Towards Static Assumption Based Cryptosystem in Pairing Setting: Further Applications of DéjàQ and Dual-Form Signature

Sanjit Chatterjee and R Kabaleeshwaran✉

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

Abstract. A large number of parameterized complexity assumptions have been introduced in the bilinear pairing setting to design novel cryptosystems and an important question is whether such "q-type" assumptions can be replaced by some static one. Recently Ghadafi and Groth captured several such parameterized assumptions in the pairing setting in a family called bilinear target assumption (BTA). We apply the DéjàQ techniques for all q-type assumptions in the BTA family. In this process, first we formalize the notion of extended adaptive parameter-hiding property and use it in the Chase-Meiklejohn's DéjàQ framework to reduce those q-type assumptions from subgroup hiding assumption in the asymmetric composite-order pairing. In addition, we extend the BTA family further into BTA1 and BTA2 and study the relation between different BTA variants. We also discuss the inapplicability of DéjàQ techniques on the q-type assumptions that belong to BTA1 or BTA2 family. We then provide one further application of Gerbush et al's dual-form signature techniques to remove the dependence on a q-type assumption for which existing DéjàQ techniques are not applicable. This results in a variant of Abe et al's structure-preserving signature with security based on a static assumption in composite order setting.

Keywords: Bilinear target assumption, q-type assumption, DejaQ technique, Dual form signature technique.

1 Introduction

Rapid development of pairing-based cryptography has witnessed an enormous number of complexity assumptions. The thurst for new complexity assumptions become somewhat unavoidable due to the role that they play in the security reduction of many novel construction of cryptographic protocols. For example, Boneh and Boyen [BB04] introduced strong Diffie-Hellman (SDH) assumption to propose a signature scheme in the standard model and Abe et al. [AGHO11] introduced a tailor-made assumption (later called as q-AGHO assumption) to prove the security of their structure preserving signature scheme.

Parameterized Assumptions. This type of non-static q-type assumptions have been extensively used in the security argument of pairing-based protocols. For example, the q-SDH assumption and its variants q-hidden SDH (HSDH), q-asymmetric hidden SDH (ADHSDH), (q, ℓ)-Poly-SDH, q-2 variable SDH (2SDH) are used in various signature schemes [BB04, Fuc09, Boy07, Oka06a]. The parameter q is typically related to the number of oracle queries given to the adversary in the security game. However, parameterized assumptions do have some implication on concrete security and may require larger size for the underlying groups. Also it is observed that the parameterized assumption becomes stronger as these parameters grows. In particular Cheon [Che06] proved that for q-SDH assumption, the secret information (say x) can be recovered using $O(\sqrt{p/q})$ group operations, where p is the underlying group order. Jao and Yoshida [JY09] proved that q-SDH assumption is equivalent to Boneh-Boyen signature

and also showed that this relation allows one to recover the secret key in time $O(p^{2/5+\epsilon})$, using $O(p^{1/5+\epsilon})$ signature queries. Hence, it's relevant to investigate whether for a particular cryptosystem one can remove the dependency on parameterized assumption. Two prominent approaches in this direction are DéjàQ [CM14, CMM16] and dual-form signature techniques [GLOW12].

BTA. Boneh, Boyen and Goh [BBG05] introduced the Uber assumption family which captures many complexity assumptions under it. Boyen [Boy08] informally suggested to extend the Uber assumption family to those assumptions with (a) flexible challenge terms, (b) rational polynomial exponents in both problem instance and challenge terms, (c) composite-order group of known or unknown factorization. Recently, Ghadafi and Groth [GG17] focused on the first two points above in the context of non-interactive computational assumptions. In the bilinear pairing setting, they formulated the bilinear target assumption (BTA) family. In the BTA family, the exponent of both problem instance and challenge terms are represented using rational polynomials and all the polynomial coefficients are given explicitly as $\mathbb{Z}_p$ elements, where p is the group order. The challenge terms are determined by the adversary's input, whose exponents are represented as coefficients of a rational polynomial. However there are many tailor-made assumptions that are not captured by this BTA family. Some examples are (q, ℓ)-Poly-SDH [Boy07], q-AGHO [AGHO11], q-simultaneous flexible pairing (SFP) [AFG$^+$10] etc.

In this work we focus on q-type assumptions that belong to BTA family for which no reduction is known from subgroup hiding (SGH) assumption. Examples of such assumptions are generalized q-co-SDH [FHS14] and Boneh-Boyen computational Diffie-Hellman (BB-CDH) assumption [BCC$^+$09].

DéjàQ. Our main approach is to use the DéjàQ framework. The seminal work of Chase and Meiklejohn [CM14] showed that certain parameterized assumptions are implied by SGH assumption in the asymmetric composite-order pairing. In particular, they gave a reduction from SGH to certain q-type assumptions such as decisional q-type assumptions which are one sided (for example, exponent q-SDH assumption) and computational q-type assumptions which are two sided (q-Diffie-Hellman inversion assumption). Also they gave a reduction for q-SDH from SGH assumption, which is having flexible challenge term. However they were not able to give a reduction for those q-type assumptions where challenge terms belong to the target group G_T (for example, q-DDHE assumption). This is solved by Chase et al's [CMM16] extended framework. Their technique treats the generators of different groups using separate ways in the asymmetric composite-order pairing. In particular separate generators are used to answer separate types of queries and because of the access to additional randomness, these generators are indistinguishable by the bounded adversary. In 2015, Wee [Wee15] came up with similar approach at protocol level instead of assumption level, but in the symmetric composite-order pairing setting.

Dual-form signatures. Our second approach for removing dependence on parameterized assumption is to utilize Gerbush et al's [GLOW12] dual-form signature techniques. For example, we consider the Abe et al's structure-preserving signature scheme [AGHO11] which is used as a building block in other cryptosystems [Gha14, Gha15]. The security of Abe et al's structure-preserving signature is proved under q-AGHO assumption. We observe that (in §4.3) we cannot apply the existing DéjàQ technique for this q-AGHO assumption, in order to reduce it from SGH assumption. Hence we construct a dual-form of Abe et al's structure-preserving signature scheme and prove its security under SGH assumption (in §5.1). We do not alter the original Abe et al's construction for the dual-form variant. The dual-form signature technique changes the scheme construction slightly from the original, as it introduces some additional randomness in the construction to argue security based on static assumption. For example, we construct the dual-form of Boyen-Waters [BW07] group signature scheme under static assumption instead of q-HSDH assumption (in §5.2).

1.1 Our Contribution

1. We extend the BTA family further (in §3.2) to capture the assumptions (q, ℓ)-Poly-SDH, q-AGHO, q-SFP and q-HSDH, which are not covered under BTA family [GG17]. Also we investigate the relation among these new variants in §3.3.
2. We formalize the extended adaptive parameter-hiding property (in §4.1). Then we use it in the Chase-Meiklejohn's DéjàQ framework to give a reduction from subgroup hiding assumption to all

the q-type assumptions that belong to BTA family (in §4.2). As a consequence, we can prove the security of Fuchsbauer et al's set commitment scheme [FHS14] under subgroup hiding assumption, instead of generalized q-co-SDH and q-co-DL assumptions used in the original proof.
3. We construct the dual-form variant of Abe et al's structure-preserving signature scheme in §5.1 (resp. Boyen-Waters group signature scheme in §5.2) whose security is proved under subgroup hiding assumption instead of q-AGHO assumption (resp. q-HSDH assumption).

2 Preliminaries

2.1 Notation

Let $\boldsymbol{X}$ denote the vector representation of m monomials $(X_1, \ldots, X_m)$. The multivariate polynomial of degree $d \geq 0$ with m variables is denoted as $q(\boldsymbol{X}) = \sum a_{k_1,\ldots,k_m} X_1^{k_1} \cdots X_m^{k_m}$, where the summation is taken over all $k_i \in [0, d]$ such that $\sum_{i=1}^{m} k_i \leq d$. The polynomial $q(\boldsymbol{X})$ is represented using the coefficients $(a_{k_1,\ldots,k_m})_{\substack{k_i \in [0,d] \\ \sum_i k_i \leq d}}$. We denote $q(\boldsymbol{x})$ to be the polynomial $q(\boldsymbol{X})$ which is evaluated at $\boldsymbol{X} = \boldsymbol{x}$, for $\boldsymbol{x} \in \mathbb{Z}_p^m$. We also denote $x \xleftarrow{\$} G$ to be the element x which is chosen uniformly at random from the group G. Similarly, for any randomized algorithm A, $y \xleftarrow{\$} A(x)$ denotes the algorithm A which takes the value x from the appropriate domain and outputs y uniformly at random. For any function f, $f.\mathcal{D}$ denotes the domain of f. For any $n \in \mathbb{N}$, we denote $[1, n]$ is a collection of all the natural numbers lies between 1 to n.

2.2 Definitions

We first begin by recalling the definition of a bilinear group generator from [CM14].

Definition 1 *A bilinear group generator $\mathcal{G}$ is a probabilistic polynomial time (PPT) algorithm which takes the security parameter λ as input and outputs (N, G, H, G_T, e, μ), where N is either prime or composite, G, H and G_T are the groups such that $|G| = |H| = k_1 N$ and $|G_T| = k_2 N$ for $k_1, k_2 \in \mathbb{N}$, all the elements of G, H, G_T are of order atmost N and $e : G \times H \longrightarrow G_T$ is a bilinear map and it satisfies, (i) Bilinearity: For all $g, g' \in G$ and $h, h' \in H$, one has $e(g \cdot g', h \cdot h') = e(g, h) \cdot e(g, h') \cdot e(g', h) \cdot e(g', h')$, (ii) Non degeneracy: If a fixed $g \in G$ satisfies $e(g, h) = 1$ for all $h \in H$, then $g = 1$ and similarly for elements of H and (iii) Computability: The map e is efficiently computable. The additional information μ is optional and defined as follows. Whenever the groups G and H are cyclic, then μ contains their respective generators g and h. Whenever the groups G and H are decomposed into its cyclic subgroups $G_1, \ldots, G_n$ and $H_1, \ldots, H_n$ respectively, then μ contains the description of these subgroups and/or their generators.*

The bilinear group generator $\mathcal{G}$ is said to be composite-order (resp. prime-order), if N is composite (resp. prime). In this paper we use both prime-order and composite-order bilinear group generator simultaneously. Hence for the ease of readability, we use the following notation to differentiate between these two settings. In the prime-order setting, we denote $\mathbb{G}_1 = G$, $\mathbb{G}_2 = H$, $\mathbb{G}_T = G_T$ and we could obtain only trivial subgroups, hence μ contains the generators g and h of the respective groups $\mathbb{G}_1$ and $\mathbb{G}_2$. In the composite-order setting, we decompose the groups $G \cong G_1 \oplus \ldots \oplus G_n$ and $H \cong H_1 \oplus \ldots \oplus H_n$ for $N = p_1 \ldots p_n$ with μ containing required subgroup information i.e., μ contains $\{g_i, h_i\}_{i=1}^{n}$, where g_i (resp. h_i) is the generator of the subgroup G_i (resp. H_i).

Now we define the subgroup hiding assumption in the composite-order pairing and one can see that it is equivalent to the definition given by [CM14].

Definition 2 *For a composite-order bilinear group generator $\mathcal{G}$ which takes λ as input and outputs (N, G, H, G_T, e, μ). Now $\mathcal{G}$ is said to satisfy the subgroup hiding assumption in G for subgroup G_1 with respect to μ, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in the security parameters,*

$$Adv_{\mathcal{A}}^{SGH_G} = \left| Pr[\mathcal{A}(N, G, x, \mu) = 1 : x \in G] - Pr[\mathcal{A}(N, G, x, \mu) = 1 : x \in G_1] \right|$$

where $g_1 \in \mu$. Similarly we can define the subgroup hiding assumption in H.

From the above definition it is clear that the choice of μ might make the subgroup hiding easy. In the above definition, if μ contains h_2 then one can easily decide whether the given element x is from the subgroup G_1 or from the group G, by checking $e(x, h_2) \stackrel{?}{=} 1$. Without loss of generality we assume that μ does not contain such elements which are harmful against its hardness of the subgroup hiding assumption.

In this paper we will be using many of the computational parameterized (q-type) assumptions. We recall the definition of the assumptions in §A.

3 Bilinear Target Assumption and Its Extension

Boneh et al. [BBG05] introduced Uber assumption (we call it classical Uber) and argued its security in the generic group model. However Boyen [Boy08, Section 6] suggested (informally) to capture the Uber assumptions which have (a) challenge terms with adversary's input and b) rational polynomial representation in the exponent. In 2017, Ghadafi and Groth [GG17] formalized an assumption family which captures the above features in both cyclic group setting and in the bilinear group setting of prime-order. The first assumption type is said to be target assumption family. Whereas the second assumption is known as bilinear target assumption (BTA) family and here we focus on this assumption.

In this section we recall the definition of BTA family [GG17]. We identify some of the concrete computational assumptions that will not fall under this BTA family. Hence we extend the BTA definition to capture such computational assumptions. We also look at the possible relation among the BTA family and its extension.

First we fix some notation which will be used in this section. Let us denote the generator of the groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ by $[1]_1$, $[1]_2$ and $[1]_T$ respectively. Hence $[a]_1$ denote the group element having discrete logarithm of a with respect to its generator $[1]_1$ in $\mathbb{G}_1$. Similarly we denote $[a]_2$ (resp. $[a]_T$) for the group element in $\mathbb{G}_2$ (resp. $\mathbb{G}_T$). The group operation $[a]_1 \cdot [b]_1$ is denoted as $[a + b]_1$ in $\mathbb{G}_1$. For the other groups $\mathbb{G}_2$ and $\mathbb{G}_T$, we follow the similar notation. The pairing operation is denoted as $e([a]_1, [b]_2) = [ab]_T$.

3.1 Definition

Now we fix some notation to define the BTA assumption. As we know that the BTA is a computational assumption and is defined in the cyclic group of order p prime. Hence any group element can be written as its discrete logarithm value exponentiated with a fixed random generator of that group. Ghadafi and Groth [GG17] represented those exponent values using some multivariate rational polynomials of bounded degree. Let $\boldsymbol{X}$ be the indeterminates with m variables and $a(\boldsymbol{X})$, $b(\boldsymbol{X})$ denotes the multivariate polynomials of degree $d \geq 0$ over $\mathbb{Z}_p$. For randomly chosen $\boldsymbol{x}$ from $\mathbb{Z}_p^m$ such that $b(\boldsymbol{x}) \neq 0$, we denote $\left[\frac{a(\boldsymbol{x})}{b(\boldsymbol{x})}\right]_j$ be the group element from $\mathbb{G}_j$ having exponent which is represented using the rational polynomial $\frac{a(\boldsymbol{X})}{b(\boldsymbol{X})}$ evaluated at $\boldsymbol{X} = \boldsymbol{x}$, for $j \in \{1, 2, T\}$. Since the polynomials are represented using the coefficients, we denote that $[a(\boldsymbol{X})]_i$ (resp. $[b(\boldsymbol{X})]_i$) be the coefficient representation of the polynomial $a(\boldsymbol{X})$ (resp. $b(\boldsymbol{X})$) in the group $\mathbb{G}_i$, for $i, j \in \{1, 2, T\}$.

Recall the BTA assumption [GG17] in which exponent of both problem instance and challenge term are represented using rational polynomials and all the polynomial coefficients are given explicitly as $\mathbb{Z}_p$ elements. More formally we define as follows.

Assumption 1 ***BTA.*** *Let $\Theta = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be the output of a bilinear group generator $\mathcal{G}$ on the input λ. For $\iota \in \{1, 2, T\}$, $\mathcal{G}$ is said to satisfy bilinear target assumption [GG17] in $\mathbb{G}_\iota$, if for every PPT adversary $\mathcal{A}$, the advantage of BTA is defined as,*

$$Adv_{\mathcal{A}}^{BTA_{\mathbb{G}_\iota}} := Pr[\mathcal{A}(\Gamma) \rightarrow \Delta : \Delta \text{ satisfies Equation 1}] = negl(\lambda),$$

where the problem instance Γ and the challenge term Δ are defined as

$$\Gamma = \left(\Theta, \left\{\left\{\left[\frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})}\right]_j, \frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})}\right\}_{i=1}^{n_j}\right\}_{j\in\{1,2,T\}}, \mathtt{pub}\right) \text{ and } \Delta = \left(\left[\frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}\right]_\iota, r(\boldsymbol{X}), s(\boldsymbol{X}), \mathtt{sol}\right).$$

The condition is defined as,

$$r(\boldsymbol{X})\prod_{i=1}^{n_\iota} b_i^{(\iota)}(\boldsymbol{X}) \notin Span\left(\left\{s(\boldsymbol{X})a_i^{(\iota)}(\boldsymbol{X})\prod_{l\neq i} b_l^{(\iota)}(\boldsymbol{X})\right\}_{i=1}^{n_\iota}\right). \tag{1}$$

The condition from Equation 1 is used to avoid the trivial attacks due to generic group operations. The above defined BTA assumption is parameterized with (d, m, n_1, n_2, n_T), where d denotes the degree of polynomials (from both problem instance and challenge terms[1]) and m denotes the number of indeterminates in $\boldsymbol{X}$ and for $j \in \{1, 2, T\}$, n_j denotes the total number of elements from $\mathbb{G}_j$ which are present in the problem instance. Once the parameter is clear from the context, for simplicity we ignore this parameter. In the above definition, `pub` contains all the coefficients of the polynomials presented in the problem instance and `sol` contains some additional information in order to validate the challenge term. The secret vector $\boldsymbol{x}$ that are used in the assumption should not be given explicitly as part of the problem instance.

