Research Article

Sanjit Chatterjee*, M. Prem Laxman Das and R. Kabaleeshwaran

Converting pairing-based cryptosystems from composite to prime order setting – A comparative analysis

https://doi.org/10.1515/jmc-2017-0042
Received August 2, 2017; revised April 13, 2018; accepted April 13, 2018

Abstract: Composite order pairing setting has been used to achieve cryptographic functionalities beyond what is attainable in prime order groups. However, such pairings are known to be significantly slower than their prime order counterparts. Thus emerged a new line of research – developing frameworks to convert cryptosystems from composite to prime order pairing setting. In this work, we analyse the intricacies of efficient prime order instantiation of cryptosystems that can be converted using existing frameworks. To compare the relative efficacy of these frameworks we mainly focus on some representative schemes: the Boneh–Goh–Nissim (BGN) homomorphic encryption scheme, ring and group signatures as well as a blind signature scheme. Our concrete analyses lead to several interesting observations. We show that even after a considerable amount of research, the projecting framework implicit in the very first work of Groth–Sahai still remains the best choice for instantiating the BGN cryptosystem. Protocols like the ring signature and group signature which use both projecting and cancelling setting in composite order can be most efficiently instantiated in the Freeman prime-order projecting only setting. In contrast, while the Freeman projecting setting is sufficient for the security reduction of the blind signature scheme, the simultaneous projecting and cancelling setting does provide some efficiency advantage.

Keywords: Pairing-based cryptography, projecting, cancelling, BGN encryption, ring signature, blind signature

MSC 2010: 94A60, 11T71, 68P25

Communicated by: Alfred Menezes

1 Introduction

Bilinear pairing was initially proposed and used for cryptographic constructions [7, 24] in the prime order groups. Boneh, Goh and Nissim [8] were the first to demonstrate a novel cryptographic application of pairing defined over composite order groups. They constructed a partially homomorphic public key encryption scheme. Subsequently, numerous other specialized cryptosystems like predicate encryption [9, 25, 36] and signature schemes with additional properties [10, 11, 30, 35] were proposed in such a setting.

***Corresponding author: Sanjit Chatterjee,** Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India, e-mail: sanjit@iisc.ac.in
M. Prem Laxman Das, Society for Electronic Transactions and Security, Chennai, India, e-mail: prem.lax@gmail.com
R. Kabaleeshwaran, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India, e-mail: kabaleeshwar@iisc.ac.in

The composite order pairing groups have two or more distinct subgroups. The corresponding pairing setting has certain special features with respect to these subgroups which have been used to construct novel cryptosystems. For example, the fact that pairing acts like an inner product on the subgroups was used for designing the Katz–Sahai–Waters (KSW) predicate encryption scheme [25]. In the Boneh–Goh–Nissim (BGN) cryptosystem [8], the message is masked with a suitable subgroup element. The compatibility of exponentiation with the underlying pairing was used to obtain the desired homomorphic property in the BGN scheme, while using the subgroup order as the private key allows chopping-off the masking factor in decryption. The natural problem of distinguishing whether an element comes from the whole group or a distinguished subgroup (the so-called subgroup decision problem) forms the basis of the semantic security of BGN cryptosystem or, for that matter, other cryptosystems in the composite order setting.

The subgroup decision problem necessarily requires that the factorization of the group order must be hard which in turn makes composite order pairing a very costly object to compute [18, 22]. Freeman provides sample parameter sizes of bilinear groups for various security levels [18, Section 4] and mentions that for 80-bit security level a composite order Tate pairing on a 1024-bit supersingular curve would be approximately 50 times slower than a prime order Tate pairing on a 170-bit MNT curve. In [22] it was reported that a composite order pairing for 128-bit security level was approximately 254 times slower than its prime order counterpart for the same security level.

This efficiency bottleneck motivated Freeman's work [18]:

> ... for efficient implementations we seek versions of protocols that use *only* prime-order elliptic curve groups. Developing these protocols is the *main goal* of this paper. [emphasis added]

Freeman abstracted out two properties in the composite order setting: projecting and cancelling. He then demonstrated how these properties can be achieved in structures built on prime order pairing groups. A projecting framework was used in [18] to convert the BGN cryptosystem [8] and it was noticed that Groth and Sahai [21] used a similar setting for instantiating their non-interactive witness indistinguishable (NIWI) proof system. A cancelling framework was used to convert the predicate encryption scheme of Katz, Sahai and Waters [25].

After Freeman, several researchers have worked on the composite-to-prime-order conversion theme. In their ASIACRYPT 2010 paper, Meiklejohn, Shacham and Freeman [30] posit certain limitations of such a conversion agenda. They designed a round optimal blind signature scheme in composite order setting which would require a simultaneously projecting and cancelling framework for conversion and examined the improbability of such a setting in prime order groups. However, in a follow-up work at TCC 2012, Seo and Cheon [34] introduced the so-called translating property and obtained a conversion of a slightly modified Meiklejohn–Shacham–Freeman construction. They also proposed a setting that is simultaneously projecting and cancelling using techniques which lie outside the restrictions of the Meiklejohn–Shacham–Freeman impossibility result [30].

As noted by Freeman himself, certain cryptosystems were not amenable to his conversion framework. A prominent example is the Lewko–Waters identity-based encryption (IBE) [29] which requires that the subgroups of the composite order group have relatively prime order. In her EUROCRYPT 2012 paper, Lewko [27] used the setting of dual pairing vector space (DPVS) [31, 32] to achieve a conversion of Lewko–Waters IBE. In their follow-up work of PKC 2015, Lewko and Meiklejohn [28] formalized the notion of parameter hiding. They generalize the Seo–Cheon framework [34] in the DPVS setting to propose an abstract framework that is simultaneously parameter hiding, cancelling and projecting.

The starting point for bilinear group constructions with some additional property like projecting, is an atomic, prime order pairing setting. One then considers "vector spaces" over the groups. A pairing is suitably defined over the augmented structure and distinguished subgroups are appropriately defined to achieve the requisite property. Some researchers have studied the question of optimality of projecting pairing setting. In his ASIACRYPT 2012 paper, Seo [33] investigated this question in both symmetric and asymmetric settings. In their subsequent work, published in CRYPTO 2014, Herold, Hesse, Hofheinz, Ràfols and Rupp [23] proposed a "polynomial" interpretation of the Freeman framework which enables them to circumvent the lower bound

result of Seo in the symmetric setting, albeit based on some non-standard hardness assumptions. Whether their polynomial interpretation can lead to a similar result in the asymmetric setting was left as an open problem in [23].

1.1 Motivation and our contributions

Taking Freeman (or his predecessor Groth–Sahai) as benchmark, we feel it is worthwhile to investigate what has been achieved on the problem of efficient prime order instantiation of converted cryptosystems. This naturally requires a protocol-centric comparative analysis of conversion frameworks. In particular, we focus on cryptosystems in projecting and simultaneously projecting and cancelling frameworks. Some protocols like the BGN cryptosystem and round optimal blind signature have received significant attention in the previous studies [18, 23, 30, 33, 34], whereas protocols such as the ring and group signature schemes from [10, 11, 35] hardly received any attention. We have incorporated both type of cryptosystems in our analysis to shed some more light on the concrete advances in the conversion agenda.

The BGN cryptosystem plays a flagship role in composite order pairing setting. Perhaps that explains why the associated projecting setting is the most studied framework in the context of conversion to prime order groups. After Freeman (and the implicit framework in Groth–Sahai), researchers have continuously strived to improve upon the previous results (see [23, 33, 34]).

In Section 3, we undertake a comprehensive comparison of projecting frameworks vis-a-vis BGN cryptosystem. In the symmetric setting we compare the Groth–Sahai, Freeman, Seo and Herold–Hesse–Hofheinz–Ràfols–Rupp frameworks and in the asymmetric, Freeman, Groth–Sahai and the polynomial asymmetric setting, the last one after recasting the Herold–Hesse–Hofheinz–Ràfols–Rupp framework [23] in the asymmetric pairing setting. In the symmetric setting Herold, Hesse, Hofheinz, Ràfols and Rupp [23] are able to circumvent the previous lower bound result through a polynomial interpretation of Freeman's framework. However, we observe that this interpretation does not yield any such benefit in the asymmetric setting. In fact, the polynomial interpretation is effectively the same as the Groth–Sahai framework. We then compare the cost of instantiating the BGN cryptosystem [8] in each of these settings. Comprehensive comparisons are provided in Table 1 for the asymmetric and in Table 7 for the symmetric setting.

Recall that the question of (im)probability of a simultaneous projecting and cancelling setting has been studied in the literature [30, 34] in the context of the blind signature scheme of [30]. There are other protocols like the Shacham–Waters ring signature [35] that make use of both projecting and cancelling properties in the composite order setting. Thus the question of whether we *necessarily* need a simultaneous projecting and cancelling framework to convert such protocols forms the main focus of Section 4. We show that the Shacham–Waters ring signature can be converted to the prime order setting using projecting property alone. However, the underlying projection framework must allow a complete decomposition of the source groups. We give an appropriate definition and show that it can be achieved in the existing projecting frameworks. A similar conversion strategy works for the other candidates for a simultaneous projecting and cancelling setting, namely the group signature protocols in [10, 11].

We revisit the case of round optimal blind signature in Section 5 and consider the two approaches for security argument: The first one reduces the one-more unforgeability proof to the Diffie–Hellman problem. For an efficient instantiation we introduce an "unbalanced" projecting setting in the asymmetric pairing setting. The proof follows the Seo–Cheon strategy [34] formulated in the symmetric pairing setting. While Seo and Cheon introduced the "translating" property, we show that one does not need such an abstraction. The second approach, used in [30], reduces one-more unforgeability to the security of Waters signature and uses a group generator which is both projecting and cancelling. We compare efficiency of the scheme when instantiated under the two approaches. Our analysis shows that KeyGen and signing (which includes User and Signer computation time) are more efficient in the projecting and cancelling setting while signature size is smaller in the projecting setting. Even though, as observed in Section 4, there is indeed no need to resort to a projecting and cancelling setting for the desired cryptographic functionality, one can still accrue some efficiency benefit by doing so.

In several cases our analysis requires recasting an existing framework and/or a security argument in the asymmetric pairing setting. To improve readability and to focus on the main ideas, we relegate some of the detailed security arguments to the Appendix. We recall some preliminary notions in Section 2. Definitions of various hard problems referred to in this work are given in Appendix A.

2 Preliminary

Notations

As usual, we denote primes by p, q, etc. By source (resp. target) group, we mean the source (resp. target) group of the pairing. By atomic pairing, we mean the prime order pairing which has been used to construct the prime-power setting. By r-fold of a group $\mathbb{G}$, we mean the group $\mathbb{G}^r$, which is a collection of r-tuples endowed with the natural group operation. The notation $[1, n]$ denotes the set $\{1, \ldots, n\}$. Let $\vec{x} = (x_1, \ldots, x_n)$ be a vector over $\mathbb{Z}_p$. We denote $\vec{x}_{|k}$ to be the k-th component of $\vec{x}$, for $k \in [1, n]$. For any group element g, $g^{\vec{x}}$ denotes $(g^{x_1}, \ldots, g^{x_n})$. For a non-empty set A, $a \xleftarrow{\$} A$ denotes the element a is chosen uniformly at random from A. For an algorithm B and any x from the appropriate domain, $B(\cdot) \to x$ denotes that B takes some input and outputs x. We use the notation $\odot$ to denote the component-wise group operation. Let $(g_1, \ldots, g_r)$ and $(g'_1, \ldots, g'_r)$ be from the group $\mathbb{G}^r$; we set

$$(g_1, \ldots, g_r) \odot (g'_1, \ldots, g'_r) = (g_1 g'_1, \ldots, g_r g'_r).$$

We denote tensor product operation by $\otimes$ and is defined as

$$(g_1, \ldots, g_m) \otimes (g'_1, \ldots, g'_n) = (g_1 g'_1, \ldots, g_1 g'_n, \ldots, g_m g'_1, \ldots, g_m g'_n)$$

for $(g_1, \ldots, g_m)$ and $(g'_1, \ldots, g'_n)$ from $\mathbb{G}^m$ and $\mathbb{G}^n$, respectively. In the same way it can be extended for matrices. Let $A = (a_{ij})$ be a matrix of order n, for $(\mathfrak{g}_1, \ldots, \mathfrak{g}_n) \in \mathbb{G}^n$ define

$$(\mathfrak{g}_1, \ldots, \mathfrak{g}_n)^A = (\mathfrak{g}_1^{a_{11}} \mathfrak{g}_1^{a_{21}} \ldots \mathfrak{g}_n^{a_{n1}}, \ldots, \mathfrak{g}_1^{a_{1n}} \mathfrak{g}_1^{a_{2n}} \ldots \mathfrak{g}_n^{a_{nn}}).$$

For any group G with identity element 1, kernel of a function $f : G \to G$ is defined as $\{g \in G : f(g) = 1\}$ and denoted as $\mathrm{Ker}(f)$.

We recall the definitions of bilinear group generator and projection and cancelling property [18].

Definition 1. A bilinear group generator $\mathcal{G}$ is an algorithm which takes as input a security parameter 1^λ to output abelian groups G, H and G_T together with subgroups $G_1 \subset G$ and $H_1 \subset H$ and a bilinear map $e : G \times H \to G_T$. The group descriptions allow efficient group operation and sampling. The properties of the map e are as follows:

- Bilinearity: For all $g, g' \in G$ and $h, h' \in H$, one has

$$e(g \cdot g', h \cdot h') = e(g, h) \cdot e(g, h') \cdot e(g', h) \cdot e(g', h').$$

- Non-degeneracy: If a fixed $g \in G$ satisfies $e(g, h) = 1$ for all $h \in H$, then $g = 1$ and similarly for elements of H.
- Computability: The map is efficiently computable.

The groups involved in the above definition may be of prime or composite order. If $G = H$, then the pairing is said to be symmetric, otherwise it is said to be asymmetric. Due to its simplicity, symmetric pairing has been used in many concrete protocols starting from Joux's protocol and Boneh–Franklin's IBE. However, symmetric pairing over small characteristic fields are effectively broken due to recent advances in solving the discrete-log problem (DLP) in some of these fields (see, for example, [1, 2]). The asymmetric pairing, on the other hand, has several options in terms of the choice of curves and are significantly faster than their symmetric counterparts at higher security levels. Hence, in this paper we will primarily focus on the asymmetric pairing setting.

Projecting property

As a property for aiding conversion, projection was first abstracted by Freeman [18], who used it to convert the Boneh, Goh and Nissim [8] cryptosystem. We recall the definition from Freeman [18, Definition 3.1].

Definition 2. Let $\mathcal{G}$ be a bilinear group generator (Definition 1). Then $\mathcal{G}$ is said to be projecting if it outputs subgroup $G_T' \subset G_T$ and non-trivial homomorphisms π_G, π_H and π_T defined on G, H and G_T to themselves such that

- $G_1 \subseteq \mathrm{Ker}(\pi_G)$, $H_1 \subseteq \mathrm{Ker}(\pi_H)$ and $G_T' \subseteq \mathrm{Ker}(\pi_T)$,
- $e(\pi_G(g), \pi_H(h)) = \pi_T(e(g, h))$, for all $g \in G$ and $h \in H$.

Remark 1. Some variants of the projection property are available in the literature. For example, the non-trivial condition in the definition was introduced by Seo and Cheon [34]. Seo's definition [33, Definition 2] requires that G_T' exists but it need not be explicitly output by the group generator. Herold, Hesse, Hofheinz, Ràfols and Rupp [23, Definition 4] considered the projection property in symmetric setting, where the underlying pairing is multilinear. They assume that the kernel of the projection on the source group is equal to the subgroup output by the bilinear group generator. Lewko and Meiklejohn [28, Definitions 2.1, 2.2, 2.3] define, in addition to Freeman's version of projecting, two other notions which they call weak and full projecting. Weak projecting considers projection maps only on the source groups and not on the target groups. The commutation condition also is not required to hold. Their full projection bilinear group generator outputs decompositions $G = \bigoplus_{i=1}^{n} G_i$, $H = \bigoplus_{i=1}^{n} H_i$ and $G_T = \bigoplus_{i=1}^{n} G_{T,i}$ and non-trivial maps $\pi_{G_i} : G \to G_i$, $\pi_{H_i} : H \to H_i$ and $\pi_{G_{T,i}} : G_T \to G_{T,i}$ such that $e(\pi_{G_i}(g), \pi_{H_i}(h)) = \pi_{G_{T,i}}(e(g, h))$, for all $g \in G$ and $h \in H$.

Cancelling property

We consider a bilinear pairing on groups (G, H, G_T) of composite order, say, $N = pq$, where p and q are primes. An arbitrary element from subgroup G_p of G (resp. H_q of H) can be represented as $g^{\alpha_1 q}$ (resp. $h^{\alpha_2 p}$) for some α_1, α_2 from $\mathbb{Z}_N$. It is now easy to see that the pairing of an element of G_p and H_q would yield the trivial element. This "orthogonality" of the two distinguished subgroups was abstracted by Freeman [18, Definition 3.5] as cancelling property who used it to convert the Katz–Sahai–Waters predicate encryption scheme [25].

Definition 3. A bilinear group generator $\mathcal{G}$ is said to satisfy the r-cancelling property if, it in addition, outputs groups G_i, H_i, $i = 1, \ldots, r$, such that

- $G \cong G_1 \times \cdots \times G_r$ and $H \cong H_1 \times \cdots \times H_r$ and
- $e(g_i, h_j) = 1$, whenever $g_i \in G_i$, $h_j \in H_j$ and $i \neq j$.

3 Projecting setting and BGN cryptosystem

Boneh, Goh and Nissim [8] were the first to use composite order pairing setting for cryptographic construction. Freeman [18] abstracted the projecting property to convert their homomorphic public key encryption scheme into prime order setting. He also noticed that Groth and Sahai [21] implicitly follow a similar approach for their NIWI proof system. Efficient implementation being the primary goal, Freeman focused solely on the asymmetric pairing setting. Subsequently, other constructions were proposed [23, 33] to achieve the projecting property in the symmetric pairing setting. These later works aimed at achieving optimal construction of symmetric projecting setting. The polynomial setting [23] achieves the most efficient construction and extending this approach to asymmetric setting was left as an open problem.

In this section we provide a natural polynomial interpretation based construction of asymmetric projecting setting followed by a comparative analysis of Freeman, Groth–Sahai and polynomial construction with a focus on the BGN homomorphic encryption scheme. A similar comparative analysis of the symmetric projection setting is provided in Appendix B.

3.1 Asymmetric projecting frameworks

We discuss three candidate constructions for asymmetric projecting bilinear setting, namely, Freeman [18], Groth–Sahai [21] and polynomial.

All of these constructions are defined on a 2-fold of the source atomic groups to the 4-fold of the target group and require four atomic pairings. However, it is Groth–Sahai which is most efficient for BGN instantiation due to its efficient subgroup representation and hence efficient projection maps.

Recall that the projection map was defined by Freeman as operating from G to G (or H to H). Defining the map from the two-fold of the atomic group to the atomic group ($G \to \mathbb{G}_1$) can result in efficiency gain. The commutation of the pairing with the projection map can be suitably redefined. This idea has been used by Groth and Sahai [21]. We examine the efficiency gained due to this idea in BGN instantiation.

3.1.1 Freeman construction

We briefly recall the Freeman projection framework (see [18, Example 3.3] for further details). Let $\mathcal{P}(1^\lambda)$ be an asymmetric prime order bilinear group generator, which outputs $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$. Let $\mathfrak{g}$ be a generator of $\mathbb{G}_1$ and let $\mathfrak{h}$ be that of $\mathbb{G}_2$. Define the groups $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$ and $G_T = \mathbb{G}_T^4$. The bilinear pairing $e : G \times H \to G_T$ is defined as,

$$e((\mathfrak{g}_1, \mathfrak{g}_2), (\mathfrak{h}_1, \mathfrak{h}_2)) := (\hat{e}(\mathfrak{g}_1, \mathfrak{h}_1), \hat{e}(\mathfrak{g}_1, \mathfrak{h}_2), \hat{e}(\mathfrak{g}_2, \mathfrak{h}_1), \hat{e}(\mathfrak{g}_2, \mathfrak{h}_2))$$

for $(\mathfrak{g}_1, \mathfrak{g}_2) \in G$ and $(\mathfrak{h}_1, \mathfrak{h}_2) \in H$.

Choose two pairs of linearly independent vectors $\{(a_1, b_1), (c_1, d_1)\}$ and $\{(a_2, b_2), (c_2, d_2)\}$ uniformly at random from $\mathbb{Z}_p^2$ such that $a_i d_i - b_i c_i = 1$, for $i = [1, 2]$. The subgroup G_1 is generated by $(\mathfrak{g}^{a_1}, \mathfrak{g}^{b_1})$ and H_1 by $(\mathfrak{h}^{a_2}, \mathfrak{h}^{b_2})$.

Consider the matrices

$$A := \begin{pmatrix} -b_1 c_1 & -b_1 d_1 \\ a_1 c_1 & a_1 d_1 \end{pmatrix} \quad \text{and} \quad B := \begin{pmatrix} -b_2 c_2 & -b_2 d_2 \\ a_2 c_2 & a_2 d_2 \end{pmatrix}. \tag{3.1}$$

The projection map $\pi_G : G \to G$ is defined as

$$\pi_G(\mathfrak{g}_1, \mathfrak{g}_2) := (\mathfrak{g}_1, \mathfrak{g}_2)^A = (\mathfrak{g}_1^{-b_1 c_1} \mathfrak{g}_2^{a_1 c_1}, \mathfrak{g}_1^{-b_1 d_1} \mathfrak{g}_2^{a_1 d_1})$$

for $(\mathfrak{g}_1, \mathfrak{g}_2) \in G$. Similarly, $\pi_H : H \to H$ is defined as

$$\pi_H(\mathfrak{h}_1, \mathfrak{h}_2) := (\mathfrak{h}_1, \mathfrak{h}_2)^B = (\mathfrak{h}_1^{-b_2 c_2} \mathfrak{h}_2^{a_2 c_2}, \mathfrak{h}_1^{-b_2 d_2} \mathfrak{h}_2^{a_2 d_2})$$

for $(\mathfrak{h}_1, \mathfrak{h}_2) \in H$. Observe that $G_1 \subseteq \operatorname{Ker} \pi_G$ and $H_1 \subseteq \operatorname{Ker} \pi_H$.

