Déjà Q: Using Dual Systems to Revisit q-Type Assumptions

Melissa Chase
Microsoft Research Redmond
melissac@microsoft.com

Sarah Meiklejohn[*]
UC San Diego
smeiklej@cs.ucsd.edu

Abstract

After more than a decade of usage, bilinear groups have established their place in the cryptographic canon by enabling the construction of many advanced cryptographic primitives. Unfortunately, this explosion in functionality has been accompanied by an analogous growth in the complexity of the assumptions used to prove security. Many of these assumptions have been gathered under the umbrella of the "uber-assumption," yet certain classes of these assumptions — namely, *q-type* assumptions — are stronger and require larger parameter sizes than their static counterparts.

In this paper, we show that in certain bilinear groups, many classes of q-type assumptions are in fact implied by subgroup hiding (a well-established, static assumption). Our main tool in this endeavor is the *dual-system* technique, as introduced by Waters in 2009. As a case study, we first show that in composite-order groups, we can prove the security of the Dodis-Yampolskiy PRF based solely on subgroup hiding and allow for a domain of arbitrary size (the original proof only allowed a logarithmically-sized domain). We then turn our attention to classes of q-type assumptions and show that they are implied — when instantiated in appropriate groups — solely by subgroup hiding. These classes are quite general and include assumptions such as q-SDH. Concretely, our result implies that every construction relying on such assumptions for security (e.g., Boneh-Boyen signatures) can, when instantiated in appropriate composite-order bilinear groups, be proved secure under subgroup hiding instead.

1 Introduction

For the past decade, bilinear groups — i.e., groups equipped with a bilinear map, or pairing — have allowed for the efficient construction of a wide variety of advanced cryptographic primitives, including (but by no means limited to): signatures [13, 6, 8, 38], group signatures [10, 16, 23], zero-knowledge proofs [24, 25], (hierarchical) identity-based encryption [11, 7, 9, 34], and functional and attribute-based encryption [32, 36, 37]. As such, pairings are now used as a standard general-purpose tool in cryptographic constructions.

Unfortunately, this growth in the complexity of cryptographic primitives has been accompanied by an analogous growth in the complexity of the assumptions required to prove security. While assumptions such as Bilinear Diffie Hellman (BDH) [11] and Decision Linear [10] have become relatively standard, the use of pairings has also ushered in various classes of assumptions such as *q-type assumptions*, in which the size of the assumption grows dynamically, or *interactive assumptions*, in which the adversary is given access to some oracle(s). For example, in the q-DBDHI (Decisional Bilinear Diffie Hellman Inversion) assumption, the adversary is given $(g, g^x, g^{x^2}, \ldots, g^{x^q})$ and is asked to produce $e(g,g)^{1/x}$. While the "uber-assumption" [9, 15] generalizes many q-type assumptions (as well as many static assumptions) and provides a lower bound for their security in the generic group model [46], such assumptions nevertheless remain less understood than their static counterparts.

[*] Work done as an intern at Microsoft Research Redmond.

Beyond the lack of understanding of such assumptions, the fact that they scale asymptotically with the security of the scheme can be problematic. In a reduction, the value of q is frequently tied to the number of queries that the adversary makes to an oracle. As a result, q must scale with some parameter of the system; e.g., for identity-based encryption, q must be at least as big as the number of parties that the adversary is able to corrupt. As it is typically the case that an assumption parameterized by q' implies the same assumption parameterized by q for $q' > q$ (as the assumption parameterized by q' gives out strictly more information), this means that the assumption gets stronger as the adversary is able to corrupt more parties. In some cases, this correlation is more striking. For example, Dodis and Yampolskiy [20] use the $a(\lambda)$-DBDHI assumption to prove the security of their pseudorandom function (PRF), where $a(\lambda)$ is the size of the domain of the PRF (and λ is the security parameter); as a result, the domain is restricted to be of logarithmic size. This correlation is furthermore not always an artifact of proof techniques, as Jao and Yoshida [27] showed that Boneh-Boyen signatures were in fact equivalent to the q-SDH assumption that they rely on for security. Finally, Cheon [19] showed that the time required to recover a secret key scales inversely with q,so that if recovering a secret key takes time t when using $q = 1$ (e.g, it takes t steps to recover x given g and g^x), then it takes time $t/\sqrt{q}$ in the general case (e.g., given $(g, g^x, \ldots, g^{x^q})$). This means that constructions rely on asymptotically stronger assumptions to obtain stronger security guarantees, so the parameters must grow appropriately in order to maintain a constant level of security (e.g., 128-bit security).

On the positive side, one technique that has proved particularly effective at avoiding q-type assumptions — and boosting security as a result — is the *dual-system* technique, which was introduced by Waters [47] in 2009 and has been used extensively since [34, 32, 33, 36, 31, 37]. Briefly, this technique takes advantage of *subgroup hiding* in bilinear groups [12]; i.e., the assumption, in a group of composite order $N = p_1p_2$, that a random element of the full group is indistinguishable from a random element of order p_1. (Subgroup hiding can also be defined, albeit in a more complex way, for vector spaces over prime-order bilinear groups.) Using this core assumption, the dual-system technique begins with a scheme in a particular subgroup (for concreteness, the subgroup of elements of order p_1); i.e., a scheme in which all elements are contained solely within the subgroup. To prove security, a "shadow" copy of the original scheme is first added in a new subgroup (e.g., the subgroup of order p_2); the addition of this shadow copy goes unnoticed by subgroup hiding. Using a property called *parameter hiding* [31], this shadow copy is then randomized, so the value in the additional subgroup is now unstructured; in Waters' terminology, this object is now *semi-functional.* This randomness is then pushed back into the original subgroup, again using subgroup hiding, and is used to blind the structure of the original scheme; e.g., in an IND-CPA game it can be used to obscure all information about the challenge message.

Our contributions. In this paper, we expand the usage of the dual-system technique. Rather than work at the level of constructions, we show directly that many q-type assumptions can be implied — with a crucial looseness of q — by subgroup hiding. In some sense, we thus interpret our approach as *absorbing* rather than avoiding q-type assumptions, and believe our work takes a (perhaps surprising) step in expanding the power of the dual-system technique.

As a first exercise, we prove in Section 3 that the Dodis-Yampolskiy PRF — unmodified, but instantiated in a composite-order group — can be proved secure using only the subgroup hiding assumption. Because of the limitations (described above) in the original security proof, our result not only moves to a static assumption, but also boosts security to allow for domains of arbitrary size, which is useful in and of itself for the many applications of the Dodis-Yampolskiy PRF [17, 5, 18, 29].

Next, in Section 4, we look beyond cryptographic primitives and instead focus directly on the underlying assumptions, and in particular on the class of q-type assumptions that are instantiations of the uber-assumption. Here we show that many instantiations of the uber-assumption can be reduced — following a modified version of the dual-system technique, which still assumes subgroup hiding — to instantiations that are significantly weaker; in fact, in many cases we can reduce to an assumption so weak that it actually holds by a statistical argument. As examples, we revisit two well-known q-type assumptions. By applying our general theorem to these assumptions, we can reduce them to assumptions in which all secret information (e.g., the exponent x in q-DBDHI) is statistically hidden, so an adversary can do no better than a random guess and the security of the entire assumption collapses down to subgroup hiding.

Finally, in Section 5, we discuss the concrete implications of our work; i.e., in which concrete bilinear settings the abstract requirements of the dual-system technique (namely, subgroup hiding and parameter hiding) can be expected to hold. Due to current limitations in the parameter hiding supported by prime-order bilinear groups, our strongest results can be applied only in asymmetric composite-order bilinear groups [14, 40].

Putting it all together, we obtain the following concrete results:

- In a composite-order group (such as the target group of a composite-order pairing, or any composite-order elliptic curve group without a pairing), subgroup hiding implies any q-type assumption where the exponents are linearly independent rational functions.
- In an asymmetric composite-order bilinear group, subgroup hiding implies any q-type assumption where the adversary is given elements with secret exponents on only one side of the pairing, where the exponents are linearly independent rational functions, and where the adversary must distinguish a particular value of the group (or target group) from a random value.
- In an asymmetric composite-order bilinear group, subgroup hiding implies any q-type assumption where the exponents are linearly independent rational functions and the adversary must compute a value in the source group.

Related work. As mentioned above, the dual-system technique was first introduced by Waters in 2009 [47], and was applied subsequently to achieve a wide variety of results [34, 32, 44, 33, 36, 35, 31, 45], all involving randomized public-key primitives (e.g., identity-based encryption) in bilinear groups.

To the best of our knowledge, we are the first to systematically apply the dual-system technique directly to assumptions, and in particular to q-type assumptions. Boneh, Boyen, and Goh [9] analyzed the security of the uber-assumption — which includes many q-type assumptions — in the generic group model, and derived generic lower bounds on the runtime of an adversary that could break the uber-assumption; this work was later extended by Jager and Rupp [26], who showed the equivalence of many assumptions in the *semi-generic* group model. Our result is somewhat orthogonal to theirs, as we seek to show that in certain concrete (i.e., non-generic) settings these assumptions actually reduce to subgroup hiding. Anecdotally, several results use the dual-system technique to eliminate the requirement on q-type assumptions for specific primitives or constructions: for example, Gerbush et al. [21] obtained Camenisch-Lysyanskaya signatures under static assumptions, as opposed to the interactive LRSW assumption; Abe et al. [1] achieve efficient structure preserving signatures under DLIN while previous efficient constructions [3, 2] required q-type or interactive assumptions; Attrapadung and Libert achieved the first identity-based broadcast encryption scheme with short ciphertexts [4]; and the original result of Waters [47] achieved the first secure HIBE under non-q-type assumptions.

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.

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 Bilinear groups

We refer to a *bilinear group* as a tuple $\mathbb{G} = (N, G, H, G_T, e)$, where N can be either prime or composite, $|G| = |H| = kN$ and $|G_T| = \ell N$ for some $k, \ell \in \mathbb{N}$, and $e : G \times H \to G_T$ is a bilinear map, meaning it is (1) efficiently computable; (2) satisfies bilinearity: $e(x^a, y^b) = e(x, y)^{ab}$ for all $x \in G$, $y \in H$, and $a, b \in \mathbb{Z}/N\mathbb{Z}$; and (3) satisfies non-degeneracy: if $e(x, y) = 1$ for all $y \in H$ then $x = 1$ and if $e(x, y) = 1$ for all $x \in G$ then $y = 1$. When G and H are cyclic, we may include in $\mathbb{G}$ generators g and h of G and H respectively, and when the groups G and H decompose into cyclic subgroups $G = G_1 \oplus G_2$ and $H = H_1 \oplus H_2$, we may additionally include descriptions of these subgroups and/or their generators. In what follows, we use $\mathsf{BilinearGen}$ to denote the algorithm by which bilinear groups are generated, and provide it with an argument n that specifies the number of subgroups.

There are two additional structural properties of bilinear groups that are exploited in the dual-system technique: subgroup hiding and parameter hiding. Subgroup hiding is a computational assumption that requires that, if G (respectively H) decomposes into two subgroups, then distinguishing between a random element of the full group and a random element of one of the subgroups should be hard. (This is actually the specific simple case of subgroup hiding originally introduced by Boneh, Goh, and Nissim [12]; more general definitions exist as well [31, 30].)

In fact, subgroup hiding can be defined for arbitrary groups; e.g., groups over general elliptic curves, or composite-order subgroups of finite fields. As some of our results (in particular, our PRF in Section 3) apply to these more general settings, we define a more general version of subgroup hiding and treat the version in bilinear groups as a special case.

Let $\mathsf{GroupGen}$ denote an algorithm that, on input 1^λ and an integer $n \in \mathbb{N}$, outputs (N, G, μ), where G is a group of order N that decomposes into n subgroups, and μ is any relevant additional information; e.g., $(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, n)$ can be cast as $(N, G, \mu) \xleftarrow{\$} \mathsf{GroupGen}(1^\lambda, n)$ for $\mu = (H, G_T, e)$, and for regular finite fields μ might contain generator(s) of G or its subgroups.

Assumption 2.1 (Subgroup hiding)**.** For a group generation algorithm $\mathsf{GroupGen}(\cdot, \cdot)$, *subgroup hiding* holds if no PT adversary $\mathcal{A}$ has a non-negligible chance of distinguishing a random element of the subgroup G_1 from a random element of the group G. Formally, define $\mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{A}}(\lambda) = 2\Pr[\mathrm{SGH}^{\mathcal{A}}_{\mu}(\lambda)] - 1$, where $\mathrm{SGH}^{\mathcal{A}}_{\mu}(\lambda)$ is defined as follows:

```
MAIN $\mathrm{SGH}^{\mathcal{A}}_{\mu}(\lambda)$
$b \xleftarrow{\$} \{0,1\}$; $(N, G, G_1, \mu) \xleftarrow{\$} \mathsf{GroupGen}(1^\lambda, 2)$
if $(b = 0)$ then $T \xleftarrow{\$} G$
if $(b = 1)$ then $T \xleftarrow{\$} G_1$
$b' \xleftarrow{\$} \mathcal{A}(N, G, \mu, T)$
return $(b' = b)$
```

Then *subgroup hiding* holds with respect to $\mathsf{GroupGen}$ and the auxiliary information μ if for all PT adversaries $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $\mathbf{Adv}^{\mathrm{sgh}}_{\mathcal{A}}(\lambda) < \nu(\lambda)$. (Subgroup hiding in G_2 is defined analogously.)

