Interactively Secure Groups from Obfuscation

Thomas Agrikola and Dennis Hofheinz

Karlsruhe Institute of Technology (KIT)
thomas.agrikola@kit.edu, dennis.hofheinz@kit.edu

Abstract. We construct a mathematical group in which an interactive variant of the very general Uber assumption holds. Our construction uses probabilistic indistinguishability obfuscation, fully homomorphic encryption, and a pairing-friendly group in which a mild and standard computational assumption holds. While our construction is not practical, it constitutes a feasibility result that shows that under a strong but generic, and a mild assumption, groups *exist* in which very general computational assumptions hold. We believe that this grants additional credibility to the Uber assumption.

Keywords: indistinguishability obfuscation, Uber assumption

1 Introduction

Cyclic groups in cryptography Cyclic groups (such as subgroups of the multiplicative group of a finite field, or certain elliptic curves) are a popular mathematical building block in cryptography. Countless cryptographic constructions are formulated in a cyclic group setting. Usually these constructions are accompanied by a security reduction that transforms any adversarial algorithm that breaks the scheme into an algorithm that solves a computational problem in that group. Among the more popular computational problems are the (computational or decisional) Diffie-Hellman problem [26], or the discrete logarithm problem.

The currently known security reductions of several relevant cryptographic schemes require somewhat more exotic computational assumptions, however. For instance, the security of the Digital Signature Algorithm is only proven in a generic model of computation [14] (see also [15]). Moreover, the semi-adaptive (i.e., IND-CCA1) security of the ElGamal encryption scheme requires a "one-more type" assumption [34]. The currently most efficient structure-preserving signature schemes require complex interactive assumptions [1, 2]. Finally, some proofs (e.g., [24, 32, 5, 25]) even require "knowledge assumptions" that essentially state that the only way to generate new group elements is as linear combinations of given group elements (with extractable coefficients).

While more exotic assumptions can thus be very helpful for constructing cryptographic schemes, their use also has a downside: reductions to more exotic (and less investigated) assumptions tend to lower our confidence in the corresponding scheme. (See [12] and [33] for two very different views on this matter.)

The Uber-assumption family An example of a somewhat exotic but very general and strong class of computational assumptions in a cyclic group setting is the "Uber" assumption family ([9], see also [12]). Essentially, this assumption states that no efficient adversary $\mathcal{A}$ can win the following guessing game significantly better than with probability 1/2. The game is formulated in a group $\mathcal{G} = \langle g \rangle$ of order q, and is parameterized over polynomials $P_1, \ldots, P_l, P^* \in \mathbb{Z}_q[X_1, \ldots, X_m]$. Initially, the game chooses secret exponents $s_1, \ldots, s_m \in \mathbb{Z}_q$ uniformly, and hands $\mathcal{A}$ the group elements $g^{P_i(s_1,\ldots,s_m)}$, and a challenge element $Z \in \mathcal{G}$ with either $Z = g^{P^*(s_1,\ldots,s_m)}$ or independently random Z. Given these elements, $\mathcal{A}$ has to guess if Z is random or not.[1]

Depending on the number m of variables, and the concrete polynomials P_i and P^*, the Uber assumption generalizes many popular existing assumptions, such as the Decisional Diffie-Hellman assumption, the k-Linear family of assumptions, and so-called "q-type assumptions". However, it is a priori not at all clear how plausible such general assumptions are. In fact, there are indications that, e.g., q-type assumptions are indeed easier to break than, say, the discrete log assumption [22].

Fortunately, a number of cryptographic constructions that rely on q-type assumptions can be transported into composite-order groups, with the advantage that now their security holds under a simpler, subgroup indistinguishability assumption [21, 20]. However, this change of groups will not work for every cryptographic construction, and currently we only know how to perform this technique for a subclass of q-type assumptions.

Our contribution In this work, we shed new light on the plausibility of Uber-style assumptions. Concretely, we construct a group in which an interactive variant of Uber-style assumptions (in which the adversary may choose the P_i and P^* adaptively) holds. We believe that this lends additional credibility to the Uber assumption itself, and also strengthens plausibility results obtained from the Uber assumption (see [12] for an overview).

Our construction assumes subexponentially secure indistinguishability obfuscation (iO, a very strong but generic assumption), a perfectly correct additively homomorphic encryption scheme for addition modulo a given prime, and a pairing-friendly group in which a standard assumption (SXDH, the symmetric external Diffie-Hellman assumption) holds. We stress that we consider our result as a feasibility result. Indeed, due to the use of indistinguishability obfuscation, our construction is far from practical. Still, our result shows that even interactive generalizations of the Uber assumption family are no less plausible than indistinguishability obfuscation (plus a standard assumption in cyclic groups and additively homomorphic encryption).