Example 1 *We recall the q-co-SDH problem [FHS14] defined in Table 1: given the instance* $\left(\left\{[1]_j, \left\{[x^i]_j\right\}_{i=1}^{q}\right\}_{j=1}^{2}\right)$ *compute* $\left(r(X), s(X), \left[\frac{r(x)}{s(x)}\right]_1\right)$ *such that* $0 \leq deg\ r(X) < deg\ s(X) \leq q$. *Note that this assumption is same as q-bilinear simple fractional assumption (BSFrac) [GG17] defined in* $\mathbb{G}_1$. *We represent the exponent values as a polynomial in X which is evaluated at* $X = x$. *Hardness of this problem ensures that the challenge term satisfies Equation 1. Thus q-co-SDH assumption belongs to BTA family with* $d = q$, $m = 1$, $n_1 = n_2 = q + 1$, $n_T = 0$. □

Similarly it is easy to check that the assumptions such as q-Diffie-Hellman inversion (DHI), q-Diffie-Hellman exponent (DHE) q-modified SDH (mSDH), q-modified double SDH (mDSDH) and BB-CDH (see Table 1) are examples for BTA family, since all the polynomial coefficients of both problem instance and challenge term are given explicitly.

3.2 BTA Extension

Recall that in the BTA definition all the polynomial coefficients in both problem instance and challenge term are given explicitly. However there are many assumptions in which not all the polynomial coefficients from problem instance and challenge terms are given explicitly. Some examples of such assumptions are q-HSDH, q-SFP, q-Triple Diffie-Hellman (TDH), q-simultaneous pairing (SP). See the complete list of such assumptions in Table 1.

Before defining the variants of BTA we observe that we could extend Ghadafi and Groth's BTA definition by including more number of challenge terms, in particular polynomial number of terms. However one can see that this extension is equivalent to the original BTA assumption. As BTA implies to this new variant is trivial and for the other direction simply run the BTA solver in polynomial many times.

As in BTA, each element from problem instance and challenge terms are written as its discrete logarithm with respect to some fixed generator of the cyclic group of prime-order. We follow this notion throughout this paper. Thus expressing computational assumption in this format allows us to classify these assumptions into appropriate BTA family and its extension such as BTA1 and BTA2 families defined as follows.

First we motivate the definition of BTA1 with a concrete assumption. Recall that, Abe et al. [AGHO11] defined a variant of q-AGHO assumption defined in Table 1. In the following example we show that it belongs to BTA1 family.

[1] For BTA in $\mathbb{G}_T$, the challenge term polynomials degree are bounded by $2d$, as given the d degree polynomials in both source groups, one can use the pairing to compute the product of these polynomials in $\mathbb{G}_T$.

Example 2 *We recall the q-AGHO′ problem defined in Table 1: given* $\Big([1]_1,[1]_2,[w]_2,[x]_2,[y]_2,\Big\{[x-a_iw-r_iy]_1,[a_i]_1,[r_i]_1,[a_i^{-1}]_2\Big\}_{i=1}^{q}\Big)$ *compute* $\big([x-a^*w-r^*y]_1,[a^*]_1,[r^*]_1,[(a^*)^{-1}]_2\big)$*. As in Example 1, the exponent values are represented using polynomials in* W*,* X *and* Y *which are evaluated at* $W=w$*,* $X=x$ *and* $Y=y$*. The exponent values such as* a_i, r_i *from the instance and* a^*, r^* *from the challenge terms are the coefficients of the polynomials. In this assumption, none of the polynomial coefficients are given explicitly rather given in the exponent of the source group element. Here the parameters can be computed with* $d=1$*,* $m=3$*,* $n_1=3q+1$*,* $n_2=q+4$*,* $n_{c_1}=3$*,* $n_{c_2}=1$ *and* $n_T=n_{c_T}=0$*, where* n_{c_j} *denotes the total number of challenge terms in* $\mathbb{G}_j$*, for* $j\in\{1,2,T\}$*.* □

From the above example, we define a new family of BTA variant called BTA1, in which not all the polynomial coefficients in both problem instance and challenge terms are given explicitly, rather given in the exponent of some source group element. In this paper we focus on the BTA1 family defined only in the source groups, since all the parameterized assumptions described in Table 1 are defined in the source groups.

Assumption 2 ***BTA1.*** *Let* $\Theta=(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e)\xleftarrow{\$}\mathcal{G}(\lambda)$*. For* $\iota\in[1,2]$*,* $\mathcal{G}$ *is said to satisfy bilinear target assumption-1 (BTA1) in* $\mathbb{G}_\iota$*, if for every PPT adversary* $\mathcal{A}$*, the advantage of BTA1 is defined as,* $Adv_{\mathcal{A}}^{BTA1_{\mathbb{G}_\iota}}:=Pr[\mathcal{A}(\Gamma)\rightarrow\Delta:\Delta$ *satisfies either Equation 2 or 3*$]=negl(\lambda)$*, where the problem instance* Γ *is defined as*

$$\Bigg(\Theta,\Bigg\{\Bigg\{\Bigg[\frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})}\Bigg]_j,\Big(\Big\{\Big[a_i^{(j)}(\boldsymbol{X})\Big]_{j_a}\Big\}_{j_a=1}^{2}\ or\ a_i^{(j)}(\boldsymbol{X})\Big),\Big(\Big\{\Big[b_i^{(j)}(\boldsymbol{X})\Big]_{j_b}\Big\}_{j_b=1}^{2}\ or\ b_i^{(j)}(\boldsymbol{X})\Big)\Bigg\}_{i=1}^{n_j}\Bigg\}_{j\in\{1,2,T\}},\mathtt{pub}\Bigg)$$

and the challenge terms Δ *is defined as*

$$\Bigg(\Bigg\{\Bigg[\frac{r_t^{(\iota)}(\boldsymbol{x})}{s_t^{(\iota)}(\boldsymbol{x})}\Bigg]_\iota,\Big(\Big\{\Big[r_t^{(\iota)}(\boldsymbol{X})\Big]_{\iota_r}\Big\}_{\iota_r=1}^{2}\ or\ r_t^{(\iota)}(\boldsymbol{X})\Big),\Big(\Big\{\Big[s_t^{(\iota)}(\boldsymbol{X})\Big]_{\iota_s}\Big\}_{\iota_s=1}^{2}\ or\ s_t^{(\iota)}(\boldsymbol{X})\Big)\Bigg\}_{t=1}^{n_{c_\iota}},\mathtt{sol}\Bigg).$$

The condition is stated as follows. There exists $t\in[1,n_{c_\iota}]$ *with atleast one of the following condition should satisfy, either* $\big[r_t^{(\iota)}(\boldsymbol{X})\big]_\iota$ *or* $\big[s_t^{(\iota)}(\boldsymbol{X})\big]_\iota$ *or*

$$\Bigg[\frac{r_t^{(j)}(\boldsymbol{X})}{s_t^{(j)}(\boldsymbol{X})}\Bigg]_\iota\notin Span\Bigg(\Bigg\{\Bigg[\frac{a_i^{(\iota)}(\boldsymbol{X})}{b_i^{(\iota)}(\boldsymbol{X})}\Bigg]_\iota\Bigg\}_{i=1}^{n_\iota},\Big\{\Big[a_{i_1}^{(\iota_1)}(\boldsymbol{X})\Big]_\iota,\Big[b_{i_2}^{(\iota_2)}(\boldsymbol{X})\Big]_\iota\Big\}_{\substack{\iota_1,\iota_2\in\{1,2\}\\ i_1,i_2\in[1,n_1]\cup[1,n_2]}}\Bigg)\tag{2}$$

and for $\varsigma=3-\iota$*, either* $\big[r_t^{(\varsigma)}(\boldsymbol{X})\big]_\varsigma$ *or*

$$\big[s_t^{(\varsigma)}(\boldsymbol{X})\big]_\varsigma\notin Span\Bigg(\Bigg\{\Bigg[\frac{a_i^{(\varsigma)}(\boldsymbol{X})}{b_i^{(\varsigma)}(\boldsymbol{X})}\Bigg]_\varsigma\Bigg\}_{i=1}^{n_\varsigma},\Big\{\Big[a_{i_1}^{(\iota_1)}(\boldsymbol{X})\Big]_\varsigma,\Big[b_{i_2}^{(\iota_2)}(\boldsymbol{X})\Big]_\varsigma\Big\}_{\substack{\iota_1,\iota_2\in\{1,2\}\\ i_1,i_2\in[1,n_1]\cup[1,n_2]}}\Bigg).\tag{3}$$

This BTA1 family is parameterized with $(d,m,n_1,n_2,n_T,n_{c_1},n_{c_2},n_{c_T})$, where n_{c_j} denotes the total number of challenge terms from $\mathbb{G}_j$, for $j\in\{1,2,T\}$ and the remaining parameters are defined as in BTA family. The condition from Equations 2 and 3 are used to avoid the trivial attacks due to generic group operation. As some of the challenge terms are given as in the exponent instead of $\mathbb{Z}_p$ element, this will have more flexible to mount some trivial attacks due to generic group operations. As a concrete example, we explain this attack for the variants of q-SDH assumption described in the following remark.

Remark 1 *Consider the following, given* $\Big(\big\{[1]_j,[x]_j\big\}_{j=1}^{2},\big\{\big[\frac{1}{x+a_i}\big]_1,[a_i]_1,[a_i]_2\big\}_{i=1}^{q}\Big)$ *whether can we compute the challenge terms* $\big(\big[\frac{1}{x+a}\big]_1,[a]_1\big)$ *or not. As in Examples 1, we represent the exponent values as polynomials in* X *which are evaluated at* $X=x$*. Since the polynomial coefficients of the challenge*

term is given in the exponent of first source group element, anyone can break this assumption by computing $\left[\frac{1}{x+a}\right]_1 = [x]_1[a_i]_1$ *and* $[a]_1 = \left[\frac{1}{x+a_i}\right]_1/[x]_1$ *for some* $i \in [1,q]$*. This attack is captured in Equation 2 by checking* $[s_1^{(1)}(\boldsymbol{X})]_1 = [a]_1$ *is in the space spanned by the elements* $\left[\frac{1}{x+a_i}\right]_1$ *and* $[x]_1$ *from the problem instance.*

Now we consider the q-2SDH$_S$ problem [Oka06a], given $\left(\left\{[1]_j, [x]_j, [y]_j\right\}_{j=1}^{2}, \left\{\left[\frac{y+b_i}{x+a_i}\right]_j, [a_i]_j, b_i\right\}_{i=1}^{q}\right)$ *whether can we compute* $\left(\left[\frac{y+d}{x+c}\right]_1, [c]_1, [c]_2, d\right)$*, for* $d \neq b_i$*. One can break this problem by computing* $\left[\frac{y+d}{x+c}\right]_1 = [y]_1[d]_1$*,* $[c]_1 = [1-x]_1$ *and* $[c]_2 = [1-x]_2$*, for* $d \xleftarrow{\$} \mathbb{Z}_p$ *with* $d \neq b_i$*, for all* $i \in [1,q]$*.*

From the definition of BTA1, it is easy to see that assumptions such as q-HSDH, q-ADHSDH, q-SFP and q-AGHO (defined in Table 1) belong to BTA1 family, since not all the polynomial coefficients in both problem instance and challenge terms are given explicitly.

Now consider the BB-HSDH assumption [BCC$^+$09] in which all the polynomial coefficients of the problem instance are given explicitly, whereas all the polynomial coefficients of the challenge terms are given in the exponent of both source groups. Thus BB-HSDH assumption will not fall under BTA1 family. This motivate us to define the other variant of BTA family, called BTA2. In this family, all the polynomial coefficients of the problem instance are given explicitly, whereas not all the polynomial coefficients of the challenge terms are given explicitly. There are many assumptions such as q-TDH, q-SP, (q,ℓ,ℓ')-Pluri-SDH and (q,ℓ)-Poly-SDH (defined in Table 1) fall in this family.

Assumption 3 ***BTA2****: Let* $\Theta = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \xleftarrow{\$} \mathcal{G}(\lambda)$*. For* $\iota \in [1,2]$*,* $\mathcal{G}$ *is said to satisfy bilinear target assumption-2 (BTA2) in* $\mathbb{G}_\iota$*, if for every PPT adversary* $\mathcal{A}$*, the advantage of BTA2 is defined as,* $Adv_{\mathcal{A}}^{BTA2_{\mathbb{G}_\iota}} := Pr[\mathcal{A}(\Gamma) \to \Delta : \Delta \text{ satisfies either Equation 2 or 3}] = negl(\lambda)$*, where the problem instance* Γ *is defined as*

$$\left(\Theta, \left\{\left\{\left[\frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})}\right]_j, a_i^{(j)}(\boldsymbol{X}), b_i^{(j)}(\boldsymbol{X})\right\}_{i=1}^{n_j}\right\}_{j\in\{1,2,T\}}, \texttt{pub}\right)$$

and challenge terms Δ *is defined as*

$$\left(\left\{\left[\frac{r_t^{(\iota)}(\boldsymbol{x})}{s_t^{(\iota)}(\boldsymbol{x})}\right]_\iota, \left(\left\{\left[r_t^{(\iota)}(\boldsymbol{X})\right]_{\iota_r}\right\}_{\iota_r=1}^{2} \text{ or } r_t^{(\iota)}(\boldsymbol{X})\right), \left(\left\{\left[s_t^{(\iota)}(\boldsymbol{X})\right]_{\iota_s}\right\}_{\iota_s=1}^{2} \text{ or } s_t^{(\iota)}(\boldsymbol{X})\right)\right\}_{t=1}^{n_{c_\iota}}, \texttt{sol}\right).$$

The condition for this assumption remains same as in BTA1, as the polynomial coefficients of problem instance are given explicitly, anyone can exponentiate these coefficients in any of the groups.

As similar to BTA1, BTA2 family is parameterized with $(d, m, n_1, n_2, n_T, n_{c_1}, n_{c_2}, n_{c_T})$ and condition from Equations 2 and 3 avoids the trivial attacks due to generic group operations. In order to understand the importance of Equations 2 and 3, we consider the following assumption. Given $\left(\left\{[1]_j, [x]_j\right\}_{j=1}^{2}, \left\{\left[\frac{1}{x+a_i}\right]_1, a_i\right\}_{i=1}^{q}\right)$ whether can we compute the challenge terms $\left(\left[\frac{1}{x+a}\right]_1, [a]_1, [a]_2\right)$ or not. As similar to Remark 1, one can break this assumption using generic group operation.

Also we observe that there are assumptions in which challenge terms do not output all the polynomial coefficients even in the exponent of a group element. As a concrete example we describe the (q,ℓ)-Poly-SDH assumption and we show that it belongs to BTA2 family.

Example 3 *Recall that* (q,ℓ)*-Poly-SDH problem defined in Table 1: given the instance* $\left([1]_1, [1]_2, \left\{[x_i]_1, [x_i]_2\right\}_{i=1}^{\ell}, \left\{[\frac{1}{x_i+c_{ij}}]_1, c_{ij}\right\}_{i,j=1}^{\ell,q}\right)$ *compute* $\left(\left\{[\frac{\gamma_i}{x_i+c_i}]_1, c_i\right\}_{i=1}^{\ell}\right)$ *such that* $\sum_{i=1}^{\ell}\gamma_i = 1$*. As similar to previous examples, the exponent values which are having terms like* x_i *are represented as polynomials in* $\{X_i\}_{i=1}^{\ell}$ *which are evaluated at* $X_i = x_i$ *and all the remaining exponent values are the coefficients of the polynomials. In this assumption, none of the numerator's polynomial coefficients of the challenge terms (i.e.,* γ_i*) are given explicitly. However the condition* $\sum_{i=1}^{\ell}\gamma_i = 1$ *is included as part of* sol*, which ensure the well-formedness of the challenge terms by* $\prod_{i=1}^{\ell} e\left(\left[\frac{\gamma_i}{x_i+c_i}\right]_1, [x_i]_2[c_i]_2\right) \stackrel{?}{=} e([1]_1, [1]_2)$*. The hardness of this assumption [Boy07] makes sure that it also satisfy the Equations 2 and 3. Thus* (q,ℓ)*-Poly-SDH assumption belongs to BTA2 family.* □

3.3 Relation among BTA variants

In this section we discuss the relation among newly defined variants of BTA assumptions. Since BTA is a family of assumptions, Ghadafi and Groth used the following notion to prove the reduction. For any assumption $\mathcal{P}$ there exists a assumption $\mathcal{Q}$ such that $\mathcal{Q}$ implies $\mathcal{P}$. Using this reduction, we prove that the assumptions in BTA2 family could be a possible candidate Uber assumption as compared to the assumptions in BTA and BTA1 families. We emphasize that while describing BTA and its variants we use both assumption and family interchangeably.

Lemma 1 *We prove that (i) for any (d, m, n_1, n_2, n_T)-BTA assumption, there exists $(d, m, n_1, n_2, n_T, n_{c_1}, n_{c_2}, n_{c_T})$-BTA2 assumption such that BTA2 implies BTA, (ii) for any $(d, m, n_1, n_2, n_T, n_{c_1}, n_{c_2}, n_{c_T})$-BTA1 assumption, there exists $(d, m, n_1, n_2, n_T, n_{c_1}, n_{c_2}, n_{c_T})$-BTA2 assumption such that BTA2 implies BTA1.*

Proof of this lemma can be found in §B.