There are often limits to the auxiliary information that can be provided to $\mathcal{A}$; e.g., in the bilinear setting, if $\mathcal{A}$ is attempting to distinguish $T = g_1^r$ from $T = g^r$ for $r \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and has access to a canceling pairing $e(\cdot,\cdot)$ — i.e., a pairing such that $e(G_1, H_2) = e(G_2, H_1) = 1$ — and $h_2 \in \mu$, it can easily distinguish between these elements by checking if $e(T, h_2) = 1$ or not. Thus, if an adversary is trying to distinguish between a random element of G_1 and a random element of $G_1 \oplus G_2$ (analogously, if it is trying to distinguish between G_2 and $G_1 \oplus G_2$), the problem becomes easy if μ includes h_2 (analogously, h_1).

Parameter hiding, unlike subgroup hiding, is a statistical property of the group that allows certain distributions across subgroups to be independent. In composite-order groups, for example, the Chinese Remainder Theorem tells us that the values of $x \bmod p_1$ and $x \bmod p_2$ are independent, so that given g_1^x, the value of g_2^x is unconstrained. In prime-order groups, Lewko [31] demonstrated how to support parameter hiding with respect to linear functions; i.e., how — using appropriate constructions of G_1 and G_2 — the distributions of g_2^{ax} and g_2^r for $a, r \xleftarrow{\$} \mathbb{F}_p$ are identical, even given x and g_1^a. The first formal notion of parameter hiding with respect to these linear functions was later given by Lewko and Meiklejohn [30]; we generalize their notion as follows:

Definition 2.2 (Parameter hiding)**.** *For a group* $(N, G, G_1, G_2, \mu) \in [\mathsf{GroupGen}(1^\lambda, 2)]$*,* parameter hiding *holds with respect to a family of functions* $\mathcal{F}$ *if for all* $g_1 \in G_1$ *and* $g_2 \in G_2$*, the distribution* $\{g_1^{f(x_1,\ldots,x_n)} g_2^{f(x_1,\ldots,x_n)}\}_{f\in\mathcal{F}}$ *is identical to* $\{g_1^{f(x_1,\ldots,x_n)} g_2^{f(x'_1,\ldots,x'_n)}\}_{f\in\mathcal{F}}$ *for* $x_1, x'_1 \ldots, x_n, x'_n \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$*.*

As a simple example, if $\mathcal{F} = \{1, x_1\}$, then for $x_1, x'_1 \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, the distributions of $(g_1 g_2, g_1^{x_1} g_2^{x_1})$ and $(g_1 g_2, g_1^{x_1} g_2^{x'_1})$ are identical.

This definition of parameter hiding works well for many classes of functions $\mathcal{F}$, and applies to many of the concrete q-type assumptions we consider in Section 4. For more interactive settings, however, in which some of the inputs might be provided by an adversary — for example, for our PRF in Section 3, in which the adversary provides the input x — we can no longer model all inputs to the function as uniformly random. This in and of itself might not be a problem (in fact, if the class of functions satisfies the above definition, then we can still capture adversarial inputs without requiring a new definition), but for certain classes of functions an unbounded adversary might be able to easily distinguish the two distributions if allowed to query on all possible adversarial inputs. As this is the case for the class of functions we consider in Section 3, a new definition is required.

We now consider a potentially unbounded adversary that is allowed to see only polynomially many evaluations, but on inputs that it can choose adaptively. To match our intuition, we switch things around slightly: rather than consider all functions f and uniformly random inputs, we consider randomly sampled functions f and f' — with potentially different but overlapping domains $f.\mathcal{D}$ and $f'.\mathcal{D}$ — applied to adversarially chosen inputs $(x_1, \ldots, x_n)$, and require that the values $g_1^{f(x_1,\ldots,x_n)} g_2^{f(x_1,\ldots,x_n)}$ and $g_1^{f(x_1,\ldots,x_n)} g_2^{f'(x_1,\ldots,x_n)}$ are statistically indistinguishable. Our resulting definition, which we call *adaptive* parameter hiding, follows a game-based approach, but we stress that this notion of parameter hiding is still a statistical property, as we allow the distinguisher to be computationally unbounded.

Definition 2.3 (Adaptive parameter hiding)**.** *For a group* (N, G, G_1, G_2, μ) *and functions* f, f' *in a family* $\mathcal{F}$*, let* $\mathcal{O}(\cdot)$ *return* $g_1^{f(\cdot)} g_2^{f(\cdot)}$ *if the input is in* $f.\mathcal{D}$ *and* 1 *otherwise, and let* $\mathcal{O}'(\cdot)$ *return* $g_1^{f(\cdot)} g_2^{f'(\cdot)}$ *if the input is in* $f.\mathcal{D} \cap f'.\mathcal{D}$ *and* 1 *otherwise. Then* adaptive parameter hiding *holds with respect to* $\mathcal{F}$ *if for all* $\lambda \in \mathbb{N}$*,* $(N, G, G_1, G_2, \mu) \in [\mathsf{GroupGen}(1^\lambda, 2)]$*, and* $g_1 \in G_1$*,* $g_2 \in G_2$*, the oracles* $\mathcal{O}$ *and* $\mathcal{O}'$

are statistically indistinguishable if they are queried polynomially many times; i.e., for any (potentially unbounded) distinguisher D making poly(λ) queries, there exists a negligible function $\nu(\cdot)$ such that

$$Pr[f \xleftarrow{\$} \mathcal{F} : D^{\mathcal{O}(\cdot)} = 1] - Pr[f, f' \xleftarrow{\$} \mathcal{F} : D^{\mathcal{O}'(\cdot)} = 1] < \nu(\lambda).$$

We use these definitions in Sections 4, and discuss the different families of functions that can be supported in different types of bilinear groups in Section 5.

2.3 Pseudorandom functions

A pseudorandom function family [22] F specifies the algorithms $\mathsf{F.Pg}$, $\mathsf{F.Keys}$, $\mathsf{F.Dom}$, $\mathsf{F.Rng}$, and $\mathsf{F.Ev}$. Via $\mathit{fp} \xleftarrow{\$} \mathsf{F.Pg}(1^\lambda)$ one generates a description fp of a function $\mathsf{F.Ev}(1^\lambda, \mathit{fp})\colon \mathsf{F.Keys}(1^\lambda, \mathit{fp}) \times \mathsf{F.Dom}(1^\lambda, \mathit{fp}) \to \mathsf{F.Rng}(1^\lambda, \mathit{fp})$. The evaluation algorithm $\mathsf{F.Ev}$ is PT and deterministic.

Definition 2.4. *For a function family* F *and an adversary* $\mathcal{A}$*, let* $\mathbf{Adv}^{prf}_{\mathsf{F},\mathcal{A}}(\lambda) = 2\Pr[PRF^{\mathcal{A}}_{\mathsf{F}}(\lambda)] - 1$*, where* $PRF^{\mathcal{A}}_{\mathsf{F}}(\lambda)$ *is defined as follows:*

```
MAIN $PRF^{\mathcal{A}}_{\mathsf{F}}(\lambda)$
$b \xleftarrow{\$} \{0,1\}; \mathit{fp} \xleftarrow{\$} \mathsf{F.Pg}(1^\lambda); sk \xleftarrow{\$} \mathsf{F.Keys}(1^\lambda, \mathit{fp})$
$b' \xleftarrow{\$} \mathcal{A}^{\mathrm{FN}}(1^\lambda, \mathit{fp})$
return $(b' = b)$

Procedure $\mathrm{FN}_{sk}(x)$
if $b = 0$ $y \xleftarrow{\$} \mathsf{F.Rng}(1^\lambda, \mathit{fp})$
if $b = 1$ $y \leftarrow \mathsf{F.Ev}(1^\lambda, \mathit{fp}, sk, x)$
return $y$
```

Then F *is* pseudorandom *if for all PT algorithms* $\mathcal{A}$ *there exists a negligible function* $\nu(\cdot)$ *such that* $\mathbf{Adv}^{prf}_{\mathsf{F},\mathcal{A}}(\lambda) \le \nu(\lambda)$.

3 Pseudorandom Functions

In this section, we explore the security of the Dodis-Yampolskiy PRF [20]. First, we recall the Dodis-Yampolskiy PRF, instantiated for our purposes in a group of composite order $N = p_1 p_2$:

- $\mathsf{F.Pg}(1^\lambda)$: Sample $(N, G, (G_1, G_2)) \xleftarrow{\$} \mathsf{GroupGen}(1^\lambda, 2)$ and $u \xleftarrow{\$} G$; output (N, G, u). With this setup, $\mathsf{F.Keys} = \mathsf{F.Dom} = \mathbb{Z}/N\mathbb{Z}$, and $\mathsf{F.Rng} = G$.
- $\mathsf{F.Ev}(1^\lambda, \mathit{fp}, sk, x)$: Output $u^{(sk+x)^{-1}}$. If $(sk + x)^{-1}$ is undefined in $\mathbb{Z}/N\mathbb{Z}$, output 1.

Dodis and Yampolskiy originally showed that, when instantiated in the target group of a bilinear pairing (i.e., when using $\mathsf{BilinearGen}$ and $u \xleftarrow{\$} G_T$), this is a *verifiable* random function — a more powerful primitive than a PRF, as it comes with the additional ability to prove that the PRF value was computed correctly — under the q-DBDHI assumption, which states that when given $(g, g^x, \ldots, g^{x^q})$, it should be hard to distinguish $e(g,g)^{1/x}$ from random. Their reduction, however, is quite loose: if the size of the PRF domain is $a(\lambda)$, they use the $a(\lambda)$-DBDHI assumption and show that

$$\mathbf{Adv}^{\text{pr-vrf}}_{\mathsf{F},\mathcal{A}}(\lambda) \le a(\lambda) \cdot \mathbf{Adv}^{a(\lambda)\text{-DBDHI}}_{\mathcal{A}}(\lambda),$$

which means that the scheme is provably secure only if the domain is restricted to be of polynomial size (i.e., its size is polynomial in the size of the security parameter).

We instead show that

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{F},\mathcal{A}}(\lambda) \le q \cdot \mathbf{Adv}^{\text{sgh}}_{\mathcal{A}}(\lambda)$$

for an adversary $\mathcal{A}$ that makes q queries to the PRF oracle; while the reduction is still not tight, our approach nevertheless allows for a domain of arbitrary size. We have made two minor modifications: First, as mentioned above, Dodis and Yampolskiy considered the PRF in the target group G_T of a symmetric prime-order bilinear pairing, while we require an abstract group G in which subgroup hiding and parameter hiding hold. Our second modification is to use, rather than the "canonical" generator $e(g, h)$, a random generator $u \in G$. We stress that these modifications are purely syntactical and do not fundamentally alter the spirit of the construction (and, in particular, do not affect its usage in applications). They do, however, allow us to prove the following two results:

Lemma 3.1. *For all $\lambda \in \mathbb{N}$ and $(N, g, u) \in [\mathsf{F.Pg}(1^\lambda)]$, if $N = p_1 p_2$ for distinct primes $p_1, p_2 \in \Omega(2^{poly(\lambda)})$, then adaptive parameter hiding holds with respect to $\{f_{sk}(\cdot) : f_{sk}(x) = \frac{1}{sk+x}\}_{sk \in \mathsf{F.Keys}}$, where the domain for each function is $\mathcal{D}_{sk} = \{x \mid \gcd(x + sk, N) = 1\}$.*

Proof. We first show that for x such that $\gcd(s + x, N) = 1$,

$$((s + x)^{-1} \bmod N) \bmod p_1 = (s \bmod p_1 + x \bmod p_1)^{-1} \bmod p_1.$$

This is fairly straightforward: by definition, $(s+x)^{-1} \bmod N$ is a value $y \in \mathbb{Z}$ such that $(s+x)y + Nz = 1$ for some $z \in \mathbb{Z}$. Since $N = p_1 p_2$, $(s + x)y + p_1 p_2 z = 1$, and thus $(s + x)y + p_1 z' = 1$ for $z' = p_2 z$, which means $y \equiv (s + x)^{-1} \bmod p_1$ as well. Since $\gcd(x + s, N) = 1$, this inverse is well defined.

With this established, we define $f_{sk}(x) = \frac{1}{sk+x}$ and $\mathcal{D}_{sk} = \{x \mid \gcd(x + sk, N) = 1\}$, and observe that whenever $\gcd(x + sk, N) = 1$, $u_1^{f_{sk}(x)}$ can be computed knowing only the value of $sk \bmod p_1$, and that similarly $u_2^{f_{sk}(x)}$ can be computed knowing only the value of $sk \bmod p_2$. By the Chinese Remainder Theorem, these values are independent and thus, for any $sk, sk', x \in \mathbb{Z}/N\mathbb{Z}$ such that $\gcd(sk+x, N) = 1$ and $\gcd(sk' + x, N) = 1$, $u_1^{f_{sk}(x)} u_2^{f_{sk}(x)}$ and $u_1^{f_{sk}(x)} u_2^{f_{sk'}(x)}$ are distributed identically.

Now, suppose we choose $f, f' \xleftarrow{\$} \mathcal{F}$ and consider a hybrid oracle $\mathcal{O}_H$ that outputs $g_1^{f(\cdot)} g_2^{f(\cdot)}$ for inputs $x \in f.\mathcal{D} \cap f'.\mathcal{D}$ and 1 otherwise. For this choice of domain, $\mathcal{O}_H$ is statistically close to $\mathcal{O}$ as long as the distinguisher makes only polynomially many queries. To see this, note that as long as the distinguisher does not query $x \in f.\mathcal{D} \setminus f'.\mathcal{D}$, the two oracles are indistinguishable. Then, since $f'.\mathcal{D} = \{x \mid \gcd(sk' + x, N)\} = 1$, N is the product of two exponentially large primes, and sk' is not used in the rest of the oracle response, an adversary making only polynomially many queries has negligible probability of querying on x such that $\gcd(sk' + x, N) = 1$.