Let $g = (\mathfrak{g}_1, \mathfrak{g}_2) \in G$, $g_1 = (\mathfrak{g}^{a_1}, \mathfrak{g}^{b_1}) \in G_1$, $h = (\mathfrak{h}_1, \mathfrak{h}_2) \in H$ and $h_1 = (\mathfrak{h}^{a_2}, \mathfrak{h}^{b_2}) \in H_1$. Then the subgroup G_T' of G_T is generated by $\{e(g, h_1), e(g_1, h), e(g_1, h_1)\}$. The projection map $\pi_T : G_T \to G_T$ is defined as

$$\pi_T(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4) = (\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4)^{A \otimes B}.$$

Observe that $G_T' \subseteq \operatorname{Ker} \pi_T$. Output $(G, H, G_T, G_1, H_1, G_T', e, \pi_G, \pi_H, \pi_T)$.

Hardness of the subgroup decision problem (SDP) is proved under the SXDH assumption in the atomic pairing setting. See [18, Proposition 3.4] for further details.

Remark 2. We comment on an alternative definition of the projecting property, which actually leads to some efficiency gain. Recall that the projection maps were defined as operating from G to G. They may also be defined as follows. Consider an atomic pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let $G = \mathbb{G}_1^2$. The projection map π_G now operates from $G \to \mathbb{G}_1$. The map $\pi_H : H \to \mathbb{G}_2$ and $\pi_T : G_T \to \mathbb{G}_T$ are suitably defined. The commutation condition now may be redefined as

$$\pi_T(e(g, h)) = \hat{e}(\pi_G(g), \pi_H(h)), \quad g \in G \text{ and } h \in H.$$

We do not distinguish this definition from Freeman's definition of projecting property hereafter.

3.1.2 Groth–Sahai construction

We have already mentioned that the Groth–Sahai construction [21] of the projecting property predates the Freeman construction. It was, however, not used in the context of conversion. This construction uses the projection as defined in Remark 2. With the atomic pairing, groups G and H and pairing as in Freeman setting, we recall the Groth–Sahai construction here.

Choose α and β uniformly at random from $\mathbb{Z}_p^*$ and define the subgroup G_1 which is generated by $(\mathfrak{g}, \mathfrak{g}^\alpha)$ and the subgroup H_1 is by $(\mathfrak{h}, \mathfrak{h}^\beta)$. The projection map $\pi_G : G \to \mathbb{G}_1$ is defined as

$$\pi_G(\mathfrak{g}_1, \mathfrak{g}_2) := \mathfrak{g}_1^{-\alpha}\mathfrak{g}_2$$

and the map $\pi_H : H \to \mathbb{G}_2$ as

$$\pi_H(\mathfrak{h}_1, \mathfrak{h}_2) := \mathfrak{h}_1^{-\beta}\mathfrak{h}_2.$$

The subgroup G_T' of G_T is generated similarly as in Freeman's projection construction. The projection map $\pi_T : G_T \to \mathbb{G}_T$ is defined as

$$\pi_T(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4) = \mathcal{T}_1^{\alpha\beta}\mathcal{T}_2^{-\alpha}\mathcal{T}_3^{-\beta}\mathcal{T}_4$$

for $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4) \in G_T$. It is easy to see that $G_1 = \operatorname{Ker}\pi_G$, $H_1 = \operatorname{Ker}\pi_H$ and $G_T' = \operatorname{Ker}\pi_T$. Finally, the bilinear group generator outputs $(G, H, G_T, G_1, H_1, G_T', e, \pi_G, \pi_H, \pi_T)$.

The SDP hardness can be proved under SXDH assumption in the atomic groups as in the Freeman case.

3.1.3 Polynomial framework

In Appendix B, we describe how Herold, Hesse, Hofheinz, Ràfols and Rupp [23] could obtain an improved construction of projecting symmetric setting by using a univariate polynomial interpretation of vectors. They asked whether a similar result can be obtained in asymmetric setting. We use the two variable polynomial-based representation of vectors to obtain a projecting setting. However, based on SXDH assumption we obtain similar parameters as that of Freeman and Groth–Sahai. In the exposition we use the atomic pairing, groups and pairing definition as in Freeman's projection description.

Choose the hidden parameters $s, s' \xleftarrow{\$} \mathbb{Z}_p$ and define the subgroup of G as

$$G_1^{(s)} := \{(\mathfrak{g}^{a_0}, \mathfrak{g}^{a_1}) \in G : \mathfrak{g}^{a_0}(\mathfrak{g}^{a_1})^s = 1,\ a_0, a_1 \in \mathbb{Z}_p\}$$

and the subgroup of H as

$$H_1^{(s')} := \{(\mathfrak{h}^{b_0}, \mathfrak{h}^{b_1}) \in H : \mathfrak{h}^{b_0}(\mathfrak{h}^{b_1})^{s'} = 1,\ b_0, b_1 \in \mathbb{Z}_p\}.$$

Then define the projection map $\pi_G^{(s)} : G \to \mathbb{G}_1$ by

$$\pi_G^{(s)}(\mathfrak{g}^{a_0}, \mathfrak{g}^{a_1}) := \mathfrak{g}^{a_0}(\mathfrak{g}^{a_1})^s$$

and the map $\pi_H^{(s')} : H \to \mathbb{G}_2$ by

$$\pi_H^{(s')}(\mathfrak{h}^{b_0}, \mathfrak{h}^{b_1}) := \mathfrak{h}^{b_0}(\mathfrak{h}^{b_1})^{s'}.$$

Observe that $G_1^{(s)} = \operatorname{Ker}\pi_G^{(s)}$ and $H_1^{(s')} = \operatorname{Ker}\pi_H^{(s')}$.

Now define the subgroup of G_T as

$$G_T^{(s,s')} := \{(\mathfrak{g}_T^{c_0}, \ldots, \mathfrak{g}_T^{c_3}) \in G_T : \mathfrak{g}_T^{c_0}(\mathfrak{g}_T^{c_1})^{s'}(\mathfrak{g}_T^{c_2})^{s}(\mathfrak{g}_T^{c_3})^{ss'} = 1,\ c_0, \ldots, c_3 \in \mathbb{Z}_p\}.$$

The projection map $\pi_T^{(s,s')} : G_T \to \mathbb{G}_T$ is defined as

$$\pi_T^{(s,s')}(\mathfrak{g}_T^{c_0}, \ldots, \mathfrak{g}_T^{c_3}) := \mathfrak{g}_T^{c_0}(\mathfrak{g}_T^{c_1})^{s'}(\mathfrak{g}_T^{c_2})^{s}(\mathfrak{g}_T^{c_3})^{ss'}.$$

Output $(G, H, G_T, e, G_1^{(s)}, H_1^{(s')}, \pi_G^{(s)}, \pi_H^{(s')}, \pi_T^{(s,s')})$.

Using the proof idea similar to [18, Theorem 2.5], one can show that SXDH assumption in the atomic pairing setting implies the hardness of SDP in the polynomial projecting setting described above. In [23], Herold, Hesse, Hofheinz, Ràfols and Rupp proved a more general result. In the symmetric multilinear setting, they showed that hardness of SDP is implied by any matrix decisional Diffie–Hellman (MDDH) assumption [17]. That result can be extended to the asymmetric bilinear pairing setting using the above idea.

3.1.4 Discussion

We note that the polynomial asymmetric framework is essentially a reformulation of the Groth–Sahai framework. Instead of representing elements of G and H as abstract 2-vectors, they are respectively interpreted as linear polynomials in X and Y. The tensor product of two vectors and multiplication of two such polynomials are the same. Subgroups are interpreted as the set of polynomials which vanish at a fixed hidden parameter (different parameters for G and H). Nevertheless, polynomial interpretation does allow a slightly simpler projection map computation in terms of polynomial evaluation.

For all the three asymmetric projecting frameworks, we have focused on the concrete case where the source group G (resp. H) can be expressed as $G_1 \oplus G_2$ (resp. $H_1 \oplus H_2$) – by allowing, if necessary, a complete decomposition (see Definition 4).

In a typical cryptographic application (see BGN of Section 3.2 for a concrete example), G_1 and H_1 are used as the "masking" subgroups while G_2 and H_2 act as the "unmasking" subgroup. In all the above constructions, both masking and unmasking subgroups are of rank one. There is no practical motivation to go beyond rank one masking subgroup as it covers instantiation of all known protocols employing projecting setting.[1] In this concrete setting, it is easy to see that polynomial asymmetric framework is equivalent to the Freeman framework and hence, the Groth–Sahai framework.

Recall that Seo [33] came up with the following theorem.

Theorem 1 ([33, Theorem 3]). *Under the assumption that for the atomic pairing the k-linear assumption holds, the image of the asymmetric projecting pairing is at least $(k+1)^2$-tuple.*

Since 1-linear assumption is the SXDH assumption, one concludes that Freeman as well as Groth–Sahai and polynomial framework are optimal in terms of the target group size and number of atomic pairings. Hence, unlike the symmetric setting, the polynomial interpretation due to [23] does not provide a more efficient realization of the projecting framework.

3.2 BGN scheme

We recall the abstract description of the construction in asymmetric pairing setting. This will be in terms of elements of the groups, group operations, projection maps and pairing computations. We use this description to benchmark various projecting settings. The same model is used in the symmetric setting with obvious interpretation.

- KeyGen(1^λ): Let $\mathcal{G}(1^\lambda)$ output the tuple $(G, H, G_T, e, G_1, H_1, G'_T, \pi_G, \pi_H, \pi_T)$, a asymmetric projecting bilinear group. Let $g \xleftarrow{\$} G$ and $h \xleftarrow{\$} H$ be the random group elements and let $g_1 \xleftarrow{\$} G_1$ and $h_1 \xleftarrow{\$} H_1$ be the random subgroup elements. Set the public key PK $= (G, H, e, g, h, g_1, h_1)$ and the corresponding secret key SK $= (\pi_G, \pi_H, \pi_T)$.
- Enc(PK, m): Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$ and compute the ciphertext as CT $= (g^m \cdot g_1^r, h^m \cdot h_1^s)$.
- Multiply(PK, C, C'): Let $C \in G$ and $C' \in H$ be from two ciphertexts. Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$ and output

$$e(C, C') \cdot e(g_1, h)^r \cdot e(g, h_1)^s \in G_T.$$

- Add(PK, C, C'): Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$. Then do the following:
  - If $C, C' \in G$, then output $C \cdot C' \cdot g_1^r \in G$.
  - If $C, C' \in H$, then output $C \cdot C' \cdot h_1^s \in H$.
  - If $C, C' \in G_T$, then output $C \cdot C' \cdot e(g_1, h)^r \cdot e(g, h_1)^s \in G_T$.

1 Apart from BGN cryptosystem, examples of such protocols are the Groth–Sahai proof system [21], the Shacham–Waters ring signature [35], the Boyen–Waters group signature [10, 11], Meiklejohn–Shacham–Freeman's round optimal blind signature [30]. The only scheme in the asymmetric setting which requires a masking subgroup (H_1) of rank two is our conversion of the round optimal blind signature in Section 5.

- Dec(SK, C): Check for membership of the ciphertext in various groups and decrypt as follows:
  - If $C \in G$, then output $m = \mathrm{dlog}_{\pi_G(g)}(\pi_G(C))$.
  - If $C \in H$, then output $m = \mathrm{dlog}_{\pi_H(h)}(\pi_H(C))$.
  - If $C \in G_T$, then output $m = \mathrm{dlog}_{\pi_T(e(g,h))}(\pi_T(C))$.

3.2.1 Correctness

We describe correctness with respect to the Groth–Sahai framework, a similar argument holds for the Freeman setting. For any key pair (PK, SK) output by KeyGen algorithm and for any message M from the message space, we have Dec(SK, Enc(PK, M))= M. Consider, for example, the case when the ciphertext is in G. The group element g is represented as $(\mathfrak{g}^{x_1}, \mathfrak{g}^{x_2})$ whereas the subgroup element g_1 as $(\mathfrak{g}^{r}, \mathfrak{g}^{r\alpha})$. The projection map π_G evaluated on g outputs $\mathfrak{g}^{-\alpha x_1 + x_2}$ and evaluation of π_G on $C = g^M g_1$ outputs $\mathfrak{g}^{M(-\alpha x_1 + x_2)}$. Thus the discrete logarithm of $\pi_G(C)$ to the base $\pi_G(g)$ returns the message M.

3.2.2 Instantiation

We briefly describe several strategies for an efficient instantiation of BGN cryptosystem. To obtain the ciphertext component, say $g^M \cdot g_1^r$ in G, the naive approach would be to compute g^M. However, in the prime order setting g is represented as $(\mathfrak{g}_1, \mathfrak{g}_2) \in \mathbb{G}_1^2$. Rather than computing g^M, for example in the Groth–Sahai framework, a more efficient approach is to exponentiate $(0, M)$ with g whence the decryption algorithm uses projection map applied on $g^{(0,1)} = (1, \mathfrak{g}_2)$ instead of g. A similar strategy was suggested in [23, Appendix F.2].

While instantiating BGN in the Freeman framework, one can encrypt either $(M, 0)$ or $(0, M)$. If $(M, 0)$ is used in encryption algorithm, then in decryption one uses the projection map on $g^{(1,0)} = (\mathfrak{g}_1, 1)$. Recall that decryption uses the projection map π_G which is defined based on the matrix A (see equation (3.1)). Using this technique, the projection map computes only the first component $\pi_G(\mathfrak{g}_1, \mathfrak{g}_2)|1$ and outputs $\mathfrak{g}_1^{-b_1 c_1} \cdot \mathfrak{g}_2^{a_1 c_1}$. A similar optimization works for H and G_T. This strategy, first used in [33, Section 5.3], improves the performance over the naive decryption method by a factor of two in G, H and a factor of four in G_T.

After taking into account the above mentioned optimizations, we compare the performance of Freeman versus Groth–Sahai/Polynomial framework for the instantiation of BGN cryptosystem [8]. The public key size and ciphertext size are the same in both cases. In terms of computation cost, both incur the same cost for encryption and homomorphic operations. However, the key generation is slightly faster in the Groth–Sahai framework due to the fact that the masking subgroup generators are constructed with only one hidden secret instead of two as in the Freeman framework. Similarly, decryption is faster in the Groth–Sahai framework, because of the more efficient projection map computation. See Table 1 for a detailed comparison.

Recall that the Groth–Sahai NIWI proof system [21] also uses the projecting setting. However, the projection map is not used explicitly in the construction. Hence instantiating this proof system in the asymmetric projecting setting results in the same efficiency in the above two frameworks.

4 Cryptosystems in projecting-and-cancelling setting

Meiklejohn, Shacham and Freeman [30] proposed a blind signature scheme which requires both projecting and cancelling properties for the security argument. They also address the difficulty of getting a prime order group generator satisfying both these properties and summarize by stating an improbability result under suitable assumptions [30, Theorem 6.5]. However, Seo and Cheon [34] constructed a projecting and cancelling group generator which is outside the restrictions of this improbability result.

We first briefly recall the Seo–Cheon construction and show that it can be based on the SXDH assumption. Then we revisit the cryptographic constructions such as the Shacham–Waters ring signature [35] and the

		Freeman	Groth–Sahai/Polynomial
SK size		$8\lvert\mathbb{F}_p\rvert$	$2\lvert\mathbb{F}_p\rvert$
PK size		$4\lvert\mathbb{G}_1\rvert + 4\lvert\mathbb{G}_2\rvert$	
CT size		$2\lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{G}_2\rvert$ $4\lvert\mathbb{G}_T\rvert$	
KeyGen		$4E_{\mathbb{G}_1} + 4E_{\mathbb{G}_2}$	$3E_{\mathbb{G}_1} + 3E_{\mathbb{G}_2}$
Enc	$\mathbb{G}_1$	$2E_{\mathbb{G}_1} + 1M_{\mathbb{G}_1}$	
	$\mathbb{G}_2$	$2E_{\mathbb{G}_2} + 1M_{\mathbb{G}_2}$	
Dec†	$\mathbb{G}_1$	$3E_{\mathbb{G}_1} + 1M_{\mathbb{G}_1}$	$1E_{\mathbb{G}_1} + 1M_{\mathbb{G}_1}$
	$\mathbb{G}_2$	$3E_{\mathbb{G}_2} + 1M_{\mathbb{G}_2}$	$1E_{\mathbb{G}_2} + 1M_{\mathbb{G}_2}$
	$\mathbb{G}_T$	$1\mathbb{P} + 5E_{\mathbb{G}_T} + 3M_{\mathbb{G}_T}$	$1\mathbb{P} + 3E_{\mathbb{G}_T} + 3M_{\mathbb{G}_T}$
Add	$\mathbb{G}_1$	$2E_{\mathbb{G}_1} + 4M_{\mathbb{G}_1}$	
	$\mathbb{G}_2$	$2E_{\mathbb{G}_2} + 4M_{\mathbb{G}_2}$	
	$\mathbb{G}_T$	$8\mathbb{P} + 12M_{\mathbb{G}_T} + 4E_{\mathbb{G}}$	
Multiply		$12\mathbb{P} + 8M_{\mathbb{G}_T} + 4E_{\mathbb{G}}$	

Table 1: BGN instantiation in asymmetric projection frameworks. (For any group $X \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$, we denote by E_X, M_X and $|X|$ the exponentiation, multiplication in X and the bit size of X, respectively, and $\mathbb{P}$ denotes the atomic asymmetric pairing.
† Excluding the final discrete logarithm computation.)

Boyen–Waters group signature [10, 11] that also apparently require both these properties in the composite order setting. To the best of our knowledge, the question of prime order instantiation of these cryptosystems, which incidentally predate [30], was never seriously addressed in the literature.

4.1 Seo–Cheon framework

We describe in Algorithm 1 a variant of the Seo–Cheon construction for the case $n = 2$ which suffices for all known instantiations of cryptosystems in the projecting and cancelling setting. This framework is of relevance in the context of round optimal blind signature also (see Section 5).

Seo and Cheon relied on a tailor-made assumption for the security of their framework [34, Definition 12]. Following the strategy outlined in [28], we show that the security can be based on SXDH assumption.

Theorem 2. *If $\mathcal{P}$ satisfies the SXDH assumption, then $\mathcal{G}_{PC}$ satisfies the subgroup decision assumption.*

Proof. Given the DDH problem instance $\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, \mathfrak{g}^c$ in $\mathbb{G}_1$, the simulator's goal is to decide whether $\mathfrak{g}^c$ is $\mathfrak{g}^{ab}$ or not. The simulator $\mathcal{B}$ randomly chooses the matrices X_i, Y_j, D from $\mathrm{GL}_2(\mathbb{Z}_p)$ as described in the construction. Then $\mathcal{B}$ embeds the DDH instance to construct 4-fold groups. In particular, the subgroup G_1 is generated using $\mathfrak{g}^{(\chi_{11}+a\cdot\chi_{21}, \chi_{12}+a\cdot\chi_{22})}$ and the subgroup H_1 is generated using $\mathfrak{h}^{(\delta_{11}, \delta_{12})}$, where χ_{ij} is the i-th row of X_j and δ_{ij} is the i-th row of Y_j, for $i, j \in [1, 2]$. Now $\mathcal{B}$ sets $\mathfrak{g}^{(\chi_{21}, \chi_{22})}$ to generate the subgroup G_2 and implicitly sets $\mathfrak{h}^{(\delta_{21}-d_3d_1^{-1}a\delta_{11}, \delta_{22}-d_4d_2^{-1}a\delta_{12})}$ to generate the subgroup H_2 (these subgroups description will not be given to the SDP solver $\mathcal{A}$). Hence this construction ensures that G_i is orthogonal to H_j for $i \neq j$ and $i, j \in [1, 2]$. Now $\mathcal{B}$ sends the SDP instance (G, G_1, H, H_1, e, G_T) along with the challenge term $\mathfrak{g}^{(b\chi_{11}+c\chi_{21}, b\chi_{12}+c\chi_{22})}$ to $\mathcal{A}$. Whenever $\mathfrak{g}^c$ is $\mathfrak{g}^{ab}$, then the above challenge term belongs to the subgroup G_1, otherwise it belongs to the group G. Similarly we can reduce the DDH problem in $\mathbb{G}_2$ to the SDP problem in H. □

4.2 Shacham–Waters ring signature

A ring signature enables a user to anonymously sign a message on behalf of a group of users called a "ring" formed in an ad-hoc manner. The key security attributes are anonymity and unforgeability. Informally speaking, anonymity (against full key exposure) ensures that the adversary cannot distinguish between two target

Algorithm 1. Bilinear group generator $\mathcal{G}_{PC}$ in the Seo–Cheon setting.

Input: The security parameter 1^λ.

Output: $(G, H, G_T, e, \{G_i, H_i, G_{T,i}, \pi_{G,i}, \pi_{H,i}, \pi_{T,i}\}_{i=1}^2, \Omega)$.

1 Run $\mathcal{P}(1^\lambda) \to (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$ to obtain prime order bilinear groups. Let $\mathfrak{g}, \mathfrak{h}$ be the random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

2 Choose $X_1, X_2, D \xleftarrow{\$} GL_2(\mathbb{Z}_p)$. For each $i = 1, 2$, define $D_i \in \mathrm{Mat}_2(\mathbb{Z}_p)$ to be a diagonal matrix having the i-th column vector D as its diagonal and define $Y_i = D_i(X_i^{-1})^\top$.