Now we observe that it is difficult to give the following reductions. In particular, reductions from BTA to BTA1 (resp. BTA1 to BTA) and from BTA to BTA2 (resp. BTA1 to BTA2) are difficult to prove, as it require to compute discrete logarithm for the challenge terms (resp. problem instance) in the appropriate groups.

4 BTA in DéjàQ Framework

In this section we prove that subgroup hiding implies all the q-type assumptions that belong to bilinear target assumption (BTA) family. Recall that Chase-Meiklejohn's [CM14] DéjàQ framework ensure the reduction from SGH to q-SDH and q-generalized Diffie-Hellman exponent (q-GDHE) assumptions. However they did not consider the assumptions such as q-co-SDH, q-mDSDH and BB-CDH. We notice that q-co-SDH assumption was used to prove the security of Fuchsbauer et al's set commitment scheme [FHS14]. With our knowledge no literature proved that q-co-SDH assumption is implied by SGH assumption. As we have seen that all these q-type assumptions belonging to BTA family. See the Table 1 for some concrete assumptions. Hence it is worth pursuing that for the parameterized assumptions that belong to BTA family, can we give a reduction from SGH assumption using DéjàQ techniques.

First we formalize the extended adaptive parameter-hiding property and use this property in Chase-Meiklejohn's DéjàQ techniques [CM14]. We also discuss in applicability of the existing DéjàQ techniques for the concrete q-type assumptions that fall in either BTA1 or BTA2 family.

4.1 Extended Adaptive Parameter-Hiding Property

The parameter-hiding is a statistical property which ensures that the elements in one subgroup should not reveal anything about related elements in other subgroups. Chinese Remainder Theorem (CRT) ensures the same in the composite-order pairing setting. Lewko [Lew12] informally used parameter-hiding property to convert Lewko-Waters IBE scheme from composite-order to prime-order pairing. In 2014, Chase and Meiklejohn [CM14] defined parameter-hiding property for any polynomial function in the composite-order setting and used it to prove SGH implies decisional q-type assumption which are one-sided[2], such as exponent q-SDH assumption [ZSS04]. Also they defined *extended parameter-hiding* property and used it to prove SGH implies the computational q-type assumptions which are two-sided[2], such as q-SDH assumption. Informally, this property says that the distributions $\{g_1^{f(\boldsymbol{x})} g_2^{f(\boldsymbol{x})}\}$ and $\{g_1^{f(\boldsymbol{x})} g_2^{f(\boldsymbol{x}')}\}$ are identical, even if some auxiliary informations are given in the exponent of h_1. The other interesting definition by Chase and Meiklejohn is the *adaptive parameter-hiding* property.

[2] We say that the BTA assumption defined in the asymmetric pairing is said to be one-sided, if the secret vector $\boldsymbol{x}$ associated with the polynomial representation occurs in exactly one of the source group. Otherwise we say that the assumption is two-sided.

Informally, this property ensures that any unbounded adversary who makes only polynomial number of queries can statistically indistinguish between the distributions $\{g_1^{f(\boldsymbol{x})} g_2^{f(\boldsymbol{x})}\}$ and $\{g_1^{f(\boldsymbol{x})} g_2^{f'(\boldsymbol{x})}\}$, for any f, f' from the family of functions $\mathcal{F}$. In particular, they have used this property for rational polynomial function of the form $\frac{1}{x+c}$ with c being chosen by the adversary.

Now we consider the computational q-type assumptions that belong to BTA family which are two-sided in which all the polynomial coefficients of the challenge terms are chosen by the adversary. Hence it is natural to use the adaptive parameter-hiding property along with some auxiliary information. We note that this has been already pointed out by Chase and Meiklejohn [CM14, footnote 5] to prove SGH implies q-SDH assumption. Similarly we can use the adaptive parameter-hiding property for the computational q-type assumptions which are one-sided. Now we formally define this property for any function as follows.

Definition 3 *Let $\mathcal{G}$ be a bilinear group generator and functions f, f' are chosen at random from a family of functions $\mathcal{F}$. Let Aux denote the auxiliary information. Let $\mathcal{O}(\cdot)$ be the oracle and it returns $g_1^{f(\cdot)} g_2^{f(\cdot)}$ if the input is in the domain $f.\mathcal{D}$ and 1 otherwise. Similarly, let $\mathcal{O}'(\cdot)$ be the oracle and it returns $g_1^{f(\cdot)} g_2^{f'(\cdot)}$ if the input is in the domain $f.\mathcal{D} \cap f'.\mathcal{D}$ and 1 otherwise. Then we say that $\mathcal{G}$ satisfies extended adaptive parameter-hiding with respect to $\mathcal{F}$ and Aux, if for all $\Theta = (N, G, H, G_T, e, \mu) \xleftarrow{\$} \mathcal{G}(\lambda)$[3] with $\mu = \{g_1, g_2\}$ where $g_1 \in G_1$, $g_2 \in G_2$ and $G \cong G_1 \oplus G_2$, the oracles $\mathcal{O}$ and $\mathcal{O}'$ are statistically indistinguishable, if they are given with auxiliary information Aux and queried polynomially many times. In other words, for any unbounded adversary $\mathcal{A}$ that makes poly(λ) queries, there exists a negligible function $\nu(\cdot)$ such that*

$$\left| Pr[f \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{\mathcal{O}(\cdot)}(\Theta, \mathsf{Aux}) = 1] - Pr[f, f' \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{\mathcal{O}'(\cdot)}(\Theta, \mathsf{Aux}) = 1] \right| < \nu(\cdot).$$

We emphasize that the above definition is applicable for any function, in particular they can be applied for rational polynomial functions in the following way. Thus we consider the functions f and f' which take rational polynomial coefficients as input and evaluate on some random vectors $\boldsymbol{x}$ and $\boldsymbol{x'}$ from $\mathbb{Z}_N^m$, i.e., the function f is defined as $f(r(\boldsymbol{X}), s(\boldsymbol{X})) := \frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}$ and f' is defined as $f'(r(\boldsymbol{X}), s(\boldsymbol{X})) := \frac{r(\boldsymbol{x'})}{s(\boldsymbol{x'})}$, where $r(\boldsymbol{X})$ and $s(\boldsymbol{X})$ denote the coefficient representation of the polynomials of degree d (defined over $\mathbb{Z}_N$) with m many monomials. As we know that in the BTA family, adversarial inputs determine the rational polynomials of the challenge term as its coefficients. Hence we can apply the extended adaptive parameter-hiding property for this rational polynomials. Also we consider the auxiliary information as $\mathsf{Aux} = \{h_1^{\zeta^{(j)}(\boldsymbol{x})}\}_{j=2,T}$ for BTA assumption defined in G, where $\zeta^{(j)}(\boldsymbol{x}) \in \left\{ \frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})} \right\}_{i=1}^{n_j}$.

4.2 SGH implies BTA

In this section we prove that all the q-type assumptions that belong to BTA family defined over composite-order pairing can be reduced from SGH assumption. This reduction uses the extended adaptive parameter-hiding property defined in Definition 3. As mentioned earlier, instead of polynomial function we apply this property for rational polynomial function.

For the q-type assumption that belongs to BTA family, it is guaranteed from the BTA definition that atleast one of the parameter from $\{n_1, n_2, n_T\}$ can be written as some function of q, where $q = poly(\lambda)$. Now we consider the BTA assumption defined in G. As a concrete example, we consider q-co-SDH assumption described in Example 1 that belongs to BTA family with $n_1 = n_2 = q + 1$ and $n_T = 0$. Now without loss of generality, it is sufficient to consider the BTA assumption defined in G with n_1 being expressed as some function of q. Since the other possible choices can be covered by constructing a strong assumption using the polynomials of the other group exponents from the weaker assumption and then apply the DéjàQ techniques on this stronger assumption. For example, we consider a BTA assumption (say $\mathcal{P}1$) defined in G with n_1 and n_T are some constants but n_2 is expressed as some

[3] Even if $N = p_1 \dots p_n$, we decompose G using two of its subgroups G_1 and G_2 such that G_1 (resp. G_2) is a subgroup of order $p_1 \dots p_{n-1}$ (resp. p_n).

function of q (say $n_2(q)$). First one can construct a stronger assumption (say $\mathcal{P}2$) from $\mathcal{P}1$ assumption by including all its $n_2(q)$ many exponents of H component to the exponent of G in the $\mathcal{P}2$ assumption.

Now we proceed with Chase-Meiklejohn's DéjàQ framework along with the extended adaptive parameter-hiding property on BTA assumption with n_1 being expressed as some function of q. First we define a variant of BTA assumption, which will be useful while proving SGH implies BTA assumption.

Assumption 4 *Let $\Theta = (N, G, H, G_T, e, \mu) \xleftarrow{\$} \mathcal{G}(\lambda)$ with $\mu = \{G_1, G_2\}$. $\mathcal{G}$ is said to satisfy a variant of bilinear target assumption (vBTA)[4] in G, if for every PPT adversary $\mathcal{A}$ and for all $\ell = poly(\lambda)$, the advantage of this assumption is defined as, $Adv_{\mathcal{A}}^{vBTA_{G_\iota}} := Pr[\mathcal{A}(\Gamma) \to \Delta] = negl(\lambda)$, where*

$$\Gamma = \Bigg(\Theta, g_1 g_2^{\sum_{i=1}^{\ell} r_i}, h_1, \left\{ g_1^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}} g_2^{\sum_{j=1}^{\ell} r_j \frac{a_i^{(1)}(\boldsymbol{x}_j)}{b_i^{(1)}(\boldsymbol{x}_j)}} \right\}_{i=1}^{n_1}, \left\{ h_1^{\frac{a_i^{(2)}(\boldsymbol{x})}{b_i^{(2)}(\boldsymbol{x})}} \right\}_{i=1}^{n_2},$$
$$\left\{ e(g_1, h_1)^{\frac{a_i^{(T)}(\boldsymbol{x})}{b_i^{(T)}(\boldsymbol{x})}} \right\}_{i=1}^{n_T}, \left\{ \left\{ \frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})} \right\}_{i=1}^{n_j} \right\}_{j \in \{1,2,T\}}, \mathsf{pub} \Bigg),$$

for $g_1 \xleftarrow{\$} G_1$, $g_2 \xleftarrow{\$} G_2 \setminus \{1\}$ and $r_j \xleftarrow{\$} \mathbb{Z}_N$, $\boldsymbol{x}, \boldsymbol{x}_j \xleftarrow{\$} \mathbb{Z}_N^m$ and the output Δ is $\left(g_1^{\frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}} g_2^{\sum_{j=1}^{\ell} r_j \frac{r(\boldsymbol{x}_j)}{s(\boldsymbol{x}_j)}}, r(\boldsymbol{X}), s(\boldsymbol{X}) \right)$ and sol.

Now we prove that BTA assumption defined in G is implied by Assumption 4 using subgroup hiding assumption and extended adaptive parameter-hiding property.

Theorem 2 *For a bilinear group generator $\mathcal{G}(\lambda) \xrightarrow{\$} (N, G, H, G_T, e, \mu)$, consider $\mathcal{G}$ satisfies (d, m, n_1, n_2, n_T) bilinear target assumption in G. Suppose that if $\mathcal{G}$ satisfies the following, (i) subgroup hiding assumption for subgroup G_1 with respect to $\mu = \{g_2, h_1\}$ and for subgroup H_1 with respect to $\mu = \{g_1\}$ and (ii) extended adaptive parameter-hiding with respect to*

$$\mathcal{F} = \left\{ \left\{ \frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})} \right\}_{i=1}^{n_1}, \frac{r(\boldsymbol{x})}{s(\boldsymbol{x})} \right\} \text{ and } \mathsf{Aux} = \left\{ h_1^{\zeta} \right\}_{\zeta \in \left\{ \left\{ \frac{a_i^{(2)}(\boldsymbol{x})}{b_i^{(2)}(\boldsymbol{x})} \right\}_{i=1}^{n_2}, \left\{ \frac{a_i^{(T)}(\boldsymbol{x})}{b_i^{(T)}(\boldsymbol{x})} \right\}_{i=1}^{n_T} \right\}}$$

for any $h_1 \in H_1$ and if G_2 is of prime-order, then the BTA assumption is implied by the Assumption 4.

Proof sketch. The detailed proof can be found in §C and it uses the hybrid argument using sequence of games. The intuitive argument is as follows, consider the BTA assumption defined over composite-order bilinear groups, first translate all the elements from the group of composite-order to its subgroup G_1. Thus the elements of G and H are shifted to subgroups G_1 and H_1 and this shifting goes unnoticed under subgroup hiding in G and H respectively. Notice that the challenge term of BTA belongs to the group G, as BTA is defined in G. Since the exponent of the group elements are interpreted as rational polynomials that are evaluated at some secret vector $\boldsymbol{x}$, then the translation of elements from G_1 into G_2 retains the same polynomial evaluation as its shadow copy in the exponent of G_2. This transition is unnoticed under subgroup hiding in G. Now the shadow copy of the rational polynomials that corresponds to the subgroup G_2's exponents are evaluated using different secret vector $\boldsymbol{x}_1$ and is statistically identical to its previous state. This transition is achieved by using the extended adaptive parameter-hiding property defined in Definition 3. We repeat the above procedure polynomial many times (say ℓ) and prove the theorem. □

Now we provide the enough entropy by taking ℓ to be $n_1 + 2$ which will guarantee the hardness of Assumption 4 in the following corollary. This implies that the BTA assumption is reduced from SGH assumption.

[4] As similar to BTA assumption, hardness of vBTA assumption ensures that the instance and challenge terms should satisfy certain linearly independent condition that corresponds to Equation 1. However we omit such condition here and prove the hardness of vBTA assumption in Corollary 3.

Corollary 3 *For a bilinear group generator* $\mathcal{G}(\lambda) \xrightarrow{\$} (N, G, H, G_T, e, \mu)$*, we prove that* $\mathcal{G}$ *satisfies* (d, m, n_1, n_2, n_T)*-BTA assumption in* G*, if (i)* $N = p_1 \dots p_n$ *for distinct primes* $p_1, \dots, p_n \in \Omega(2^\lambda)$ *and* $\mathcal{G}$ *satisfies the following, (ii) subgroup hiding for subgroup* G_1 *with respect to* $\mu = \{g_2, h_1\}$ *and for subgroup* H_1 *with respect to* $\mu = \{g_1\}$*, (iii) extended adaptive parameter-hiding with respect to class* $\mathcal{F}$ *and* Aux *which are defined as in Theorem 2 and (iv) the polynomials in* $\mathcal{F}$ *are linearly independent and have maximum degree poly(*λ*).*

Proof. From the requirements (ii) and (iii), Theorem 2 tell us that BTA assumption is implied by the Assumption 4. In order to prove this corollary, it is sufficient to prove that the advantage of the Assumption 4 is negligible in the security parameter. Now for the sake of simplicity we assume that g_1 and $\boldsymbol{x}$ are public, hence adversary can compute the G_1 component of any challenge term. Now this boils down to computing $g_2^{\sum_{j=1}^{\ell} r_j \frac{r(\boldsymbol{x}_j)}{s(\boldsymbol{x}_j)}}$. Also note that the auxiliary information Aux doesn't provide any advantage in computing the above element, since they operate on different groups with completely independent set of variables. Consider the following matrix from the G_2 component of Assumption 4,

$$V = \begin{pmatrix} 1 & \frac{a_1^{(1)}(\boldsymbol{x}_1)}{b_1^{(1)}(\boldsymbol{x}_1)} & \cdots & \frac{a_{n_1}^{(1)}(\boldsymbol{x}_1)}{b_{n_1}^{(1)}(\boldsymbol{x}_1)} & \frac{r(\boldsymbol{x}_1)}{s(\boldsymbol{x}_1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \frac{a_1^{(1)}(\boldsymbol{x}_\ell)}{b_1^{(1)}(\boldsymbol{x}_\ell)} & \cdots & \frac{a_{n_1}^{(1)}(\boldsymbol{x}_\ell)}{b_{n_1}^{(1)}(\boldsymbol{x}_\ell)} & \frac{r(\boldsymbol{x}_\ell)}{s(\boldsymbol{x}_\ell)} \end{pmatrix}.$$

Here we set ℓ as $n_1 + 2$. From the requirement (iv), [CM14, Lemma 4.4] ensures that the above matrix V is nonsingular. For randomly chosen vector $\boldsymbol{r}$ from $\mathbb{Z}_N^{n_1+2}$ (it was chosen during the simulation), we define $\boldsymbol{y} = \boldsymbol{r} \cdot V$. Thus from this matrix relation, the first $n_1 + 1$ elements are given to $\mathcal{A}$ and his goal is to compute the last element of $\boldsymbol{y}$. Since V is invertible and $\boldsymbol{r}$ is chosen uniformly at random from $\mathbb{Z}_N^{n_1+2}$. This makes the vector $\boldsymbol{y}$ uniformly at random, in particular the last element is uniformly distributed. Hence probability of computing such challenge term is negligible in the security parameter. □

Remark 2 *We can also prove that SGH implies all the q-type assumptions belonging to BTA assumption defined in target group* G_T*. This can be further simplified by Ghadafi and Groth [GG17] result, as they proved that all the BTA assumption in* G_T *are implied by bilinear simple fractional (*BSFrac_T*) and bilinear gap (BGap) assumptions defined in* G_T*. Now we can use Chase et al's [CMM16] techniques to give a reduction from SGH assumption to both q-BGap and q-BSFrac assumptions. In particular for the reduction of q-BGap assumption from SGH, we can use the extended parameter-hiding property, as its challenge term do not contain any adversarial input. However for the reduction of q-SFrac assumption from SGH we can use extended adaptive parameter-hiding property, as the polynomial coefficients in the challenge terms are chosen by the adversary.*

Remark 3 *Having proved that SGH implies BTA assumption in the composite-order pairing setting, the natural question arises for the similar reduction in the prime-power setting. We don't have a positive answer, since emulating the parameter-hiding property for any polynomial and rational polynomial functions other than the affine function of the form* $f(x) = ax + b$ *[LM15] is still an interesting open problem.*

4.3 DéjàQ for BTA1 and BTA2

In this section we explain why the existing DéjàQ techniques are not applicable for q-type assumptions that belong to either BTA1 or BTA2 family. We explain the difficulty of proving the reduction with a concrete example. Here we could identify two issues for this difficulty.