We finally argue that interactions with $\mathcal{O}_H$ and $\mathcal{O}'$ are identically distributed. To see this, suppose we choose $sk_1, sk'_1 \xleftarrow{\$} \mathbb{Z}/p_1\mathbb{Z}$ and $sk_2, sk'_2 \xleftarrow{\$} \mathbb{Z}/p_2\mathbb{Z}$, and set sk, sk' such that $sk = sk_1 \mod p_1$, $sk = sk_2 \mod p_2$, $sk' = sk'_1 \mod p_1$, and $sk' = sk'_2 \mod p_2$; this is identical to the choice of f, f' above. As described above, $\mathcal{O}_H$ outputs $g_1^{(sk_1+x)^{-1}} g_2^{(sk_2+x)^{-1}}$ on inputs $x \notin \{-sk_1, -sk'_2\} \mod p_1$ and $x \notin \{-sk_2, -sk'_2\} \mod p_2$, and 1 otherwise. Finally, if we instead set sk, sk' such that $sk = sk_1 \mod p_1$, $sk = sk'_2 \mod p_2$, $sk' = sk'_1 \mod p_1$, and $sk' = sk_2 \mod p_2$ (which again corresponds to $f, f' \xleftarrow{\$} \mathcal{F}$), $\mathcal{O}'$ will perform exactly the same computation. We thus conclude that the distribution of the two oracles is identical.

Putting this together with the above, we conclude that no distinguisher that makes polynomially many queries can distinguish $\mathcal{O}$ and $\mathcal{O}'$ with more than negligible probability. □

Theorem 3.2. *For all $\lambda \in \mathbb{N}$ and $fp \in [\mathsf{F.Pg}(1^\lambda)]$, if subgroup hiding holds with respect to* GroupGen *and $N = p_1 p_2$ for distinct primes $p_1, p_2 \in \Omega(2^{poly(\lambda)})$, then* F *is a pseudorandom function family.*

A proof of Theorem 3.2 can be found in Appendix A. Intuitively, our approach amplifies the only unknown value present in the PRF — namely, the sk value — as follows: first, we switch to using $u \in G_1$. Then this secret value sk is replicated in the G_2 subgroup, which is indistinguishable from the original by subgroup hiding. The secret value in the G_2 subgroup is then decoupled from the secret value in the G_1 subgroup, which is indistinguishable by adaptive parameter hiding. Finally, the new secret value from the G_2 subgroup is moved back into G_1, which is again indistinguishable by subgroup hiding. At this point, we now have one additional secret value in the PRF values we return. By repeating the process, we can embed polynomially many secret values (in particular, we embed as many values as there are oracle queries), at which point we have enough entropy to argue that the values returned by the PRF are statistically indistinguishable from truly random values.

One interesting feature of our approach is that — because we are using a deterministic primitive — we do not need to follow the traditional dual-system structure and adhere to a "query hybrid," in which each query to the oracle must be treated separately. Nevertheless, we do need to add enough additional degrees of randomness to cover all of the adversary's queries, so we still end up with a looseness of q in our reduction (but where q is the number of queries, not the size of the PRF domain).

4 Reducing q-Type Assumptions to Subgroup Hiding

Our main result in this section is to show that — if subgroup hiding holds and parameter hiding holds with respect to certain functions in the exponent — certain q-type assumptions are equivalent to significantly weaker assumptions. In fact, these equivalent assumptions are often so weak that they hold by a purely statistical argument, so the original assumption is fully implied by subgroup hiding.

We begin by recalling the uber-assumption, which serves as an umbrella for many q-type assumptions. We then describe two approaches: roughly, the first reduces any uber-assumption to subgroup hiding, but only if the assumption gives out meaningful functions on only one side of the pairing (or in the target group), and the second reduces any computational uber-assumption in the source group to subgroup hiding. Both of our reductions incur a looseness of q in the reduction, so we can think of them as "absorbing" the factor of q from the assumption rather than eliminating it outright.

4.1 The uber-assumption

We are able to examine many q-type assumptions at the same time using the "uber-assumption" [9, 15], which was first introduced by Boneh, Boyen, and Goh as a way to reason generally about a wide variety of pairing-based assumptions. They prove that if the parameters of the uber-assumption meet certain independence requirements then the assumption is hard in the generic group model, which eliminates the need to prove generic lower bounds for every individual instantiation of the assumption that is introduced. Our motivation, on the other hand, is to prove that many common instantiations of the assumption are in fact implied — assuming subgroup hiding holds in the bilinear group — by weaker versions of the assumption.

Formally, for a bilinear group $\mathbb{G} = (N, G, H, G_T, e, g, h)$ (where N can be either prime or composite) the uber-assumption is parameterized by five values: an integer $c \in \mathbb{N}$, three sets R, S, and T of polynomials over $\mathbb{Z}/N\mathbb{Z}$ (which represent the values we are given in G, H, and G_T respectively), and a polynomial f over $\mathbb{Z}/N\mathbb{Z}$. For the sets of polynomials, we write $R = \langle \rho_1(x_1, \ldots, x_c), \ldots, \rho_r(x_1, \ldots, x_c) \rangle$ and as shorthand use $\rho_i(\vec{x}) = \rho_i(x_1, \ldots, x_c)$ and $g^{R(x_1,\ldots,x_c)} = \{g^{\rho_i(\vec{x})}\}_{i=1}^{r}$ (and similarly for S and T).

Assumption 4.1 (Computational)**.** For an adversary $\mathcal{A}$, define $\mathbf{Adv}_{\mathcal{A}}^{\text{uber}}(\lambda) = \Pr[\text{c-UBER}_{c,R,S,T,f}^{\mathcal{A}}(\lambda)]$, where $\text{c-UBER}_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$ is defined as follows:

```
MAIN $\text{c-UBER}_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g \xleftarrow{\$} G, h \xleftarrow{\$} H$; $x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$y \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), g^{R(x_1,\ldots,x_c)}, h^{S(x_1,\ldots,x_c)}, e(g,h)^{T(x_1,\ldots,x_c)})$
return $(y = e(g,h)^{f(x_1,\ldots,x_c)})$
```

Then the uber-assumption holds if for all PT algorithms $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that $\mathbf{Adv}_{\mathcal{A}}^{\text{uber}}(\lambda) < \nu(\lambda)$. If instead the adversary must compute $y = g^{f(x_1,\ldots,x_c)}$, we call it the computational uber-assumption in the source group.

As an example, CDH in a symmetric group G uses $c = 2$, $R = S = \langle 1, x_1, x_2 \rangle$, $T = \langle 1 \rangle$, and $f(x_1, x_2) = x_1x_2$, so that given (g, g^{x_1}, g^{x_2}), it should be hard to compute $g^{x_1x_2}$. As a more complicated example, exponent q-SDH [48] in the target space G_T uses $c = 1$, $R = S = \langle 1 \rangle$, $T = \langle 1, \{x^i\}_{i=1}^{q} \rangle$, and $f(x) = x^{q+1}$, so that given $(e(g,h), e(g,h)^x, \ldots, e(g,h)^{x^q})$, it should be hard to compute $e(g,h)^{x^{q+1}}$. As long as R and S both include 1, the computational uber-assumption in the target group implies the computational uber-assumption in the source group, since given $X = g^{f(\vec{x})}$ one can always compute $e(X, h) = e(g, h)^{f(\vec{x})}$.

The game d-UBER$_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$ for the decisional uber-assumption is defined analogously, except rather than compute $e(g,h)^{f(x_1,\ldots,x_c)}$ at the end, the adversary has only to distinguish it from random.

Assumption 4.2 (Decisional)**.** For an adversary $\mathcal{A}$, define $\mathbf{Adv}_{\mathcal{A}}^{\text{uber}}(\lambda) = 2\Pr[\text{d-UBER}_{c,R,S,T,f}^{\mathcal{A}}(\lambda)] - 1$, where d-UBER$_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$ is defined as follows:

```
MAIN d-UBER$_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$
$b \xleftarrow{\$} \{0,1\}$; $(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g \xleftarrow{\$} G$, $h \xleftarrow{\$} H$; $x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
if $(b = 0)$ then $y \xleftarrow{\$} G$
if $(b = 1)$ then $y \leftarrow g^{f(x_1,\ldots,x_c)}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e, g, h), g^{R(x_1,\ldots,x_c)}, h^{S(x_1,\ldots,x_c)}, e(g,h)^{T(x_1,\ldots,x_c)}, y)$
return $(b' = b)$
```

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

Unlike the computational version, the decisional uber-assumption in the source group implies the decisional uber-assumption in the target group, since one can use a decider between $e(g,h)^{f(\vec{x})}$ and R_T to decide between $g^{f(\vec{x})}$ and R by computing the pairing. Furthermore, the decisional uber-assumption (in either group) implies the computational uber-assumption, since the ability to compute the target value immediately implies the ability to distinguish it from random. The strongest version of the uber-assumption, and the one we therefore choose to aim for in the next section, is the decisional assumption in either of the source groups.

4.2 A first approach: functions on one side of the pairing

Our first approach shows that certain classes of the uber-assumption are equivalent to significantly weaker classes, and that in fact these weaker classes are so weak that the assumption holds by a statistical argument. The subclass of uber-assumptions we cover includes q-type assumptions such as exponent q-SDH (defined above), and implies that any schemes that currently rely on such assumptions can be instantiated so that they rely solely on subgroup hiding.

Theorem 4.3. *For a bilinear group $\mathbb{G} = (N, G, H, G_T, e, G_1, G_2) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$, consider the decisional uber-assumption parameterized by c, $R = \langle 1, \rho_1(\vec{x}), \ldots, \rho_r(\vec{x}) \rangle$, $S = T = \langle 1 \rangle$, and $f(\vec{x})$. Then, if subgroup hiding holds in $\mathbb{G}$ with respect to $\mu = \{g_1, g_2\}$ and parameter hiding holds with respect to $R \cup \{f\}$, this assumption is implied by the decisional uber-assumption parameterized by ℓc, $R' = \langle \sum_{i=1}^{\ell} r_i, \sum_{i=1}^{\ell} r_i\rho_1(\vec{x}_i), \ldots, \sum_{i=1}^{\ell} r_i\rho_r(\vec{x}_i) \rangle$, S, T, and $f' = \sum_{i=1}^{\ell} r_i f(\vec{x}_i)$ for all $\ell = poly(\lambda)$ and for $r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$.* [1]

[1] In fact, the proof for this theorem can easily be extended to the case that adaptive parameter hiding holds for $R \cup \{f\}$.

A proof of this theorem can be found in Appendix B, and also applies when $R = S = \langle 1 \rangle$ and only T contains meaningful functions, or more generally in the case when there might not be an efficiently computable pairing (i.e., in the case when we use the more general $\mathsf{GroupGen}$, provided subgroup hiding holds for $\mu = g_1$ and for $\mu = g_2$).This theorem also trivially generalizes to the case when $T \neq \langle 1 \rangle$, as long as parameter hiding holds with respect to $R \cup T \cup \{f\}$. We are not aware of any concrete instantiations of the uber-assumption that have meaningful R and T though, so we focus on the above case for the sake of a cleaner exposition.

Intuitively, the transitions rely on the same modified dual-system technique that we used in the proof of Theorem 3.2. First, we switch to a setting where all elements exist only in the G_1 subgroup, operating over the original set of variables $\vec{x}_1$. A shadow copy of these elements is then added into the G_2 subgroup, which goes unnoticed by subgroup hiding. This shadow copy is then switched to operate over a new set of variables $\vec{x}_2$, which is identical by parameter hiding. These new values are then folded back into the G_1 subgroup, which is again indistinguishable by subgroup hiding. Finally, the G_2 component is eliminated, which is once again indistinguishable by subgroup hiding. The result is now a G_1 component that operates over both $\vec{x}_1$ and $\vec{x}_2$ (which can be shifted back into the full group G using another application of subgroup hiding), and the effect is analogous to the extra degree of randomness we obtain in the proof of Theorem 3.2. Repeating this process $\ell - 1$ more times proves the theorem.

To now show why this theorem is useful, we illustrate that the resulting game is often statistically hard, and thus the original uber-assumption is implied solely by subgroup hiding. To start, consider

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

We then have the following lemma, which relates the linear independence of the polynomials with the invertibility of the matrix:

Lemma 4.4. *For all $\lambda \in \mathbb{N}$, if the functions in $R \cup \{f\}$ are linearly independent and of maximum degree poly(λ), $\ell = q + 2$ for $q = poly(\lambda)$, and $N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$, then with all but negligible probability the matrix V is invertible.*

Proof. If the matrix V is invertible in $\mathbb{Z}/p_i\mathbb{Z}$ for each prime $p_i \mid N$, then it is also invertible in $\mathbb{Z}/N\mathbb{Z}$. To see that V is invertible (with all but negligible probability) in $\mathbb{Z}/p_i\mathbb{Z}$ for all i, define $F = \mathbb{Z}/p_i\mathbb{Z}$ (or, in the case that N is itself prime, define $F = \mathbb{Z}/N\mathbb{Z}$); then V is a matrix over F, where $|F|$ is exponential in λ. If we consider V instead as a matrix over the polynomial ring $F[x_{1,1}, \ldots, x_{1,c}, \ldots, x_{q+2,c}]$, then we can define its determinant to be the polynomial $D(\vec{x}_1, \ldots, \vec{x}_{q+2})$. By the definition of polynomial linear independence, the columns of V are linearly independent, so D is not the zero polynomial.

To consider the linear independence of the matrix over F, we must consider an assignment of concrete values $\vec{a}_1, \ldots, \vec{a}_{q+2}$ for the variables $\vec{x}_1, \ldots, \vec{x}_{q+2}$. To see that $D(\vec{a}_1, \ldots, \vec{a}_{q+2}) \neq 0$ with all but negligible probability — and thus the matrix V is invertible — consider $d = \max_{i=0}^{q}(d_i)$, where $d_0 = \deg(f)$ and $d_i = \deg(\rho_i)$ for all $\rho_i \in R$; then $\deg(D) \leq (q+1)d$. By the Schwartz-Zippel lemma, $\Pr[D(\vec{a}_1, \ldots, \vec{a}_{q+2}) = 0] \leq (q+1)d/|F|$ for $\vec{a}_1, \ldots, \vec{a}_{q+2} \xleftarrow{\$} F$. As $|F|$ is exponential in λ and both q and d are polynomial in λ, the probability is bounded by a negligible function in λ. □

We then have the following corollary, which indicates when we can show that the original decisional assumption is implied by subgroup hiding.

