Full version of an extended abstract published in Asiacrypt 2016.

Déjà Q All Over Again: Tighter and Broader Reductions of q-Type Assumptions

Melissa Chase
Microsoft Research Redmond
melissac@microsoft.com

Mary Maller
University College London
mary.maller.15@ucl.ac.uk

Sarah Meiklejohn
University College London
s.meiklejohn@ucl.ac.uk

Abstract

In this paper, we demonstrate that various cryptographic constructions — including ones for broadcast, attribute-based, and hierarchical identity-based encryption — can rely for security on only the static subgroup hiding assumption when instantiated in composite-order bilinear groups, as opposed to the dynamic q-type assumptions on which their security previously was based. This specific goal is accomplished by more generally extending the recent Déjà Q framework (Chase and Meiklejohn, Eurocrypt 2014) in two main directions. First, by teasing out common properties of existing reductions, we expand the q-type assumptions that can be covered by the framework; i.e., we demonstrate broader classes of assumptions that can be reduced to subgroup hiding. Second, while the original framework applied only to asymmetric composite-order bilinear groups, we provide a reduction to subgroup hiding that works in symmetric (as well as asymmetric) composite-order groups. As a bonus, our new reduction achieves a tightness of $\log(q)$ rather than q.

1 Introduction

In cryptography, the provable security paradigm crucially relies on the existence of hard mathematical problems. To prove the security of a candidate cryptographic construction, one must demonstrate that any adversary that can break its security can be used to construct another adversary that can break the underlying mathematical problem; if the problem is assumed to be hard, then it logically follows that the construction is secure.

To be confident in the security of a construction, we must therefore also be confident in the underlying assumption; i.e., the assumption that the given mathematical problem is hard. Cryptographic assumptions come in many forms, and confidence in them can be gained through various means: one can perform cryptanalysis on the problem and attempt to break it, prove its security in the generic group model [40], or generalize multiple assumptions using a construct like the *uber-assumption* [12, 16] to provide general lower bounds on security.

As a field, cryptography has in the past decade become increasingly tolerant of assumptions that are new, not particularly well understood, and in some cases even "hard to untangle from the constructions which utilize them" [26]. While there are of course good reasons for doing so (e.g., driving the state of the art forward), and it is demonstrably impossible to reduce every construction to a simple assumption like DDH, the growth in the volume and complexity of new assumptions nevertheless provides an opportunity to revisit this landscape of assumptions and attempt to simplify and systematize it where possible.

Our specific focus in this paper is the class of *q-type assumptions*, in which the assumption is not static, but rather can grow dynamically; e.g., the decisional q-wBDHI (weak Bilinear Diffie-Hellman Inversion) assumption [12] says that given $(g, g^c, g^b, g^{b^2}, \dots, g^{b^q})$, it should be hard to distinguish $e(g,g)^{b^{q+1}c}$ from random. These assumptions are closely tied to the schemes that rely on them for security, as the value q is often equal to the number of oracle calls that can be made in a reduction; e.g., in identity-based encryption (IBE), a distinct value from the assumption is used within the reduction to respond to each of q key extraction queries. Moreover, q-type assumptions become stronger as q grows, and the time to recover the discrete logarithm scales inversely with q [22].

In a recent paper [18], Chase and Meiklejohn demonstrated the potential to move away from q-type assumptions by demonstrating that certain types of q-type assumptions (under the umbrella of the uber-assumption) were implied by the static subgroup hiding assumption [14] in asymmetric composite-order groups. Specifically, they demonstrated a reduction — with looseness q — to the subgroup hiding assumption from all q-type assumptions that either (1) gave out functions on only one side of the pairing and asked the adversary to distinguish elements in the source group or (2) gave out functions on both sides of the pairing and asked the adversary to compute an element in the source group. Following Wee [44], we dub their set of techniques and results the "Déjà Q framework."

1.1 Our contributions

In this paper, we seek to expand the applicability of the Déjà Q framework to encompass wider classes of assumptions and to apply to settings that are used more commonly in cryptographic constructions. In particular, we provide the following three main contributions:

Broader classes of assumptions. In terms of specific schemes and assumptions, the original Déjà Q framework implied that the Dodis-Yampolskiy PRF [23] and the q-SDH assumption [11] could be reduced to subgroup hiding. To broaden not only the class of assumptions but also the concrete applicability of the framework, we capture computational and decisional uber-assumptions in the target group, including commonly used q-type assumptions such as q-BDHE [12] and q-wBDHI. We also demonstrate techniques for translating concrete schemes — in particular, the BGW broadcast encryption scheme [13], the BBG hierarchical identity-based encryption scheme [12], the Waters attribute-based encryption scheme [41], and the ACF identity-based key encapsulation mechanism [1] — that rely on the symmetric versions of these assumptions for security into asymmetric composite-order bilinear groups, where they can then be reduced to subgroup hiding.

Tighter reductions. We provide a new reduction from both computational and decisional uber-assumptions in the target group to subgroup hiding. Our new reduction requires adding at least one additional prime to the factorization of N, but it achieves logarithmic — rather than linear — tightness. These results can then be applied to any scheme based on these assumptions, including the ones mentioned above, which directly gives a tightly (or almost tightly, depending on ones preferred terminology) secure instantiation, albeit in a somewhat inefficient setting.

Symmetric and asymmetric groups. The original Déjà Q framework could operate only in asymmetric composite-order bilinear groups (or composite-order groups where no pairing existed), of which only one construction is known [15, 37]. Our new proof works in both symmetric and asymmetric settings, thus allowing us to consider the more "usual" instantiations of composite-order bilinear groups.

1.2 Our techniques

In terms of the techniques we use, our proof in Section 3 that computational and decisional uber-assumptions in the target group can be reduced to subgroup hiding is closely based on the proof in the original Déjà Q framework for computational uber-assumptions in the source group. To achieve this, we observe that reductions frequently treat group generators in separate ways; i.e., separate sets of generators are used to answer separate types of queries, and the reduction crucially relies on this separation to ensure that the adversary can't test the relationships between different objects as they (separately) incorporate additional randomness or otherwise shift in value. By explicitly acknowledging this usage in our statement of the uber-assumption, we can treat the separate generators in different ways in our reductions and thus extend the results to the target group. To further demonstrate how to securely move symmetric constructions into the asymmetric setting, where they can then be covered by these results, we rely on a recent set of techniques due to Abe et al. [4] for doing automated symmetric-to-asymmetric translations.

Next, in Section 4, we consider a modified version of this proof strategy, where in each game hop we double the amount of randomness included in the assumption. To do this, we require three subgroups instead of two, meaning we can write $G = G_1 \times G_2 \times G_3$. As in the original Déjà Q framework, we start by shifting the variables used in the q-type assumption from G_1 into G_2 and G_3, which following the usual dual-system technique we can argue goes unnoticed by *subgroup hiding* [14]. We then change the variables in G_2 and G_3 to take on entirely new values, which again following the dual-system technique we can argue goes unnoticed by *parameter hiding* [29]. Now, however, instead of continuing to shift the same variables from G_1 into G_2 and change them one by one, we shift the new variables from G_3 into G_2, so that G_2 has effectively doubled the number of new variables it contains. By repeating this process of shifting all the variables from G_2 into

G_3, changing them, and shifting them back, we achieve the same outcome as the original framework of having ℓ sets of variables in G_2, but using $\log_2(\ell)$ game transitions instead of ℓ.

While one additional subgroup suffices to achieve this tighter reduction in asymmetric bilinear groups, our reduction relies on the use of subgroup generators that would break subgroup hiding in symmetric groups. To address this, our new reduction brings in certain aspects of the more traditional application of the dual-system technique to constructions (rather than assumptions) [32, 31, 29, 10, 20], and in particular a recent result due to Wee [44] that used an adaption of the Déjà Q framework to reduce both an IBE scheme and a broadcast encryption scheme to subgroup hiding. We thus demonstrate that by folding in random values from a fourth subgroup, we can sufficiently "mask" the subgroups to push through the same reduction in symmetric groups. Thus, while our results in Section 3 apply to versions of concrete constructions translated into the asymmetric setting (but otherwise unmodified), our results in Section 4 provide tighter reductions for the (original) symmetric versions in which additional randomness is incorporated when instantiated in groups with two additional subgroups, or for asymmetric versions with an additional subgroup (but no additional randomness).

1.3 Related work

Our work closely builds on the Déjà Q framework due to Chase and Meiklejohn [18]. In order to go beyond the original set of contributions, we draw on certain aspects of the dual-system technique [42, 32, 31], the notion of parameter hiding [29, 30], and the general notion of subgroup hiding [9]. For our results in the symmetric setting, we draw on ideas in a recent work by Wee [44], who extended the original Déjà Q framework but focused specifically on constructions for broadcast encryption and IBE.

The search for tight reductions goes back to the paper of Bellare and Rogaway [8], and the results are extensive. To compare with the results most similar to ours, we focus on results for pairing-based primitives, where much related work has provided (almost) tight reductions for various primitives, including identity-based encryption [21, 3, 10, 28, 35], inner product encryption [38], authenticated key exchange [7], and quasi-adaptive non-interactive zero-knowledge proofs [36, 24]. Each of these results focuses on a specific construction, and employs a specific set of techniques to achieve tight security. (One exception is a paper by Attrapadung, Hanoaka, and Yamada [6] that gives an abstraction from which several different IBE variants can be constructed. This work, however, is still focused on IBE and on a particular construction approach.) By presenting our results at the level of assumptions, we can instead prove tight security for an entire class of constructions; i.e., constructions that are instantiated in appropriate groups and have been previously proved secure under an appropriate class of q-type assumptions. To the best of our knowledge, we are thus the first to use the dual-system technique to provide a tightly secure reduction in a more general setting. Finally, we note that while much of the previous work has focused on reductions whose running time is linear in the security parameter, our reduction is linear in $\log(q)$, which in practice may be a much smaller number.

2 Definitions and Notation

2.1 Preliminaries

If x is a binary string then $|x|$ denotes its bit length. If S is a finite set then $|S|$ denotes its size and $x \xleftarrow{\$} S$ denotes sampling a member uniformly from S and assigning it to x. $\lambda \in \mathbb{N}$ denotes the security parameter and 1^λ denotes its unary representation. $[n]$ denotes the set $\{1, \ldots, n\}$.

Algorithms are randomized unless explicitly noted otherwise. "PT" stands for "polynomial-time." By $y \leftarrow A(x_1, \ldots, x_n; R)$ we denote running algorithm A on inputs $x_1, \ldots, x_n$ and random coins R and assigning its output to y. By $y \xleftarrow{\$} A(x_1, \ldots, x_n)$ we denote $y \leftarrow A(x_1, \ldots, x_n; R)$ for coins R sampled uniformly at random. By $[A(x_1, \ldots, x_n)]$ we denote the set of values that have positive probability of being output by A on inputs $x_1, \ldots, x_n$. Adversaries are algorithms.

We use games in definitions of security and in proofs. A game G has a MAIN procedure whose output is the output of the game. $\Pr[\mathsf{G}]$ denotes the probability that this output is true.

2.2 Basic bilinear groups

A *bilinear group* is a tuple $\mathbb{G} = (N, G, H, G_T, e)$, where N is either prime or composite, $|G| = |H| = kN$ and $|G_T| = \ell N$ for $k, \ell \in \mathbb{N}$, all elements of G, H, and G_T are of order at most N, and $e : G \times H \to G_T$ is a *bilinear map*: it is efficiently computable, satisfies $e(A^x, B^y) = e(A, B)^{xy}$ for all $A \in G$, $B \in H$, and $x, y \in \mathbb{Z}/N\mathbb{Z}$ (bilinearity), and if $e(A, B) = 1$ for all $B \in H$ then $A = 1$ and vice versa if this holds for all $A \in G$ (non-degeneracy). We use $\mathsf{BilinearGen}$ to denote the algorithm by which bilinear groups are generated.

When G and H are cyclic, the description of the group may include their respective generators g and h. If the groups can be decomposed as $G = G_1 \times G_2$ and $H = H_1 \times H_2$, the description of the group may include information about these subgroups and their generators; additionally, the number of cyclic subgroups may be provided as an argument n to $\mathsf{BilinearGen}$.

2.3 Subgroup hiding and parameter hiding

We highlight two structural properties of bilinear groups — *subgroup hiding* and *parameter hiding* — that are essential to the Déjà Q framework, using adapted versions of the definitions given by Chase and Meiklejohn [18].