Before describing our results in more detail, we remark that the group we construct actually has non-unique element encodings (much like in a "graded encoding scheme" [27], only without any notion of multilinear map). It is hence possible to compare and operate with group elements, but it is not directly

[1] Owing to the original application, the Uber assumption family was formulated in [9] in a setting with a pairing-friendly group, with a final challenge in the target group.

possible to use, e.g., the encoding of group elements to hide an encrypted message. (In particular, it is not immediately possible to implement, say, the ElGamal encryption scheme with our group as there is no obvious way to decrypt ciphertexts. Signature schemes, however, do not require unique encodings of group elements and can hence be implemented using our group.) Furthermore, due to technical reasons our construction requires the maximum degree of the adversarially chosen polynomials to be bounded a priori.

Related work Pass et al. [37] introduce semantically secure multilinear (and graded) encoding schemes (of groups). A semantically secure encoding scheme guarantees security of a class of algebraic decisional assumptions. On a high level, the security property requires that encodings are computationally indistinguishable whenever there is no way to distinguish the corresponding elements using only generic operations. The generic multilinear encoding model implies semantic security of a multilinear encoding scheme. Furthermore, Pass et al. show that many existing iO candidates [28, 13, 4] that are proven secure in the generic multilinear encoding model can also be proven secure assuming semantically secure encoding schemes. Hence, this result relaxes the necessary assumptions to prove the security of certain iO constructions. Bitansky et al. [7] slightly strengthen the security property of encoding schemes formulated in [37]. Assuming the resulting security property allows to prove that existing obfuscation candidates [4] provide virtual grey-box security[2].

In [3] Albrecht et al. construct a group scheme providing a multilinear map from iO. This result complements earlier results that construct iO from multilinear maps [28, 40]. The notion of encoding schemes used in [3] is a direct adaption of the "cryptographic" multilinear group setting from [10]. In contrast to [37, 7], the encoding scheme of Albrecht et al. provides an extraction algorithm producing a unique string for all encodings that are equal with respect to the equality relation of the scheme. Furthermore, [3] requires a publicly available sampling algorithm that produces encodings for given exponents. Hence, the encoding scheme of [3] grants adversaries slightly more power.

In this paper we use a similar notion of encoding schemes as in [3]. Furthermore, [37, 7] define the security property for encoding schemes implicitly. We, in contrast, consider a concrete strong interactive hardness assumption that holds in our encoding scheme.

Technical approach The assumption we consider is defined similarly to the Uber assumption above, only with an interactive and adaptive choice of arbitrary (multivariate) polynomials P_i, P^* over $\mathbb{Z}_q$, where q is the order of the group. That is, there is a secret point $\boldsymbol{s} := (s_1, \ldots, s_m) \in \mathbb{Z}_q^m$, and $\mathcal{A}$ may freely and

[2] An obfuscator $\mathcal{O}$ satisfies virtual grey-box security for a class of circuits $\mathcal{C}$ if for any circuit $C \in \mathcal{C}$, a PPT adversary given $\mathcal{O}(C)$ can not compute significally more about C than a simulator given unbounded computational resources and polynomially many queries to the circuit C.

adaptively choose the P_i and P^* during the course of the security game. To avoid trivialities, we require that P^* does not lie in the linear span of the polynomials P_i. We call this assumption the *Interactive Uber assumption.* For convenience only, we will describe our approach assuming only univariate polynomials in the Interactive Uber assumption. However, we will see that similar techniques yield security even for multivariate polynomials.

Our starting point is a recent work by Albrecht et al. [3], which constructs a group with a multilinear map from (probabilistic) iO, an additively homomorphic encryption scheme, a dual mode NIZK proof system, and a group $\mathcal{G}$ in which (a variant of) the Strong Diffie-Hellman assumption [8] holds. For our purposes, we are not interested in obtaining a multilinear map, however, and we would also like to avoid relying on a strong (i.e., q-type) assumption to begin with. Moreover, [3] only proves relatively mild computational assumptions in the constructed group.

In a nutshell, a group element in the construction of [3] has the form

$$(g^z,\, C = \textsc{Enc}(z),\, \pi), \tag{1}$$

where $z \in \mathbb{Z}$ is the discrete logarithm of that group element, $g \in \mathcal{G}$ is a generator of the used existing group $\mathcal{G}$, $\textsc{Enc}$ is the encryption algorithm of an additively homomorphic encryption scheme, and π is a non-interactive zero-knowledge proof of consistency. Concretely, π proves that C encrypts the discrete logarithm z of g^z, *or* that C encrypts a polynomial f with $f(w) = z$, for a fixed value w committed to in the public parameters.

