Beyond the Limitation of Prime-Order Bilinear Groups, and Round Optimal Blind Signatures*

Jae Hong Seo[1] and Jung Hee Cheon[2]

[1] National Institute of Information and Communications Technology, Tokyo, Japan
jaehong@nict.go.jp
[2] ISaC & Dep. of Mathematical Sciences, Seoul National University, Seoul, Korea
jhcheon@snu.ac.kr

Abstract. At Eurocrypt 2010, Freeman proposed a transformation from pairing-based schemes in composite-order bilinear groups to equivalent ones in prime-order bilinear groups. His transformation can be applied to pairing-based cryptosystems exploiting only one of two properties of composite-order bilinear groups: cancelling and projecting. At Asiacrypt 2010, Meiklejohn, Shacham, and Freeman showed that prime-order bilinear groups according to Freeman's construction cannot have two properties simultaneously except negligible probability and, as an instance of implausible conversion, proposed a (partially) blind signature scheme whose security proof exploits both the cancelling and projecting properties of composite-order bilinear groups.

In this paper, we invalidate their evidence by presenting a security proof of the prime-order version of their blind signature scheme. Our security proof follows a different strategy and exploits only the projecting property. Instead of the cancelling property, a new property, that we call *translating*, on prime-order bilinear groups plays an important role in the security proof, whose existence was not known in composite-order bilinear groups. With this proof, we obtain a 2-move (i.e., round optimal) (partially) blind signature scheme (without random oracle) based on the decisional linear assumption in the common reference string model, which is of independent interest.

As the second contribution of this paper, we construct prime-order bilinear groups that possess both the cancelling and projecting properties at the same time by considering more general base groups. That is, we take a rank n $\mathbb{Z}_p$-submodule of $\mathbb{Z}_p^{n^2}$, instead of $\mathbb{Z}_p^n$, to be a base group G, and consider the projections into its rank 1 submodules. We show that the subgroup decision assumption on this base group G holds in the generic bilinear group model for $n = 2$, and provide an efficient membership-checking algorithm to G, which was trivial in the previous setting. Consequently, it is still open whether there exists a cryptosystem on composite-order bilinear groups that cannot be constructed on prime-order bilinear groups.

1 Introduction

Since Boneh, Goh, and Nissim [11] introduced composite-order bilinear groups in 2005, they have been used to solve many challenging problems in cryptography. Cryptographic systems using composite-order bilinear groups mostly utilize one of two properties, called *cancelling* and *projecting*, which Freeman [18] identified. (Though Freeman named two properties recently, these properties were already used before.) The security of almost all crypto systems using composite-order bilinear groups is based on the subgroup decision assumption, introduced by Boneh, Goh, and Nissim [11], or its variants.

Recently, some literature has aimed at constructing mathematical structures using prime-order bilinear groups with properties similar to (or richer than) composite-order bilinear groups [33, 25, 18, 20]. In particular, Freeman [18] proposed two product groups of prime-order bilinear groups with separately defined bilinear maps. He showed that two proposed product groups satisfy the subgroup decision assumption (in the sense that given g, it is infeasible to determine whether g is in a subgroup or the whole product group), and each product group with a bilinear map satisfies *cancelling* and *projecting*, respectively. One direct benefit of

* An extended abstract of this paper was presented at TCC 2012 [34]. This is the full version.

this approach is efficiency improvements of group operations and pairing computations. Loosely speaking, in bilinear groups of composite order, the group order N must be infeasible to factor so that group operations and pairing computations are less efficient than those of bilinear groups of prime order for the same security level. See [18, 20] for detailed efficiency comparison between composite-order groups and prime-order groups.

On the other hand, Meiklejohn, Shacham, and Freeman [31] gave a negative result, that is, an evidence of the limitation of constructing in some class of bilinear groups with both the *cancelling* and *projecting* properties, which is constructed on prime-order bilinear groups. To impart meaning to their result, they also proposed a round optimal blind signature scheme in composite-order bilinear groups whose security proof exploits both the *cancelling* and *projecting* properties of the composite-order bilinear group.[1] Their round optimal blind signature scheme is of independent interest since it is the first practical scheme of this type based on static assumptions (not based on q-type assumptions) in the common reference string model. They left two open questions: (1) whether the instantiation in prime-order groups of their round optimal blind signature scheme is provably secure or insecure, and (2) whether their limitation result can be applied to a wider class of bilinear groups constructed from prime-order groups.

In this paper, we answer both questions. We propose a (partially) blind signature scheme in a prime-order bilinear group setting. The proposed scheme can be considered as an adapted version of the scheme in [31] to the prime-order group setting. However, we prove the one-more unforgeability of the proposed scheme by using a completely different strategy from [31]. Our proof does not require the *cancelling* property, and instead we use another property, that we call *translating*, on prime order groups. Informally, the *translating* property is that given $g_1, g_1^a \in G_1, g_2 \in G_2$, where G_1 and G_2 are distinct subgroups of G, there exists a map $\mathcal{T}$ outputting g_2^a. The *translating* property is used, in an essential way, to prove the one-more unforgeability of the proposed scheme. With this proof, we obtain a round optimal (partially) blind signature scheme (without relying on the random oracle heuristic) based on the decisional linear assumption in the common reference string model, which is of independent interest. Our blind signature scheme is more efficient than [31]. For example, our scheme has a shorter signature size (six elements in the prime-order group vs. two elements in the composite-order group). Moreover, the security of our blind signature scheme does not rely on the factoring assumption. (The blindness of the signature scheme in [31] based on the subgroup hiding assumption, which requires that the factorization of group order N is infeasible.)

As the second contribution, we show that there exists a more general class of bilinear groups than Meiklejohn, Shacham, and Freeman considered, and some of theses can be both *cancelling* and *projecting*. That is, we take a rank n $\mathbb{Z}_p$-submodule of $\mathbb{Z}_p^{n^2}$, instead of $\mathbb{Z}_p^n$, to be a base group G, and consider the projections into its rank 1 submodules. In this case, we should carefully consider group membership tests of a subgroup. We provide an efficient membership-checking algorithm to G, which was trivial in the previous setting, and we show that the subgroup decision assumption on this base group G holds in the generic bilinear group model for $n = 2$. Consequently, it is still open as to whether there exists a cryptosystem on composite-order bilinear groups that cannot be constructed on prime-order bilinear groups.

We note that although we construct a structure satisfying both *cancelling* and *projecting*, our construction can not be applied directly to the scheme in [31] to transform it to prime-order setting. The proof of [31] uses a property of composite-order group such that two subgroups' order are relatively prime, and our construction does not support such property so that we could not apply our construction to the round optimal blind signature scheme in [31].

Related Work: Blind Signatures. Since Chaum [12, 13] introduced the concept of blind signatures in 1982, it has been studied extensively [6, 1, 7, 8, 17, 29, 32, 26, 5, 19, 4, 2, 22, 31, 3, 21] because of its numerous applications, such as electronic voting [14] and electronic cash [15]. Blind signatures are interactive protocols between a user and a signer. In blind signatures, informally, the user can obtain a signature (signed by the signer) on a message (chosen by the user) without revealing the message to the signer that is signed during the protocol; that is, the signer learns nothing about the message after finishing the protocol.

[1] The scheme in [31] itself does not use *cancelling* and *projecting*. Only the proof of security uses both *cancelling* and *projecting* properties. Thus, the authors do not rule out the existence of different proof strategy.

In particular, round optimal (i.e., 2-move) blind signature schemes have received attention since the round complexity is an important measurement of efficiency in the computer network, and round optimal blind signature schemes directly imply that they are concurrently secure. In the random oracle model, there are elegant round optimal blind signatures by Chaum [13] and Boldyreva [8]. Without relying on the random oracle heuristic, there is an approach using general NIZKs for NP, and its security depends on the assumption that a common reference string exists [17, 5]. Very recently, Garg et al. proposed the first round optimal blind signature in the standard model (without random oracle and a setup assumption such as a common reference string) [21]. These approaches without random oracle, however, are not as efficient as an approach, in which we are interested, using a bilinear map [10, 11].

In recent years several efficient round optimal blind signatures [19, 4, 2, 31, 3] have been proposed in the common reference string model, using a bilinear map, by combining signature schemes with efficient NIWI proofs [24, 23, 25]. These approaches using a bilinear map either rely on q-type dynamic assumptions [19, 4, 2, 3] or working on the composite-order group [31]. Though there is an analysis of a family of q-type dynamic assumptions by Cheon [16], the security of q-type assumptions still remains obscure. (q-type assumptions used in the above schemes hold in the generic group model [36] and these can be strong evidence for believing such assumptions. However, we believe that as the next step, constructing schemes without relying on such strong assumptions is an encouraging research approach.) In [31], a round optimal blind signature scheme based on static assumptions (not on q-type assumptions) using composite-order groups is proposed.

2 Notations and Definitions

Throughout this paper, we use notation $\oplus$ for the internal direct product: for an abelian group G, we write $G = G_1 \oplus G_2$ when G_1 and G_2 are subgroups of G and $G_1 \cap G_2 = \{1_G\}$ for the identity 1_G of G. In this case, every element g in G can be uniquely written by $g = g_1 \cdot g_2$ for some $g_1 \in G_1$ and $g_2 \in G_2$, where $\cdot$ is a group operation in G, and will be omitted sometimes. We use notation $x \xleftarrow{\$} A$. If A is a group $\mathbb{G}$, then it means that an element x is randomly chosen from $\mathbb{G}$, and if A is an algorithm, then it means that A outputs x. $[i, j]$ denotes a set of integers $\{i, \cdots, j\}$. We denote an abelian group generated by $g_1, \cdots, g_n$ by $\langle g_1, \cdots, g_n \rangle$.

We give formal definitions of bilinear group generators, and properties and cryptographic assumptions defined on the bilinear group.

Definition 1 *We say that $\mathcal{G}(\cdot, \cdot)$ is a* bilinear group generator *if it takes as input a security parameter λ and a positive integer $n \geq 1$, and it outputs a tuple $(G, G_i, H, H_i, G_t, e, \sigma | \, i \in [1, n]) \xleftarrow{\$} \mathcal{G}(\lambda, n)$, where G, H, G_t are finite abelian groups, G_i and H_i are cyclic subgroups of G and H of same order, respectively, such that $G = \oplus_{i \in [1,n]} G_i$ and $H = \oplus_{i \in [1,n]} H_i$, and $e : G \times H \to G_t$ is a non-degenerate bilinear map, that is, it satisfies*

$$\begin{aligned} &\textit{Bilinearity:} && e(g_1 g_2, h_1 h_2) = e(g_1, h_1) e(g_1, h_2) e(g_2, h_1) e(g_2, h_2) \\ & && \quad \textit{for } g_1, g_2 \in G \textit{ and } h_1, h_2 \in H, \\ &\textit{Non-degeneracy:} && \textit{for } g \in G, \textit{ if } e(g, h) = 1 \textit{ for any } h \in H, \textit{ then } g = 1, \\ & && \textit{for } h \in H, \textit{ if } e(g, h) = 1 \textit{ for any } g \in G, \textit{ then } h = 1, \end{aligned}$$

and σ is additional information for group membership-check. Moreover, we assume that group operations, random samplings, and membership-checks in G, H, and G_t and computation of e can be efficiently performed (i.e. polynomial-time in λ).

We do not exclude the case that $G = H$. When $G = H$, we say that $\mathcal{G}$ is a symmetric bilinear group generator.

Definition 2 *We say that an algorithm $\mathcal{G}_1$ is a bilinear group generator of* prime order *if $\mathcal{G}_1(\lambda) = \mathcal{G}(\lambda, 1)$, and $\mathcal{G}_1$ outputs groups G, G_1, H, H_1, G_t of prime order p and a map e. Then, $G = G_1$, $H = H_1$. We denote the three distinct groups G, H, G_t by $\mathbb{G}, \mathbb{H}, \mathbb{G}_t$, respectively, and a bilinear map e by $\hat{e}$.*

Now, we provide definitions of two properties, called *cancelling* and *projecting*, which are introduced by Freeman [18].

Definition 3 *A bilinear group generator $\mathcal{G}$ is* cancelling *if $e(g_i, h_j) = 1_t$ whenever $g_i \in G_i$, $h_j \in H_j$, and $i \neq j$, where 1_t is the identity of G_t.*

Definition 4 *A bilinear group generator $\mathcal{G}$ is* projecting *if there exist subgroups $G' \subset G$, $H' \subset H$, and $G'_t \subset G_t$, and non-trivial[2] homomorphisms $\pi : G \to G$, $\bar{\pi} : H \to H$, and $\pi_t : G_t \to G_t$ such that*

1. *$G' \subset \ker(\pi)$, $H' \subset \ker(\bar{\pi})$, and $G'_t \subset \ker(\pi_t)$.*
2. *$\pi_t(e(g, h)) = e(\pi(g), \bar{\pi}(h))$ for $\forall g \in G$ and $\forall h \in H$.*

If $\mathcal{G}$ is a symmetric bilinear group generator, that is, $G = H$, then set $G' = H'$ and $\pi = \bar{\pi}$.

To prove the security of the proposed blind signature scheme, we need two widely-known assumptions, the Computational Diffie-Hellman assumption, and k-Linear assumption which is introduced by Hofheinz and Kiltz and Shacham [27, 35], in the bilinear group setting.

Definition 5 *Let $\mathcal{G}_1$ be a bilinear group generator of prime order. We define the advantage of an algorithm $\mathcal{A}$ in solving Computational Diffie-Hellman (CDH) problem in G, denoted by $Adv^{CDHP_G}_{\mathcal{A},\mathcal{G}_1}$, is to be*

$$\Pr\Big[\mathcal{A}(G, H, G_t, e, g, g^a, g^b) \to g^{ab} : (G, H, G_t, e) \xleftarrow{\$} \mathcal{G}_1, g \xleftarrow{\$} G, a, b, \xleftarrow{\$} \mathbb{Z}_p\Big].$$

We say that $\mathcal{G}$ satisfies the Computational Diffie-Hellman (CDH) assumption in G if for any PPT algorithm $\mathcal{A}$, $Adv^{CDHP_G}_{\mathcal{A},\mathcal{G}_1}$ is a negligible function of λ.

Definition 6 *Let $\mathcal{G}_1$ be a bilinear group generator of prime order and $k \geq 1$. We define the advantage of an algorithm $\mathcal{A}$ in solving the k-Linear problem in G, denoted by $Adv^{k\text{-}Lin_G}_{\mathcal{A},\mathcal{G}_1}$, is to be*

$$\begin{aligned}
&\Big|\Pr\Big[\mathcal{A}(G, H, G_t, e, g, u_i, u_i^{a_i}, g^b, h \text{ for } i \in [1, k]) \to 1 : \\
&\qquad (G, H, G_t, e) \xleftarrow{\$} \mathcal{G}_1, g, u_i \xleftarrow{\$} G, h \xleftarrow{\$} H, a_i \xleftarrow{\$} \mathbb{Z}_p \text{ for } i \in [1, k], b \xleftarrow{\$} \mathbb{Z}_p\Big] \\
&- \Pr\Big[\mathcal{A}(G, H, G_t, e, g, u_i, u_i^{a_i}, g^b, h \text{ for } i \in [1, k]) \to 1 : \\
&\quad (G, H, G_t, e) \xleftarrow{\$} \mathcal{G}_1, g, u_i \xleftarrow{\$} G, h \xleftarrow{\$} H, a_i \xleftarrow{\$} \mathbb{Z}_p \text{ for } i \in [1, k], b = \textstyle\sum_{i\in[1,k]} a_i\Big]\Big|.
\end{aligned}$$

Then, we say that $\mathcal{G}$ satisfies the k-Linear assumption in G if for any PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ $Adv^{k\text{-}Lin_G}_{\mathcal{A},\mathcal{G}_1}$ is a negligible function of λ.

We can analogously define the CDH assumption and the k-Linear assumption in H. The *1-Linear assumption* in G is the *DDH assumption* in G and the *2-Linear assumption* in G is the *decisional linear assumption* in G.

Next, we provide the definition of the *subgroup decision assumption*, adapted from [18] to fit our purpose.

Definition 7 *Let $\mathcal{G}$ be a bilinear group generator. We define the advantage of an algorithm $\mathcal{A}$ in solving the* (n, k)-subgroup decision problem on the left*, denoted by $Adv^{SDA_L}_{\mathcal{A},\mathcal{G}}$, is to be*

$$\begin{aligned}
&\Big|\Pr\Big[\mathcal{A}(G, G', H, H', G_t, e, \sigma, g) \to 1 : \\
&\quad (G, G_i, H, H_i, G_t, e, \sigma) \xleftarrow{\$} \mathcal{G}(\lambda, n), G' := \oplus_{i\in[1,k]} G_i, H' := \oplus_{i\in[1,k]} H_i, g \xleftarrow{\$} G\Big] \\
&- \Pr\Big[\mathcal{A}(G, G', H, H', G_t, e, \sigma, g') \to 1 : \\
&\quad (G, G_i, H, H_i, G_t, e, \sigma) \xleftarrow{\$} \mathcal{G}(\lambda, n), G' := \oplus_{i\in[1,k]} G_i, H' := \oplus_{i\in[1,k]}, g' \xleftarrow{\$} G'\Big]\Big|.
\end{aligned}$$

We say that $\mathcal{G}$ satisfies the (n, k)-subgroup decision assumption on the left *if for any PPT algorithm $\mathcal{A}$, its advantage $Adv^{SDA_L}_{\mathcal{A},\mathcal{G}}$ is a negligible function in λ.*

[2] The non-triviality does not appear in the original definition [18]. Without this, however, every bilinear group can be *projecting* by using the trivial homomorphisms.

We analogously define the (n,k)-*subgroup decision assumption on the right.*

Definition 8 *We say that a bilinear group generator $\mathcal{G}(\cdot,\cdot)$ satisfies the* (n,k)-subgroup decision assumption *if $\mathcal{G}(\cdot,n)$ satisfies both the* (n,k)-subgroup decision assumptions on the left *and* on the right.

We will often omit (n,k) term, if it is clear in the context.

3 Round-Optimal Blind Signature in Prime-Order group

3.1 Symmetric Bilinear Group with Projecting Pairing

We construct a symmetric bilinear group generator with the *projecting* property. (The symmetric bilinear groups mean that $G = H$, and $G_i = H_i$ in our definition of bilinear groups.) We borrow some notations from Freeman's paper [18]. Let $\mathbb{G}$ be a group, $\mathfrak{g}, \mathfrak{g}_1, \cdots, \mathfrak{g}_n$ be elements in $\mathbb{G}$, $\overrightarrow{\alpha} = (a_1, \cdots, a_n)$ be a vector in $\mathbb{Z}_p^n$, and $M = (m_{ij})$ be an $n \times n$ matrix. We denote $\mathfrak{g}^{\overrightarrow{\alpha}} := (\mathfrak{g}^{a_1}, \cdots, \mathfrak{g}^{a_n}) \in \mathbb{G}^n$ and $(\mathfrak{g}_1, \cdots, \mathfrak{g}_n)^M := (\prod_{i\in[1,n]} \mathfrak{g}_i^{m_{i1}}, \cdots, \prod_{i\in[1,n]} \mathfrak{g}_i^{m_{in}})$. We can see that $(\mathfrak{g}^{\overrightarrow{\alpha}})^M = \mathfrak{g}^{(\overrightarrow{\alpha}M)}$. We newly define some notations useful to explain product groups. Let $G = \oplus_{i\in[1,n]} G_i$ and $H = \oplus_{j\in[1,n]} H_j$, where G_i and H_j are cyclic groups of same order. Let $e(G_i, H_j)$ be a set $\{e(g_i, h_j) | g_i \in G_i, h_j \in H_j\}$; hence $e(G_i, H_j)$ is a cyclic group since G_i and H_j are cyclic groups. In particular, when G_i and H_j have prime order p, $e(G_i, H_j)$ is a cyclic group of order p or 1.

Now, we construct a symmetric bilinear group generator $\mathcal{G}_{SP}(\lambda, 3)$, which is a generalization of Groth and Sahai's instantiation based on the decisional linear assumption [25], and is also a symmetric version of Freeman's asymmetric bilinear group generator with the *projecting* property [18].