3 For each $i = 1, 2$, let χ_{ij} be the i-th row of X_j and let δ_{ij} be the i-th row of Y_j for $j \in [1, 2]$. Let $\chi_i = (\chi_{i1}, \chi_{i2})$ and $\delta_i = (\delta_{i1}, \delta_{i2})$. Now define $G_i = \langle \mathfrak{g}^{\chi_i} \rangle$ to be a cyclic subgroup of $\mathbb{G}_1^4$. Similarly define $H_i = \langle \mathfrak{h}^{\delta_i} \rangle$ to be a cyclic subgroup of $\mathbb{G}_2^4$. Note that the order of G_i and H_i is p.

4 Define $G := G_1 \oplus G_2$, $H := H_1 \oplus H_2$ and $G_T := \mathbb{G}_T^2$. Then the bilinear map $e : G \times H \to G_T$ is defined as

$$e(\mathfrak{g}^\Gamma, \mathfrak{h}^\Lambda) := \left(\hat{e}(\mathfrak{g}^{\alpha_{11}}, \mathfrak{h}^{\beta_{11}})\, \hat{e}(\mathfrak{g}^{\alpha_{12}}, \mathfrak{h}^{\beta_{12}}), \hat{e}(\mathfrak{g}^{\alpha_{21}}, \mathfrak{h}^{\beta_{21}})\, \hat{e}(\mathfrak{g}^{\alpha_{22}}, \mathfrak{h}^{\beta_{22}})\right)$$

for any vectors $\Gamma = (\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22})$ and $\Lambda = (\beta_{11}, \beta_{12}, \beta_{21}, \beta_{22})$ from $\mathbb{Z}_p^4$.

5 Let $\{\delta_3, \delta_4\}$ be a random basis of $\langle \chi_1, \chi_2 \rangle^\perp$. Similarly let $\{\chi_3, \chi_4\}$ be a random basis of $\langle \delta_1, \delta_2 \rangle^\perp$. Let $\Omega := \{\hat{e}, \{\mathfrak{g}^{\chi_3}, \mathfrak{g}^{\chi_4}\}, \{\mathfrak{h}^{\delta_3}, \mathfrak{h}^{\delta_4}\}\}$.

6 The subgroup $G_{T,1}$ is generated using $\hat{e}(\mathfrak{g}, \mathfrak{h})^{(d_1, d_2)}$ and $G_{T,2}$ is generated using $\hat{e}(\mathfrak{g}, \mathfrak{h})^{(d_3, d_4)}$, where (d_1, d_2) and (d_3, d_4) are the first and second row of the matrix D.

7 The projection maps are defined as natural projection maps, i.e., $\pi_{G,i}(g) := g^{M^{-1}U_iM}$ and $\pi_{H,i}(h) := h^{N^{-1}U_iN}$ for $g \in G$, $h \in H$, where M (resp. N) is the matrix of order 4 whose j-th row is χ_j (resp. δ_j) and U_i is the boolean matrix of order 4 whose (i, i)-th entry is 1 and the other entries are zero for $i \in [1, 2]$ and $j \in [1, 4]$.

8 The projection map $\pi_{T,i} : G_T \to G_{T,i}$ is defined as $\pi_{T,i}(g_T) := g_T^{D^{-1}V_iD}$ for $g_T \in G_T$, where V_i is a matrix of order 2 whose (i, i)-th entry is 1 and all other entries are zero.

signers even when she/he is given all the private keys in the ring. Unforgeability (with respect to insider corruption), on the other hand, ensures that the adversary cannot forge a signature on behalf of an uncorrupted user. See Bender, Katz and Morselli [5] for formal definition and security properties of ring signature.

We start with an abstract description of the ring signature scheme in the asymmetric pairing setting. This is in terms of a bilinear group generator $\mathcal{G}$ which outputs $(G, H, G_T, e, \{G_i, H_i, G_{T,i}\}_{i\in[1,2]}, \pi_G, \pi_H, \pi_T)$. Here $|G| = |H| = |G_T| = n$, and G_i, H_i and $G_{T,i}$ are distinguished subgroups of G, H and G_T, respectively, and π_G, π_H, π_T are projection maps defined in G, H and G_T, respectively. Setting $G = H$ and $n = pq$, for primes p and q, one gets the original construction [35] in composite order symmetric pairing setting.

4.2.1 Scheme construction

RS-Global-Setup(1^λ). The setting up authority runs the bilinear group generator $\mathcal{G}$ to obtain

$$(G, H, G_T, e, \{G_i, H_i, G_{T,i}\}_{i\in[1,2]}, \pi_G, \pi_H, \pi_T).$$

The authority chooses $g \xleftarrow{\$} G$, $g_1 \xleftarrow{\$} G_1$ and $h \xleftarrow{\$} H$, $h_1 \xleftarrow{\$} H_1$, $a, b_0 \xleftarrow{\$} \mathbb{Z}_n$ and sets $A_G = g^a$, $A_H = h^a$, $B_G = g^{b_0}$, $B_H = h^{b_0}$ and $A_{G,1} = g_1^a$. The authority also chooses $\mathfrak{U}_0, \mathfrak{U}_1, \ldots, \mathfrak{U}_k$ as the Waters hash generators from G. Let $\mathcal{H} : \{0, 1\}^* \to \{0, 1\}^k$ be a collision resistant hash function (CRHF). Finally, the authority publishes the system parameters

$$(G, H, G_T, e, g, h, g_1, h_1, \{\mathfrak{U}_j\}_{j=0}^k, \mathcal{H}, A_{G,1}, A_G, B_G, A_H, B_H).$$

Anyone can check that $(A_G, A_{G,1}, A_H)$ is properly formed by checking

$$e(A_G, h) \stackrel{?}{=} e(g, A_H) \quad \text{and} \quad e(A_{G,1}, h) \stackrel{?}{=} e(g_1, A_H).$$

A similar check can be performed for B_G and B_H.

RS-KeyGen(*PP*). The user chooses $b \xleftarrow{\$} \mathbb{Z}_n$ and computes $pk_G = g^b$ and $sk_G = A_G^b$ in G and $pk_H = h^b$ in H. The user sets $pk = (pk_G, pk_H)$ and $sk = sk_G$.

RS-Sign(pk, sk, R, M). The signer first ensures that there is no repetition in R and $pk \in R$. Let $l = |R|$; the signer parses R as $(pk_{G,i}, pk_{H,i}) \in G \times H$, $i \in [1, l]$. Let i^* be such that $(pk_{G,i^*}, pk_{H,i^*}) = pk$. Now for each $i \in [1, l]$ define f_i such that $f_{i^*} = 1$ and $f_i = 0$ for all $i \neq i^*$. The signer then computes $(m_1, \ldots, m_k) \leftarrow \mathcal{H}(M, R)$. For each $i \in [1, l]$, she/he chooses $t_i, s_i, r_{i,1}, r_{i,2} \xleftarrow{\$} \mathbb{Z}_n$ and computes

$$C_{G,i} = (pk_{G,i}/B_G)^{f_i} g_1^{t_i}, \quad C_{H,i} = (pk_{H,i}/B_H)^{f_i} h_1^{s_i} \tag{4.1}$$

and

$$\begin{aligned} \Theta_{G,i,1} &= (pk_{G,i}/B_G)^{(f_i-1)s_i} g_1^{r_{i,1}}, \quad & \Theta_{H,i,1} &= ((pk_{H,i}/B_H)^{f_i} h_1^{s_i})^{t_i} h_1^{-r_{i,1}}, \\ \Theta_{G,i,2} &= (pk_{G,i}/B_G)^{f_i s_i} g_1^{r_{i,2}}, & \Theta_{H,i,2} &= ((pk_{H,i}/B_H)^{f_i-1} h_1^{s_i})^{t_i} h_1^{-r_{i,2}}. \end{aligned} \tag{4.2}$$

Finally, the signer chooses $r \xleftarrow{\$} \mathbb{Z}_n$ and computes

$$S_{G,1} = sk\left(\mathcal{U}_0 \prod_{j=1}^{k} \mathcal{U}_j^{m_j}\right)^r A_{G,1}^t, \quad S_{G,2} = g^r \quad \text{and} \quad S_{H,2} = h^r,$$

where $t = \sum_{i=1}^{l} t_i$. The output is

$$\sigma = (\{C_{G,i}, C_{H,i}, \Theta_{G,i,1}, \Theta_{G,i,2}, \Theta_{H,i,1}, \Theta_{H,i,2}\}_{i=1}^{l}, S_{G,1}, S_{G,2}, S_{H,2}).$$

RS-Verify(R, M, σ). Let $l = |R|$; parse R as $(pk_{G,i}, pk_{H,i}) \in G \times H$, $i \in [1, l]$. Ensure that there is no repetitions in R. Compute $(m_1, \ldots, m_k) \leftarrow \mathcal{H}(M, R)$. Parse the signature σ as

$$(\{C_{G,i}, C_{H,i}, \Theta_{G,i,1}, \Theta_{G,i,2}, \Theta_{H,i,1}, \Theta_{H,i,2}\}_{i=1}^{l}, S_{G,1}, S_{G,2}, S_{H,2}).$$

For every $i \in [1, l]$ verify that

$$\begin{aligned} e(C_{G,i}/(pk_{G,i}/B_G), C_{H,i}) &\stackrel{?}{=} e(\Theta_{G,i,1}, h_1) e(g_1, \Theta_{H,i,1}), \\ e(C_{G,i}, C_{H,i}/(pk_{H,i}/B_H)) &\stackrel{?}{=} e(\Theta_{G,i,2}, h_1) e(g_1, \Theta_{H,i,2}). \end{aligned}$$

If any of the above verification fails, output reject. Otherwise, compute $C_G = \prod_{i=1}^{l} C_{G,i}$ and verify that

$$e(B_G C_G, A_H) \stackrel{?}{=} e(S_{G,1}, h) e\left(\mathcal{U}_0 \prod_{j=1}^{k} \mathcal{U}_j^{m_j}, S_{H,2}^{-1}\right) \quad \text{and} \quad e(S_{G,2}, h) \stackrel{?}{=} e(g, S_{H,2}). \tag{4.3}$$

If any of the above verification fails, output "reject"; otherwise output "accept".

The ring signature basically consists of the Waters signature under the actual signer's public key pk and the Groth–Sahai NIWI proof components which convince a verifier that one of the signing keys corresponding to the ring of public keys R is used to produce the signature. In the asymmetric setting one can use a variant of the Waters signature (termed as Waters-3b in [12]). The correctness of the scheme can easily be verified from equation (4.3). Observe that neither projection nor cancelling property is used in the construction.

4.2.2 Necessity of cancelling in composite order

Recall that in the original construction [35], $G = H$ and is of order $n = pq$. The subgroup G_1 (resp. G_2) is the order-p subgroup G_p (resp. order q subgroup G_q). Only the unforgeability proof of the Shacham–Waters ring signature needs the cancelling property along with projecting. The proof basically considers two complementary events (termed as Type II and Type III forgery). In the former the adversary outputs a forgery for which either more than one or no member of the ring signed the message and in the latter only one member signed the message.

In the Type II case, the reduction solves an instance of the CDH problem, say $g_2, g_2^{\alpha}, g_2^{\beta} \in G_q$. The simulator chooses a random r_2 from $\mathbb{Z}_p$ and sets $A_G = g_1^{r_2} g_2^{\alpha}$, where g_1 is the generator of G_p. As the simulator has no way to compute g_1^{α}, she/he sets $A_{G,1} = g_1^{r_2/r_1}$, $g = g_1^{r_1} g_2$ for some random r_1 from $\mathbb{Z}_p$. Thus, unlike in the scheme description, A_G and $A_{G,1}$ do not share a common exponent in the simulation. However, the can-

celling property will ensure the well-formedness of A_G and $A_{G,1}$.[2] The simulator embeds g_2^{β} to construct B_G and generates the key pairs for each member of the ring. So she/he can answer all the corruption queries and signatures are generated as in the construction. Once the forger outputs a valid forgery, the simulator uses the projection map to retrieve the CDH solution $g_2^{\alpha\beta}$ in G_q.

For a Type III forgery, the reduction produces a forgery of the Waters signature defined in G_q. Given the Waters public key components $g_2^{\alpha}, g_2^{\beta}$ from G_q, the simulator uses g_2^{α} to construct A_G, as in the case of Type II forgery above, and also $A_{G,1}$. Again, the cancelling property is used to ensure that $A_G, A_{G,1}$ are well-formed. The simulator suitably combines g_2^{β} with some random G_p element to construct the public key of the target user in G. For all other users, key-pairs are generated as in the scheme construction, so both corruption and signing queries can easily be answered for these users. For any signing query made on the target user, the simulator passes the same to the Waters signature challenger. Given the signature with components in G_q, the simulator cannot form a proper signature in G, because she/he cannot compute g_1^{α}. However, she/he can manipulate the ring signature in such a way that it passes the verification in equation (4.3). Once again the cancelling property is crucial for this pairing based verification to go through. Finally, given a valid ring signature forgery, the simulator uses the projection map to retrieve the Waters signature forgery in G_q.

To summarize, in both Type II and Type III forgeries, the cancelling property seems necessary in the composite order setting to establish well-formedness of A_G, $A_{G,1}$, and in Type III to pass the signature verification. The projecting property, on the other hand, is used in both cases to extract the desired solution from the forged ring signature.

4.3 Ring signature in prime order

We show that the cancelling property is not necessary to argue the security in the prime order setting. The crux of the matter is the following. In the composite order setting the unforgeability proof is based on the hardness of CDH in G_q and the simulator is *provided* with $G_1 = G_p$, $G_2 = G_q$. In contrast, in the prime order setting one can start with a hard problem in the prime order groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and then the simulator can appropriately *construct* the subgroups G_1 and G_2. This, however, requires a minor modification in the original definition of projection because the subgroups G_2 and H_2 were not explicitly defined in [18].

Definition 4. Let $\mathcal{G}$ be a bilinear group generator (Definition 1). Then $\mathcal{G}$ is said to be projecting with source group decomposition if it additionally outputs subgroups G_2, H_2 and $G_{T,1}, G_{T,2}$ and non-trivial homomorphisms π_G, π_H and π_T defined on G, H and G_T to themselves such that

- $G \cong G_1 \oplus G_2$, $H \cong H_1 \oplus H_2$,
- $G_1 \subseteq \mathrm{Ker}(\pi_G)$, $H_1 \subseteq \mathrm{Ker}(\pi_H)$ and $G_{T,1} \subseteq \mathrm{Ker}(\pi_T)$,
- $e(\pi_G(g), \pi_H(h)) = \pi_T(e(g, h))$, for all $g \in G$ and $h \in H$.

In the definition above, we allow a complete decomposition of the underlying source groups. Further, instead of defining projection maps for each decomposition (as in [28, Definition 2.3]), we define maps whose kernel contains masking subgroups. Using the proof idea of [18, Theorem 2.5], one can show that the above bilinear group generator $\mathcal{G}$ satisfies the (n, k)-subgroup decisional assumption if the k-linear assumption holds in the atomic group. In particular, the additional output, such as the description of G_2, H_2 and $G_{T,2}$, does not affect the security of the bilinear group generator $\mathcal{G}$.

4.3.1 Freeman framework

It is easy to check that the Freeman construction of Section 3.1.1 satisfies Definition 4. The ring signature scheme described in Section 4.2.1 can be instantiated in this framework without any modification. The

2 Well-formedness is checked by verifying $e(A_G, g_1) \stackrel{?}{=} e(A_{G,1}, g)$. Further, as noted in [35], the adversary does not know g_2 and hence cannot distinguish A_G from $g^{r_2/r_1} = g_1^{r_2} g_2^{r_2/r_1}$.

anonymity proof does not require projecting or cancelling and essentially follows [35]. Here we provide an intuitive justification of why the unforgeability proof requires only the projection property. See Appendix C for the security argument.

Consider, for example, the case of Type II forger. The proof strategy is similar to the original scheme. The simulator chooses key pairs for each user in the system and thus can answer for both signing and corruption queries. After a polynomial number of signing and corruption queries, the adversary outputs a valid ring signature forgery. By applying the projection map on this forgery simulator obtains the solution for the co-DHP+ problem in the subgroups G_2.

Recall that the only place where cancelling was used in [35] is to ensure that A_G and $A_{G,1}$ are correctly formed. In the prime order setting, the simulator is given the co-DHP+ problem instance $\mathfrak{g}$, $\mathfrak{h}$, $\mathfrak{g}^\alpha$, $\mathfrak{g}^\beta$, $\mathfrak{h}^\alpha$, $\mathfrak{h}^\beta$ in $\mathbb{G}_1$ and $\mathbb{G}_2$. As the simulator runs the bilinear group generator in the Freeman setting, we observe that she/he has all the necessary information to compute $A_G = g^\alpha$, $A_{G,1} = g_1^\alpha$ and $A_H = h^\alpha$, which are properly formed.

Finally, the Type II forger outputs a ring signature forgery which contains a Waters-3b signature in $G^2 \times H$ from which the simulator has to extract $\mathfrak{g}^{\alpha\beta}$. After ensuring that this is a valid forgery, she/he applies the appropriate projection map on the first components of the signature to obtain an element of G_2. One can show that this will be of the form $g_2^{\alpha\beta}$, where $g_2 = \mathfrak{g}^{(x_{21}, x_{22})}$ for some $x_{21}, x_{22} \in \mathbb{Z}_p$, which the simulator picked while running the bilinear group generator during the system setup.

A similar argument shows that in the Type III case also we can avoid the cancelling property by leveraging a complete decomposition of the source groups as in Definition 4.

4.3.2 Seo–Cheon framework

One can use the Seo–Cheon construction of Section 4.1 to instantiate the Shacham–Waters ring signature. Since the setting satisfies both the projecting and cancelling properties, it is relatively straightforward to adapt the original security argument of [35].

Note that the Seo–Cheon setting also satisfies Definition 4. Let $\mathcal{G}'_{PC}$ be the corresponding bilinear group generator in Seo–Cheon setting. On input the security parameter 1^λ, $\mathcal{G}'_{PC}$ outputs

$$(G, H, G_T, e, \{G_i, H_i, G_{T,i}\}_{i=1}^2, \pi_G, \pi_H, \pi_T).$$

The construction of $\mathcal{G}'_{PC}$ is similar to the construction of $\mathcal{G}_{PC}$ (see Algorithm 1) with $\pi_G := \pi_{G,2}$, $\pi_H := \pi_{H,2}$ and $\pi_T := \pi_{T,2}$. The only difference with $\mathcal{G}_{PC}$ is that $\mathcal{G}'_{PC}$ does not define the projection maps $\pi_{G,1}$, $\pi_{H,1}$ and $\pi_{T,1}$ and hence it does not output the description of these maps.

Thus in the Seo–Cheon setting the security argument for unforgeability will go through without using the cancelling property. Anonymity can be established under the $(2, 1)$-SDP assumption in G and H. Both security arguments are similar to the Freeman framework instantiation and hence omitted.

4.3.3 Comparison

The size of the source groups (and hence the corresponding group operations) will be double in the Seo–Cheon setting as compared to Freeman's setting. Hence public key size, signature size and time taken by the signing algorithm improve by a factor of two in Freeman's instantiation as compared to the Seo–Cheon instantiation. The verification time is also slightly better in Freeman's instantiation as compared to the Seo–Cheon instantiation.

4.4 Group signature

Group signature provides a mechanism for any member of a group to sign on a message without revealing the signer's identity. Unlike ring signature, group signature requires a distinguished entity called group man-

ager to enroll new members, revoke an existing member or to trace a signature to a particular entity. See Bellare, Micciancio, and Warinschi [4] for a formal definition and security properties of group signature such as full-anonymity and full-traceability. Informally speaking full-anonymity means an adversary cannot obtain signer's identity even if she/he is given a secret key of all the users in the system. Full-traceability, on the other hand, ensures that an adversary is unable to produce a valid forgery which cannot be traced to one of the users in the group.

Boyen and Waters used the symmetric composite order pairing setting to propose two group signatures – one with signature size logarithmic in the number of members [10] and the other with constant size signature [11]. Both schemes are constructed using a similar strategy. For example, the constant size group signature is obtained by suitably composing a constant size NIWI proof system with a two-level hierarchical signature scheme. The Boneh–Boyen signature [6] is used at the first level to generate the private key of the group members. The second level is the Waters signature [37] which corresponds to signature by one of the users in the group.

Group signatures in [10, 11] use the projection map to trace the signer if necessary. Full-anonymity is proved under the hardness of the SDP problem, which requires neither the projecting nor the cancelling property. Full-traceability, on the other hand, requires both projecting and cancelling properties. In particular, the cancelling property is used to check the well-formedness of the private key of the users (at first level) and the group signature (at second level) and the projection map is used to obtain the Waters signature forgery from group signature forgery.

We have converted both schemes and their security arguments to the asymmetric prime order setting. The strategy is quite similar to the ring signature conversion described above. For example, in the constant size group signature [11], the converted scheme uses an asymmetric variant of a constant size NIWI proof system and a two-level hierarchical signature scheme. A group signature consists of a (two level) signature Σ under the signer's identifier s_{ID} and an NIWI proof components Ω. The signer identity can be revealed by applying the appropriate projection map on a component of Σ.

Full-anonymity can be directly proved under the hardness of the SDP problem as similar to the original scheme [11]. Full-traceability is proved under the unforgeability of a two-level hierarchical signature scheme, which is defined in the atomic groups $\mathbb{G}_1$ and $\mathbb{G}_2$. As in the case of ring signature in Section 4.2, all the subgroup generators are constructed by the simulator and hence she/he can translate the hierarchical signature defined in $\mathbb{G}_1$ and $\mathbb{G}_2$ to the group signature defined in 2-fold groups G and H. Also simulator can compute the projection maps, as she/he knows all subgroup generator exponents. This helps the simulator to retrieve the users identity and convert the group signature forgery defined in G and H to two-level hierarchical signature forgery defined in $\mathbb{G}_1$ and $\mathbb{G}_2$. The well-formedness of the group signature and private key of the users are ensured without using cancelling property.

