \boldsymbol{b}_1}$ and $T_2 = g^{\tau_1 \boldsymbol{b}_2}$, then the k-th signature σ_k is distributed as an output of Sign_A. Thus $\mathcal{B}$ is simulating $\mathtt{Game}_{k-1}$. Suppose $T_1 = g^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_3}$ and $T_2 = g^{\tau_1 \boldsymbol{b}_2 + \tau_2 \boldsymbol{b}_4}$, with $\tau_2 \neq 0$. Then the coefficient of σ_k with respect to the basis $(\boldsymbol{d}_3, \boldsymbol{d}_4)$ is $\tau_2 A^{-1}(1, \frac{-1}{y_1}(x + m_{k1} + \sum_{j=2}^{\ell} m_{kj} y_j))^\top$. Since the matrix A is chosen uniformly at random, we obtain the coefficient of the σ_k with respect to the basis $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$ are random. By taking $r = \tau_1$ and $(\delta_3, \delta_4) = \tau_2 A^{-1}(1, \frac{-1}{y_1}(x + m_{k1} + \sum_{j=2}^{\ell} m_{kj} y_j))$, it is easy to see that the signature σ_k is properly distributed as an output of $\mathtt{Sign}_B$. Thus $\mathcal{B}$ is simulating $\mathtt{Game}_k$.

Finally $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$. As before, $\mathcal{B}$ checks (i) the forgery is valid and (ii) the message $\boldsymbol{m}^*$ is not queried earlier. Notice that $\mathcal{B}$ is not given with the elements $h^{\boldsymbol{b}_3^*}$ and $h^{\boldsymbol{d}_4^*}$ to check the forgery types. Hence, as similar to Lemma 3, $\mathcal{B}$ cannot compare the signature σ_k constructed above with the signature obtained by using SK components, to break the underlying assumption. In other words, $\mathcal{B}$ has to rely on the advantage of $\mathcal{A}$.

As long as $\mathcal{A}$ distinguishes between Game_{k-1} and Game_k with some non-negligible probability, $\mathcal{B}$ leverages $\mathcal{A}$ to break the Assumption 5. Thus we have

$$|Adv_{\mathcal{A}}^{\mathsf{Game}_{k-1}} - Adv_{\mathcal{A}}^{\mathsf{Game}_k}| \leq Adv_{\mathcal{B}}^{Ass\ 5}. \tag{11}$$

□

Lemma 8 *If $DDH_{\mathbb{H}}$ assumption holds in $\mathcal{P}$, then Adv^{Game_q} is negligible.*

Proof. Suppose that, there exists a PPT adversary $\mathcal{A}$ playing Game_q and winning with some non-negligible probability. Then we construct a simulator $\mathcal{B}$ to break the $\mathrm{DDH}_{\mathbb{H}}$ assumption as follows. $\mathcal{B}$ is given $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h, h^a, h^b, h^{ab+\theta})$ and his goal is to decide whether $\theta = 0$ or not. Now $\mathcal{B}$ chooses $(\mathbb{F}, \mathbb{F}^*)$ uniformly at random from $\mathsf{Dual}(\mathbb{Z}_p^4)$, where $\mathbb{F} = \{\boldsymbol{f}_i\}_{i=1}^4$ and $\mathbb{F}^* = \{\boldsymbol{f}_i^*\}_{i=1}^4$. Next $\mathcal{B}$ defines the orthogonal basis $(\mathbb{D}, \mathbb{D}^*)$ as,

$$\begin{aligned} \boldsymbol{d}_1 &= \boldsymbol{f}_1 - a\boldsymbol{f}_3, \quad \boldsymbol{d}_2 = \boldsymbol{f}_2, \quad \boldsymbol{d}_3 = \boldsymbol{f}_3, \quad \boldsymbol{d}_4 = \boldsymbol{f}_4, \\ \boldsymbol{d}_1^* &= \boldsymbol{f}_1^*, \quad \boldsymbol{d}_2^* = \boldsymbol{f}_2^*, \quad \boldsymbol{d}_3^* = \boldsymbol{f}_3^* + a\boldsymbol{f}_1^*, \quad \boldsymbol{d}_4^* = \boldsymbol{f}_4^*. \end{aligned}$$