Assumption 2.1 (Subgroup hiding)**.** *For $n \in \mathbb{N}$ and a bilinear group generation algorithm* $\mathsf{BilinearGen}(\cdot,\cdot)$*, define $\mathbf{Adv}^{sgh}_{\mathcal{A}}(\lambda) = 2Pr[SGH^{\mathcal{A}}_{\mu}(\lambda)] - 1$, where $SGH^{\mathcal{A}}_{\mu}(\lambda)$ is defined as follows:*

```
MAIN $SGH^{\mathcal{A}}_{\mu}(\lambda)$
$b \xleftarrow{\$} \{0,1\}$; $(N, G, H, G_T, e, \mu) \xleftarrow{\$} \mathsf{BilinearGen}(1^{\lambda}, n)$
if $(b = 0)$ then $w \xleftarrow{\$} G$
if $(b = 1)$ then $w \xleftarrow{\$} G_1$
$b' \xleftarrow{\$} \mathcal{A}(N, G, H, G_T, e, \mu, w)$
return $(b' = b)$
```

Then subgroup hiding *holds in G_1 with auxiliary information μ if for all PT adversaries $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $\mathbf{Adv}^{sgh}_{\mathcal{A}}(\lambda) < \nu(\lambda)$.*

Subgroup hiding is defined analogously for G_2, $G_{1,T}$, and $G_{2,T}$ (where $G_{1,T}$ and $G_{2,T}$ are cyclic subgroups of G_T), and the auxiliary information μ is designed to capture additional subgroup generators that may also be given out (with the observation that revealing certain subgroup generators might allow one to trivially distinguish subgroups when using a canceling pairing, so one must be careful with what μ contains). If we switch between different subgroups rather than one subgroup and the full group — e.g., between G_2 and G_{23}, as we do in Section 4 — then we say subgroup hiding holds *between* the subgroups.

To elaborate on the point about μ, subgroup hiding can be trivially broken if the adversary has knowledge of certain generators; e.g., if an adversary is given a value w and asked to determine if it is in G or G_1, knowledge of the generator h_2 allows it to check if $e(w, h_2) = 1$ and trivially break subgroup hiding. To avoid this, the many variants of subgroup hiding used in the literature often specify which subgroup elements the adversary can see [25, 34, 33, 27, 17, 39], and the rules about which generators can be given out have been codified in the *general subgroup decision* assumption due to Bellare, Waters, and Yilek [9]. The variants of subgroup hiding that we use in Sections 3 and 4 are specific instantations of this general assumption.

Definition 2.2 (Extended parameter hiding)**.** *For $m, n \in \mathbb{N}$ and a bilinear group $(N, G, H, G_T, e, \mu) \in [\mathsf{BilinearGen}(1^{\lambda}, n)]$, we say* extended parameter hiding *holds with respect to a family of functions $\mathcal{F}$, auxiliary information* aux*, and a pair of subgroups (G_{i_1}, G_{i_2}) if for all $g_{i_1} \in G_{i_1}$ and $g_{i_2} \in G_{i_2}$, the distribution $\{g_{i_1}^{f(\vec{x})} g_{i_2}^{f(\vec{x})}, a(\vec{x})\}_{f \in \mathcal{F}, a \in \mathsf{aux}}$ is identical to $\{g_{i_1}^{f(\vec{x})} g_{i_2}^{f(\vec{x}')}, a(\vec{x})\}_{f \in \mathcal{F}, a \in \mathsf{aux}}$ for $\vec{x}, \vec{x}' \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^m$.*

Chase and Meiklejohn proved [18, Lemma 5.2] that their original definition of extended parameter hiding (which used $n = 2$) holds in composite-order bilinear groups with respect to all polynomial functions and the version of aux that we require in Section 3. In Section 4, however, we consider a group with $n > 2$ subgroups and we want parameter hiding to hold across subgroups beyond G_1 and G_2. We thus prove that parameter hiding still holds in this setting as long as the orders of G_{i_1} and G_{i_2} have no primes in common and the auxiliary information is not in G_{i_2}.

Lemma 2.3. *For all $m, n \in \mathbb{N}$, bilinear groups $(N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^{\lambda}, n)]$ where $N = p_1 \cdot \ldots \cdot p_n$, (i_1, i_2) such that $1 \leq i_1, i_2 \leq n$, and for the class $\mathcal{F}$ of all polynomials $f(\cdot)$ over $\mathbb{Z}/N\mathbb{Z}$, if $\gcd(p_{i_1}, p_{i_2}) = 1$ and if for all $a \in \mathsf{aux}$, $a(\cdot) \in A$ such that $\gcd(|A|, p_{i_2}) = 1$, then the distribution over $\{g_{i_1}^{f(\vec{x})} g_{i_2}^{f(\vec{x})}, a(\vec{x})\}_{f \in \mathcal{F}, a \in \mathsf{aux}}$ is identical to the distribution over $\{g_{i_1}^{f(\vec{x})} g_{i_2}^{f(\vec{x}')}, a(\vec{x})\}_{f \in \mathcal{F}, a \in \mathsf{aux}}$ for $\vec{x}, \vec{x}'_1 \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^m$.*

Proof. For any polynomial $f(\cdot)$, one can compute $g_{i_1}^{f(\vec{x})}$ knowing just the value of $x_j \bmod p_{i_1}$ for all j, $1 \leq j \leq m$, and can similarly compute $g_{i_2}^{f(\vec{x})}$ knowing just the value of $x_j \bmod p_{i_2}$ for all j, $1 \leq j \leq m$. If $\gcd(p_{i_1}, p_{i_2}) = 1$ and the functions in aux reveal no information about $x_j \bmod p_{i_2}$, then by the Chinese Remainder theorem the values of $x_j \bmod p_{i_2}$ are independent of all the other values, so this is identical to using an independent x'_j for the g_{i_2} values. □

3 Uber-assumptions in the target group

In this section, we consider how to capture new classes of assumptions within the Déjà Q framework [18]. In particular, we first prove in Section 3.1 that decisional and computational uber-assumptions in the target group are implied — through the repeated application of subgroup hiding and parameter hiding — by assumptions with significant amounts of randomness folded into particular subgroups. (The framework previously covered only computational assumptions in the source group, which are implied by computational assumptions in the target group, or "one-sided" decisional assumptions in the source group; i.e., assumptions where meaningful functions could be given out on only one side of the pairing.)

Next, in Section 3.2, we show that the computational variant of the transitioned uber-assumption is so weak that it holds by a statistical argument; thus, the computational uber-assumption can be implied solely by subgroup hiding. By relying on an additional mild subgroup hiding assumption in the target, we can show the same results for decisional variants as well; i.e., we can show that the decisional uber-assumption is implied by three variants of subgroup hiding.

Finally, in Section 3.3, we observe that many examples of uber-assumptions (including widely used q-type assumptions) have been used only in symmetric bilinear groups to date, making it difficult to cover them directly with our analysis. (In Section 4, we do provide ways to cover the symmetric setting, but this requires an extra prime in the order of the group.) We thus demonstrate how to convert popular symmetric assumptions into asymmetric variants using techniques due to Abe et al. [4]. All of our converted symmetric schemes — e.g., the BGW broadcast encryption scheme [13] and the Waters attribute-based encryption scheme [41] — rely for security on q-type decisional uber-assumptions of the appropriate form, so our results demonstrate the security of these schemes when instantiated in groups where subgroup hiding holds.

3.1 Reducing asymmetric assumptions to weaker variants

In the uber-assumption [18, Assumption 4.1], the adversary is given three sets of values with respect to a set of c variables $\vec{x}$: a generator $g \in G$ raised to a set of functions $R(\vec{x})$, a generator $h \in H$ raised to a set of functions $S(\vec{x})$, and the value $e(g,h)$ raised to a set of functions $T(\vec{x})$ (where $g^{R(\vec{x})}$ is used as shorthand for $\{g^{\rho_i(\vec{x})}\}_{i=1}^{r}$ for $R = \langle \rho_1(\vec{x}), \dots, \rho_r(\vec{x}) \rangle$, and similarly for S and T). The adversary is then asked to either compute $e(g,h)^{f(\vec{x})}$ (in the computational assumption in the target group) or distinguish it from random.

This definition captures a broad range of q-type assumptions, but in some cases it may be instructive to explicitly identify the qualities of the assumption that are used in the reduction. In particular, constructions that use the dual-system technique must add noise into group elements in such a way that valuable information is hidden but one can nevertheless continue to correctly perform operations (e.g., decryption) without noticing the added noise. This is often accomplished by using two separate generators that are primarily used for separate operations — e.g., in the case of identity-based encryption, one generator is used to create the parameters and the other to form the challenge ciphertext — and this separation is acknowledged in the assumption. For example, the (symmetric) q-BDHE assumption [12] says that given $(g, g^s, \{g^{a^i}\}_{i \in [2q], i \neq q+1})$, it should be hard to distinguish $e(g,g)^{a^{q+1}s}$ from random.

We thus modify slightly the original definition of the uber-assumption to (1) make explicit the role of two generators h and $\hat{h}$, the former of which we move into a subgroup to provide the necessary correctness and the latter of which we keep in the full group to provide the necessary hiding guarantee, and (2) combine computational and decisional assumptions into the same definition so we can cover them both in our main theorem.

Assumption 3.1 (Uber-assumption). *For an adversary $\mathcal{A}$, let $\mathbf{Adv}_{\mathcal{A}}^{comp\text{-}uber}(\lambda) = Pr[comp\text{-}UBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)]$ and $\mathbf{Adv}_{\mathcal{A}}^{dec\text{-}uber}(\lambda) = 2Pr[dec\text{-}UBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda) - 1$, where $\mathsf{type}\text{-}UBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$ is defined as follows for $\mathsf{type} \in \{comp, dec\}$:*

```
$\underline{\textsc{main}\ \text{type-}UBER^{\mathcal{A}}_{c,R,S,T,f}(\lambda)}$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g \xleftarrow{\$} G$, $h, \hat{h} \xleftarrow{\$} H$
$x_1, \dots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g, \hat{h}, g^{R(\vec{x})}, h^{S(\vec{x})}, e(g, h)^{T(\vec{x})})$
$\mathsf{chal} \leftarrow e(g, \hat{h})^{f(\vec{x})}$
$\textit{return}\ \text{type-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

```
$\underline{\textit{comp}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})}$
$y \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{inputs})$
$\textit{return}\ (y = \mathsf{chal})$
```