1. $\mathcal{G}_1(\lambda) \xrightarrow{\$} (p, \mathbb{G}, \mathbb{G}_t, \hat{e})$.
2. Set $G = \mathbb{G}^3, G_t = \mathbb{G}_t^9$.
3. Choose linearly independent vectors $\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3 \in \mathbb{Z}_p^3$, and set $G_1 = \langle \mathfrak{g}^{\overrightarrow{x}_1} \rangle$, $G_2 = \langle \mathfrak{g}^{\overrightarrow{x}_2} \rangle$ and $G_3 = \langle \mathfrak{g}^{\overrightarrow{x}_3} \rangle$. Then, $G = G_1 \oplus G_2 \oplus G_3$.
4. Define a map $e : G \times G \to G_t$ by
$$= e((\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3), (\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3))$$
$$\Big(\hat{e}(\mathfrak{g}_1, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_1, \mathfrak{h}_2)^{1/2},\ \hat{e}(\mathfrak{g}_1, \mathfrak{h}_3)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_2)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_3)^{1/2},$$
$$\hat{e}(\mathfrak{g}_3, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_3, \mathfrak{h}_2)^{1/2},\ \hat{e}(\mathfrak{g}_3, \mathfrak{h}_3)^{1/2}\Big)$$
$$\cdot\Big(\hat{e}(\mathfrak{g}_1, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_3, \mathfrak{h}_1)^{1/2},\ \hat{e}(\mathfrak{g}_1, \mathfrak{h}_2)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_2)^{1/2},\ \hat{e}(\mathfrak{g}_3, \mathfrak{h}_2)^{1/2},$$
$$\hat{e}(\mathfrak{g}_1, \mathfrak{h}_3)^{1/2},\ \hat{e}(\mathfrak{g}_2, \mathfrak{h}_3)^{1/2},\ \hat{e}(\mathfrak{g}_3, \mathfrak{h}_3)^{1/2}\Big).$$
Then, $e(\mathfrak{g}^{\overrightarrow{x}}, \mathfrak{g}^{\overrightarrow{y}}) = \hat{e}(\mathfrak{g}, \mathfrak{g})^{1/2(\overrightarrow{x}\otimes\overrightarrow{y})+1/2(\overrightarrow{y}\otimes\overrightarrow{x})}$, where $\otimes$ is a tensor product (Kronecker product) of two 3-dimensions vectors.
5. For $i \in [1,3]$, define maps $\pi_i : G \to G$ and $\pi_{t,i} : G_t \to G_t$ by
$$\pi_i(g) = g^{M^{-1}U_iM} \text{ and } \pi_{t,i}(g_t) = g_t^{(M^{-1}U_iM)\otimes(M^{-1}U_iM)}, \text{ respectively,}$$
where M is a 3×3 matrix having $\overrightarrow{x}_i$ as its i-th row, U_i is a 3×3 matrix with 1 in the (i,i) entry and zeroes elsewhere, and $\otimes$ is a tensor product of matrices: For $\ell_1 \times \ell_2$ matrix $A = (a_{i,j})$ and $\ell_3 \times \ell_4$ matrix $B = (b_{i,j})$, $A \otimes B$ is a $\ell_1\ell_3 \times \ell_2\ell_4$ matrix whose (i,j)-th block is equal to $a_{i,j}B$, where we consider $A \otimes B$ as $\ell_1 \times \ell_2$ blocks. Then, π_i is a projection such that for $g_1 \in G_1, g_2 \in G_2, g_3 \in G_3$, $\pi_i(g_1g_2g_3)$ is equal to g_i.
6. Output $(p, G, G_1, G_2, G_3, G_t, e, \pi_1, \pi_2, \pi_3, \pi_{t,1}, \pi_{t,2}, \pi_{t,3})$.

We provide a useful lemma to understand the structure of the image of e.

Lemma 1 *The image of e generated by $\mathcal{G}_{SP}$ is equal to $\oplus_{1\le i\le j\le 3} e(G_i, G_j)$, and each $e(G_i, G_j)$'s order is p.*

We provide the proof of Lemma 1 in Appendix B. Non-degeneracy of e is directly coming from the lemma 1. (That is, $e(g,h) \neq 1_t$ for any non-identity elements $g, h \in G$. If not, the image is not equal to $\oplus_{1\leq i\leq j\leq 3} e(G_i, G_j)$.) The bilinear property of e can be easily checked from the bilinear property of the tensor product. Further, $\mathcal{G}_{SP}$ satisfies the *projecting* property: Let $G' = G_2 \oplus G_3$, $G'_t = \oplus_{2\leq i\leq j\leq 3} e(G_i, G_j)$, $\pi = \pi_1$, and $\pi_t = \pi_{t,1}$, where G', G'_t, π, and π_t are defined in the definition 4. Then, $G' \subset \ker(\pi)$ and $G'_t \subset \ker(\pi_t)$, and e, π, π_t satisfy the following commutative property.

$$\pi_t(e(\mathfrak{g}^{\overrightarrow{x}}, \mathfrak{g}^{\overrightarrow{y}})) = e(\pi(\mathfrak{g}^{\overrightarrow{x}}), \pi(\mathfrak{g}^{\overrightarrow{y}})).$$

We can check this commutative property as follows:

$$\begin{aligned}
&\pi_t(e(\mathfrak{g}^{\overrightarrow{x}}, \mathfrak{g}^{\overrightarrow{y}})) \\
&= \pi_{t,1}(e(\mathfrak{g}^{\overrightarrow{x}}, \mathfrak{g}^{\overrightarrow{y}})) \\
&= \pi_{t,1}(\hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}\otimes\overrightarrow{y})+1/2(\overrightarrow{y}\otimes\overrightarrow{x})}) \\
&= (\hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}\otimes\overrightarrow{y})+1/2(\overrightarrow{y}\otimes\overrightarrow{x})})^{(M^{-1}U_iM)\otimes(M^{-1}U_iM)} \\
&= \hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}\otimes\overrightarrow{y})((M^{-1}U_iM)\otimes(M^{-1}U_iM))+1/2(\overrightarrow{y}\otimes\overrightarrow{x})((M^{-1}U_iM)\otimes(M^{-1}U_iM))} \\
&= \hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}M^{-1}U_iM)\otimes(\overrightarrow{y}M^{-1}U_iM)+1/2(\overrightarrow{y}M^{-1}U_iM)\otimes(\overrightarrow{x}M^{-1}U_iM)} \\
&= e(\mathfrak{g}^{(\overrightarrow{x}M^{-1}U_iM)}, \mathfrak{g}^{(\overrightarrow{y}M^{-1}U_iM)}) \\
&= e((\mathfrak{g}^{\overrightarrow{x}})^{M^{-1}U_iM}, (\mathfrak{g}^{\overrightarrow{y}})^{M^{-1}U_iM}) \\
&= e(\pi_1(\mathfrak{g}^{\overrightarrow{x}}), \pi_1(\mathfrak{g}^{\overrightarrow{y}})) = e(\pi(\mathfrak{g}^{\overrightarrow{x}}), \pi(\mathfrak{g}^{\overrightarrow{y}})).
\end{aligned}$$

The fifth equality comes from the property of the tensor product such as $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, where A and B are matrices having ℓ columns and C and D are matrices having ℓ rows for some ℓ. (We can consider a vector as a matrix having one row.)

In contrast to the composite order bilinear group, our product group of prime order group has an additional property, we name *translating* and define as follow.

Definition 9 *A bilinear group generator $\mathcal{G}$ is* (i,j)-translating *if there exists efficiently computable (that is, polynomial time in λ) maps $\mathcal{T}_{i,j} : G_i^2 \times G_j \to G_j$ defined by $(g_i, g_i^a, g_j) \mapsto g_j^a$ and $\bar{\mathcal{T}}_{i,j} : H_i^2 \times H_j \to H_j$ defined by $(h_i, h_i^a, h_j) \mapsto h_j^a$ for an integer $a \in \mathbb{Z}$. If $\mathcal{G}$ is a symmetric bilinear group generator, then set $\bar{\mathcal{T}}_{i,j} = \mathcal{T}_{i,j}$.*

We show that the above $\mathcal{G}_{SP}$ construction satisfies *translating* property.

Theorem 1 *$\mathcal{G}_{SP}(\lambda, 3)$ satisfies* translating *property for all $i, j \in [1,3]$.*

Proof. We first construct $\mathcal{T}_{3,1}$. Given g_3^a and a 3×3 matrix M defined as in the description of $\mathcal{G}_{SP}$, we can compute g_1^a without knowing a as follows:

$$(g_3^a)^{M^{-1}} = ((\mathfrak{g}^{\overrightarrow{x}_3})^a)^{M^{-1}} = (\mathfrak{g}^{a\overrightarrow{e}_3 M})^{M^{-1}} = \mathfrak{g}^{a\overrightarrow{e}_3} = (1, 1, \mathfrak{g}^a),$$

$$(\mathfrak{g}^a, 1, 1)^M = (\mathfrak{g}^{a\overrightarrow{e}_1})^M = \mathfrak{g}^{a\overrightarrow{x}_1} = g_1^a,$$

where $\overrightarrow{e}_i$ is the canonical i-th vector in $\mathbb{Z}_p^3$, for example, $\overrightarrow{e}_1 = (1,0,0)$. We can construct other $\mathcal{T}_{i,j}$ analogously. □

Moreover, $\mathcal{G}_{SP}$ satisfies $(3,2)$-subgroup decision assumption when the underlying group generator $\mathcal{G}_1$ satisfies the decisional linear assumption.

Lemma 2 *If $\mathcal{G}_1$ satisfies the decisional linear assumption, then $\mathcal{G}_{SP}$ satisfies the $(3,2)$-subgroup decision assumption.*

We relegate the proof of Lemma 2 in Appendix B.

Remark 1. Note that $\mathcal{G}_{SP}$ does not satisfy the *cancelling* property since $e(G_i, G_j)$ is not equal to $\{1_t\}$ for $i \neq j$ (Lemma 1).

3.2 Construction

The abstract of our scheme looks very similar to the Meiklejohn et al.'s construction in the composite order bilinear group [31]. We slightly changed the Meiklejohn et al.'s construction to adapt in the prime order bilinear group setting.

(Partially) blind signature schemes in the common reference model consist of five (interactive) algorithms: Setup, KeyGen, User, Signer, and Verify. We provide the formal definition of (partially) blind signature schemes, and concurrently security, in Appendix A. We follow the security definition of [31], which is slightly stronger than [6], by allowing the adversary to choose the public key in the *blindness* definition. As a definition of the blind signature, [31] is modified from [28]; (1) it strengthens the *blindness* game to allow the adversary to generate the public key, and (2) it weakens the *one-more unforgeability* game to require that the messages (instead of pairs of message and signature) must all be distinct.[3]

The proposed partially blind signature scheme for a message space $\mathcal{M} = \{0,1\}^m$ is as follows.[4]:

- $\mathsf{Setup}(\lambda)$: $\mathcal{G}_{SP}(\lambda, 3) \xrightarrow{\$} (p, G, G_1, G_2, G_3, G_t, e, \pi_i, \pi_{t,i})$. Choose $g, u', u_1, \cdots, u_m, v_1 \cdots, v_m \xleftarrow{\$} G$, $h_1 \xleftarrow{\$} G_1$ and $h_2 \xleftarrow{\$} G_2$. Define

$$CRS = (p, G, G_t, e, g, u', u_1, \cdots, u_m, v_1, \cdots, v_m, h_1, h_2).$$

- $\mathsf{KeyGen}(CRS)$: Choose $g' \xleftarrow{\$} G$. Set $A = e(g, g')$. The public key is $PK = \{A\}$, and the secret key is $SK = \{g'\}$.
- $\mathsf{User}(CRS, PK, info, Msg)$: Let $info$ be an m_0 bits string and Msg be an $m - m_0$ bit string. We write $info$ bitwise as $b_0 \cdots b_{m_0}$ and Msg as $b_{m_0+1} \cdots b_m$. For $i \in [m_0 + 1, m]$, pick random integers $t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2}, r_i, r'_i \xleftarrow{\$} \mathbb{Z}_p$, and compute

$$\begin{aligned}
&c_i = (u_i)^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, \quad d_i = (v_i)^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}, \\
&\theta_{i,1} = u_i^{b_i s_{i,1}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}, \quad \theta_{i,2} = u_i^{b_i s_{i,2}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}, \\
&\theta_{i,3} = u_i^{(b_i - 1) s_{i,1}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}, \quad \theta_{i,4} = u_i^{(b_i - 1) s_{i,2}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}.
\end{aligned}$$

  Let $\overrightarrow{\theta}_i = (\theta_{i,1}, \cdots, \theta_{i,4})$, and send $req = \{(c_i, d_i, \overrightarrow{\theta}_i)\}_{i \in [m_0+1, m]}$ to the signer and save $state = \{(t_{i,1}, t_{i,2})\}_{i \in [m_0+1, m]}$.
- $\mathsf{Signer}(CRS, SK, info, req)$: Write $req = \{(c_i, d_i, \overrightarrow{\theta}_i)\}_{i \in [m_0+1, m]}$ and $info = b_1 \cdots b_{m_0}$. For each $i \in [m_0 + 1, m]$, verify c_i is a commitment of 0 or 1 by checking that

$$e(c_i, d_i v_i^{-1}) \stackrel{?}{=} e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2}) \text{ and } e(c_i u_i^{-1}, d_i) \stackrel{?}{=} e(h_1, \theta_{i,3}) e(h_2, \theta_{i,4}).$$

  If for some i the above equation does not hold, abort the protocol and output $\perp$. Otherwise, compute

$$c = \left(u' \prod_{i \in [1, m_0]} u_i^{b_i} \right) \left(\prod_{i \in [m_0+1, m]} c_i \right),$$

  choose a random integer $r \xleftarrow{\$} \mathbb{Z}_p$, compute

$$K_1 = g' c^r, \quad K_2 = g^{-r}, \quad K_{3,1} = h_1^{-r}, \quad K_{3,2} = h_2^{-r},$$

  send $(K_1, K_2, K_{3,1}, K_{3,2})$ to the user, and output *success* and $info$.

[3] This weakened definition is necessary if the output signature can be re-randomized. [31]'s partially blind signature and ours are in the case.

[4] For large message spaces, we can use a collision resistance hash function first.

- $\mathsf{User}(state, (K_1, K_2, K_{3,1}, K_{3,2}))$: Write $state = \{(t_{i,1}, t_{i,2})\}_{i\in[m_0+1,m]}$. Check that

$$e(K_{3,1}, g) \stackrel{?}{=} e(K_2, h_1) \text{ and } e(K_{3,2}, g) \stackrel{?}{=} e(K_2, h_2).$$

If one of two above equations is fail to hold, then abort the protocol and output $\perp$. Otherwise, unblind the signature by computing

$$S_1 = K_1 \cdot (\prod_{i\in[m_0+1,m]} K_{3,1}^{t_{i,1}} K_{3,2}^{t_{i,2}}) \text{ and } S_2 = K_2.$$

Check the validity of the signature (S_1, S_2) by running Verify. If it outputs *accept*, then go to the next step. Otherwise, abort the protocol and output $\perp$. Finally re-randomize the signature by picking a random $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and computing

$$S_1' = S_1 \cdot (u' \prod_{i\in[1,m]} u_i^{b_i})^s \text{ and } S_2' = S_2 \cdot g^{-s}.$$

Output the signature $sig = (S_1', S_2'), info$, and *success*.

- $\mathsf{Verify}(CRS, PK, info, Msg, sig)$: Write $PK = \{A\}$, $info = b_1 \cdots b_{m_0}$, $Msg = b_{m_0} \cdots b_m$, and $sig = (S_1, S_2)$. Check that

$$e(S_1, g) \cdot e(S_2, u' \prod_{i\in[1,m]} u_i^{b_i}) \stackrel{?}{=} A.$$

If the above equality holds, then output *accept*. Otherwise, output $fail$.

In the first procedure of the user, c_i and d_i are GS-commitment to b_i, and $\overrightarrow{\theta}_i$ is GS-proof that b_i satisfies the equation $b_i(b_i - 1) = 0$ so that $b_i = 0$ or $b_i = 1$. More precisely, when b_i and b_i' are openings of c_i and d_i, respectively, $\overrightarrow{\theta}_i$ is a proof that $b_i(b_i' - 1) = 0$ and $(b_i' - 1)b_i = 0$. Then, ($b_i = 0$ or $b_i' = 1$) $\bigwedge$($b_i = 1$ or $b_i' = 0$) so that $b_i = b_i' = 0$ or $b_i = b_i' = 1$. We provide three theorems to prove the security of the proposed (partially) blind signature scheme.

Theorem 2 *The above blind signature is correct.*

Theorem 3 *If $\mathcal{G}_1$ satisfies the decisional linear assumption, then the above blind signature satisfies blindness.*

The proof of Theorem 2 and 3 are similar to the previous ones [31]. We provide the proof in Appendix C.

Theorem 4 *If $\mathcal{G}_1$ satisfies the the CDH assumption, then the above blind signature is one-more unforgeable.*

We provide the proof of Theorem 4 in Appendix C. Now, we briefly explain our idea to prove the one-more unforgeability, and the reason why we cannot apply the Meiklejohn et al. proof strategy to the proposed scheme. At the end of the interaction, the user obtains a Waters-signature, which is existentially unforgeable based on the CDH assumption. If the user obtains only a Waters signature, then the proposed scheme is, loosely speaking, also one-more unforgeable. However, the user obtains not only a Waters signature (of the form $g'(u\prod_{i\in[1,m]} u_i^{b_i})^r$ and g^{-r} for message $b_1 \cdots b_m$), but also some additional information, that is, it eventually gets

$$g'(u \prod_{i\in[1,m]} u_i^{b_i})^r (\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^r, \quad g^{-r}, \quad h_1^{-r}, \text{ and } h_2^{-r}$$

for some (unknown and uniformly distributed) $r \in \mathbb{Z}_p$, and $t_{i,1}$, $t_{i,2}$, and b_i chosen by itself. Therefore, we should show that h_1^{-r}, h_2^{-r}, and $(\prod_{i\in[m_0,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^r$ will not be helpful for the user to break the one-more unforgeability. In [31], a pairing e satisfies the *cancelling* property, and orders of subgroups are relatively prime so that each part contained in each subgroup in a signature scheme is independent. [31] essentially utilized this independence. If, in our scheme, the $G_1 \oplus G_2$ part and G_3 part were independent, the user

could not obtain any additional information about the part in G_3 from the above information. (Since all information other than a Waters signature, which the user gets at the end of the protocol, is related to h_1 and h_2, which are elements in $G_1 \oplus G_2$, this information will not be helpful for forging the Waters signature in the G_3 part.) Hence, the one-more unforgeability of the scheme can be reduced to the existential unforgeability of the Waters signature (in G_3 in the case of our scheme). However, we cannot apply this Meiklejohn et al. proof strategy to our scheme since our bilinear map e does not have the *cancelling* property and each subgroup has the same order p. Instead, we prove the one-more unforgeability using a completely different strategy. Our simulation basically follows the simulation for the existential unforgeability of the Waters signature, and at the same time simulates directly additional information h_1^{-r}, h_2^{-r}, and $(\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^r$. It seems hard to simulate $(\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^r$ since $t_{i,1}$ and $t_{i,2}$ are chosen by the user and r is usually not known to the simulator during the simulation. (r is usually of the form $Ra + S$ for some unknown a and constants R and S, where a is given by the form $\mathfrak{g}^a$.) We circumvent this obstacle by using the *projecting* property and the *translating* property mentioned in section 3.1. To simulate this additional information, the simulator first extracts the message, that is, recovers $b_1 \cdots b_m$ by computing $log_{\pi_1(u_i)}\pi_1(c_i) = b_i$, and second computes $\pi_j(c_i/u_i^{b_i}) = h_j^{t_{i,j}}$ and

$$\text{if } b_i = 0, \begin{cases} \pi_3(\theta_{i,1}^{-1}) = \pi_3(v_i)^{t_{i,1}} \\ \pi_3(\theta_{i,2}^{-1}) = \pi_3(v_i)^{t_{i,2}}, \end{cases} \qquad \text{if } b_i = 1, \begin{cases} \pi_3(\theta_{i,3}) = \pi_3(v_i)^{t_{i,1}} \\ \pi_3(\theta_{i,4}) = \pi_3(v_i)^{t_{i,2}}. \end{cases}$$

Though $\pi_3(v_i)^{t_{i,j}}$ is contained in G_3, we can change it to be of the form $h_j^{at_{i,j}}$ for some unknown a by using the *translating* property mentioned in section 3.1 when v_i contains a in the exponent. The simulator can generate $(\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^r$ by using $h_j^{t_{i,j}}$ and $h_j^{at_{i,j}}$.

Remark 2. The decisional linear assumption implies the CDH assumption. (The decisional linear assumption implies the computational linear assumption, and the computational linear assumption implies the CDH assumption. Reductions are quite straightforward.)

Remark 3. In the user's first procedure, the GS-commitment and proof appear to have redundant parts. It would be more natural to change them to

$$c_i = (u_i)^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, \theta_{i,1} = (u_i^{2b_i-1} h_1^{t_{i,1}} h_2^{t_{i,2}})^{t_{i,1}} h_2^{r_i}, \theta_{i,2} = (u_i^{2b_i-1} h_1^{t_{i,1}} h_2^{t_{i,2}})^{t_{i,2}} h_1^{-r_i},$$

and it can be verified by $e(c_i, c_i u_i^{-1}) \stackrel{?}{=} e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2})$. This commitment and proof is GS commitment and proof for $b_i \in \{0, 1\}$. However, we note that in this case, we could not prove the one-more unforgeability based on the CDH assumption. We only proved the one-more unforgeability based on the decisional linear assumption and augmented CDH assumption. (Augmented CDH assumption roughly says that given $\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, \mathfrak{g}^{a^2}$, it is infeasible to compute $\mathfrak{g}^{ab}$.) To avoid requiring $\mathfrak{g}^{a^2}$, in the simulation, that is, to prove the one-more unforgeability based on the CDH assumption, we modified the commitment and the proof to the current form.