$\mathcal{B}$ computes $\{g^{\boldsymbol{d}_i}\}_{i=2}^4$ and $\{h^{\boldsymbol{d}_i^*}\}_{i=1}^4$, whereas he cannot compute $g^{\boldsymbol{d}_1}$, as he does not know g^a. $\mathcal{B}$ chooses μ_1', μ_2' uniformly at random from $\mathbb{Z}_p$ and computes,

$$U := g^{\mu_1'\boldsymbol{f}_1 + \mu_2'\boldsymbol{f}_3} = g^{\mu_1'\boldsymbol{d}_1 + (\mu_2' + a\mu_1')\boldsymbol{d}_3} = g^{\mu_1\boldsymbol{d}_1 + \mu_2\boldsymbol{d}_3},$$

where $\mathcal{B}$ implicitly sets $\mu_1 = \mu_1'$ and $\mu_2 = \mu_2' + a\mu_1'$ modulo p. Next $\mathcal{B}$ chooses x, y_j uniformly at random from $\mathbb{Z}_p$, for $j \in [1, \ell]$ and defines the PK and SK as described in Table 3, while SK does not contain any random element of G_1. After receiving PK, $\mathcal{A}$ makes signing queries for some message $\boldsymbol{m}_i = (m_{i1}, \ldots, m_{i\ell})$. Since $\mathcal{B}$ knows $U = g^{\mu_1\boldsymbol{d}_1 + \mu_2\boldsymbol{d}_3}$ and $\{g^{\boldsymbol{d}_i}\}_{i=2}^4$, he can answer for the Sign_B queries by computing,

$$\sigma_i := U^{r'}(g^{\boldsymbol{d}_2})^{-\mu_1\frac{r'}{y_1}(x + m_{i1} + \sum_{j=2}^{\ell} m_{ij}y_j)}(g^{\boldsymbol{d}_3})^{\delta_3'}(g^{\boldsymbol{d}_4})^{\delta_4},$$

for r', δ_3', δ_4 uniformly chosen from $\mathbb{Z}_p$ and $r' \neq 0$. Now, it is easy to check that the above signature is properly distributed by implicitly setting the randomness r and δ_3 by $\mu_1' r'$ and $\delta_3' + (\mu_2' + a\mu_1')r'$ respectively. After q many signing queries, $\mathcal{A}$ returns a forgery $(\boldsymbol{m}^*, \sigma^*)$, where $\boldsymbol{m}^* = (m_1^*, \ldots, m_\ell^*)$.

As before, $\mathcal{B}$ checks (i) the forgery is valid and (ii) $\boldsymbol{m}^*$ is not queried earlier. If any of these checks fail to hold, $\mathcal{B}$ aborts. Otherwise, first $\mathcal{B}$ checks whether $\mathcal{A}$ returns Type-I forgery by checking $e(\sigma^*, h^{\boldsymbol{d}_3^*}) = 1$ and $e(\sigma^*, h^{\boldsymbol{d}_4^*}) = 1$.

As mentioned in Equation 4, for the valid forgery, $\mathcal{B}$ writes the signature as, $\sigma^* = g^{s\boldsymbol{d}_1 - \frac{s}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2}$, for some $s \in \mathbb{Z}_p^*$. Since $\mathcal{B}$ knows $\{\boldsymbol{f}_i, \boldsymbol{f}_i^*\}_{i=1}^4$, he computes g^s and g^{sa} as follows. Expand the σ^* in-terms of the orthogonal basis $(\mathbb{F}, \mathbb{F}^*)$, we have

$$\begin{aligned} \sigma^* =& g^{s\boldsymbol{d}_1 - \frac{s}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{d}_2} = g^{s(\boldsymbol{f}_1 - a\boldsymbol{f}_3) - \frac{s}{y_1}(x + m_1^* + \sum_{j=2}^{\ell} m_j^* y_j)\boldsymbol{f}_2} \\ \Rightarrow & g^s = (\sigma^*)^{(\boldsymbol{f}_1^*)^\top} \text{ and } g^{sa} = (\sigma^*)^{-(\boldsymbol{f}_3^*)^\top}. \end{aligned}$$

In the above equation, we use the dual orthonormal basis $\{\boldsymbol{f}_i, \boldsymbol{f}_i^*\}$ such that $\boldsymbol{f}_i \cdot \boldsymbol{f}_i^* = 1$, for $i \in [1,4]$ and $\boldsymbol{f}_i \cdot \boldsymbol{f}_j^* = 0$, for $i \neq j$. The condition $e(\sigma^*, h^{\boldsymbol{d}_1^*}) \neq 1$ ensures that $s \neq 0$ modulo p.