In their security analysis, Albrecht et al. [3] crucially use a "switching lemma" that states that different encodings $(g^z, \textsc{Enc}(z), \pi)$ and $(g^{f(w)}, \textsc{Enc}(f), \pi')$ are computationally indistinguishable whenever $f(w) = z$. This allows to switch to, and argue about encodings with higher-degree f. Note, however, that any such encoding must also carry a valid $g^z = g^{f(w)}$. Hence, changing the values $z = f(w)$ in such encodings with higher-degree f (as is often required to prove security) would seem to already necessitate Uber-style assumptions. Indeed, Albrecht et al. require a variant of the Strong Diffie-Hellman assumption, a q-type assumption.

Group elements in our group To avoid making Uber-style assumptions in the first place, we simply omit the initial g^z value in encodings of group elements, and modify the consistency proof from Eq. (1). That is, group elements in our group are of the form

$$(C = \textsc{Enc}(z),\, \pi), \tag{2}$$

where $\textsc{Enc}$ is the encryption algorithm of an additively homomorphic encryption scheme, and π is a proof of knowledge of some (potentially constant) polynomial f' with $f'(w) = z$ or $f'(w) = f(w)$ (in case C encrypts a polynomial f). The value w is some point in $\mathbb{Z}_q$ that is fixed, but hidden, in the public parameters of our group, where q is the group order. The proof of knowledge is realized through an additional encryption C' that contains the polynomial f'. Hence, group elements are actually of the form

$$(C = \textsc{Enc}(z),\, C' = \textsc{Enc}(f'),\, \pi). \tag{3}$$

In a nutshell, such an encoding implicitly represents the group element $g^{f(w)} = g^{f'(w)}$, where f and f' are the polynomials defined by C and C' respectively. For clarity, we sometimes omit the component C' in this overview.

More precisely, C and C' contain representation vectors $\overrightarrow{f}$ and $\overrightarrow{f'}$ of the polynomials f and f' with respect to a basis $\{a_1, \ldots, a_d\}$ of $\mathbb{Z}_q^d$. That is, given a vector $\overrightarrow{f}$ that is encrypted in C, the coefficients of the corresponding polynomial f are defined as follows

$$(a_1 \mid \ldots \mid a_d)^{-1} \cdot \overrightarrow{f} \tag{4}$$

using the homomorphic mapping between polynomials over $\mathbb{Z}_q$ and vectors in $\mathbb{Z}_q^d$. This basis is not public, but committed to in the public parameters. The reason for using a hidden basis is that we need to deal with adaptive queries. We postpone the details to a subsequent paragraph. In this overview, however, we will pretend the ciphertexts C and C' contain mere polynomials.

Intuitively, the crux of the matter for the proof of security will be to remove the dependency on the point w. This changes the group structure to be isomorphic to $\mathbb{Z}_q^d$ which makes it possible to argue with linear algebra.

A public sampling algorithm allows to produce arbitrary encodings of group elements. Given an exponent z, the sampling algorithm produces the ciphertexts C and C' using the constant polynomials $f := f' := z$ and produces the consistency proof accordingly. We remark that our group allows for re-randomization of encodings assuming some natural additional properties of the homomorphic encryption scheme.

The group operation is performed in a similar way to [3]. Namely, suppose we want to add two encodings $(\textsc{Enc}(f_1), \pi_1)$ and $(\textsc{Enc}(f_2), \pi_2)$. The resulting $(\textsc{Enc}(f_3), \pi_3)$ should satisfy $f_3 = f_1 + f_2$ as abstract polynomials. Hence, $\textsc{Enc}(f_3)$ can be computed homomorphically from $\textsc{Enc}(f_1)$ and $\textsc{Enc}(f_2)$. To compute the proof π_3, however, we require an obfuscated circuit C_{Add} that extracts f_1, f_2, and generates a fresh proof using the knowledge of $f_3 = f_1 + f_2$ as witness. Thus, the implementation of C_{Add} needs to know both decryption keys for C and C'. (The details are somewhat technical and similar to [3], so we omit them in this overview.) We prove that it is possible to implement a circuit C''_{Add} that has almost the same functionality as C_{Add} but produces a simulated proof of consistency that is identically distributed to a real one. Hence, the implementation of C''_{Add} does not need to know the decryption keys. Therefore, exploiting the security of the used obfuscator, we are able to unnoticeably replace the obfuscation of C_{Add} with an obfuscation of C''_{Add}.

We note that our modification to omit the entry g^z from the encodings in Eq. (1) makes it nontrivial to decide whether two given encodings represent the same group element, or, equivalently, to decide whether a given encoding represents the identity element of the group. Recall that an encoding $(C = \textsc{Enc}(f), \pi)$ represents the group element $g^{f(w)}$. (This operation is trivial in the setting of Albrecht et al., since their encodings carry a value $g^z = g^{f(w)}$.) Thus, our construction needs to provide a public algorithm that tests whether a given encoding $(C = \textsc{Enc}(f), \pi)$ represents the identity element of the group, i.e. that tests whether $f(w) = 0$.