```
$\underline{\textit{dec}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})}$
$b \xleftarrow{\$} \{0, 1\}$
$\textit{if}\ (b = 0)\ \textit{then}\ y \xleftarrow{\$} G_T$
$\textit{if}\ (b = 1)\ \textit{then}\ y \leftarrow \mathsf{chal}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{inputs}, y)$
$\textit{return}\ (b' = b)$
```

Then the uber-assumption in the target group holds if for all PT algorithms $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $\mathbf{Adv}^{uber}_{\mathcal{A}}(\lambda) < \nu(\lambda)$.

We now proceed to prove a theorem analogous to the one in the original Déjà Q framework [19, Theorem 4.8], but which treats these different bases in H in different ways. For ease of exposition, we make explicit the original assumption used in this proof, which (with our additional generator $\hat{h}$ added) is as follows:

Assumption 3.2. *For a bilinear group $\mathbb{G} = (N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$, $\ell \in \mathbb{N}$, and classes of functions R, S, T, and f (as defined in the uber-assumption in Assumption 3.1), given*

$$\mathsf{inputs} = (\mathbb{G}, g_1 g_2^{\sum_{i=1}^{\ell} r_i}, \hat{h}, \{g_1^{\rho_k(\vec{x})} g_2^{\sum_{i=1}^{\ell} r_i \rho_k(\vec{x}_i)}\}_{k=1}^{r}, h_1^{S(\vec{x})}, e(g_1, h_1)^{T(\vec{x})})$$

for $g_1 \xleftarrow{\$} G_1$, $g_2 \xleftarrow{\$} G_2 \backslash \{1\}$, $\hat{h} \xleftarrow{\$} H$, $h_1 \xleftarrow{\$} H_1 \backslash \{1\}$, and $r_1, \dots, r_\ell, \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, $\vec{x}, \vec{x}_1, \dots, \vec{x}_\ell \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c$, no PT adversary has more than negligible advantage when playing type-$\textsc{play}(\lambda, \mathsf{inputs}, e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)})$.

Theorem 3.3. *For a bilinear group $\mathbb{G} = (N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$, consider the uber-assumption in the target group parameterized by (c, R, S, T, f). Then this is implied by Assumption 3.2 if*

1. *subgroup hiding holds in G_1 with $\mu = \{g_2, h_1\}$;*
2. *subgroup hiding holds in H_1 with $\mu = \{g_1\}$; and*
3. *extended parameter hiding holds with respect to $\mathcal{F} = R \cup \{f\}$ and $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ for all $h_1 \in H_1$.*

In particular, for $\ell \in \mathbb{N}$ we have that

$$\mathbf{Adv}^{uber}_{\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{sgh}_{\mathcal{B}_0}(\lambda) + \mathbf{Adv}^{sgh}_{\mathcal{C}_0}(\lambda) + \ell \mathbf{Adv}^{sgh}_{\mathcal{B}_i}(\lambda) + \mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda).$$

A proof of this theorem can be found in Appendix A. Intuitively, the outline is similar to that of the original proof: to start, all elements in G are first shifted into G_1, and elements using h as the base are shifted into H_1. Elements using $\hat{h}$ remain in the full group H (this is our main point of divergence from the original Déjà Q proof). We argue that both of these changes go unnoticed by subgroup hiding. Then, the elements in G_1 are added into G_2, which we again argue goes unnoticed by subgroup hiding. The elements in G_2 are then switched to use a new set of variables $\vec{x}_1$, which we argue is identical by parameter hiding. Now, we repeat this process of adding the original elements from G_1 into G_2 and switching them to a new set of variables, until — after ℓ transitions — we end up with ℓ sets of variables in G_2.

3.2 Reducing asymmetric assumptions to subgroup hiding

We now deal separately with the case of computational and decisional assumptions, as decisional assumptions require an extra assumption on the indistinguishability of random elements in $G_{2,T}$ and random elements in

G_T (we use $G_{i,T}$ to denote the i^{th} subgroup of G_T). For both, however, we first recall two relevant components from the Déjà Q framework: the matrix V defined as

$$V = \begin{bmatrix} 1 & \rho_1(\vec{x}_1) & \rho_2(\vec{x}_1) & \cdots & \rho_q(\vec{x}_1) & f(\vec{x}_1) \\ 1 & \rho_1(\vec{x}_2) & \rho_2(\vec{x}_2) & \cdots & \rho_q(\vec{x}_2) & f(\vec{x}_2) \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ & & & \ddots & & \\ 1 & \rho_1(\vec{x}_\ell) & \rho_2(\vec{x}_\ell) & \cdots & \rho_q(\vec{x}_\ell) & f(\vec{x}_\ell) \end{bmatrix} \tag{1}$$

and a lemma that relates the linear independence of the polynomials with the invertibility of V as follows:

Lemma 3.4. [18] *For all $\lambda \in \mathbb{N}$, if the functions in $R \cup \{f\}$ are linearly independent and of maximum degree $poly(\lambda)$, $\ell = q+2$ for $q = poly(\lambda)$, and $N = p_1 \cdot \ldots \cdot p_n$ for $n = poly(\lambda)$ distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$, then with all but negligible probability the matrix V is invertible.*

We also make explicit the argument used in the Déjà Q framework concerning the multiplication of this matrix with a random vector.

Lemma 3.5. *If V is invertible, then the distribution over $\vec{r} \cdot V$ for $r_1, \ldots, r_{q+2} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ is uniformly random.*

Proof. Define $\vec{y} \leftarrow \vec{r} \cdot V$, and consider the set of all vectors of length $q+2$ over $\mathbb{Z}/N\mathbb{Z}$. Since $\vec{r}$ and $\vec{y}$ are both members of this set, multiplication by V maps the set to itself; as V is furthermore invertible, it is a permutation over this set. Thus, sampling $\vec{r}$ uniformly at random and multiplying by V yields a vector $\vec{y}$ that is also distributed uniformly at random. □

3.2.1 Computational assumptions.

For computational assumptions, we can now argue directly that, by transitioning to Assumption 3.2, we reach an assumption so weak that it holds by a statistical argument. Thus, the computational uber-assumption reduces directly to subgroup hiding.

Proposition 3.6. *For a bilinear group $\mathbb{G}$ of order N, the computational uber-assumption parameterized by (c, R, S, T, f) holds in the target group if*

1. *subgroup hiding holds in G_1 with $\mu = \{g_2, h_1\}$;*
2. *subgroup hiding holds in H_1 with $\mu = \{g_1\}$;*
3. *extended parameter hiding holds with respect to $\mathcal{F} = R \cup f$ and* aux $= \{h_1^{\sigma(\cdot)}\}_{\forall \sigma \in S \cup T}$ *for all $h_1 \in H_1$;*
4. *$N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$; and*
5. *the polynomials in $R \cup f$ are linearly independent and have maximum degree $poly(\lambda)$.*

Proof. By requirements (1)-(3), Theorem 3.3 tells us that the original assumption is implied by the computational variant of Assumption 3.2. We make the problem strictly easier if we assume that g_1 and $\vec{x}$ are public, in which case $g_1^{R(\vec{x})}$, $h_1^{S(\vec{x})}$, and $e(g_1, h_1)^{T(\vec{x})}$ provide no additional information, and $\mathcal{A}$ can compute the $G_{1,T}$ component of chal directly.

We thus consider a problem where $\mathcal{A}$ is given $g_2^{\sum r_i}$ and $\{g_2^{\sum_{i=1}^{q+2} r_i \rho_k(\vec{x}_i)}\}_{k=0}^{r}$ and we must argue that it is hard for it to compute $e(g_2, \hat{h})^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$. If we let $\ell = q+2$, requirements (4)-(5) and Lemma 3.4 imply that V is invertible with all but negligible probability, and Lemma 3.5 then tells us that the distribution over $\vec{y} \leftarrow \vec{r} \cdot V$ is uniformly random. As $\mathcal{A}$ is given values in G_2 raised to the first $q+1$ entries of $\vec{y}$ and is asked to compute $e(g_2, \hat{h})$ raised to the last, it is thus given uniformly random values and asked to compute something uniformly random, which it has at most negligible probability in doing. □

3.2.2 Decisional assumptions.

Finally, to enable an argument about the decisional assumption in the target, we introduce the following assumption:

Assumption 3.7. *For $\ell \in \mathbb{N}$ and a bilinear group $\mathbb{G} = (N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$, consider the* inputs *given to $\mathcal{A}$ in Assumption 3.2. Given the same set of* inputs, *it is difficult to distinguish $e(g_1, \hat{h})^{f(\vec{x})}$ $e(g_2, \hat{h})^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)}$ from $e(g_1, \hat{h})^{f(\vec{x})} \cdot R$ for $R \xleftarrow{\$} G_{2,T}$.*

We now prove the following lemma:

```
$\underline{\text{MAIN } \mathsf{G}^{\mathcal{A}}_{3.2}(\lambda) \ / \ \mathsf{G}^{\mathcal{A}}_{0}(\lambda) \ / \ \mathsf{G}^{\mathcal{A}}_{1}(\lambda)}$
if $(b = 0)$ then $\mathsf{chal} \xleftarrow{\$} G_T$                                                          // $\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)$
if $(b = 0)$ then $R \xleftarrow{\$} G_T$; $\mathsf{chal} \leftarrow \boxed{e(g_1, \hat{h})^{f(\vec{x})}} \cdot R$           // $\mathsf{G}^{\mathcal{A}}_{0}(\lambda)$
if $(b = 0)$ then $\boxed{R \xleftarrow{\$} G_{2,T}}$; $\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} \cdot R$          // $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$
```

Figure 1: Games for the proof of Lemma 3.8. Each game introduces the boxed code on its corresponding line.

Lemma 3.8. *If subgroup hiding holds in $G_{2,T}$ with $\mu = \{g_1, g_2, h_1\}$, then Assumption 3.2 is implied by Assumption 3.7.*

Proof. Let $\mathcal{A}$ be a PT adversary playing game $\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)$, and let $\mathbf{Adv}^{3.7}_{\mathcal{A}}(\lambda)$ denote its advantage in the game specified in Assumption 3.7. We build a PT adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{3.7}_{\mathcal{A}}(\lambda)$$

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{B}$ such that

$$\Pr[\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{0}(\lambda)] = 0 \tag{2}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{0}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)] \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) \tag{3}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)] = \mathbf{Adv}^{3.7}_{\mathcal{A}}(\lambda). \tag{4}$$

We then have that

$$\begin{aligned}\mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda) &= \Pr[\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)] \\ &= (\Pr[\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{0}(\lambda)]) + (\Pr[\mathsf{G}^{\mathcal{A}}_{0}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)]) + \Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)] \\ &\leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{3.7}_{\mathcal{A}}(\lambda).\end{aligned}$$

Equation 2: $\mathsf{G}^{\mathcal{A}}_{3.2}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{0}(\lambda)$

This follows trivially, as the values $\mathsf{chal} \cdot A$ and chal are identically distributed for $\mathsf{chal} \xleftarrow{\$} G_T$ and $A \in G_{1,T}$.

Equation 3: $\mathsf{G}^{\mathcal{A}}_{0}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$

$\mathcal{B}$ behaves as follows:

```
$\underline{\mathcal{B}(1^\lambda, N, G, H, G_T, e, g_1, g_2, h_1, w)}$
$b \xleftarrow{\$} \{0,1\}$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_\ell \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c$, $r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{\ell} r_j \rho_k(\vec{x}_j)}$ $\forall k \in [r]$          (Here we define $\rho_0 = 1$.)
$y_k \leftarrow h_1^{\sigma_k(\vec{x})}$ $\forall k \in [s]$
$z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}$ $\forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
if $(b = 0)$ then $\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} \cdot w$
if $(b = 1)$ then $\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{j=1}^{\ell} r_j f(\vec{x}_j)}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{inputs}, \mathsf{chal})$
return $(b' = b)$
```

If $w \xleftarrow{\$} G_T$, then this is identical to $\mathsf{G}^{\mathcal{A}}_{0}(\lambda)$. If $w \xleftarrow{\$} G_{2,T}$, then this is identical to $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$. □

Proposition 3.9. *For a bilinear group $\mathbb{G}$ of order N, the decisional uber-assumption parameterized by (c, R, S, T, f) holds in the target group if*

1. *subgroup hiding holds in G_1 with $\mu = \{g_2, h_1\}$;*

Scheme	Assumption	Elements in public key	
		symmetric	asymmetric
BGW [13]	q-BDHE	$2q + A$	$4q + A$
BBG [12]	q-wBDHI	$q + 4$	$2q + 7$
Waters [41]	q-BDHE	$3 + U$	$5 + 2U$
ACF [1]	q-wBDHI	$2 + 2q$	$3 + 2q$

Table 1: Examples of schemes whose reductions are compatible with the desired conversion from symmetric to asymmetric assumptions, along with the assumptions they rely on and the numbers of group elements in both the symmetric and asymmetric variants of the public key. The value A refers to the number of parallel instances of the system being run in the BGW scheme, and the value U refers to the maximum number of system attributes in Waters' scheme.

2. *subgroup hiding holds in H_1 with $\mu = \{g_1\}$;*
3. *subgroup hiding holds in $G_{2,T}$ with $\mu = \{g_1, g_2, h_1\}$;*
4. *extended parameter hiding holds with respect to $\mathcal{F} = R \cup f$ and* aux $= \{h_1^{\sigma(\cdot)}\}_{\forall \sigma \in S \cup T}$ *for all $h_1 \in H_1$;*
5. *$N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$; and*
6. *the polynomials in $R \cup f$ are linearly independent and have maximum degree $poly(\lambda)$.*

Proof. By requirements (1)-(4), Theorem 3.3 and Lemma 3.8 tell us that the original assumption is implied by Assumption 3.7. We make the problem strictly easier if we assume that g_1 and $\vec{x}$ is public, in which case $g_1^{R(\vec{x})}$, $h_1^{S(\vec{x})}$, and $e(g_1, h_1)^{T(\vec{x})}$ provide no additional information, and $\mathcal{A}$ can compute the $G_{1,T}$ component of chal directly (which is the same in either case).

We thus consider a problem where $\mathcal{A}$ is given $g_2^{\sum r_i}$ and $\{g_2^{\sum_{i=1}^{q+2} r_i \rho_k(\vec{x}_i)}\}_{k=0}^{r}$ and we must argue that it is hard for it to distinguish $e(g_2, \hat{h})^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$ from random. If we let $\ell = q + 2$, requirements (5)-(6) and Lemmas 3.4 and 3.5 imply that the distribution over $\vec{y} \leftarrow \vec{r} \cdot V$ is uniformly random with all but negligible probability. As $\mathcal{A}$ is given values in G_2 raised to the first $q + 1$ entries of $\vec{y}$ and is asked to distinguish $e(g_2, \hat{h})$ raised to the last from random, it is thus given uniformly random values and asked to distinguish two uniformly random things, which it has at most negligible advantage in doing. □

3.3 Converting symmetric uber-assumptions

As mentioned earlier, most schemes that rely on q-type assumptions do so in the symmetric setting, whereas our analysis above works only in the asymmetric setting. To nevertheless capture these useful examples of q-type assumptions, we use the technique of Abe et al. [4] to convert the assumptions from the symmetric to the asymmetric setting so that they can be covered by our analysis.

To perform this conversion, we must of course do so in a way that respects the underlying reduction; i.e., we must ensure that the asymmetric variant of the scheme can still be proved secure under the asymmetric variant of the assumption. The main technique for doing this revolves around the idea of *dependency graphs* that reflect the usage of all values in the source groups and how they interact with each other and with the pairing. Thus, all of the dependencies in both the scheme and its security reduction are represented in a directed graph Γ, with pairings represented by two nodes (one for each side of the pairing). To find an asymmetric variant that respects these dependencies, one must search for a *valid split* of Γ into Γ_0 and Γ_1; this is defined as a split in which

- No nodes or edges are lost; i.e., merging Γ_0 and Γ_1 recovers Γ,
- For every pair of pairing nodes, if one node is in Γ_0, the other node is exclusively in Γ_1, and
- For every node X in each split graph, the ancestor subgraph of X in Γ is included in the same graph.

For more details on this technique and the process of automating it, we refer to the original paper of Abe et al. or to a paper by Akinyele et al. [5] that proposes a tool, AutoGroup+, that improves on the tool developed by Abe et al. and applies the technique to additional schemes.

To demonstrate the coverage of our analysis, we have identified four influential schemes that rely on symmetric uber-assumptions and demonstrated their conversion to asymmetric variants that fit into the class of uber-assumptions our analysis can cover. These are:

- The general construction of the Boneh-Gentry-Waters broadcast encryption scheme [13], based on the q-BDHE assumption;

- the Boneh-Boyen-Goh hierarchical identity-based encryption scheme with constant-sized ciphertexts [12], based on the q-wBDHI assumption;
- the version of Waters' attribute-based encryption scheme [41] that uses the q-BDHE assumption (as opposed to the more efficient construction that uses the q-parallel BDHE assumption [43], which we cannot cover); and
- the Abdalla-Catalano-Fiore identity-based key encapsulation mechanism [1], based on the q-wBDHI assumption.

These schemes are given in Table 1, along with the assumptions they rely on for security, and the number of elements in both the symmetric and the asymmetric variants of the public key. As an example of our analysis, we include in Figure 2 the dependency graph for the Boneh-Boyen-Goh HIBE. In the graph, the shape of the node indicates which side of the split each element goes on: triangle nodes are in G, inverted triangle nodes are in H, and diamond nodes are replicated across G and H. Pairing equations are denoted by $pn[i]$, where $n \in \mathbb{N}$ indicates a particular usages of the pairing and $i \in \{0,1\}$ indicates the side of the pairing in which the element is used. The nodes with an i included represent multiple (related) values; e.g., the node yi represents $\{g^{\alpha^i}\}_i$.

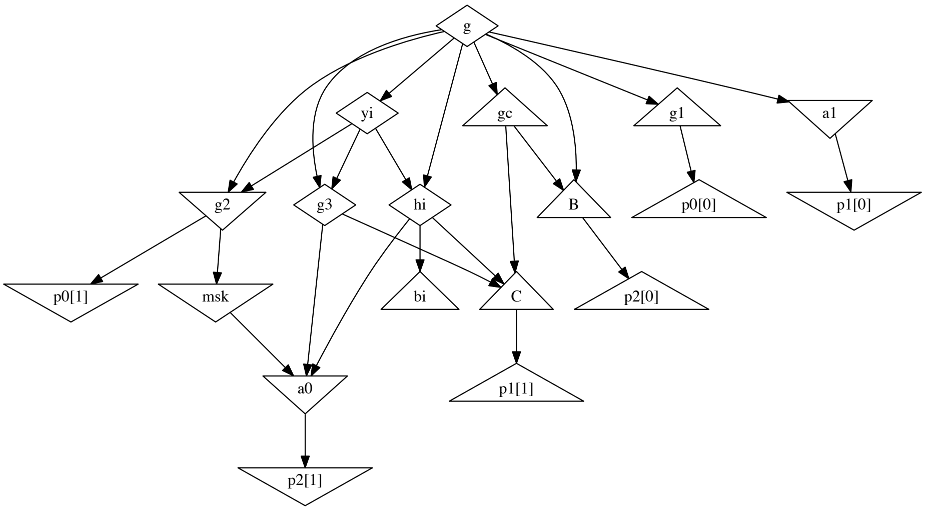

Figure 2: Dependency graph for the BBG HIBE scheme [12]. The public key consists of g, $g1$, $g2$, $g3$, and hi, the master secret key is denoted msk, and the secret keys consist of $a0$, $a1$, and bi. Encryption uses the pairing $p0$ and produces B and C, and decryption uses the pairings $p1$ and $p2$. In the reduction, yi and gc are derived from the q-wBDHI assumption.

The original q-wBDHI assumption states that given $(g, g^c, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^q})$, it should be hard to distinguish $e(g,g)^{\alpha^{q+1}c}$ from random. Looking at the graph in Figure 2, in which these quantities are represented by yi and gc, we see that the yi nodes must be replicated across G and H but gc can remain in only one source group. Writing h^c as $\hat{h}$, the asymmetric q-wBDHI assumption thus states that given $(g, h, \hat{h}, g^\alpha, h^\alpha, \dots, g^{\alpha^q}, h^{\alpha^q})$, it should be hard to distinguish $e(g, \hat{h})^{\alpha^{q+1}}$ from random. This same converted version of the assumption also works for the Abdalla-Catalano-Fiore IB-KEM (whose dependency graph is included in Appendix C).

A similar analysis works for the schemes that rely on the q-BDHE assumption (whose dependency graphs are also included in Appendix C), which states that given $(g, g^c, \{g^{\alpha^i}\}_{i\in[2q], i\neq q+1})$, it should be hard to distinguish $e(g,g)^{\alpha^{q+1}c}$ from random. Here we find that the asymmetric variant states that — again, rewriting h^c as $\hat{h}$ — given $(g, h, \hat{h}, \{g^{\alpha^i}, h^{\alpha^i}\}_{i\in[2q], i\neq q+1}$, it should be hard to distinguish $e(g, \hat{h})^{\alpha^{q+1}}$ from random.

As each of the converted assumptions fits the set of requirements for the uber-assumption needed for Proposition 3.9, we thus obtain as a corollary that, when instantiated in asymmetric composite-order bilinear groups, the security of each of these schemes can rely solely on (three variants of) the subgroup hiding assumption.

4 Tighter Reductions in (A)symmetric Groups

The results in the previous section already demonstrate a broader application of the Déjà Q framework, but two fundamental restrictions remain: it can be applied directly to assumptions only in asymmetric composite-order bilinear groups, and it introduces a looseness of q into the reduction. In this section, we address both of these restrictions. In particular, we show that by adding more primes into the factorization of N, we can achieve a tighter reduction — one with $\log(q)$ looseness instead of q — in symmetric composite-order bilinear groups.

Our inspiration for the conversion to symmetric groups comes from Wee [44], who applied the Déjà Q framework at the level of constructions rather than assumptions, and thus was able to make use of two key features of traditional dual-system reductions: fresh randomness across queries and a third subgroup used to hide additional information. To maintain the most generality, we continue in Section 4.1 to work at the level of assumptions, but we nevertheless attempt to capture these additional features by using a variant of the uber-assumption in which extra randomness is added into components in G. We then define an assumption with significant randomness added into various subgroups in G (analogous to Assumption 3.2). Finally, we diverge completely from [44] and prove that — in only a logarithmic number of game hops — this assumption implies these additionally randomized computational and decisional uber-assumptions in the target group.

Next, in Section 4.2, we show — in a manner almost completely analogous to that in Section 3.2 — that the computational variant of the transitioned uber-assumption is so weak that it holds by a statistical argument; thus the computational randomized uber-assumption is implied by two variants of subgroup hiding. In the case of the decisional uber-assumption, we transition to an assumption analogous to Assumption 3.7 and show that it is implied by three variants of subgroup hiding.

Finally, in Section 4.3, we briefly discuss the implications of our results for the concrete schemes presented in Section 3.3. Although our discussion here is not as formal as our symmetric-to-asymmetric conversions, we nevertheless suggest ways to transform existing schemes to provide them with tight reductions to subgroup hiding.

4.1 Reducing randomized assumptions to weaker variants

We begin by formalizing the *randomized* uber-assumption as follows:

Assumption 4.1 (Randomized uber-assumption)**.** *For an adversary $\mathcal{A}$, we define its computational advantage to be $\mathbf{Adv}_{\mathcal{A}}^{comp\text{-}r\text{-}uber}(\lambda) = Pr[comp\text{-}RandUBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)]$ and its decisional advantage to be $\mathbf{Adv}_{\mathcal{A}}^{dec\text{-}r\text{-}uber}(\lambda) = 2Pr[dec\text{-}RandUBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)] - 1$, where for $\mathsf{type} \in \{comp, dec\}$, $\mathsf{type}\text{-}RandUBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$ is defined as follows (with the omitted end games comp-PLAY and dec-PLAY the same as in Definition 3.1):*

```
MAIN $\mathsf{type}\text{-}RandUBER_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 4)$; $g \xleftarrow{\$} G$, $g_4 \xleftarrow{\$} G_4$, $h_{123}, \hat{h} \xleftarrow{\$} H_{123}$
$x_1, \dots, x_c, \chi_1, \dots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g, g_4, \hat{h}, g^{R(\vec{x})} g_4^{\vec{\chi}}, h_{123}^{S(\vec{x})}, e(g, h_{123})^{T(\vec{x})})$
$\mathsf{chal} \leftarrow e(g, \hat{h})^{f(\vec{x})}$
return $\mathsf{type}\text{-PLAY}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

The the randomized uber-assumption in the target group holds if for PT algorithms $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $\mathbf{Adv}_{\mathcal{A}}^{r\text{-}uber}(\lambda) < \nu(\lambda)$.

The main difference from the regular uber-assumption is the additional randomness in G_4 (hence the name), and the fact that h and $\hat{h}$ are now sampled from the subgroup H_{123} rather than the full group H. As discussed further in Section 4.3, this latter change is needed to balance out the former, as the canceling property of the pairing means that we can still obtain meaningful values in G_T (i.e., values without added randomness) by pairing an element with a random G_4 component with an element in H_{123}. To maintain full generality, we also continue to write G and H separately, but in a symmetric pairing they would be the same group.

Assumption 4.2. *For a bilinear group $\mathbb{G} = (N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 4)]$, $\ell \in \mathbb{N}$, and classes of functions R, S, T, and f (as defined in the uber-assumption in Assumption 4.1), given*

$$\mathsf{inputs} = (\mathbb{G}, g_1 g_2^{\sum_{i=1}^{\ell} r_i} g_4^{\chi}, g_4, \hat{h}, \{g_1^{\rho_k(\vec{x})} g_2^{\sum_{i=1}^{\ell} r_i \rho_k(\vec{x}_i)} g_4^{\chi_k}\}_{k=1}^{r}, h_1^{S(\vec{x})}, e(g_1, h_1)^{T(\vec{x})}),$$

for $g_1 \xleftarrow{\$} G_1$, $g_2 \xleftarrow{\$} G_2\backslash\{1\}$, $g_4 \xleftarrow{\$} G_4$, $h_1 \xleftarrow{\$} H_1\backslash\{1\}$, $\hat{h} \xleftarrow{\$} H_{123}$*;* $\vec{x}, \dots, \vec{x}_\ell \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c$, $r_1, \dots, r_\ell, \chi, \chi_1, \dots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$*, there is no PT adversary that has better than negligible advantage when playing* $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)})$*.*

In addition to the extra subgroups, our new reduction also makes use of a different class of functions for extended parameter hiding. In particular, our old proof added variables into G_2 one at a time, which allowed us to fold in a freshly random coefficient r_j in this step. As we now add many variables at a time, however, the extra randomness added by the subgroup hiding transition is not sufficient, so we instead use parameter hiding to argue that the randomness can be "freshened up" in the new subgroup instead. In the main parameter hiding step, we thus want to transition the quantity $\sum_j r_j \rho_k(\vec{x}_j)$ to $\sum_j r'_j \rho_k(\vec{x}'_j)$, which we accomplish using the set of functions defined as

$$\mathcal{F} = \left\{ p'(y_1, \vec{y}_1, \dots, y_m, \vec{y}_m) = \sum_{i=1}^{m} y_m p(\vec{y}_m) \right\}_{p \in R \cup \{f\}}. \tag{5}$$

Theorem 4.3. *For a bilinear group* $(N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 4)]$*, consider the randomized uber-assumption parameterized by* (c, R, S, T, f)*. Then this is implied by Assumption 4.2 if*

1. *subgroup hiding holds between* H_1 *and* H_{123} *with* $\mu = \{g_4, h_{123}\}$*;*
2. *subgroup hiding holds between* G_{24} *and* G_{34} *with* $\mu = \{g_1, g_{24}, g_4, h_1, h_{123}\}$*;*
3. *extended parameter hiding holds with respect to* $R \cup \{f\}$*,* $\mathsf{aux} = \{g_3^{\rho(\cdot)}, h_1^{\sigma(\cdot)}\}_{\rho \in R \cup \{f\}, \sigma \in S \cup T}$ *for all* $g_3 \in G_3$ *and* $h_1 \in H_1$*, and subgroups* (G_1, G_2)*;*
4. *extended parameter hiding holds with respect to* $R \cup \{f\}$*,* $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ *for all* $h_1 \in H_1$*, and subgroups* (G_1, G_3)*; and*
5. *extended parameter hiding holds with respect to the* $\mathcal{F}$ *defined in Equation 5,* $\mathsf{aux} = \emptyset$*, and subgroups* (G_2, G_3)*.*

In particular, we have that

$$\mathbf{Adv}_{\mathcal{A}}^{r\text{-}uber}(\lambda) \leq \mathbf{Adv}_{\mathcal{C}_0}^{sgh}(\lambda) + \mathbf{Adv}_{\mathcal{C}_1}^{sgh}(\lambda) + \log_2(\ell)(\mathbf{Adv}_{\mathcal{B}_i}^{sgh}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i+1}}^{sgh}(\lambda)) + \mathbf{Adv}_{\mathcal{A}}^{4.2}(\lambda).$$

Our two subgroup hiding variants are valid instantiations of the general subgroup decision assumption [9] discussed in Section 2. Similarly, we proved in Lemma 2.3 that in composite-order groups extended parameter hiding holds for all polynomials and the aux and subgroups that we use here, so the three variants all hold and are listed separately solely for insight into the reduction.