Now $\mathcal{B}$ checks whether $e(g^s, h^{ab+\theta}) \stackrel{?}{=} e(g^{sa}, h^b)$. Notice that the equality holds only when $\theta = 0$. Hence we obtain,

$$Adv_{\mathcal{A}}^{\mathsf{Game_q}} \leq Adv_{\mathcal{B}}^{DDH_{\mathbb{H}}}. \tag{12}$$

□

Remark 2 *In the above Lemma 8, we can use the following computational assumption. Given $p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h, h^a$ it is hard to compute (g^s, g^{sa}), for some $s \in \mathbb{Z}_p^*$. It is easy to see that the $DDH_{\mathbb{H}}$ assumption implies this computational assumption.*

4.4 Another Variant

Here we present a variant of the above signature scheme, denoted as `PS-RRS` scheme. This construction is derived from [PS16] rerandomizable signature scheme. In particular, we use the $(\ell+1)$-wise pairwise independent function of the form $(x + \sum_{i=1}^{\ell} m_i y_i)$ in our construction. Along with the above structure, the random term r is given in the exponent of the subgroup G_1's generator $g^{\boldsymbol{d}_1}$. The public key components will be defined appropriately to validate the signature. We describe the `PS-RRS` scheme in Table 4.

Table 4. `RRS` scheme in the prime-order setting.

KeyGen(λ) Run $\mathcal{P}_{(\perp,4)} \to (\Theta, \{G_i, H_i\}_{i=1}^4)$, where $\Theta = (p, G, H, G_T, e, g, h)$. Choose $x, \{y_j\}_{j=1}^{\ell} \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, $g^{\boldsymbol{d}_i} \stackrel{\$}{\leftarrow} G_i$ and $h^{\boldsymbol{d}_i^*} \stackrel{\$}{\leftarrow} H_i$, for $i \in [1,2]$. Set $PK := \{\Theta, h^{\boldsymbol{d}_1^*}, h^{\boldsymbol{d}_2^*}, X := h^{x\boldsymbol{d}_1^*}, \{Y_j := h^{y_j \boldsymbol{d}_1^*}\}_{j=1}^{\ell}\}$ and $SK := \{x, \{y_j\}_{j=1}^{\ell}, g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}\}$. Return (SK, PK).	**Sign**($SK, \boldsymbol{m} = (m_1, \ldots, m_\ell)$) Choose $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and set, $\sigma := g^{r\boldsymbol{d}_1 - r(x + \sum_{j=1}^{\ell} m_j y_j)\boldsymbol{d}_2}$. Return $(\boldsymbol{m}, \sigma)$. **Ver**($PK, \boldsymbol{m} = (m_1, \ldots, m_\ell), \sigma$) Accept if $e(\sigma, h^{\boldsymbol{d}_1^*}) \neq 1$ and $e(\sigma, X \prod_{j=1}^{\ell} Y_j^{m_j} h^{\boldsymbol{d}_2^*}) = 1$.

One can easily check the correctness of the scheme and that the signature components are rerandomizable. We only give a high level idea for the unforgeability proof of `PS-RRS` scheme, as it essentially mimics that of Theorem 5. Recall that in the proof of Lemma 6, 7 and 8, the variables x, y_j's are chosen by the simulator. Hence it does not matter whether we are arguing the unforgeability of the signature whose exponent structure is of the form $r(x + m_1 + \sum_{i=2}^{\ell} m_i y_i)/y_1$ (for `RRS` scheme) or $r(x + \sum_{i=1}^{\ell} m_i y_i)$ (for `PS-RRS` scheme). Thus one can use the same set of assumptions to argue the unforgeability proof of `PS-RRS` scheme.