To illustrate the first issue, we consider the q-SFP′ problem (which is equivalent to q-SFP problem [AFG+10] except with negligible probability, see §A) defined in Table 1 in which, given the instance $\left(g, h, g^z, g^f, g^r, g^u, g^a, g^b, h^c, h^d, \left\{g^{\frac{ac - zz_i - rr_i}{t_i}}, g^{\frac{bd - fz_i - uu_i}{w_i}}, h^{z_i}, h^{r_i}, h^{t_i}, h^{u_i}, h^{w_i}\right\}_{i=1}^{q}\right)$ compute

$\left(g^{\frac{ac-zz^*-rr^*}{t^*}}, g^{\frac{bd-fz^*-uu^*}{w^*}}, h^{z^*}, h^{r^*}, h^{t^*}, h^{u^*}, h^{w^*}\right)$ for $r^* \neq r_i$. The challenge terms of q-SFP$'$ problem also satisfy the two sets of PPEs. For simplicity we consider the first PPE, $e(g^{\frac{ac-zz^*-rr^*}{t^*}}, h^{t^*}) \stackrel{?}{=} e(g^a, h^c)e(g^z, h^{-z^*})\ e(g^r, h^{-r^*})$. One can use Chase et al's [CMM16] techniques to prove a result analogous to Theorem 2. However one can observe that from the above pairing equation, columns of the matrix V that correspond to the elements g^a, g^z, g^r and $g^{\frac{ac-zz^*-rr^*}{t^*}}$ will be linearly dependent. This makes the matrix V to be singular. However in order to complete the reduction from SGH assumption, Chase et al's technique requires non-singular matrix V. Hence we cannot use the existing DéjàQ techniques for q-SFP$'$ problem.

To illustrate the second issue, we consider the q-AGHO$'$ problem (which is equivalent to q-AGHO problem [AGHO11]) defined in Table 1. It is stated as, given the instance $\left(g, h, h^w, h^x, h^y, \left\{g^{x-a_i w-r_i y}, g^{a_i}, g^{r_i}, h^{a_i^{-1}}\right\}_{i=1}^{q}\right)$ compute $\left(g^{x-a^* w-r^* y}, g^{a^*}, g^{r^*}, h^{(a^*)^{-1}}\right)$. The challenge terms of q-AGHO$'$ problem should satisfy the following PPEs such as $e(g^{x-a^* w-r^* y}, h) \stackrel{?}{=} e(g, h^x)\ e(g^{a^*}, h^w)^{-1}\ e(g^{r^*}, h^y)^{-1}$ and $e(g^{a^*}, h^{(a^*)^{-1}}) \stackrel{?}{=} e(g, h)$. First we try to apply Chase-Meiklejohn's [CM14] techniques. During the reduction simulator chooses the secret w, x, y to generate an instance and sends to adversary who breaks the q-AGHO$'$ assumption. Then adversary outputs the challenge terms. Now in order to verify the well-formedness of the challenge terms, simulator must use the above PPEs as he does not know the exponents a^* and r^*. This is the main bottleneck, as the corresponding PPEs cannot be used to check the well-formedness of the challenge terms and hence it cannot distinguish the intermediate games. Thus we cannot apply Chase-Meiklejohn's technique. Now we apply the Chase et al's [CMM16] technique which uses two types of generators. The first type generator is analogous to the one used in Chase-Meiklejohn techniques, whereas the second type generator is defined in such a way that it is fixed throughout the reduction. One can use the second type generator for g^{a^*} and g^{r^*}, but we cannot use for $g^{x-a^* w-r^* y}$, as it leaks the information about w, x, y mod p_2. Hence one have to use the first type generator for $g^{x-a^* w-r^* y}$. This results to PPEs which are unsatisfiable and hence we cannot use these PPEs to check the well-formedness of the challenge terms. Thus one cannot distinguish between the intermediate games. Similar issue arises for all the q-type problems except the q-SFP$'$, that are listed in Table 1 and that belong to either BTA1 or BTA2 family.

5 Dual-Form Signature Variants

Here we consider two protocols whose security is proved under q-type assumptions that belong to either BTA1 or BTA2 family. The first one is Abe et al's [AGHO11] structure-preserving signature (SPS) scheme which is secure under q-AGHO assumption. The second one is Boyen-Waters [BW07] group signature (GS) scheme which is secure under q-HSDH assumption. We apply the dual-form signature techniques of Gerbush et al's [GLOW12] to construct a dual-form SPS scheme (and a dual-form GS) where security is based on some static assumption.

5.1 Dual-Form Abe et al's Structure-Preserving Signature Scheme

Structure-preserving signature (SPS) is used as a building block to construct several cryptographic primitives such as group signature, blind signature, anonymous credentials etc. SPS is a special type signature scheme where the message, public key and signature components belong to the underlying bilinear groups and the signature is verified using pairing product equations over the public key, the message and the signature.

Gerbush et al. introduced dual-form signature [GLOW12] which is defined using two signing algorithms, namely Sign_A and Sign_B that will respectively return two forms of signature and both will verify under the same public key. The security definition categorizes the forgeries into two types, Type I and Type II which typically correspond to the signatures returned by Sign_A and Sign_B respectively. See §D.1 for the definition of dual-form signature and its security and §E.1 for the definition of SPS scheme and its security.

Informally, we directly instantiate the original Abe et al's SPS scheme [AGHO11] in the asymmetric composite-order pairing and using dual-form signature techniques we prove its security under static assumption. Without loss of generality, in the following we assume that the signer chooses the message M from the group G. However the same techniques can be extended for the message vectors from either or both of the source groups G and H. Let $\Theta := (N = p_1p_2, G, H, G_T, e, \mu = \{g_1', g_2', h_1'\}) \xleftarrow{\$} \mathcal{G}(\lambda)$, where g_i' (resp. h_1') is a random element from the p_i-order subgroup G_i (resp. H_1) of G (resp. H) and pairing is defined as $e : G \times H \to G_T$. We instantiate the dual-form SPS scheme using the above mentioned bilinear group generator $\mathcal{G}$. In this construction, the public key and signatures returned by Sign_A algorithm resides in the subgroup of order p_1, whereas the signature returned by Sign_B algorithm resides in the group of order N. The dual-form SPS scheme consists of four PPT algorithms, which are defined as follows.

KeyGen(Θ). Choose g_i (resp. h_1) uniformly at random from G_i (resp. H_1). Choose w, x, y_1, y_2 uniformly at random from $\mathbb{Z}_N$ and compute $W = h_1^w, X = h_1^x$, $Y_1 = h_1^{y_1}$ and $Y_2 = h_1^{y_2}$. Return the secret key $SK = (w, x, y_1, y_2, g_2)$ and public key $PK = (g_1, h_1, W, X, Y_1, Y_2)$.

Sign$_A$(SK, M). Choose r (resp. a) uniformly at random from $\mathbb{Z}_N$ (resp. $\mathbb{Z}_N^*$). Compute $A = g_1^a$, $D = h_1^{1/a}$, $B = g_1^{x-aw-ry_1}M^{-y_2}$ and $R = g_1^r$. Return the signature $\sigma = (A, D, B, R)$ along with the message M.

Sign$_B$(SK, M). Choose $r, \gamma_1, \gamma_2, \gamma_3$ (resp. a) uniformly at random from $\mathbb{Z}_N$ (resp. $\mathbb{Z}_N^*$). Compute $A = g_1^a g_2^{\gamma_1}$, $D = h_1^{1/a}$, $B = g_1^{x-aw-ry_1}M^{-y_2}g_2^{\gamma_2}$ and $R = g_1^r g_2^{\gamma_3}$. Return the signature $\sigma = (A, D, B, R)$ along with the message M.

Verify(PK, M, σ). Parse the signature and check $A \overset{?}{\in} G, D \overset{?}{\in} H_1$[5] and $B, R \overset{?}{\in} G$. If any of the above checks fail to hold, then abort, else checks

$$e(R, h_1) \neq 1,\ e(A, D) \overset{?}{=} e(g_1, h_1) \text{ and } e(B, h_1)e(A, W)e(R, Y_1)e(M, Y_2) \overset{?}{=} e(g_1, X). \tag{4}$$

If all the above equations hold then return *accept*, otherwise return *reject*.

The signature returned by both Sign_A and Sign_B algorithms can be verified using Equation 4. It is easy to check the correctness of the scheme from Equation 4. As similar to Abe et al's [AGHO11] SPS scheme, we prove the above dual-form SPS scheme is secure in the sense of strongly unforgeable, i.e., adversary can forge on the queried message but the corresponding signature obtained from the signing oracle should differ from the forgery signature.

We define the variant of SGH assumptions such as Assumptions 5, 6 and 7 in Appendix E.2. Now we state the security of dual-form SPS scheme under these assumptions in the following theorem. The detailed proof can be found in Appendix E.2.

Theorem 4 *The dual-form of Abe et al's SPS scheme satisfies A-I matching, Dual-oracle invariance and B-II matching if $\mathcal{G}$ satisfies Assumption 5, 6 and 7 respectively.*

Proof sketch. The A-I matching is proved under Assumption 5. Given the instance, simulator $\mathcal{B}$ constructs the PK for dual-form SPS scheme by choosing the random exponents. Thus $\mathcal{B}$ knows all the components of SK except g_2. Hence $\mathcal{B}$ can answer for all the Sign_A queries. Once the adversary returns a valid forgery of Type-II, $\mathcal{B}$ uses it to solve Assumption 5. Similarly we can prove B-II matching under Assumption 7. Notice that the condition $e(R, h_1) \neq 1$ in the Verify algorithm ensures that the simulator in the B-II matching proof computes the non-trivial solution for Assumption 7. Whereas dual-oracle invariance is proved under Assumption 6. Here simulator $\mathcal{C}$ chooses all the random exponents to construct PK and the problem instance contains g_2. Thus $\mathcal{C}$ knows the entire SK components and hence he can answer for all the signing queries of both types. $\mathcal{C}$ embeds the challenge terms of the Assumption 6 while answering for a challenge query. Again based on the adversary's forgery types, $\mathcal{C}$ solves the underlying assumption. In all cases, simulator uses the suitable backdoor verification test (BVT) to check the forgery types returned by the adversary. □

[5] First we check A (resp. D) belongs to G (resp. H) by verifying $A^N = 1_G$ (resp. $D^N = 1_H$). Then the pairing equation $e(A, D) = e(g_1, h_1)$ ensures that D indeed belongs to subgroup H_1.

5.2 Dual-Form Boyen-Waters Group Signature Scheme

We discuss the case of Boyen-Waters [BW07] group signature which was originally proved secure under q-HSDH assumption. Since q-HSDH belongs to BTA1 family, we cannot apply the existing DéjàQ technique (see §4.3). As before, we apply Gerbush et al's [GLOW12] dual-form signature technique on this scheme and prove its security under static assumption.

Now we recall the Boyen-Waters [BW07] group signature scheme. The formal definition of group signature and its security properties such as full anonymity and full traceability is recalled in §F.1. The Boyen-Waters group signature was constructed using two-level hierarchical signature scheme (HSS) along with Groth-Sahai's [GS08] non-interactive witness indistinguishable (NIWI) proof. The security of NIWI proof system ensures the full anonymity and security of two-level HSS (it was proved under q-HSDH assumption) ensures the full traceability of group signature scheme. Note that in the Boyen-Waters [BW07] construction, signer constructs the NIWI proof ensuring that one of the user has signed the message. In particular, signer commits to the second level signatures returned by the appropriate types of signing algorithm. From the verification equations of second level signature, signer could construct the NIWI proof components that verify to the above commited values.

First we focus on constructing dual-form variant of two-level HSS [BW07] under subgroup hiding assumption. The two-level HSS consists of Boneh-Boyen signature [BB04] at first level and Waters signature [Wat05] at second level. In 2012, Yuen et al. [YCZY14] used dual-form signature techniques to obtain the dual-form of Boneh-Boyen signature scheme under SGH assumption instead of q-SDH assumption. In their construction, the original Boneh-Boyen signature component is modified to include some additional random elements to avoid the dependence on q-type assumption. We can directly use this dual-form Boneh-Boyen signature [YCZY14] along with the Waters signature [Wat05] to obtain the dual-form two-level HSS as in the original Boyen-Waters [BW07] construction and it is described in §F.2.

Now we focus on constructing dual-form variant of Boyen-Waters group signature scheme. As similar to the original construction, the dual-form two-level HSS scheme outputs a constant size signature and hence we could obtain the constant size dual-form variant of group signature. The scheme is defined over symmetric composite-order pairing with product of four primes. Let G_i be the subgroup of G, for any subset $i \subset [1,4]$. As similar to Yuen et al's construction [YCZY14], we use the subgroup G_3 to provide additional randomness and the subgroup G_2 to define the second type group signatures.

Now we present the dual-form group signature which consists of seven PPT algorithms which are defined as follows.

Setup(λ). Run the bilinear group generator $\mathcal{G}$ on λ which outputs (N, G, G_T, e, μ), where $N = p_1p_2p_3p_4$ with large primes $p_i > 2^\lambda$, for $i \in [1,4]$ and $\mu = \{g_{1,4}, g_{2,3}, g_3, g_4\}$. Choose g, h, u, w, v_0, $\{v_i\}_{i=1}^m$ (resp. α) uniformly at random from the subgroup[5] $G_{1,4}$ (resp. $\mathbb{Z}_N$), for $m = poly(\lambda)$. Now define the public parameter PP as $(N, G, G_T, e, g, u, A = e(g,h), g^\alpha, w, v_0, \{v_i\}_{i=1}^m, g_4)$ and master enrollment key MK as $(\alpha, h, g_3, g_{2,3}, \{s_i\}_{i=1}^{2^k})$ and tracing key TK as prime p_4, where $s_i \in \mathbb{Z}_N$ is a unique identifier of the user $ID \in \{0,1\}^k$ in the system, for $k = poly(\lambda)$. Then output PP, MK and TK.

Enroll$_A$(PP, MK, ID). Choose X_3, X_3' (resp. r) uniformly at random from G_3 (resp. $\mathbb{Z}_N$). Compute and output the private signing key $K_{ID} = (K_1, \ldots, K_4)$, where

$$K_1 = (hu^{-r})^{\frac{1}{\alpha + s_{ID}}} X_3, \quad K_2 = g^r X_3', \quad K_3 = g^{s_{ID}}, \quad K_4 = w^{s_{ID}}.$$

Enroll$_B$(PP, MK, ID). This algorithm is same as that of Enroll$_A$ except that K_1 and K_2 are defined as $K_1 = (hu^{-r})^{\frac{1}{\alpha + s_{ID}}} X_{2,3}$ and $K_2 = g^r X_{2,3}'$, for randomly chosen $X_{2,3}, X_{2,3}'$ from $G_{2,3}$.

Sign$_A$(PP, K_{ID}, M). Parse the message $M = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m$ and choose s uniformly at random from $\mathbb{Z}_N$. First compute the initial signature components $\theta = (\theta_1, \ldots, \theta_5)$, where $\theta_i = K_i$,

[5] Given a subgroup generator g_i as part of μ, the random element x of the subgroup G_i is generated by computing $x = g_i^r$, for $r \xleftarrow{\$} \mathbb{Z}_N$. This holds for any subset $i \subset [1,4]$.

for $i \in [1,3]$, $\theta_4 = K_4\big(v_0 \prod_{i=1}^{m} v_i^{\mu_i}\big)^s$ and $\theta_5 = g^{-s}$. Choose t_i uniformly at random from $\mathbb{Z}_N$ and compute

$$S_i = \theta_i g_4^{t_i}, \text{for } i \in [1,5], \quad \pi_1 = (\theta_1)^{t_3}(\theta_3 g^{\alpha})^{t_1} u^{t_2} g_4^{t_1 t_3} \quad \text{and} \quad \pi_2 = w^{t_3} g^{-t_4}\big(v_0 \prod_{j=1}^{m} v_j^{\mu_j}\big)^{-t_5}.$$

Then output the signature as $\sigma = (S_1, \ldots, S_5, \pi_1, \pi_2)$.

Sign$_B(PP, K_{ID}, M)$**.** This algorithm is same as that of Sign$_A$ except that it uses K_{ID} returned by the Enroll$_B$ algorithm.

Verify(PP, M, σ)**.** Parse the message M and signature σ and check that

$$e(S_1, S_3 g^{\alpha}) e(S_2, u) A^{-1} \stackrel{?}{=} e(\pi_1, g_4) \text{ and } e(S_3, w) e(S_4, g)^{-1} e\big(v_0 \prod_{j=1}^{m} v_j^{\mu_j}, S_5\big)^{-1} \stackrel{?}{=} e(\pi_2, g_4). \quad (5)$$

If any of the above checks fail to hold, then output reject, otherwise output accept.