4 Bilinear Group: Both Cancelling and Projecting

4.1 Interpreting Limitation Result in [31]

In [31], the authors consider the cases that the bilinear group generator $\mathcal{G}(\lambda, n)$ is defined as follows:

1. $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e}) \stackrel{\$}{\leftarrow} \mathcal{G}_1(\lambda)$
2. $G = \mathbb{G}^n$, $H = \mathbb{G}^n$, and $G_t = \mathbb{G}_t^m$ for some positive integer m.
3. a bilinear map $e : G \times G \to G_t$ is defined by

$$\begin{aligned} e((\mathfrak{g}_1, \cdots, \mathfrak{g}_n), (\mathfrak{h}_1, \cdots, \mathfrak{h}_n)) &= (\cdots, e((\mathfrak{g}_1, \cdots, \mathfrak{g}_n), (\mathfrak{h}_1, \cdots, \mathfrak{h}_n))^{(\ell)}, \cdots) \\ &= (\cdots, \textstyle\prod_{i,j\in[1,n]} \hat{e}(\mathfrak{g}_i, \mathfrak{h}_j)^{e_{ij}^{(\ell)}}, \cdots), \end{aligned}$$

   where $e_{ij}^{(\ell)} \in \mathbb{Z}_p$ for all $i, j \in [1, n]$ and $\ell \in [1, m]$.

The authors showed that e can be both the *cancelling* and *projecting* only with negligible probability when e is defined as the above. In the above $\mathcal{G}$ construction, to generate a rank n $\mathbb{Z}_p$-module, G is defined as $\mathbb{G}^n$. In the proof for the limitation result ([31, Proposition 6.4 and Theorem 6.5]), the authors used, in an essential way, the fact that a rank n $\mathbb{Z}_p$-module is of the form $\mathbb{G}^n$.

We can, however, also define, in a different way, a rank n $\mathbb{Z}_p$-module G. First generate a rank $n'(> n)$ $\mathbb{Z}_p$-module $\tilde{G}$, and then define G as a rank n $\mathbb{Z}_p$-submodule of $\tilde{G}$. For example, define $\tilde{G} = \mathbb{G}^4$ and

$$G = \langle (\mathfrak{g}^{a_1}, \mathfrak{g}^{b_1}, \mathfrak{g}^{c_1}, \mathfrak{g}^{d_1}), (\mathfrak{g}^{a_2}, \mathfrak{g}^{b_2}, \mathfrak{g}^{c_2}, \mathfrak{g}^{d_2}), (\mathfrak{g}^{a_3}, \mathfrak{g}^{b_3}, \mathfrak{g}^{c_3}, \mathfrak{g}^{d_3}) \rangle,$$

where $\{(a_i, b_i, c_i, d_i)\}_{i\in[1,3]}$ is a set of linearly independent vectors in $\mathbb{Z}_p^4$. Then, G is a rank 3 $\mathbb{Z}_p$-submodule of a rank 4 $\mathbb{Z}_p$-module $\tilde{G}$. This example is not included in the case of the above $\mathcal{G}$ construction. In this example, we should argue about the membership check of G since any group should be easy to check for its membership to be used for cryptographic applications. If there is no additional information, the membership check of G is infeasible since it is equivalent to the decisional 3-linear problem. However, we should not rule out this case when some additional information for membership check is given. Our construction is exactly such a case.

4.2 Our Construction

First, we give an instructive intuition of our construction. To construct a bilinear group generator with *projecting*, we should consider the order of image of a bilinear map, which should be larger than prime p.[5] We start from a bilinear group generator with the *cancelling* property [18]. We consider n different bilinear group generators (of rank n) with *cancelling* property. Let $G^{(i)} = \oplus_{j\in[1,n]} G_{ij}$ (rank n $\mathbb{Z}_p$-module), $H^{(i)} = \oplus_{j\in[1,n]} H_{ij}$ (rank n $\mathbb{Z}_p$-module) and $\bar{e}_i$ (bilinear map) be the output of i-th bilinear group generator. Let $G_{ij} = \langle g_{ij} \rangle$ that is a rank 1 $\mathbb{Z}_p$-submodule of a rank n $\mathbb{Z}_p$-module. Let G_j be $\langle (g_{1j}, \cdots, g_{nj}) \rangle$, which is a rank 1 $\mathbb{Z}_p$-submodule of a rank n^2 $\mathbb{Z}_p$-module (n direct product of n $\mathbb{Z}_p$-modules). Define H_j similarly, and define $G = \oplus_{j\in[1,n]} G_j$ and $H = \oplus_{j\in[1,n]} H_j$. We define a map e by using bilinear maps $\bar{e}_i$ defined over each $G^{(i)} \times H^{(i)}$ as follows:

$$e((g_1, \cdots, g_n),\ (h_1, \cdots, h_n)) \ = \ (\bar{e}_1(g_1, h_1), \cdots, \bar{e}_n(g_n, h_n)),$$

where $g_i \in G^{(i)}$ and $h_i \in H^{(i)}$. This construction also satisfies the *cancelling* property. If we can control the basis of the image of e so that the order of image is not prime p, then we may obtain the *projecting* property.

For vectors $\mathit{\Gamma} = (\overrightarrow{\alpha}_1, \cdots, \overrightarrow{\alpha}_n) = (\alpha_{11}, \cdots, \alpha_{nn})$ and $\mathit{\Lambda} = (\overrightarrow{\beta}_1, \cdots, \overrightarrow{\beta}_n) = (\beta_{11}, \cdots, \beta_{nn}) \in \mathbb{Z}_p^{n^2}$, and a group element $\mathfrak{g} \in \mathbb{G}$, we define a notation $\mathit{\Gamma} \circ \mathit{\Lambda} := (\overrightarrow{\alpha}_1 \cdot \overrightarrow{\beta}_1, \cdots, \overrightarrow{\alpha}_n \cdot \overrightarrow{\beta}_n) \in \mathbb{Z}_p^n$, where $\overrightarrow{\alpha}_j$'s and $\overrightarrow{\beta}_j$'s are vectors in $\mathbb{Z}_p^n$, and $\overrightarrow{\alpha}_j \cdot \overrightarrow{\beta}_j = \sum_{\ell\in[1,n]} \alpha_{j\ell}\beta_{j\ell}$. Now, we describe our construction $\mathcal{G}_{CP}$.

1. Take a security parameter and a positive integer n as inputs, run $\mathcal{G}_1$, and obtain $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e})$.
2. Choose generators $\mathfrak{g}$ and $\mathfrak{h}$ at random from $\mathbb{G}$ and $\mathbb{H}$, respectively.
3. Choose $X_1, \cdots, X_n$ and D from $GL_n(\mathbb{Z}_p)$ at random. Define $D_i \in Mat_n(\mathbb{Z}_p)$ be a diagonal matrix having D's i-th column vector as its diagonal. Define Y_i by $D_i(X_i^{-1})^t$.
4. Let $\overrightarrow{\psi}_{ij}$ be the i-th row of X_j and $\overrightarrow{\phi}_{ij}$ be the i-th row of Y_j. Let $\mathit{\Psi}_i = (\overrightarrow{\psi}_{i1}, \cdots, \overrightarrow{\psi}_{in})$ and $\mathit{\Phi}_i = (\overrightarrow{\phi}_{i1}, \cdots, \overrightarrow{\phi}_{in})$. Then, define G_i by a cyclic subgroup in $\mathbb{G}^{n^2}$ generated by $\langle \mathfrak{g}^{\mathit{\Psi}_i} \rangle$, and define H_i by a cyclic group in $\mathbb{H}^{n^2}$ generated by $\langle \mathfrak{h}^{\mathit{\Phi}_i} \rangle$.
5. Define G and H by the internal direct product of G_i's and H_i's, respectively. That is, $G = \oplus_{i\in[1,n]} G_i \subset \mathbb{G}^{n^2}$, and $H = \oplus_{i\in[1,n]} H_i \subset \mathbb{H}^{n^2}$. Define G_t by $\mathbb{G}_t^n$.
6. Define a map $e : G \times H \to G_t$ as follows:

$$e(\mathfrak{g}^{\mathit{\Gamma}}, \mathfrak{h}^{\mathit{\Lambda}}) := (\prod_{\ell\in[1,n]} \hat{e}(\mathfrak{g}^{\alpha_{1\ell}}, \mathfrak{h}^{\beta_{1\ell}}), \cdots, \prod_{\ell\in[1,n]} \hat{e}(\mathfrak{g}^{\alpha_{n\ell}}, \mathfrak{h}^{\beta_{n\ell}})) \ = \ \hat{e}(\mathfrak{g}, \mathfrak{h})^{\mathit{\Gamma}\circ\mathit{\Lambda}},$$

for any $\mathit{\Gamma} = (\alpha_{11}, \cdots, \alpha_{nn})$ and $\mathit{\Lambda} = (\beta_{11}, \cdots, \beta_{nn})$.

[5] If the image of a bilinear map is prime p, it cannot satisfy *projecting* property [31].

7. Take a basis of $\langle \Psi_1, \cdots, \Psi_n \rangle^{\perp}$ at random, say $\{\hat{\Psi}_1, \cdots, \hat{\Psi}_{n^2-n}\}$, and take a basis of $\langle \Phi_1, \cdots, \Phi_n \rangle^{\perp}$ at random, say $\{\hat{\Phi}_1, \cdots, \hat{\Phi}_{n^2-n}\}$, where the notation $\langle \Gamma_1, \cdots, \Gamma_n \rangle^{\perp}$ means a set of all orthogonal vectors to $\langle \Gamma_1, \cdots, \Gamma_n \rangle$. Define
$$\sigma := (\hat{e}, \{\mathfrak{h}^{\hat{\Psi}_1}, \cdots, \mathfrak{h}^{\hat{\Psi}_{n^2-n}}\}, \{\mathfrak{g}^{\hat{\Phi}_1}, \cdots, \mathfrak{g}^{\hat{\Phi}_{n^2-n}}\}).$$
8. Output $(G, G_1, \cdots, G_n, H, H_1, \cdots, H_n, G_t, e, \sigma)$.

In the description of $\mathcal{G}_{CP}$ each G_i and H_i is defined to be rank 1, as $\mathbb{Z}_p$-submodules of $\mathbb{G}^{n^2}$, and for $i \neq j$, $G_i \cap G_j = H_i \cap H_j = \{1_{\mathbb{G}^{n^2}}\}$, where $1_{\mathbb{G}^{n^2}}$ is the identity of $\mathbb{G}^{n^2}$. Therefore, in the step 5, $G = \oplus_{i\in[1,n]} G_i$ and $H = \oplus_{i\in[1,n]} H_i$ are well-defined and rank n $\mathbb{Z}_p$-submodules of $\mathbb{G}^{n^2}$.

4.3 Cancelling, Projecting, and Translating

It is straightforward to check that e is a non-degenerate bilinear map. We show that e satisfies *cancelling*, *projecting* and *translating*.

Theorem 5 *Let $(G = \oplus_{i\in[1,n]} G_i, G_i, H = \oplus_{i\in[1,n]} H_i, H_i, G_t, e, \sigma)$ be the output of the above $\mathcal{G}_{CP}$. Then, e is both* cancelling *and* projecting.

Proof. Let $X_1, \cdots, X_n, Y_1, \cdots, Y_n$ and D be generated in the step 3 of Section 4.2. These satisfy the following three conditions.

(1) X_ℓ and Y_ℓ are in $GL_n(\mathbb{Z}_p)$ for $\ell \in [1, n]$.
(2) For $\ell \in [1, n]$ each $X_\ell \cdot Y_\ell^{\top}$ is a diagonal matrix with a diagonal $\mathbf{d}_\ell$.
(3) $D = (\mathbf{d}_1 \cdots \mathbf{d}_n)$, that is, the i-th column vector of D is $\mathbf{d}_i$.

From the condition (1) we can see that Ψ_i's are linearly independent and Φ_i's are linearly independent and so $G = \oplus_{i\in[1,n]} G_i$ and $H = \oplus_{i\in[1,n]} H_i$ are well-defined. The condition (2) guarantees that e is a *cancelling* bilinear map: For $i \neq j$, $\Psi_i \circ \Phi_j := (\overrightarrow{\psi}_{i1} \cdot \overrightarrow{\phi}_{j1}, \cdots, \overrightarrow{\psi}_{in} \cdot \overrightarrow{\phi}_{jn}) = 0$ and so $e(\mathfrak{g}^{\Psi_i}, \mathfrak{h}^{\Phi_j}) = e(\mathfrak{g}, \mathfrak{h})^{\Psi_i \circ \Phi_j} = (1_{\mathbb{G}_t}, \cdots, 1_{\mathbb{G}_t})$ is equal to the identity of the product group $(\mathbb{G}_t)^n$. The third condition (3) implies that $\{\Psi_i \circ \Phi_i\}_{i\in[1,n]}$ is a set of linearly independent vectors in $\mathbb{Z}_p^n$; hence, any pair of groups $e(G_i, H_i) = \langle e(\mathfrak{g}, \mathfrak{h})^{\Psi_i \circ \Phi_i} \rangle = \langle (\mathfrak{g}, \mathfrak{h})^{(d_{i1}, \cdots, d_{in})} \rangle$ has no common element except the identity so that $Im(e) = \oplus_{i\in[1,n]} e(G_i, H_i) = G_t$. We can consider natural projections $\pi_i : G \to G_i$, $\bar{\pi}_i : H \to H_i$, and $\pi_{t,i} : G_t \to e(G_i, H_i)$. We can construct these projections, in a similar way as the construction of the projections in the subsection 3.1. We leave the details in Appendix D. Let $G' = \oplus_{[2,n]} G_i$, $H' = \oplus_{[2,n]} H_j$, $G'_t = e(G', H')$, $\pi = \pi_i$, $\bar{\pi} = \bar{\pi}_i$, and $\pi_t = \pi_{t,i}$. Then, e satisfies the definition 4. □

Theorem 6 *$\mathcal{G}_{CP}(\lambda, n)$ satisfies* translating *property for all $i, j \in [1, n]$.*

Proof. We will construct $\mathcal{T}_{3,1}$. We can construct other $\mathcal{T}_{i,j}$ and $\bar{\mathcal{T}}_{i,j}$ similarly. Given g_3, g_3^a and $n \times n$ matrices X_i defined as in the description of $\mathcal{G}_{CP}$, we can compute g_1^a without knowing a as follows:

$$\text{Parse } g_3^a \text{ as } (\mathfrak{g}^{\Psi_3})^a = ((\mathfrak{g}^{\overrightarrow{\psi}_{31}})^a, \cdots, (\mathfrak{g}^{\overrightarrow{\psi}_{3n}})^a), \text{ and compute}$$

$$\text{for } j \in [1, n],\ ((\mathfrak{g}^{\overrightarrow{\psi}_{3j}})^a)^{X_j^{-1}} = (\mathfrak{g}^{a\overrightarrow{e}_3 X_j})^{X_j^{-1}} = \mathfrak{g}^{a\overrightarrow{e}_3} = (1, 1, \mathfrak{g}^a, \cdots, 1),$$

$$(\mathfrak{g}^a, 1, \cdots, 1)^{X_j} = (\mathfrak{g}^{a\overrightarrow{e}_1})^{X_j} = \mathfrak{g}^{a\overrightarrow{\psi}_{1j}},$$

$$\text{then } (\mathfrak{g}^{a\overrightarrow{\psi}_{11}}, \cdots, \mathfrak{g}^{a\overrightarrow{\psi}_{1n}}) = (\mathfrak{g}^{\Psi_1})^a = g_1^a.$$

where $\overrightarrow{e}_i$ is the canonical i-th vector in $\mathbb{Z}_p^n$, for example, $\overrightarrow{e}_1 = (1, 0, 0, \cdots, 0)$. □

We show that anyone knowing σ can test membership of elements in G and H (membership test for G_t is trivial) in Appendix D. Finally, we should show that $\mathcal{G}$ satisfies the subgroup decision assumption, but it is not easy to prove that $\mathcal{G}$ satisfies the subgroup decision for any n. Instead, in Appendix D we give

a proof that, for $n = 2$, $\mathcal{G}$ satisfies the $(2,1)$-subgroup decision assumption in the generic bilinear group model [36] (that is, we assume that the adversary should access the oracles for group operations of $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_t$ and pairing computations for $\hat{e}$, where $\mathcal{G}_1 \to (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e})$). Though we give a proof for the case $n = 2$, we are positive that $\mathcal{G}_{CP}$ satisfies the subgroup decision assumption for $n > 2$. For $n > 2$, there are several variables, particularly in σ, we should consider for the subgroup decision assumption, so these make it hard to prove for the case $n > 2$, even in the generic bilinear group model.[6]

5 Conclusions and Further Work

In this paper, we answered two open questions left by Meiklejohn, Shacham, and Freeman. First, we showed that the security of the Meiklejohn et al.'s (partial) blind signature can be proved in the prime-order bilinear group setting.[7] Second, we showed that there exist bilinear group generators that are both *cancelling* and *projecting* in the prime-order bilinear group setting.

The proof of the Meiklejohn-Shacham-Freeman blind signature scheme, and the Lewko-Waters identity-based encryption scheme [30] essentially use the fact that orders of subgroups are relatively prime as well as the projecting and/or cancelling properties. For each scheme, the adapted version in prime-order bilinear groups is proposed, with a different security proof strategy, in this paper and [30], respectively. It would be interesting to find a general procedure to transform such schemes using relatively prime orders in composite-order groups to schemes in prime-order groups.

We proposed a new mathematical framework with both *cancelling* and *projecting* in a prime-order bilinear group setting, and gave the proof that the $(2,1)$ subgroup decision assumption holds in the generic bilinear group model when $n = 2$. This research leaves many interesting open problems. We ask if the subgroup decision assumption holds when $n > 2$, and if the subgroup decision assumption can be reduced to the simple assumption such as the (decisional) k-linear assumption. We did not find good cryptographic applications of this framework. It would be interesting to design cryptographic schemes based on the proposed framework. We expect that this research will provide other directions for our primitive question: whether there exists a cryptosystem on composite-order bilinear groups that cannot be constructed on prime-order bilinear groups.

References

1. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, 2001.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, 2010.
3. M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signature in asymmetric bilinear groups. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, 2011.
4. M. Abe, K. Haralambiev, and M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. In *Cryptology ePrint Archive, Report 2010/133*. http://eprint.iacr.org/2010/133, 2010.
5. M. Abe and M. Ohkubo. A framework for universally composable non-committing blind signatures. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 435–450. Springer, 2009.
6. M. Abe and T. Okamoto. Provably secure partially blind signatures. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, 2000.
7. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. In *Journal of Cryptology*, volume 16, pages 185–215, 2003.
8. A. Boldyreva. Trheshold signature, multisignature and blind signature schemes based on the gap-diffie-hellman-group signature scheme. In *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, 2003.

[6] All variables in σ is public, so to show that $\mathcal{G}_{CP}$ satisfies the subgroup decision assumption, the simulator should simulate σ in the proof.

[7] We modified their scheme slightly to prove its security under the CDH assumption. We remark that, however, the security of the direct instantiation of their scheme in the prime-order bilinear group can also be proven secure under the decisional linear assumption and the augmented CDH assumption, which is stronger than the CDH assumption.

9. D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertexts. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer-Verlag, 2005.
10. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–23. Springer-Verlag, 2001.
11. D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC 2005*, volume 3378 of *LNCS*. Springer-Verlag, 2005.
12. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.
13. D. Chaum. Blind signature system. In *CRYPTO 1983*, 1983.
14. D. Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking rsa. In *EUROCRYPT*, volume 330 of *LNCS*, pages 177–182. Springer, 1988.
15. D. Chaum, A. Fiat, and M. Naor. Untraceable electoronic cash. In *CRYPTO*, volume 403 of *LNCS*, pages 319–327. Springer, 1988.
16. J. H. Cheon. Discrete logarithm problems with auxiliary inputs. In *Journal of Cryptology*, volume 23, pages 457–476, 2010.
17. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, 2006.
18. D. M. Freeman. Converting pairing-based cryptosystems from composite-orde groups to prime-order groups. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, (full version is available from http://eprint.iacr.org/2009/540) (full version is available from http://eprint.iacr.org/2009/540), 2010.
19. G. Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. In *Cryptology ePrint Archive, Report 2009/320*. http://eprint.iacr.org/2009/320, 2009.
20. S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In *ACM Conference on Computer and Communications Security*, pages 121–130. ACM, 2010.
21. S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signature. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648. Springer, 2011.
22. E. Ghadafi and N. Smart. Efficient two-move blind signatures in the common reference string model. In *Cryptology ePrimt Archive, Report 2010/568*. http://eprint.iacr.org/2010/568, 2010.
23. J. Groth, R. Ostrovsky, and A. Sahai. Non-interactive zaps and new techniques for nizk. In *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, 2006.
24. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero-knolwedge for np. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 339–358. Springer, 2006.
25. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
26. C. Hazay, J. Katz, C.-Y. Koo, and Y. Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In *TCC 2007*, volume 4392 of *LNCS*, pages 323–341. Springer, 2007.
27. D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, volume 4622 of *LNCS*, pages 553–571. Springer, 2007.
28. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 150–164. Springer, 1997.
29. A. Kiayias and H.-S. Zhou. Concurrently-secure blind signatures without random oracles. In *SCN 2006*, volume 4116 of *LNCS*, pages 49–62. Springer, 2006.
30. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, 2010.
31. S. Meiklejohn, H. Shacham, and D. M. Freeman. Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 519–538. Springer, 2010.
32. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, 2006.
33. T. Okamoto and K. Takashima. Homomorphic encryption and signature from vector decomposition. In *Pairing*, volume 5209 of *LNCS*, pages 57–74. Springer, 2008.
34. J. H. Seo and J. H. Cheon. Beyond the limitation of prime-order groups, and round optimal blind signatures. In *TCC*, volume 7194 of *LNCS*, pages 133–150. Springer, 2012.
35. H. Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. In *Cryptology ePrimt Archive, Report 2007/074*. http://eprint.iacr.org/2007/074, 2007.
36. V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266. Springer, 1997.

37. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, 2005.

A Definition of Blind Signatures

In this section, we recall the definition of partially blind signature from [31]. We assume that both signer and user agree on the common information, denoted as $info$, and $info$ is decided outside of the partially blind signature scheme. In some applications, $info$ is may be decided by the signer, or the user. In our definition of the (concurrently secure) partially blind signature, we want the signature scheme to be secure regardless of the process of deciding $info$.

Definition 10 *A partially blind signature in the common reference string (CRS) model is a collection of five (interactive) algorithms.*

- *Setup is a PPT algorithm that takes a security parameter λ and outputs a common reference string, denoted as CRS.*
- *KeyGen is a PPT algorithm, on input CRS, outputs a public and secret key pair (PK, SK).*
- *Signer and User are PPT (interactive) algorithms. Signer takes CRS, SK and $info$ as input, and User takes $CRS, PK, info$, and a message $Msg \in \{0,1\}^m$ as input. At the end of interaction, if the interaction is successful, then Signer outputs 'success', and User outputs 'success' and the unblinded signature 'sig'.*
- *Verify is a (probabilistic) polynomial-time algorithm that takes $(CRS, PK, info, Msg, sig)$ and outputs either 'accept' or 'fail'.*

Definition 11 *We say that a partially blind signature scheme (in the CRS model) is concurrently secure if for all PPT algorithm $\mathcal{A}$ there exists a negligible function $\eta(\cdot)$ and a security parameter λ_0 such that for all $\lambda > \lambda_0$ the following three properties hold:*

1. *Correctness: For all $CRS \xleftarrow{\$} Setup(\lambda)$, $info \in \{0,1\}^m$ ($info$ is actually m_{info} bit string such that $m_{info} < m$) and $(PK, SK) \xleftarrow{\$} \mathsf{KeyGen}(CRS)$, if sig is the output of $\mathsf{User}(CRS, PK, info, Msg) \leftrightarrow \mathsf{Signer}(CRS, SK, info)$ for an honest user and an honest signer, then $\mathsf{Verify}(CRS, PK, info, Msg, sig)$ outputs accept with probability 1.*
2. *Blindness: Let $b \xleftarrow{\$} \{0,1\}$ be unknown to $\mathcal{A}$ (roll of a signer). Define the following game:*
   (a) $CRS \xleftarrow{\$} \mathsf{Setup}(\lambda)$.
   (b) $(info, Msg_0, Msg_1, PK) \leftarrow \mathcal{A}(CRS)$.
   (c) $\mathcal{A}$ engages in two arbitrary interleaved signing protocols; one with $\mathsf{User}(CRS, PK, info, Msg_b)$ and one with $\mathsf{User}(CRS, PK, info, Msg_{1-b})$ (where both users act honestly).
   (d) If the first user outputs sig_b and the second user outputs sig_{1-b} (i.e., both users succeed) then $\mathcal{A}$ is given sig_0 and sig_1.
   (e) In the end of interaction, $\mathcal{A}$ outputs a bit b'.
   The signature scheme is considered blind if the probability that $b' = b$ is at most $\frac{1}{2} + \eta(\lambda)$, where the probability goes over the choices of b, the randomness used in Setup, and the randomness used by $\mathcal{A}$ and users.
3. *One-more unforgeability: Define the following game for the adversary $\mathcal{A}$ (roll of a user).*
   (a) $CRS \xleftarrow{\$} \mathsf{Setup}(\lambda)$.
   (b) $(PK, SK) \xleftarrow{\$} \mathsf{KeyGen}(CRS)$.
   (c) $\mathcal{A}$, on input CRS and PK, engages in $poly(\lambda)$ arbitrarily interleaved executions of the signing protocol with polynomially many copies of $\mathsf{Signer}(CRS, SK, info)$, where $info$ is sent by $\mathcal{A}$, (on and messages of its choice). Let q_{info} denote the number of executions in which for a common input $info$ the signer outputs success at the end. For $info$ that have never sent to the signer, define $q_{info} = 0$. (For $info = \perp$ (that is, $m_{info} = 0$), q_{info} is defined in the same manner.)
   (d) For some $info$, $\mathcal{A}$ outputs a collection of message-signature pairs $\{(Msg_i, sig_i)\}_{i \in [1,q']}$ such that $Msg_i \neq Msg_j$ for all $i \neq j$, and $\mathsf{Verify}(CRS, PK, info, Msg_i, sig_i) = success$ for all $i \in [1, q']$.
   We say that the signature scheme is one-more unforgeable if the probability that $q' > q_{info}$ is at most $\eta(\lambda)$, where the probability is taken over the randomness used in Setup, KeyGen, $\mathcal{A}$ and Signer.

B Proof of Lemmas

Proof of Lemma 1. We define notation: For subgroups $S_1, \cdots, S_n$ of an abelian group S with group operator $\cdot$, let $\prod_{i\in[1,n]} S_i$ be a subgroup $\{s_1 \cdots s_n | s_i \in S_i \text{ for } \forall i \in [1,n]\}$.

From the bilinearity of e, the image of e is equal to $\prod_{1\le i\le j\le 3} e(G_i, G_j)$. Let $G_i = \langle \mathfrak{g}^{\overrightarrow{x}_i} \rangle$. Since $e(\mathfrak{g}^{\overrightarrow{x}_i}, \mathfrak{g}^{\overrightarrow{x}_j}) = \hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}_i \otimes \overrightarrow{x}_j + \overrightarrow{x}_j \otimes \overrightarrow{x}_i)}$, showing that for $1 \le i \le j \le 3$, $\overrightarrow{x}_i \otimes \overrightarrow{x}_j + \overrightarrow{x}_j \otimes \overrightarrow{x}_i$'s are linear independent vectors in $\mathbb{Z}_p^9$ is sufficient to prove the lemma. Suppose that for $1 \le i \le j \le 3$, $\overrightarrow{x}_i \otimes \overrightarrow{x}_j + \overrightarrow{x}_j \otimes \overrightarrow{x}_i$'s are linear dependent. Then, there exists $a_{ij} \in \mathbb{Z}_p$ for $i, j \in [1,3]$ such that $\sum_{i,j\in[1,3]} a_{ij}(\overrightarrow{x}_i \otimes \overrightarrow{x}_j) = 0$ and a_{ij}'s are not all zero. Let $\overrightarrow{x}'_j$ be $\sum_{i\in[1,3]} a_{ij} \overrightarrow{x}_i$. Then, $\sum_{j\in[1,3]} \overrightarrow{x}'_j \otimes \overrightarrow{x}_j = 0$. At least one of $\overrightarrow{x}'_1$, $\overrightarrow{x}'_2$ and $\overrightarrow{x}'_3$ is a non-zero vector since a_{ij}'s are not all zero, and $\overrightarrow{x}_1$, $\overrightarrow{x}_2$ and $\overrightarrow{x}_3$ are linearly independent. Without loss of generality, we assume that $\overrightarrow{x}'_1$ is a non-zero vector, and the first entry of $\overrightarrow{x}'_1$ is non-zero. From the first three entries of $\sum_{j\in[1,3]} \overrightarrow{x}'_j \otimes \overrightarrow{x}_j$, we obtain an equation $x'_{11}\overrightarrow{x}_1 + x'_{21}\overrightarrow{x}_2 + x'_{31}\overrightarrow{x}_3 = 0$, where x'_{j1} is the first entry of $\overrightarrow{x}'_j$. Since $x'_{11} \neq 0$, it is a contradiction to the linearly independency of $\overrightarrow{x}_1$, $\overrightarrow{x}_2$ and $\overrightarrow{x}_3$. Therefore, for $1 \le i \le j \le 3$, $\overrightarrow{x}_i \otimes \overrightarrow{x}_j + \overrightarrow{x}_j \otimes \overrightarrow{x}_i$'s are linear independent vectors so that for $1 \le i \le j \le 3$, each $e(G_i, G_j) = \langle \hat{e}(\mathfrak{g},\mathfrak{g})^{1/2(\overrightarrow{x}_i \otimes \overrightarrow{x}_j + \overrightarrow{x}_j \otimes \overrightarrow{x}_i)} \rangle$ is mutually disjoint (except the identity) to the other subgroup. Therefore, the image of e is equal to $\oplus_{1\le i\le j\le 3} e(G_i, G_j)$. □

Proof of Lemma 2. Suppose that there exist an algorithm $\mathcal{A}$ to break the subgroup decision assumption of $\mathcal{G}_{SP}$. We construct an algorithm $\mathcal{B}$ to attack the decisional linear assumption of $\mathcal{G}_1$ by using $\mathcal{A}$. First, $\mathcal{B}$ is given the decisional linear problem $(p, \mathbb{G}, \mathbb{G}_t, e, \mathfrak{g}, \mathfrak{g}^{z_1}, \mathfrak{g}^{z_2}, \mathfrak{g}^{z_1 z_3}, \mathfrak{g}^{z_2 z_4}, \mathfrak{g}^{z})$. The goal of $\mathcal{B}$ is to determine whether $z = z_3 + z_4$ or $z \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{B}$ sets G, G_t and e according to the description of $\mathcal{G}_{SP}$. $\mathcal{B}$ chooses random integers $r_1, r_2, s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p$, and computes $h_1 = (\mathfrak{g}^{z_1 z_3}, \mathfrak{g}^{z_1 r_1}, \mathfrak{g}^{z_1 s_1})$ and $h_2 = (\mathfrak{g}^{z_2 z_4}, \mathfrak{g}^{z_2 r_2}, \mathfrak{g}^{z_2 s_2})$ as generators of G_1 and G_2, respectively. Next, $\mathcal{B}$ sets $\tilde{g} = (\mathfrak{g}^{z}, \mathfrak{g}^{r_1+r_2}, \mathfrak{g}^{s_1+s_2})$ and send $\tilde{g}$ along with the group description to $\mathcal{A}$. Finally, $\mathcal{B}$ receives $\mathcal{A}$'s result and outputs it as his result. If $z = z_3 + z_4$, then $\tilde{g}$ is uniformly distributed in $\langle h_1, h_2 \rangle = G_1 \oplus G_2$, and otherwise, $\tilde{g}$ is uniformly distributed in G. Therefore, $\mathcal{B}$ can attack the subgroup decision assumption with the same advantage as $\mathcal{A}$'s advantage to solve the decisional linear problem. □

C Proof of Theorems

C.1 Proof of Theorem 2

We show that the correctness of the proposed blind signature in the section 3.2. If the user correctly constructs c_i, $\overrightarrow{\theta}_i$ in the first procedure, then they will pass the signer's test since

$$
\begin{aligned}
e(c_i, d_i v_i^{-1}) &= e(u_i^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= e(u_i^{b_i}, v_i^{b_i-1}) e(u_i^{b_i}, h_1^{s_{i,1}} h_2^{s_{i,2}}) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= 1 \cdot e((u_i^{b_i})^{s_{i,1}}, h_1) e((u_i^{b_i})^{s_{i,2}}, h_2) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= 1 \cdot e((u_i^{b_i})^{s_{i,1}}, h_1) e((u_i^{b_i})^{s_{i,2}}, h_2) e(h_1, (v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, (v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \\
&= e(h_1, u_i^{b_i s_{i,1}} (v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, u_i^{b_i s_{i,2}} (v_i^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \\
&= e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2}),
\end{aligned}
$$

$$
\begin{aligned}
e(c_i u_i^{-1}, d_i) &= e(u_i^{b_i-1} h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= e(u_i^{b_i-1}, v_i^{b_i}) e(u_i^{b_i-1}, h_1^{s_{i,1}} h_2^{s_{i,2}}) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= 1 \cdot e((u_i^{b_i-1})^{s_{i,1}}, h_1) e((u_i^{b_i-1})^{s_{i,2}}, h_2) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= 1 \cdot e((u_i^{b_i-1})^{s_{i,1}}, h_1) e((u_i^{b_i-1})^{s_{i,2}}, h_2) e(h_1, (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \\
&= e(h_1, u_i^{(b_i-1)s_{i,1}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, u_i^{(b_i-1)s_{i,2}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \\
&= e(h_1, \theta_{i,3}) e(h_2, \theta_{i,4}).
\end{aligned}
$$

Further, if the signer correctly follows the protocol, $K_1 = g'(u'(\prod_{i\in[1,m]} u_i^{b_i})(\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}}))^r$, $K_2 = g^{-r}$, $K_{3,1} = h_1^{-r}$, and $K_{3,2} = h_2^{-r}$. They will pass the user's test in the second procedure since

$$e(K_{3,1}, g) = e(h_1^{-r}, g) = e(h_1, K_2) \text{ and } e(K_{3,2}, g) = e(h_2^{-r}, g) = e(K_2, g).$$

Finally, the user outputs a signature as

$$S_1 = K_1 \cdot (\prod_{i\in[m_0+1,m]} K_{3,1}^{t_{i,1}} K_{3,2}^{t_{i,2}}) \cdot (u' \prod_{i\in[1,m]} u_i^{b_i})^s = g'(u' \prod_{i\in[1,m]} u_i^{b_i})^{r+s} \text{ and } S_2 = K_2 \cdot g^{-s} = g^{-(r+s)},$$

which is a valid signature with the randomness $r + s$. □

C.2 Proof of Theorem 3

We show that if $\mathcal{G}_1$ satisfies the decisional linear assumption, then the proposed blind signature satisfies the blindness definition.

We prove the theorem by using hybrid arguments. First, we define a sequence of games. The only different part between adjacent games is the distribution of CRS. For $i \in [0, m]$,

$Game_i$: Group description in CRS is equal to the real game, that is, p, G, G_t, e are normally generated. Generate $g, u' \xleftarrow{\$} G$,$h_1 \xleftarrow{\$} G_1$, and $h_2 \xleftarrow{\$} G_2$. Set $u_1, \cdots, u_i, v_1, \cdots, v_i \xleftarrow{\$} G_1 \oplus G_2$, and $u_{i+1}, \cdots, u_m, v_{i+1}, \cdots, v_m \xleftarrow{\$} G$. Then, normally follow the game procedure.

$Game'_i$: Same game to $Game_i$ except choosing $v_i \xleftarrow{\$} G$.

$Game_0$ is equal to the real game, and in $Game_m$ all u_i and v_i are randomly chosen from $G_1 \oplus G_2$. By Lemma 2, any polynomial time algorithm has negligibly different advantage between $Game'_i$ and $Game_i$, and between $Game_i$ and $Game'_{i+1}$. Then, the triangle inequality law implies the following lemma.

Lemma 3 *No polynomial time algorithm has negligibly different advantage between $Game_0$ and $Game_m$.*

Next, we will show that any algorithm cannot obtain any information about the user's message in $Game_m$. That is, we will prove the following lemma.

Lemma 4 *Any (unbounded) algorithm has no advantage in $Game_m$.*

From the above two lemmas, we can complete the proof of theorem.

Now, we prove the lemma 4. In the game of blindness, the only chances that the adversary can obtain information about each user's message are (1) the user's commitment to the message and its proof, (2) the user's response, that is, whether the user accepts the adversary's output $(K_1, K_2, K_{3,1}, K_{3,2})$, and (3) the user's output signature. Now, we show that no adversary can obtain any information from the three aforementioned resources in $Game_m$ even for the unbounded adversary.

The user's commitment to the message and its proof. We used GS-commitment and proof so that in $Game_m$ each user's commitment and proof identically distribute regardless of their witness. ($Game_m$ is a *witness indistinguishable setting of GS-proof.*) More precisely, the distribution of $c_i, d_i, \theta_{i,1}, \theta_{i,2}, \theta_{i,3}$ and $\theta_{i,4}$ when $b_i = 0$, is identical to the distribution when $b_i = 1$. Further, all tuple of $(c_i, d_i, \theta_{i,1}, \theta_{i,2}, \theta_{i,3}, \theta_{i,4})$ for $i \in [m_0 + 1, m]$ are independent. Therefore, the adversary cannot obtain any information about the message from the user's commitment to the message and its proof. Let us explain the detail.

When $b_i = 0$,

$$c_i = h_1^{t_{i,1}} h_2^{t_{i,2}}, \quad \theta_{i,1} = (v_i^{-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}, \quad \theta_{i,2} = (v_i^{-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}.$$

$$d_i = h_1^{s_{i,1}} h_2^{s_{i,2}}, \quad \theta_{i,3} = u_i^{(-1)s_{i,1}} (h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}, \quad \theta_{i,4} = u_i^{(-1)s_{i,2}} (h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}.$$

where $t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2}, r_i$ and r'_i are uniformly and independently distributed in $\mathbb{Z}_p$.

When $b_i = 1$,

$$c_i = u_i h_1^{t_{i,1}} h_2^{t_{i,2}}, \quad \theta_{i,1} = u_i^{s_{i,1}} (h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}, \quad \theta_{i,2} = u_i^{s_{i,2}} (h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}.$$

$$d_i = v_i h_1^{s_{i,1}} h_2^{s_{i,2}}, \quad \theta_{i,3} = (v_i h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}, \quad \theta_{i,4} = (v_i h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}.$$

where $t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2}, r_i$ and r'_i are uniformly and independently distributed in $\mathbb{Z}_p$. Let $u_i = h_1^a h_2^b$ and $v_i = h_1^c h_2^d$ since $u_i, v_i \in \langle h_1, h_2 \rangle = G_1 \oplus G_2$ in $Game_m$. Let $\tilde{t}_{i,1} = t_{i,1} + a$, $\tilde{t}_{i,2} = t_{i,2} + b$, $\tilde{s}_{i,1} = s_{i,1} + c$, $\tilde{s}_{i,2} = s_{i,2} + d$, $\tilde{r}_i = b(\tilde{s}_{i,1} - c) + a(\tilde{s}_{i,2} - d) + r_i$ and $\tilde{r}' = \tilde{s}_{i,2}(\tilde{t}_{i,1} - a) + r'$. Then, $\tilde{t}_{i,1}$, $\tilde{t}_{i,2}$, $\tilde{s}_{i,1}$, $\tilde{s}_{i,2}$, $\tilde{r}_i$ and $\tilde{r}'_i$ are uniformly and independently distributed in $\mathbb{Z}_p$, and

$$c_i = h_1^{\tilde{t}_{i,1}} h_2^{\tilde{t}_{i,2}}, \quad \theta_{i,1} = (v_i^{-1} h_1^{\tilde{s}_{i,1}} h_2^{\tilde{s}_{i,2}})^{\tilde{t}_{i,1}} h_2^{\tilde{r}_i}, \quad \theta_{i,2} = (v_i^{-1} h_1^{\tilde{s}_{i,1}} h_2^{\tilde{s}_{i,2}})^{\tilde{t}_{i,2}} h_1^{-\tilde{r}_i}.$$

$$d_i = h_1^{\tilde{s}_{i,1}} h_2^{\tilde{s}_{i,2}}, \quad \theta_{i,3} = u_i^{(-1)\tilde{s}_{i,1}} (h_1^{\tilde{s}_{i,1}} h_2^{\tilde{s}_{i,2}})^{\tilde{t}_{i,1}} h_2^{\tilde{r}'_i}, \quad \theta_{i,4} = u_i^{(-1)\tilde{s}_{i,2}} (h_1^{\tilde{s}_{i,1}} h_2^{\tilde{s}_{i,2}})^{\tilde{t}_{i,2}} h_1^{-\tilde{r}'_i},$$

and hence the distribution when $b_i = 1$ is identical to the distribution when $b_i = 0$.