5 Round optimal blind signature

This section revisits the round optimal blind signature (ROBS) cryptosystem of Meiklejohn, Shacham and Freeman [30]. Recall that the scheme was proposed as a concrete example whose security argument requires both projecting and cancelling properties. Later Seo and Cheon [34] provided an alternative proof in the symmetric prime order setting which avoided the cancelling property but made use of the so-called translating property, also introduced in [34].

We investigate the question of secure and efficient instantiation of the scheme in asymmetric prime order setting. We observe that a Freeman-type projecting setting satisfying Definition 4 is sufficient for the security reduction. The translating property introduced in [34] specifically for this purpose does not require a separate treatment as it is trivially achieved in such a setting.

We also provide a Seo–Cheon setting instantiation and perform a detailed efficiency comparison in the two settings. One may expect, like the cryptosystems discussed in Section 4, that a projecting and cancelling instantiation of ROBS is unlikely to outperform a projecting only setting. However, we show (in Table 5) that

while signature size and verification time are comparatively less in the projecting setting, key generation and signing (both user and signer side computations) can be performed more efficiently in the Seo–Cheon setting. This is because, unlike the Seo–Cheon projecting and cancelling setting (see Section 4.1), the SXDH based instantiation of the projecting setting (see Section 3.1.1) seems not to be sufficient for the security argument to go through. We, thus, introduce an *unbalanced* Freeman-type projecting setting to instantiate the scheme, albeit with some additional cost.

Thus our concrete analysis brings one back to the projecting and cancelling setting for the ROBS cryptosystem. But the reason now is efficiency and not functionality/security as envisaged in [30].

5.1 An unbalanced projecting setting

Ghadafi, Smart and Warinschi [20] discussed (and attributed to Groth) an instantiation of the Groth–Sahai proof system in the Type 2 pairing setting [19]. The construction uses DDH in $\mathbb{G}_1$ and DLIN in $\mathbb{G}_2$. Applying the technique suggested by Chatterjee and Menezes [13, 14], we convert this NIWI proof system to the Type 3 pairing setting. This gives the unbalanced projecting bilinear group generator in the asymmetric pairing setting detailed in Algorithm 2. We argue the security of this construction in Theorem 6 of Appendix D.

Algorithm 2. Bilinear group generator $\mathcal{G}_{\mathrm{UP}}$ in the unbalanced projecting setting.

Input: The security parameter 1^λ.

Output: $(G, H, G_T, e, G_1, \mathbb{H}_1, \mathbb{H}_2, G_T', \pi_G, \pi_H, \pi_T)$.

1 The atomic, prime order pairing is generated by running $\mathcal{P}(1^\lambda) \to (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, \mathfrak{g}, \mathfrak{h})$, where $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map and $\mathbb{G}_1 = \langle \mathfrak{g} \rangle$, $\mathbb{G}_2 = \langle \mathfrak{h} \rangle$ and $\mathbb{H} = \langle (\mathfrak{g}, \mathfrak{h}) \rangle$.

2 Choose linearly independent vectors $\vec{x}_1, \vec{x}_2 \xleftarrow{\$} \mathbb{Z}_p^2$ and $\vec{y}_1, \vec{y}_2, \vec{y}_3 \xleftarrow{\$} \mathbb{Z}_p^3$. Define the subgroups $G_i = \langle \mathfrak{g}^{\vec{x}_i} \rangle$ of $\mathbb{G}_1^2$ and $\mathbb{H}_j = \check{H}_j \times H_j$ of $\mathbb{H}^3$, where $\check{H}_j = \langle \mathfrak{g}^{\vec{y}_j} \rangle$ is a subgroup of $\mathbb{G}_1^3$ and $H_j = \langle \mathfrak{h}^{\vec{y}_j} \rangle$ is a subgroup of $\mathbb{G}_2^3$ for all $i \in [1, 2]$ and $j \in [1, 3]$.

3 Define $G := G_1 \oplus G_2 \cong \mathbb{G}_1^2$, $H := H_1 \oplus H_2 \oplus H_3 \cong \mathbb{G}_2^3$ and $G_T := \mathbb{G}_T^6$. The bilinear pairing $e : G \times H \to G_T$ is defined for any $(\mathfrak{g}_1, \mathfrak{g}_2) \in G$ and $(\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3) \in H$ as

$$e((\mathfrak{g}_1, \mathfrak{g}_2), (\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3)) := (\hat{e}(\mathfrak{g}_1, \mathfrak{h}_1), \hat{e}(\mathfrak{g}_1, \mathfrak{h}_2), \hat{e}(\mathfrak{g}_1, \mathfrak{h}_3), \hat{e}(\mathfrak{g}_2, \mathfrak{h}_1), \hat{e}(\mathfrak{g}_2, \mathfrak{h}_2), \hat{e}(\mathfrak{g}_2, \mathfrak{h}_3)).$$

4 Let M be a matrix of order 2 whose i-th row is $\vec{x}_i$ and U be a matrix of order 2, whose $(2, 2)$-th entry is 1 and all other entries are zero. The projection map $\pi_G : G \to G$ is defined as $\pi_G(\mathfrak{g}_1, \mathfrak{g}_2) = (\mathfrak{g}_1, \mathfrak{g}_2)^{M^{-1}UM}$ for $(\mathfrak{g}_1, \mathfrak{g}_2) \in G$. Let N be a matrix of order 3 whose j-th row is $\vec{y}_j$ and let V be a matrix of order 3 whose $(3, 3)$-th entry is 1 and all other entries are zero. The projection map $\pi_H : H \to H$ is defined as

$$\pi_H(\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3) = (\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3)^{N^{-1}VN}$$

for $(\mathfrak{h}_1, \mathfrak{h}_2, \mathfrak{h}_3) \in H$.

5 Let us denote $g_1 = \mathfrak{g}^{\alpha \vec{x}_1}$ and $h_i = \mathfrak{h}^{\alpha_i \vec{y}_i}$ for $\alpha, \alpha_i \in \mathbb{Z}_p^*$ with $i \in [1, 2]$. Define the subgroup $G_T' \subseteq G_T$, where $G_T' = \langle e(g_1, h), e(g, h_1), e(g, h_2) \rangle$ for $g \xleftarrow{\$} G$ and $h \xleftarrow{\$} H$. The projection map $\pi_T : G_T \to G_T$ is defined as

$$\pi_T(\mathcal{T}_1, \ldots, \mathcal{T}_6) = (\mathcal{T}_1, \ldots, \mathcal{T}_6)^{(M^{-1}UM) \otimes (N^{-1}VN)}$$

for $(\mathcal{T}_1, \ldots, \mathcal{T}_6) \in G_T$.

5.2 Prime order instantiation

We give a unified description of the blind signature scheme in the asymmetric pairing setting. The concrete structure of common reference string (CRS), NIWI commitment and proofs as well as their verification will depend upon the choice of bilinear group generator. Table 2 (resp. Table 3) provides the corresponding details for the Seo–Cheon (resp. projecting) framework.

Setup(1^λ). First the atomic bilinear group generator is run. Let $\mathfrak{g}$ and $\mathfrak{h}$ be the generator of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Then either the group generator for the unbalanced projecting or the Seo–Cheon framework is executed. Based on the output of the corresponding group generator, the authority prepares the CRS. The output of the group generator and details of CRS computation are in Table 3 (resp. Table 2) for the projecting (resp. Seo–Cheon) framework.

KeyGen(CRS). Choose $g' \xleftarrow{\$} G$ and define the public key $\mathrm{PK} := e(g', h) = A$ and the secret key $\mathrm{SK} := g'$. Then output (PK, SK).

User(CRS, PK, info, *M*). Let $\mathrm{info} = b_1 \dots b_{m_0}$ be a bit string of length m_0 and let $M = b_{m_0+1} \dots b_m$ be a bit string of length $(m - m_0)$. For each $i \in [m_0 + 1, m]$, the user constructs req, which comprises of the NIWI commitments $(C_{G,i}, C_{H,i})$ and corresponding proofs. (See Table 3 (resp. Table 2) for the projecting (resp. Seo–Cheon) framework.) Let t_i be the random choice of the user in the construction of $C_{G,i}$ which are saved as $\mathrm{state} = \{t_i\}_{i\in[m_0+1,m]}$. The user sends req to the signer.

Signer(CRS, SK, info, req). The signer parses the req and info and, for each $i \in [m_0 + 1, m]$, verifies that $C_{G,i}$ and $C_{H,i}$ are commitments of $b_i = 0$ or $b_i = 1$. (See the details of the proof verification in Table 3 (resp. Table 2) for the projecting (resp. Seo–Cheon) framework.) If the above checks fail, the signer aborts and outputs $\perp$. Otherwise, the signer chooses $r \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$C_G := \mathcal{U}_{G,0}\Big(\prod_{i\in[1,m_0]} \mathcal{U}_{G,i}^{b_i}\Big)\Big(\prod_{i\in[m_0+1,m]} C_{G,i}\Big)$$

and

$$K_{G,1} := g' C_G^r, \quad K_{G,2} := g^{-r}, \quad K_{H,2} := h^{-r}, \quad K_{G,3} := g_1^{-r}.$$

The signer then sends the blinded signature

$$\mathrm{BSig} = (K_{G,1}, K_{G,2}, K_{H,2}, K_{G,3})$$

to the user and outputs *success* and *info*.

User(state, $(K_{G,1}, K_{G,2}, K_{H,2}, K_{G,3})$). The user verifies that

$$e(K_{G,2}, h) \stackrel{?}{=} e(g, K_{H,2}) \quad \text{and} \quad e(K_{G,3}, h) \stackrel{?}{=} e(g_1, K_{H,2}).$$

If any of the above equations fail to hold, the user aborts and outputs $\perp$. Otherwise, the user unblinds the signature by computing

$$S'_{G,1} := K_{G,1}\Big(\prod_{i\in[m_0+1,m]} K_{G,3}^{t_i}\Big), \quad S'_{G,2} := K_{G,2} \quad \text{and} \quad S'_{H,2} := K_{H,2}.$$

The user then checks the validity of the signature $(S'_{G,1}, S'_{G,2}, S'_{H,2})$ by running the **Verify** algorithm as described below. If the output is *reject*, then the user aborts and outputs $\perp$. Otherwise, the user re-randomizes the signature by choosing $z \xleftarrow{\$} \mathbb{Z}_p$ and computing

$$S_{G,1} := S'_{G,1}\Big(\mathcal{U}_{G,0}\prod_{i\in[1,m]} \mathcal{U}_{G,i}^{b_i}\Big)^z, \quad S_{G,2} := S'_{G,2} g^{-z} \quad \text{and} \quad S_{H,2} := S'_{H,2} h^{-z}.$$

Finally, the user outputs $\sigma = (S_{G,1}, S_{G,2}, S_{H,2})$, info and success.

Verify(CRS, PK, info, *M*, σ). The verifier parses info, M and σ and checks that

$$e(S_{G,1}, h)\, e\Big(\mathcal{U}_{G,0}\prod_{i\in[1,m]} \mathcal{U}_{G,i}^{b_i}, S_{H,2}\Big) \stackrel{?}{=} A \quad \text{and} \quad e(S_{G,2}, h) \stackrel{?}{=} e(g, S_{H,2}).$$

If the above two equalities hold, then outputs *accept*, otherwise *reject*.

$\mathcal{G}_{\mathbf{PC}}(1^\lambda)$	$(G, H, G_T, e, \{G_i, H_i, G_{T,i}, \pi_{G,i}, \pi_{H,i}, \pi_{T,i}\}_{i=1}^2, \Omega)$
CRS	The authority chooses $g \xleftarrow{\$} G$, $h \xleftarrow{\$} H$ and chooses $g_1 \xleftarrow{\$} G_1$ and $h_1 \xleftarrow{\$} H_1$. It chooses $\mathcal{U}_{G,i} \xleftarrow{\$} G$ for $i \in [0, m]$. It also chooses $\bar{v}_j \xleftarrow{\$} \mathbb{Z}_p^4$ and computes $\mathcal{V}_{H,j} := \mathfrak{h}^{\bar{v}_j}$ for $j \in [1, m]$, then $\mathsf{CRS}_{\mathsf{SC}} = (G, H, G_T, e, g, h, g_1, h_1, \mathcal{U}_{G,0}, \{\mathcal{U}_{G,i}, \mathcal{V}_{H,i}\}_{i\in[1,m]})$.
Commitment, NIWI proofs	$C_{G,i} := \mathcal{U}_{G,i}^{b_i} g_1^{t_i}, \quad C_{H,i} := \mathcal{V}_{H,i}^{b_i} h_1^{s_i}$ $\Theta_{G,i,1} := \mathcal{U}_{G,i}^{(b_i-1)s_i} g_1^{r_{i,1}}, \quad \Theta_{H,i,1} := (\mathcal{V}_{H,i}^{b_i} h_1^{s_i})^{t_i} h_1^{-r_{i,1}}$ $\Theta_{G,i,2} := \mathcal{U}_{G,i}^{b_i s_i} g_1^{r_{i,2}}, \quad \Theta_{H,i,2} := (\mathcal{V}_{H,i}^{b_i-1} h_1^{s_i})^{t_i} h_1^{-r_{i,2}}$ for $i \in [m_0 + 1, m]$
Verification	$e(C_{G,i}/\mathcal{U}_{G,i}, C_{H,i}) \stackrel{?}{=} e(\Theta_{G,i,1}, h_1)\, e(g_1, \Theta_{H,i,1})$ $e(C_{G,i}, C_{H,i}/\mathcal{V}_{H,i}) \stackrel{?}{=} e(\Theta_{G,i,2}, h_1)\, e(g_1, \Theta_{H,i,2})$ for $i \in [m_0 + 1, m]$

Table 2: Instantiation specifics in the Seo–Cheon setting.

$\mathcal{G}_{\mathbf{UP}}(1^\lambda)$	$(G, H, G_T, e, G_1, \mathbb{H}_1, \mathbb{H}_2, G_T', \pi_G, \pi_H, \pi_T)$
CRS	The authority chooses $g \xleftarrow{\$} G$, $h \xleftarrow{\$} H$ and chooses $g_1 \xleftarrow{\$} G_1$, $\mathbf{h}_{\mathbb{H},1} \xleftarrow{\$} \mathbb{H}_1$, $\mathbf{h}_{\mathbb{H},2} \xleftarrow{\$} \mathbb{H}_2$. It then chooses $\mathcal{U}_{G,i} \xleftarrow{\$} G$, $i \in [0, m]$. It also chooses elements $\bar{v}_j \xleftarrow{\$} \mathbb{Z}_p^3$ and computes $\mathcal{V}_{\mathbb{H},j} = (\mathcal{V}_{\mathbb{G}_1,j}, \mathcal{V}_{H,j}) := (\mathfrak{g}^{\bar{v}_j}, \mathfrak{h}^{\bar{v}_j})$ for $j \in [1, m]$, then CRS is given by $\mathsf{CRS}_{\mathsf{FUP}} = (G, H, G_T, e, g, h, g_1, \mathbf{h}_{\mathbb{H},1}, \mathbf{h}_{\mathbb{H},2}, \mathcal{U}_{G,0}, \{\mathcal{U}_{G,i}, \mathcal{V}_{\mathbb{H},i}\}_{i=1}^m)$.
Commitment, NIWI proofs	$C_{G,i} := \mathcal{U}_{G,i}^{b_i} g_1^{t_i}, \quad C_{H,i} := \mathcal{V}_{H,i}^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}}$ $\Theta_{G,i,1} := \mathcal{U}_{G,i}^{(b_i-1)s_{i,1}} g_1^{r_{i,1}}, \quad \Theta_{H,i,1} := (\mathcal{V}_{H,i}^{b_i} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_i} h_1^{-r_{i,1}} h_2^{-r_{i,2}}$ $\Theta_{G,i,2} := \mathcal{U}_{G,i}^{(b_i-1)s_{i,2}} g_1^{r_{i,2}}, \quad \Theta_{H,i,2} := (\mathcal{V}_{H,i}^{b_i-1} h_1^{s_{i,1}} h_2^{s_{i,2}})^{t_i} h_1^{-r_{i,3}} h_2^{-r_{i,4}}$ $\Theta_{G,i,3} := \mathcal{U}_{G,i}^{b_i s_{i,1}} g_1^{r_{i,3}}, \quad \widetilde{\Theta}_{\mathbb{G}_1,i,1} := (\mathcal{V}_{\mathbb{G}_1,i}^{b_i} h_{\mathbb{G}_1,1}^{s_{i,1}} h_{\mathbb{G}_1,2}^{s_{i,2}})^{t_i} h_{\mathbb{G}_1,1}^{-r_{i,1}} h_{\mathbb{G}_1,2}^{-r_{i,2}}$ $\Theta_{G,i,4} := \mathcal{U}_{G,i}^{b_i s_{i,2}} g_1^{r_{i,4}}, \quad \widetilde{\Theta}_{\mathbb{G}_1,i,2} := (\mathcal{V}_{\mathbb{G}_1,i}^{b_i-1} h_{\mathbb{G}_1,1}^{s_{i,1}} h_{\mathbb{G}_1,2}^{s_{i,2}})^{t_i} h_{\mathbb{G}_1,1}^{-r_{i,3}} h_{\mathbb{G}_1,2}^{-r_{i,4}}$ $\bar{\Theta}_{G,i} = (\Theta_{G,i,1}, \ldots, \Theta_{G,i,4}), \quad \bar{\boldsymbol{\Theta}}_{\mathbb{H},i} = (\Theta_{\mathbb{H},i,1}, \Theta_{\mathbb{H},i,2})$ where $\Theta_{\mathbb{H},i,j} = (\widetilde{\Theta}_{\mathbb{G}_1,i,j}, \Theta_{H,i,j})$ for $i \in [m_0 + 1, m]$ and $j \in [1, 2]$
Verification	$e(C_{G,i}/\mathcal{U}_{G,i}, C_{H,i}) \stackrel{?}{=} e(\Theta_{G,i,1}, h_1)\, e(\Theta_{G,i,2}, h_2)\, e(g_1, \Theta_{H,i,1})$ $e(C_{G,i}, C_{H,i}/\mathcal{V}_{H,i}) \stackrel{?}{=} e(\Theta_{G,i,3}, h_1)\, e(\Theta_{G,i,4}, h_2)\, e(g_1, \Theta_{H,i,2})$ $\hat{e}(\mathfrak{g}, (\Theta_{H,i,j})_{\vert k}) \stackrel{?}{=} \hat{e}((\widetilde{\Theta}_{\mathbb{G}_1,i,j})_{\vert k}, \mathfrak{h})$ for $i \in [m_0 + 1, m]$, $j \in [1, 2]$ and $k \in [1, 3]$

Table 3: Instantiation specifics in the unbalanced projecting setting.

5.3 Security arguments

A blind signature scheme must satisfy two security attributes, namely blindness and one-more unforgeability (OMU). Informally speaking, blindness assures that the signer does not learn any information about the underlying message. OMU assures the conservation of signature – an adversary should not be able to generate any additional signature based on the signatures generated. Refer to [30] for the formal definitions.

The blindness proof is relatively straightforward as it depends on the NIWI security and does not explicitly use projecting or cancelling property. The case for unbalanced setting is discussed in Theorem 7 of Appendix D. The unforgeability proof in the Seo–Cheon projecting and cancelling setting follows the line of argument in [30]. Whereas the unforgeability proof in the unbalanced asymmetric projecting setting to a large extent follows the strategy used in [34] with one exception.

For this reduction Seo and Cheon abstracted the *translating* property [34, Definition 9]. We are not aware of any other application of the translating property in composite to prime order conversion literature. In fact, it is easy to see that, unlike projecting or cancelling, there is no composite order analogue of this property. Also, the original definition [34, Definition 9] requires some modification. As stated, an application of $\tau_{i,j}$ followed by $\tau_{j,i}$ will render the CDH problem easy in the subgroup G_1. This issue, however, can be resolved easily if the map is evaluated with respect to the fixed generators as defined below.

Definition 5. Let $\mathcal{G}$ be a bilinear group generator. We say that $\mathcal{G}$ is (i, j)-translating, for $i \neq j$, if there exist computable maps $\tau_{i,j} : G_i \to G_j$ such that $\tau_{i,j} : g_i^a \mapsto g_j^a$, where g_i and g_j are fixed generators of G_i and G_j, respectively, and $\overline{\tau}_{i,j} : H_i \to H_j$ such that $\overline{\tau}_{i,j} : h_i^a \mapsto h_j^a$, where h_i and h_j are fixed generators of H_i and H_j, respectively. In the symmetric case, $G_i = H_i$ for all i and hence $\tau_{i,j} = \overline{\tau}_{i,j}$.

However, as shown in Claim 1, in the projecting setting of Definition 4, the translating property essentially boils down to few group exponentiations. Our security reduction does not use this abstraction.