Corollary 4.5. *The decisional uber-assumption parameterized by (c, R, S, T, f) holds with all but negligible probability if (1) subgroup hiding holds in $\mathbb{G}$ with respect to $\mu = \{g_1, g_2\}$, where $\mathbb{G}$ is of order N for $N = p_1 \cdot \ldots \cdot p_n$ for distinct primes $p_1, \ldots, p_n \in \Omega(2^{poly(\lambda)})$, (2) parameter hiding holds with respect to $R \cup \{f\}$, (3) $S = T = \langle 1 \rangle$, and (4) the polynomials in $R \cup \{f\}$ are linearly independent and of maximum degree $poly(\lambda)$.*

Proof. By requirements (1), (2), and (3), Theorem 4.3 tells us that the (c, R, S, T, f)-uber assumption is equivalent to the $(\ell c, R', S, T, f')$-uber-assumption. In this latter assumption, the adversary sees values with exponents of the form $\vec{y} = \vec{r} \cdot V$, where $\vec{r}$ is a random vector of length ℓ and V is the $\ell \times (q+2)$ matrix defined in Equation 1. If we use $\ell = q + 2$, then by requirement (4), Lemma 4.4 tells us that V is invertible with all but negligible probability.

We can now use a bijection argument similar to the one in the proof of Theorem 3.2: $\vec{r}$ and $\vec{y}$ are both members of the set $\mathcal{S}$ containing all sets of size $q+2$ over $\mathbb{Z}/N\mathbb{Z}$, so multiplication by V maps $\mathcal{S}$ to itself. As V is invertible, the map is invertible as well, and is thus a permutation over $\mathcal{S}$. Sampling $\vec{r}$ uniformly at random and then multiplying by V thus yields a vector $\vec{y}$ that is distributed uniformly at random over $\mathbb{Z}/N\mathbb{Z}$.

When V is invertible, an adversary $\mathcal{A}$ thus has no advantage in distinguishing between $\vec{y}$ and a uniformly random vector in $\mathcal{S}$, as the distributions over the two are identical, and thus has only negligible overall advantage in $\text{d-UBER}^{\mathcal{A}}_{\ell c, R', S, T, f'}(\lambda)$. □

As observed by Boneh, Boyen, and Goh, if f is not linearly independent from all polynomials in $R \cup T$, then the assumption becomes trivially false. It furthermore unnecessarily expands the size of the tuple to use polynomials in R or T that are linearly dependent, as, e.g., g^{2x} is redundant given g^x. We therefore believe that the requirement that the polynomials in $R \cup T \cup \{f\}$ be linearly independent is not restrictive, and in fact — to the best of our knowledge — it is satisfied by all existing instantiations of the uber-assumption.

As a concrete example, we examine the exponent q-SDH assumption.

Example 4.6. For exponent q-SDH, $R = \langle 1, \alpha, \alpha^2, \ldots, \alpha^q \rangle$ and $f(\alpha) = \alpha^{q+1}$. Plugging these values into the matrix V gives

$$V = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^q & \alpha^{q+1} \\ 1 & \gamma_2 & \gamma_2^2 & \cdots & \gamma_2^q & \gamma_2^{q+1} \\ 1 & \gamma_3 & \gamma_3^2 & \cdots & \gamma_3^q & \gamma_2^{q+1} \\ & & \ddots & & & \\ \vdots & \vdots & & & \vdots & \vdots \\ & & & \ddots & & \\ 1 & \gamma_{q+2} & \gamma_{q+2}^2 & \cdots & \gamma_{q+2}^q & \gamma_{q+2}^{q+1} \end{bmatrix}$$

This is a Vandermonde matrix, which is invertible. By Corollary 4.5, exponent q-SDH is thus implied by subgroup hiding (with $\mu = g_1$ and $\mu = g_2$), assuming parameter hiding holds with respect to the set $\{f_k(\alpha) = \alpha^k\}_{k=0}^{q+1}$ (which, given our discussion in Section 5, currently restricts us to composite-order groups).

4.3 A second approach: computational assumptions in the source group

Although our results in the previous section have potentially broad implications, the requirements for Theorem 4.3 — and in particular the requirement that $S = \langle 1 \rangle$ — are somewhat restrictive, as many q-type assumptions require meaningful functions on both sides of the pairing. We furthermore do not seem able to relax this requirement using our current proof strategy: briefly, the fact that we need subgroup hiding between both G_1 and $G_1 \times G_2$ and between $G_1 \times G_2$ and G_2 means that we cannot give out the subgroup generators h_1 and h_2 on the other side of the pairing. To get around this restriction and allow meaningful functions on both sides of the pairing, we now consider an alternate approach in which we require subgroup hiding only between G_1 and $G_1 \times G_2$, which allows us to give out h_1.

Now, however, we may be giving meaningful information about the variables $\vec{x}$ in the group H as well as in G. This means we will need to extend our definition of parameter hiding to allow for the case where some additional information about $\vec{x}$ is revealed.

Definition 4.7 (Extended parameter hiding)**.** *For a group* $(N, G, G_1, G_2, \mu) \in [\mathsf{GroupGen}(1^\lambda, 2)]$, extended parameter hiding *holds with respect to a family of functions* $\mathcal{F}$ *and auxiliary information defined by* Aux *if for all* $g_1 \in G_1$ *and* $g_2 \in G_2$*, the distribution* $\{g_1^{f(x_1,\ldots,x_n)} g_2^{f(x_1,\ldots,x_n)}, a(x_1,\ldots,x_n)\}_{f\in\mathcal{F},a\in\mathsf{Aux}}$ *is identical to* $\{g_1^{f(x_1,\ldots,x_n)} g_2^{f(x_1',\ldots,x_n')}, a(x_1,\ldots,x_n)\}_{f\in\mathcal{F},a\in\mathsf{Aux}}$ *for* $x_1, x_1' \ldots, x_n, x_n' \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$.

For our purposes, we need this to hold for $\mathsf{Aux} = \{h_1^{f(\cdot)}\}_{f\in S\cup T}$, where h_1 is a generator of H_1 and S and T are the sets of polynomial functions specified by the uber-assumption (note that $h_1^{\tau(\vec{x})}$ is sufficient to compute $e(g_1, h_1)^{\tau(\vec{x})}$). Fortunately, we can — and do, in Lemma 5.2 — show that this holds in certain bilinear groups.

Theorem 4.8. *For a bilinear group* $\mathbb{G} = (N, G, H, G_T, e) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$*, consider the computational uber-assumption in the source group parameterized by* c*,* $R = \langle 1, \rho_1(\vec{x}), \ldots, \rho_r(\vec{x})\rangle$*,* S*,* T*, and* f*. Then, if subgroup hiding holds in* $\mathbb{G}$ *with respect to* $\mu = \{g_1, g_2, h_1\}$ *and 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 any* $h_1 \in H_1$*, this is implied by the following assumption for all* $\ell = poly(\lambda)$*: given*

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

for $g_1 \xleftarrow{\$} G_1, g_2 \xleftarrow{\$} G_2$*, and* $\vec{x}, r_1, \vec{x}_1, \ldots, r_\ell, \vec{x}_\ell \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$*, it is difficult to compute* $g_1^{f(\vec{x})} g_2^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)}$*.*[2]

A proof of this theorem can be found in Appendix C. Intuitively, the starting point is the same as in our previous proofs: all elements are first shifted to a setting in which they exist only in the G_1 subgroup, operating over the original set of variables $\vec{x}$, and a shadow copy of these elements is added into the G_2 subgroup, which goes unnoticed by subgroup hiding. This shadow copy is then switched to operate over a new set of variables $\vec{x}_1$, which is identical by parameter hiding. Now, rather than attempt to move these new variables back into G_1, we simply repeat the process of adding and re-randomizing the original set of variables into the G_2 subgroup, until we end up with ℓ sets of variables there.

Once again, the usefulness of this theorem is revealed only when we examine what this more complex assumption provides. Interestingly, it is not clear how to show that the decisional assumption holds by a statistical argument, as the isolation of the $\vec{x}$ variables in the G_1 subgroup provides a potentially detectable distribution. Instead, we restrict our attention to computational assumptions in the source group, in which the adversary is required to compute $g_1^{f(\vec{x})} g_2^{\sum_{i=1}^{\ell} r_i f(\vec{x}_i)}$ rather than distinguish it from random. In this setting, we have the following corollary, analogous to Corollary 4.5.

Corollary 4.9. *The computational uber-assumption parameterized by* (c, R, S, T, f) *holds in the source group with all but negligible probability if (1) subgroup hiding holds in* $\mathbb{G}$ *with respect to* $\mu = \{g_1, g_2, h_1\}$*, (2) 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 any* $h_1 \in H_1$*, (3) the polynomials in* $R \cup \{f\}$ *are linearly independent and have maximum degree poly*(λ)*.*

[2] Again, the proof for this theorem can easily be extended to the case that adaptive parameter hiding holds for $R \cup \{f\}$.

Proof. Let $\rho_0 = 1$. By requirements (1) and (2), Theorem 4.8 tells us that the original assumption is equivalent to one in which the adversary is given $\{g_1^{\rho_k(\vec{x})} g_2^{\sum_{i=1}^{q+2} r_i \rho_k(\vec{x}_i)}\}_{k=0}^{r}$, $h_1^S = \{h_1^{\sigma_m(\vec{x})}\}_{m=1}^{s}$, and $e(g_1, h_1)^T = \{e(g_1, h_1)^{\tau_j(\vec{x})}\}_{j=1}^{t}$, and is asked to compute $g_1^{f(\vec{x})} g_2^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$. Assume, for the sake of simplicity, that $g_1, \vec{x}$ are public, so $\mathcal{A}$ can compute the G_1 component of this target, and all it needs to compute is $g_2^{\sum_{i=1}^{q+2} r_i f(\vec{x}_i)}$. Immediately, it is clear that h_1^S and $e(g_1, h_1)^T$ provide no advantage, as they operate in different groups over a completely independent set of variables (namely, $\vec{x}$ as opposed to $\vec{x}_i$ for $i = 1 \dots q+2$).

In the G_2 subgroup, if we use $\ell = q+2$ and define $\vec{y} = \vec{r} \cdot V$ — where $\vec{r}$ is a random vector of length $q+2$ and V is the matrix defined in Equation 1 — then $\mathcal{A}$ is given the first $q+1$ entries of $\vec{y}$ and is asked to compute the last. By requirement (3), Lemma 4.4 tells us that V is invertible with all but negligible probability. We can now apply an analysis similar to the proof of Corollary 4.5: by the same bijection argument (i.e., the argument that if V is invertible then it is a permutation over vectors of length $q+2$ over $\mathbb{Z}/N\mathbb{Z}$), the fact that $\vec{r}$ is distributed uniformly at random means that the vector $\vec{y}$ is distributed uniformly at random as well; in particular, the distribution over both the values that $\mathcal{A}$ is given and the target value that it is trying to compute is uniformly random. $\mathcal{A}$ therefore has at most negligible probability in computing this target value. □

To bring everything together, we examine the q-SDH assumption, as defined by Boneh and Boyen [8].

Example 4.10. The q-SDH assumption uses $R = \langle 1, \alpha, \dots, \alpha^q \rangle$, $S = \langle 1, \alpha \rangle$, $T = \langle 1 \rangle$, and asks $\mathcal{A}$ to compute $(c, u^{\frac{1}{\alpha+c}})$. Using Theorem 4.8,[3] this is equivalent (under subgroup and parameter hiding) to an assumption in which $\mathcal{A}$ is given $(u_1 g_2^{\sum_{i=1}^{q+2} r_i}, u_1^{\alpha} g_2^{\sum_{i=1}^{q+2} r_i \gamma_i}, \dots, u_1^{\alpha^q} g_2^{\sum_{i=1}^{q+2} r_i \gamma_i^q}, v_1, v_1^{\alpha})$, where $\gamma_1, \dots, \gamma_{q+2} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, and is asked to compute $(c, u_1^{\frac{1}{\alpha+c}} g_2^{\sum_i \frac{r_i}{\gamma_i+c}})$. Applying the same analysis as above, we can ignore G_1 and focus on G_2, in which we use the matrix

$$A = \begin{bmatrix} 1 & \gamma_1 & \cdots & \gamma_1^q & \frac{1}{\gamma_1+c} \\ 1 & \gamma_2 & \cdots & \gamma_2^q & \frac{1}{\gamma_2+c} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \gamma_\ell & \cdots & \gamma_\ell^q & \frac{1}{\gamma_\ell+c} \end{bmatrix}$$

First, note that for any c such that $\gcd(\gamma_i + c, N) = 1$ for all i, this matrix is invertible. To see this, consider the additional matrices

$$B = \begin{bmatrix} \gamma_1 + c & \gamma_1(\gamma_1 + c) & \cdots & \gamma_1^q(\gamma_1 + c) & 1 \\ \gamma_2 + c & \gamma_2(\gamma_2 + c) & \cdots & \gamma_2^q(\gamma_2 + c) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_\ell + c & \gamma_\ell(\gamma_\ell + c) & \cdots & \gamma_\ell^q(\gamma_\ell + c) & 1 \end{bmatrix}, \quad C = \begin{bmatrix} \gamma_1 & \gamma_1(\gamma_1 + c) & \cdots & \gamma_1^q(\gamma_1 + c) & 1 \\ \gamma_2 & \gamma_2(\gamma_2 + c) & \cdots & \gamma_2^q(\gamma_2 + c) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_\ell & \gamma_\ell(\gamma_\ell + c) & \cdots & \gamma_\ell^q(\gamma_\ell + c) & 1 \end{bmatrix}, \text{ and}$$

$$D = \begin{bmatrix} \gamma_1 & \gamma_1(\gamma_1) & \cdots & \gamma_1^q(\gamma_1) & 1 \\ \gamma_2 & \gamma_2(\gamma_2) & \cdots & \gamma_2^q(\gamma) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_\ell & \gamma_\ell(\gamma_\ell) & \cdots & \gamma_\ell^q(\gamma_\ell) & 1 \end{bmatrix}$$

Each of these matrices is obtained from the previous one by elementary row or column operations, so if D is invertible then A is as well. Since D is a Vandermonde matrix with shifted columns, it is invertible

[3] Technically, this assumption doesn't meet the requirements of the theorem, as $\mathcal{A}$ produces a new value c rather than a function $f(\vec{x})$. The proof of the theorem can, however, be trivially extended to support assumptions of this type as well, as long as the group satisfies adaptive parameter hiding.

with all but negligible probability (following the argument in Lemma 4.4), so A is invertible with all but negligible probability.