Trace(PP, TK, σ)**.** Compute and check $(S_2)^{p_4} \stackrel{?}{=} (g^{s_{ID_i}})^{p_4}$ for every suspicious identity ID_i and return ID_i if the above check holds, else output $\perp$. This algorithm can be optimized further using lookup table with preprocessing the exponentiation of all the user identity.

The group signature consists of commitment components $(S_1, \ldots, S_5)$ under the signer's identifier s_{ID} (associated with user index) and some additional NIWI proof components, namely, π_1 and π_2. Without revealing s_{ID}, the proof components convinces the verifier that one of the signing key corresponding to some user s_{ID} in the system produced the signature. The scheme correctness can be verified using Equation 5. Let $\mathcal{V}$ be the collection of all message and signature pairs such that it satisfies Equation 5. The Type I forgery is defined as a collection of message and signature pairs in $\mathcal{V}$ such that $e(g_2, S_1^*) = 1$ and $e(g_2, S_2^*) = 1$ holds, otherwise it is said to be Type II forgery.

The full anonymity of the group signature scheme can be proved under subgroup hiding assumption. Given the SGH instance, simulator setup an experiment against full anonymity adversary. If the challenge term belongs to subgroup G_4 of $G_{1,4}$ then the experiment is identical to the original anonymity game, else we can prove that even an unbounded adversary cannot win the experiment as similar to [BW07].

Theorem 5 *Consider the bilinear group generator $\mathcal{G}(\lambda) \stackrel{\$}{\to} (N, G, G_T, e, \mu)$. If $\mathcal{G}$ satisfies subgroup hiding assumption, then the above described dual-form group signature is fully traceable.*

Proof sketch. As similar to [BW07], we can establish a reduction for each security property of dual-form group signature such as A-I matching, dual oracle invariance and B-II matching from that of dual-form two-level HSS. Notice that the dual-form two-level HSS is defined only in the subgroup of order $p_1 p_2 p_3$, but still it uses the bilinear pairing defined over N. Hence the reduction extensively uses the independent random exponents between the subgroups G_1 and G_4. This helps the reduction to translate PP, first and second level signatures from the dual-form two-level HSS into PP, private signing key and signatures of dual-form group signatures respectively. Also the reduction uses the exponentiation by p_4 as a function which always kill the component from the subgroup G_4. This helps the reduction to translate the forgery returned by the adversary from the group G to the subgroup $G_{1,2,3}$. □

References

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223, pages 209–236, Springer, 2010.

[AGHO11] Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841, pages 649–666, Springer, 2011.

[BB04] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In Cachin and Camenisch, editor, *EUROCRYPT*, volume 3027, pages 56–73, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494, pages 440–456, Springer, 2005.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677, pages 108–125, Springer, 2009.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and Noninteractive Anonymous Credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948, pages 356–374, Springer, 2008.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005* volume 3621, pages 258–275, Springer, 2005.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, EUROCRYPT, volume 2656, pages 614–629, Springer, 2003.

[Boy07] Xavier Boyen. Mesh Signatures. In Moni Naor, editor, *EUROCRYPT*, volume 4515, pages 210–227, Springer, 2007.

[Boy08] Xavier Boyen. The Uber-Assumption Family. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing 2008*, volume 5209, pages 39–56, Springer, 2008.

[BW07] Xavier Boyen and Brent Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450, pages 1–15, Springer, 2007.

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

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings - The role of Ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.

[CM14] Melissa Chase and Sarah Meiklejohn. Déjà Q: Using Dual Systems to Revisit q-Type Assumptions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441, pages 622–639, Springer, 2014.

[CMM16] Melissa Chase, Mary Maller, and Sarah Meiklejohn. Déjà Q All Over Again: Tighter and Broader Reductions of q-Type Assumptions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016*, volume 10032, pages 655–681, Springer, 2016.

[FHS14] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *IACR Cryptology ePrint Archive*, 2014:944, 2014.

[FPV09] Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Transferable Constant-Size Fair E-Cash. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 2009* volume 5888, pages 226–247, Springer, 2009.

[Fuc09] Georg Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. *IACR Cryptology ePrint Archive*, 2009:320, 2009.

[GG17] Essam Ghadafi and Jens Groth. Towards a classification of non-interactive computational assumptions in cyclic groups. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017*, volume 10625, pages 66–96, Springer, 2017.

[Gha14] Essam Ghadafi. Efficient distributed tag-based encryption and its application to group signatures with efficient distributed traceability. In Diego F. Aranha and Alfred Menezes, editors, LATINCRYPT, volume 8895, pages 327–347. Springer, 2014.

[Gha15] Essam Ghadafi. Stronger security notions for decentralized traceable attribute-based signatures and more efficient constructions. In Kaisa Nyberg, editor, *CT-RSA*, volume 9048, pages 391–409. Springer, 2015.

[GL07] Jens Groth and Steve Lu. A Non-interactive Shuffle with Pairing Based Verifiability. In Kurosawa, editor, *ASIACRYPT*, volume 4833, pages 51–67, 2007.

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

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Smart, editor, *EUROCRYPT 2008*, pages 415–432, Springer, 2008.

[JY09] David Jao and Kayo Yoshida. Boneh-boyen signatures and the strong diffie-hellman problem. In Hovav Shacham and Brent Waters, editors, *Pairing 2009*, volume 5671, pages 1–16, Springer, 2009.

[Lew12] Allison B. Lewko. Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237, pages 318–335, Springer, 2012.

[LM15] Allison B. Lewko and Sarah Meiklejohn. A Profitable Sub-prime Loan: Obtaining the Advantages of Composite Order in Prime-Order Bilinear Groups. In Jonathan Katz, editor, *PKC*, volume 9020, pages 377–398, Springer, 2015.

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 85(2):481–484, 2002.

[Oka06a] Tatsuaki Okamoto. Efficient Blind and Partially Blind Signatures Without Random Oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876, pages 80–99, Springer, 2006.

[Oka06b] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. *IACR Cryptology ePrint Archive*, 2006:102, 2006.

[Wat05] Brent Waters. Efficient Identity-Based Encryption Without Random Oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, pages 114–127, Springer, 2005.

[Wee15] Hoeteck Wee. Déjà Q: encore! un petit IBE. *IACR Cryptology ePrint Archive*, 2015:1064, 2015.

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

[ZSS04] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *PKC 2004*, volume 2947, pages 277–290, Springer, 2004.

A Complexity Assumptions

We recall some of the computational assumptions definition that are used in this paper. All these assumptions are defined in the asymmetric prime-order setting using bilinear group generator $\mathcal{G}$. Let $\Theta = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h) \xleftarrow{\$} \mathcal{G}(\lambda)$, where g (resp. h) is a generator of the group $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). Now we describe some of the computational q-type problems in Table 1, which includes Θ as part of the problem instance.

We say that the computational problem is said to have discrete logarithm representation, if each element in both instance and challenge terms are represented with an explicit discrete logarithm of that element specified in the exponent of some fixed generator. Now we explain that both the discrete logarithm representation and avoiding pairing product equations (PPEs) will help to classify the concrete q-type assumptions in to BTA1 or BTA2 families. For example, the variant of q-AGHO and q-SFP problems use the discrete logarithm representation and avoid using the corresponding PPEs, as opposed to its respective original counter-part from [AGHO11] and [AFG$^+$10]. These variants are denoted as q-AGHO$'$ and q-SFP$'$ respectively and both are classified into BTA1 family. Similarly the variant of q-SP problem avoids the PPEs as opposed to its original counter-part [GL07] and is denoted as q-SP$'$ problem. This variant is classified into BTA2 family. Similarly the variant of q-ADHSDH problem, denoted as q-ADHSDH$'$, uses the discrete logarithm representation for challenge terms as opposed to its original counter-part [Fuc09] and classified into BTA1 family. Now we prove the equivalence of these new variants with their respective original counter-parts.

Problem	Instance (Γ)	Challenge (Ω)
	BTA family	
q-DHI [MSK02]	$h^x, \{g^{x^i}\}_{i=1}^{q}$	$g^{\frac{1}{x}}$
q-GDHE [BGW05]	$h^x, \{g^{x^i}\}_{i=1, i\neq q+1}^{2q}$	$g^{x^{q+1}}$
q-SDH [BB04]	$h^x, \{g^{x^i}\}_{i=1}^{q}$	$\left(c, g^{\frac{1}{x+c}}\right)$
q-mSDH [BW07]	$g^x, h^x, \{g^{\frac{1}{x+a_i}}, a_i\}_{i=1}^{q}$	$\left(c, g^{\frac{1}{x+c}}\right)$ s.t $c \neq a_i$
q-2SDH[6] [Oka06a]	$h^y, \{g^{x^i}, h^{x^i}, g^{yx^i}\}_{i=1}^{q}, g^{\frac{y+b}{x+a}}, a, b$	$\left(c, g^{\frac{1}{x+c}}\right)$
q-mDSDH[7] [FPV09]	$g^x, h^x, g^y, \{g^{\frac{y+b_i}{x+a_i}}, a_i, b_i\}_{i=1}^{q}$	$\left(g^{\frac{y+d}{x+c}}, c, d\right)$ s.t $c \neq a_i$
BB-CDH [BCC+09]	$g^x, g^y, h^x, \{g^{\frac{1}{x+c_i}}, c_i\}_{i=1}^{q}$	g^{xy}
q-co-SDH [FHS14]	$\{g^{x^i}, h^{x^i}\}_{i=1}^{q}$	$\left(r(X), s(X), g^{\frac{r(x)}{s(x)}}\right)$ s.t $0 \leq \deg r(X) < \deg s(X) \leq q$
	BTA1 family	
q-HSDH [BW07]	$g^x, h^x, u, \{g^{\frac{1}{x+a_i}}, u^{a_i}, h^{a_i}\}_{i=1}^{q}$	$\left(g^{\frac{1}{x+c}}, u^c, h^c\right)$ s.t $c \neq a_i$
q-ADHSDH′[9] [Fuc09]	$g^x, h^x, g^y, u, \{g^{\frac{y+b_i}{x+a_i}}, u^{a_i}, h^{a_i}, g^{b_i}, h^{b_i}\}_{i=1}^{q}$	$g^{\frac{y+d}{x+c}}, u^c, h^c, g^d, h^d$ s.t $(c,d) \neq (a_i, b_i)$
q-AGHO′[9] [AGHO11]	$h^w, h^x, h^y, \{g^{x-a_i w - r_i y}, g^{a_i}, g^{r_i}, h^{a_i^{-1}}\}_{i=1}^{q}$	$g^{x-a^* w - r^* y}, g^{a^*}, g^{r^*}, h^{(a^*)^{-1}}$
q-SFP′[8] [AFG+10]	$\left(g^z, g^f, g^r, g^u, g^a, g^b, h^c, h^d, \{g^{\frac{ac - zz_i - rr_i}{t_i}}, g^{\frac{bd - fz_i - uu_i}{w_i}}, h^{z_i}, h^{r_i}, h^{t_i}, h^{u_i}, h^{w_i}\}_{i=1}^{q}\right)$	$\left(g^{\frac{ac - zz^* - rr^*}{t^*}}, g^{\frac{bd - fz^* - uu^*}{w^*}}, h^{z^*}, h^{r^*}, h^{t^*}, h^{u^*}, h^{w^*}\right)$, $r^* \neq r_i$
	BTA2 family	
BB-HSDH [BCC+09]	$g^x, h^x, u, \{g^{\frac{1}{x+a_i}}, a_i\}_{i=1}^{q}\big)$	$g^{\frac{1}{x+c}}, u^c, h^c$ s.t $c \neq a_i$
q-TDH [BCKL08]	$g^x, g^y, h^x, \{c_i, g^{1/(x+c_i)}\}_{i=1}^{q}$	$h^{\mu x}, g^{\mu y}, g^{\mu xy}$
q-SP′[8] [GL07]	$\{g^{x_i}, g^{x_i^2}\}_{i=1}^{q}$	$\left(\{h^{y_i}\}_{i=3}^{q}, h^{\frac{\sum_{i=3}^{q} x_i(x_i - x_2)y_i}{x_1(x_2 - x_1)}}, h^{\frac{\sum_{i=3}^{q} x_i(x_i - x_1)y_i}{x_2(x_1 - x_2)}}\right)$
q-2SDH[6] [Oka06b]	$g^x, h^x, g^y, h^y, \{g^{\frac{y+b_i}{x+a_i}}, h^{\frac{y+bi}{x+a_i}}, a_i, b_i\}_{i=1}^{q}$	$\left(g^{\frac{y+d}{\theta x+\rho}}, g^{x/\theta+\tau}, h^{(\theta\tau+\rho/\theta)x+\rho\tau}, h^{\theta x+\rho}, d\right)$ s.t $d \neq b_i$
(q,ℓ)-Poly-SDH [Boy07]	$\{g^{x_i}, h^{x_i}\}_{i=1}^{\ell}, \{g^{\frac{1}{x_i + c_{ij}}}, c_{ij}\}_{i,j=1}^{\ell,q}$	$\{g^{\frac{\gamma_i}{x_i+c_i}}, c_i\}_{i=1}^{\ell}, \sum_{i=1}^{\ell} \gamma_i = 1$
(q,ℓ,ℓ')-Pluri-SDH [Boy07]	$\{g^{x_i}, h^{x_i}\}_i, \{g^{\frac{1}{x_{i'} + c_{i'j}}}, c_{i'j}\}_{i',j=1}^{q}$ $i \in [-\ell', \ell] - \{0\}$, $i' \in [-\ell', -1]$	$\{g^{\frac{\gamma_i}{x_i+c_i}}, c_i\}_i$ s.t $\sum_i \gamma_i = 1$ $i \in [-\ell', \ell] - \{0\}$

Table 1: Computational problems classification in BTA and its variants.

First we prove that q-SFP problem [AFG+10] is equivalent to q-SFP′ problem described in Table 1. Informally q-SFP problem says that, given $G_Z, F_Z, G_R, F_U, A, \tilde{A}, B, \tilde{B}$ and $\{Z_i, R_i, S_i, T_i, U_i, V_i, W_i\}_{i=1}^{q}$ which satisfies Equation 6, compute (Z, R, S, T, U, V, W) satisfying Equation 6. Here both instance and challenge terms should satisfy the following PPEs and hence we use the common variables $\left(Z^*, R^*, S^*, T^*, U^*, V^*, W^*\right)$.

$$e(A, \tilde{A}) \stackrel{?}{=} e(G_Z, Z^*)e(G_R, R^*)e(S^*, T^*) \text{ and } e(B, \tilde{B}) \stackrel{?}{=} e(F_Z, Z^*)e(F_U, U^*)e(V^*, W^*). \tag{6}$$

[6] Chatterjee-Menezes [CM11] technique can be used to convert these problems from Type-II pairing to Type-III pairing.

[7] We rename the problem in Definition 15 of [FPV09] as q-mDSDH problem.

[8] We argue the equivalence of these assumptions with its original counter-part except negligible probability.

[9] We argue that these assumptions are equivalent to its original counter-part.

As we are in the cyclic group of prime-order pairing setting, each element in the source groups can be represented using some fixed generators (say) g and h. For simplicity, we consider the elements of problem instance corresponding to the first PPE from Equation 6. Thus elements $A, \tilde{A}, G_Z, G_R, Z_i, R_i, S_i$ and T_i from the instance are represented as $A = g^a, \tilde{A} = h^c, G_Z = g^z, G_R = g^r, Z_i = h^{z_i}, R_i = h^{r_i}, S_i = g^{s_i}$ and $T_i = h^{t_i}$ respectively, for the appropriate randomness from $\mathbb{Z}_p$. Now applying this representation on first PPE, we get

$$\begin{aligned} e(g^a, h^c) \stackrel{?}{=} e(g^z, h^{z_i}) e(g^r, h^{r_i}) e(g^{s_i}, h^{t_i}) &\Leftrightarrow e(g,h)^{ac} \stackrel{?}{=} e(g,h)^{zz_i + rr_i + s_i t_i} \\ &\Leftrightarrow ac = zz_i + rr_i + s_i t_i \bmod p, \end{aligned} \tag{7}$$

where the first equivalent condition holds due to bilinearity of the map e and the second equivalent condition holds from the fact that in a cyclic group of order p, $g^x = g^y$ if and only if p divides $(x - y)$.

Now we can see that the following events are equal,

$$\begin{aligned} \{ac = zz_i + rr_i + s_i t_i \bmod p\} = &\{t_i = 0\} \wedge \{ac = zz_i + rr_i \bmod p\} \\ &+ \{t_i \neq 0\} \wedge \{s_i = (ac - zz_i - rr_i)/t_i \bmod p\} \end{aligned} \tag{8}$$

The above equality of the events corresponds to the first PPE described in Equation 6. Similarly we can obtain the equivalent conditions and events analogous to Equation 7, for the other set of instance elements that corresponds to second PPE and for the challenge terms of q-SFP problem.

With this background, now we prove the following.