Claim 1. *For a bilinear group generator satisfying projecting with source group decomposition as in Definition 4, translating maps may be computed easily between any two subgroups of the same group.*

Proof. We give the proof for the Freeman projecting setting with $n = 2$ (see Section 3.1.1). A similar proof can be extended for any projecting setting and any $n > 1$. Recall that the bilinear group generator outputs subgroups $G_1 = \langle \mathfrak{g}^{(a_1,b_1)} \rangle$, $G_2 = \langle \mathfrak{g}^{(c_1,d_1)} \rangle$, $H_1 = \langle \mathfrak{h}^{(c_2,d_2)} \rangle$ and $H_2 = \langle \mathfrak{h}^{(c_2,d_2)} \rangle$, where $\mathfrak{g}$ generates $\mathbb{G}_1$ and $\mathfrak{h}$ generates $\mathbb{G}_2$. Observe that $\mathfrak{g}^{c_1} = \mathfrak{g}^{a_1 \cdot (c_1/a_1)}$ and $\mathfrak{g}^{d_1} = \mathfrak{g}^{b_1 \cdot (d_1/b_1)}$. Note that the bilinear group generator can compute, say, the translating map $\tau_{1,2} : G_1 \to G_2$. Let $(g_1, g_2) \in G_1$ be given. Then $g_1 = \mathfrak{g}^{a_1 \cdot \theta}$ and $g_2 = \mathfrak{g}^{b_1 \cdot \theta}$ for some $\theta \in \mathbb{Z}_p$. Observe that $g_1^{c_1/a_1} = \mathfrak{g}^{c_1 \cdot \theta}$ and $g_2^{d_1/b_1} = \mathfrak{g}^{d_1 \cdot \theta}$ are simple group exponentiations. This is possible because the bilinear group generator knows the exponents (a_1, b_1) and (c_1, d_1), which are the exponents of the respective generator of the subgroups G_1 and G_2. Now, it is easy to see that $\tau_{1,2} : (g_1, g_2) \mapsto (g_1^{c_1/a_1}, g_2^{d_1/b_1})$ is a valid translation map, with generators of G_1 and G_2 as described above. Similarly the above claim can be proved for any pairs of the subgroups of H. □

Now we outline the one-more unforgeability proof of blind signature under the co-DHP* assumption.

Theorem 3. *The blind signature scheme instantiated in the unbalanced projecting setting is one-more unforgeable if $\mathcal{P}$ satisfies the co-DHP* assumption.*

Proof. The simulator $\mathcal{S}$ is given with a co-DHP* instance $\mathfrak{g}$, $\mathfrak{h}$, $\mathfrak{g}^a$, $\mathfrak{h}^a$, $\mathfrak{g}^b$ along with $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$; $\mathcal{S}$ interacts with a forger $\mathcal{F}$ in the one-more unforgeability game as follows. The simulator $\mathcal{S}$ constructs a unbalanced setting as described in Algorithm 2. This involves choosing linearly independent vectors $\{\vec{x}_i\}$ from $\mathbb{Z}_p^2$ and $\{\vec{y}_j\}$ from $\mathbb{Z}_p^3$ and setting $\hat{g}_i := \mathfrak{g}^{\vec{x}_i}$ and $\hat{\mathbf{h}}_{\mathbb{H},j} = (\hat{h}_{\mathbb{G}_1,j}, \hat{h}_j) = (\mathfrak{g}^{\vec{y}_j}, \mathfrak{h}^{\vec{y}_j})$ for $i \in [1, 2]$, $j \in [1, 3]$. The groups G, H and G_T, subgroups $G_i, H_j, \check{H}_j$ and the pairing maps are defined as in the unbalanced setting. Also, $\mathcal{S}$ defines $R \in G$ and $S \in H$. The simulator $\mathcal{S}$ chooses $\zeta, \zeta_1, \zeta_2 \xleftarrow{\$} \mathbb{Z}_p$ and defines the random subgroup elements $g_1 := \hat{g}_1^{\zeta}$, $\mathbf{h}_{\mathbb{H},i} = (h_{\mathbb{G}_1,i}, h_i) = (\hat{h}_{\mathbb{G}_1,i}^{\zeta_i}, \hat{h}_i^{\zeta_i})$ for $i \in [1, 2]$.

As in the unforgeability proof of Waters signatures [37], $\mathcal{S}$ defines functions $F(M)$ and $J_j(M)$, $j \in [1, 2]$. Using the co-DHP* problem instance, $\mathcal{S}$ computes for $i \in [1, 2]$ and $j \in [1, 3]$,

$$A_{G,i} := (\mathfrak{g}^a)^{\vec{x}_i} = \hat{g}_i^a, \quad B_{G,i} := (\mathfrak{g}^b)^{\vec{x}_i} = \hat{g}_i^b \quad \text{and} \quad \mathbf{A}_{\mathbb{H},j} = (A_{\mathbb{G}_1,j}, A_{H,j}) = ((\mathfrak{g}^a)^{\vec{y}_j}, (\mathfrak{h}^a)^{\vec{y}_j}) = (\hat{h}_{\mathbb{G}_1,j}^a, \hat{h}_j^a)$$

The simulator $\mathcal{S}$ constructs the Waters hash generators $\mathcal{U}_{G,0}$ and $\mathcal{U}_{G,i}$, $\mathcal{V}_{\mathbb{H},i} = (\mathcal{V}_{\mathbb{G}_1,i}, \mathcal{V}_{H,i})$ for $i \in [1, m]$, chooses $a' \xleftarrow{\$} \mathbb{Z}_p$, computes the public key as $A = e(\hat{g}_1^{a'} B_{G,2}, A_{H,1} A_{H,2} A_{H,3})$ and sends the public parameters along with the public key to $\mathcal{F}$. See Appendix D for details about the computation of these parameters.

The forger $\mathcal{F}$ makes signing queries to $\mathcal{S}$ by sending blinded message in terms of commitment $(C_{G,i}, C_{H,i})$ and NIWI proofs $(\bar{\Theta}_{G,i}, \bar{\mathbf{\Theta}}_{\mathbb{H},i})$. The simulator $\mathcal{S}$ processes the committed message and proof given by $\mathcal{F}$ to infer that they indeed have the intended structure. This can be proved by a variant of [34, Lemma 5] in the asymmetric pairing setting which we state as Lemma 8 in Appendix D.

The simulator $\mathcal{S}$ applies the projection map to extract the message M and then constructs the unblinded signature $(S_{G,1}, S_{G,2}, S_{H,2})$ (see Appendix D for details). From this, $\mathcal{S}$ constructs a blinded signature components $K_{G,2} := S_{G,2}$, $K_{H,2} := S_{H,2}$ and $K_{G,3} := g_1^{-r} A_{G,1}^{\zeta/F(M)}$, where r is the randomizer used in the construction of the Waters signature.

To construct $K_{G,1}$, $\mathcal{S}$ needs to compute $g_1^{t_i}$ and $g_1^{at_i}$ without the knowledge of a and t_i, for $i \in [m_0 + 1, m]$. Since $C_{G,i} = \mathcal{U}_{G,i}^{b_i} g_1^{t_i}$, $\mathcal{S}$ can compute $g_1^{t_i}$ as $C_{G,i} / \mathcal{U}_{G,i}^{b_i}$ by using the knowledge of b_i. Now, $\mathcal{S}$ computes

$$h_{\mathbb{G}_1,3}^{at_i} = A_{\mathbb{G}_1,3}^{t_i} = \begin{cases} (\widetilde{\Theta}_{\mathbb{G}_1,i,1}^{N^{-1}VN})^{1/\bar{w}_{3i}}, & \text{if } b_i = 1, \\ (\widetilde{\Theta}_{\mathbb{G}_1,i,2}^{N^{-1}VN})^{-1/\bar{w}_{3i}}, & \text{if } b_i = 0, \end{cases}$$

where N is a matrix of order 3 whose j-th row is $\vec{y}_j$ and V is a matrix of order 3 whose $(3,3)$-th element is 1 and all other elements are zero. The simulator $\mathcal{S}$ computes $g_1^{at_i}$ as follows:

$$(h_{\mathbb{G}_1,3}^{at_i})^{N^{-1}UM} = (\mathfrak{g}^{at_i\vec{y}_3})^{N^{-1}UM} = (\mathfrak{g}^{at_i(0,0,1)N})^{N^{-1}UM} = \mathfrak{g}^{at_i(0,0,1)UM}$$
$$= \mathfrak{g}^{at_i(1,0)M} = \mathfrak{g}^{at_i\vec{x}_1} = g_1^{at_i}.$$

Here U is a boolean matrix of order 3×2 whose $(3,1)$-th entry is 1 and all other entries are zero. Recall that M is a matrix of order 2 whose i-th row is $\vec{x}_i$. Finally, $\mathcal{S}$ computes

$$K_{G,1} := S_{G,1} \prod_{i=m_0+1}^{m} (g_1^{t_i})^r (g_1^{at_i})^{-\frac{1}{F(M)}}.$$

The well-formedness of the above (blinded) signature can be verified using bilinear map, which does not require the cancelling property.

The simulator $\mathcal{S}$ stores all the queried messages in a list called L. After receiving the forgery list from $\mathcal{F}$, $\mathcal{S}$ finds at least one message, say M^*, which is not in L. Let $(S_{G,1}^*, S_{G,2}^*, S_{H,2}^*)$ be the corresponding signature on M^*. Then $\mathcal{S}$ checks whether $F(M^*) = 0$; if not, $\mathcal{S}$ aborts the game. Otherwise, $\mathcal{S}$ computes

$$\hat{g}_2^{ab} = \pi_G(S_{G,1}^*(S_{G,2}^*)^{J_2(M^*)})$$

and checks that it indeed belongs to the subgroup G_2. Since $\mathcal{S}$ had chosen $\vec{x}_2$, she/he can easily compute $\mathfrak{g}^{ab}$ from $\hat{g}_2^{ab}$ and return as a solution for the co-DHP* problem. □

Remark 3. We briefly comment why we need the DLIN assumption in $\mathbb{H}$ and thus work in the unbalanced setting. The structure of the Groth–Sahai proof system in the asymmetric pairing setting mandates that $\Theta_{H,i,j} \in H$ for $j \in [1,2]$ be provided as part of the NIWI proof. On the other hand, $\mathcal{S}$ needs the corresponding $\tilde{\Theta}_{\mathbb{G}_1,i,j} \in \mathbb{G}_1^3$ to extract $g_1^{at_i}$. This is achieved by providing the NIWI proof components $\Theta_{\mathbb{H},i,j} \in \mathbb{H}$ (see Table 3). Recall that $\mathbb{H} = \langle(\mathfrak{g}, \mathfrak{h})\rangle$, which means the underlying elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ will share the same exponent. In particular, the unknown value (at_i) will be in the exponent of $\mathbb{G}_1$ and $\mathbb{G}_2$. In other words, DDH and hence the subgroup decision problem will be easy in $\mathbb{H}$ and we cannot use the projecting settings of Section 3.1.

5.4 Comparison

We compare the two frameworks in Table 4 based on size of group elements and various operations. In the following, for a group X we use E_X, M_X and I_X to denote respectively group exponentiation, multiplication and inversion in X; $\mathbb{P}$ (resp. P) is used to denote a pairing in atomic (resp. prime power) setting.

Recall that m_0 (resp. k) denotes the bit-length of common information (resp. hash digest of the actual message) whence $m = k + m_0$ is the total message length. The size of various parameters like CRS and σ and the computational complexity of the algorithms are compared in Table 5.

As described in Table 4, the elements of G, H and G_T as well as various operations involving these groups can be described in terms of the atomic groups. Based on the concrete analysis at the 128-bit security level from [12, Table 2], we work out the relative performance of the two frameworks. In [12], all the operations

Framework	Unbalanced	Seo–Cheon
$1\|G\|, 1\|H\|, 1\|G_T\|$	$2\|\mathbb{G}_1\|, 3\|\mathbb{G}_2\|, 6\|\mathbb{G}_T\|$	$4\|\mathbb{G}_1\|, 4\|\mathbb{G}_2\|, 2\|\mathbb{G}_T\|$
$1E_G, 1E_H, 1E_{G_T}$	$2E_{\mathbb{G}_1}, 3E_{\mathbb{G}_2}, 6E_{\mathbb{G}_T}$	$4E_{\mathbb{G}_1}, 4E_{\mathbb{G}_2}, 2E_{\mathbb{G}_T}$
$1M_G, 1M_H, 1M_{G_T}$	$2M_{\mathbb{G}_1}, 3M_{\mathbb{G}_2}, 6M_{\mathbb{G}_T}$	$4M_{\mathbb{G}_1}, 4M_{\mathbb{G}_2}, 2M_{\mathbb{G}_T}$
$1I_G, 1I_H, 1I_{G_T}$	$2I_{\mathbb{G}_1}, 3I_{\mathbb{G}_2}, 6I_{\mathbb{G}_T}$	$4I_{\mathbb{G}_1}, 4I_{\mathbb{G}_2}, 2I_{\mathbb{G}_T}$
$1P$	$6\mathbb{P}$	$4\mathbb{P} + 2M_{\mathbb{G}_T}$

Table 4: Comparison of the unbalanced projecting and Seo–Cheon's projection and cancelling framework.

	(A) Unbalanced framework	(B) Seo–Cheon framework	Ratio (A/B)†
|CRS|	$(5m+12)\|\mathbb{G}_1\| + (3m+9)\|\mathbb{G}_2\|$	$(4m+16)\|\mathbb{G}_1\| + (4m+12)\|\mathbb{G}_2\|$	0.92
|Key|	$2\|\mathbb{G}_1\| + 6\|\mathbb{G}_T\|$	$4\|\mathbb{G}_1\| + 2\|\mathbb{G}_T\|$	2.16
|req|	$16k\|\mathbb{G}_1\| + 9k\|\mathbb{G}_2\|$	$12k(\|\mathbb{G}_1\| + \|\mathbb{G}_2\|)$	0.94
|BSig|	$6\|\mathbb{G}_1\| + 3\|\mathbb{G}_2\|$	$12\|\mathbb{G}_1\| + 4\|\mathbb{G}_2\|$	0.6
$\|\sigma\|$	$4\|\mathbb{G}_1\| + 3\|\mathbb{G}_2\|$	$8\|\mathbb{G}_1\| + 4\|\mathbb{G}_2\|$	0.62
Setup	$(5m+10)E_{\mathbb{G}_1} + (3m+7)E_{\mathbb{G}_2}$	$(4m+16)E_{\mathbb{G}_1} + (4m+12)E_{\mathbb{G}_2}$	0.91
KeyGen	$6\mathbb{P} + 2E_{\mathbb{G}_1}$	$4\mathbb{P} + 2M_{\mathbb{G}_T} + 4E_{\mathbb{G}_1}$	1.4
User	$48\mathbb{P} + 6M_{\mathbb{G}_T} + [34k+4]E_{\mathbb{G}_1}$ $+[28k+4m_0+4]M_{\mathbb{G}_1}$ $+[18k+3](E_{\mathbb{G}_2} + M_{\mathbb{G}_2})$	$40\mathbb{P} + 24M_{\mathbb{G}_T} + [14k+24]E_{\mathbb{G}_1}$ $+[18k+8m_0+16]M_{\mathbb{G}_1}$ $+[10k+12]E_{\mathbb{G}_2} + [6k+4]M_{\mathbb{G}_2}$	1.51
Signer	$52k\mathbb{P} + 24kM_{\mathbb{G}_T} + 6E_{\mathbb{G}_1}$ $+[4k+2m_0+2]M_{\mathbb{G}_1} + 3E_{\mathbb{G}_2}$ $+3kM_{\mathbb{G}_2}$	$[24k+8]\mathbb{P} + [16k+6]M_{\mathbb{G}_T}$ $+[12k+16]E_{\mathbb{G}_1} + [20k+4m_0]M_{\mathbb{G}_1}$ $+[12k+8]E_{\mathbb{G}_2} + [16k-4]M_{\mathbb{G}_2}$	1.88
Verify	$24\mathbb{P} + 6M_{\mathbb{G}_T} + 2mM_{\mathbb{G}_1}$	$24\mathbb{P} + 16M_{\mathbb{G}_T} + 12E_{\mathbb{G}_1} + [4m+4]M_{\mathbb{G}_1} + 8E_{\mathbb{G}_2}$	0.88

Table 5: Comparison of blind signature instantiation in the two frameworks. († The ratio A/B is computed for $m_0 = 100$ and $k = 256$ using [12, Table 2].)

such as group exponentiation and pairing computation are expressed in terms of field multiplications. Thus, using that result, one can easily calculate the ratio A/B for different values of m_0 and k. As expected, the ratio remains more or less the same for various concrete values of the message digest k and common information m_0.

In the above table we have computed the ratio for $m_0 = 100$ and $k = 256$. The target group size is 1024 bits and size of the first and second source groups are 257-bits and 513-bits, respectively. For $m = m_0 + k = 356$ bits, the CRS size for Seo–Cheon and unbalanced settings turn out to be respectively 1,013,045 and 1,106,748 bits and the ratio is approximately 0.92. The size of the other components are calculated in a similar way. Similarly, the running time of various algorithms in the blind signature scheme are obtained using [12, Table 2]. In particular, pairing computation takes 15,175 field multiplications and addition in the first, second and target group respectively take 11, 30 and 54 field multiplications and exponentiation in the first and second source group respectively take 1533 and 3052 field multiplications. Note that User, Singer and Verify algorithms need to perform a group membership test. In the Seo–Cheon setting, group membership is checked using pairing based batch verification (see [34, Appendix E]) under the SXDH assumption. Thus the cost of this check is also incorporated in Table 5. Chatterjee, Hankerson, Knapp and Menezes [12] mentioned that membership test in the second source group takes 3,052 field multiplications, whereas the cost is ignored for the first source group as it takes very few field multiplications. By applying these values in each algorithm (such as Setup, KeyGen, User, Signer and Verify), we obtain the appropriate ratio A/B as described in our Table 5.

6 Concluding remarks

For the projecting property, several frameworks have been proposed in the symmetric pairing setting with each one improving upon the previous proposal. However, in the asymmetric setting, which really matters for efficient instantiation, the Freeman and its predecessor Groth–Sahai construction still remain the best.

We have several interesting observations in the context of a simultaneously projecting and cancelling setting. Ring and group signatures [10, 11, 35] that also require the cancelling property in composite order can be converted to prime order in a projecting alone setting. This is because, unlike in the composite order setting where subgroups are given, a prime order projecting setting allows the simulator to construct the

subgroups. We are not aware of any other cryptosystem that uses both projecting and cancelling properties in the composite order setting, and hence requires a projecting and cancelling framework in the prime order setting for functionality and/or security.

In this context it is worth to briefly comment on two recent works. Herold, Hesse, Hofheinz, Ràfols and Rupp [23] proposed a polynomial interpretation based projecting and cancelling framework in the symmetric setting (but did not propose any concrete application). We extended their projecting and cancelling framework to the asymmetric pairing setting and observed that a pairing computation in their framework will require nine atomic pairing as opposed to four in the Seo–Cheon framework. Lewko and Meiklejohn [28] used their simultaneously parameter hiding, cancelling and projecting framework to instantiate two cryptosystems in the prime order setting – a leakage resilient variant of BGN and an IBE with CCA1 security. However, neither of the two cryptosystems have any composite order counterpart. In fact, it is difficult to conceive of a *natural* counterpart of the two cryptosystems in the composite order setting. Thus the relevance of the above two frameworks in the context of composite-to-prime order conversion is yet to be established.

Finally, our concrete analysis of the blind signature indicates that even though a projecting and cancelling setting may not be necessary for functionality and/or security of a cryptosystem, still that may be a preferred choice from the view point of efficiency.

A Relevant hard problems

We recall the definitions of major hard problems used in this work.

Definition 6. Let $\Psi = (n, \mathbb{G}, g)$ be the output of a group generator $\mathcal{G}(1^\lambda)$. A PPT algorithm $\mathcal{A}$ is said to have advantage $\mathrm{Adv}_{\mathrm{CDH}_\mathbb{G}}[\mathcal{A}, \mathcal{G}]$ in solving the computational Diffie–Hellman problem (CDH), where the advantage is defined as

$$\Pr[\mathcal{A}(\Psi, g^a, g^b) \to g^{ab} : \Psi \xleftarrow{\$} \mathcal{G}; g \xleftarrow{\$} \mathbb{G}; a, b \xleftarrow{\$} \mathbb{Z}_p].$$

Further, $\mathcal{G}$ is said to satisfy the CDH assumption in $\mathbb{G}$ if $\mathrm{Adv}_{\mathrm{CDH}_\mathbb{G}}[\mathcal{A}, \mathcal{P}]$ is negligible in λ for any PPT algorithm $\mathcal{A}$.

Definition 7. Let $\Psi = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be the output of a prime order bilinear group generator $\mathcal{P}(1^\lambda)$. Let $k \geq 1$. An algorithm $\mathcal{A}$ taking $(2k + 2)$ elements from $\mathbb{G}_1$ as inputs has advantage $\mathrm{Adv}_{k\text{-Lin}_{\mathbb{G}_1}}[\mathcal{A}, \mathcal{P}]$ defined to be equal to

$$\begin{aligned}\Big|\Pr[\mathcal{A}(\Psi, g_1, \ldots, g_k, g_1^{r_1} \ldots, g_k^{r_k}, g_0, g_0^{r_1+\cdots+r_k}) = 1 : \Psi \xleftarrow{\$} \mathcal{P}, g_1, \ldots, g_k, g_0 \xleftarrow{\$} \mathbb{G}_1; r_1, \ldots, r_k \xleftarrow{\$} \mathbb{Z}_p] \\ - \Pr[\mathcal{A}(\Psi, g_1, \ldots, g_k, g_1^{r_1} \ldots, g_k^{r_k}, g_0, g_0^{s}) = 1 : \Psi \xleftarrow{\$} \mathcal{P}, g_1, \ldots, g_k, g_0 \xleftarrow{\$} \mathbb{G}_1; r_1, \ldots, r_k, s \xleftarrow{\$} \mathbb{Z}_p]\Big|\end{aligned}$$

in solving the k-linear problem (k-Lin) on $\mathbb{G}_1$. The group generator $\mathcal{P}$ is said to satisfy the k-linear assumption in $\mathbb{G}_1$ if $\mathrm{Adv}_{k\text{-Lin}_{\mathbb{G}_1}}[\mathcal{A}, \mathcal{P}]$ is negligible in λ for any PPT algorithm $\mathcal{A}$ (similar definition in $\mathbb{G}_2$).