A proof of this theorem can be found in Appendix B. To start, all elements using h as the base are shifted into the H_1 subgroup, but elements using $\hat{h}$ or in G remain unchanged. Using the first two variants of parameter hiding, we now switch the variables in G_2 to $\vec{x}'$ and in G_3 to $\vec{x}''$, and — using subgroup hiding — fold the $\vec{x}''$ elements into G_2. At this point we now have the original variables $\vec{x}$ in G_1, two new sets of variables in G_2, nothing in G_3, and random values in G_4.

Our reduction now proceeds by exploiting this "semi-functional" subgroup G_3 and the masking effect provided by the randomness in G_4. First, a shadow copy of *all of the variables* in G_2 is added to G_3, which we argue goes unnoticed by subgroup hiding. Second, the variables in G_3 are changed to a new set of variables, which is identical by the third variant of parameter hiding. Finally, we fold all of the new variables back into G_2, which we again argue goes unnoticed by subgroup hiding. By working with all of the variables at once — as opposed to the one-at-a-time approach of the original Déjà Q framework — we double the number of new variables in the G_2 subgroup after each iteration, so after only $\log_2(\ell)$ transitions we end up with ℓ sets of variables in the G_2 subgroup.

As described, we move new variables from G_3 to G_2 while using the generator g_2 to compute the existing variables in the G_2 subgroup. In symmetric groups with a canceling pairing, however, one could use knowledge of this generator to violate subgroup hiding by checking if $e(g_2, w) = 1$. The G_4 subgroup is thus needed to mask this transition, so in symmetric groups we transition from G_{34} to G_{24} instead, and argue that the randomness in G_4 "absorbs" the variables that are added there. In an asymmetric setting, however, knowledge of g_2 does not provide the ability to distinguish G_2 and G_3, so the masking effect of G_4 is unnecessary and the same reduction goes through without it. We thus state the simplified version of Theorem 4.3 for asymmetric groups as the following corollary:

Corollary 4.4. *For an asymmetric bilinear group* $(N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 3)]$*, consider the uber-assumption parameterized by* (c, R, S, T, f)*. Then this is implied by a version of Assumption 3.2 (using* $\mathsf{BilinearGen}(1^\lambda, 3)$*) if*

1. subgroup hiding holds between H and H_1 with $\mu = \{\ \}$;
2. subgroup hiding holds between G_2 and G_3 with $\mu = \{g_1, g_2, h_1\}$;
3. extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{g_3^{\rho(\cdot)}, h_1^{\sigma(\cdot)}\}_{\rho \in R \cup \{f\}, \sigma \in S \cup T}$ for all $g_3 \in G_3$ and $h_1 \in H_1$, and subgroups (G_1, G_2);
4. extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ for all $h_1 \in H_1$, and subgroups (G_1, G_3); and
5. extended parameter hiding holds with respect to the $\mathcal{F}$ defined in Equation 5, $\mathsf{aux} = \emptyset$, and subgroups (G_2, G_3).

In particular, we have that

$$\mathbf{Adv}_{\mathcal{A}}^{uber}(\lambda) \leq \mathbf{Adv}_{\mathcal{C}_0}^{sgh}(\lambda) + \mathbf{Adv}_{\mathcal{C}_1}^{sgh}(\lambda) + \log_2(\ell)(\mathbf{Adv}_{\mathcal{B}_i}^{sgh}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i+1}}^{sgh}(\lambda)) + \mathbf{Adv}_{\mathcal{A}}^{3.2}(\lambda).$$

Thus, under the conditions in Propositions 3.6 and 3.9, we get tight reductions in the asymmetric setting with $N = p_1p_2p_3$.

For the rest of this section we will focus on the symmetric setting.

4.2 Reducing randomized assumptions to subgroup hiding

As in Section 3, we now treat computational and decisional assumptions separately.

4.2.1 Computational assumptions.

Our argument that the computational randomized uber-assumption holds is nearly identical to our previous argument that the (regular) computational uber-assumption holds.

Proposition 4.5. *For a bilinear group $\mathbb{G}$ of order N, the computational uber-assumption parameterized by (c, R, S, T, f) holds in the target group if*

1. subgroup hiding holds between H_1 and H_{123} with $\mu = \{g_4, h_{123}\}$;
2. subgroup hiding holds between G_{34} and G_{24} with $\mu = \{g_1, g_{24}, g_4, h_1, h_{123}\}$;
3. extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{g_3^{\rho(\cdot)}, h_1^{\sigma(\cdot)}\}_{\rho \in R \cup \{f\}, \sigma \in S \cup T}$ for all $g_3 \in G_3$ and $h_1 \in H_1$, and subgroups (G_1, G_2);
4. extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ for all $h_1 \in H_1$, and subgroups (G_1, G_3);
5. extended parameter hiding holds with respect to the $\mathcal{F}$ defined in Equation 5, $\mathsf{aux} = \emptyset$, and subgroups (G_2, G_3);
6. $N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$; and
7. the polynomials in $R \cup f$ are linearly independent and have maximum degree $poly(\lambda)$.

Proof. By requirements (1)-(5), Theorem 4.3 tells us that the computational uber-assumption is implied by the computational variant of Assumption 4.2. We make the problem strictly easier if we assume that g_1, g_4, $\vec{x}$, and $\vec{\chi}$ are public, in which case $g_1^{R(\vec{x})}$, $g_4^{\vec{\chi}}$, $h_1^{S(\vec{x})}$ and $e(g_1, h_1)^{T(\vec{x})}$ provide no additional information. In this case $\mathcal{A}$ can also compute the $G_{1,T}$ component of chal directly, so we need only to argue that it is hard for it to compute $e(g_2, \hat{h})^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$. The rest of the argument can thus proceed as in the proof of Proposition 3.6. □

4.2.2 Decisional assumptions.

To enable an argument about the decisional assumption in the target group, we introduce an assumption analogous to Assumption 3.7.

Assumption 4.6. *For a bilinear group $(N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 4)]$, $\ell \in \mathbb{N}$, consider the values given to $\mathcal{A}$ in Assumption 4.2. Given the same set of values, it is difficult to distinguish $e(g_1, \hat{h}_1)^{f(\vec{x})}$ $e(g_2, \hat{h}_2)^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)}$ from $e(g_1, \hat{h}_1)^{f(\vec{x})} \cdot R$ for $R \xleftarrow{\$} G_{2,T}$.*

We now prove the following lemma:

Lemma 4.7. *If subgroup hiding holds in $G_{2,T}$ with $\mu = \{g_1, g_2, g_4, h_1, h_{123}\}$, then Assumption 4.2 is implied by Assumption 4.6.*

Proof. Let $\mathcal{A}$ be a PT adversary playing game $\mathsf{G}^{\mathcal{A}}_{4.2}(\lambda)$, and let $\mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda)$ denote its advantage in the game specified in Assumption 4.2. We build a PT adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{4.6}_{\mathcal{A}}(\lambda)$$

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{B}$ such that

$$\Pr[\mathsf{G}^{\mathcal{A}}_{4.2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{4.6}(\lambda)] \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) \tag{6}$$

We then have that

$$\mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda).$$

Equation 6: $\mathsf{G}^{\mathcal{A}}_{4.2}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{4.6}(\lambda)$
$\mathcal{B}$ behaves as follows (again assuming $\rho_0 = 1$):

```
$\underline{\mathcal{B}(1^\lambda, N, G, H, G_T, e, g_1, g_2, g_4, h_1, w)}$
$b \xleftarrow{\$} \{0,1\}$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_\ell \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c$, $r_1, \ldots, r_\ell, \chi_1, \ldots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{\ell} r_j \rho_k(\vec{x}_j)} g_4^{\chi_k}\ \forall k \in [r]$
$y_k \leftarrow h_1^{\sigma_k(\vec{x})}\ \forall k \in [s]$
$z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}\ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
if $(b = 0)$ then $\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} \cdot w$
if $(b = 1)$ then $\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{j=1}^{\ell} r_j f(\vec{x}_j)}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{inputs}, \mathsf{chal})$
return $(b' = b)$
```

If $w \xleftarrow{\$} G_T$, then this is identical to $\mathsf{G}^{\mathcal{A}}_{4.2}(\lambda)$. If $w \xleftarrow{\$} G_{2,T}$, then this is identical to $\mathsf{G}^{\mathcal{A}}_{4.6}(\lambda)$. □

Proposition 4.8. *For a bilinear group $\mathbb{G}$ of order N, the decisional uber-assumption parameterized by (c, R, S, T, f) holds in the target group if*

1. *subgroup hiding holds between H_{123} and H_1 with $\mu = \{g_4, h_{123}\}$;*
2. *subgroup hiding holds between G_{24} and G_{34} with $\mu = \{g_1, g_{24}, g_4, h_1, h_{123}\}$;*
3. *subgroup hiding holds in $G_{2,T}$ with $\mu = \{g_1, g_2, g_4, h_1, h_{123}\}$;*
4. *extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{g_3^{\rho(\cdot)}, h_1^{\sigma(\cdot)}\}_{\rho \in R \cup \{f\}, \sigma \in S \cup T}$ for all $g_3 \in G_3$ and $h_1 \in H_1$, and subgroups (G_1, G_2);*
5. *extended parameter hiding holds with respect to $R \cup \{f\}$, $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ for all $h_1 \in H_1$, and subgroups (G_1, G_3);*
6. *extended parameter hiding holds with respect to the $\mathcal{F}$ defined in Equation 5, $\mathsf{aux} = \emptyset$, and subgroups (G_2, G_3);*
7. *$N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$; and*
8. *the polynomials in $R \cup f$ are linearly independent and have maximum degree $poly(\lambda)$.*

Proof. By requirements (1)-(6), Theorem 4.3 and Lemma 4.7 tell us that the original assumption is implied by Assumption 4.2. We make the problem strictly easier if we assume that g_1, g_4, $\vec{x}$, and $\vec{\chi}$ are public, in which case $g_1^{R(\vec{x})}$, $g_4^{\vec{\chi}}$, $h_1^{S(\vec{x})}$ and $e(g_1, h_1)^{T(\vec{x})}$ provide no additional information. In this case $\mathcal{A}$ can also compute the $G_{1,T}$ component of chal directly (which is the same in either case), so we need only to argue that it is hard for it to distinguish $e(g_2, \hat{h})^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$ from random. The rest of the argument can thus proceed as in the proof of Proposition 3.9. □

4.3 Application to existing schemes

In Section 3.3, we demonstrated how to convert schemes that rely on symmetric version of the uber-assumption to work in asymmetric groups and thus be covered by our overall results in Section 3. Here, we briefly demonstrate how to convert schemes to be covered by our results in this section as well.

Suppose we have a scheme and corresponding reduction that work in asymmetric groups and performs only group operations, pairings, and equality tests between group elements. We can then modify both the scheme and reduction as follows: instead of sampling elements from H we sample them from H_{123}; when we multiply any elements in G we also include a freshly random element in G_4; and when we compare two elements g and g' in G for equality, rather than return $(g = g')$ we return $(e(g, h_{123}) = e(g', h_{123}))$. In particular, this last alteration — combined with the fact that $e(g_4, h_{123}) = 1$ and an asymmetric scheme only ever pairs elements of G with elements of H — allows us to preserve the functionality of the original scheme despite the fact that additional randomness is added into the G_4 subgroup.

If the original assumption relied on for security is a case of the uber-assumption (Definition 3.1), then the resulting assumption is a case of the randomized uber-assumption (Definition 4.1). Thus, the concrete schemes presented in Section 3.3 can be instantiated either in asymmetric groups of order $N = p_1p_2p_3$ under the asymmetric variants of their original (symmetric) assumptions, or in symmetric groups of order $N = p_1p_2p_3p_4$ under the randomized variants. In either case, the results of Theorem 4.3 and Corollary 4.4 imply a tight reduction to the appropriate variants of the subgroup hiding assumption.

Acknowledgments

Mary Maller is supported by a scholarship from Microsoft Research and Sarah Meiklejohn is supported in part by EPSRC Grant EP/M029026/1.

References

[1] M. Abdalla, D. Catalano, and D. Fiore. Verifiable random functions from identity-based key encapsulation. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 554–571, Cologne, Germany, Apr. 26–30, 2009. Springer, Heidelberg, Germany.

[2] M. Abdalla, D. Catalano, and D. Fiore. Verifiable random functions: Relations to identity-based key encapsulation and new constructions. *Journal of Cryptology*, 27(3):544–593, July 2014.

[3] M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 312–331, Nara, Japan, Feb. 26 – Mar. 1, 2013. Springer, Heidelberg, Germany.

[4] M. Abe, J. Groth, M. Ohkubo, and T. Tango. Converting cryptographic schemes from symmetric to asymmetric bilinear groups. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 241–260, Santa Barbara, CA, USA, Aug. 17–21, 2014. Springer, Heidelberg, Germany.

[5] J. A. Akinyele, C. Garman, and S. Hohenberger. Automating fast and secure translations from type-I to type-III pairing schemes. In I. Ray, N. Li, and C. Kruegel:, editors, *ACM CCS 15*, pages 1370–1381, Denver, CO, USA, Oct. 12–16, 2015. ACM Press.

[6] N. Attrapadung, G. Hanaoka, and S. Yamada. A framework for identity-based encryption with almost tight security. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, pages 521–549, 2015.