The user's response. When the adversary gives $(K_1, K_2, K_{3,1}, K_{3,2})$, the user performs two types of tests. The one type of test is

$$e(K_{3,1}, g) \stackrel{?}{=} e(K_2, h_1) \text{ and } e(K_{3,2}, g) \stackrel{?}{=} e(K_2, h_2),$$

the other one is $\mathsf{Verfy}(CRS, PK, info, Msg, (S_1, S_2))$, where

$$S_1 = K_1 \cdot (\prod_{i \in [m_0+1, m]} K_{3,1}^{t_{i,1}} K_{3,2}^{t_{i,2}}) \text{ and } S_2 = K_2,$$

and $t_{i,1}$ and $t_{i,2}$ are used in the commitment c_i and its proof $\theta_{i,1}$ and $\theta_{i,2}$. We will show that the adversary can perform these two types of test by himself so that the adversary cannot obtain any information from the user's response. The adversary can trivially check the first type of test by himself. Now, we consider the user's second test $\mathsf{Verfy}(CRS, PK, info, Msg, (S_1, S_2))$. Let $info \| Msg = b_1 \| \cdots \| b_m$.

$$\begin{aligned}
A &\stackrel{?}{=} e(S_1, g) e(S_2, u' \textstyle\prod_{i \in [1,m]} u_i^{b_i}) \\
&= e(K_1 \cdot (\textstyle\prod_{i \in [m_0+1,m]} K_{3,1}^{t_{i,1}} K_{3,2}^{t_{i,2}}), g) e(K_2, u' \textstyle\prod_{i \in [1,m]} u_i^{b_i}) \\
&= e(K_1, g) (\textstyle\prod_{i \in [m_0+1,m]} e(K_{3,1}, g)^{t_{i,1}} e(K_{3,2}, g)^{t_{i,2}})) e(K_2, u' \textstyle\prod_{i \in [1,m]} u_i^{b_i}) \\
&= e(K_1, g) (\textstyle\prod_{i \in [m_0+1,m]} e(K_2, h_1)^{t_{i,1}} e(K_2, h_2)^{t_{i,2}})) e(K_2, u' \textstyle\prod_{i \in [1,m]} u_i^{b_i}) \\
&= e(K_1, g) (\textstyle\prod_{i \in [m_0+1,m]} e(K_2, h_1^{t_{i,1}} h_2^{t_{i,2}})) e(K_2, u' \textstyle\prod_{i \in [1,m]} u_i^{b_i}) \\
&= e(K_1, g) e(K_2, u' (\textstyle\prod_{i \in [1,m]} u_i^{b_i}) (\textstyle\prod_{i \in [m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})) \\
&= e(K_1, g) e(K_2, u' (\textstyle\prod_{i \in [1,m_0]} u_i^{b_i}) (\textstyle\prod_{i \in [m_0+1,m]} c_i))
\end{aligned}$$

The adversary can see whether the fourth equality holds or not, from the first test. Since the user performs the second test only if the first test are passed, the adversary can check by testing the first test and testing $A \stackrel{?}{=} e(K_1, g) e(K_2, u' (\prod_{i \in [1,m_0]} u_i^{b_i}) (\prod_{i \in [m_0+1,m]} c_i))$ whether the user accepts the second test.

The user's output signature. The output signature of each user is re-randomized so that the randomness uniformly distributes. Therefore the adversary cannot obtain any information about the underlying message from two signatures. □

C.3 Proof of Theorem 4

We show that if $\mathcal{G}_1$ satisfies the the CDH assumption, then the proposed blind signature is one-more unforgeable (in the sense of the one-more unforgeability definition in Appendix A).

Suppose that there exists a polynomial algorithm $\mathcal{A}$ to break the one-more unforgeable property in the game defined in Appendix A, with ϵ success probability. Let q_{info} be the number of signing queries for the common information $info$, and let q be the sum of q_{info} for all $info$ issued by $\mathcal{A}$. Let ϵ be the success probability of $\mathcal{A}$ in the one-more unforgeability game. We construct an algorithm $\mathcal{B}$ to attack the CDH assumption by using $\mathcal{A}$, with more than $\frac{\epsilon}{8(m+1)q}$ success probability so that we prove there is no such polynomial time adversary $\mathcal{A}$ to break the one-more unforgeable property in the sense of the definition in Appendix A.

$\mathcal{B}$ starts with receiving $(p, \mathbb{G}, \mathbb{G}_t, e, \mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b)$. The goal of $\mathcal{B}$ is to compute $\mathfrak{g}^{ab}$.

Setup. $\mathcal{B}$ first generates CRS of the blind signature. $\mathcal{B}$ sets an integer, $\ell = 4q$, chooses a random integer $k \xleftarrow{\$} [0, m]$, $z', z_j \xleftarrow{\$} [0, \ell - 1]$, and $w'_i, w_{ij}, \bar{w}_{ij} \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [1, 3]$ and $j \in [1, m]$.

Define five functions

$$
\begin{aligned}
&F(Msg) = (p - \ell k) + z' + \textstyle\sum_{i\in[1,m]} b_i z_i,\\
&J_1(Msg) = w'_1 + \textstyle\sum_{i\in[1,m]} b_i w_{1,i},\\
&J_2(Msg) = w'_2 + \textstyle\sum_{i\in[1,m]} b_i w_{2,i},\\
&J_3(Msg) = w'_3 + \textstyle\sum_{i\in[1,m]} b_i w_{3,i},\\
&K(Msg) = \begin{cases} 0, & \text{if } z' + \sum_{i\in[1,m]} b_i z_i \equiv 0 \ (\mod \ell)\\ 1, & \text{otherwise,}\end{cases}\\
&\text{where } Msg \text{ is bitwise equal to } b_1 \cdots b_m \text{ for } b_i \in \{0, 1\}.
\end{aligned}
$$

These functions are not used in the setup phase, however, they will be used to simulate Waters-signature in the signing oracle phase and to extract $\mathfrak{g}^{ab}$ from the adversary's output.

$\mathcal{B}$ chooses linearly independent random vectors $\overrightarrow{x}_1, \overrightarrow{x}_2$ and $\overrightarrow{x}_3 \xleftarrow{\$} \mathbb{Z}_p^3$, chooses random integers ζ_1, ζ_2 from $\mathbb{Z}_p$, computes

$$
\begin{array}{lll}
g_1 = \mathfrak{g}^{\overrightarrow{x}_1}, & g_2 = \mathfrak{g}^{\overrightarrow{x}_2}, & g_3 = \mathfrak{g}^{\overrightarrow{x}_3},\\
A_1 = (\mathfrak{g}^a)^{\overrightarrow{x}_1} = g_1^a, & A_2 = (\mathfrak{g}^a)^{\overrightarrow{x}_2} = g_2^a, & A_3 = (\mathfrak{g}^a)^{\overrightarrow{x}_3} = g_3^a,\\
B_1 = (\mathfrak{g}^b)^{\overrightarrow{x}_1} = g_1^b, & B_2 = (\mathfrak{g}^b)^{\overrightarrow{x}_2} = g_2^b, & B_3 = (\mathfrak{g}^b)^{\overrightarrow{x}_3} = g_3^b,
\end{array}
$$

and then generates CRS by defining $G = \mathbb{G}^3$, $G_t = \mathbb{G}_t^9$ and a map e according to the description in $\mathcal{G}_{SP}$, and by computing

$$
g = g_1 g_2 g_3,\ u' = (B_1^{p-k\ell+z'} g_1^{w'_1})(g_2^{w'_2})(g_3^{w'_3}),\ u_i = (B_1^{z_i} g_1^{w_{1i}})(g_2^{w_{2i}})(g_3^{w_{3i}}), v_i = g_1^{\bar{w}_{1i}} g_2^{\bar{w}_{2i}} A_3^{\bar{w}_{3i}},
$$

$$
h_1 = g_1^{\zeta_1},\ h_2 = g_2^{\zeta_2}.
$$

Then, $g, u', u_1, v_1, \cdots, u_m, v_m$ are uniformly distributed in G, and h_1 and h_2 are uniformly distributed in $\langle g_1 \rangle = G_1$ and $\langle g_2 \rangle = G_2$, respectively, so that the distribution of CRS is identical to the distribution of the real output of Setup algorithm.

We note that $\mathcal{B}$ knows $\overrightarrow{x}_1, \overrightarrow{x}_2$, and $\overrightarrow{x}_3$ so that he can construct $\pi_1, \pi_2, \pi_3, \pi_{t,1}, \pi_{t,2}$ and $\pi_{t,3}$.

KeyGen. Next, $\mathcal{B}$ generates the public key PK by choosing random integers $b', c' \xleftarrow{\$} \mathbb{Z}_p$ and computing $A = e(A_1 A_2 A_3, B_1 g_2^{b'} g_3^{c'}) = e(g^a, g_1^b g_2^{b'} g_3^{c'}) = e(g, g_1^{ab} A_2^{b'} A_3^{c'})$. Then, the secret key g' is $g_1^{ab} A_2^{b'} A_3^{c'}$, which is uniformly distributed in G and unknown to $\mathcal{B}$ since $\mathcal{B}$ does not know g_1^{ab}.

Sining Oracle. $\mathcal{B}$ receives $info = b_1 \cdots b_{m_0}$ (actually m_0 is depending on each $info$, however, for notational convenience, we use notation m_0 instead of m_{info}.) and a tuple $req = (c_{m_0+1}, d_{m_0+1}, \overrightarrow{\theta}_{m_0+1}, \cdots, c_m, d_m, \overrightarrow{\theta}_m)$ from $\mathcal{A}$. For each $i \in [m_0+1, m]$, $\mathcal{B}$ tests $e(c_i, d_i v_i^{-1}) \stackrel{?}{=} e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2})$ and $e(c_i u_i^{-1}, d_i) \stackrel{?}{=} e(h_1, \theta_{i,3}) e(h_2, \theta_{i,4})$. If these two equalities do not hold, then $\mathcal{B}$ aborts the protocol and outputs $\perp$. If req passes the test, for each $i \in [m_0+1, m]$ $c_i, d_i, \theta_{i,1}, \theta_{i,2}, \theta_{i,3}$, and $\theta_{i,4}$ can be written as the forms

$$c_i = (u_i)^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, \quad \theta_{i,1} = u_i^{b_i s_{i,1}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}, \quad \theta_{i,2} = u_i^{b_i s_{i,2}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}.$$

$$d_i = (v_i)^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}, \quad \theta_{i,3} = u_i^{(b_i - 1) s_{i,1}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}, \quad \theta_{i,4} = u_i^{(b_i - 1) s_{i,2}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}.$$

for some $b_i \in \{0,1\}$ and $t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2}, r_i, r'_i \in \mathbb{Z}_p$. We will show this in the lemma 5.

$\mathcal{B}$ first computes the Waters-signature (S_1, S_2), and then computes the signer's output $(K_1, K_2, K_{3,1}, K_{3,2})$. The simulation to compute the Waters-signature is similar to the proof of the original Waters' signature scheme [37]. Given c_i, $\mathcal{B}$ applies π_3 and obtain $\pi_3(u_i^{b_i})$. Since $\mathcal{B}$ can compute $\pi_3(u_i)$, $\mathcal{B}$ can obtain b_i (recall, $b_i \in \{0,1\}$) so that $\mathcal{B}$ has the whole message Msg' by applying similarly for all $i \in [m_0+1, m]$. From now, we use notation $Msg = b_1 \cdots b_m$ to denote $info \| Msg'$.

If $K(Msg) = 0$, $\mathcal{B}$ aborts and outputs $\perp$. Otherwise, $\mathcal{B}$ chooses a random integer $r \in \mathbb{Z}_p$ and $\mathcal{B}$ constructs (S_1, S_2) as

$$(S_1, S_2) = (A_1^{\frac{-J_1(Msg)}{F(Msg)}} A_2^{b' - \frac{J_2(Msg)}{F(Msg)}} A_3^{c' - \frac{J_3(Msg)}{F(Msg)}} (u' \prod_{i \in [1,m]} u_i^{b_i})^r, \quad g^{-r} (A_1 A_2 A_3)^{\frac{1}{F(Msg)}}).$$

This is equal to a Waters-signature on $Msg = b_1 \cdots b_m$ with the randomness $\tilde{r} = r - \frac{a}{F(Msg)}$. More precisely,

$$\begin{aligned}
S_1 &= A_1^{\frac{-J_1(Msg)}{F(Msg)}} A_2^{b' - \frac{J_2(Msg)}{F(Msg)}} A_3^{c' - \frac{J_3(Msg)}{F(Msg)}} (u' \textstyle\prod_{i \in [1,m]} u_i^{b_i})^r \\
&= A_1^{\frac{-J_1(Msg)}{F(Msg)}} (B_1^{F(Msg)} g_1^{J_1(Msg)})^r A_2^{b' - \frac{J_2(Msg)}{F(Msg)}} (g_2^{J_2(Msg)})^r A_3^{c' - \frac{J_3(Msg)}{F(Msg)}} (g_3^{J_3(Msg)})^r \\
&= g_1^{ab} (B_1^{F(Msg)} g_1^{J_1(Msg)})^{r - \frac{a}{F(Msg)}} A_2^{b'} (g_2^{J_2(Msg)})^{r - \frac{a}{F(Msg)}} A_3^{c'} (g_3^{J_3(Msg)})^{r - \frac{a}{F(Msg)}} \\
&= g_1^{ab} A_2^{b'} A_3^{c'} (B_1^{F(Msg)} g_1^{J_1(Msg)} g_2^{J_2(Msg)} g_3^{J_3(Msg)})^{r - \frac{a}{F(Msg)}} \\
&= g'(u' \textstyle\prod_{i \in [1,m]} u_i^{b_i})^{\tilde{r}},
\end{aligned}$$

$$\begin{aligned}
S_2 &= g^{-r} (A_1 A_2 A_3)^{\frac{1}{F(Msg)}} \\
&= g^{-r + \frac{a}{F(Msg)}} \\
&= g^{-\tilde{r}}.
\end{aligned}$$

Next, $\mathcal{B}$ constructs $K_1, K_2, K_{3,1}$ and $K_{3,2}$, by using S_1 and S_2. Since $K_2 = S_2$ and $K_{3,i} = h_i^{-\tilde{r}} = h_i^{-r} h_i^{\frac{a}{F(Msg)}} = h_i^{-r} A_i^{\frac{\zeta_i}{F(Msg)}}$, $\mathcal{B}$ can construct $K_2, K_{3,1}$, and $K_{3,2}$. The remaining part is to construct K_1, which is quite technical. We know that $K_1 = S_1 \cdot (\prod_{i \in [m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^{\tilde{r}}$. To construct K_1, $\mathcal{B}$ needs to compute $h_1^{t_{i,1}}, h_2^{t_{i,2}}, h_1^{a t_{i,1}}$ and $h_2^{a t_{i,2}}$ for $i \in [m_0+1, m]$. Since $\mathcal{B}$ knows b_i, he can compute $h_j^{t_{i,j}}$ by computing $\pi_j(c_i / u_i^{b_i})$. $\mathcal{B}$ computes $h_j^{a t_{i,j}}$ by the following procedures. First, $\mathcal{B}$ computes

$$\text{if } b_i = 0, \begin{cases} \pi_3(\theta_{i,1}) = \pi_3(v_i^{-t_{i,1}}) = A_3^{-\bar{w}_{3,i} t_{i,1}}, \\ \pi_3(\theta_{i,2}) = \pi_3(v_i^{-t_{i,2}}) = A_3^{-\bar{w}_{3,i} t_{i,2}}, \end{cases}$$

$$\text{if } b_i = 1, \begin{cases} \pi_3(\theta_{i,3}) = \pi_3(v_i^{t_{i,1}}) = A_3^{\bar{w}_{3,i} t_{i,1}}, \\ \pi_3(\theta_{i,4}) = \pi_3(v_i^{t_{i,2}}) = A_3^{\bar{w}_{3,i} t_{i,2}}. \end{cases}$$

Second, $\mathcal{B}$ computes $A_3^{t_{i,1}}$ and $A_3^{t_{i,2}}$. (Since $\mathcal{B}$ knows $\bar{w}_{3,i}$ and $\bar{w}_{3,i}$ is a non-zero with overwhelming probability, $\mathcal{B}$ can compute $A_3^{t_{i,j}}$.) Third, $\mathcal{B}$ computes $A_1^{t_{i,1}} = \mathcal{T}_{3,1}(A_3, A_3^{t_{i,1}}, A_1)$ by using the *translating* property in the

definition 9. More precisely, let M be a 3×3 matrix with having $\overrightarrow{x}_i$ as its i-th row, and $\overrightarrow{e}_i$ be the i-th canonical vector in $\mathbb{Z}_p^3$. Then, we can see the following two equalities.

$$(A_3^{t_{i,1}})^{M^{-1}} = (\mathfrak{g}^{at_{i,1}\overrightarrow{x}_3})^{M^{-1}} = (\mathfrak{g}^{at_{i,1}\overrightarrow{e}_3 M})^{M^{-1}} = \mathfrak{g}^{at_{i,1}\overrightarrow{e}_3} = (1, 1, \mathfrak{g}^{at_{i,1}}),$$
$$(\mathfrak{g}^{at_{i,1}}, 1, 1)^M = (\mathfrak{g}^{at_{i,1}\overrightarrow{e}_1})^M = \mathfrak{g}^{at_{i,1}\overrightarrow{x}_1} = A_1^{t_{i,1}}.$$

$\mathcal{B}$ can compute $A_1^{t_{i,j}}$ by using M and $A_3^{t_{i,j}}$ according to the above equalities. Fourth, $\mathcal{B}$ computes $h_1^{at_{i,1}}$ by computing $(A_1^{t_{i,1}})^{\zeta_1}$. Similarly, $\mathcal{B}$ computes $h_2^{at_{i,2}} = (\mathcal{T}_{3,2}(A_3, A_3^{t_{i,2}}, A_2))^{\zeta_2}$. Next, $\mathcal{B}$ compute K_1 by using $h_1^{t_{i,1}}, h_2^{t_{i,2}}, h_1^{at_{i,1}}$ and $h_2^{at_{i,2}}$ for $i \in [m_0+1, m]$ as follows:

$$\begin{aligned} K_1 &= S_1 \cdot (\textstyle\prod_{i\in[m_0+1,m]} (h_1^{t_{i,1}} h_2^{t_{i,2}})^r (h_1^{at_{i,1}} h_2^{at_{i,2}})^{-\frac{1}{F(Msg)}}) \\ &= S_1 \cdot (\textstyle\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^{r-\frac{a}{F(Msg)}} \\ &= S_1 \cdot (\textstyle\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^{\tilde{r}}. \end{aligned}$$

Finally, $\mathcal{B}$ sends $(K_1, K_2, K_{3,1}, K_{3,2})$ and outputs *success* and $info$.

Output. At the end of the interaction, for some $info$ $\mathcal{A}$ outputs q' pairs of a message and a signature such that all q' messages are distinct and all tuples of $info$, message and signature pass the Verify algorithm. If q' is strictly larger than q_{info}, which is the number of signing queries issued by $\mathcal{A}$ for $info$, by the pigeonhole principle there would exist a pair of Msg' and a signature (S_1^*, S_2^*) such that $\mathcal{A}$ did not obtain them from $\mathcal{B}$. We use Msg^* to denote $info \| Msg'$. For such $Msg^* = b_1^* \cdots b_m^*$, if $z' + \sum_{i\in[1,m]} b_i^* z_i \neq k\ell$, then $\mathcal{B}$ outputs $\perp$. Otherwise, we have $F(Msg^*) \equiv 0 \pmod p$. $\mathcal{B}$ computes $C = (\pi_1(S_1^* \cdot (S_2^*)^{J_1(Msg)}))^{M^{-1}}$. Let $C = (\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3)$. Then, $\mathcal{B}$ outputs $\mathfrak{g}_1$. When $\mathcal{A}$ fails to outputs $q' > q_{info}$ distinct messages and their signature for the same common information $info$, $\mathcal{B}$ outputs $\perp$.

Analysis. We need to argue two things. One is that the simulated transcript by $\mathcal{B}$ is indistinguishable from the real transcript in the view of $\mathcal{A}$. The another one is that the success probability of $\mathcal{B}$ to output $\mathfrak{g}^{ab}$ is more than $\frac{\epsilon}{8(m+1)q}$.

In the view of $\mathcal{A}$, the whole simulated transcript is identical to the transcript in the real game. In Setup phase, each parameter is uniformly distributed in the suitable group. In the KeyGen phase, $g' = g_1^{ab} A_2^{b'} A_3^{c'}$ is uniformly distributed in G because of b, b' and c'. Therefore, it's distribution is identical to that of the real game. In Sining Oracle phase, $\mathcal{B}$'s output is

$$(g'(u' \prod_{i\in[1,m]} u_i^{b_i})^{\tilde{r}} (\prod_{i\in[m_0+1,m]} h_1^{t_{i,1}} h_2^{t_{i,2}})^{\tilde{r}}, \quad g^{-\tilde{r}}, \quad h_1^{-\tilde{r}}, \quad h_2^{-\tilde{r}}),$$

whose distribution is identical to that of the real output of the signer. Therefore, the overall distribution is identical to that of the real game defined in Appendix A.