The *decisional Diffie–Hellman (DDH) assumption* is the 1-linear assumption. The *decisional linear (DLin) assumption* is the 2-linear assumption. When the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$, the *symmetric external Diffie–Hellman assumption (SXDH)* is said to hold.

Definition 8. Let $\mathcal{G}$ be a bilinear group generator (of non-prime order). Consider the following distribution:

$$\Psi = (G, G_1, H, H_1, G_T, e) \xleftarrow{\$} \mathcal{G}(1^\lambda),\ T_0 \xleftarrow{\$} G,\ T_1 \xleftarrow{\$} G_1.$$

The advantage of a PPT algorithm $\mathcal{A}$ in solving the SDP problem on the left is denoted by $\mathrm{Adv}_{\mathrm{SDP}_L}[\mathcal{A}, \mathcal{G}]$ and defined as

$$|\Pr[\mathcal{A}(\Psi, T_0) = 1] - \Pr[\mathcal{A}(\Psi, T_1) = 1]|.$$

Then $\mathcal{G}$ is said to satisfy the subgroup decision on the left if the above advantage is negligible in λ. A similar definition holds for subgroup decision on the right. Consequently, $\mathcal{G}$ is said to satisfy the subgroup decision assumption if both the corresponding problems on the left and the right are hard.

Further, if G is an n-fold group and the subgroup G_1 is generated using the k-linearly independent group element, then we say that $\mathcal{G}$ satisfies the (n, k)-subgroup decision assumption [18] in G. Similarly we can define in H.

Definition 9. Let $\Psi = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{P}(1^\lambda)$. An algorithm $\mathcal{A}$ has advantage $\mathrm{Adv}_{\text{co-DHP}^*}$ defined to be equal to

$$\Pr[\mathcal{A}(\Psi, g, g^a, g^b, h, h^a) \to g^{ab} : \Psi \xleftarrow{\$} \mathcal{P}, g \xleftarrow{\$} \mathbb{G}_1, h \xleftarrow{\$} \mathbb{G}_2; a, b \xleftarrow{\$} \mathbb{Z}_p]$$

in solving the co-DHP* problem [12] on both $\mathbb{G}_1$ and $\mathbb{G}_2$. Then $\mathcal{P}$ is said to satisfy the co-DHP* assumption if the above advantage is negligible in λ for any $\mathcal{A}$.

Definition 10. Let $\Psi = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{P}(1^\lambda)$. An algorithm $\mathcal{A}$ has advantage $\mathrm{Adv}_{\text{co-DHP+}}$ defined to be equal to

$$\Pr[\mathcal{A}(\Psi, g, g^a, g^b, h, h^a, h^b) \to g^{ab} : \Psi \xleftarrow{\$} \mathcal{P}, g \xleftarrow{\$} \mathbb{G}_1, h \xleftarrow{\$} \mathbb{G}_2; a, b \xleftarrow{\$} \mathbb{Z}_p]$$

in solving the co-DHP+ problem on both $\mathbb{G}_1$ and $\mathbb{G}_2$. Then $\mathcal{P}$ is said to satisfy the co-DHP+ assumption if the above advantage is negligible in λ for any $\mathcal{A}$.

Ghadafi, Smart and Warinschi [20] proposed a variant of the DLIN problem in the asymmetric pairing setting. In the following we will define the problem in terms of $\mathbf{h} \in \mathbb{H}$, where $\mathbb{H} = \langle (g, h) \rangle$.

Definition 11. Let $\Psi = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{P}(1^\lambda)$. An algorithm $\mathcal{A}$ has advantage $\mathrm{Adv}_{\mathrm{DLin}_{\mathbb{H}}}$ defined to be equal to

$$\begin{aligned}\big|&\Pr[\mathcal{A}(\Psi, \mathbf{h}, \mathbf{h}^{z_1}, \mathbf{h}^{z_2}, \mathbf{h}^{z_1 z_3}, \mathbf{h}^{z_2 z_4}, \mathbf{h}^{z_3 + z_4}) = 1 : \Psi \xleftarrow{\$} \mathcal{P}, \mathbf{h} = (g, h) \xleftarrow{\$} \mathbb{H}; z_1, z_2, z_3, z_4 \xleftarrow{\$} \mathbb{Z}_p] \\ &- \Pr[\mathcal{A}(\Psi, \mathbf{h}, \mathbf{h}^{z_1}, \mathbf{h}^{z_2}, \mathbf{h}^{z_1 z_3}, \mathbf{h}^{z_2 z_4}, \mathbf{h}^{z_5}) = 1 : \Psi \xleftarrow{\$} \mathcal{P}, \mathbf{h} = (g, h) \xleftarrow{\$} \mathbb{H}; z_1, z_2, z_3, z_4, z_5 \xleftarrow{\$} \mathbb{Z}_p]\big|\end{aligned}$$

in solving the DLin problem on $\mathbb{H}$. Then $\mathcal{P}$ is said to satisfy the DLin assumption if the above advantage is negligible in λ for any $\mathcal{A}$.

B Symmetric projecting frameworks

While Freeman considered only an asymmetric version of projecting setting, it is easy to adapt his strategy in the symmetric setting as shown, for example, in [34]. In this section, we compare the various symmetric projecting frameworks with an eye on the instantiation of BGN cryptosystem. Our discussion also narrates how researchers have been able to progressively improve the parameters for symmetric projecting frameworks [23, 33, 34].

As we mentioned in Section 2 symmetric pairings over small characteristic fields are effectively broken due to recent advances in solving the DLP in some of the fields. However, those results do not affect symmetric pairings defined over large characteristic fields. Also note that pairing over an embedding degree one curve [15] has been proposed in recent times. This can be a conservative choice as recent advances in the efficiency of NFS and its variants for solving the medium prime discrete log [26] have cast some doubt on the concrete security of asymmetric pairing defined over BN curves [3]. Pairing over embedding degree one as well as embedding degree two curves (for example, [16]) can be used to instantiate cryptosystems that are built on symmetric pairing. Hence, the question of efficient instantiation in the symmetric prime order setting still merits some discussion.

B.1 Groth–Sahai framework

Consider a symmetric prime order bilinear group generator $\mathcal{P}_s(1^\lambda)$ which outputs $(\mathbb{G}, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is the atomic symmetric pairing. Define $G = \mathbb{G}^3$

and $G_T = \mathbb{G}_T^9$. The pairing map $e : G \times G \to G_T$ is defined using the tensor product

$$e(\mathfrak{g}^{\vec{x}}, \mathfrak{g}^{\vec{y}}) = \hat{e}(\mathfrak{g}, \mathfrak{g})^{\frac{1}{2}(\vec{x}\otimes\vec{y})+\frac{1}{2}(\vec{y}\otimes\vec{x})},$$

where $\mathfrak{g} \in \mathbb{G}$, $\vec{x}, \vec{y} \in \mathbb{Z}_p^3$. Note that, the way pairing is defined makes it commutative.

Choose two random elements α, β from $\mathbb{Z}_p^*$ and set $\vec{x}_1 = (\alpha, 0, 1)$ and $\vec{x}_2 = (0, \beta, 1)$. The subgroup G_1 is generated by $\mathfrak{g}^{\vec{x}_1} = (\mathfrak{g}^\alpha, 1, \mathfrak{g})$ and $\mathfrak{g}^{\vec{x}_2} = (1, \mathfrak{g}^\beta, \mathfrak{g})$. The projection map $\pi : G \to \mathbb{G}$ is defined as

$$\pi(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) = \mathfrak{g}_1^{-1/\alpha}\mathfrak{g}_2^{-1/\beta}\mathfrak{g}_3$$

for any $(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) \in G$. Observe that $G_1 = \operatorname{Ker}\pi$. Similarly the subgroup G_T' is generated using $e(\mathfrak{g}^{\vec{x}_1}, \mathfrak{g})$ and $e(\mathfrak{g}^{\vec{x}_2}, \mathfrak{g})$ and the projection map $\pi_T : G_T \to \mathbb{G}_T$ is defined as

$$\pi_T\begin{pmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{pmatrix} = (z_{33}z_{13}^{-1/\alpha}z_{23}^{-1/\beta})(z_{31}z_{11}^{-1/\alpha}z_{21}^{-1/\beta})^{-1/\alpha}(z_{32}z_{12}^{-1/\alpha}z_{22}^{-1/\beta})^{-1/\beta}$$

for any $(z_{11}, \ldots, z_{33}) \in G_T$. Observe that $G_T' = \operatorname{Ker}\pi_T$ and the projection maps π, π_T commute with the pairing. Finally, the bilinear group generator outputs $(G, G_T, e, G_1, G_T', \pi, \pi_T)$.

B.2 Seo framework

In the Groth–Sahai framework, the symmetric property of the pairing leads to six distinct subgroups of G_T even though G_T is a 9-fold of $\mathbb{G}_T$. This was observed by Seo and Cheon [34], who gave a symmetric variant of the Freeman asymmetric framework. In a follow-up work, Seo [33] further optimized the projecting setting by using a 6-fold target group for pairing definition.

The underlying groups G and G_T are as in the Groth–Sahai framework of Section B.1. Let $\{\vec{x}_i\}_{i\in[1,3]}$ be linearly independent vectors from $\mathbb{Z}_p^3$. The subgroup G_1 is generated using $\mathfrak{g}^{\vec{x}_1}$ and $\mathfrak{g}^{\vec{x}_2}$. The projection map $\pi : G \to G$ is defined as

$$\pi(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) = (\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3)^{M^{-1}U_3M},$$

where M is a matrix of order 3 whose i-th row is $\vec{x}_i$ and U_3 is a matrix of order 3 whose $(3, 3)$-th entry is 1 and the remaining entries are zero. Observe that $G_1 \subseteq \operatorname{Ker}\pi$. The bilinear map $e : G \times G \to G_T$ is defined as

$$\begin{aligned} e(\mathfrak{g}^{\vec{x}}, \mathfrak{g}^{\vec{y}}) := (&\hat{e}(\mathfrak{g}^{x_1}, \mathfrak{g}^{y_1}), \hat{e}(\mathfrak{g}^{x_1}, \mathfrak{g}^{y_2})\hat{e}(\mathfrak{g}^{x_2}, \mathfrak{g}^{y_1}), \hat{e}(\mathfrak{g}^{x_2}, \mathfrak{g}^{y_2}), \\ &\hat{e}(\mathfrak{g}^{x_1}, \mathfrak{g}^{y_3})\hat{e}(\mathfrak{g}^{x_3}, \mathfrak{g}^{y_1}), \hat{e}(\mathfrak{g}^{x_2}, \mathfrak{g}^{y_3})\hat{e}(\mathfrak{g}^{x_3}, \mathfrak{g}^{y_2}), \hat{e}(\mathfrak{g}^{x_3}, \mathfrak{g}^{y_3})) \end{aligned} \tag{B.1}$$

for any $\mathfrak{g}^{\vec{x}} = (\mathfrak{g}^{x_1}, \mathfrak{g}^{x_2}, \mathfrak{g}^{x_3})$, $\mathfrak{g}^{\vec{y}} = (\mathfrak{g}^{y_1}, \mathfrak{g}^{y_2}, \mathfrak{g}^{y_3}) \in G$. The above pairing exponent can be viewed as a 6-tuple vector $(\vec{x} \otimes \vec{y})B$, for the boolean matrix B is of order 9×6 (see [33, Example 2]). The projection map $\pi_T : G_T \to G_T$ is defined as

$$\pi_T(\mathcal{T}_1, \ldots, \mathcal{T}_6) := (\mathcal{T}_1, \ldots, \mathcal{T}_6)^{D^{-1}VD}$$

for $(\mathcal{T}_1, \ldots, \mathcal{T}_6) \in G_T$. Here V is a diagonal matrix of order 6 with 1 in the entry $(1, 1)$ and zeros elsewhere. The matrix D is defined as $D = ZB$, where Z is the 6×9 matrix such that its l-th row is $\vec{x}_i \otimes \vec{x}_j$ for $l = \frac{1}{2}(4 - i)(3 - i) + 4 - j$ and $1 \le i \le j \le 3$. The security of this framework is proved under the DLin assumption [33, Theorem 1]. Finally, the bilinear group generator outputs $(G, G_T, e, G_1, \pi, \pi_T)$. Note that the description of G_1 and e suffice to describe G_T', which is not provided explicitly.

B.3 Polynomial framework

In 2014, Herold, Hesse, Hofheinz, Ràfols and Rupp [23] introduced their framework using a polynomial interpretation. This framework can be seen as an optimized version of the Groth–Sahai construction.

Elements of G are represented by the polynomial $f(X_1, X_2) = -X_2 - X_1 + X_1X_2$, where X_1, X_2 are the indeterminates. The pairing map is defined as in equation (B.1) which is interpreted as a product of f with itself. Thus the elements of G_T are represented using the polynomial

$$f_T(X_1, X_2) = X_2^2 + 2X_1X_2 + X_1^2 - 2X_1X_2^2 - 2X_1^2X_2 + X_1^2X_2^2.$$

Two hidden parameters s_1, s_2 are chosen uniformly at random from $\mathbb{Z}_p^*$. The subgroup $G_1^{(s_1,s_2)}$ is defined using the generators $(\mathfrak{g}^{s_1}, 1, \mathfrak{g})$ and $(1, \mathfrak{g}^{s_2}, \mathfrak{g})$ for $\mathfrak{g} \in \mathbb{G}$. The projection map $\pi^{(s_1,s_2)} : G \to \mathbb{G}$ is defined as

$$\pi^{(s_1,s_2)}(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) = \mathfrak{g}_1^{-s_2}\mathfrak{g}_2^{-s_1}\mathfrak{g}_3^{s_1s_2}$$

for $(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) \in G$. The map $\pi^{(s_1,s_2)}$ is interpreted as evaluation of the polynomial f at $X_1 = s_1$ and $X_2 = s_2$. The subgroup $G_T'^{(s_1,s_2)}$ is generated using $e((\mathfrak{g}^{s_1}, 1, \mathfrak{g}), g)$ and $e((1, \mathfrak{g}^{s_2}, \mathfrak{g}), g)$ for fixed $g = (\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) \in G$. The projection map $\pi_T^{(s_1,s_2)} : G_T \to \mathbb{G}_T$ is defined using evaluation of the polynomial f_T at $X_1 = s_1$ and $X_2 = s_2$. Observe that $G_1^{(s_1,s_2)} = \operatorname{Ker}\pi^{(s_1,s_2)}$ and $G_T'^{(s_1,s_2)} = \operatorname{Ker}\pi_T^{(s_1,s_2)}$. It is also shown that, if $\mathcal{P}_s$ satisfies the DLin assumption, then the above bilinear group generator satisfies the $(3, 2)$-subgroup decision assumption [23, Theorem 1, Example 2]. We denote this framework as "Polynomial-I".

This work also proposed a more efficient construction using univariate polynomial representation under the seemingly stronger 2SCasc assumption. A single hidden parameter s is chosen uniformly at random from $\mathbb{Z}_p^*$ to define the subgroups and projection maps. The elements of G (resp. G_T) are represented by the polynomial $f(X) = 1 + X + X^2$ (resp. $f_T(X) = 1 + X + X^2 + X^3 + X^4$). The pairing $e : G \times G \to G_T$ is defined as

$$e((\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3), (\mathfrak{g}_1', \mathfrak{g}_2', \mathfrak{g}_3')) := (\hat{e}(\mathfrak{g}_1, \mathfrak{g}_1'), \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2')\hat{e}(\mathfrak{g}_2, \mathfrak{g}_1'), \hat{e}(\mathfrak{g}_1, \mathfrak{g}_3')\hat{e}(\mathfrak{g}_2, \mathfrak{g}_2')\hat{e}(\mathfrak{g}_3, \mathfrak{g}_1'),$$
$$\hat{e}(\mathfrak{g}_2, \mathfrak{g}_3')\hat{e}(\mathfrak{g}_3, \mathfrak{g}_2'), \hat{e}(\mathfrak{g}_3, \mathfrak{g}_3'))$$

for $(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3), (\mathfrak{g}_1', \mathfrak{g}_2', \mathfrak{g}_3') \in G$. The subgroup $G_1^{(s)}$ is defined using the generators $(\mathfrak{g}^{-s}, \mathfrak{g}, 1)$ and $(1, \mathfrak{g}^{-s}, \mathfrak{g})$. The projection map $\pi^{(s)} : G \to \mathbb{G}$ is defined for $(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) \in G$ as

$$\pi^{(s)}(\mathfrak{g}_1, \mathfrak{g}_2, \mathfrak{g}_3) := \mathfrak{g}_1\mathfrak{g}_2^s\mathfrak{g}_3^{s^2}.$$

Similarly, the subgroup $G_T'^{(s)}$ is defined using the generators $e((\mathfrak{g}^{-s}, \mathfrak{g}, 1), g)$ and $e((1, \mathfrak{g}^{-s}, \mathfrak{g}), g)$ for $g \in G$. The projection map $\pi_T^{(s)} : G_T \to \mathbb{G}_T$ is defined as

$$\pi_T^{(s)}(\mathcal{T}_1, \ldots, \mathcal{T}_5) := \mathcal{T}_1\mathcal{T}_2^s\mathcal{T}_3^{s^2}\mathcal{T}_4^{s^3}\mathcal{T}_5^{s^4}$$

for $(\mathcal{T}_1, \ldots, \mathcal{T}_5) \in G_T$. It is easy to see that $G_1^{(s)} = \operatorname{Ker}\pi^{(s)}$ and $G_T'^{(s)} = \operatorname{Ker}\pi_T^{(s)}$. Observe that both projection maps can be interpreted as (respective) polynomial evaluation at $X = s$. This framework allows representing G_T in terms of five elements of $\mathbb{G}_T$. It is also shown that, if $\mathcal{P}_s$ satisfies the 2SCasc assumption, then the above bilinear group generator satisfies the $(3, 2)$-subgroup decision assumption [23, Theorem 1, Example 1]. We denote this framework as "Polynomial-II".

B.4 Comparison

We compare the above mentioned symmetric projecting frameworks in Table 6 based on the following metrics: size of the target group, number of pairing and cost of computing the projection maps. We also compare the performance of these frameworks when used to instantiate BGN cryptosystem and tabulated the cost in Table 7. One can apply an efficient encryption method (as described in Section 3.2) to optimize the BGN instantiation using the above described frameworks. But the efficient decryption technique, as described in the asymmetric setting, is applicable only in Seo's framework.

Remark 4. Recall that the Groth–Sahai NIWI proof system does not explicitly use the projection map. In their instantiation of the proof system, Herold, Hesse, Hofheinz, Ràfols and Rupp [23] used the implicit representation of G_T where the pairing computation uses the "evaluate-multiply" method in the polynomial projecting setting. However, the BGN scheme uses the projection map π_T in the Dec algorithm, which requires the

Scheme	Groth–Sahai	Seo	Polynomial-I	Polynomial-II
Assumption		DLin		2SCasc
Size of G_T	$9\lvert\mathbb{G}_T\rvert$		$6\lvert\mathbb{G}_T\rvert$	$5\lvert\mathbb{G}_T\rvert$
1P	$9\mathbb{P}_S + 3M_{\mathbb{G}_T} + 6E_{\mathbb{G}}$	$9\mathbb{P}_S + 3M_{\mathbb{G}_T}$	$6\mathbb{P}_S + 32M_{\mathbb{G}_T} + 18M^{\dagger}_{\mathbb{G}}$	$5\mathbb{P}_S + 28M_{\mathbb{G}_T} + 20M^{\dagger}_{\mathbb{G}}$
Projection π_G	$2E_{\mathbb{G}} + 2M_{\mathbb{G}}$	$3E_{\mathbb{G}} + 2M_{\mathbb{G}}$	$3E_{\mathbb{G}} + 2M_{\mathbb{G}}$	$2E_{\mathbb{G}} + 2M_{\mathbb{G}}$
Projection π_T	$8E_{\mathbb{G}_T} + 8M_{\mathbb{G}_T}$	$6E_{\mathbb{G}_T} + 5M_{\mathbb{G}_T}$	$6E_{\mathbb{G}_T} + 5M_{\mathbb{G}_T}$	$4E_{\mathbb{G}_T} + 4M_{\mathbb{G}_T}$

Table 6: Comparing symmetric projection frameworks. († We use the evaluate-multiply-interpolate method to compute the pairing as opposed to [23] which uses the evaluate-multiply method.)