Lemma 6 *The q-SFP problem [AFG+10] is equivalent to q-SFP′ problem defined in Table 1 except with some negligible probability.*

Proof. Assume that there exists a polynomial time adversary $\mathcal{A}_{SFP'}$ for q-SFP′ problem. Now we construct a simulator $\mathcal{B}_{SFP}$ for q-SFP problem. Given the instance of q-SFP problem, simulator $\mathcal{B}_{SFP}$ relays the same as an instance of q-SFP′ problem to $\mathcal{A}_{SFP'}$, except when atleast one t_i or w_i is zero. From the contradiction assumption, $\mathcal{A}_{SFP'}$ returns a valid challenge terms of q-SFP′ problem which are used as a challenge terms of q-SFP problem by $\mathcal{B}_{SFP}$. Thus from the probability calculation, we have

$$\begin{aligned} Adv^{SFP'} &= \Pr[\mathcal{A}_{SFP'}(\Gamma) = \Omega] \\ &= \Pr\Big[\bigwedge_{i=1}^{q} (t_i \neq 0 \wedge w_i \neq 0) \wedge \mathcal{A}_{SFP'}(\Gamma) = \Omega\Big] + \Pr\Big[\bigvee_{i=1}^{q} (t_i = 0 \vee w_i = 0) \wedge \mathcal{A}_{SFP'}(\Gamma) = \Omega\Big] \\ &= \Pr\big[\mathcal{B}_{SFP}(\Gamma) = \Omega\big] + \Pr\Big[\bigvee_{i=1}^{q} (t_i = 0 \vee w_i = 0) \wedge \mathcal{A}_{SFP'}(\Gamma) = \Omega\Big] \\ &\leq Adv^{SFP} + \sum_{i=1}^{q} (\Pr[t_i = 0] + \Pr[w_i = 0]) = Adv^{SFP} + 2q/p. \end{aligned}$$

In the above computation, the third equality comes from the above simulation for q-SFP assumption and the fourth inequality comes from the basic probability results. We note that the restriction for the above simulation is due to the event that atleast one of t_i or w_i is zero, for $i \in [1, q]$. This restriction contribute to the additional term $2q/p$ in the above relation.

Now for the other direction, there exists a polynomial time adversary $\mathcal{A}_{SFP}$ for q-SFP problem. Now we construct a simulator $\mathcal{B}_{SFP'}$ for q-SFP′ problem. Given the instance of q-SFP′ problem, $\mathcal{B}_{SFP'}$ relays the same as an instance of q-SFP problem to $\mathcal{A}_{SFP}$. From the contradiction assumption, $\mathcal{A}_{SFP}$ returns a valid challenge terms of q-SFP problem with some non-negligible advantage. From the definition of q-SFP′ it is sufficient to look for those exponents t and w such that both are non-zero in $\mathbb{Z}_p$. The probability calculation is similar to the previous direction but for a single pairs t_i, w_i instead of q pairs. This directly imply that

$$Adv^{SFP} \leq Adv^{SFP'} + 2/p.$$

Since q is polynomial in the security parameter λ and for large prime $1/p$ is negligible, hence $2/p$ and $2q/p$ are negligible. Thus q-SFP problem is equivalent to q-SFP$'$ problem except with negligible probability. □

Similarly one can prove that q-AGHO and q-AGHO$'$ problems are equivalent. This is mainly because the PPE for q-AGHO problem do not have a rational expression and hence there is no restriction while simulating the reduction.

Now we consider the q-SP problem [GL07], in which both instance and challenge terms use the discrete logarithm representation. However from the corresponding PPEs we can simplify and obtain the explicit form of the challenge exponents, so that we can avoid using PPEs. This result to q-SP$'$ problem which can be classified into BTA2 family. Recall that q-SP problem says that given $g, h, \{g^{x_i}, g^{x_i^2}\}_{i=1}^{q}$, compute $\{h^{y_i}\}_{i=1}^{q}$ such that $\prod_{i=1}^{q} e(g^{x_i}, h^{y_i}) \stackrel{?}{=} 1$ and $\prod_{i=1}^{q} e(g^{x_i^2}, h^{y_i}) \stackrel{?}{=} 1$. Now we interpret these PPEs as a system of equations having two equations with q many variables. We solve this system of equations and obtain

$$y_1 = \frac{\sum_{i=3}^{q} x_i(x_i - x_2)s_i}{x_1(x_2 - x_1)}, y_2 = \frac{\sum_{i=3}^{q} x_i(x_i - x_1)s_i}{x_2(x_1 - x_2)} \text{ and } y_i = s_i \xleftarrow{\$} \mathbb{Z}_p, \text{for } i \in [3, q].$$

From this representation, we can construct q-SP$'$ problem as described in Table 1. Here the additional restriction is $x_1 \neq x_2$ and this happens with probability $1/p$ which is negligible. Thus we can justify that q-SP and q-SP$'$ problems are equivalent except with negligible probability.

Similarly we can argue that q-ADHSDH$'$ problem is obtained from q-ADHSDH problem by expressing its challenge terms using discrete logarithm representation. The q-ADHSDH$'$ problem does not contain any additional restriction and hence we can prove that it is equivalent to q-ADHSDH problem.

B Proof of Lemma 1

To prove the Lemma 1 we follow the reduction techniques from Ghadafi and Groth [GG17]. Hence we follow their computational assumption definition which emphasize the private information in `priv`. The non-interactive computational assumption consists of $(\mathcal{I}, \mathcal{V})$, in which the instance generator $\mathcal{I}$ takes λ as input and outputs an instance Γ along with `pub` and `priv`, where `priv` contains some secret information such as the secret vectors $\boldsymbol{x}$ that was used to evaluate the rational polynomials, `pub` contains information about the underlying bilinear groups and its order. The instance verifier $\mathcal{V}$ takes the challenge terms Ω, `pub`, `priv` as input and outputs 1 or 0. Adversary takes the instance Γ and `pub` as input and outputs the challenge terms Ω. From the definition of BTA and its variants BTA1 and BTA2, we include the secret vector $\boldsymbol{x}$ and all the polynomial coefficients of the instance are included as part of `priv`.

Proof. (i.) **BTA2 implies BTA**. First we prove the existence result. From the definition of BTA and BTA2, instance of both BTA and BTA2 problems contain all the rational polynomials whose coefficients are given explicitly. Hence any one can construct an instance of BTA2 from BTA. However all the polynomial coefficients of the challenge term of BTA are given explicitly. Hence any one can compute the challenge terms for BTA2 by exponentiating each of those polynomial coefficients in both source groups. Thus constructing a BTA2 problem from that of BTA directly ensures the existence result.

Now we prove the reduction using contradiction method. Assume that there exists a polynomial time adversary $\mathcal{A}$ that breaks a BTA problem with some non-negligible advantage. Now we construct a BTA2 solver, say $\mathcal{B}$ as follows. Given the instance of BTA2, $\mathcal{B}$ randomizes this instance and passes to $\mathcal{A}$. Observe that since $\mathcal{B}$ has to construct the challenge terms of BTA2 from that of BTA, hence the appropriate parameters (which includes n_{c_1}, n_{c_2} and n_{c_T}) are given as part of `pub`. Now $\mathcal{B}$ chooses n_m such that $n_m = \max\{n_{c_1}, n_{c_2}, n_{c_T}\}$. Thus $\mathcal{B}$ uses $\mathcal{A}$ as a subroutine in exactly n_m many times in order to construct the challenge terms for BTA2 problem. Let us denote the BTA2 instance as

$$\Gamma = \left(\left\{ [1]_j, \left\{ \left[\frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})} \right]_j, a_i^{(j)}(\boldsymbol{X}), b_i^{(j)}(\boldsymbol{X}) \right\}_{i=1}^{n_j} \right\}_{j \in \{1,2,T\}}, \texttt{pub} \right),$$

where $\mathtt{pub}$ contains the additional information such as description of bilinear groups Θ and $\mathtt{priv}$ contains the secret vector $\boldsymbol{x}$ from $\mathbb{Z}_p^m$. For each $t \in [1, n_m]$, $\mathcal{B}$ chooses θ_{tj} uniformly at random from $\mathbb{Z}_p^*$ and constructs a random BTA instance as

$$\Gamma_t = \left\{ \theta_{tj}[1]_j, \left\{ \theta_{tj} \left[\frac{a_i^{(j)}(\boldsymbol{x})}{b_i^{(j)}(\boldsymbol{x})} \right]_j, \theta_{tj} a_i^{(j)}(\boldsymbol{X}), \theta_{tj} b_i^{(j)}(\boldsymbol{X}) \right\}_{i=1}^{n_j} \right\}_{j \in \{1,2,T\}}$$

along with the public (resp. private) information is denoted as $\mathtt{pub}_t$ (resp. $\mathtt{priv}_t$). Given the instance Γ_t, $\mathtt{pub}_t$, $\mathtt{priv}_t$, $\mathcal{A}$ outputs $\Omega_t = \left(\left\{ \left[\frac{r^{(j)}(\boldsymbol{x})}{s^{(j)}(\boldsymbol{x})} \right]_j, r^{(j)}(\boldsymbol{X}), s^{(j)}(\boldsymbol{X}) \right\}_{j \in \{1,2,T\}}, \mathtt{sol}_t' \right)$ with some non-negligible probability. Since n_m is chosen in such a way that $n_m \geq n_{c_j}$, for $j \in \{1, 2, T\}$. Hence without loss of generality, $\mathcal{B}$ runs $\mathcal{A}$ as subroutine n_m many times and collects the first n_{c_j} number of polynomials from $\mathcal{A}$'s output collections. From this collection $\mathcal{B}$ constructs the challenge terms for BTA2 problem as follows. Now $\mathcal{B}$ exponentiates the coefficients of those selected polynomials $r^{(j)}(\boldsymbol{X})$ and $s^{(j)}(\boldsymbol{X})$ using the generators $[1]_1$ and $[1]_2$, as these coefficients are given explicitly. Thus $\mathcal{B}$ computes the challenge terms Ω for BTA2 problem as,

$$\left(\left\{ \left\{ \theta_{tj}^{-1} \left[\frac{r_t^{(j)}(\boldsymbol{x})}{s_t^{(j)}(\boldsymbol{x})} \right]_j, \left(\left\{ \theta_{tj'}^{-1} \left[r_t^{(j)}(\boldsymbol{X}) \right]_{j'} \right\}_{j'=1}^{2} \text{ or } r_t^{(j)}(\boldsymbol{X}) \right), \right. \right. \\ \left. \left. \left(\left\{ \theta_{tj'}^{-1} \left[s_t^{(j)}(\boldsymbol{X}) \right]_{j'} \right\}_{j'=1}^{2} \text{ or } s_t^{(j)}(\boldsymbol{X}) \right) \right\}_{t=1}^{n_{c_j}} \right\}_{j \in \{1,2,T\}}, \mathtt{sol} \right) \tag{9}$$

In the above expression, $\mathcal{B}$ removes the randomness θ_{tj} that was used to generate the BTA instance earlier. Also we notice that $\mathtt{sol}_t'$ contains additional information such as how the polynomials $r_t^{(j)}(\boldsymbol{X})$ and $s_t^{(j)}(\boldsymbol{X})$ were constructed. However it is sufficient to include all the polynomial coefficients that are not given explicitly and hence we have $\mathtt{sol} \subseteq \{\mathtt{sol}_t'\}_{t=1}^{n_m}$.

From the definition of BTA, output of $\mathcal{A}$ satisfies Equations 2 and 3. Now we prove that the challenge terms constructed in Equation 9 should satisfy Equations 2 and 3. Suppose that the challenge terms constructed above do not satisfy Equations 2 and 3. In other words, for all $t \in [1, n_{c_j}]$ and $j_1, j_2 \in \{1, 2, T\}$, each element $\left[\frac{r_t^{(j)}(\boldsymbol{X})}{s_t^{(j)}(\boldsymbol{X})} \right]_j$, $\left[r_t^{(j_1)}(\boldsymbol{X}) \right]_j$ and $\left[s_t^{(j_2)}(\boldsymbol{X}) \right]_j$ can be written as a linear combination of $\left[\frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})} \right]_j$, $\left[a_{i_1}^{(\iota_1)}(\boldsymbol{X}) \right]_j$, $\left[b_{i_2}^{(\iota_2)}(\boldsymbol{X}) \right]_j$. In particular, the above linear relation for the element $\left[\frac{r_t^{(j)}(\boldsymbol{X})}{s_t^{(j)}(\boldsymbol{X})} \right]_j$ violates Equations 2 and 3. This contradicts the assumption that adversary $\mathcal{A}$ outputs a valid challenge term for BTA problem with non-negligible probability.

(ii.) **BTA2 implies BTA1**. Proof is similar to the previous case, hence we give the high level idea. Given the BTA2 instance, all the polynomial coefficients are exponentiated in both source groups to construct the BTA1 instance. Once the adversary returns a valid challenge term, the same can be forwarded as BTA2's challenge term.

C Proof of Theorem 2

We prove this theorem using hybrid techniques using sequence of games. All the intermediate games are described in Figure 1.1 and 1.2. Let $\mathcal{A}$ be a polynomial time adversary playing the game BTA (resp. vBTA) that corresponds to the bilinear target assumption (resp. Assumption 4) defined in G and its advantage is denoted as $Adv_{\mathcal{A}}^{\mathsf{BTA}}$ (resp. $Adv_{\mathcal{A}}^{\mathsf{vBTA}}$).

We provide polynomial time adversaries $\mathcal{B}_0, \mathcal{C}_0$ and $\{\mathcal{B}_k\}_{k=1}^{\ell}$ such that

$$Adv_{\mathcal{A}}^{\mathsf{BTA}}(\lambda) \leq Adv_{\mathcal{B}_0}^{SGH}(\lambda) + Adv_{\mathcal{C}_0}^{SGH}(\lambda) + (\ell - 1) Adv_{\mathcal{B}_k}^{SGH}(\lambda) + Adv_{\mathcal{A}}^{\mathsf{vBTA}}(\lambda)$$

for all $\lambda \in \mathbb{N}$ and which is enough to prove the Theorem2. In particular, we construct the simulators

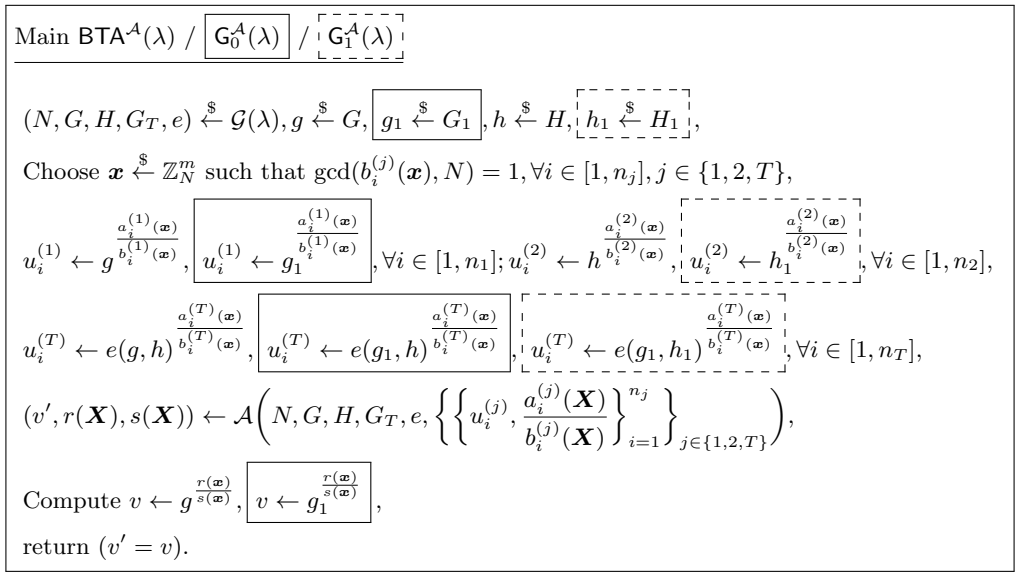

Figure 1.1: Games for the proof of Theorem 2 (for both Equations 10 and 11). The boxed (resp. dotted boxed) game uses the boxed (resp. dotted boxed) code and other game do not.