4.5 Comparison

We compare our rerandomizable signature schemes instantiated in the prime-order setting with the existing schemes in Table 5.

Table 5. Comparing rerandomizable signatures for multiple block messages.

	$\|PK\|$	$\|\sigma\|$	Cost of Sign.	Cost of Verification	Rand.	Hardness Ass.
CL-RRS [CL04]	$(\ell+2)\|\mathbb{G}\|$	$(2\ell+1)\|\mathbb{G}\|$	$(2\ell+1)E_{\mathbb{G}}$	$4\ell\mathbb{P}+\ell E_{\mathbb{G}}+\ell M_{\mathbb{G}}$	Full	LRSW
mCL-RRS [PS18]	$(\ell+3)\|\mathbb{G}\|$	$(2\ell+3)\|\mathbb{G}\|$ $+1\|\mathbb{Z}_p\|$	$(2\ell+3)E_{\mathbb{G}}$	$4(\ell+1)\mathbb{P}+(\ell+1)E_{\mathbb{G}}$ $+(\ell+1)M_{\mathbb{G}}$	Weak	q-MSDH-1
PS-RRS [PS16]	$(\ell+2)\|\mathbb{H}\|$	$2\|\mathbb{G}\|$	$2E_{\mathbb{G}}$	$2\mathbb{P}+\ell E_{\mathbb{H}}+\ell M_{\mathbb{H}}$	Full	PS
mPS-RRS [PS18]	$(\ell+3)\|\mathbb{H}\|$	$2\|\mathbb{G}\|+1\|\mathbb{Z}_p\|$	$2E_{\mathbb{G}}$	$2\mathbb{P}+(\ell+1)E_{\mathbb{H}}$ $+(\ell+1)M_{\mathbb{H}}$	Weak	q-MSDH-2
`PS-RRS` §4.4	$(4\ell+13)\|\mathbb{H}\|+1\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$8E_{\mathbb{G}}+4M_{\mathbb{G}}$	$8\mathbb{P}+6M_{\mathbb{G}_T}+4\ell E_{\mathbb{H}}$ $+4(\ell+1)M_{\mathbb{H}}$	Full	SXDH
`RRS` §4.1	$(4\ell+9)\|\mathbb{H}\|+1\|\mathbb{G}\|$					

For any group $X \in \{\mathbb{G}, \mathbb{H}, \mathbb{G}_T\}$, E_X, M_X respectively denote the cost of the exponentiation, multiplication in X and $|X|$ is the bit size of X whereas $\mathbb{P}$ denotes pairing computation cost.

Notice that both CL-RRS [CL04] and modified CL-RRS (denoted as mCL-RRS) [PS18, Section 6.2] are defined in the symmetric prime-order setting. However, the signature size in both CL-RRS and mCL-RRS schemes depends on the message block length ℓ. The remaining schemes such as PS-RRS [PS16] and modified PS-RRS (denoted as mPS-RRS) [PS18, Section 4.2] are defined in the asymmetric prime-order setting, whose signature size is independent of the message block length ℓ. However, the unforgeability of CL-RRS and PS-RRS (resp. mCL-RRS and mPS-RRS) schemes is proved under interactive (resp. parameterized) assumption. Notice that both mCL-RRS and mPS-RRS schemes achieve only weakly rerandomizable property.

In contrast, both `RRS` and `PS-RRS` schemes are instantiated in the asymmetric prime-order setting. Both schemes ensure full rerandomizable property and unforgeability under the SXDH assumption. Also the signature size of `RRS` and `PS-RRS` schemes is constant and thus independent of the message block length. `PS-RRS` scheme is having one more public key component (i.e., four atomic group elements) as compared to the `RRS` scheme. Hence `RRS` scheme is slightly better than `PS-RRS` scheme.

5 Concluding Remark

We proposed the first construction of rerandomizable signature scheme in the standard model based on the SXDH assumption in the prime order bilinear pairing setting. This is achieved by applying the dual form signature technique in the DPVS setting on an RRS inspired by Ghadafi's SPS. Our proposal retains the desirable properties of RRS, namely full randomizability and constant size signature on a block of messages.

References

ASM06. Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116, pages 111–125, Springer, 2006.