Scheme		Groth–Sahai	Seo	Polynomial-I	Polynomial-II
SK size		$2\lvert\mathbb{F}_p\rvert$	$9\lvert\mathbb{F}_p\rvert$	$2\lvert\mathbb{F}_p\rvert$	$1\lvert\mathbb{F}_p\rvert$
PK size		$6\lvert\mathbb{G}\rvert$	$9\lvert\mathbb{G}\rvert$	$6\lvert\mathbb{G}\rvert$	$5\lvert\mathbb{G}\rvert$
CT size		$9\lvert\mathbb{G}_T\rvert$	$3\lvert\mathbb{G}\rvert$ $6\lvert\mathbb{G}_T\rvert$		$5\lvert\mathbb{G}_T\rvert$
KeyGen		$5E_{\mathbb{G}}$	$9E_{\mathbb{G}}$	$5E_{\mathbb{G}}$	$4E_{\mathbb{G}}$
Enc		$3E_{\mathbb{G}}+1M_{\mathbb{G}}$	$6E_{\mathbb{G}}+4M_{\mathbb{G}}$	$3E_{\mathbb{G}}+1M_{\mathbb{G}}$	$4E_{\mathbb{G}}+2M_{\mathbb{G}}$
Dec‡	$\mathbb{G}$	$2E_{\mathbb{G}}+2M_{\mathbb{G}}$	$4E_{\mathbb{G}}+2M_{\mathbb{G}}$		$2E_{\mathbb{G}}+2M_{\mathbb{G}}$
	$\mathbb{G}_T$	$1\mathbb{P}_S+8E_{\mathbb{G}_T}+8M_{\mathbb{G}_T}$	$1\mathbb{P}_S+6E_{\mathbb{G}_T}+5M_{\mathbb{G}_T}+3E_{\mathbb{G}}+2M_{\mathbb{G}}$		$1\mathbb{P}_S+4E_{\mathbb{G}_T}+4M_{\mathbb{G}_T}$
Add	$\mathbb{G}$	$3E_{\mathbb{G}}+6M_{\mathbb{G}}$	$6E_{\mathbb{G}}+9M_{\mathbb{G}}$	$3E_{\mathbb{G}}+6M_{\mathbb{G}}$	$4E_{\mathbb{G}}+7M_{\mathbb{G}}$
	$\mathbb{G}_T$	$9\mathbb{P}_S+21M_{\mathbb{G}_T}+9E_{\mathbb{G}}$	$9\mathbb{P}_S+15M_{\mathbb{G}_T}+6E_{\mathbb{G}}+3M_{\mathbb{G}}$	$6\mathbb{P}_S+44M_{\mathbb{G}_T}+3E_{\mathbb{G}}+18M_{\mathbb{G}}$	$5\mathbb{P}_S+40M_{\mathbb{G}_T}+4E_{\mathbb{G}}+21M_{\mathbb{G}}$
Multiply		$18\mathbb{P}_S+15M_{\mathbb{G}_T}+15E_{\mathbb{G}}$	$18\mathbb{P}_S+12M_{\mathbb{G}_T}+6E_{\mathbb{G}}+3M_{\mathbb{G}}$	$12\mathbb{P}_S+70M_{\mathbb{G}_T}+3E_{\mathbb{G}}+36M_{\mathbb{G}}$	$10\mathbb{P}_S+62M_{\mathbb{G}_T}+4E_{\mathbb{G}}+41M_{\mathbb{G}}$
Assumption		DLin			2SCasc

Table 7: Comparing BGN in symmetric projection frameworks. (For any group $X \in \{\mathbb{G}, \mathbb{G}_T\}$, we denote by E_X, M_X and $|X|$ the exponentiation, multiplication in X and the bit size of X, respectively, and $\mathbb{P}_S$ denotes the atomic symmetric pairing computation time. ‡ Excluding the final discrete logarithm computation.)

explicit representation of G_T elements. Hence the pairing computation here uses the "evaluate-multiply-interpolate" method. The latter is marginally slower than the former, as the interpolation involves additional multiplications in the target group (see Table 6).

C Ring signature security

Shacham and Waters [35] assume a trusted global setup by an authority. Security is defined in terms of two games. As stated earlier, anonymity (against full key exposure) informally ensures that the adversary cannot distinguish between two target signers even when she/he is given all the private keys in the ring. Whereas unforgeability (with respect to insider corruption) ensures that the adversary cannot forge a signature for an uncorrupted user. See [35] for the formal definitions. Barring a few details in terms of the underlying algebraic structure, the security arguments remain the same for Freeman and Seo–Cheon frameworks. Whenever necessary we give those details for the Freeman construction.

Theorem 4. *The ring signature scheme instantiated in the prime order setting is anonymous against full key exposure attack if the corresponding bilinear group generator $\mathcal{G}_P$ satisfies the $(2, 1)$-subgroup decision assumption.*

Proof. The argument essentially mimics the original proof of [35] in the asymmetric pairing setting. We provide a sketch below.

Given the SDP challenge instance (G, H, G_T, e, g_1, h_1), the simulator $\mathcal{S}$ chooses $g, \mathcal{U}_{G,i}$ for $i \in [0, k]$ from G and h from H uniformly at random. Then $\mathcal{S}$ constructs $A_G = g^a$, $A_H = h^a$, $B_G = g^b$, $B_H = h^b$ and $A_{G,1} = g_1^a$ for

a, b uniformly at random from $\mathbb{Z}_p$. The simulator $\mathcal{S}$ gives the public parameter to adversary $\mathcal{A}$. $\mathcal{S}$ runs the key generation algorithm to obtain public-secret key pairs and answers for signing oracle queries.

When $\mathcal{A}$ makes a challenge query with two indices i_0, i_1, then $\mathcal{S}$ chooses a bit β uniformly at random and constructs a ring signature on behalf of i_β and gives to $\mathcal{A}$. When $\mathcal{A}$ outputs a guess β', $\mathcal{S}$ outputs $1 \oplus \beta \oplus \beta'$. When $\{g_1, h_1\} \in G_1 \times H_1$, one can show that the game is same as the anonymity game defined in [35]. Hence $\mathcal{A}$ wins with the same advantage as in the anonymity game.

When $\{g_1, h_1\}$ are the random elements from $G \times H$, then one can show that the NIWI proof components given as a part of the challenge signature perfectly hides the signer information. Let the secret signing key used to generate the challenge is $sk_{i_\beta} = g^{ab_{i_\beta}}$, for some $b_{i_\beta} \in \mathbb{Z}_p$. Hence, one can express the Waters-3b signature components as

$$S_{G,1} = g^{ab_{i_\beta}} \left(\mathcal{U}_0 \prod_{j=1}^{k} \mathcal{U}_j^{m_j} \right)^r g_1^{at}, \quad S_{G,2} = g^r \quad \text{and} \quad S_{H,2} = h^r.$$

Obviously $S_{G,2}$ and $S_{H,2}$ do not contain any information about i_β. Hence the only possibility for $\mathcal{A}$ to retrieve the signer information is from $S_{G,1}$. Suppose that $\mathcal{A}$ is unbounded and hence can compute the discrete logarithm; $\mathcal{A}$ can retrieve $g^{ab_{i_\beta}} g_1^{at}$ from $S_{G,1}$. However, the hiding property of GS-commitment ensures that even an unbounded adversary cannot extract t_i from $C_{G,i}$ and hence $\mathcal{A}$ cannot compute $t = \sum_i t_i$ or g_1^{at}. Since g_1 is a random group element, the term $g^{ab_{i_\beta}} g_1^{at}$ is uniformly random irrespective of the choice of i_β. This ensures that in this case $\mathcal{A}$ cannot win the game with a probability better than $\frac{1}{2}$. □

Theorem 5. *The ring signature scheme instantiated in the prime order setting is existential unforgeable with respect to insider corruption if $\mathcal{H}$ is a collision resistant hash function and $\mathcal{P}$ satisfies the co-DHP+ assumption.*

Proof. The overall proof strategy is analogous to [35]. However, giving a direct reduction to a hard problem defined in the atomic groups $\mathbb{G}_1, \mathbb{G}_2$ and the knowledge of the exponents used to define the subgroups $\{G_i, H_i\}_{i=1}^2$ allow us to avoid the cancelling property as detailed below.

Like in the original proof [35, Appendix A] we classify the forger in three types. The Type I forger outputs two different message-ring pairs (M, R) and (M', R') such that $\mathcal{H}(M, R) = \mathcal{H}(M', R')$ and thus breaks the collision resistance property of $\mathcal{H}$.

The Type II forger $\mathcal{F}$ outputs a forgery with either $f_i = 0$ for all i or $\sum_i f_i > 1$. The simulator $\mathcal{S}$ is given a co-DHP+ problem instance $\mathfrak{g}, \mathfrak{g}^\alpha, \mathfrak{g}^\beta, \mathfrak{h}, \mathfrak{h}^\alpha, \mathfrak{h}^\beta$ in $\mathbb{G}_1, \mathbb{G}_2$. Then $\mathcal{S}$ chooses linearly independent random $\{\vec{x}_1, \vec{x}_2\}$ and $\{\vec{y}_1, \vec{y}_2\}$ from $\mathbb{Z}_p^2$ and constructs the Freeman projection setting as described in Section 3.1.1. The simulator $\mathcal{S}$ defines $g_1 := \mathfrak{g}^{z\vec{x}_1}$, $g_2 := \mathfrak{g}^{\vec{x}_2}$, $g := \mathfrak{g}^{z\vec{x}_1 + \vec{x}_2}$ and $h_1 := \mathfrak{h}^{z'\vec{y}_1}$, $h_2 := \mathfrak{h}^{\vec{y}_2}$, $h := \mathfrak{h}^{z'\vec{y}_1 + \vec{y}_2}$ for random z, z' from $\mathbb{Z}_p$. Further, the simulator $\mathcal{S}$ uses $(\mathfrak{g}^\alpha, \mathfrak{h}^\alpha)$ to construct $A_G := (\mathfrak{g}^\alpha)^{z\vec{x}_1 + \vec{x}_2} = g^\alpha$, $A_{G,1} := (\mathfrak{g}^\alpha)^{z\vec{x}_1} = g_1^\alpha$, $A_H := (\mathfrak{h}^\alpha)^{z'\vec{y}_1 + \vec{y}_2} = h^\alpha$ and $(\mathfrak{g}^\beta, \mathfrak{h}^\beta)$ to construct $B_G := (\mathfrak{g}^\beta)^{z\vec{x}_1 + \vec{x}_2} = g^\beta$, $B_H := (\mathfrak{h}^\beta)^{z'\vec{y}_1 + \vec{y}_2} = h^\beta$. Well-formedness of these components can be checked using pairing and, unlike the composite order setting, without using the cancelling property. The simulator $\mathcal{S}$ chooses u_j uniformly at random from $\mathbb{Z}_p$ to define the Waters hash generator $\mathcal{U}_j := \mathfrak{g}^{(z\vec{x}_1 + \vec{x}_2)u_j} = g^{u_j}$ for $j \in [0, k]$. Note that $\mathcal{S}$ knows the discrete logarithm of $\mathcal{U}_j$ with respect to $\mathfrak{g}$. Finally, $\mathcal{S}$ selects a CRHF $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ and gives the system parameters to $\mathcal{F}$.

For each user $\mathcal{S}$ runs the key generation algorithm which outputs public and secret key pairs. Hence $\mathcal{S}$ can answer for all the signing and corruption queries. Finally, $\mathcal{F}$ returns a valid ring signature forgery (denoted as $(S^*_{G,1}, S^*_{G,2}, \ldots)$) on the message $(m_1^*, \ldots, m_k^*)$. As in the original proof of [35], $\mathcal{S}$ retrieves f_i^* from GS-commitment $C^*_{G,i}$ or $C^*_{H,i}$ using appropriate projection maps. From the forged signature components $(S^*_{G,1}, S^*_{G,2})$, $\mathcal{S}$ can apply the projection map to compute

$$\hat{g}_2 = \pi_G \left(S^*_{G,1} (S^*_{G,2})^{(u_0 + \sum_{i=1}^k u_i m_i^*)} A_G^{\sum_{i=1}^{l^*} b_i f_i^*} \right)^{\frac{1}{1-f}},$$

where b_i is the random exponent of i-th user's public key and $f = \sum_i f_i^*$. The simulator $\mathcal{S}$ verifies that $\hat{g}_2$ indeed belongs to G_2 and an easy but tedious calculation shows that $\hat{g}_2 = g_2^{\alpha\beta}$. Recall that $g_2 = \mathfrak{g}^{\vec{x}_2}$, where $\vec{x}_2$ is chosen by $\mathcal{S}$. Hence $\mathcal{S}$ can extract $\mathfrak{g}^{\alpha\beta}$ from $g_2^{\alpha\beta}$.

The Type III forger $\mathcal{F}$ outputs a forgery with exactly one of $\{f_i\}$ equal to 1. The simulator $\mathcal{S}$ is given a co-DHP+ problem instance $\mathfrak{g}, \mathfrak{g}^\alpha, \mathfrak{g}^\beta, \mathfrak{h}, \mathfrak{h}^\alpha, \mathfrak{h}^\beta$ in $\mathbb{G}_1$ and $\mathbb{G}_2$. Similar to the Type II case, $\mathcal{S}$ constructs the

Freeman projection setting with complete decomposition of the source groups by choosing $\{\vec{x}_i, \vec{y}_i\}$ from $\mathbb{Z}_p^2$ for $i \in [1, 2]$. The simulator $\mathcal{S}$ defines g_1, g_2, g, h_1, h_2 and h in exactly the same way as in the Type II case. Similarly, $\mathcal{S}$ defines $A_G = g^\alpha$, $A_{G,1} = g_1^\alpha$ and $A_H = h^\alpha$ using the co-DHP+ problem instance but chooses b from $\mathbb{Z}_p$ and defines $B_G = g^b$ and $B_H = h^b$.

Let q be the upper bound of singing queries. As in the security argument of the Waters signature [37], $\mathcal{S}$ chooses a_i, b_i from $\mathbb{Z}_p$ for $i \in [0, k]$ and defines the functions

$$F(M) = a_0 - \tilde{t}q + \sum_{i=1}^{k} a_i m_i \quad \text{and} \quad J(M) = b_0 + \sum_{i=1}^{k} b_i m_i,$$

where $\tilde{t}$ is chosen uniformly at random from $[0, k]$ and $M = (m_1, \ldots, m_k)$. The Waters hash parameters are defined as $\mathcal{U}_{G,0} = [(\mathfrak{g}^\beta)^{a_0 - \tilde{t}q} \mathfrak{g}^{b_0}]^{z\vec{x}_1 + \vec{x}_2}$ and $\mathcal{U}_{G,i} = [(\mathfrak{g}^\beta)^{a_i} \mathfrak{g}^{b_i}]^{z\vec{x}_1 + \vec{x}_2}$. It is easy to see that the above constructed public parameters are properly distributed. The simulator $\mathcal{S}$ randomly chooses one of the users (indexed as i^*) and uses $(\mathfrak{g}^\beta, \mathfrak{h}^\beta)$ from the co-DHP+ instance to construct the corresponding public key as $pk_{i^*} = ((\mathfrak{g}^\beta)^{z\vec{x}_1 + \vec{x}_2}, (\mathfrak{h}^\beta)^{z'\vec{y}_1 + \vec{y}_2}) = (g^\beta, h^\beta)$. This implicitly sets sk_{i^*} as $g^{\alpha\beta}$ which is unknown to $\mathcal{S}$. For the remaining users, $\mathcal{S}$ runs the key generation algorithm to obtain public-secret key pairs. Then $\mathcal{S}$ sends public parameters along with all the users public key to the forger. Note that $\mathcal{S}$ can answer for all the corruption queries except for the target user index i^*.

Suppose that $\mathcal{F}$ makes signing query on the message $M = (m_1, \ldots, m_k)$ with user index s in the ring R. If $s \neq i^*$, then $\mathcal{S}$ can easily respond by using the corresponding secret key. For $s = i^*$, $\mathcal{S}$ follows the strategy used in the proof of Waters signature security [37]. In particular, $\mathcal{S}$ evaluates $F(M)$ and aborts if $F(M) = 0$. Otherwise, $\mathcal{S}$ computes the ring signature as follows: construct the GS-commitment and proof components as defined in equation (4.1) and equation (4.2) by choosing t_i, s_i uniformly at random from $\mathbb{Z}_p$. Then compute the signature components which will be of the form

$$S_{G,1} := (g^\alpha)^{-\frac{J(M)}{F(M)}} \left(\mathcal{U}_0 \prod_{i=1}^{k} \mathcal{U}_i^{m_i} \right)^{\tilde{r}} (g_1^\alpha)^t, \quad S_{G,2} := g^{\tilde{r}} (g^\alpha)^{-\frac{1}{F(M)}}, \quad S_{H,2} := h^{\tilde{r}} (h^\alpha)^{-\frac{1}{F(M)}},$$

where $\tilde{r} = r + \frac{\alpha}{F(M)}$ for r chosen uniformly at random from $\mathbb{Z}_p$ and $t = \sum_i t_i$. The well-formedness of the above signature can be verified without using the cancelling property. The simulator $\mathcal{S}$ sends the above signature components along with GS-commitment and NIWI proof components to $\mathcal{F}$. Finally, $\mathcal{F}$ outputs a valid ring signature forgery of the form (σ^*, M^*, R^*). At this point $\mathcal{S}$ checks whether the forgery is for target user i^*. If not $\mathcal{S}$ aborts the game, otherwise, $\mathcal{S}$ evaluates $F(M^*)$. If $F(M^*) \neq 0$, then $\mathcal{S}$ aborts the game, else retrieves the co-DHP+ solution as follows. From the valid ring signature forgery, the simulator computes by

$$\hat{g}_2 = \pi_G\big(S_{G,1}^* (S_{G,2}^*)^{-J(M^*)}\big).$$

As in the case of the Type II forger, one can show that $\hat{g}_2 = g_2^{\alpha\beta}$ from which $\mathcal{S}$ can extract $\mathfrak{g}^{\alpha\beta}$. □

D Blind signature security

D.1 Unbalanced projecting setting

Theorem 6. *Suppose that $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and the DLin assumption in $\mathbb{H}$. Then $\mathcal{G}_{\mathrm{UP}}$ satisfies the $(2, 1)$-SDP assumption in G and the $(3, 2)$-SDP assumption in $\mathbb{H}^3 \subseteq \mathbb{G}_1^3 \times H$.*

Proof. The proof is divided into two parts.

Given a polynomial time algorithm $\mathcal{A}_1$ for the $(2, 1)$ subgroup decision problem in G with non-negligible advantage ε_1, we construct a polynomial time solver $\mathcal{B}_1$ for the DDH problem in $\mathbb{G}_1$. Given the DDH instance $\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, \mathfrak{g}^c$ from $\mathbb{G}_1$ for $a, b \xleftarrow{\$} \mathbb{Z}_p$, $\mathcal{B}_1$'s goal is to decide whether $\mathfrak{g}^c$ is $\mathfrak{g}^{ab}$ or not. The solver $\mathcal{B}_1$ defines the groups G as $\mathbb{G}_1^2$, H as $\mathbb{G}_2^3$ and G_T as $\mathbb{G}_T^6$. Then $\mathcal{B}_1$ defines the bilinear map $e : G \times H \to G_T$ as in Algorithm 2. The solver $\mathcal{B}_1$ defines the subgroup G_1 of G such that it is generated by $(\mathfrak{g}^\alpha, \mathfrak{g}^{\alpha\alpha})$ for some $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Further, $\mathcal{B}_1$ chooses $\gamma, r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$ and defines the subgroups $\mathbb{H}_1$ and $\mathbb{H}_2$ such that they are generated

by $(\mathfrak{g}^{r_1\gamma}, 1, \mathfrak{g}^{\gamma})\times(\mathfrak{h}^{r_1\gamma}, 1, \mathfrak{h}^{\gamma})$ and $(1, \mathfrak{g}^{r_2\gamma}, \mathfrak{g}^{\gamma})\times(1, \mathfrak{h}^{r_2\gamma}, \mathfrak{h}^{\gamma})$, respectively. Now $\mathcal{B}_1$ uses the DDH instance to construct a $(2, 1)$-SDP instance $(G, H, G_T, e, G_1, \mathbb{H}_1, \mathbb{H}_2, (\mathfrak{g}^{b\alpha}, \mathfrak{g}^{c\alpha}))$ and sends to $\mathcal{A}_1$. From our initial assumption, $\mathcal{A}_1$ decides whether $(\mathfrak{g}^{b\alpha}, \mathfrak{g}^{c\alpha})$ belongs to G_1 or the whole group G with non-negligible advantage ε_1. The solver $\mathcal{B}_1$ outputs 1, only when $\mathcal{A}_1$ outputs 1, otherwise $\mathcal{B}_1$ outputs 0. Hence $\mathrm{Adv}_{\mathrm{DDH}} \le \mathrm{Adv}_{\mathrm{SDP}_G} = \varepsilon_1$, which is non-negligible.

Given a polynomial time algorithm $\mathcal{A}_2$ for the $(3, 2)$ subgroup decision problem in $\mathbb{H}^3$ with non-negligible advantage ε_1, we construct a polynomial time solver $\mathcal{B}_2$ for the DLin problem in $\mathbb{H}$. The solver $\mathcal{B}_2$ is given the description of the groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the pairing map $\hat{e}$ along with the DLin problem instance $(\mathfrak{g}, \mathfrak{h}), (\mathfrak{g}^{z_1}, \mathfrak{h}^{z_1}), (\mathfrak{g}^{z_2}, \mathfrak{h}^{z_2}), (\mathfrak{g}^{z_1z_3}, \mathfrak{h}^{z_1z_3}), (\mathfrak{g}^{z_2z_4}, \mathfrak{h}^{z_2z_4}), (\mathfrak{g}^{z_5}, \mathfrak{h}^{z_5})$ from $\mathbb{H}$ for $z_1, \ldots, z_4 \xleftarrow{\$} \mathbb{Z}_p$. Then $\mathcal{B}_2$'s goal is to decide whether $(\mathfrak{g}^{z_5}, \mathfrak{h}^{z_5}) = (\mathfrak{g}^{z_3+z_4}, \mathfrak{h}^{z_3+z_4})$ or not. The solver $\mathcal{B}_2$ constructs the groups G, H, G_T and the bilinear map e as in the previous reduction. Then $\mathcal{B}_2$ chooses $\beta \xleftarrow{\$} \mathbb{Z}_p$ and constructs the subgroups $\mathbb{H}_1$ and $\mathbb{H}_2$ of $\mathbb{H}^3$ such that they are generated by $(\mathfrak{g}^{z_1\beta}, 1, \mathfrak{g}^{\beta}) \times (\mathfrak{h}^{z_1\beta}, 1, \mathfrak{h}^{\beta})$ and $(1, \mathfrak{g}^{z_2\beta}, \mathfrak{g}^{\beta}) \times (1, \mathfrak{h}^{z_2\beta}, \mathfrak{h}^{\beta})$, respectively. In addition, $\mathcal{B}_2$ chooses $a, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and constructs the subgroup G_1 of G such that it is generated by $(\mathfrak{g}^{\alpha}, \mathfrak{g}^{a\alpha})$. Finally, $\mathcal{B}_2$ constructs a $(3, 2)$-SDP instance

$$(G, H, G_T, e, G_1, \mathbb{H}_1, \mathbb{H}_2, (\mathfrak{g}^{z_1z_3\beta}, \mathfrak{g}^{z_2z_4\beta}, \mathfrak{g}^{z_5\beta}) \times (\mathfrak{h}^{z_1z_3\beta}, \mathfrak{h}^{z_2z_4\beta}, \mathfrak{h}^{z_5\beta}))$$

and sends to $\mathcal{A}_2$. From our initial assumption $\mathcal{A}_2$ decides the subgroup membership of above instance with non-negligible advantage ε_2. The solver $\mathcal{B}_2$ outputs 1, only when $\mathcal{A}_2$ outputs 1, otherwise $\mathcal{B}_2$ outputs 0. Hence $\mathrm{Adv}_{\mathrm{DLin}_{\mathbb{H}}} \le \mathrm{Adv}_{\mathrm{SDP}} = \varepsilon_2$, which is non-negligible. □

D.2 Blindness

Seo and Cheon [34] established the blindness property in the symmetric prime order setting using a hybrid argument through a series of $m + 1$ games. The argument does not make use of the projecting, cancelling or translating property. One can recast their argument in the asymmetric setting. However, we avoid the m intermediate games and give a tight reduction by using the random self-reducibility of DDH and DLin (see, for example, [17, Lemma 1]).