Now, we argue that $\mathcal{B}$'s success probability is non-negligible whenever $\mathcal{A}$'s success probability to break the one-more unforgeability is non-negligible. First, we argue that $\mathcal{B}$ outputs $\mathfrak{g}^{ab}$ if $\mathcal{B}$ does not abort and $\mathcal{A}$ successfully breaks the one-more unforgeability. Second, we argue the overall success probability of $\mathcal{B}$.

In Output phase, $\mathcal{B}$ receives (S_1^*, S_2^*) such that $\mathsf{Verify}(CRS, PK, Msg^*, (S_1^*, S_2^*)) = accept$ and $F(Msg^*) \equiv 0 \pmod p$. $\mathsf{Verify}(CRS, PK, Msg^*, (S_1^*, S_2^*)) = accept$ implies an equality

$$e(S_1^*, g) e(S_2^*, u' \prod_{i\in[1,m]} u_i^{b_i^*}) = A,$$

where $Msg^* = b_1^* \cdots b_m^*$. Apply a projection $\pi_{t,1}$ in the both side of the above equality, and then use the commutative property of projections and e, so we obtain that[8]

$$\begin{aligned} e(\pi_1(S_1^*), \pi_1(g)) e(\pi_1(S_2^*), \pi_1(u' \textstyle\prod_{i\in[1,m]} u_i^{b_i})) &= \pi_{t,1}(A) \\ &= e(\pi_1(g), \pi_1(g_1^{ab} A_2^{b'} A_3^{c'})) \\ &= e(\pi_1(g), g_1^{ab}). \end{aligned}$$

[8] That is, we utilize the *projecting* property.

Let $\pi_1(S_2) = g_1^{r^*}$ for some unknown $r^* \in \mathbb{Z}_p$. (Since G_1 is a cyclic group of order p, we can always write $\pi_1(S_2) = g_1^{r^*}$ for some $r^* \in \mathbb{Z}_p$.) Then,

$$e(\pi_1(S_1^*), g_1)e(\pi_1(u' \prod_{i\in[1,m]} u_i^{b_i}), g_1^{r^*}) = e(g_1^{ab}, g_1)$$

By little changing the above equality using the bilinear property of e, we obtain

$$e(\pi_1(S_1^*)(\pi_1(u' \textstyle\prod_{i\in[1,m]} u_i^{b_i}))^{r^*} g_1^{-ab}, g_1) = 1_t.$$

Since e is a non-degenerate bilinear map, $\pi_1(S_1^*)(\pi_1(u' \prod_{i\in[1,m]} u_i^{b_i}))^{r^*} g_1^{-ab} = 1$ so that

$$\pi_1(S_1^*)(\pi_1(u' \prod_{i\in[1,m]} u_i^{b_i}))^{r^*} = g_1^{ab}.$$

In the case that $F(Msg^*) \equiv 0 \ (\mod p)$,

$$\begin{aligned}(\pi_1(u' \textstyle\prod_{i\in[1,m]} u_i^{b_i}))^{r^*} &= (g_1^{J_1(Msg)})^{r^*} \\ &= \pi_1((S_2^*)^{J_1(Msg)}).\end{aligned}$$

Therefore, from the above two equalities we obtain $\pi_1(S_1^*)\pi_1((S_2^*)^{J_1(Msg)}) = g_1^{ab}$. Further,

$$(g_1^{ab})^{M^{-1}} = (\mathfrak{g}^{ab\overrightarrow{x}_1})^{M^{-1}} = (\mathfrak{g}^{ab\overrightarrow{e}_1 M})^{M^{-1}} = (\mathfrak{g}^{ab\overrightarrow{e}_1}) = (\mathfrak{g}^{ab}, 1, 1),$$

so that $\mathcal{B}$ outputs $\mathfrak{g}^{ab}$.

Next, we argue about the success probability of $\mathcal{B}$. Since $\mathcal{B}$ sometimes aborts, it is not easy to calculate the success probability of $\mathcal{B}$. The event that $\mathcal{B}$ does not abort, however, is only related to the case that $K(Msg_i) = 1$ for all $i \in [1, q]$ and $z' + \sum_{i\in[1,m]} b_i^* z_i = k\ell$, where $Msg_1 \cdots, Msg_q$ are messages used in the signature sining query, that is, $Msg_i = info_i \| Msg_i'$ and $\mathcal{A}$ sent the commitment to Msg_i' to $\mathcal{B}$ during the signing query phase, and $Msg^* = b_1^* \cdots b_m^*$. This situation is identical to the simulation in the proof of the existentially unforgeability of Waters-signature [37]. Therefore, we follow Waters' approach to analyze the success probability of the simulation. First we describe a second simulator, which is easy to be analyzed and has the same success probability of simulator as that of the first simulation. We define a function $\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*)$ by

$$\begin{cases} 0 & \text{if } K(Msg_i) = 1 \text{ for } \forall i \in [1, q], \text{ and } z' + \sum_{i\in[1,m]} b_i^* z_i = k\ell, \\ 1 & \text{otherwise,} \end{cases}$$

where $Msg^* = b_1^* \cdots b_m^*$. We can easily check that $\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0$ if and only if the simulator in the first simulation is not abort.

Now, we describe the second simulation. In the second simulation the simulator, we assume that $\mathcal{B}_2$ receives a DDH-tuple $(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, \mathfrak{g}^{ab})$ before starting simulation. Then, $\mathcal{B}_2$ behaves as the real challenger in the one-more unforgeability game. That is, $\mathcal{B}_2$ generates CRS in the setup phase, a signing key pair PK, SK in the KeyGen phase, answers signature signing queries by using SK. Let $info_i$ and Msg_i' be the common input string and the message used in Signing Oracle phase by $\mathcal{A}$. We simply denote $info_i$ and Msg_i' as $Msg_i = info_i \| Msg_i'$. Let E_{succA} be the event that $\mathcal{A}$ wins in the one-more unforgeability game, that is, $\mathcal{A}$ outputs q' message and signature pairs passing Verify algorithm for the same $info$ such that all messages are mutually distinct and $q' > q_{info}$, where q_{info} is the number of issuing signing queries for $info$ by $\mathcal{A}$. At the end of interactions, the event E_{succA} occurs with ϵ probability. If the event E_{succA} occurs, then there exists at least one pair $(Msg^*, (S_1^*, S_2^*))$ such that $\mathcal{B}_2$ have not signed on the message $Msg^* = b_1^* \cdots b_m^*$ $(= info \| Msg$, for some $Msg)$. At this point $\mathcal{B}_2$ compute $\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*)$ and if $\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 1$, then $\mathcal{B}_2$ aborts and outputs $\perp$. Otherwise, $\mathcal{B}_2$ outputs $\mathfrak{g}^{ab}$. Further, if the event E_{succA} does not occur, then $\mathcal{B}_2$ outputs $\perp$.

Claim 1. $\Pr[(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, T)$ is a DDH-tuple. : $\mathcal{B}_i \xrightarrow{\$} T]$ is identical regardless i, where $\mathcal{B}_i$ is a simulator in the i-th simulation.

Claim 2. ([37], Claim 2.) $\Pr_{(z',z_i)}[\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0] \geq \frac{1}{8(m+1)q}$ for all $(Msg_1, \cdots, Msg_q, Msg^*)$ so that the probability of the simulation not aborting is at least $\frac{1}{8(m+1)q}$.

From the above two claims, we conclude that the success probability of $\mathcal{B}_1$ is more than $\frac{\epsilon}{8(m+1)q}$ as follows:

$$\begin{aligned}
&\Pr[(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, T) \text{ is a DDH-tuple. : } \mathcal{B}_1 \xrightarrow{\$} T] \\
&= \Pr[(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, T) \text{ is a DDH-tuple. : } \mathcal{B}_2 \xrightarrow{\$} T] \\
&\geq \Pr[E_{succA} \bigwedge \Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0] \\
&= \Pr[E_{succA}] \cdot \Pr[\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0 | E_{succA}] \\
&\geq \epsilon \cdot \tfrac{1}{8(m+1)q}.
\end{aligned}$$

The last inequalities come from the fact that $\Pr[\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0] \geq \frac{1}{8(m+1)q}$ for all $(Msg_1, \cdots, Msg_q, Msg^*)$ so that $\Pr[\Gamma(z', z_1, \cdots, z_m, Msg_1, \cdots, Msg_q, Msg^*) = 0 | E_{succA}] \geq \frac{1}{8(m+1)q}$. ($z', z_1, \cdots, z_m$ are independent from E_{succA} since $(z', z_1, \cdots, z_m)$ is completely hided from the view of $\mathcal{A}$.) □

Lemma 5 *For $c_i, d_i, \theta_{i,1}, \theta_{i,2}, \theta_{i,3}$ and $\theta_{i,4} \in G$, if $e(c_i, d_i v_i^{-1}) = e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2})$ and $e(c_i u_i^{-1}, d_i) = e(h_1, \theta_{i,3}) e(h_2, \theta_{i,4})$, then $c_i, d_i, \theta_{i,1}, \theta_{i,2}, \theta_{i,3}$ and $\theta_{i,4}$ can be uniquely written as the forms*

$$c_i = (u_i)^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, \quad \theta_{i,1} = u_i^{b_i s_{i,1}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}, \quad \theta_{i,2} = u_i^{b_i s_{i,2}} (v_i^{b_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}.$$

$$d_i = (v_i)^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}, \quad \theta_{i,3} = u_i^{(b_i - 1) s_{i,1}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}, \quad \theta_{i,4} = u_i^{(b_i - 1) s_{i,2}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}.$$

for some $b_i \in \{0, 1\}$ and $t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2}, r_i, r'_i \in \mathbb{Z}_p$.

Proof. We can always write c_i and d_i by $u_i^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}$ and $v_i^{b'_i} h_1^{s_{i,1}} h_2^{s_{i,2}}$, respectively, for some $b_i, b'_i, t_{i,1}, t_{i,2}, s_{i,1}, s_{i,2} \in \mathbb{Z}_p$ since $\langle u_i, h_1, h_2 \rangle = \langle v_i, h_1, h_2 \rangle = G$. From the equality $e(c_i, d_i v_i^{-1}) = e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2})$, we obtain

$$\begin{aligned}
&e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2}) \\
&= e(c_i, d_i v_i^{-1}) \\
&= e(u_i^{b_i} h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= e(u_i^{b_i}, v_i^{b'_i - 1}) e(u_i^{b_i}, h_1^{s_{i,1}} h_2^{s_{i,2}}) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= e(u_i^{b_i}, v_i^{b'_i - 1}) e((u_i^{b_i})^{s_{i,1}}, h_1) e((u_i^{b_i})^{s_{i,2}}, h_2) e(h_1^{t_{i,1}} h_2^{t_{i,2}}, v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}}) \\
&= e(u_i^{b_i}, v_i^{b'_i - 1}) e((u_i^{b_i})^{s_{i,1}}, h_1) e((u_i^{b_i})^{s_{i,2}}, h_2) e(h_1, (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \\
&= e(u_i^{b_i}, v_i^{b'_i - 1}) e(h_1, u_i^{b_i s_{i,1}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}}) e(h_2, u^{b_i s_{i,2}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}).
\end{aligned}$$

Let $G_{t,i,j}$ be a prime order cyclic group $e(G_i, G_j)$. We already know that the image of e is equal to $G_{t,1,1} \oplus G_{t,1,2} \oplus G_{t,1,3} \oplus G_{t,2,2} \oplus G_{t,2,3} \oplus G_{t,3,3}$ from the lemma 1. We first show that $b_i = 0$ or $b'_i = 1$. Since $e(h_1, \theta_{i,1}) e(h_2, \theta_{i,2}) \in G_{t,1,1} \oplus G_{t,1,2} \oplus G_{t,1,3} \oplus G_{t,2,2} \oplus G_{t,2,3}$, $e(u_i^{b_i}, v_i^{b'_i - 1})$ should be the identity so that $b_i = 0$ or $b'_i = 1$. (If $e(u_i^{b_i}, v_i^{b'_i - 1}) \neq 1_t$, $e(u_i^{b_i}, v_i^{b'_i - 1})$'s decomposition has a non-identity element in $G_{t,3,3}$ so that the above equality cannot hold.)

Therefore, after setting $e(u_i^{b_i}, v_i^{b'_i - 1}) = 1$ from the above equality we obtain an equality

$$e(h_1, \theta_{i,1} u_i^{-b_i s_{i,1}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{-t_{i,1}}) = e(h_2, \theta_{i,2}^{-1} u^{b_i s_{i,2}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}).$$

The left side of the equality is contained in $G_{t,1,1} \oplus G_{t,1,2} \oplus G_{t,1,3}$ and the right side of the equality is contained in $G_{t,1,2} \oplus G_{t,2,2} \oplus G_{t,2,3}$. Since $G_{t,1,1}, G_{t,1,2}, G_{t,1,3}, G_{t,2,2}$, and $G_{t,2,3}$ are mutually disjoint subgroups

(except the identity), $e(h_1, \theta_{i,1} u_i^{-b_i s_{i,1}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{-t_{i,1}}) = e(h_2, \theta_{i,2}^{-1} u^{b_i s_{i,2}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}}) \in G_{t,1,2}$. It means that $\theta_{i,1} u_i^{-b_i s_{i,1}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{-t_{i,1}} = h_2^{r_i}$ and $\theta_{i,2}^{-1} u^{b_i s_{i,2}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} = h_1^{r_i}$ for some $r_i \in \mathbb{Z}_p$, and hence $\theta_{i,1} = u_i^{b_i s_{i,1}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r_i}$ and $\theta_{i,2} = u^{b_i s_{i,2}} (v_i^{b'_i - 1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r_i}$.

Analogously, from the equality $e(c_i u_i^{-1}, d_i) = e(h_1, \theta_{i,3}) e(h_2, \theta_{i,4})$, we can show that (1) $b_i = 1$ or $b'_i = 0$ and (2) $\theta_{i,3} = u_i^{(b_i - 1) s_{i,1}} (v_i^{b'_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,1}} h_2^{r'_i}$ and $\theta_{i,4} = u^{(b_i - 1) s_{i,2}} (v_i^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_{i,2}} h_1^{-r'_i}$ for some $r'_i \in \mathbb{Z}_p$. Since ($b_i = 0$ or $b'_i = 1$) and ($b_i = 1$ or $b'_i = 0$), we can be convinced that $b_i = b'_i \in \{0, 1\}$. Further, $\theta_{i,1}$, $\theta_{i,2}$, $\theta_{i,3}$, and $\theta_{i,4}$ can be written as the desired form.

Moreover, we can easily check that $G = \langle u_i \rangle \oplus \langle h_1 \rangle \oplus \langle h_2 \rangle = \langle v_i \rangle \oplus \langle h_1 \rangle \oplus \langle h_2 \rangle$. Therefore, it implies that the uniqueness of b_i, $t_{i,1}$, $t_{i,2}$, $s_{i,1}$, $s_{i,2}$, r_i and r'_i. □

D Projections, Membership Tests, and Subgroup Decision Assumption

We show how to construct natural projections π_i, $\bar{\pi}_i$ and $\pi_{t,i}$. Furthermore, we will show how to check group membership of G and H. Intuitively, it seems like that it is infeasible to determine the group membership of G (H, respectively) for given $g \in \mathbb{G}^{n^2}$ ($h \in \mathbb{H}^{n^2}$, respectively) since it is a kind of the subgroup decision problem of $\mathbb{G}^{n^2}$ ($\mathbb{H}^{n^2}$, respectively). We propose a novel technique to manage this problem about group membership. Finally, we will show that our bilinear group generator satisfies $(2, 1)$-*subgroup decision assumption* in generic group model.

Constructions of Projections. Next, we describe how to construct the natural projections $\pi_i : G = \oplus_{i \in [1,n]} G_i \to G_i$ for $i \in [1, n]$. Let a n^2-by-n^2 matrix M has Ψ_j as its the j-th row. Then, M is invertible. Let U_i be a n^2-by-n^2 matrix with 1 in the (i, i) entry and zeroes elsewhere. First, we choose additional $\{\Psi_j\}_{j \in [n+1, n^2]}$ and $\{\Phi_j\}_{j \in [n+1, n^2]}$ such that $\{\Psi_j\}_{j \in [n^2]}$ and $\{\Phi_j\}_{j \in [n^2]}$ are sets of linearly independent vectors. The natural projection π_i is defined by

$$\pi_i(g) = g^{M^{-1} U_i M}.$$

Since $g = \prod_{i \in [1,n]} (\mathfrak{g}^{\Psi_i})^{\alpha_i} = \mathfrak{g}^{\sum_{i \in [1,n]} \alpha_i \Psi_i} = \mathfrak{g}^{\sum_{i \in [1,n]} \alpha_i \overrightarrow{e}_i M}$ for some α_i, where $\overrightarrow{e}_i$ is the i-th canonical vector in $(\mathbb{Z}_p)^{n^2}$,

$$g^{M^{-1} U_i M} = \mathfrak{g}^{\sum_{i \in [1,n]} \alpha_i \overrightarrow{e}_i U_i M} = \mathfrak{g}^{\alpha_i \overrightarrow{e}_i M} = \mathfrak{g}^{\alpha_i \Psi_i}$$

is the desired output of the natural projection map.

We can construct the natural projection $\bar{\pi}_i : H = \oplus_{i \in [1,n]} H_i \to H_i$ analogously. Since Φ_j's are linearly independent, we can define an invertible n^2-by-n^2 matrix M' having Φ_j as its j-th row, and then we define $\bar{\pi}_j$ by $h \mapsto h^{M'^{-1} U_j M'}$.

The natural projection in G_t, $\pi_{t,i} : G_t = \oplus_{i \in [1,n]} e(G_i, H_i) \to e(G_i, H_i)$ can be also constructed by the similar way. Since $e(G_i, H_i) = \langle e(\mathfrak{g}, \mathfrak{h})^{\Psi_i \circ \Phi_i} \rangle = \langle (\mathfrak{g}, \mathfrak{h})^{(d_{i1}, \cdots, d_{in})} \rangle$ and $D = (d_{i\ell})$ is invertible n-by-n matrix, $\pi_{t,i}(g_t)$ can be computed by $g_t^{D^{-1} U_i D}$ for $g_t \in G_t$.

Group Membership Test. Anyone who knows the group description of G and σ can determine whether given g is contained in G as follows:

1. If $g \in \mathbb{G}^{n^2}$ (by using the group membership test of $\mathcal{G}_1$), then go to the next step. Otherwise, output 0.
2. Let $g = \mathfrak{g}^{\Gamma}$, where $\Gamma = (\alpha_{11}, \cdots, \alpha_{nn}) \in \mathbb{Z}_p^{n^2}$. For $\forall i \in [1, n^2 - n]$, test $\prod_{j,\ell \in [1,n]} \hat{e}(\mathfrak{g}^{\alpha_{j\ell}}, \mathfrak{h}^{\hat{\psi}_{ij\ell}}) = \hat{e}(\mathfrak{g}, \mathfrak{h})^{\Gamma \cdot \hat{\Psi}_i} \stackrel{?}{=} 1_{\mathbb{G}_t}$, where $\hat{\Psi}_i = (\hat{\psi}_{i11}, \cdots, \hat{\psi}_{inn})$ and $\mathfrak{h}^{\hat{\Psi}_i}$ is contained in σ. If yes, output 1. Otherwise, output 0.

It is not hard to show that given g, the above Group Membership Test (GMT) algorithm outputs 1 if and only if $g \in G$. (For $g \in \mathbb{G}^{n^2}$, $g = \mathfrak{g}^{\Gamma} \in G$ if and only if $\Gamma \in \langle \Psi_1, \cdots, \Psi_n \rangle$ if and only if $\Gamma \cdot \hat{\Psi}_i = 0$ for

all $i \in [1,n]$, where $\{\hat{\Psi}_i\}_{i\in[1,n^2-n]}$ is a basis of $\langle \Psi_1, \cdots, \Psi_n\rangle^{\perp}$.) To test a g's membership in G, we need to compute $(n^2-n)n^2$ bilinear map $\hat{e}$ and $(n^2-n)(n^2-1)$ multiplication in $\mathbb{G}_t$ if we ignore the cost for computing the step 1 since it is respectively smaller than the cost for computing the step 2.

The above GMT is a perfect algorithm in the sense that it outputs without any errors. If we allow negligible errors, then we can improve the efficiency of GMT by using some batch verification technique. Basing on the k-linear assumption, we can further improve the efficiency of GMT. We relegate these improvements and the batch verification technique to Appendix E.

Subgroup Decision Assumption. The proposed bilinear group generator satisfies the *subgroup decision assumption* in generic group model when $n = 2$ and $k = 1$. We propose a new assumption which guarantees the *subgroup decision assumption*. We provide an evidence why we believe this new assumption is secure by showing that it holds in the generic group model.