Now, the game chooses $\vec{r} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, and $\mathcal{A}$ is given the first $q+1$ entries of $\vec{r} \cdot A$; these values $y_1, \ldots, y_{q+1}$ are uniformly distributed by the same bijection argument as in Corollary 4.9. The adversary then chooses c, which defines a permutation (assuming $\gcd(c + \gamma_i, N) = 1$ for all i), and then must produce $\hat{y}_{q+2}$. The adversary wins only if $y_1, \ldots, y_{q+1}, \hat{y}_{q+2}$ is consistent with the original $\vec{r}$. However, as multiplication with A is a permutation, every value $\hat{y}_{q+2}$ corresponds to some vector $\vec{r}'$, and the probability that the adversary chooses a value consistent with the original $\vec{r}$ is then $1/N$.

Finally, note that $y_1, \ldots, y_{q+1}$ are uniformly distributed and independent of $\gamma_1, \ldots, \gamma_{q+2}$, so the probability that $\mathcal{A}$ produces c such that $\gcd(\gamma_i + c, N) = 1$ for some i is at most $(q+2)(p_1 + p_2 - 1)/N$, which is negligible as long as p_1 and p_2 are exponential.

Putting this together, we get that $\mathcal{A}$ can produce the correct value with at most negligible probability, which implies that for these groups q-SDH is implied by subgroup hiding (with appropriate μ).

5 Instantiating Our Results

Abstractly, our results provide quite a strong guarantee: as long as subgroup hiding and parameter hiding hold, many instantiations of the uber-assumption hold (as well as non-uber-assumptions, such as q-SDH), as they reduce to assumptions that hold by a statistical argument. Concretely, we need to examine which groups support these underlying assumptions.

Parameter hiding. Our strongest requirement in our analysis was the generality of parameter hiding: to reason about any q-type assumption, we need a group where parameter hiding holds for all rational functions. While this seems hard to achieve in general, it does hold for any composite-order group (e.g., any group of order $N = p_1 p_2$ for primes p_1 and p_2).

Lemma 5.1. *For all groups G of order $N = p_1 p_2$ with subgroups $G_1 = \langle g_1 \rangle$ and $G_2 = \langle g_2 \rangle$, all $c \in \mathbb{N}$, if $\mathcal{F}$ is the class of all polynomial functions $f(\cdot)$ over $\mathbb{Z}/N\mathbb{Z}$, the distribution over $\{g_1^{f(x_1,\ldots,x_c)} g_2^{f(x_1,\ldots,x_c)}\}_{f \in \mathcal{F}}$ is identical to the distribution over $\{g_1^{f(x_1,\ldots,x_c)} g_2^{f(x_1',\ldots,x_c')}\}_{f \in \mathcal{F}}$ for $x_1, x_1', \ldots, x_c, x_c' \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$.*

Proof. This follows directly from the Chinese Remainder theorem. For a polynomial $p(\cdot)$, one can compute $g_2^{p(x_1,\ldots,x_c)}$ knowing just the values $x_i \bmod p_2$, which (by the Chinese Remainder Theorem) are independent of their values modulo p_1. □

Moreover, we can show that in the bilinear setting, extended parameter hiding holds when Aux is the set $\{h_1^{\sigma(\cdot)}\}_{\sigma \in S}$ for any $h_1 \in H_1$ and set of polynomials S.

Lemma 5.2. *For all groups $(N, G, H, G_T, e, g_1, g_2, h_1) \in [\mathsf{BilinearGen}(1^\lambda, 2)]$ where $N = p_1 p_2$, all $c \in \mathbb{N}$, and the class $\mathcal{F}$ of all polynomials $f(\cdot)$ over $\mathbb{Z}/N\mathbb{Z}$, the distribution over $\{g_1^{f(x_1,\ldots,x_c)} g_2^{f(x_1,\ldots,x_c)}, h_1^{f(x_1,\ldots,x_c)}\}_{f \in \mathcal{F}}$ is identical to the distribution over $\{g_1^{f(x_1,\ldots,x_c)} g_2^{f(x_1',\ldots,x_c')}, h_1^{f(x_1,\ldots,x_c)}\}_{f \in \mathcal{F}}$ for $x_1, x_1', \ldots, x_c, x_c' \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$.*

Proof. Again, this follows directly from the Chinese Remainder Theorem. The generator h_1 has order p_1, so the auxiliary information reveals information about only $\vec{x} \bmod p_1$, which is information-theoretically independent of the values modulo p_2. □

Subgroup hiding. In groups without a pairing — such as the target group of a bilinear tuple or a group over a non-pairing-friendly elliptic curve — subgroup decision is fairly straightforward. In groups with a pairing, however, the concerns mentioned in Section 2 (in which certain subgroup generators on the other side of the pairing could render subgroup decision easy) mean we have to be more careful. Our first approach in Section 4.2 relies on being unable to distinguish random elements of both G_1 and G_2 from $G_1 \times G_2$, even when given g_1 and g_2. This cannot hold, for example, in a symmetric bilinear group, so this assumption is reasonable only in the asymmetric setting. Our second approach in Section 4.3 requires that subgroup hiding holds even given h_1 and g_2, so it again requires an asymmetric pairing.

Instantiations. As mentioned above, our results in Sections 3 and 4 can be applied in any composite-order group where we can assume subgroup hiding. Reasonable candidates for such a group include composite-order elliptic curve groups without efficient pairings, the target group of a composite-order bilinear group, or composite-order subgroups of finite fields.

In the case where we do have a pairing, we need an asymmetric composite-order bilinear group in order to make subgroup hiding a reasonable assumption. Although most composite-order bilinear groups are symmetric (as they are groups of points on supersingular curves), ordinary composite-order curves were first introduced by Boneh, Rubin, and Silverberg [14], and their applicability for cryptography — and in particular an examination of the nature of the resulting asymmetric composite-order bilinear group — was very recently explored by Meiklejohn and Shacham [40].

Applications. In asymmetric composite-order bilinear groups we can prove a wide range of constructions secure based on just subgroup hiding. For example, our examination of q-SDH means that the Boneh-Boyen signature, the Boneh-Boyen-Shacham group signature [10], and the attribute-based signature due to Maji et al. [39] can all be proved secure under subgroup hiding, and the fact that q-DHI [41] — which states that given $(g, g^x, \dots, g^{x^q})$ it should be hard to compute $g^{1/x}$ — is also equivalent to subgroup hiding implies the Dodis-Yampolskiy VUF and the Jarecki-Liu PRF [28] can also both be proved secure based on subgroup hiding.

6 Conclusions and Open Problems

This paper demonstrated the applicability of the dual-system technique (and variants on it) by first proving the security of the Dodis-Yampolskiy PRF — using a domain of arbitrary size — under subgroup hiding, and then proving equivalence between many classes of the uber-assumption. This latter result further implies that many of these classes are in fact implied solely by subgroup hiding, as they reduce to assumptions that hold by a purely statistical argument. Our paper thus demonstrates that many common q-type assumptions — and the constructions that rely on them for security — can be implied directly by subgroup hiding when instantiated in the appropriate bilinear groups.

As our paper is a first step, many interesting directions and open problems remain. For example, we currently cannot prove anything about, e.g., decisional assumptions — such as q-DDHE — that require meaningful functions on both sides of the pairing. Perhaps the biggest open problem is obtaining more robust forms of parameter hiding in prime-order groups. Prime-order groups have the benefit of being significantly more efficient, and it is possible to construct groups with the appropriate subgroup hiding requirements using dual pairing vector spaces [42, 43], as exemplified most recently by Lewko and Meiklejohn [30].

For parameter hiding in prime-order bilinear groups, however, it is currently known how to obtain parameter hiding only for linear functions. Papers that have focused on translating these structural properties into prime-order settings, however, have indicated that they focus on such simple functions to keep their "constructions. . . simple and tailored to the requirements that [they] need" [30], so we consider constructing parameter hiding for more robust functions in the prime-order setting an interesting open problem rather than an impossibility.

Acknowledgments

We thank Michael Naehrig for his valuable feedback, and the anonymous reviewers for their helpful suggestions.

References

[1] M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *ASIACRYPT 2012*, LNCS, pages 4–24. Springer, Dec. 2012.

[2] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, Aug. 2010.

[3] M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, Aug. 2011.

[4] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, May 2010.

[5] M. H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In R. D. Prisco and M. Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 111–125. Springer, Sept. 2006.

[6] A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Jan. 2003.

[7] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Aug. 2004.

[8] 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. Springer, May 2004.

[9] 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. Springer, May 2005.

[10] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Aug. 2004.

[11] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Aug. 2001.

[12] 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. Springer, Feb. 2005.

[13] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Dec. 2001.

[14] 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.

[15] 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. Springer, Sept. 2008.

[16] X. Boyen and B. Waters. Compact group signatures without random oracles. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, May / June 2006.

[17] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer, May 2005.

[18] J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. In *2007 IEEE Symposium on Security and Privacy*, pages 101–115. IEEE Computer Society Press, May 2007.

[19] J. H. Cheon. Security analysis of the strong Diffie-Hellman problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11. Springer, May / June 2006.

[20] 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. Springer, Jan. 2005.

[21] M. Gerbush, A. Lewko, A. O'Neill, and B. Waters. Dual form signatures: an approach for proving security from static assumptions. In *Proceedings of Asiacrypt 2012*, pages 25–42, 2012.

[22] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *25th FOCS*, pages 464–479. IEEE Computer Society Press, Oct. 1984.

[23] J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Dec. 2006.

[24] J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, May / June 2006.

[25] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Apr. 2008.

[26] T. Jager and A. Rupp. The semi-generic group model and applications to pairing-based cryptography. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 539–556. Springer, Dec. 2010.

[27] D. Jao and K. Yoshida. Boneh-Boyen signatures and the strong Diffie-Hellman problem. In H. Shacham and B. Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 1–16. Springer, Aug. 2009.

[28] S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 577–594. Springer, Mar. 2009.

[29] M. Z. Lee, A. M. Dunn, J. Katz, B. Waters, and E. Witchel. Anon-pass: practical anonymous subscriptions. In *Proceedings of IEEE Symposium on Security and Privacy*, 2013.

[30] A. Lewko and S. Meiklejohn. A profitable sub-prime loan: Obtaining the advantages of composite-order in prime-order bilinear groups. Cryptology ePrint Archive, Report 2013/300, 2013. `http://eprint.iacr.org/2013/300`.

[31] 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. Springer, Apr. 2012.

[32] 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. Springer, May 2010.

[33] A. B. Lewko, Y. Rouselakis, and B. Waters. Achieving leakage resilience through dual system encryption. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 70–88. Springer, Mar. 2011.

[34] 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. Springer, Feb. 2010.

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

[36] 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. Springer, May 2011.

[37] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Aug. 2012.

[38] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485. Springer, May / June 2006.

[39] H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In A. Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 376–392. Springer, Feb. 2011.

[40] 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`.

[41] S. Mitsunari, R. Saka, and M. Kasahara. A new traitor tracing. *IEICE Transactions*, E85-A(2):481–484, Feb. 2002.

[42] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In S. D. Galbraith and K. G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 57–74. Springer, Sept. 2008.

[43] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, Dec. 2009.

```
$\underline{\textsc{main}\ \mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)\ /\ \boxed{\mathsf{G}_1^{\mathcal{A}}(\lambda)}}$
$(N, G, G_1, G_2, \mu) \xleftarrow{\$} \mathsf{GroupGen}(1^\lambda)$; $g \xleftarrow{\$} G$, $\boxed{u_1 \xleftarrow{\$} G_1,\ u_2 \xleftarrow{\$} G_2}$; $b \xleftarrow{\$} \{0,1\}$
$sk_1, r_1 \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\textsc{Fn}(\cdot)}(1^\lambda)$

$\underline{\text{Procedure } \textsc{Fn}(x)}$    // $\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_1^{\mathcal{A}}(\lambda)}$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
  if $\gcd(x + sk_1, N) \neq 1$ then $y \leftarrow 1$
  else $y \leftarrow g^{\frac{1}{sk_1 + x}}$, $\boxed{y \leftarrow u_1^{\frac{r_1}{sk_1 + x}}}$
```

Figure 1: Games for the proof of Theorem 3.2 (Equation 2). The boxed game uses the boxed code and the other game does not.

[44] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Aug. 2010.

[45] S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of Waters' dual system primitives using asymmetric pairings. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 298–315. Springer, May 2012.

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

[47] 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. Springer, Aug. 2009.