[7] C. Bader, D. Hofheinz, T. Jager, E. Kiltz, and Y. Li. Tightly-secure authenticated key exchange. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 629–658, Warsaw, Poland, Mar. 23–25, 2015. Springer, Heidelberg, Germany.

[8] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Heidelberg, Germany.

[9] M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 235–252, Providence, RI, USA, Mar. 28–30, 2011. Springer, Heidelberg, Germany.

[10] O. Blazy, E. Kiltz, and J. Pan. (Hierarchical) identity-based encryption from affine message authentication. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425, Santa Barbara, CA, USA, Aug. 17–21, 2014. Springer, Heidelberg, Germany.

[11] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[12] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[13] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275, Santa Barbara, CA, USA, Aug. 14–18, 2005. Springer, Heidelberg, Germany.

[14] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 325–341, Cambridge, MA, USA, Feb. 10–12, 2005. Springer, Heidelberg, Germany.

[15] D. Boneh, K. Rubin, and A. Silverberg. Finding ordinary composite order elliptic curves using the Cocks-Pinch method. *Journal of Number Theory*, 131(5):832–841, 2011.

[16] X. Boyen. The uber-assumption family (invited talk). In S. D. Galbraith and K. G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56, Egham, UK, Sept. 1–3, 2008. Springer, Heidelberg, Germany.

[17] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307, Santa Barbara, CA, USA, Aug. 20–24, 2006. Springer, Heidelberg, Germany.

[18] M. Chase and S. Meiklejohn. Déjà Q: Using dual systems to revisit q-type assumptions. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 622–639, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[19] M. Chase and S. Meiklejohn. Déjà Q: Using dual systems to revisit q-type assumptions. Cryptology ePrint Archive, Report 2014/570, 2014. `http://eprint.iacr.org/2014/570`.

[20] J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624, Sofia, Bulgaria, Apr. 26–30, 2015. Springer, Heidelberg, Germany.

[21] J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460, Santa Barbara, CA, USA, Aug. 18–22, 2013. Springer, Heidelberg, Germany.

[22] J. H. Cheon. Security analysis of the strong Diffie-Hellman problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[23] Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In S. Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 416–431, Les Diablerets, Switzerland, Jan. 23–26, 2005. Springer, Heidelberg, Germany.

[24] R. Gay, D. Hofheinz, E. Kiltz, and H. Wee. Tightly cca-secure encryption without pairings. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 1–27, 2016.

[25] M. Gerbush, A. B. Lewko, A. O'Neill, and B. Waters. Dual form signatures: An approach for proving security from static assumptions. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 25–42, Beijing, China, Dec. 2–6, 2012. Springer, Heidelberg, Germany.

[26] S. Goldwasser and Y. T. Kalai. Cryptographic assumptions: A position paper. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 505–522, Tel Aviv, Israel, Jan. 10–13, 2016. Springer, Heidelberg, Germany.

[27] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88, Seoul, South Korea, Dec. 4–8, 2011. Springer, Heidelberg, Germany.

[28] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607, Santa Barbara, CA, USA, Aug. 19–23, 2012. Springer, Heidelberg, Germany.

[29] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 318–335, Cambridge, UK, Apr. 15–19, 2012. Springer, Heidelberg, Germany.

[30] A. B. Lewko and S. Meiklejohn. A profitable sub-prime loan: Obtaining the advantages of composite order in prime-order bilinear groups. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 377–398, Gaithersburg, MD, USA, Mar. 30 – Apr. 1, 2015. Springer, Heidelberg, Germany.

[31] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[32] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479, Zurich, Switzerland, Feb. 9–11, 2010. Springer, Heidelberg, Germany.

[33] A. B. Lewko and B. Waters. Decentralizing attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[34] A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[35] B. Libert, M. Joye, M. Yung, and T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 1–21, Kaoshiung, Taiwan, R.O.C., Dec. 7–11, 2014. Springer, Heidelberg, Germany.

[36] B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707, Auckland, New Zealand, Nov. 30 – Dec. 3, 2015. Springer, Heidelberg, Germany.

[37] S. Meiklejohn and H. Shacham. New trapdoor projection maps for composite-order bilinear groups. Cryptology ePrint Archive, Report 2013/657, 2013. `http://eprint.iacr.org/2013/657`.

[38] T. Okamoto and K. Takashima. Efficient (hierarchical) inner-product encryption tightly reduced from the decisional linear assumption. *IEICE Transactions*, 96-A(1):42–52, 2013.

[39] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.

[40] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.

```
$\textsc{main}\ \mathsf{type}\text{-}\mathrm{UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)\ /\ \boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}\ /\ \boxed{\mathsf{G}^{\mathcal{A}}_1(\lambda)}$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$
$g \xleftarrow{\$} G,\ \boxed{g_1 \xleftarrow{\$} G_1,\ g_2 \xleftarrow{\$} G_2 \setminus \{1\}}$                              // $\boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}$
$h, \hat{h} \xleftarrow{\$} H,\ \boxed{h_1 \xleftarrow{\$} H_1,\ \hat{h} \xleftarrow{\$} H}$                                   // $\boxed{\mathsf{G}^{\mathcal{A}}_1(\lambda)}$
$x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g^{\rho_k(\vec{x})},\ \boxed{v_k \leftarrow g_1^{\rho_k(\vec{x})}}\ \forall k \in [r]$                        // $\boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}$
$y_k \leftarrow h^{\sigma_k(\vec{x})},\ \boxed{y_k \leftarrow h_1^{\sigma_k(\vec{x})}}\ \forall k \in [s]$                      // $\boxed{\mathsf{G}^{\mathcal{A}}_1(\lambda)}$
$z_k \leftarrow e(g, h)^{\tau_k(\vec{x})},\ \boxed{z_k \leftarrow e(g_1, h)^{\tau_k(\vec{x})}} \forall k \in [t]$             // $\boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}$
$z_k \leftarrow e(g_1, h)^{\tau_k(\vec{x})},\ \boxed{z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}}\ \forall k \in [t]$         // $\boxed{\mathsf{G}^{\mathcal{A}}_1(\lambda)}$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g, \hat{h})^{f(\vec{x})},\ \boxed{\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})}}$                      // $\boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

Figure 3: Games for the proof of Theorem 3.3 (Equations 7 and 8). Each game uses the boxed code on its corresponding line.

[41] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290, 2008. `http://eprint.iacr.org/2008/290`.

[42] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636, Santa Barbara, CA, USA, Aug. 16–20, 2009. Springer, Heidelberg, Germany.

[43] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70, Taormina, Italy, Mar. 6–9, 2011. Springer, Heidelberg, Germany.

[44] H. Wee. Déjà Q: Encore! Un petit IBE. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 237–258, Tel Aviv, Israel, Jan. 10–13, 2016. Springer, Heidelberg, Germany.

A A Proof of Theorem 3.3

Proof. Let $\mathcal{A}$ be a PT adversary playing game $\mathsf{type}\text{-}\mathrm{UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$, and let $\mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda)$ denote its advantage in Assumption 3.2. We provide PT adversaries $\mathcal{B}_0$ and $\mathcal{C}_0$ and a family of PT adversaries $\mathcal{B}_i$ such that

$$\mathbf{Adv}^{\mathrm{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}_0}(\lambda) + \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{C}_0}(\lambda) + \ell\mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}_i}(\lambda) + \mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda)$$

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{B}_0$, $\mathcal{C}_0$, and $\mathcal{B}_i$ for all i, $1 \leq i \leq \ell$, such that

$$\Pr[\mathsf{type}\text{-}\mathrm{UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda) - \Pr[\mathsf{G}^{\mathcal{A}}_0(\lambda)] \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}_0}(\lambda) \qquad (G_1;\ \mu = \{\ \}) \tag{7}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_0(\lambda) - \Pr[\mathsf{G}^{\mathcal{A}}_1(\lambda)] \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{C}_0}(\lambda) \qquad (H_1;\ \mu = \{g_1\}) \tag{8}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_i(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] \leq \mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{B}_i}(\lambda) \qquad (G_1;\ \mu = \{g_2, h_1\}) \tag{9}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)] = 0 \tag{10}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_\ell(\lambda)] = \mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda). \tag{11}$$

```
MAIN $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$ / $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_i, r_1, \ldots, r_i \boxed{, x_{i+1,1}, \ldots, x_{i+1,c}, r_{i+1}} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)} \; \forall k \in [r]$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{r_{i+1}\rho_k(\vec{x}) + \sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)}} \; \forall k \in [r]$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{i+1} r_j \rho_k(\vec{x}_j)} \; \forall k \in [r]$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, h)^{\sum_{j=1}^{i} r_j f(\vec{x}_j)}$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{r_{i+1} f(\vec{x}) + \sum_{j=1}^{i} r_j f(\vec{x}_j)}}$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{j=1}^{i+1} r_j f(\vec{x}_j)}$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
```

Figure 4: Games for the proof of Theorem 3.3 (Equations 9 and 10). Each game uses the boxed code on its corresponding line.

We then have that

$$\begin{aligned}
\mathbf{Adv}^{\text{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) &= \Pr[\text{type-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] \\
&= (\Pr[\text{type-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - \Pr[\mathsf{G}_0^{\mathcal{A}}(\lambda)]) + (\Pr[\mathsf{G}_0^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_1^{\mathcal{A}}(\lambda)]) \\
&\quad + \left(\sum_{i=1}^{\ell} ((\Pr[\mathsf{G}_i^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)]) + (\Pr[\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)]))\right) + \Pr[\mathsf{G}_\ell^{\mathcal{A}}(\lambda)] \\
&\leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_0}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_0}(\lambda) + \ell \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_i}(\lambda) + \mathbf{Adv}^{3.2}_{\mathcal{A}}(\lambda).
\end{aligned}$$

We also for simplicity let $\rho_0 = 1$, so $R = \langle \rho_0, \ldots, \rho_q \rangle$.

Equation 7: type-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ to $\mathsf{G}_0^{\mathcal{A}}(\lambda)$

$\mathcal{B}_0$ behaves as follows:

```
$\mathcal{B}_0(1^\lambda, N, G, H, G_T, e, w)$
$x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}; h, \hat{h} \xleftarrow{\$} H$
$v_k \leftarrow w^{\rho_k(\vec{x})} \; \forall k \in [r]$
$y_k \leftarrow h^{\sigma_k(\vec{x})} \; \forall k \in [s]$
$z_k \leftarrow e(w, h)^{\tau_k(\vec{x})} \; \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(w, \hat{h})^{f(\vec{x})}$
return type-PLAY($\lambda$, inputs, chal)
```

If $w \xleftarrow{\$} G$ then this is identical to type-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$. If instead $w \xleftarrow{\$} G_1$ then this is identical to $\mathsf{G}_0^{\mathcal{A}}(\lambda)$.

Equation 8: $\mathsf{G}_0^{\mathcal{A}}(\lambda)$ to $\mathsf{G}_1^{\mathcal{A}}(\lambda)$

$\mathcal{C}_0$ behaves as follows:

```
$\mathcal{C}_0(1^\lambda, N, G, H, G_T, e, g_1, w)$
$x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$; $\hat{h} \xleftarrow{\$} H$
$v_k \leftarrow g_1^{\rho_k(\vec{x})}\ \forall k \in [r]$
$y_k \leftarrow w^{\sigma_k(\vec{x})}\ \forall k \in [s]$
$z_k \leftarrow e(g_1, w)^{\tau_k(\vec{x})}\ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

If $w \xleftarrow{\$} H$ then this is identical to $\mathsf{G}_0^{\mathcal{A}}(\lambda)$. If instead $w \xleftarrow{\$} H_1$ then this is identical to $\mathsf{G}_1^{\mathcal{A}}(\lambda)$.