Definition 12 *Let $\mathcal{G}_1$ be a bilinear group generator. Define distribution $\mathcal{D}_b$ by*

$$\mathcal{G}_1(\lambda) \xrightarrow{\$} (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e}), \mathfrak{g} \xleftarrow{\$} \mathbb{G}, \mathfrak{h} \xleftarrow{\$} \mathbb{H},$$
$$\mathfrak{g}^{x_1}, \mathfrak{g}^{x_2}, \mathfrak{g}^{d_1 d_2 (x_1 x_4 - x_2 x_3)}, \mathfrak{g}^{x_1 y}, T_b \in \mathbb{G}^5,$$
$$M_{\mathfrak{g}} = \begin{pmatrix} \mathfrak{g}^{d_2 x_1 x_4 - d_1 x_2 x_3} & \mathfrak{g}^{(d_2-d_1)x_2 x_4} \\ \mathfrak{g}^{(d_1-d_2)x_1 x_3} & \mathfrak{g}^{d_1 x_1 x_4 - d_2 x_2 x_3} \end{pmatrix} \in Mat_2(\mathbb{G}),$$
$$\mathfrak{h}^{x_3}, \mathfrak{h}^{x_4}, \mathfrak{h}^{d_1 x_3}, \mathfrak{h}^{d_1 x_4}, \mathfrak{h}^{x_1 x_4 - x_2 x_3} \in \mathbb{H}^5,$$
$$M_{\mathfrak{h}} = \begin{pmatrix} \mathfrak{h}^{d_1 x_1 x_4 - d_2 x_2 x_3} & \mathfrak{h}^{(d_2-d_1)x_1 x_3} \\ \mathfrak{h}^{(d_1-d_2)x_2 x_4} & \mathfrak{h}^{d_2 x_1 x_4 - d_1 x_2 x_3} \end{pmatrix} \in Mat_2(\mathbb{H}),$$

where $x_1, x_2, x_3, x_4, y, d_1, d_2 \xleftarrow{\$} \mathbb{Z}_p$, $T_0 = \mathfrak{g}^{x_2 y}$, $T_1 \xleftarrow{\$} \mathbb{G}$, and $b \in \{0,1\}$. Define the advantage an algorithm $\mathcal{A}$, denoted by $Adv^1_{\mathcal{A},\mathcal{G}_1}$, to distinguish $\mathcal{D}_0$ and $\mathcal{D}_1$ by

$$Adv^1_{\mathcal{A},\mathcal{G}_1} = \Big| \Pr[\mathcal{A}(\mathcal{D}_0) \to 1] - \Pr[\mathcal{A}(\mathcal{D}_1) \to 1] \Big|.$$

We say that $\mathcal{G}_1$ satisfies the assumption 1 in $\mathbb{G}$ if for any probabilistic polynomial time algorithm $\mathcal{A}$, $Adv^1_{\mathcal{A},\mathcal{G}_1}$ is a negligible function of λ.

We prove that $\mathcal{G}_1$ satisfies the assumption 1 in the generic group model by applying the Boneh-Boyen-Goh master theorem [9], more precisely, its generalized version by Freeman [18]. Freeman gave the general definition of the independency and the master theorem using his definition of the independency as follows:

Definition 13 *[18, Definition D.1] Let $P = (u_1, \cdots, u_r), Q = (v_1, \cdots, v_s), R = (w_1, \cdots, w_t), S = (\chi_1, \cdots, \chi_m)$ be tuples of polynomials in $\mathbb{Z}_p[X_1, \cdots, X_n]$. Let f be a polynomial in $\mathbb{Z}_p[X_1, \cdots, X_n]$. We say that $f \cdot S$ is* dependent on (P, Q, R) *if there exist integers $a_{i,j}$ for $1 \in [1,r]$ and $j \in [1,s]$, integers b_k for $k \in [1,t]$, and integers c_ℓ with $\ell \in [1,m]$, such that*

$$\sum_{i\in[1,r]} \sum_{j\in[1,s]} a_{i,j} u_i v_j + \sum_{k\in[1,t]} b_k w_k + \sum_{\ell[1,m]} c_\ell \chi_\ell Y$$

is non-zero in $\mathbb{Z}_p[X_1, \cdots, X_n, Y]$ but become zero when we set $Y = f$.

We say that $f \cdot S$ is independent of (P, Q, R) if $f \cdot S$ is not dependent on (P, Q, R).

Definition 14 *[18, Definition D.2] Let $\mathcal{G}_1$ be a prime-order bilinear group generator, and let P, Q, R, f be as in Definition 13. Define the following distribution:*

$$GD = (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_t, e) \xleftarrow{\$} \mathcal{G}_1(\lambda), \mathfrak{g} \xleftarrow{\$} \mathbb{G}, \mathfrak{h} \xleftarrow{\$} \mathbb{H}, \mathfrak{g}_t = \hat{e}(\mathfrak{g}, \mathfrak{h}), \overrightarrow{x} \xleftarrow{\$} \mathbb{Z}_p^{\ell},$$

$$Z = (\mathfrak{g}^{u_1(\vec{x})}, \cdots, \mathfrak{g}^{u_r(\vec{x})}, \mathfrak{h}^{v_1(\vec{x})}, \cdots, \mathfrak{h}^{v_s(\vec{x})}, \mathfrak{g}_t^{w_1(\vec{x})}, \cdots, \mathfrak{g}_t^{w_t(\vec{x})}),$$
$$T_0 = \mathfrak{g}_1^{f(\vec{x})}, \quad T_1 \xleftarrow{\$} \mathbb{G}.$$

We define the advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0,1\}$ in solving the (P,Q,R,f)-decision Diffie-Hellman problem in $\mathbb{G}$ *to be*

$$(P,Q,R,f) - \text{DDH-Adv}[\mathcal{A}, \mathcal{G}_1] = \Big| \Pr[\mathcal{A}(GD, Z, T_0) \to 1] - \Pr[\mathcal{A}(GD, Z, T_1) \to 1] \Big|.$$

Theorem 7 *[18, Theorem D.3] Let (P,Q,R) be as in Definition 13, let $f \in \mathbb{Z}_p[X_1, \cdots, X_n]$, and let p be a prime. Let $d = 2 \cdot max\{\deg \alpha : \alpha \in P \cup Q \cup R \cup \{f\}\}$. If $f \cdot Q$ is independent of (P,Q,R), then any algorithm that solves the* (P,Q,R,f)-decision Diffie-Hellman problem in $\mathbb{G}$ *with advantage* $1/2$ *in a generic bilinear group of order p must take time at least $\Omega(\sqrt{p/d} - n)$, asymptotically as $p \to \infty$.*

By applying the above definitions and the theorem to the assumption 1, we obtain the following theorem.

Theorem 8 *Assumption* 1 *holds in the generic group model.*

Proof. We define sets of polynomials in $\mathbb{Z}_p[x_1, x_2, x_3, x_4, d_1, d_2, y]$ as follows:

$$P = \{1, x_1, x_2, d_1d_2(x_1x_4 - x_2x_3), x_1y, d_2x_1x_4 - d_1x_2x_3, (d_2 - d_1)x_2x_4, (d_1 - d_2)x_1x_3, d_1x_1x_4 - d_2x_2x_3\},$$
$$Q = \{1, x_3, x_4, d_1x_3, d_1x_4, x_1x_4 - x_2x_3, d_1x_1x_4 - d_2x_2x_3, (d_2 - d_1)x_1x_3, (d_1 - d_2)x_2x_4, d_2x_1x_4 - d_1x_2x_3\},$$
$$R = \{1\} \text{ and } f = \{x_2y\}.$$

Then, the assumption 1 is equivalent to a (P,Q,R,f)-*decision Diffie-Hellman problem in* $\mathbb{G}$. We will show that $f \cdot Q$ is independent of (P,Q,R). By the theorem 7, this is sufficient to show that $\mathcal{G}_1$ satisfies (P,Q,R,f)-*decision Diffie-Hellman problem in* $\mathbb{G}$. To show the independency, first we observe the variable y. The variable y is used only for x_1y in P and for x_2y in f. We will show that $f \cdot Q$ is independent of (P', Q, R) where $P' = \{x_1y\}$. This is sufficient to show that $f \cdot Q$ is independent of (P,Q,R). We will check this independency step-by-step. First, we will compute all possible products between polynomials in $\{x_1y, x_2y\}$ and in Q. For a set S of polynomials, we say S is independent if there is no non-trivial sum of polynomials in S to be equal to zero. Second, we will show that a set of the resulting polynomials (of the first step) is independent.

A set S of all possible products between polynomials in $\{x_1y, x_2y\}$ and in Q are as follows:

$$S = \{x_1y,\ x_1x_3y,\ x_1x_4y,\ d_1x_1x_3y,\ d_1x_1x_4y,\ (x_1^2x_4 - x_1x_2x_3)y, (d_1x_1^2x_4 - d_2x_1x_2x_3)y,$$
$$(d_2 - d_1)x_1^2x_3y,\ (d_1 - d_2)x_1x_2x_4y,\ (d_2x_1^2x_4 - d_1x_1x_2x_3)y,$$
$$x_2y,\ x_2x_3y,\ x_2x_4y,\ d_1x_2x_3y,\ d_1x_2x_4y,\ (x_1x_2x_4 - x_2^2x_3)y,\ (d_1x_1x_2x_4 - d_2x_2^2x_3)y,$$
$$(d_2 - d_1)x_1x_2x_3y,\ (d_1 - d_2)x_2^2x_4y,\ (d_2x_1x_2x_4 - d_1x_2^2x_3)y\}.$$

We separate the above set according to degrees of polynomials.

$$S_2 = \{x_1y,\ x_2y\}$$
$$S_3 = \{x_1x_3y,\ x_1x_4y,\ x_2x_3y,\ x_2x_4y\}$$
$$S_4 = \{d_1x_1x_3y,\ d_1x_1x_4y,\ (x_1^2x_4 - x_1x_2x_3)y,\ d_1x_2x_3y,\ d_1x_2x_4y,\ (x_1x_2x_4 - x_2^2x_3)y\}$$
$$S_5 = \{(d_1x_1^2x_4 - d_2x_1x_2x_3)y,\ (d_2 - d_1)x_1^2x_3y,\ (d_1 - d_2)x_1x_2x_4y,\ (d_2x_1^2x_4 - d_1x_1x_2x_3)y,$$
$$(d_1x_1x_2x_4 - d_2x_2^2x_3)y,\ (d_2 - d_1)x_1x_2x_3y,\ (d_1 - d_2)x_2^2x_4y,\ (d_2x_1x_2x_4 - d_1x_2^2x_3)y\}.$$

Then, $S = \cup_{i \in [2,5]} S_i$. It is easy to show that each S_i is independent, so we omit details. Each polynomial in S_i consists of only sum of monomials with degree i so that any summation of polynomials chosen from different S_i cannot be equal to zero. Therefore, the independency of S_i implies the independency of S so that we complete the proof for the independency of $f \cdot Q$ of (P,Q,R). □

Theorem 9 *$\mathcal{G}$ satisfies* $(2,1)$-subgroup decision assumption *if $\mathcal{G}_1$ satisfies the assumption* 1.

Proof. We define some notation useful to simplify proof. For matrices, $M = (m_{ij})$, $N = (n_{ij})$ and $L \in Mat_2(\mathbb{Z}_p)$, $\mathfrak{g}^N := (\mathfrak{g}^{n_{ij}})$, ${}^M(\mathfrak{g}^N) := (\prod_{k\in[1,2]}(\mathfrak{g}^{n_{kj}})^{m_{ik}}) = \mathfrak{g}^{MN}$, and $(\mathfrak{g}^N)^L := (\prod_{k\in[1,2]}(\mathfrak{g}^{n_{ik}})^{m_{kj}}) = \mathfrak{g}^{NL}$. Note that we can extend this notation for the case when MNL is well-defined. Given an algorithm $\mathcal{A}$ breaking $(2,1)$-*subgroup decision assumption* and $\mathcal{D}_b$, we describe an algorithm $\mathcal{B}$ determining b with same advantage as $\mathcal{A}$'s. First, $\mathcal{B}$ will compute generators for G, H, G_1 and H_1, and the additional information for group membership check σ. That is, $\mathcal{B}$ will compute

$$\text{generators in } G \text{ and } H : \mathfrak{g}^{A'X_1}, \mathfrak{g}^{A'X_2}, \mathfrak{h}^{B'Y_1}, \mathfrak{h}^{B'Y_2},$$

$$G_1 = \langle(\mathfrak{g}^{a'\overrightarrow{x}_1}, \mathfrak{g}^{a'\overrightarrow{x}_2})\rangle, H_1 = \langle(\mathfrak{h}^{b'\overrightarrow{y}_1}, \mathfrak{h}^{b'\overrightarrow{y}_2})\rangle,$$

$$\sigma = \{(\mathfrak{g}^{\overrightarrow{w}'_1(Y_1^{-1})^t}, \mathfrak{g}^{-\overrightarrow{w}'_1(Y_2^{-1})^t}), (\mathfrak{g}^{\overrightarrow{w}'_2(Y_1^{-1})^t}, \mathfrak{g}^{-\overrightarrow{w}'_2(Y_2^{-1})^t}), (\mathfrak{h}^{\overrightarrow{z}'_1(X_1^{-1})^t}, \mathfrak{h}^{-\overrightarrow{z}'_1(X_2^{-1})^t}), (\mathfrak{h}^{\overrightarrow{z}'_2(X_1^{-1})^t}, \mathfrak{h}^{-\overrightarrow{z}'_2(X_2^{-1})^t})\},$$

where $X_i(Y_i)^t = D_i$, which is a random diagonal matrix, $\overrightarrow{x}_i$ is the first row of X_i, $\overrightarrow{y}_i$ is the first row of Y_i, and $A', B' \xleftarrow{\$} GL_2(\mathbb{Z}_p)$, $a', b' \xleftarrow{\$} \mathbb{Z}_p$, $\overrightarrow{w}'_1, \overrightarrow{w}'_2, \overrightarrow{z}'_1, \overrightarrow{z}'_2 \xleftarrow{\$} \mathbb{Z}_p^2$.

Now, we describe $\mathcal{B}$'s procedure. First it randomly chooses $A, B, C \xleftarrow{\$} GL_2(\mathbb{Z}_p), a, b \xleftarrow{\$} \mathbb{Z}_p, \overrightarrow{w}, \overrightarrow{z} \xleftarrow{\$} \mathbb{Z}_p^2$, and computes

$$\mathfrak{g}^A, \mathfrak{g}^{AC}, (\mathfrak{h}^{x_1x_4-x_2x_3})^B, {}^B(M_\mathfrak{h})^{(C^{-1})^t},$$

$$G_1 = \langle((\mathfrak{g}^{(ax_1,ax_2)}), (\mathfrak{g}^{(ax_1,ax_2)})^C)\rangle, H_1 = \langle(\mathfrak{h}^{(bx_4,-bx_3)}, (\mathfrak{h}^{(bd_1x_4,-bd_1x_3)})^{(C^{-1})^t})\rangle,$$

$$\sigma = \{((\mathfrak{g}^{d_1d_2(x_1x_4-x_2x_3)})^{\overrightarrow{w}_i}, {}^{-\overrightarrow{w}_i}(M_\mathfrak{g})^C), (\mathfrak{h}^{\overrightarrow{z}_i}, \mathfrak{h}^{-\overrightarrow{z}_i(C^{-1})^t}) \text{ for } i \in [1,2]\},$$

and computes $((\mathfrak{g}^{x_1y}, T_b), (\mathfrak{g}^{x_1y}, T_b)^C)$ as the challenge. Then, $\mathcal{B}$ sends the above all to $\mathcal{A}$ and transfers $\mathcal{A}$'s output as its result.

We argue that $\mathcal{B}$'s simulated transcript is identical to the real transcript. $\mathcal{B}$'s construction is distributed as if

$$X_1 = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}, \quad X_2 = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} C, \quad A' = A(X_1)^{-1},$$

$$D_1 = \begin{pmatrix} d_2r & 0 \\ 0 & d_1s \end{pmatrix}, \quad D_2 = \begin{pmatrix} d_1d_2r & 0 \\ 0 & d_1d_2s \end{pmatrix} C, \quad \text{for some random } r, s \xleftarrow{\$} \mathbb{Z}_p.$$

$$Y_1 = D_1(X_1^{-1})^t, Y_2 = D_2(X_2^{-1})^t, B' = (x_1x_4 - x_2x_3)BX_1^tD_1^{-1},$$

$$a' = a, b' = b\frac{x_1x_4 - x_2x_3}{d_2r}, \overrightarrow{w}'_i = d_1d_2(x_1x_4 - x_2x_3)\overrightarrow{w}_iX_1^{-1}D_1, \text{ and } \overrightarrow{z}'_i = \overrightarrow{z}_iX_1^t.$$

Then, the distribution of the simulated transcript is identical to the distribution of the real transcript.

Let us explain details. First, we consider about $M_\mathfrak{h}$ and $M_\mathfrak{g}$.

$$\begin{pmatrix} d_1x_1x_4 - d_2x_2x_3 & (d_2 - d_1)x_1x_3 \\ (d_1 - d_2)x_2x_4 & d_2x_1x_4 - d_1x_2x_3 \end{pmatrix} = \begin{pmatrix} x_1 & x_3 \\ x_2 & x_4 \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} x_4 & -x_3 \\ -x_2 & x_1 \end{pmatrix}$$

$$= (x_1x_4 - x_2x_3)X_1^tD_1^{-1}D_2(X_1^{-1})^t,$$

so $M_\mathfrak{h} = \mathfrak{h}^{(x_1x_4-x_2x_3)X_1^tD_1^{-1}D_2(X_1^{-1})^t}$, and similarly

$$\begin{pmatrix} d_2x_1x_4 - d_1x_2x_3 & (d_2 - d_1)x_2x_4 \\ (d_1 - d_2)x_1x_3 & d_1x_1x_4 - d_2x_2x_3 \end{pmatrix} = \begin{pmatrix} x_4 & -x_2 \\ -x_3 & x_1 \end{pmatrix} \begin{pmatrix} d_2 & 0 \\ 0 & d_1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}$$

$$= d_1d_2(x_1x_4 - x_2x_3)X_1^{-1}D_1D_2^{-1}X_1,$$

so $M_\mathfrak{g} = \mathfrak{g}^{d_1d_2(x_1x_4-x_2x_3)X_1^{-1}D_1D_2^{-1}X_1}$.

If we consider the exponent parts (that is, discrete logarithm based on $\mathfrak{g}$ or $\mathfrak{h}$,) of elements generated by $\mathcal{B}$, we can verify the following equalities.

$$(A, AC) = (A'X_1, A'X_2),$$

$$((x_1x_4 - x_2x_3)B, (x_1x_4 - x_2x_3)BX_1^tD_1^{-1}D_2(X_1^{-1})^t(C^{-1})^t)$$
$$= (B'D_1(X_1^{-1})^t, B'D_2(X_1^{-1})^t(C^{-1})^t) = (B'Y_1, B'Y_2),$$
$$((ax_1, ax_2), (ax_1, ax_2)C) = (a'\overrightarrow{x}_1, a'\overrightarrow{x}_2),$$
$$((bx_4, -bx_3), (bd_1x_4, -bd_1x_3)(C^{-1})^t)$$
$$= ((\frac{b'd_2rx_4}{(x_1x_4 - x_2x_3)}, \frac{-b'd_2rx_3}{(x_1x_4 - x_2x_3)}), (\frac{b'd_1d_2rx_4}{(x_1x_4 - x_2x_3)}, \frac{-b'd_1d_2rx_3}{(x_1x_4 - x_2x_3)})(C^{-1})^t) = (b'\overrightarrow{y}_1, b'\overrightarrow{y}_2),$$
$$(d_1d_2(x_1x_4 - x_2x_3)\overrightarrow{w}_i, -d_1d_2(x_1x_4 - x_2x_3)\overrightarrow{w}_iX_1^{-1}D_1D_2^{-1}X_1C)$$
$$= (\overrightarrow{w}_i'D_1^{-1}X_1, -\overrightarrow{w}_iD_2^{-1}X_1C) = (\overrightarrow{w}_i'(Y_1^{-1})^t, -\overrightarrow{w}_i'(Y_2^{-1})^t),$$
$$(\overrightarrow{z}_i, -\overrightarrow{z}_i(C^{-1})^t) = (\overrightarrow{z}_i'(X_1^{-1})^t, -\overrightarrow{z}_i'(X_1^{-1})^t(C^{-1})^t) = (\overrightarrow{z}_i'(X_1^{-1})^t, -\overrightarrow{z}_i'(X_2^{-1})^t).$$

The variable y in the challenge $((\mathfrak{g}^{x_1y}, T_b), (\mathfrak{g}^{x_1y}, T_b)^C)$ is a independent from the adversarial view. If $T_b = \mathfrak{g}^{x_2y}$, then the challenge $((\mathfrak{g}^{x_1y}, T_b), (\mathfrak{g}^{x_1y}, T_b)^C) = ((\mathfrak{g}^{x_1}, \mathfrak{g}^{x_2}), (\mathfrak{g}^{x_1}, \mathfrak{g}^{x_2})^C)^y$, and thus the challenge is uniformly distributed in G_1. Otherwise, T_b is uniformly distributed in $\mathbb{G}$, and then the challenge is uniformly distributed in G. Therefore, $\mathcal{B}$ can determine b with the same advantage of $\mathcal{A}$. □

E Other Group Membership Tests

E.1 Batch Group Membership Test

Given $g_1, \cdots, g_m$, we describe Batch Group Membership Test (BGMT).