[48] F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In F. Bao, R. Deng, and J. Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 277–290. Springer, Mar. 2004.

A A Proof of Theorem 3.2

Proof. Let $\mathcal{A}$ be a PT adversary playing game $\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)$ that makes $q = q(\lambda)$ queries to its $\textsc{Fn}$ oracle. We provide PT adversaries $\mathcal{B}_0$ and $\mathcal{B}_{\mathsf{final}}$, a family of PT adversaries $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$, and $\mathcal{B}_{i,3}$, and negligible functions $\nu_2(\cdot)$ and $\nu(\cdot)$ such that

$$\begin{aligned}\mathbf{Adv}_{\mathsf{F},\mathcal{A}}^{\mathrm{prf}}(\lambda) \ \leq\ & 2(\mathbf{Adv}_{\mathcal{B}_0}^{\mathrm{sgh}}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{\mathsf{final}}}^{\mathrm{sgh}}(\lambda)) + (2q)(\mathbf{Adv}_{\mathcal{B}_{i,1}}^{\mathrm{sgh}}(\lambda) + \nu_2(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i,2}}^{\mathrm{sgh}}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i,3}}^{\mathrm{sgh}}(\lambda)) \\ & + (q^2 + q)\nu(\lambda)\end{aligned}$$

```
MAIN $\boxed{\mathsf{G}_i^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)}$
$sk_1, r_1, \ldots, sk_i, r_i, \boxed{sk_{i+1}, r_{i+1}} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$

Procedure Fn($x$)    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
if $b = 1$ then
    if $\gcd(x + sk_j, N) \neq 1$ for $j \in [i]$ then $y \leftarrow 1$
    else $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}}$, $\boxed{y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_i+x}}}$

Procedure Fn($x$)    // $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)}$
if $b = 1$ then
    if $\gcd(x + sk_j, N) \neq 1$ for $j \in [i]$, $\boxed{\text{for } j \in [i+1]}$ then $y \leftarrow 1$
    else $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} u_2^{\frac{r_i}{sk_i+x}}$, $\boxed{y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}}$

Procedure Fn($x$)    // $\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)}$
else $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}$, $\boxed{y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}}$

Procedure Fn($x$)    // $\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)}$
else $y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}$, $\boxed{y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}}}$
```

Figure 2: Games for the proof of Theorem 3.2 (Equations 3 through 6). The boxed games use the boxed code and the other games do not.

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{B}_0, \mathcal{B}_{\mathsf{final}}$, $\nu_2(\lambda)$, $\nu(\lambda)$, and $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$, and $\mathcal{B}_{i,3}$ for all i, $1 \leq i \leq q$, such that

$$\Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_1^{\mathcal{A}}(\lambda)] \leq \mathbf{Adv}_{\mathcal{B}_0}^{\mathrm{sgh}}(\lambda) \tag{2}$$

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

$$\Pr[\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)] \leq \nu_2(\lambda) \tag{4}$$

$$\Pr[\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)] \leq \mathbf{Adv}_{\mathcal{B}_{i,2}}^{\mathrm{sgh}}(\lambda) \tag{5}$$

$$\Pr[\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)] \leq \mathbf{Adv}_{\mathcal{B}_{i,3}}^{\mathrm{sgh}}(\lambda) \tag{6}$$

$$\Pr[\mathsf{G}_{q+1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)] \leq \mathbf{Adv}_{\mathcal{B}_{\mathsf{final}}}^{\mathrm{sgh}}(\lambda) \tag{7}$$

$$2\Pr[\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)] - 1 = (q^2 + q)\nu(\lambda). \tag{8}$$

```
MAIN $\mathsf{G}_{q+1}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)}$
$(N, G, G_1, G_2, \mu) \xleftarrow{\$} \mathsf{GroupGen}(1^\lambda)$; $u_1 \xleftarrow{\$} G_1$, $u_2 \xleftarrow{\$} G_2$; $\boxed{g \xleftarrow{\$} G}$, $b \xleftarrow{\$} \{0,1\}$
$sk_1, r_1, sk_{q+1}, r_{q+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\text{FN}(\cdot)}(1^\lambda)$

Procedure FN$(x)$    // $\mathsf{G}_{q+1}^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)}$
else $y \leftarrow u_1^{\sum_{j=1}^{q+1} \frac{r_j}{sk_j+x}}$, $\boxed{y \leftarrow g^{\sum_{j=1}^{q+1} \frac{r_j}{sk_j+x}}}$
```

Figure 3: Games for the proof of Theorem 3.2 (Equation 7). The boxed game uses the boxed code and the other game does not.

We then have, defining $\mathsf{G}_{i,0}^{\mathcal{A}}(\lambda) = \mathsf{G}_{i}^{\mathcal{A}}(\lambda)$ and $\mathsf{G}_{i,4}^{\mathcal{A}}(\lambda) = \mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$, that

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{F},\mathcal{A}}^{\mathrm{prf}}(\lambda) &= 2\Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)] - 1 \\
&= 2\left(\Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_1^{\mathcal{A}}(\lambda)]\right) + 2\left(\sum_{i=1}^{q}\sum_{j=0}^{3}(\Pr[\mathsf{G}_{i,j}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{i,j+1}^{\mathcal{A}}(\lambda)])\right) \\
&\quad + 2\left(\Pr[\mathsf{G}_{q+1}^{\mathcal{A}}(\lambda)] - \Pr[\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)]\right) + 2\Pr[\mathsf{G}_{\mathsf{final}}^{\mathcal{A}}(\lambda)] - 1 \\
&\le 2(\mathbf{Adv}_{\mathcal{B}_0}^{\mathrm{sgh}}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{\mathsf{final}}}^{\mathrm{sgh}}(\lambda)) + (2q)(\mathbf{Adv}_{\mathcal{B}_{i,1}}^{\mathrm{sgh}}(\lambda) + \nu_2(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i,2}}^{\mathrm{sgh}}(\lambda) + \mathbf{Adv}_{\mathcal{B}_{i,3}}^{\mathrm{sgh}}(\lambda)) \\
&\quad + (q^2 + q)\nu(\lambda).
\end{aligned}$$

Equation 2.

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

```
$\mathcal{B}_0(1^\lambda, N, G, G_1, T)$
$b \xleftarrow{\$} \{0,1\}$; $sk \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\text{SIMFN}(\cdot)}(1^\lambda)$
return $(b' \neq b)$

Procedure SIMFN$(x)$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
    if $\gcd(sk + x, N) \neq 1$ then $y \leftarrow 1$
    else $y \leftarrow T^{\frac{1}{sk+x}}$
return $y$
```

If $T \xleftarrow{\$} G$, then this is identical to the value in $\mathrm{PRF}_{\mathsf{F}}^{\mathcal{A}}(\lambda)$. If instead $T \xleftarrow{\$} G_1$, then this is identical to the value in $\mathsf{G}_1^{\mathcal{A}}(\lambda)$.

Equation 3.
$\mathcal{B}_{i,1}$ behaves as follows:

```
$\mathcal{B}_{i,1}(1^\lambda, N, G, G_1, u_1, T)$
$b \xleftarrow{\$} \{0,1\}$; $sk_1, r_1, \ldots, sk_i, r_i \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\textsc{SimFn}(\cdot)}(1^\lambda)$
return $(b' \neq b)$

Procedure $\textsc{SimFn}(x)$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
  if $\gcd(sk_j + x, N) \neq 1$ for $j \in [i]$ then $y \leftarrow 1$
  else $y \leftarrow u_1^{\sum_{j=1}^{i-1} \frac{r_j}{sk_j + x}} \cdot T^{\frac{1}{sk_i + x}}$
return $y$
```

If $T \xleftarrow{\$} G_1$, then $T = u_1^{r_i}$ for $r_i \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and $u_1 \in G_1$, so $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j + x}}$, which is identical to the value in $\mathsf{G}_i^{\mathcal{A}}(\lambda)$. If instead $T \xleftarrow{\$} G$, then $T = u_1^{r_i} u_2$ for $r_i \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, $u_1 \in G_1$, and $u_2 \xleftarrow{\$} G_2$, so $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j + x}} u_2^{\frac{1}{sk_i + x}}$, which is identical to the value in $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$.

Equation 4.
When $b = 0$, the two games are identical. When $b = 1$, consider the class of functions $\mathcal{F} = \{f_{sk}(x) = (x + sk)^{-1}\}_{sk \in \mathsf{F.Keys}}$ with domains $f_{sk}.\mathcal{D} = \{x \mid \gcd(sk + x, N) = 1\}$. For $r_1, sk_1, \ldots, r_{i-1}, sk_{i-1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and for $\mathcal{O}, \mathcal{O}'$ constructed as in Definition 2.3, consider an oracle that on input x outputs 1 if $\gcd(x + sk_j) = 1$ for some $j \in [i-1]$, and otherwise outputs the result of $\mathcal{O}$ or $\mathcal{O}'$. If we use $\mathcal{O}$, the result is identical to the $\textsc{Fn}$ oracle in $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$ with $b = 1$, and if we use $\mathcal{O}'$, it is identical to the $\textsc{Fn}$ oracle in $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$ with $b = 1$. This means that to prove the equation it suffices to show that this function family satisfies adaptive parameter hiding, which follows from Lemma 3.1.

Equation 5.
$\mathcal{B}_{i,2}$ behaves as follows:

```
$\mathcal{B}_{i,2}(1^\lambda, N, G, G_1, u_1, T)$
$b \xleftarrow{\$} \{0,1\}$; $sk_1, r_1, \ldots, sk_i, r_i, sk_{i+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\textsc{SimFn}(\cdot)}(1^\lambda)$
return $(b' \neq b)$

Procedure $\textsc{SimFn}(x)$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
  if $\gcd(sk_j + x, N) \neq 1$ for $j \in [i+1]$ then $y \leftarrow 1$
  else $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j + x}} \cdot T^{\frac{1}{sk_{i+1} + x}}$
return $y$
```

If $T \xleftarrow{\$} G_2$ then $T = u_2$ for $u_2 \xleftarrow{\$} G_2$, so $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}$, which is identical to the value in $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$. If instead $T \xleftarrow{\$} G$, then $T = u_1^{r_{i+1}} u_2$ for $r_{i+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and $u_2 \xleftarrow{\$} G_2$, so $y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}$, which is identical to the value in $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$.

Equation 6.
$\mathcal{B}_{i,3}$ behaves as follows:

```
$\mathcal{B}_{i,3}(1^\lambda, N, G, G_1, u_1, T)$
$b \xleftarrow{\$} \{0,1\}$; $sk_1, r_1, \ldots, sk_i, r_i, sk_{i+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\textsc{SimFn}(\cdot)}(1^\lambda)$
return $(b' \neq b)$

Procedure $\textsc{SimFn}(x)$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
   if $\gcd(sk_j + x, N) \neq 1$ for $j \in [i+1]$ then $y \leftarrow 1$
   else $y \leftarrow u_1^{\sum_{j=1}^{i} \frac{r_j}{sk_j+x}} \cdot T^{\frac{1}{sk_{i+1}+x}}$
return $y$
```

If $T \xleftarrow{\$} G$, then $T = u_1^{r_{i+1}} u_2$ for $u_1 \in G_1$, $r_{i+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, and $u_2 \xleftarrow{\$} G_2$, so $y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}} u_2^{\frac{1}{sk_{i+1}+x}}$, which is identical to the value in $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$. If instead $T \xleftarrow{\$} G_1$ then $T = u_1^{r_{i+1}}$ for $u_1 \in G_1$ and $r_{i+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, so $y \leftarrow u_1^{\sum_{j=1}^{i+1} \frac{r_j}{sk_j+x}}$, which is identical to the value in $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$.

Equation 7.
Finally, $\mathcal{B}_{\mathsf{final}}$ behaves as follows:

```
$\mathcal{B}_{\mathsf{final}}(1^\lambda, N, G, G_1, T)$
$b \xleftarrow{\$} \{0,1\}$; $sk_1, r_1, \ldots, sk_{q+1}, r_{q+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$b' \xleftarrow{\$} \mathcal{A}^{\textsc{SimFn}(\cdot)}(1^\lambda)$
return $(b' \neq b)$

Procedure $\textsc{SimFn}(x)$
if $b = 0$ then $y \xleftarrow{\$} G$
if $b = 1$ then
   if $\gcd(sk_j + x, N) \neq 1$ for $j \in [q+1]$ then $y \leftarrow 1$
   else $y \leftarrow T^{\sum_{j=1}^{q+1} \frac{r_j}{sk_j+x}}$
return $y$
```

If $T \xleftarrow{\$} G_1$, then this y is identical to the value in $\mathsf{G}^{\mathcal{A}}_{q+1}(\lambda)$. If instead $T \xleftarrow{\$} G$, then this y is identical to the value in $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$.

Equation 8.
We begin with a modified version of $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$: rather than pick $r_1, \ldots, r_{q+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$, pick $y_1, \ldots, y_{q+1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and set $r_j := y_j \prod_{k \neq j} \frac{1}{s_k - s_j}$ for all j, $1 \leq j \leq q+1$. (Note that as long as $\gcd(s_i - s_j, N) = 1$ for

all pairs s_i, s_j, this will be distributed identically to the original choice of $r_1, \ldots, r_{q+1} \mod p_1$. Thus, the adversary has less than $q^2 * (p_1 + p_2 - 1)/N$ advantage in distinguishing this game from $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$.) Then in the exponent (as long as $sk_i + x$ has an inverse in $\mathbb{Z}/N\mathbb{Z}$ for all i), we have the expression

$$\sum_{i=1}^{q+1} \frac{r_i}{sk_i + x} = \frac{\sum_{i=1}^{q+1} r_i \prod_{j \neq i}(sk_j + x)}{\prod_{i=1}^{q+1}(sk_i + x)} = \frac{\sum_{i=1}^{q+1} y_i \prod_{j \neq i} \frac{sk_j + x}{sk_j - sk_i}}{\prod_{i=1}^{q+1}(sk_i + x)}.$$

The numerator is the formula for the Lagrange interpolating polynomial through the points $(-sk_i, y_i)$; i.e., the polynomial $p(\cdot)$ such that $p(-sk_i) = y_i$ for all i, $1 \leq i \leq q+1$. As the y_i values are distributed uniformly at random, $p(\cdot)$ is therefore a degree-q polynomial with random coefficients.