Equation 9: $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ to $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$

$\mathcal{B}_i$ behaves as follows:

```
$\mathcal{B}_i(1^\lambda, N, G, H, G_T, e, g_2, h_1, w)$
$x_1, \ldots, x_c, x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i+1,1}, \ldots, x_{i+1,c} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$; $\hat{h} \xleftarrow{\$} H$
$v_k \leftarrow w^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)}\ \forall k \in [r]$
$y_k \leftarrow h_1^{\sigma_k(\vec{x})}\ \forall k \in [s]$
$z_k \leftarrow e(w, h_1)^{\tau_k(\vec{x})}\ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(w, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{j=1}^{i} r_j f(\vec{x}_j)}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

If $w = g_1 \xleftarrow{\$} G_1$, then

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} g_2^{\sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)}, \\ z_k &\leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}, \text{ and} \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\sum_{j=1}^{i} r_j f(\vec{x}_j)}, \end{aligned}$$

which are identical to the values in $\mathsf{G}_i^{\mathcal{A}}(\lambda)$. If instead $w \xleftarrow{\$} G$, then $w = g_1 g_2^{r_{i+1}}$ for uniformly distributed $g_1 \in G_1$ and $r_{i+1} \in \mathbb{Z}/N\mathbb{Z}$, and

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} g_2^{r_{i+1}\rho_k(\vec{x}) + \sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)}, \\ z_k &\leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}, \text{ and} \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{r_{i+1} f(\vec{x}) + \sum_{j=1}^{i} r_j f(\vec{x}_j)}, \end{aligned}$$

which are identical to the values in $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$.

Equation 10: $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$ to $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$.

The value $g_2^{\sum_{j=1}^{i} r_j \rho_k(\vec{x}_j)}$ is independent from $\vec{x}$ and $\vec{x}_{i+1}$, as is the corresponding value using $f(\cdot)$. By extended parameter hiding with respect to $R \cup \{f\}$ and $\{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$ (which allows one to compute the necessary components in H and G_T), the distributions over $g_1^{\rho_k(\vec{x})} g_2^{r_{i+1}\rho_k(\vec{x})}$ and $g_1^{\rho_k(\vec{x})} g_2^{r_{i+1}\rho_k(\vec{x}_{i+1})}$ are identical for all $k \in [r]$, as are the corresponding values using $f(\cdot)$.

Equation 11.

This follows by definition, as the R, S, T, and f values have now changed to the form specified in Assumption 3.2. □

B A Proof of Theorem 4.3

Proof. Let $\mathcal{A}$ be a PT adversary playing game $\mathsf{type}\text{-}\mathrm{RandUBER}_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$, and let $\mathbf{Adv}_{\mathcal{A}}^{4.2}(\lambda)$ denote its advantage in Assumption 4.2. We provide PT adversaries $\mathcal{C}_0$ and $\mathcal{C}_1$, and a family of PT adversaries $\mathcal{B}_i$ and

```
MAIN $\text{type-RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ / $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$ / $\boxed{\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)}$ / $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 4)$
$g \xleftarrow{\$} G,$ $\boxed{g_1 \xleftarrow{\$} G_1,\ g_2 \xleftarrow{\$} G_2 \setminus \{1\},\ g_3 \xleftarrow{\$} G_3 \setminus \{1\}}$, $g_4 \xleftarrow{\$} G_4$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$h, \hat{h} \xleftarrow{\$} H_{123},$ $\boxed{h_1 \xleftarrow{\$} H_1 \setminus \{1\}}$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$x_1, \ldots, x_c, \chi_1, \ldots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g^{\rho_k(\vec{x})} g_4^{\chi_k}\ \forall k \in [r]$    // $\text{type-RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$
$\boxed{v_k \leftarrow g_{123}^{\rho_k(\vec{x})} g_4^{\chi_k}}$ $\forall k \in [r]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$\boxed{v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\rho_k(\vec{x}')} g_3^{\rho(\vec{x}'')} g_4^{\chi_k}}$ $\forall k \in [r]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\rho_k(\vec{x}')+r''\rho(\vec{x}'')} g_4^{\chi_k}\ \forall k \in [r]$    // $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$
$y_k \leftarrow h^{\sigma_k(\vec{x})},$ $\boxed{y_k \leftarrow h_1^{\sigma_k(\vec{x})}}$ $\forall k \in [s]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$z_k \leftarrow e(g, h)^{\tau_k(\vec{x})},$ $\boxed{z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}}$ $\forall k \in [t]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g, \hat{h})^{f(\vec{x})}$    // $\text{type-RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$
$\boxed{\mathsf{chal} \leftarrow e(g_{123}, \hat{h})^{f(\vec{x})}}$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)}$
$\boxed{\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{f(\vec{x}')} e(g_3, \hat{h})^{f(\vec{x}'')}}$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)}$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{f(\vec{x}')+r''f(\vec{x}'')}$    // $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$
return type-PLAY$(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

Figure 5: Games for the proof of Theorem 4.3 (Equations 12-14). For the terms in G, each game uses the boxed code on its corresponding line, and for the terms in H and G_T the boxed game introduces the boxed code on its corresponding line.

```
MAIN $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$ / $\boxed{\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)}$ / $\boxed{\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)}$ / $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_{2^i}$ $\boxed{, \vec{x}_{2^i+1}, \ldots, \vec{x}_{2^{i+1}}}$ $\xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c,\ r_1, \ldots, r_{2^i}$ $\boxed{, r_{2^i+1}, \ldots r_{2^{i+1}}}$ $\xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k}\ \forall k \in [r]$    // $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$
$\boxed{v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_2 g_3)^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k}}$ $\forall k \in [r]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)}$
$\boxed{v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_3^{\sum_{j=2^i+1}^{2^{i+1}} r_j \rho_k(\vec{x}_j)} g_4^{\chi_k}}$ $\forall k \in [r]$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_{i+1}[\rho_k]} g_4^{\chi_k}\ \forall k \in [r]$    // $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_i[f]}$    // $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$
$\boxed{\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2 g_3, \hat{h})^{\mathbf{sum}_i[f]}}$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)}$
$\boxed{\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_i[f]} e(g_3, \hat{h})^{\sum_{j=2^i+1}^{2^{i+1}} r_j f(\vec{x}_j)}}$    // $\boxed{\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)}$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_{i+1}[f]}$    // $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$
```

Figure 6: Games for the proof of Theorem 4.3 (Equations 15-17), using $\mathbf{sum}_i[\rho_k] = \sum_{j=1}^{2^i} r_j \rho_k(\vec{x}_j)$. Each game uses the boxed code on its corresponding line.

B_{i+1} such that

$$\begin{aligned}\mathbf{Adv}^{\text{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) \leq &\mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_0}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_1}(\lambda) + \mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda)\\ &+ \log_2(\ell)\left(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_i}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i+1}}(\lambda)\right)\end{aligned}$$

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{C}_0$ and $\mathcal{C}_1$, and $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$ for all i, $1 \leq i \leq \ell$, such that

$$\Pr[\textsf{type-}\mathrm{RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)] \leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_0}(\lambda) \tag{12}$$

$$(H_1, H_{123}; \mu = \{g_4, h_{123}\})$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda) - \Pr[\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)] = 0 \tag{13}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)] \leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_1}(\lambda) \tag{14}$$

$$(G_{24}, G_{34}; \mu = \{g_1, g_{24}, g_4, h_1, h_{123}\})$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{i}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] \leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_i}(\lambda) \tag{15}$$

$$(G_{24}, G_{34}; \mu = \{g_1, g_{24}, g_4, h_1, h_{123}\})$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)] = 0 \tag{16}$$

$$\Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)] \leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i+1}}(\lambda) \tag{17}$$

$$(G_{24}, G_{34}; \mu = \{g_1, g_{24}, g_4, h_1, h_{123}\})$$

We then have that

$$\begin{aligned}\mathbf{Adv}^{\text{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) &= \Pr[\textsf{type-}\mathrm{RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)]\\ &= (\Pr[\textsf{type-}\mathrm{RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)])\\ &\quad + (\Pr[\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)]) + (\Pr[\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{1}(\lambda)])\\ &\quad + \Bigg(\sum_{i=1}^{\log_2(\ell)} ((\Pr[\mathsf{G}^{\mathcal{A}}_{i}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)]) + (\Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)])\\ &\quad + (\Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)])\Bigg) + \Pr[\mathsf{G}^{\mathcal{A}}_{\ell+1}(\lambda)]\\ &\leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_0}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_1}(\lambda) + \log_2(\ell)(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_i}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i+1}}(\lambda))\\ &\qquad + \mathbf{Adv}^{4.2}_{\mathcal{A}}(\lambda).\end{aligned}$$

To ease exposition, we use $\mathbf{sum}_i[\phi]$ to denote $\mathbf{sum}_i[\phi] = \sum_{j=1}^{2^i} r_j \phi(\vec{x}_j)$, and use $\hat{h}$ as auxiliary input rather than h_{123}.

Equation 12: $\textsf{type-}\mathrm{UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)$

$\mathcal{C}_0$ behaves as follows (in attacking the first variant of subgroup hiding in Theorem 4.3, in which we transition from H_{123} to H_1 using $\mu = \{g_4, h_{123}\}$):

```
$\underline{\mathcal{C}_0(1^\lambda, N, G, H, G_T, e, g_4, \hat{h}, w)}$
$x_1, \dots, x_c, \chi_1, \dots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}, g \xleftarrow{\$} G$
$v_k \leftarrow g^{\rho_k(\vec{x})} {g_4}^{\chi_k}\ \forall k \in [r]$
$y_k \leftarrow w^{\sigma_k(\vec{x})}\ \forall k \in [s]$
$z_k \leftarrow e(g, w)^{\tau_k(\vec{x})}\ \forall k \in [t]$
$\textsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \dots, v_r, y_1, \dots, y_s, z_1, \dots, z_t)$
$\textsf{chal} \leftarrow e(g, \hat{h})^{f(\vec{x})}$
return $\textsf{type-}\mathrm{PLAY}(\lambda, \textsf{inputs}, \textsf{chal})$
```

If $w \xleftarrow{\$} H_{123}$ this is identical to $\textsf{type-}\mathrm{RandUBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$, and if instead $w \xleftarrow{\$} H_1$ this is identical to $\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)$.

Equation 13: $\mathsf{G}^{\mathcal{A}}_{0,1}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)$.

By extended parameter hiding with respect to $R \cup \{f\}$, and $\mathsf{aux} = \{h_1^{\sigma(\cdot)}, g_3^{\rho(\cdot)}\}_{\rho \in R \cup \{f\}, \sigma \in S \cup T}$, and subgroups (G_1, G_2), the distributions over $g_1^{\rho_k(\vec{x})} g_2^{\rho_k(\vec{x})} g_3^{\rho_k(\vec{x})}$ and $g_1^{\rho_k(\vec{x})} g_2^{\rho_k(\vec{x}')} g_3^{\rho_k(\vec{x})}$ for all $k \in [r]$ are identical, and this similarly holds if we also consider the corresponding values using $f(\cdot)$. The extra h_1 values allow us to compute the necessary values in H and G_T (and $\hat{h}$ is independent of all of these values, as it depends only on the description of H).

Similarly, by extended parameter hiding for $R \cup \{f\}$, $\mathsf{aux} = \{h_1^{\sigma(\cdot)}\}_{\sigma \in S \cup T}$, and subgroups (G_1, G_3), the distributions over $g_1^{\rho_k(\vec{x})} g_2^{\rho_k(\vec{x}')} g_3^{\rho_k(\vec{x})}$ and ${g_1}^{\rho_k(\vec{x})} {g_2}^{\rho_k(\vec{x}')} {g_3}^{\rho_k(\vec{x}'')}$ for all $k \in [r]$ are identical, and this similarly holds if we consider the corresponding values using $f(\cdot)$. (The $g_2^{\rho_k(\vec{x}')}$ values are independent of $\vec{x}$ and $\vec{x}''$, so they can be sampled independently.) Again, the extra h_1 values allow us to compute the necessary values in H and G_T.

Equation 14: $\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$

$\mathcal{C}_1$ behaves as follows (in attacking the second variant of subgroup hiding in Theorem 4.3, in which we transition from G_{24} to G_{34} using $\mu = \{g_1, g_{24}, g_4, h_1, \hat{h}\}$):

```
$\mathcal{C}_1(1^\lambda, N, G, H, G_T, e, g_1, g_{24}, g_4, h_1, \hat{h}, w)$
$x_1, x_1', x_1'', \ldots, x_c, x_c', x_c'', \chi_1, \ldots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_{24})^{\rho_k(\vec{x}')} w^{\rho_k(\vec{x}'')} g_4^{\chi_k} \ \forall k \in [r]$
$y_k \leftarrow h_1^{\sigma_k(\vec{x})} \ \forall k \in [s]$
$z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})} \ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_{24}, \hat{h})^{f(\vec{x}')} e(w, \hat{h})^{f(\vec{x}'')}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

If $w \xleftarrow{\$} G_{34}$ then this is distributed identically to $\mathsf{G}^{\mathcal{A}}_{0,2}(\lambda)$. If instead $w \xleftarrow{\$} G_{24}$ then this is distributed identically to $\mathsf{G}^{\mathcal{A}}_{1}(\lambda)$. (Note that in both games the extra g_4 values in the v_k can be absorbed into χ_k and the extra g_4 values in chal cancel out when paired with $\hat{h}$.)

Equation 15: $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$

$\mathcal{B}_i$ behaves as follows (in attacking the second variant of subgroup hiding in Theorem 4.3, in which we transition from G_{24} to G_{34} using $\mu = \{g_1, g_{24}, g_4, h_1, \hat{h}\}$):

```
$\mathcal{B}_i(1^\lambda, N, G, H, G_T, e, g_1, g_{24}, g_4, h_1, \hat{h}, w)$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_{2^i} \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c, r_1, \ldots r_{2^i}, \chi_1, \ldots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_{24} w)^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k} \ \forall k \in [r]$
$y_k \leftarrow h_1^{\sigma_k(\vec{x})} \ \forall k \in [s]$
$z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})} \ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_{24} w, \hat{h})^{\mathbf{sum}_i[f]}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

If $w \xleftarrow{\$} G_{24}$, then $w = g_2^{\chi^*} g_4^{\chi'}$ for uniformly distributed $\chi^*, \chi' \in \mathbb{Z}/N\mathbb{Z}$, and

$$v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_2 g_2^{\chi^*} g_4 g_4^{\chi'})^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k} = g_1^{\rho_k(\vec{x})} g_2^{\chi^{**} \mathbf{sum}_i[\rho_k]} g_4^{\chi_k'} \quad \text{and}$$
$$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2 g_2^{\chi^*} g_4 g_4^{\chi'}, \hat{h})^{\mathbf{sum}_i[f]} = e(g_1, \hat{h})^{f(\vec{x})} e(g_2^{\chi^{**}}, \hat{h})^{\mathbf{sum}_i[f]},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$. If instead $w \xleftarrow{\$} G_{34}$, then $w = g_3 g_4^{\chi'}$ for uniformly distributed $g_3 \in G_3$ and $\chi' \in \mathbb{Z}/N\mathbb{Z}$, and

$$v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_2 g_3^{\chi^*} g_4 g_4^{\chi'})^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k} = g_1^{\rho_k(\vec{x})} (g_2 g_3)^{\mathbf{sum}_i[\rho_k]} g_4^{\chi_k'} \quad \text{and}$$
$$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2 g_3 g_4 g_4^{\chi'}, \hat{h})^{\mathbf{sum}_i[f]} = e(g_1, \hat{h})^{f(\vec{x})} e(g_2 g_3, \hat{h})^{\mathbf{sum}_i[f]},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$.