Theorem 7. *The blind signature scheme instantiated in the unbalanced projecting setting satisfies the blindness property assuming the underlying NIWI proof system is secure.*

Proof (Sketch). The CRS of the NIWI proof system is given to the simulator $\mathcal{S}$, which includes u_1, u_2 from $G = \mathbb{G}_1^2$, $\mathbf{v}_{\mathbb{H},i} = (v_{\mathbb{G}_1,i}, v_i)$ from $\mathbb{H}^3$, where $v_{\mathbb{G}_1,i}$ is from $\mathbb{G}_1^3$ and v_i is from $H = \mathbb{G}_2^3$ for $i \in [1, 3]$ along with (G, H, G_T, e). The goal of $\mathcal{S}$ is to distinguish between the binding and hiding settings. The simulator $\mathcal{S}$ constructs the subgroup G_1 of G generated by u_1, the subgroup $\check{H}_i$ of $\mathbb{G}_1^3$ generated by $v_{\mathbb{G}_1,i}$ and H_i of H by v_i for $i \in [1, 2]$. The simulator $\mathcal{S}$ defines $u_G := u_2 \odot (1, \mathfrak{g})$ and $\mathbf{v}_{\mathbb{H}} = (v_{\mathbb{G}_1}, v_H)$, where $v_{\mathbb{G}_1} := v_{\mathbb{G}_1,3} \odot (1, 1, \mathfrak{g})$ and $v_H := v_3 \odot (1, 1, \mathfrak{h})$. Using the random self-reducibility of DDH and DLin, $\mathcal{S}$ chooses α_i, β_j uniformly at random from $\mathbb{Z}_p$ and defines the Waters hash generators $\mathcal{U}_{G,i} := u_G^{\alpha_i}$ and $\mathcal{V}_{\mathbb{H},j} = (\mathcal{V}_{\mathbb{G}_1,j}, \mathcal{V}_{H,j}) = (v_{\mathbb{G}_1}^{\beta_j}, v_H^{\beta_j})$ for $i \in [0, m]$ and $j \in [1, m]$. Then $\mathcal{S}$ chooses g (resp. h) uniformly at random from the group G (resp. H). Also $\mathcal{S}$ chooses ζ, ζ_i uniformly at random from $\mathbb{Z}_p$ and defines $g_1 := u_1^{\zeta}$, $\mathbf{h}_{\mathbb{H},i} = (h_{\mathbb{G}_1,i}, h_i) = (v_{\mathbb{G}_1,i}^{\zeta_i}, v_i^{\zeta_i})$ for $i \in [1, 2]$. Finally, $\mathcal{S}$ sends the public parameter to the adversary $\mathcal{A}$. When $\mathcal{A}$ makes a challenge query by sending $(\mathrm{info}, M_0, M_1)$, $\mathcal{S}$ chooses one of the messages randomly and interacts with $\mathcal{A}$. The successful interaction outputs a valid signature, say σ_b. Similarly $\mathcal{S}$ uses the other message and interacts with $\mathcal{A}$. Again the successful interaction outputs a valid signature σ_{1-b}. Now $\mathcal{A}$ is given with both signatures σ_0 and σ_1. The simulator $\mathcal{S}$ outputs 1 if $\mathcal{A}$ correctly guesses, otherwise outputs 0.

Now we briefly analyse the above experiment under the security of the Groth–Sahai NIWI proof system from [20] in terms of two games. If $\mathcal{S}$ is given a binding key, then we call it Game-0. In Game-0, the elements $\{\mathcal{U}_{G,i}\}_{i=0}^m$ and $\{\mathcal{V}_{\mathbb{H},j}\}_{j=1}^m$ are from the respective groups G and $\mathbb{H}^3$. This is the actual blindness security game and hence $\mathcal{A}$ has a non-negligible advantage. If $\mathcal{S}$ is given a hiding key, then we are in Game-1. Here $\{\mathcal{U}_{G,i}\}_{i=0}^m$ and $\{\mathcal{V}_{\mathbb{H},j}\}_{j=1}^m$ are from the respective subgroups G_1 and $\mathbb{H}_1 \oplus \mathbb{H}_2$, where $\mathbb{H}_l = (\check{H}_l, H_l)$, for $l \in [1, 2]$. One can show that even for an unbounded $\mathcal{A}$ the advantage in Game-1 will be zero. The argument essentially mimics

the symmetric setting proof of Seo and Cheon [34, Lemma 4]. There are three possibilities for $\mathcal{A}$ to obtain information about the secret bit b. The first one is users commitment and NIWI proof components. Since we are in hiding key setting, one can see that commitment and NIWI proof components do not reveal any information about the message. The second one is the user (pretend as $\mathcal{S}$) response. When $\mathcal{A}$ returns the blinded signature $(K_{G,1}, K_{G,2}, K_{H,2}, K_{G,3})$, $\mathcal{S}$ performs two pairing based verification. It is easy to see that $\mathcal{A}$ can perform the same verification without any involvement of $\mathcal{S}$. Thus this step does not provide any additional information about b from $\mathcal{S}$. The third possibility is to infer b from the output of two unblinded signatures. However, re-randomization of the signatures will ensure the uniform distribution of randomness. Hence $\mathcal{A}$ cannot obtain any additional information about the message. Thus the advantage of $\mathcal{A}$ in Game-1 will be zero. □

D.3 More details for the unforgeability proof

Subgroup descriptions. The subgroups $G_i, H_j, \check{H}_j$ are generated by $\hat{g}_i$, $\hat{h}_j$ and $\hat{h}_{\mathbb{G}_1,j}$, respectively, for $i \in [1,2]$ and $j \in [1,3]$. Then $\mathcal{S}$ defines the group element $R := \hat{g}_1\hat{g}_2$ in G and $S := \hat{h}_1\hat{h}_2\hat{h}_3$ in H.

Waters signature. Let q_{info} be the number of signing queries for the common information info and let q be the sum of all q_{info} for all info issued by the forger. Now the simulator sets $l = 4q$ and chooses $k \xleftarrow{\$} [0,m]$, $z', z_i \xleftarrow{\$} [0, l-1]$, $w'_j, w_{ji} \xleftarrow{\$} \mathbb{Z}_p$. Let $M = b_1, b_2, \ldots, b_m$, define

$$F(M) := (p - lk) + z' + \sum_{i\in[1,m]} b_i z_i \quad \text{and} \quad J_j(M) := w'_j + \sum_{i\in[1,m]} b_i w_{ji}$$

for $i \in [1,m]$, $j \in [1,2]$.

Some computation details. Set

$$\mathcal{U}_{G,0} := \hat{g}_1^{w'_1}\hat{g}_2^{w'_2}B_{G,2}^{(p-lk+z')}, \quad \mathcal{U}_{G,i} := \hat{g}_1^{w_{1i}}\hat{g}_2^{w_{2i}}B_{G,2}^{z_i},$$

and

$$\mathcal{V}_{\mathbb{H},i} = (\mathcal{V}_{\mathbb{G}_1,i}, \mathcal{V}_{H,i}) = (\hat{h}_{\mathbb{G}_1,1}^{\bar{w}_{1i}}\hat{h}_{\mathbb{G}_1,2}^{\bar{w}_{2i}}A_{\mathbb{G}_1,3}^{\bar{w}_{3i}}, \hat{h}_1^{\bar{w}_{1i}}\hat{h}_2^{\bar{w}_{2i}}A_{H,3}^{\bar{w}_{3i}}),$$

where $\bar{w}_{ji} \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [1,m]$ and $j \in [1,3]$. The public key is $A = e(\hat{g}_1^{a'}B_{G,2}, A_{H,1}A_{H,2}A_{H,3})$, which can be re-written as $e(A_{G,1}^{a'}\hat{g}_2^{ab}, S)$. The corresponding secret key is $g' = A_{G,1}^{a'}\hat{g}_2^{ab}$ which is unknown to $\mathcal{S}$.

To construct the signature, $\mathcal{S}$ first checks whether $F(M) \neq 0$; if not $\mathcal{S}$ aborts the game. Otherwise, $\mathcal{S}$ constructs the unblinded signature

$$S_{G,1} = A_{G,1}^{a' - \frac{J_1(M)}{F(M)}}A_{G,2}^{-\frac{J_2(M)}{F(M)}}\left(\mathcal{U}_{G,0}\prod_{i\in[1,m]}\mathcal{U}_{G,i}^{b_i}\right)^r, \quad S_{G,2} = R^{-r}(A_{G,1}A_{G,2})^{\frac{1}{F(M)}}, \quad S_{H,2} = S^{-r}(A_{H,1}A_{H,2}A_{H,3})^{\frac{1}{F(M)}}.$$

Extraction of message bits by using the projection map. Consider the commitment of b_i, $C_{G,i} = \mathcal{U}_{G,i}^{b_i}g_1^{t_i}$ as described in Table 3. From the definition of the projection map, we know that $G_1 \subseteq \mathrm{Ker}(\pi_G)$. Hence applying π_G on $C_{G,i}$ will kill the G_1 component. Since $\mathcal{U}_{G,i} \in G = G_1 \oplus G_2$, $g_1^{t_i} \in G_1$, we obtain that $\pi_G(C_{G,i}) = 1$ if and only if $b_i = 0$, otherwise $b_i = 1$.

Lemma 8. *If the commitments $C_{G,i}, C_{H,i}$ and proofs $\bar{\Theta}_{G,i}, \bar{\boldsymbol{\Theta}}_{\mathbb{H},i}$ satisfy the verification equations described in Table 3, then they can be uniquely expressed as*

$$\begin{aligned}
C_{G,i} &:= \mathcal{U}_{G,i}^{b_i}g_1^{t_i}, & C_{H,i} &:= \mathcal{V}_{H,i}^{b_i}h_1^{s_{i,1}}h_2^{s_{i,2}},\\
\Theta_{G,i,1} &:= \mathcal{U}_{G,i}^{(b_i-1)s_{i,1}}g_1^{r_{i,1}}, & \Theta_{H,i,1} &:= (\mathcal{V}_{H,i}^{b_i}h_1^{s_{i,1}}h_2^{s_{i,2}})^{t_i}h_1^{-r_{i,1}}h_2^{-r_{i,2}},\\
\Theta_{G,i,2} &:= \mathcal{U}_{G,i}^{(b_i-1)s_{i,2}}g_1^{r_{i,2}}, & \Theta_{H,i,2} &:= (\mathcal{V}_{H,i}^{b_i-1}h_1^{s_{i,1}}h_2^{s_{i,2}})^{t_i}h_1^{-r_{i,3}}h_2^{-r_{i,4}},\\
\Theta_{G,i,3} &:= \mathcal{U}_{G,i}^{b_is_{i,1}}g_1^{r_{i,3}}, & \widetilde{\Theta}_{\mathbb{G}_1,i,1} &:= (\mathcal{V}_{\mathbb{G}_1,i}^{b_i}h_{\mathbb{G}_1,1}^{s_{i,1}}h_{\mathbb{G}_1,2}^{s_{i,2}})^{t_i}h_{\mathbb{G}_1,1}^{-r_{i,1}}h_{\mathbb{G}_1,2}^{-r_{i,2}},\\
\Theta_{G,i,4} &:= \mathcal{U}_{G,i}^{b_is_{i,2}}g_1^{r_{i,4}}, & \widetilde{\Theta}_{\mathbb{G}_1,i,2} &:= (\mathcal{V}_{\mathbb{G}_1,i}^{b_i-1}h_{\mathbb{G}_1,1}^{s_{i,1}}h_{\mathbb{G}_1,2}^{s_{i,2}})^{t_i}h_{\mathbb{G}_1,1}^{-r_{i,3}}h_{\mathbb{G}_1,2}^{-r_{i,4}}
\end{aligned}$$

for some $b_i \in \{0,1\}$ and $t_i, s_{i,1}, s_{i,2}, r_{i,1}, \ldots, r_{i,4} \in \mathbb{Z}_p$.

The proof will mimic that of [34, Lemma 5] in the asymmetric pairing setting and we omit the details.

References

[1] G. Adj, I. Canales-Martínez, N. Cruz Cortés, A. Menezes, T. Oliveira, L. Rivera-Zamarripa and F. Rodríguez-Henríquez, Computing discrete logarithms in cryptographically-interesting characteristic-three finite fields, IACR Cryptology ePrint Archive (2016), https://eprint.iacr.org/2016/914.pdf.

[2] R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic, in: *Advances in Cryptology—EUROCRYPT 2014*, Lecture Notes in Comput. Sci. 8441, Springer, Heidelberg (2014), 1–16.

[3] P. S. L. M. Barreto and M. Naehrig, Pairing-friendly elliptic curves of prime order, in: *Selected Areas in Cryptography*, Lecture Notes in Comput. Sci. 3897, Springer, Berlin (2006), 319–331.

[4] M. Bellare, D. Micciancio and B. Warinschi, Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions, in: *Advances in Cryptology—EUROCRYPT 2003*, Lecture Notes in Comput. Sci. 2656, Springer, Berlin (2003), 614–629.

[5] A. Bender, J. Katz and R. Morselli, Ring signatures: stronger definitions, and constructions without random oracles, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 3876, Springer, Berlin (2006), 60–79.

[6] D. Boneh and X. Boyen, Short signatures without random oracles, in: *Advances in Cryptology—EUROCRYPT 2004*, Lecture Notes in Comput. Sci. 3027, Springer, Berlin (2004), 56–73.

[7] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, in: *Advances in Cryptology—CRYPTO 2001*, Lecture Notes in Comput. Sci. 2139, Springer, Berlin (2001), 213–229.

[8] D. Boneh, E.-J. Goh and K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 3378, Springer, Berlin (2005), 325–341.

[9] D. Boneh and B. Waters, Conjunctive, subset, and range queries on encrypted data, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 4392, Springer, Berlin (2007), 535–554.

[10] X. Boyen and B. Waters, Compact group signatures without random oracles, in: *Advances in Cryptology—EUROCRYPT 2006*, Lecture Notes in Comput. Sci. 4004, Springer, Berlin (2006), 427–444.

[11] X. Boyen and B. Waters, Full-domain subgroup hiding and constant-size group signatures, in: *Public Key Cryptography—PKC 2007*, Lecture Notes in Comput. Sci. 4450, Springer, Berlin (2007), 1–15.

[12] S. Chatterjee, D. Hankerson, E. Knapp and A. Menezes, Comparing two pairing-based aggregate signature schemes, *Des. Codes Cryptogr.* **55** (2010), no. 2–3, 141–167.

[13] S. Chatterjee and A. Menezes, On cryptographic protocols employing asymmetric pairings—the role of Ψ revisited, *Discrete Appl. Math.* **159** (2011), no. 13, 1311–1322.

[14] S. Chatterjee and A. Menezes, Type 2 structure-preserving signature schemes revisited, in: *Advances in Cryptology—ASIACRYPT 2015. Part I*, Lecture Notes in Comput. Sci. 9452, Springer, Heidelberg (2015), 286–310.

[15] S. Chatterjee, A. Menezes and F. Rodríguez-Henríquez, On implementing pairing-based protocols with elliptic curves of embedding degree one, IACR Cryptology ePrint Archive (2016), https://eprint.iacr.org/2016/403.pdf.

[16] S. Chatterjee, P. Sarkar and R. Barua, Efficient computation of Tate pairing in projective coordinate over general characteristic fields, in: *Information security and cryptology—ICISC 2004*, Lecture Notes in Comput. Sci. 3506, Springer, Berlin (2005), 168–181.

[17] A. Escala, G. Herold, E. Kiltz, C. Ràfols and J. Villar, An algebraic framework for Diffie–Hellman assumptions, in: *Advances in Cryptology—CRYPTO 2013. Part II*, Lecture Notes in Comput. Sci. 8043, Springer, Heidelberg (2013), 129–147.

[18] D. M. Freeman, Converting pairing-based cryptosystems from composite-order groups to prime-order groups, in: *Advances in Cryptology—EUROCRYPT 2010*, Lecture Notes in Comput. Sci. 6110, Springer, Berlin (2010), 44–61.

[19] S. D. Galbraith, K. G. Paterson and N. P. Smart, Pairings for cryptographers, *Discrete Appl. Math.* **156** (2008), no. 16, 3113–3121.

[20] E. Ghadafi, N. P. Smart and B. Warinschi, Groth–Sahai proofs revisited, in: *Public Key Cryptography—PKC 2010*, Lecture Notes in Comput. Sci. 6056, Springer, Berlin (2010), 177–192.

[21] J. Groth and A. Sahai, Efficient non-interactive proof systems for bilinear groups, in: *Advances in Cryptology—EUROCRYPT 2008*, Lecture Notes in Comput. Sci. 4965, Springer, Berlin (2008), 415–432.

[22] A. Guillevic, Comparing the pairing efficiency over composite-order and prime-order elliptic curves, in: *Applied Cryptography and Network Security—ACNS 2013*, Lecture Notes in Comput. Sci. 7954, Springer, Berlin (2013), 357–372.

[23] G. Herold, J. Hesse, D. Hofheinz, C. Ràfols and A. Rupp, Polynomial spaces: A new framework for composite-to-prime-order transformations, in: *Advances in Cryptology—CRYPTO 2014. Part I*, Lecture Notes in Comput. Sci. 8616, Springer, Heidelberg (2014), 261–279. Full version is available in IACR eprint archive, https://eprint.iacr.org/2014/445.pdf.

[24] A. Joux, A one round protocol for tripartite Diffie-Hellman, in: *Algorithmic Number Theory* (Leiden 2000), Lecture Notes in Comput. Sci. 1838, Springer, Berlin (2000), 385–393.

[25] J. Katz, A. Sahai and B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: *Advances in cryptology—EUROCRYPT 2008*, Lecture Notes in Comput. Sci. 4965, Springer, Berlin (2008), 146–162.

[26] T. Kim and R. Barbulescu, Extended tower number field sieve: A new complexity for the medium prime case, in: *Advances in Cryptology—CRYPTO 2016. Part I*, Lecture Notes in Comput. Sci. 9814, Springer, Berlin (2016), 543–571.
[27] A. Lewko, Tools for simulating features of composite order bilinear groups in the prime order setting, in: *Advances in Cryptology—EUROCRYPT 2012*, Lecture Notes in Comput. Sci. 7237, Springer, Heidelberg (2012), 318–335.
[28] A. Lewko and S. Meiklejohn, A profitable sub-prime loan: Obtaining the advantages of composite order in prime-order bilinear groups, in: *Public-Key Cryptography—PKC 2015*, Lecture Notes in Comput. Sci. 9020, Springer, Heidelberg (2015), 377–398.
[29] A. Lewko and B. Waters, New techniques for dual system encryption and fully secure HIBE with short ciphertexts, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 5978, Springer, Berlin (2010), 455–479.
[30] S. Meikeljohn, H. Shacham and D. M. Freeman, Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures, in: *Advances in Cryptology—ASIACRYPT 2010*, Lecture Notes in Comput. Sci. 6477, Springer, Berlin (2010), 519–538.
[31] T. Okamoto and K. Takashima, Homomorphic encryption and signatures from vector decomposition, in: *Pairing-Based Cryptography—Pairing 2008*, Lecture Notes in Comput. Sci. 5209, Springer, Berlin (2008), 57–74.
[32] T. Okamoto and K. Takashima, Hierarchical predicate encryption for inner-products, in: *Advances in Cryptology—ASIACRYPT 2009*, Lecture Notes in Comput. Sci. 5912, Springer, Berlin (2009), 214–231.
[33] J. H. Seo, On the (im)possibility of projecting property in prime-order setting, in: *Advances in Cryptology—ASIACRYPT 2012*, Lecture Notes in Comput. Sci. 7658, Springer, Heidelberg (2012), 61–79. Full version is available in IACR eprint archive, https://eprint.iacr.org/2013/186.pdf.
[34] J. H. Seo and J. H. Cheon, Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 7194, Springer, Heidelberg (2012), 133–150. Full version is available in IACR eprint archive, https://eprint.iacr.org/2012/198.pdf.
[35] H. Shacham and B. Waters, Efficient ring signatures without random oracles, in: *Public Key Cryptography—PKC 2007*, Lecture Notes in Comput. Sci. 4450, Springer, Berlin (2007), 166–180. Full version is available in IACR eprint archive, https://eprint.iacr.org/2006/289.pdf.
[36] E. Shen, E. Shi and B. Waters, Predicate privacy in encryption systems, in: *Theory of Cryptography*, Lecture Notes in Comput. Sci. 5444, Springer, Berlin (2009), 457–473.
[37] B. Waters, Efficient identity-based encryption without random oracles, in: *Advances in Cryptology—EUROCRYPT 2005*, Lecture Notes in Comput. Sci. 3494, Springer, Berlin (2005), 114–127.