To see this argument in more detail, consider the set S_1 of all polynomials of degree q over $\mathbb{Z}/N\mathbb{Z}$; then $|S_1| = N(q+1)$ (as a degree-q polynomial can be represented by its $q+1$ coefficients, each in $\mathbb{Z}/N\mathbb{Z}$). If we also consider the set S_2 of sets of size $q+1$ over $\mathbb{Z}/N\mathbb{Z}$, then we have $|S_2| = N(q+1)$. As we can compute a unique degree-q polynomial given a set of $q+1$ output points using interpolation, and can compute a unique set of $q+1$ output points given a polynomial by evaluating the polynomial, S_1 and S_2 are bijective. This means that sampling randomly from S_2 and then applying the bijection gives us a random sample in S_1, which in turn means that, because the set $\{y_i\}_{i=1}^{q+1}$ is distributed uniformly at random, the polynomial $p(\cdot)$ defined by applying Lagrange interpolation is a random polynomial.

We can therefore rewrite the function in the exponent as $\frac{p(x)}{\prod_{i=1}^{q+1}(sk_i+x)}$. Note that as long as $\gcd(sk_i + x, N) = 1$ for all i, then the response to each query x to the $\textsc{Fn}$ oracle is $p(x)/X$ for some value $X \neq 0$, which — because $p(\cdot)$ is a random degree-q polynomial that we see at most q outputs of — is distributed uniformly at random. Given this, we can also conclude that as long as $\gcd(sk_i + x, N) = 1$ for all i, the oracle responses reveal nothing else about $sk_1, \ldots, sk_{q+1}$. Thus, the probability that in q queries, the adversary finds at least one x for which $\gcd(sk_i + x, N) = 1$ for some $sk_i \in \{sk_1, \ldots, sk_{q+1}\}$ is at most the probability that he can find such an x by random guessing: $q(q+1)(p_1 + p_2 - 1)/N$.

Putting this together with the $(q+1)(p_1 + p_2 - 1)/N$ above, we conclude that the adversary's advantage of distinguishing $b = 0$ and $b = 1$ is $O(q(q+1)(\frac{1}{p_2} + \frac{1}{p_1}))$, which is negligible as long as p_1 and p_2 are exponential. □

B A Proof of Theorem 4.3

Proof. Let $\mathcal{A}$ be a PT adversary playing game d-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$. We provide PT adversaries $\mathcal{B}_0$ and $\mathcal{B}_{\mathsf{final}}$ and a family of PT adversaries $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$, and $\mathcal{B}_{i,3}$ such that

$$\begin{aligned}\mathbf{Adv}^{\text{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) \;\leq\; & 2(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_0}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{\mathsf{final}}}(\lambda)) + 2\ell(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,1}}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,2}}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,3}}(\lambda)) \\ & + \mathbf{Adv}^{\text{uber}}_{\ell c,R,S,T',f',\mathcal{A}}(\lambda)\end{aligned}$$

```
$\text{MAIN d-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ / $\boxed{\mathsf{G}^{\mathcal{A}}_1(\lambda)}$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g \xleftarrow{\$} G$, $\boxed{g_1 \xleftarrow{\$} G_1,\ g_2 \xleftarrow{\$} G_2}$; $b \xleftarrow{\$} \{0,1\}$
$x_1, \ldots, x_c, r_1 \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g^{\rho_k(x_1,\ldots,x_c)}$, $\boxed{v_k \leftarrow g_1^{r_1\rho_k(x_1,\ldots,x_c)}}$ $\forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g^{f(x_1,\ldots,x_c)}$, $\boxed{v' \leftarrow g_1^{f(x_1,\ldots,x_c)}}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

Figure 4: Games for the proof of Theorem 4.3 (Equation 9). The boxed game uses the boxed code and the other game does not.

for all $\lambda \in \mathbb{N}$, from which the theorem follows. To do this, we build $\mathcal{B}_0$, $\mathcal{B}_{\mathsf{final}}$, and $\mathcal{B}_{i,1}$, $\mathcal{B}_{i,2}$, and $\mathcal{B}_{i,3}$ for all i, $1 \le i \le \ell$, such that

$$|\Pr[\text{d-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_1(\lambda)]| \le \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_0}(\lambda) \tag{9}$$
$$|\Pr[\mathsf{G}^{\mathcal{A}}_i(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)]| \le \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,1}}(\lambda) \tag{10}$$
$$|\Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)]| = 0 \tag{11}$$
$$|\Pr[\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)]| \le \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,2}}(\lambda) \tag{12}$$
$$|\Pr[\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)]| \le \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,3}}(\lambda) \tag{13}$$
$$|\Pr[\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{\ell}(\lambda)]| \le \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{\mathsf{final}}}(\lambda) \tag{14}$$
$$|2\Pr[\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)] - 1| = \mathbf{Adv}^{\text{uber}}_{\ell c,R,S,T',f',\mathcal{A}}(\lambda). \tag{15}$$

We then have, defining $\mathsf{G}^{\mathcal{A}}_i(\lambda) = \mathsf{G}^{\mathcal{A}}_{i,0}(\lambda)$ and $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda) = \mathsf{G}^{\mathcal{A}}_{i,4}(\lambda)$, that

$$\begin{aligned}
\mathbf{Adv}^{\text{uber}}_{c,R,S,T,f,\mathcal{A}}(\lambda) &= |2\Pr[\text{d-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - 1| \\
&= |2(\Pr[\text{d-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_1(\lambda)]) + 2(\sum_{i=1}^{\ell-1}\sum_{j=0}^{3}(\Pr[\mathsf{G}^{\mathcal{A}}_{i,j}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,j+1}(\lambda)])) \\
&\quad + 2(\Pr[\mathsf{G}^{\mathcal{A}}_{\ell}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)]) + 2\Pr[\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)] - 1| \\
&\le 2(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_0}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{\mathsf{final}}}(\lambda)) + 2\ell(\mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,1}}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,2}}(\lambda) + \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_{i,3}}(\lambda)) \\
&\quad + \mathbf{Adv}^{\text{uber}}_{\ell c,R,S,T',f',\mathcal{A}}(\lambda).
\end{aligned}$$

Finally, the description of H allows us to sample $h \xleftarrow{\$} H$, which is enough to compute $h^{S(x_1,\ldots,x_c)} = h^1$ and $e(g,h)^{T(x_1,\ldots,x_c)} = e(g,h)^1$. We can thus ignore these values in the following games as long as we include the description of H.

We also for simplicity let $\rho_0 = 1$, so $R = \langle \rho_0, \ldots, \rho_q \rangle$, and define $[q]$ to mean $\{0, \ldots, q\}$.

Equation 9: d-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_1(\lambda)$

```
MAIN $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)}$ / $\boxed{\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)}$
$(N, G, H, G_T, e, G_1, G_2) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g_1 \xleftarrow{\$} G_1, g_2 \xleftarrow{\$} G_2; b \xleftarrow{\$} \{0,1\}$
$x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i,1}, \ldots, x_{i,c}, r_1, \ldots, r_i$ $\boxed{x_{i+1,1}, \ldots, x_{i+1,c}} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\sum_{j=1}^{i} \rho_k(x_{j,1},\ldots,x_{j,c})}$ $\forall k \in [q]$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i,1},\ldots,x_{i,c})}}$ $\forall k \in [q]$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$\boxed{v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})}}$ $\forall k \in [q]$    // $\boxed{\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)}$
$\boxed{v_k \leftarrow g_1^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})}}$ $\forall k \in [q]$    // $\boxed{\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)}$
$v_k \leftarrow g_1^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})}$ $\forall k \in [q]$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})}$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{\text{if } (b = 1) \text{ then } v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i,1},\ldots,x_{i,c})}}$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$\boxed{\text{if } (b = 1) \text{ then } v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i+1,1},\ldots,x_{i+1,c})}}$    // $\boxed{\mathsf{G}_{i,2}^{\mathcal{A}}(\lambda)}$
$\boxed{\text{if } (b = 1) \text{ then } v' \leftarrow g_1^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i+1,1},\ldots,x_{i+1,c})}}$    // $\boxed{\mathsf{G}_{i,3}^{\mathcal{A}}(\lambda)}$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\ldots,x_{j,c})}$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

Figure 5: Games for the proof of Theorem 4.3 (Equations 10 through 13). Each game uses the boxed code on its corresponding line.

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

```
$\mathcal{B}_0(1^\lambda, N, G, H, G_T, e, W)$
$b \xleftarrow{\$} \{0,1\}$; $x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow W^{\rho_k(x_1,\ldots,x_c)}$ $\forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow W^{f(x_1,\ldots,x_c)}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

If $W \xleftarrow{\$} G$ then this is identical to the value in d-UBER$_{c,R,S,T,f}^{\mathcal{A}}(\lambda)$. If instead $W \xleftarrow{\$} G_1$ then this is identical to the value in $\mathsf{G}_1^{\mathcal{A}}(\lambda)$.

```
MAIN $\mathsf{G}^{\mathcal{A}}_{\ell}(\lambda)$ / $\boxed{\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)}$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2)$; $g_1 \xleftarrow{\$} G_1$, $g_2 \xleftarrow{\$} G_2$, $\boxed{g \xleftarrow{\$} G}$; $b \xleftarrow{\$} \{0,1\}$
$x_{1,1}, \ldots, x_{1,c}, \ldots, x_{\ell,1}, \ldots, x_{\ell,c}, r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\sum_{j=1}^{\ell} r_j \rho_k(x_{j,1},\ldots,x_{j,c})}$, $\boxed{v_k \leftarrow g^{\sum_{j=1}^{\ell} r_j \rho_k(x_{j,1},\ldots,x_{j,c})}}$ $\forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{\ell} r_j f(x_{j,1},\ldots,x_{j,c})}$, $\boxed{v' \leftarrow g^{\sum_{j=1}^{\ell} r_j f(x_{j,1},\ldots,x_{j,c})}}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

Figure 6: Games for the proof of Theorem 4.3 (Equation 14). The boxed game uses the boxed code and the other game does not.

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

$\mathcal{B}_{i,1}$ behaves as follows:

```
$\mathcal{B}_{i,1}(1^\lambda, N, G, G_T, e, g_1, W)$
$b \xleftarrow{\$} \{0,1\}$; $x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i,1}, \ldots, x_{i,c}, r_1, \ldots, r_{i-1} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\sum_{j=1}^{i-1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \cdot W^{\rho_k(x_{i,1},\ldots,x_{i,c})}$ $\forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{i-1} r_j f(x_{j,1},\ldots,x_{j,c})} \cdot W^{f(x_{i,1},\ldots,x_{i,c})}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

If $W \xleftarrow{\$} G_1$, then $W = g_1^{r_i}$ for uniformly distributed $r_i \in \mathbb{Z}/N\mathbb{Z}$, and

$$v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i}(\lambda)$. If instead $W \xleftarrow{\$} G$, then $W = g_1^{r_i} g_2$ for uniformly distributed $r_i \in \mathbb{Z}/N\mathbb{Z}$ and $g_2 \in G_2$, and

$$v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i,1},\ldots,x_{i,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i,1},\ldots,x_{i,c})},$$

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

Equation 11: $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$

If we define $A := g_1^{\sum_{j=1}^{i-1} r_j \rho_k(\vec{x}_j)}$, then A is independent from anything involving $\vec{x}_i$ or $\vec{x}_{i+1}$, as it operates over completely independent sets of variables. The switch from $\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$ is from $A \cdot (\hat{g_1}^{\rho_k(\vec{x}_i)} g_2^{\rho_k(\vec{x}_i)})$ to $A \cdot (\hat{g_1}^{\rho_k(\vec{x}_i)} g_2^{\rho_k(\vec{x}_{i+1})})$ (and similarly for f), where $\hat{g_1} = g_1^{r_i}$ is a uniformly random element of G_1. Since parameter hiding is assumed to hold with respect to $R \cup f$, the distributions over these values are identical.

Equation 12: $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$
$\mathcal{B}_{i,2}$ behaves as follows:

```
$\underline{\mathcal{B}_{i,2}(1^\lambda, N, G, G_T, e, g_1, W)}$
$b \xleftarrow{\$} \{0,1\}$; $x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i+1,1}, \ldots, x_{i+1,c}, r_1, \ldots, r_i \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \cdot W^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})}\ \forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} \cdot W^{f(x_{i+1,1}\cdots,x_{i+1,c})}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

If $W = g_2 \xleftarrow{\$} G_2$, then

$$v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i+1,1},\ldots,x_{i+1,c})},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i,2}(\lambda)$. If instead $W \xleftarrow{\$} G$, then $W = g_1^{r_{i+1}} g_2$ for uniformly distributed $r_{i+1} \in \mathbb{Z}/N\mathbb{Z}$ and $g_2 \in G_2$, and

$$v_k \leftarrow g_1^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i+1,1}\cdots,x_{i+1,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i+1,1}\cdots,x_{i+1,c})},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$.