Equation 16: $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$.

Here we use extended parameter hiding with respect to the function class defined in Equation 5, $\mathsf{aux} = \emptyset$, and subgroups (G_2, G_3). The $g_1^{\rho_k(\vec{x})}$ components and the y_k and z_k values all concern only $\vec{x}$, which is independent of all the $\vec{x}_j, \vec{x}'_j$ and r_j, r'_j variables. Thus, extended parameter hiding tells us that the distributions over

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_3^{\sum_{i=1}^{2^i} r_j \rho_k(\vec{x})} g_4^{\chi_k}, \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_i[f]} e(g_3, \hat{h})^{\sum_{j=1}^{2^i} r_j f(\vec{x}_j)} \end{aligned}$$

and

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_3^{\sum_{i=1}^{2^i} r'_j \rho_k(\vec{x}')} g_4^{\chi_k}, \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x}_j)} e(g_2, \hat{h})^{\mathbf{sum}_i[f]} e(g_3, \hat{h})^{\sum_{j=1}^{2^i} r'_j f(\vec{x}'_j)} \end{aligned}$$

are identical. Thus if we define $r_{2^i+j} = r'_j$ and $\vec{x}_{2^i+j} = \vec{x}'_j$ for $j \in \{1 \ldots 2^i\}$ we have that $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$ and $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$ are identically distributed.

Equation 17: $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$.

$\mathcal{B}_{i+1}$ behaves as follows (in attacking the second variant of subgroup hiding in Theorem 4.3, in which we transition from G_{34} to G_{24} using $\mu = \{g_1, g_{24}, g_4, h_1, \hat{h}\}$):

```
$\underline{\mathcal{B}_{i+1}(1^\lambda, N, G, H, G_T, e, g_1, g_{24}, g_4, h_1, \hat{h}, w)}$
$\vec{x}, \vec{x}_1, \ldots, \vec{x}_{i+1} \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^c, r_1, \ldots r_{i+1}, \chi_1, \ldots, \chi_r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(\vec{x})} (g_{24})^{\mathbf{sum}_i[\rho_k]} w^{\sum_{j=2^i+1}^{2^{i+1}} r_j \rho_k(\vec{x}_j)} g_4^{\chi_k}\ \forall k \in [r]$
$y_k \leftarrow h_1^{\sigma_k(\vec{x})}\ \forall k \in [s]$
$z_k \leftarrow e(g_1, h_1)^{\tau_k(\vec{x})}\ \forall k \in [t]$
$\mathsf{inputs} \leftarrow (N, G, H, G_T, e, g_4, \hat{h}, v_0, \ldots, v_r, y_1, \ldots, y_s, z_1, \ldots, z_t)$
$\mathsf{chal} \leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_{24}, \hat{h})^{\mathbf{sum}_i[f]} e(w, \hat{h})^{\sum_{j=2^i+1}^{2^{i+1}} r_j f(\vec{x}_j)}$
return $\mathsf{type}\text{-}\textsc{play}(\lambda, \mathsf{inputs}, \mathsf{chal})$
```

If $w \xleftarrow{\$} G_3G_4$ then $w = g_3 g_4^{\chi'}$ for uniformly distributed $g_3 \in G_3$ and $\chi' \in \mathbb{Z}/N\mathbb{Z}$, and

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} (g_{24})^{\mathbf{sum}_i[\rho_k]} (g_3 g_4^{\chi'})^{\sum_{j=2^i+1}^{2^{i+1}} r_j \rho_k(\vec{x}_j)} g_4^{\chi_k} \\ &= g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_3^{\sum_{j=2^i+1}^{2^{i+1}} r_j \rho_k(\vec{x}_j)} g_4^{\chi'_k} \quad \text{and} \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_{24}, \hat{h})^{\mathbf{sum}_i[f]} e(g_3 g_4^{\chi'}, \hat{h})^{\sum_{j=2^i}^{2^{i+1}} r_j f(\vec{x}_j)} \\ &= e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_i[f]} e(g_3, \hat{h})^{\sum_{j=2^i}^{2^{i+1}} r_j f(\vec{x}_j)}, \end{aligned}$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$. If instead $w \xleftarrow{\$} G_2G_4$ then $w = g_2^{\chi^*} g_4^{\chi'}$ for uniformly distributed $\chi^*, \chi' \in \mathbb{Z}/N\mathbb{Z}$, and

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(\vec{x})} (g_{24})^{\mathbf{sum}_i[\rho_k]} (g_2^{\chi^*} g_4^{\chi'})^{\sum_{j=2^i+1}^{2^{i+1}} r_j \rho_k(\vec{x}_j)} g_4^{\chi_k} \\ &= g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_i[\rho_k]} g_2^{\sum_{j=2^i+1}^{2^{i+1}} \chi^* r_j \rho_k(\vec{x}_j)} g_4^{\chi'_k} \\ &= g_1^{\rho_k(\vec{x})} g_2^{\mathbf{sum}_{i+1}[\rho_k]} g_4^{\chi'_k} \quad \text{and} \\ \mathsf{chal} &\leftarrow e(g_1, \hat{h})^{f(\vec{x})} e(g_{24}, \hat{h})^{\mathbf{sum}_i[f]} e(g_2^{\chi^*} g_4^{\chi'}, \hat{h})^{\sum_{j=2^i+1}^{2^{i+1}} r_j f(\vec{x}_j)} \\ &= e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_i[f]} e(g_2, \hat{h})^{\sum_{j=2^i}^{2^{i+1}} \chi^* r_j f(\vec{x}_j)} \\ &= e(g_1, \hat{h})^{f(\vec{x})} e(g_2, \hat{h})^{\mathbf{sum}_{i+1}[f]}, \end{aligned}$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$. (To see this, note that we can absorb the χ^* into each of the r_j's for $j > 2^i$.) □

C Dependency Graphs from Section 3.3

In this section, we include the rest of the dependency graphs for the converted schemes in Table 1. As a reminder from Section 3.3 (in which we included the graph for the Boneh-Boyen-Goh HIBE), the shape of the node indicates which side of the split each element goes on: triangle nodes are in G, inverted triangle nodes are in H, and diamond nodes are replicated across G and H. Pairing equations are denoted by $pn[i]$, where $n \in \mathbb{N}$ indicates a particular usage of the pairing and $i \in \{0, 1\}$ indicates the side of the pairing in which the element is used. The nodes with an i included represent multiple (related) values; e.g., the node gi represents $\{g^{\alpha^i}\}_i$.

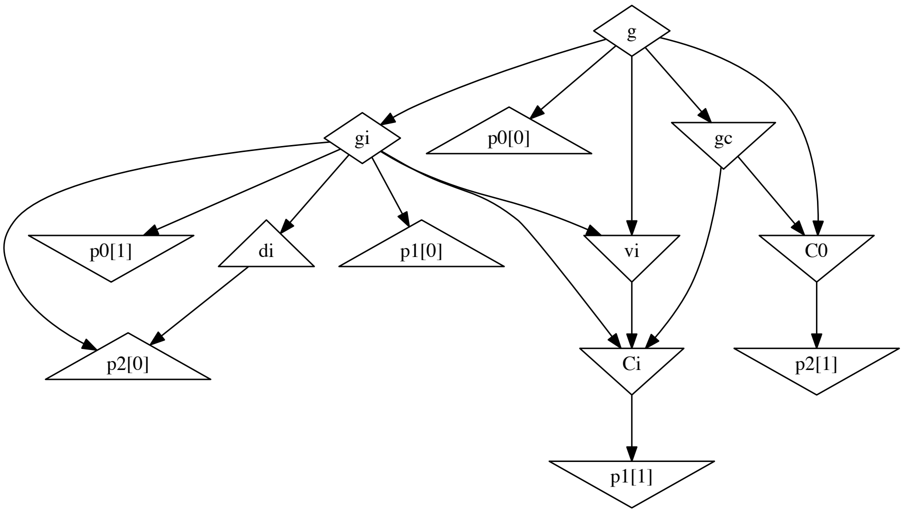

Figure 7: Dependency graph for the BGW broadcast encryption scheme [13]. The public key consists of g, gi and vi, and the secret key of di. Encryption uses the pairing $p0$ and produces $C0$ and Ci, and decryption uses the pairings $p1$ and $p2$. In the reduction, gi are derived from the q-BDHE assumption.

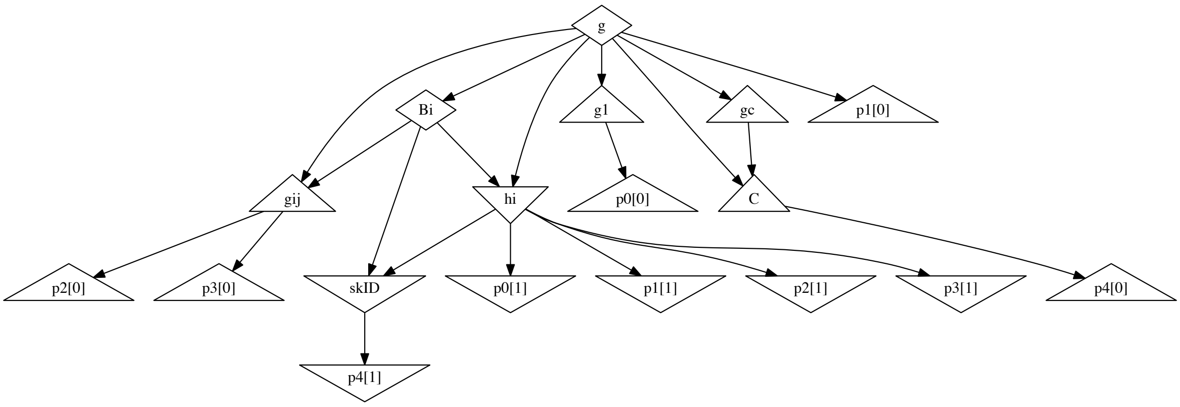

Figure 8: Dependency graph for the ACF IB-KEM [2]. The master public key consists of g, gij, and $g1$. Secret key derivation uses hi as the auxiliary information and $skID$ as the secret key for identity ID. The pairing $p0$ and the ciphertext C are used in the encapsulation process, decapsulation uses the pairings $p1$, $p2$, and $p3$, and the key is calculated from the encapsulation using $p4$. In the reduction, Bi and gc are derived from the q-wBDHI assumption.

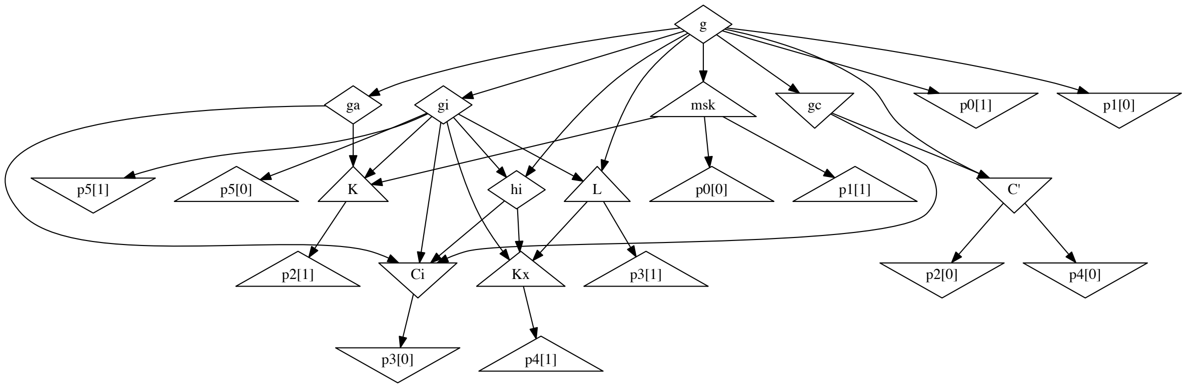

Figure 9: Dependency graph for the Waters ABE scheme [41]. The public key consists of g, ga, and hi, and is computed using the pairing $p0$. msk denotes the master secret key and the secret key consists of K, Kx, and L. Encryption uses the pairing $p1$ and produces C' and Ci, and decryption uses the pairings $p2$, $p3$, and $p4$. In the reduction, gi and gc are derived from the q-BDHE assumption, and the pairing $p5$ is used to simulate the pairing $p0$.