$\mathcal{B}_0, \mathcal{C}_0$ and $\mathcal{B}_k$ such that

$$\Pr[\mathsf{BTA}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G_0}^{\mathcal{A}}(\lambda)] \leq Adv_{\mathcal{B}_0}^{SGH}(\lambda), \quad (\text{in } G_1, \mu = \{\,\}) \tag{10}$$

$$\Pr[\mathsf{G}_0^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_1^{\mathcal{A}}(\lambda)] \leq Adv_{\mathcal{C}_0}^{SGH}(\lambda), \quad (\text{in } H_1, \mu = \{g_1\}) \tag{11}$$

$$\Pr[\mathsf{G}_k^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{k,1}^{\mathcal{A}}(\lambda)] \leq Adv_{\mathcal{B}_k}^{SGH}(\lambda), \quad (\text{in } G_1, \mu = \{g_2, h_1\}) \tag{12}$$

$$\Pr[\mathsf{G}_{k,1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{k+1}^{\mathcal{A}}(\lambda)] = 0, \quad (\text{extended adaptive parameter-hiding}) \tag{13}$$

$$\Pr[\mathsf{G}_\ell^{\mathcal{A}}(\lambda)] = Adv_{\mathcal{A}}^{\mathsf{vBTA}}(\lambda). \tag{14}$$

From the above inequalities we have,

$$\begin{aligned} Adv_{\mathcal{A}}^{\mathsf{BTA}}(\lambda) =& \Pr[\mathsf{BTA}^{\mathcal{A}}(\lambda)] \\ =& \big(\Pr[\mathsf{BTA}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_0^{\mathcal{A}}(\lambda)]\big) + \big(\Pr[\mathsf{G}_0^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_1^{\mathcal{A}}(\lambda)]\big) \\ &+ \Big(\sum_{k=1}^{\ell-1} \Pr[\mathsf{G}_k^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{k,1}^{\mathcal{A}}(\lambda)] + \Pr[\mathsf{G}_{k,1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{k+1}^{\mathcal{A}}(\lambda)]\Big) + \Pr[\mathsf{G}_\ell^{\mathcal{A}}(\lambda)] \\ \leq& Adv_{\mathcal{B}_0}^{SGH}(\lambda) + Adv_{\mathcal{C}_0}^{SGH}(\lambda) + (\ell-1)Adv_{\mathcal{B}_k}^{SGH}(\lambda) + Adv_{\mathcal{A}}^{\mathsf{vBTA}}(\lambda). \end{aligned}$$

Equation 10: $\mathsf{BTA}^{\mathcal{A}}$ vs $\mathsf{G}_0^{\mathcal{A}}$

Now we construct the simulator $\mathcal{B}_0$ which in-distinguish between the games BTA and $\mathsf{G_0}$ under SGH

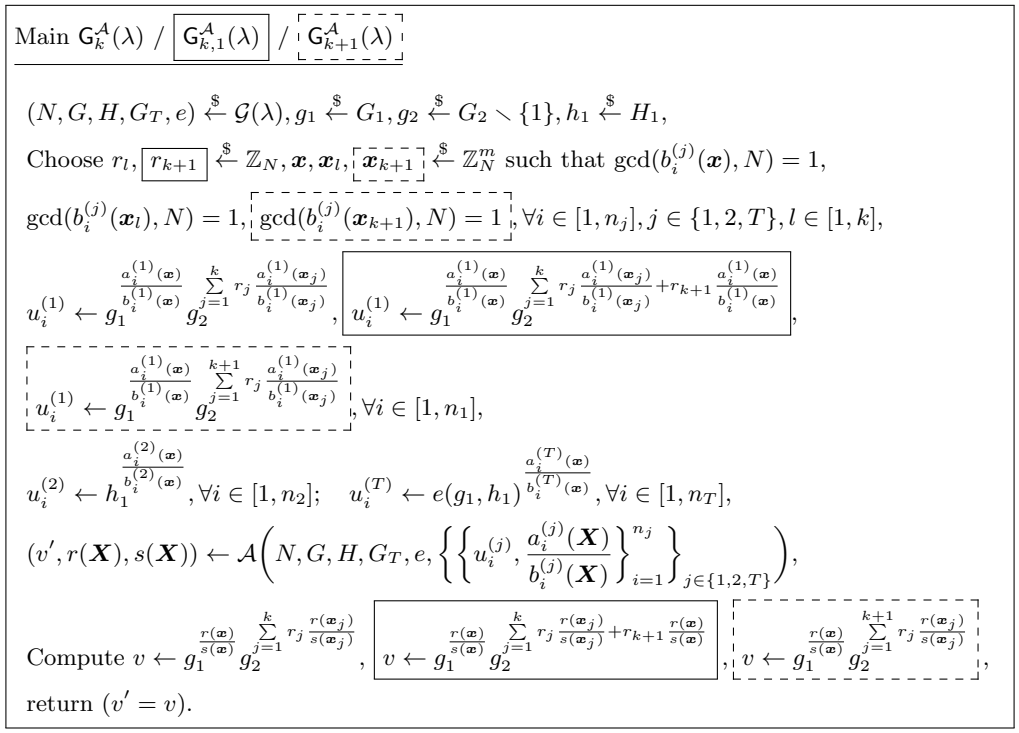

Figure 1.2: Games for the proof of Theorem 2 (for both Equations 12 and 13). The boxed (resp. dotted boxed) game uses the boxed (resp. dotted boxed) code and other game do not.

assumption for the subgroup G_1 with $\mu = \{\ \}$ as follows,

$\underline{\mathcal{B}_0(\lambda, N, G, H, G_T, e, w)}$

Choose $\boldsymbol{x} \xleftarrow{\$} \mathbb{Z}_N^m$ such that $\gcd(b_i^{(j)}(\boldsymbol{x}), N) = 1, \forall i \in [1, n_j], j \in \{1, 2, T\},$

Choose $h \xleftarrow{\$} H$ compute $u_i^{(1)} \leftarrow w^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}}, u_i^{(2)} \leftarrow h^{\frac{a_i^{(2)}(\boldsymbol{x})}{b_i^{(2)}(\boldsymbol{x})}}, u_i^{(T)} \leftarrow e(w, h)^{\frac{a_i^{(T)}(\boldsymbol{x})}{b_i^{(T)}(\boldsymbol{x})}},$

$(v', r(\boldsymbol{X}), s(\boldsymbol{X})) \leftarrow \mathcal{A}\Big(N, G, H, G_T, e, \Big\{\Big\{u_i^{(j)}, \frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})}\Big\}_{i=1}^{n_j}\Big\}_{j \in \{1,2,T\}}\Big),$

Return $(v' = w^{\frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}})$.

If $w \xleftarrow{\$} G$, then this is identical to the BTA game and if $w \xleftarrow{\$} G_1$, then this is identical to G_0.

Equation 11: G_0 vs G_1

Now we construct the simulator $\mathcal{C}_0$ which in-distinguish between the games G_0 and G_1 under SGH

assumption for the subgroup H_1 with $\mu = \{g_1\}$ as follows,

$\underline{\mathcal{C}_0(\lambda, N, G, H, G_T, e, g_1, z)}$

Choose $\boldsymbol{x} \xleftarrow{\$} \mathbb{Z}_N^m$ such that $\gcd(b_i^{(j)}(\boldsymbol{x}), N) = 1, \forall i \in [1, n_j], j \in \{1, 2, T\}$,

Compute $u_i^{(1)} \leftarrow g_1^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}}, u_i^{(2)} \leftarrow z^{\frac{a_i^{(2)}(\boldsymbol{x})}{b_i^{(2)}(\boldsymbol{x})}}, u_i^{(T)} \leftarrow e(g_1, z)^{\frac{a_i^{(T)}(\boldsymbol{x})}{b_i^{(T)}(\boldsymbol{x})}}$,

$(v', r(\boldsymbol{X}), s(\boldsymbol{X})) \leftarrow \mathcal{A}\Big(N, G, H, G_T, e, \Big\{\Big\{u_i^{(j)}, \frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})}\Big\}_{i=1}^{n_j}\Big\}_{j\in\{1,2,T\}}\Big)$,

Return $(v' = g_1^{\frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}})$.

If $z \xleftarrow{\$} H$, then this is identical to G_0 and if $z \xleftarrow{\$} H_1$, then this is identical to G_1. Here $\mathcal{A}$ distinguish the game G_0 with game G_1 based on the difference in the problem instance.

Equation 12: G_k vs $\mathsf{G}_{\mathsf{k},1}$

Now we construct a simulator $\mathcal{B}_k$ that in-distinguish between the games G_k and $\mathsf{G}_{\mathsf{k},1}$ under the SGH assumption for the subgroup G_1 with $\mu = \{g_2, h_1\}$ as follows,

$\underline{\mathcal{B}_k(\lambda, N, G, H, G_T, e, g_2, h_1, w)}$,

Choose $\boldsymbol{x}, \boldsymbol{x}_l \xleftarrow{\$} \mathbb{Z}_N^m$ such that $\gcd(b_i^{(j)}(\boldsymbol{x}), N) = 1, \gcd(b_i^{(j)}(\boldsymbol{x}_l), N) = 1$,

$\forall i \in [1, n_j], j \in \{1, 2, T\}, l \in [1, k]$,

Compute $u_i^{(1)} \leftarrow w^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}} g_2^{\sum_{j=1}^{k} r_j \frac{a_i^{(1)}(\boldsymbol{x}_j)}{b_i^{(1)}(\boldsymbol{x}_j)}}, u_i^{(2)} \leftarrow h_1^{\frac{a_i^{(2)}(\boldsymbol{x})}{b_i^{(2)}(\boldsymbol{x})}}, u_i^{(T)} \leftarrow e(w, h_1)^{\frac{a_i^{(T)}(\boldsymbol{x})}{b_i^{(T)}(\boldsymbol{x})}}$,

$(v', r(\boldsymbol{X}), s(\boldsymbol{X})) \leftarrow \mathcal{A}\Big(N, G, H, G_T, e, \Big\{\Big\{u_i^{(j)}, \frac{a_i^{(j)}(\boldsymbol{X})}{b_i^{(j)}(\boldsymbol{X})}\Big\}_{i=1}^{n_j}\Big\}_{j\in\{1,2,T\}}\Big)$,

return $(v' = w^{\frac{r(\boldsymbol{x})}{s(\boldsymbol{x})}} g_2^{\sum_{j=1}^{k} r_j \frac{r(\boldsymbol{x}_j)}{s(\boldsymbol{x}_j)}})$.

If $w = g_1 \xleftarrow{\$} G_1$, then this is identical to G_k game and if $w \xleftarrow{\$} G$, then write $w = g_1 g_2^{r_{k+1}}$, for some $g_1 \xleftarrow{\$} G_1$ and $r_{k+1} \xleftarrow{\$} \mathbb{Z}_N$. Then it is clear that the above game is identical to $\mathsf{G}_{k,1}$.

Equation 13: $\mathsf{G}_{k,1}$ vs G_{k+1}

Define $A := g_2^{\sum_{j=1}^{k} r_j \frac{a_i^{(1)}(\boldsymbol{x}_j)}{b_i^{(1)}(\boldsymbol{x}_j)}}$. It is clear that A is independent of $\boldsymbol{x}$ and $\boldsymbol{x}_{k+1}$. Then we use the extended adaptive parameter-hiding property with respect to the functions f and f' which corresponds to the evaluation of rational polynomials $\Big\{\Big\{\frac{a_i^{(1)}(\boldsymbol{X})}{b_i^{(1)}(\boldsymbol{X})}\Big\}_{i=1}^{n_1}, \frac{r(\boldsymbol{X})}{s(\boldsymbol{X})}\Big\}$ at some vectors $\boldsymbol{x}$ and $\boldsymbol{x}_{k+1}$ respectively (modulo p_n). Thus we obtain that the distributions $g_1^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}} g_2^{r_{k+1}\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}}$ and $g_1^{\frac{a_i^{(1)}(\boldsymbol{x})}{b_i^{(1)}(\boldsymbol{x})}} g_2^{r_{k+1}\frac{a_i^{(1)}(\boldsymbol{x}_{k+1})}{b_i^{(1)}(\boldsymbol{x}_{k+1})}}$ are identical. This ensure that the distribution of instance in the games $\mathsf{G}_{\mathsf{k},1}$ and $\mathsf{G}_{\mathsf{k}+1}$ are identical. Similarly we can prove that for the challenge term. Note that the auxiliary information Aux contains the elements $\Big\{\Big\{\frac{a_i^{(2)}(\boldsymbol{X})}{b_i^{(2)}(\boldsymbol{X})}\Big\}_{i=1}^{n_2}, \Big\{\frac{a_i^{(T)}(\boldsymbol{X})}{b_i^{(T)}(\boldsymbol{X})}\Big\}_{i=1}^{n_T}\Big\}$ and these elements do not affect the above distributions as they defined in different subgroup of H and G_T.

Equation 14: G_ℓ

From the definition of Assumption 4 it is clear that this Equation 14 holds.

D Dual-Form Signature

D.1 Definition of Dual-Form Signature

We recall the definition of dual-form signature from [GLOW12] which consists of four algorithms.
Setup: Given a security parameter λ generate a key pair PK and SK,
Sign$_A$**:** Given SK and message M output a signature σ,
Sign$_B$**:** Given SK and message M output a signature σ,
Verify: Given PK, a signature σ and a message M output accept or reject.

The Type I (resp. Type II) forgery will be related to the signatures returned by **Sign**$_A$ (resp. **Sign**$_B$) algorithm. We denote $\mathcal{V}$ be the set of all message and signature pairs such that **Verify** algorithm outputs accept under a fixed PK. Let $\mathcal{V} = \mathcal{V}_I \cup \mathcal{V}_{II}$, where $\mathcal{V}_I$ (resp. $\mathcal{V}_{II}$) denotes the forgery of Type I (resp. Type II).

We briefly mention the security properties of dual-form signature scheme. The formal definition can be found in [GLOW12].
A-I Matching. Given the public key and signing oracle access which returns the signature from **Sign**$_A$, for an adversary it is hard to create a forgery of Type I.
Dual Oracle Invariance. The public key and signing oracle access to both **Sign**$_A$ and **Sign**$_B$ are given to adversary. At some point adversary outputs a challenge message m and challenger returns a signature on m using either **Sign**$_A$ or **Sign**$_B$ with equal probability. Finally adversary outputs a forgery pair (m^*, σ^*), where m^* was not queried earlier to any of the signing oracles. The adversary's probability of producing a Type I forgery when the challenge signature is from **Sign**$_A$ is approximately same as when the challenge signature is from **Sign**$_B$.
B-II Matching. Given the public key and signing oracle access which returns the signature from **Sign**$_B$, for an adversary it is hard to create a Type II forgery.

E Structure-Preserving Signature

E.1 Definition of Structure-Preserving Signature

Let $\mathcal{G}$ denote the bilinear group generator which takes the security parameter λ as input and outputs $(N, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Now we recall the definition of structure-preserving signature scheme from [AGHO11]. Scheme consists of four PPT algorithms which are defined as follows,

Setup(λ) Given the security parameter λ, it outputs a public parameter PP using $\mathcal{G}$.
KeyGen(PP) Given the public parameter PP, it outputs a public and secret key pair (PK, SK) such that PK belongs to $\mathbb{G}_1$ and $\mathbb{G}_2$.
Sign(SK, M) Given the message M and secret key SK, it outputs the signature σ on M, where both message M and signature σ belongs to $\mathbb{G}_1$ and $\mathbb{G}_2$.
Verify(PK, M, σ) Given public key PK, message M and signature σ it outputs *accept* or *reject* based on the satisfiability of certain pairing product equations.

Existential unforgeability under chosen message attack (EUF-CMA) for a structure-preserving signature scheme is defined in the standard way. Informally, adversary is given PK and signing oracle access, probability of returning a valid forgery that differs from the queried message is negligible.

E.2 Security of Dual-Form Abe et al's SPS Scheme

In this section we prove the security of dual-form SPS scheme constructed in §5.1. Before that, we define the forgery classes. For an element z chosen from $\mathbb{Z}_N$, we define $z_1 = z \bmod p_1$ and $z_2 = z \bmod p_2$ and from Chinese Remainder Theorem, we write z as $(z_1, z_2) \in \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2}$. Letting $(z_1, z_2) = (0 \bmod p_1, 1 \bmod p_2)$, we define the forgery classes as follows,

- **Type I**: $\mathcal{V}_I = \{(m^*, \sigma^*) \in \mathcal{V} : (A^*)^z = 1, (B^*)^z = 1, (R^*)^z = 1\}$,

– **Type II**: $\mathcal{V}_{II} = \{(m^*, \sigma^*) \in \mathcal{V} : (A^*)^z \neq 1 \text{ or } (B^*)^z \neq 1 \text{ or } (R^*)^z \neq 1\}$.

Here $\mathcal{V}$ denotes the set of all message-signature pair such that they verifies under the public key. Now we define the subgroup hiding assumptions which will be used to prove the security of our dual-form SPS scheme.

Assumption 5 *Given* $(N, G, H, G_T, e, \mu = \{g_1, h_1, X_1X_2, Z_1Z_2\}, T)$*, it is hard to decide whether* $T \in H_1$ *or* $T \in H$*, where* $X_1X_2 \in G$ *and* $Z_1Z_2 \in H$*.*

Assumption 6 *Given* $(N, G, H, G_T, e, \mu = \{g_1, g_2, h_1, h_1^{t^{-1}}, Z_1Z_2\}, T = g_1^t g_2^{t_2})$*, it is hard to decide whether* $T \in G_1$ *or* $T \in G$*, where* $Z_1Z_2 \in H$ *and* $t \in \mathbb{Z}_N^*$*.*

Assumption 7 *Given* $(N, G, H, G_T, e, g_1, g_2, g_1^{\ell} X_2, g_1^{\ell y_1} X_2', h_1, h_1^{y_1}, h_2)$*, it is hard to compute* $(g_1^{\ell c}, g_1^{\ell y_1 c})$*, for some* $c \neq 0 \bmod p_1$*.*

Assumption 5 is a specific instance of the general subgroup decision assumption [LM15, Assumption 6.2], whereas Assumption 6 and 7 are proved to be hard in the generic group model, see Lemma 7 and 8 in §E.3.

Now we complete the proof of Theorem 4 by using the above defined assumptions in the following lemmas.

Lemma 1. *If* $\mathcal{G}$ *satisfies Assumption 5, then the dual-form SPS scheme is A-I matching.*

Proof. Suppose that there exists an adversary $\mathcal{A}$, given Sign_A oracle access, who can create a Type-II forgery with some non-negligible advantage ϵ. Then we construct an algorithm $\mathcal{B}$ that breaks the Assumption 5. Given $g_1, X_1X_2, h_1, Z_1Z_2, T$, $\mathcal{B}$ chooses w, x, y_1, y_2 uniformly at random from $\mathbb{Z}_N$ and defines all the components of SK except g_2. Then $\mathcal{B}$ computes $PK = (h_1^w, h_1^x, h_1^{y_1}, h_1^{y_2})$ and sends to $\mathcal{A}$. Note that $\mathcal{B}$ does not require g_2 to respond to Sign_A queries. After making polynomial number of signing queries, $\mathcal{A}$ returns a valid forgery $(M^*, \sigma^* = (A^*, D^*, B^*, R^*))$. First $\mathcal{B}$ checks whether (M^*, σ^*) verifies or not. If this check fails, then $\mathcal{B}$ will guess randomly, else $\mathcal{B}$ checks whether the forgery is Type-II or not.