Equation 13: $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)$
$\mathcal{B}_{i,3}$ behaves as follows:

```
$\underline{\mathcal{B}_{i,3}(1^\lambda, N, G, G_T, e, g_1, W)}$
$b \xleftarrow{\$} \{0,1\}$; $x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i,1}, \ldots, x_{i,c}, r_1, \ldots r_i \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \cdot W^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})}\ \forall k \in [q]$
if $(b = 0)$ then $v' \xleftarrow{\$} G$
if $(b = 1)$ then $v' \leftarrow g_1^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})} \cdot W^{f(x_{i+1,1}\cdots,x_{i+1,c})})$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, v')$
return $(b' = b)$
```

If $W \xleftarrow{\$} G$, then $W = g_1^{r_{i+1}} g_2$ for uniformly distributed $r_{i+1} \in \mathbb{Z}/N\mathbb{Z}$ and $g_2 \in G_2$, and

$$v_k \leftarrow g_1^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} g_2^{\rho_k(x_{i+1,1},\ldots,x_{i+1,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\ldots,x_{j,c})} g_2^{f(x_{i+1,1},\ldots,x_{i+1,c})},$$

which are identical to the values in $\mathsf{G}^{\mathcal{A}}_{i,3}(\lambda)$. If instead $W \xleftarrow{\$} G_1$, then $W = g_1^{r_{i+1}}$ for uniformly distributed $r_{i+1} \in \mathbb{Z}/N\mathbb{Z}$, and

$$v_k \leftarrow g_1^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \quad \text{and} \quad v' \leftarrow g_1^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\ldots,x_{j,c})},$$

```
MAIN c-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ / $\boxed{\mathsf{G}^{\mathcal{A}}_0(\lambda)}$
$(N,G,H,G_T,e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda,2)$; $g \xleftarrow{\$} G$, $\boxed{g_1 \xleftarrow{\$} G_1,\ g_2 \xleftarrow{\$} G_2}$, $h \xleftarrow{\$} H$
$x_1,\ldots,x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g^{\rho_k(x_1,\ldots,x_c)}$, $\boxed{v_k \leftarrow g_1^{\rho_k(x_1,\ldots,x_c)}}$ $\forall k \in [q]$
$y_k \leftarrow h^{\sigma_k(x_1,\ldots,x_c)}$ $\forall k \in [s]$
$z_k \leftarrow e(g,h)^{\tau_k(x_1,\ldots,x_k)}$, $\boxed{z_k \leftarrow e(g_1,h)^{\tau_k(x_1,\ldots,x_c)}}$ $\forall k \in [t]$
$v' \leftarrow g^{f(x_1,\ldots,x_c)}$, $\boxed{v' \leftarrow g_1^{f(x_1,\ldots,x_c)}}$
$v'' \xleftarrow{\$} \mathcal{A}(1^\lambda,(N,G,H,G_T,e),v_1,\ldots,v_q,y_1,\ldots,y_s,z_1,\ldots,z_t)$
return $(v'' = v')$
```

Figure 7: Games for the proof of Theorem 4.8 (Equation 16). The boxed game uses the boxed code and the other game does not.

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

Equation 14: $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_{\ell}(\lambda)$

Finally, $\mathcal{B}_{\mathsf{final}}$ behaves as follows:

```
$\mathcal{B}_{\mathsf{final}}(1^\lambda,N,G,G_T,e,W)$
$b \xleftarrow{\$} \{0,1\}$; $x_{1,1},\ldots,x_{1,c},\ldots,x_{\ell,1},\ldots,x_{\ell,c},r_1,\ldots,r_\ell \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow W^{\sum_{j=1}^{\ell} r_j\rho_k(x_{j,1},\ldots,x_{j,c})}$ $\forall k \in [q]$
if $(b=0)$ then $v' \xleftarrow{\$} G$
if $(b=1)$ then $v' \leftarrow W^{\sum_{j=1}^{\ell} r_j f(x_{j,1},\ldots,x_{j,c})}$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda,(N,G,H,G_T,e),v_1,\ldots,v_q,v')$
return $(b' = b)$
```

If $W \xleftarrow{\$} G_1$ then this is identical to the value in $\mathsf{G}^{\mathcal{A}}_{\ell}(\lambda)$. If instead $W \xleftarrow{\$} G$ then this is identical to the value in $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$.

Equation 15: $\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$ to d-UBER$^{\mathcal{A}}_{\ell c,R',S,T,f'}(\lambda)$

$\mathsf{G}^{\mathcal{A}}_{\mathsf{final}}(\lambda)$ gives out values of the form $v_k \leftarrow g^{\sum_{j=1}^{\ell} r_j\rho_k(x_{j,1},\ldots,x_{j,c})}$ for all k, $1 \le k \le q$, and checks at the end that $v' = g^{\sum_{j=1}^{\ell} f(x_{j,1},\ldots,x_{j,c})}$. As $\rho'_k(x_{1,1},\ldots,x_{1,c},\ldots,x_{\ell,c}) = \sum_{j=1}^{\ell} r_j\rho_k(x_{j,1},\ldots,x_{j,c})$ and $f'(x_{1,1},\ldots,x_{1,c},\ldots,x_{\ell,c}) = \sum_{j=1}^{\ell} r_j f(x_{j,1},\ldots,x_{j,c})$, this is exactly d-UBER$^{\mathcal{A}}_{\ell c,R',S,T,f'}(\lambda)$. □

C A Proof of Theorem 4.8

Proof. Let $\mathcal{A}$ be a PT adversary playing game c-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$, and let $\mathbf{Adv}^{\text{dual-sys}}_{\mathcal{A}}(\lambda)$ denote its advantage in the final game specified in the statement of Theorem 4.8. We provide PT adversaries $\mathcal{B}_0$

```
MAIN $\mathsf{G}_0^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_1^{\mathcal{A}}(\lambda)}$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2);\ g_1 \xleftarrow{\$} G_1,\ g_2 \xleftarrow{\$} G_2,\ h \xleftarrow{\$} H,\ \boxed{h_1 \xleftarrow{\$} H_1}$
$x_1, \dots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$y_k \leftarrow h^{\sigma_k(x_1,\dots,x_c)},\ \boxed{y_k \leftarrow h_1^{\sigma_k(x_1,\dots,x_c)}}\ \forall k \in [s]$
$z_k \leftarrow e(g_1, h)^{\tau_k(x_1,\dots,x_c)},\ \boxed{z_k \leftarrow e(g_1, h_1)^{\tau_k(x_1,\dots,x_c)}}\ \forall k \in [t]$
```

Figure 8: Games for the proof of Theorem 4.8 (Equation 17). The boxed game uses the boxed code and the other game does not.

```
MAIN $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ / $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$ / $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$(N, G, H, G_T, e) \xleftarrow{\$} \mathsf{BilinearGen}(1^\lambda, 2); g_1 \xleftarrow{\$} G_1, g_2 \xleftarrow{\$} G_2$
$x_1, \dots, x_c, x_{1,1}, \dots, x_{1,c}, \dots, x_{i,1}, \dots, x_{i,c}, r_1, \dots, r_i \boxed{, x_{i+1,1}, \dots, x_{i+1,c}, r_{i+1}} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(x_1,\dots,x_c)} g_2^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\dots,x_{j,c})}\ \forall k \in [q]$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{v_k \leftarrow g_1^{\rho_k(x_1,\dots,x_c)} g_2^{r_{i+1}\rho_k(x_1,\dots,x_c) + \sum_{j=1}^{i} r_j \rho_k(x_{j,1},\dots,x_{j,c})}}\ \forall k \in [q]$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$v_k \leftarrow g_1^{\rho_k(x_1,\dots,x_c)} g_2^{\sum_{j=1}^{i+1} r_j \rho_k(x_{j,1},\dots,x_{j,c})}\ \forall k \in [q]$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$v' \leftarrow g_1^{f(x_1,\dots,x_c)} g_2^{\sum_{j=1}^{i} r_j f(x_{j,1},\dots,x_{j,c})}$    // $\mathsf{G}_i^{\mathcal{A}}(\lambda)$
$\boxed{v' \leftarrow g_1^{f(x_1,\dots,x_c)} g_2^{r_{i+1} f(x_1,\dots,x_c) + \sum_{j=1}^{i} r_j f(x_{j,1},\dots,x_{j,c})}}$    // $\boxed{\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)}$
$v' \leftarrow g_1^{f(x_1,\dots,x_c)} g_2^{\sum_{j=1}^{i+1} r_j f(x_{j,1},\dots,x_{j,c})}$    // $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$
$v'' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \dots, v_q)$
return $(v' = v'')$
```

Figure 9: Games for the proof of Theorem 4.8 (Equations 18 and 19). Each game uses the boxed code on its corresponding line.

and $\mathcal{C}_0$, and a family of PT adversaries $\mathcal{B}_i$ such that

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

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

$$\begin{align}
\Pr[\text{c-UBER}^{\mathcal{A}}_{c,R,S,T,f}(\lambda) - \Pr[\mathsf{G}^{\mathcal{A}}_0(\lambda)] &\leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_0}(\lambda) \tag{16}\\
\Pr[\mathsf{G}^{\mathcal{A}}_0(\lambda) - \Pr[\mathsf{G}^{\mathcal{A}}_1(\lambda)] &\leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{C}_0}(\lambda) \tag{17}\\
\Pr[\mathsf{G}^{\mathcal{A}}_i(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] &\leq \mathbf{Adv}^{\text{sgh}}_{\mathcal{B}_i}(\lambda) \tag{18}\\
\Pr[\mathsf{G}^{\mathcal{A}}_{i,1}(\lambda)] - \Pr[\mathsf{G}^{\mathcal{A}}_{i+1}(\lambda)] &= 0 \tag{19}\\
\Pr[\mathsf{G}^{\mathcal{A}}_\ell(\lambda)] &= \mathbf{Adv}^{\text{dual-sys}}_{\mathcal{A}}(\lambda). \tag{20}
\end{align}$$

We then have that

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

We also for simplicity let $\rho_0 = 1$, so $R = \langle \rho_0, \ldots, \rho_q \rangle$, and define $[q]$ to mean $\{0, \ldots, q\}$.

Equation 16: c-UBER$^{\mathcal{A}}_{c,R,S,T,f}(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_0(\lambda)$

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

```
$\mathcal{B}_0(1^\lambda, N, G, G_T, e, h, W)$
$x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow W^{\rho_k(x_1,\ldots,x_c)}\ \forall k \in [q]$
$y_k \leftarrow h^{\sigma_k(x_1,\ldots,x_c)}\ \forall k \in [s]$
$z_k \leftarrow e(W, h)^{\tau_k(x_1,\ldots,x_c)}\ \forall k \in [t]$
$v' \leftarrow W^{f(x_1,\ldots,x_c)}$
$v'' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, y_1, \ldots, y_s, z_1, \ldots, z_t)$
return $(v'' = v')$
```

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

Equation 17: $\mathsf{G}^{\mathcal{A}}_0(\lambda)$ to $\mathsf{G}^{\mathcal{A}}_1(\lambda)$

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

```
$\underline{\mathcal{C}_0(1^\lambda, N, G, G_T, e, g_1, W)}$
$x_1, \ldots, x_c \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow g_1^{\rho_k(x_1,\ldots,x_c)}\ \forall k \in [q]$
$y_k \leftarrow W^{\sigma_k(x_1,\ldots,x_c)}\ \forall k \in [s]$
$z_k \leftarrow e(g_1, W)^{\tau_k(x_1,\ldots,x_c)}\ \forall k \in [t]$
$v' \leftarrow g_1^{f(x_1,\ldots,x_c)}$
$v'' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, y_1, \ldots, y_s, z_1, \ldots, z_t)$
return $(v'' = v')$
```

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

Equation 18: $\mathsf{G}_i^{\mathcal{A}}(\lambda)$ to $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$

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

```
$\underline{\mathcal{B}_i(1^\lambda, N, G, G_T, e, g_2, h_1, W)}$
$x_1, \ldots, x_c, x_{1,1}, \ldots, x_{1,c}, \ldots, x_{i+1,1}, \ldots, x_{i+1,c} \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
$v_k \leftarrow W^{\rho_k(x_1,\ldots,x_c)} g_2^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})}\ \forall k \in [q]$
$v' \leftarrow W^{f(x_1,\ldots,x_c)} g_2^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})}$
$y_k \leftarrow h_1^{\sigma_k(x_1,\ldots,x_c)}\ \forall k \in [s]$
$z_k \leftarrow e(W, h_1)^{\tau_k(x_1,\ldots,x_c)}\ \forall k \in [t]$
$v'' \xleftarrow{\$} \mathcal{A}(1^\lambda, (N, G, H, G_T, e), v_1, \ldots, v_q, y_1, \ldots, y_s, z_1, \ldots, z_t)$
return $(v'' = v')$
```

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

$$v_k \leftarrow g_1^{\rho_k(x_1,\ldots,x_c)} g_2^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \quad \text{and} \quad v' \leftarrow g_1^{f(x_1,\ldots,x_c)} g_2^{\sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})},$$

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

$$\begin{aligned} v_k &\leftarrow g_1^{\rho_k(x_1,\ldots,x_c)} g_2^{r_{i+1}\rho_k(x_1,\ldots,x_c) + \sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})} \quad \text{and} \\ v' &\leftarrow g_1^{f(x_1,\ldots,x_c)} g_2^{r_{i+1} f(x_1,\ldots,x_c) + \sum_{j=1}^{i} r_j f(x_{j,1},\ldots,x_{j,c})}, \end{aligned}$$

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

Equation 19: $\mathsf{G}_{i,1}^{\mathcal{A}}(\lambda)$ to $\mathsf{G}_{i+1}^{\mathcal{A}}(\lambda)$.

If we define $A = g_2^{\sum_{j=1}^{i} r_j \rho_k(x_{j,1},\ldots,x_{j,c})}$, then A is independent from $\vec{x}, \vec{x}_{i+1}$. By parameter hiding with respect to $R \cup \{f\}$, 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 (and the same is true using f), which proves Equation 19.

Equation 20.

This follows by definition, as the R, S, T, and f values have now changed to the form specified in the dual-system assumption. □