BB04. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT* 2004, volume 3027, pages 223–238, Springer, 2004.

BCN+10. Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Get shorty via group signatures without encryption. In Juan A. Garay and Roberto De Prisco, editors, *SCN* 2010, volume 6280, pages 381–398, Springer, 2010.

BFG+13. David Bernhard, Georg Fuchsbauer, Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Anonymous attestation with user-controlled linkability. *Int. J. Inf. Sec.*, 12(3):219–249, 2013.

BGOY07. Alexandra Boldyreva, Craig Gentry, Adam O'Neill, and Dae Hyun Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS*, pages 276–285, 2007.

BW07. Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC*, volume 4450, pages 1–15, Springer, 2007.

CGKW18. Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT*, volume 10820, pages 503–534, Springer, 2018.

Che06. Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004, pages 1–11, Springer, 2006.

CK18. Sanjit Chatterjee and R. Kabaleeshwaran. Towards static assumption based cryptosystem in pairing setting: Further applications of DéjàQ and dual-form signature (extended abstract). In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *ProvSec*, volume 11192, pages 220–238, Springer, 2018.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152, pages 56–72, Springer, 2004.

CLL+12. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. In Michel Abdalla and Tanja Lange, editors, *Pairing*, volume 7708, pages 122–140, Springer, 2012.

CPST15. Sébastien Canard, David Pointcheval, Olivier Sanders, and Jacques Traoré. Divisible e-cash made practical. In Jonathan Katz, editor, *PKC*, volume 9020, pages 77–100, Springer, 2015.

CW14. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *SCN*, volume 8642, pages 277–297, Springer, 2014.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110, pages 44–61, Springer, 2010.

Gha17. Essam Ghadafi. More efficient structure-preserving signatures - or: Bypassing the type-iii lower bounds. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *ESORICS*, volume 10493, pages 43–61. Springer, 2017.

GLOW12. Michael Gerbush, Allison B. Lewko, Adam O'Neill, and Brent Waters. Dual form signatures: An approach for proving security from static assumptions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658, pages 25–42, Springer, 2012.

GMR88. 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.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

Gui13. Aurore Guillevic. Arithmetic of pairings on algebraic curves for cryptography PhD thesis, École Normale Supérieure, Paris, France, 2013.

Lew12. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237, pages 318–335, Springer, 2012.

LJYP14. Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. Concise multi-challenge cca-secure encryption and signatures with almost tight security. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT*, volume 8874, pages 1–21, Springer, 2014.

LLY13. Kwangsu Lee, Dong Hoon Lee, and Moti Yung. Aggregating CL-signatures revisited: Extended functionality and better efficiency. In Ahmad-Reza Sadeghi, editor, *FCDS*, volume 7859, pages 171–188, Springer, 2013.

LMPY16. Benoît Libert, Fabrice Mouhartem, Thomas Peters, and Moti Yung. Practical "signatures with efficient protocols" from simple assumptions. In Xiaofeng Chen, XiaoFeng Wang, and Xinyi Huang, editors, *AsiaCCS*, pages 511–522, ACM, 2016.

LPY15. Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In Rosario Gennaro and Matthew Robshaw, editors, *CRYPTO*, volume 9216, pages 296–316, Springer, 2015.

LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC*, volume 5978, pages 455–479, Springer, 2010.

OT08. Tatsuaki Okamoto and Katsuyuki Takashima. Homomorphic encryption and signatures from vector decomposition. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209, pages 57–74, Springer, 2008.

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO*, volume 6223, pages 191–208, Springer, 2010.

PS16. David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *CT-RSA*, volume 9610, pages 111–126, Springer, 2016.

PS18. David Pointcheval and Olivier Sanders. Reassessing security of randomizable signatures. In Nigel P. Smart, editor, *CT-RSA*, volume 10808, pages 319–338, Springer, 2018.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. *IACR Cryptology ePrint Archive*, 2009:385, 2009.

YCZY14. Tsz Hon Yuen, Sherman S. M. Chow, Cong Zhang, and Siu-Ming Yiu. Exponent-inversion Signatures and IBE under Static Assumptions. *IACR Cryptology ePrint Archive*, 2014:311, 2014.