Now write $T = g_1^{t_1} g_2^{t_2}$ and write the forged signature components as, $A^* = g_1^{a^*} g_2^{\gamma_1}, D^* = h_1^{1/a^*}$, $B^* = g_1^{x - a^* w - r^* y_1} (M^*)^{-y_2} g_2^{\gamma_2}$ and $R^* = g_1^{r^*} g_2^{\gamma_3}$, for some $r^*, \gamma_1, \gamma_2, \gamma_3 \in \mathbb{Z}_N$ and $a^* \in \mathbb{Z}_N^*$. In order to check the forgery types, $\mathcal{B}$ verifies the following backdoor verification test (BVT),

$$e(B^*, T) e(A^*, T^w) e(R^*, T^{y_1}) e(M^*, T^{y_2}) \stackrel{?}{=} e(g_1, T^x). \tag{15}$$

If the forgery does not satisfy the above BVT, then $\mathcal{B}$ outputs 1 (i.e., $T \in H$), otherwise he flips a coin $b \in \{0, 1\}$ and outputs b. Since the forgery is valid, it satisfies the signature verification equation. Hence the exponent of the above equation is simplified as $t_2(\gamma_2 + \gamma_1 w + \gamma_3 y_1) \stackrel{?}{=} 0 \bmod p_2$.

$\mathcal{B}$ has to decide whether T is from H_1 or H based on the type of forgery returned by $\mathcal{A}$. Suppose $T \in H_1$, then whether $\mathcal{A}$ outputs Type-I or Type-II forgery, the above BVT holds. On the other hand, when T comes from the whole group H then the above BVT holds provided $\gamma_2 + \gamma_1 w + \gamma_3 y_1 = 0 \bmod p_2$. However, probability that the forgery signature satisfies the condition $\gamma_2 + \gamma_1 w + \gamma_3 y_1 = 0 \bmod p_2$ is negligible (in the security parameter), as w and y_1 are chosen uniformly at random from $\mathbb{Z}_N$. Suppose $T \in H$ and $\gamma_2 + \gamma_1 w + \gamma_3 y_1 \neq 0 \bmod p_2$, then the BVT fails to hold. Hence the forgery returned by $\mathcal{A}$ is Type-II forgery. From our initial assumption, $\mathcal{A}$ outputs a Type-II forgery with some non-negligible advantage. Hence the unsatisfiability of the BVT test over $\mathcal{A}$'s forgery signature can be used by $\mathcal{B}$ to break the Assumption 5 with non-negligible advantage. □

Lemma 2. *If* $\mathcal{G}$ *satisfies Assumption 6, then the dual-form SPS scheme is dual-oracle invariance.*

Proof. Suppose that there exists an adversary $\mathcal{A}$, given an oracle access to both Sign_A and Sign_B algorithms, who can create a Type-II forgery with non-negligible advantage. Then we construct an algorithm $\mathcal{B}$ to break the Assumption 6. Given the instance $(g_1, g_2, h_1, h_1^{t^{-1}}, Z_1Z_2, T = g_1^t g_2^{t_2})$, $\mathcal{B}$

chooses w, x, y_1, y_2 uniformly at random from $\mathbb{Z}_N$ and sets as SK along with g_2 and computes $PK = (h_1^x, h_1^w, h_1^{y_1}, h_1^{y_2})$ which then he sends to $\mathcal{A}$. Since $\mathcal{B}$ knows SK, he can answer for $\mathcal{A}$'s signing oracle queries of both types. In the challenge phase, given the message $\widetilde{M} \in G$, $\mathcal{B}$ embeds the challenge term to construct the signature as,

$$\widetilde{A} = T^{a'}, \widetilde{D} = (h_1^{t^{-1}})^{1/a'}, \widetilde{B} = g_1^x T^{-wa'-y_1 r'} \widetilde{M}^{-y_2}, \widetilde{R} = T^{r'},$$

for r' (resp. a') chosen uniformly at random from $\mathbb{Z}_N$ (resp. $\mathbb{Z}_N^*$). If $T = g_1^{t_1}$, then the challenge signature is distributed as an output of Sign_A algorithm. If $T = g_1^{t_1} g_2^{t_2}$, then the challenge signature is distributed as an output of Sign_B algorithm, since r' and a' are random mod p_2. After receiving the challenge signature, $\mathcal{A}$ makes polynomial number of signing oracle queries of both types. Finally $\mathcal{A}$ returns a valid forgery (M^*, σ^*), where $\sigma^* = (A^*, D^*, B^*, R^*)$. In order to use the output of $\mathcal{A}$ to determine the membership of T, $\mathcal{B}$ must be able to determine whether $\mathcal{A}$ returns a Type-I or Type-II forgery. First $\mathcal{B}$ checks whether (M^*, σ^*) verifies or not. If this check fails, then $\mathcal{B}$ will guess randomly, else $\mathcal{B}$ checks the following backdoor verification test (BVT),

$$e(B^*, Z_1 Z_2) e(A^*, (Z_1 Z_2)^w) e(R^*, (Z_1 Z_2)^{y_1}) e(M^*, (Z_1 Z_2)^{y_2}) \stackrel{?}{=} e(g_1, (Z_1 Z_2)^x). \tag{16}$$

If the forgery does not satisfy BVT, then $\mathcal{B}$ outputs 1, otherwise he flips a coin $b \in \{0, 1\}$ and outputs b. Now we express $T = g_1^{t_1} g_2^{t_2}$, $Z_1 Z_2 = h_1^{\theta_1} h_2^{\theta_2}$ and write the forged signature components as, $A^* = g_1^{a^*} g_2^{\gamma_1}, D^* = h_1^{1/a^*}, B^* = g_1^{x - a^* w - r^* y_1} (M^*)^{-y_2} g_2^{\gamma_2}$ and $R^* = g_1^{r^*} g_2^{\gamma_3}$, for some $r^*, \gamma_1, \gamma_2, \gamma_3, \in \mathbb{Z}_N$ and $a^* \in \mathbb{Z}_N^*$. As before, the exponent of the above BVT test can be simplified to $\theta_2(\gamma_2 + \gamma_1 w + \gamma_3 y_1) \stackrel{?}{=} 0 \mod p_2$.

Now for the forgery returned by $\mathcal{A}$, if the above BVT does not hold, then with probability 1, it is a Type-II forgery. If the above BVT holds for the forgery, then it can be either of Type-I or Type-II. However, we claim that $\mathcal{A}$ can create a Type-II forgery satisfying the above simplified BVT test only with a negligible probability. In order to create a Type-II forgery with $\gamma_2 + \gamma_1 w + \gamma_3 y_1 = 0 \mod p_2$, adversary $\mathcal{A}$ must implicitly determine $-\gamma_2 = \gamma_1 w + \gamma_3 y_1 \mod p_2$ which is a polynomial function with unknown values w and y_1. However during the query phase, no information about w, $y_1 \mod p_2$ is revealed. Hence $\mathcal{A}$ must determine the values of w, $y_1 \mod p_2$ only from the challenge signature. From the challenge signature components, $\mathcal{A}$ can implicitly determine $t_2(a'w + r'y_1) \mod p_2$, which is a single equation with two unknowns w and y_1. This ensures that $\mathcal{A}$ cannot obtain unique values of w and $y_1 \mod p_2$. Hence $\mathcal{A}$ can compute the correct values of w and $y_1 \mod p_2$ only with negligible probability. Thus if BVT holds for the forgery, then with non-negligible probability it is Type-I forgery. □

Lemma 3. *If $\mathcal{G}$ satisfies Assumption 7, then the dual-form SPS scheme is B-II matching.*

Proof. Suppose that there exists an adversary $\mathcal{A}$, given Sign_B oracle access, who can create a Type-I forgery with non-negligible probability. Then we construct an algorithm $\mathcal{B}$ to break the Assumption 7. Given the instance $(g_1, g_2, g_1^{\ell} X_2, g_1^{\ell y_1} X_2', h_1, h_1^{y_1}, h_2)$, $\mathcal{B}$'s goal is to compute $(g_1^{\ell c}, g_1^{\ell y_1 c})$, for some $c \neq 0 \mod p_1$. Now $\mathcal{B}$ chooses w, x, y_2 uniformly at random from $\mathbb{Z}_N$ and sets $PK = (h_1^w, h_1^x, h_1^{y_1}, h_1^{y_2})$ and $SK = (w, x, y_1, y_2)$. Note that $\mathcal{B}$ does not know y_1 and hence part of the secret keys are set implicitly. However $\mathcal{B}$ can answer for Sign_B queries as follows. $\mathcal{B}$ chooses r', γ_1 (resp. a) uniformly at random from $\mathbb{Z}_N$ (resp. $\mathbb{Z}_N^*$) and constructs the signature as,

$$A = g_1^a g_2^{\gamma_1}, D = h_1^{1/a}, B = g_1^{x - aw} (g_1^{\ell y_1} X_2')^{-r'} M^{-y_2}, R = (g_1^{\ell} X_2)^{r'}.$$

Here $\mathcal{B}$ implicitly sets $r = r'\ell$ and it is easy to verify that the above signature is properly distributed. After making polynomial number of signing queries, $\mathcal{A}$ returns a valid forgery (M^*, σ^*), where $\sigma^* = (A^*, D^*, B^*, R^*) \in G \times H \times G^2$. First $\mathcal{B}$ checks whether (M^*, σ^*) verifies or not. If this check fails, then $\mathcal{B}$ aborts, else $\mathcal{B}$ checks whether the forgery is Type-I or Type-II by

$$e(A^*, h_2) \stackrel{?}{=} 1, e(B^*, h_2) \stackrel{?}{=} 1 \text{ and } e(R^*, h_2) \stackrel{?}{=} 1.$$

If any of the above checks fail to hold, then the forgery is Type-II and $\mathcal{B}$ will abort. Otherwise it is a Type-I forgery which does not have G_2 components. Hence we write the forgery signature as,

$A^* = g_1^{a^*}, D^* = h_1^{1/a^*}, B^* = g_1^{x-a^*w} g_1^{-\ell y_1 r^*} (M^*)^{-y_2}$ and $R^* = g_1^{\ell r^*}$, for $r^* \in \mathbb{Z}_N$ and $a^* \in \mathbb{Z}_N^*$. Now $\mathcal{B}$ computes $g_1^{\ell r^*} = R^*$, $g_1^{\ell y_1 r^*} = (B^*)^{-1} g_1^x (A^*)^{-w} (M^*)^{-y_2}$ and returns to his challenger as a solution for Assumption 7. Since $\mathcal{A}$ returns a valid forgery, which means it should satisfy Equation 4. In particular, the condition $e(R^*, h_1) \neq 1$ ensures that $r^* \neq 0 \bmod p_1$. Thus $\mathcal{B}$ relays the non-trivial solution for Assumption 7. □

E.3 Hardness of SGH Variants

We prove the hardness of Assumption 6 and 7 defined in §E.2. We follow the notation from [GLOW12, Appendix B]. In particular, for $j \in \{1, 2, T\}$, random element x from the group G_j is denoted as $X = (X_1, X_2)_j$, where X_1 (resp. X_2) is the indeterminate that corresponds to the subgroup of order p_1 (resp. p_2).

Lemma 7 *Assumption 6 holds in the generic group model if it is hard to find a non-trivial factor of* $N = p_1 p_2$.

Proof. Adversary is given with $(1,0)_1, (0,1)_1, (L,X)_1, (LY,W)_1, (P_1,P_2)_1$, $(1,0)_2, (0,1)_2, (Y,0)_2$ and $(Q_1, Q_2)_2$. Then adversary must compute $(Lc, 0)_1, (LYc, 0)_1$ such that $c \neq 0$. In order to remove X (resp. W) from $(L,X)_1$ (resp. $(LY,W)_1$), adversary requires to have $(0,X)_1$ (resp. $(0,W)_1$), which is not possible from the generic group operations, as these terms are linearly independent with the terms that are given to the adversary. □

Lemma 8 *Assumption 7 holds in the generic group model if it is hard to find a non-trivial factor of* $N = p_1 p_2$.

Proof. Adversary is given with $(1,0)_1, (0,1)_1, (P_1,P_2)_1, (1,0)_2, (T_1^{-1}, 0)_2, (Q_1,Q_2)_2$ and $(T_1,T_2)_1$. Then adversary must decide whether T_2 is zero or not. In order to decide $(T_1,T_2)_1$, adversary requires to have $(0,a)_2$ for some $a \neq 0$, which is not possible from the generic group operations, as $(0,a)_2$ is linearly independent with the terms that are given to the adversary. □

F Group signature scheme

F.1 Definition of Group Signatures

We recall the definition of group signature scheme and its security from Boyen and Waters [BW07]. The scheme consists of five PPT algorithms which are defined as follows,

Setup(λ) Given the security parameter λ, it outputs the public parameter PP, master enrollment key MK and tracing key TK.

Enroll(PP, MK, ID) Given identity of the signer $ID \in \{0,1\}^\kappa$ and public parameter PP with master enrolling key MK, it outputs a private signing key K_{ID}, where $\kappa = poly(\lambda)$.

Sign(PP, K_{ID}, M) Given the message $M \in \{0,1\}^m$, public parameter PP and private signing key K_{ID} on ID, it outputs a signature σ, where $m = poly(\lambda)$.

Verify(PP, M, σ) Given message M and signature σ it outputs *accept* or *reject*.

Trace(PP, σ, TK) Given a signature σ and tracing key TK it outputs an identifier or $\perp$.

We briefly mention the security properties of group signature scheme. The formal definition can be found in [BMW03].

Fully anonymous. Given the public parameter, adversary is given access to both private signing key queries and signature queries. In the challenge phase adversary gives a random message along with two identities ID_0 and ID_1. Now the challenger chooses one of the identity and sign the message on behalf. Given the challenge signature adversary have to guess the correct signer identity.

Fully traceable. Given the public parameter, adversary is given access to private signing key, signature queries and tracing queries. After polynomial number of queries adversary have to output a valid forgery whose components are not been used in any of oracle queries.

F.2 Dual-Form HSS in Composite-order

The dual-form two-level HSS consists of six PPT algorithms which are defined as follows.

Setup(λ) Run the bilinear group generator $\mathcal{G}$ on λ which outputs $(N = p_1p_2p_3, G, G_T, e, g_1, g_{2,3}, g_3)$. Choose $h_1, u, u_1, v_0, \{v_i\}_{i=1}^m$ uniformly at random from G_1 and choose α uniformly at random from $\mathbb{Z}_N$, for $m = poly(\lambda)$. Then output the public key PK as $(N, G, G_T, e, g_1, u_1, e(g_1, h_1), g_1^\alpha, u, v_0, \{v_i\}_{i=1}^m)$ and secret key SK as $(\alpha, h_1, g_3, g_{2,3}, \{s_i\}_{i=1}^{2^k})$, where $s_i \in \mathbb{Z}_N$ is a unique identifier of the user in the system, for $k = poly(\lambda)$.

Extract$_A(PK, SK, ID)$ Choose r and X_3, X_3' uniformly at random from $\mathbb{Z}_N$ and G_3 respectively. Compute and output the first level signature $K_{ID} := (K_1, \ldots, K_4)$, where

$$K_1 = (h_1 u_1^{-r})^{\frac{1}{\alpha + s_{ID}}} X_3, \quad K_2 = g_1^r X_3', \quad K_3 = g_1^{s_{ID}}, \quad K_4 = u^{s_{ID}}.$$

Extract$_B(PK, SK, ID)$ Choose r and $X_{2,3}, X_{2,3}'$ uniformly at random from $\mathbb{Z}_N$ and $G_{2,3}$ respectively. Compute and output $K_{ID} := (K_1, \ldots, K_4)$, where

$$K_1' = (h_1 u_1^{-r})^{\frac{1}{\alpha + s_{ID}}} X_{2,3}, \quad K_2' = g_1^r X_{2,3}', \quad K_3 = g_1^{s_{ID}}, \quad K_4 = u^{s_{ID}}.$$

Sign$_A(PK, K_{ID}, M)$ Parse the message $M = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m$ and choose s uniformly at random from $\mathbb{Z}_N$. Compute and output the second level signature components $\sigma = (S_1, \ldots, S_5)$, where

$$S_1 = K_1, \quad S_2 = K_2, \quad S_3 = K_3, \quad S_4 = K_4\big(v_0 \textstyle\prod_{i=1}^m v_i^{\mu_i}\big)^s, \quad S_5 = g_1^s.$$

Sign$_B(PK, K_{ID}, M)$ Parse the message $M = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m$ and choose s uniformly at random from $\mathbb{Z}_N$. Compute and output $\sigma = (S_1, \ldots, S_5)$, where

$$S_1 = K_1', \quad S_2 = K_2', \quad S_3 = K_3, \quad S_4 = K_4\big(v_0 \textstyle\prod_{i=1}^m v_i^{\mu_i}\big)^s, \quad S_5 = g_1^s.$$

Verify(PK, ID, M, σ) Parse the message $ID||M$, signature σ and check that

$$e(S_1, g_1^\alpha S_3) e(S_2, u_1) \stackrel{?}{=} e(g_1, h_1) \text{ and } e(S_4, g_1) \stackrel{?}{=} e(u, S_3) e\big(v_0 \textstyle\prod_{i=1}^m v_i^{\mu_i}, S_5\big) \tag{17}$$

If any of the above checks fail to hold, then output reject, otherwise output accept.

Here we omit the security proof, as it is very similar to the one described in [YCZY14, Appendix A].