1. If $g_1, \cdots, g_m \in \mathbb{G}^{n^2}$, then go to the next step. Otherwise, output 0.
2. Let $g_i = \mathfrak{g}^{\Gamma_i}$, where $\Gamma_i = (\alpha_{i11}, \cdots, \alpha_{inn})$. Choose random integers $r_1, \cdots, r_m$, $s_1, \cdots, s_{n^2-n} \in \mathbb{Z}_p$.
3. Test $\prod_{j,\ell\in[1,n]} \hat{e}(\prod_{i\in[1,m]}(\mathfrak{g}^{\alpha_{ij\ell}})^{r_i}, \prod_{i\in[1,n^2-n]}(\mathfrak{h}^{\hat{\psi}_{ij\ell}})^{s_i}) = \hat{e}(\mathfrak{g}, \mathfrak{h})^{(\sum_{i\in[1,m]} r_i\Gamma_i)\cdot(\sum_{i\in[1,n^2-n]} s_i\hat{\Psi}_i)} \stackrel{?}{=} 1_{\mathbb{G}_t}$, where $\hat{\Psi}_i = (\hat{\psi}_{i11}, \cdots, \hat{\psi}_{inn})$ and $\mathfrak{h}^{\hat{\Psi}_i}$ is contained in σ. If yes, output 1. Otherwise, output 0.

It is easy to show that if all $g_1, \cdots, g_m \in G$, BGMT outputs 1. If $g_{i_0} = \mathfrak{g}^{\Gamma_{i_0}} \in \mathbb{G}^{n^2} \setminus G$, then there exists a $\hat{\Psi}_{i_1}$ such that $\Gamma_{i_0} \cdot \hat{\Psi}_{i_1} \neq 0$. Then, the probability that $(\sum_{i\in[1,m]} r_i\Gamma_i) \cdot (\sum_{i\in[1,n^2-n]} s_i\hat{\Psi}_i) = \sum_{i\in[1,m]}\sum_{j\in[1,n^2-n]} r_is_j(\Gamma_i \cdot \hat{\Psi}_j) = 0$ is at most $1/p$, where the probability goes over the choices of r_i's and s_j's; hence, if one of g_i is in $\mathbb{G}^{n^2} \setminus G$, then BGMT outputs 1 at most $1/p$ probability. The cost of BGMT (when we ignore the costs for the step 1 and 2) is n^2m exponentiations and $n^2(m-1)$ multiplications in $\mathbb{G}$, $n^2(n^2-n)$ exponentiations and $n^2(n^2-n-1)$ multiplications in $\mathbb{H}$, n^2-1 multiplications in $\mathbb{G}_t$, and n^2 bilinear map computations.

E.2 More Efficient Group Membership Test Based on k-Linear Assumption

We propose a More Efficient Group Membership Test (MEGMT), which is more efficient than the original GMT, and the soundness of MEGMT is proved when k-Linear assumption holds. First, we need to a little modify $\mathcal{G}$, in particular, $\sigma = \{\hat{e}, \{\mathfrak{g}^{\hat{\Phi}_i}\}_{i\in[1,k]}, \{\mathfrak{h}^{\hat{\Psi}_i}\}_{i\in[1,k]}\}$, where $\hat{\Phi}_i$'s and $\hat{\Psi}_i$'s are uniformly and independently chosen from $\langle\Phi_1, \cdots, \Phi_n\rangle^\perp$ and $\langle\Psi_1, \cdots, \Psi_n\rangle^\perp$, respectively.

Given g, we describe $MEGMT^k(\sigma, g)$.

1. If $g \in \mathbb{G}^{n^2}$, then go to the next step. Otherwise, output 0.

2. Let $g = \mathfrak{g}^{\Gamma}$, where $\Gamma = (\alpha_{11}, \cdots, \alpha_{nn})$. For $\forall i \in [1,k]$, test $\prod_{j,\ell \in [1,n]} \hat{e}(\mathfrak{g}^{\alpha_{j\ell}}, \mathfrak{h}^{\hat{\psi}_{ij\ell}}) = \hat{e}(\mathfrak{g},\mathfrak{h})^{\Gamma \cdot \hat{\Psi}_i} \stackrel{?}{=} 1_{\mathbb{G}_t}$, where $\hat{\Psi}_i = (\hat{\psi}_{i11}, \cdots, \hat{\psi}_{inn})$ and $\mathfrak{h}^{\hat{\Psi}_i}$ is contained in σ. If yes, output 1. Otherwise, output 0.

If $\mathcal{G}_1$ satisfies *k-Linear assumption* in $\mathbb{H}$, then no polynomial time algorithm $\mathcal{A}$ can generate g such that $g \in \mathbb{G}^{n^2} \setminus G$ but $MEGMT^k(\sigma, g) = 1$ with non-negligible probability. We provide the formal theorem below for this argument.

Theorem 10 *If there exists an algorithm $\mathcal{A}$ such that given $(G, G_1, \cdots, G_n, H, H_1, \cdots, H_n, G_t, e, \sigma)$ $\mathcal{A}$ outputs $g \in \mathbb{G}^{n^2} \setminus G$ passing $MEGMT^k$ with ϵ probability, then there exists an algorithm $\mathcal{B}$ solves k-*Linear problem *in $\mathbb{H}$ with $(1 - \frac{1}{p})\epsilon$ advantage. Furthermore, the running time of $\mathcal{B}$ is almost equal to the running time of $\mathcal{A}$.*

Proof. Let us overview the strategy of $\mathcal{B}$ to solve *k-Linear problem* in $\mathbb{H}$ in the bilinear group setting by using $\mathcal{A}$. Suppose that $\mathcal{B}$ receives the instances of *k-Linear problem* in $\mathbb{H}$ in the bilinear group setting, $(\mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e}, \mathfrak{h}, \mathfrak{v}, \mathfrak{v}_i^a, \mathfrak{h}^b,\ i \in [1,k])$. Define a map $\tilde{e}$ by

$$\begin{array}{rccccc} \tilde{e}: & \mathbb{G}^{n^2-n} & \times & \mathbb{H}^{n^2-n} & \to & \mathbb{G}_t \\ & (\mathfrak{g}_1, \cdots, \mathfrak{g}_{n^2-n}), & & (\mathfrak{h}_1, \cdots, \mathfrak{h}_{n^2-n}) & \mapsto & \prod_{i \in [1, n^2-n]} \hat{e}(\mathfrak{g}_i, \mathfrak{h}_i). \end{array}$$

We can see that $\tilde{e}(\mathfrak{g}^{\overrightarrow{x}}, \mathfrak{h}^{\overrightarrow{y}}) = \hat{e}(\mathfrak{g},\mathfrak{h})^{\overrightarrow{x} \cdot \overrightarrow{y}}$ for $\overrightarrow{x}$ and $\overrightarrow{y} \in \mathbb{Z}_p^{n^2-n}$.

Let $h_i' = ((\mathfrak{v}^{a_i})^{r_1}\mathfrak{v}^{s_{i,1}}, \cdots, (\mathfrak{v}^{a_i})^{r_{n^2-n}}\mathfrak{v}^{s_{i,n^2-n}}) \in \mathbb{H}^{n^2-n}$ for random integers r_i's and $s_{i,j}$'s in $\mathbb{Z}_p$, and $i \in [1,k]$. If $\mathcal{B}$ has a non-identity element $g_1' = \mathfrak{g}^{\overrightarrow{\gamma}} \in \mathbb{G}^{n^2-n}$ such that for $\forall i \in [1,k]$ $\hat{e}(g_1', h_i') = 1_{\mathbb{G}_t}$ and g_1' is independent from $\mathfrak{h}^b$, then $\mathcal{B}$ can solves *k-Linear Problem* in $\mathbb{H}$ by testing $\tilde{e}(g_1', h') \stackrel{?}{=} 1_{\mathbb{G}_t}$, where $h' = ((\mathfrak{h}^b)^{r_1}\mathfrak{h}^{\sum_{i\in[1,k]} s_{i,1}}, \cdots, (\mathfrak{h}^b)^{r_{n^2-n}}\mathfrak{h}^{\sum_{i\in[1,k]} s_{i,n^2-n}}) \in \mathbb{H}^{n^2-n}$.

From the fact that $\tilde{e}(g_1', h_i') = 1$ for $\forall i \in [1,k]$, we know that

$$\overrightarrow{\gamma} \cdot (a_i r_1 + s_{i,1}, \cdots, a_i r_{n^2-n} + s_{i,n^2-n}) = 0 \ \text{ for } \forall i.$$

If $b = \sum_{i\in[1,k]} a_i$, then $\overrightarrow{\gamma} \cdot (br_1 + \sum_{i\in[1,k]} s_{i,1}, \cdots, br_{n^2-n} + \sum_{i\in[1,k]} s_{i,n^2-n}) = 0$; hence, $\tilde{e}(g_1', h') = 1_{\mathbb{G}_t}$. Otherwise, i.e. b is random integer, then $(br_1 + \sum_{i\in[1,k]} s_{i,1}, \cdots, br_{n^2-n} + \sum_{i\in[1,k]} s_{i,n^2-n})$ is a uniformly distributed in $\mathbb{Z}_p^{n^2-n}$ and also independent from $(a_i r_1 + s_{i,1}, \cdots, a_i r_{n^2-n} + s_{i,n^2-n}) = 0$ for $\forall i$; The probability that $\overrightarrow{\gamma} \cdot (br_1 + \sum_{i\in[1,k]} s_{i,1}, \cdots, br_{n^2-n} + \sum_{i\in[1,k]} s_{i,n^2-n}) = 0$ is $\frac{1}{p}$ since we assume that g_1' is independent from $\mathfrak{h}^b$ so $(br_1 + \sum_{i\in[1,k]} s_{i,1}, \cdots, br_{n^2-n} + \sum_{i\in[1,k]} s_{i,n^2-n})$ is a random vector independent from $\overrightarrow{\gamma}$; hence $\tilde{e}(g_1', h') = 1_{\mathbb{G}_t}$ with $\frac{1}{p}$ probability.

Therefore, the goal of $\mathcal{B}$ is a finding such a $g_1' \in \mathbb{G}^{n^2-n}$ by using $\mathcal{A}$. If $\mathcal{A}$ outputs such a g_1' with ϵ probability, then $\mathcal{B}$ can solve *k-Linear Problem* with $(1 - \frac{1}{p})\epsilon$ advantage.

Now, we describe $\mathcal{B}$'s procedure to find such a g_1' by using $\mathcal{A}$. From $(\mathbb{G}, \mathbb{H}, \mathbb{G}_t, \hat{e})$, $\mathcal{B}$ normally generates all outputs of $\mathcal{G}$ except σ. Let X_i be used in the procedure of $\mathcal{G}$. $\mathcal{B}$ chooses random integers r_j and $s_{i,j}$ from $\mathbb{Z}_p$ for $j \in [1, n^2-n], i \in [1,k]$, and $\mathcal{B}$ defines $\mathfrak{h}^{\hat{\Psi}_i}$ to be satisfied that

$$\hat{\Psi}_i = \overrightarrow{z}_{i,1} \| \overrightarrow{z}_{i,2} X_1^t (X^{-1})^t \| \cdots \| \overrightarrow{z}_{i,n} X_1^t (X_n^{-1})^t,$$

and

$$\overrightarrow{z}_{i,1} \| \overrightarrow{z}_{i,2} \| \cdots \| \overrightarrow{z}_{i,n-1} = (va_i r_1 + vs_{i,1}, \cdots, va_i r_{n^2-n} + vs_{i,n^2-n}) \text{ and } \overrightarrow{z}_{i,n} = -\sum_{i\in[1,k]} \overrightarrow{z}_i,$$

where $\overrightarrow{z}_i \in \mathbb{Z}_p^n$ and $v = \log_{\mathfrak{h}} \mathfrak{v}$. Since $\mathcal{B}$ has $\mathfrak{v}, \mathfrak{v}^{a_i}$, r_j's, $s_{i,j}$'s and X_i's, $\mathcal{B}$ can compute $\mathfrak{h}^{\hat{\Psi}_i}$ as the above. After normally generating the remaining part of σ, $\mathcal{B}$ sends all outputs to $\mathcal{A}$ and receives $\mathfrak{g}^{\Gamma}$ from $\mathcal{A}$. Let $\Gamma = \overrightarrow{w}_1 \| \cdots \| \overrightarrow{w}_n$ for some $\overrightarrow{w}_i \in \mathbb{Z}_p^n$. $\mathcal{B}$ defines $g_1' \in \mathbb{G}^{n^2-n}$ by

$$(\mathfrak{g}^{\overrightarrow{w}_1}(\mathfrak{g}^{\overrightarrow{w}_n})^{-X_n X_1}, \cdots, (\mathfrak{g}^{\overrightarrow{w}_{n-1}})^{X_{n-1}^{-1} X_1}(\mathfrak{g}^{\overrightarrow{w}_n})^{-X_n X_1}).$$

Next, we argue that g'_1 is a non-identity element in $\mathbb{G}^{n^2}$ and $\tilde{e}(g'_1, h'_i) = 1_{\mathbb{G}_t}$ for $\forall i$ if $\mathfrak{g}^{\Gamma} \in \mathbb{G}^{n^2-n} \setminus G$ and $MEGMT^k(\sigma, \mathfrak{g}^{\Gamma}) = 1$, where $h'_i = \mathfrak{h}^{\overrightarrow{z}_{i,1}\|\cdots\|\overrightarrow{z}_{i,n^2-n}}$. Suppose that $\mathfrak{g}^{\Gamma} \in \mathbb{G}^{n^2} \setminus G$ and $MEGMT^k(\sigma, \mathfrak{g}^{\Gamma}) = 1$. $\mathfrak{g}^{\Gamma} \in \mathbb{G}^{n^2} \setminus G$ if and only if there exists at least one i such that $\overrightarrow{w}_i X_i^{-1} \neq \overrightarrow{w}_n X_n^{-1}$. (In other words, $\mathfrak{g}^{\Gamma'} \in G$ if and only if Γ' is of the form $\overrightarrow{w} X_1 \| \cdots \| \overrightarrow{w} X_n$.) Therefore, $\mathfrak{g}^{\Gamma} \in \mathbb{G}^{n^2} \setminus G$ implies that $g'_1 \in \mathbb{G}^{n^2-n}$ is a non-identity element in $\mathbb{G}^{n^2}$. Furthermore, since $\mathfrak{g}^{\Gamma}$ passes $MEGMT^k$,

$$
\begin{aligned}
0 &= \Gamma \cdot \hat{\Psi}_i \\
&= \sum_{j \in [1,n]} (\overrightarrow{w}_j X_j^{-1} X_1) \cdot \overrightarrow{z}_{i,j} \\
&= \sum_{j \in [1,n-1]} (\overrightarrow{w}_j X_j^{-1} X_1) \cdot \overrightarrow{z}_{i,j} + (\overrightarrow{w}_n X_n^{-1} X_1) \cdot \overrightarrow{z}_{i,n} \\
&= \sum_{j \in [1,n-1]} (\overrightarrow{w}_j X_j^{-1} X_1) \cdot \overrightarrow{z}_{i,j} - \sum_{j \in [1,n-1]} (\overrightarrow{w}_n X_n^{-1} X_1) \cdot \overrightarrow{z}_{i,j} \\
&= \sum_{j \in [1,n-1]} (\overrightarrow{w}_j X_j^{-1} X_1 - \overrightarrow{w}_n X_n^{-1} X_1) \cdot \overrightarrow{z}_{i,j}
\end{aligned}
$$

for all $i \in [1,k]$. Therefore, $\tilde{e}(g'_1, h'_i) = \hat{e}(\mathfrak{g}, \mathfrak{h})^{\sum_{j \in [1,n-1]} (\overrightarrow{w}_j X_j^{-1} X_1 - \overrightarrow{w}_n X_n^{-1} X_1) \cdot \overrightarrow{z}_{ij}} = 1_{\mathbb{G}_t}$; hence, we obtain g'_1 aforementioned at the beginning of the proof.

Finally, we argue that in the view of $\mathcal{A}$ the distribution of $\mathcal{B}$'s output is identical to the that of real $\mathcal{G}$. Showing that $\hat{\Psi}_i$ is independently and uniformly chosen from $\langle \hat{\Psi}_1, \cdots, \hat{\Psi}_n \rangle^{\perp}$ is sufficient to prove it since others are normally generated.

Let us consider about elements in $\langle \hat{\Psi}_1, \cdots, \hat{\Psi}_n \rangle^{\perp}$. For each element $\Theta \in \mathbb{Z}_p^{n^2}$, we can rewrite Θ by $\overrightarrow{u}_1 \| \overrightarrow{u}_2 X_1^t (X_2^{-1})^t \| \cdots \| \overrightarrow{u}_n X_1^t (X_n^{-1})^t$ for some $\overrightarrow{u}_i \in \mathbb{Z}_p^n$ since X_i's are n-rank matrices. Then, we can see the following equivalences:

$$
\begin{aligned}
&\Theta \in \langle \hat{\Psi}_1, \cdots, \hat{\Psi}_n \rangle^{\perp} \text{ if and only if } \hat{\Psi}_i \cdot \Theta = 0 \text{ for } \forall i \in [1,k] \\
\text{if and only if } &\sum_{j \in [1,n]} (\overrightarrow{e}_i X_1) \cdot \overrightarrow{u}_j = 0 \text{ for } \forall i \in [1,k] \text{ if and only if } \sum_{j \in [1,n]} \overrightarrow{u}_j = 0.
\end{aligned}
$$

Therefore, if we randomly choose $\overrightarrow{u}_1, \cdots \overrightarrow{u}_n$ such that $\sum_{j \in [1,n]} \overrightarrow{u}_j = 0$, and we set Θ by

$$
\overrightarrow{u}_1 \| \overrightarrow{u}_2 X_1^t (X_2^{-1})^t \| \cdots \| \overrightarrow{u}_n X_1^t (X_n^{-1})^t,
$$

then Θ is uniformly distributed in $\Theta \in \langle \hat{\Psi}_1, \cdots, \hat{\Psi}_n \rangle^{\perp}$. Since $\mathcal{B}$ defines $\hat{\Psi}_i = \overrightarrow{z}_{i1} \| \overrightarrow{z}_{i,2} X_1^t (X_2^{-1})^t \| \cdots \| \overrightarrow{z}_{i,n} X_1^t (X_n^{-1})^t$ and $\overrightarrow{z}_{i,j}$ are randomly chosen vectors from $\mathbb{Z}_p^n$ with satisfying $\sum_{j \in [1,n]} \overrightarrow{z}_{i,j}$, each $\hat{\Psi}_i$ is independently and uniformly distributed in $\langle \hat{\Psi}_1, \cdots, \hat{\Psi}_n \rangle^{\perp}$. □

We note that we can apply the batch verification technique to $MEGMT^k$ again. We call such a test by $BMEGMT^k$. Moreover, although we describe several tests for elements in G, we can analogously test for elements in H. We provide comparisons among group membership tests in the table 1.

Test	Computational Costs						Error Prob.	Assump.
	$\mathbf{pair}_{\hat{\mathbf{e}}}$	$\mathbf{exp}_{\mathbb{G}}$	$\mathbf{mul}_{\mathbb{G}}$	$\mathbf{exp}_{\mathbb{H}}$	$\mathbf{mul}_{\mathbb{H}}$	$\mathbf{mul}_{\mathbb{G}_{\mathbf{t}}}$		
GMT	$m(n^2-n)n^2$	0	0	0	0	$m(n^2-n)(n^2-1)$	0	·
$BGMT$	n^2	mn^2	$(m-1)n^2$	$(n^2-n)n^2$	$(n^2-n-1)n^2$	n^2-1	neg.	·
$MEGMT^k$	mkn^2	0	0	0	0	$mk(n^2-1)$	neg.	k-linear
$BMEGMT^k$	n^2	mn^2	$(m-1)n^2$	kn^2	$(k-1)n^2$	n^2-1	neg.	k-linear

$\mathbf{pair}_{\hat{\mathbf{e}}}, \mathbf{exp}_{\mathbb{G}}, , \mathbf{mul}_{\mathbb{G}}, \mathbf{exp}_{\mathbb{H}}, \mathbf{mul}_{\mathbb{H}}$, and $\mathbf{mul}_{\mathbb{G}_{\mathbf{t}}}$ mean the number of bilinear map $\hat{e}$ computations, exponentiations in $\mathbb{G}$, multiplications in $\mathbb{G}$, exponentiations in $\mathbb{H}$, multiplications in $\mathbb{H}$, and multiplications in $\mathbb{G}_t$, respectively.

(We ignore the costs for testing membership in $\mathbb{G}^{n^2}$.)

Table 1. Comparisons among group membership tests for m elements in $G \subset \mathbb{G}^{n^2}$