At this point two problems arise. First, this public algorithm must be able to obtain at least one of the polynomials that are encrypted in C and C' respectively. Second, the value w must not be explicitly known during the proof of security as our strategy is to remove the dependency on w. We solve both problems by using an *obfuscated* circuit C_{Zero} for testing whether a given encoding represents the identity element. More precisely, given an encoding $(C = \textsc{Enc}(f), \pi)$, C_{Zero} decrypts C (using one fixed decryption key) to obtain the polynomial f. In order to avoid the necessity to explicitly know the value w, C_{Zero} factors the univariate polynomial f (in $\mathbb{Z}_q[X]$), and obtains the small set $\{x_1, \ldots, x_n\}$ of all zeros of f.[3] As mentioned above, the value w is fixed but hidden inside the public parameters. Particularly, we store the value w in form of a point function obfuscation (i.e., in form of a publicly evaluable function $\mathsf{po}\colon \mathbb{Z}_q \to \{0,1\}$ with $\mathsf{po}(x) = 1 \Leftrightarrow x = w$, such that it is hard to determine the value w given only the function description po). The zero testing circuit C_{Zero} treats an encoding as the identity element if f is the zero polynomial or $w \in \{x_1, \ldots, x_n\}$.

Observe that this implementation of C_{Zero} only requires one decryption key allowing to apply the Naor-Yung strategy [36]. Furthermore, C_{Zero} does not need to know the value w in the clear. Hence, using an obfuscation of this implementation of C_{Zero} avoids both problems described above.

Switching of encodings Similarly to Albrecht et al. [3] we prove a "switching lemma" that states that encodings $(C_1 = \textsc{Enc}(f_1), \pi_1)$ and $(C_2 = \textsc{Enc}(f_2), \pi_2)$ are computationally indistinguishable whenever $f_1(w) = f_2(w)$. In other words, encodings of the same group element are computationally indistinguishable. To prove this lemma, we exploit the security of the used double-encryption in a similar way as in the IND-CCA proof of Naor and Yung [36]. Particularly, when using an obfuscation of the circuit C''_{Add}, it is not necessary to know both decryption keys to produce public parameters for the group. We recall that the circuit C_{Zero} only knows the decryption key to decrypt the first component of encodings. Furthermore, it is possible to produce a consistency proof without knowing the content of the ciphertexts C and C' by simply simulating it in the same way C''_{Add} does. Therefore, we can reduce to the IND-CPA security of the encryption scheme. In order to apply the same argument for the first component of encodings, we need the circuit C_{Zero} to forget about the first decryption key. We accomplish that by replacing the obfuscation of C_{Zero} with an obfuscation of the circuit $\overline{C}_{\mathsf{Zero}}$ that uses only the second decryption key instead of the first one. This is possible due to the security of the obfuscator and the soundness of the proof system. Then, we can use the same argument as above to reduce to the IND-CPA security of the encryption scheme.

Obtaining the *Interactive Uber assumption* in our group We recall that the Interactive Uber assumption (in one variable) generates one secret point

[3] We note that there are probabilistic polynomial time algorithms that factor univariate polynomials over finite fields, for instance the Cantor-Zassenhaus algorithm [18].

$s \in \mathbb{Z}_q$ uniformly at random at which all queried polynomials are evaluated. To show that the Interactive Uber assumption holds in our group, we first set up that secret point s as $c \cdot w$ for some independent random c from $\mathbb{Z}_q^\times$, where w is the secret value of our group introduced above. Hence, a polynomial P that is evaluated at $s = c \cdot w$ can be interpreted as a (different) polynomial in w. Particularly, given a polynomial $P(X)$, the polynomial $\overline{P}(X) := P(c \cdot X)$ satisfies the equation $P(s) = \overline{P}(w)$. Thus, an encoding that contains the polynomial $\overline{P}(X)$ determines the exponent of the represented group element to equal $\overline{P}(w) = P(c \cdot w) = P(s)$. This observation paves the way for using higher-degree polynomials $\overline{P}(X)$ to produce encodings for oracle answers and the challenge. As the resulting group elements (i.e. the corresponding exponents) remain the same, the "switching lemma" described above justifies that this modification is unnoticeable. Furthermore, by a similar argument as above, we simulate the proofs of consistency π for every produced encoding, in particular for the encodings that are produced by the addition circuit.[4] As the consistency proof can now be produced independently of the basis $\{a_1, \ldots, a_d\}$, we are able to unnoticeably "erase" this basis from the commitment in the public parameters.

Our goal now is to alter the structure of the group in the following sense. By definition, our group is isomorphic to the additive group $\mathbb{Z}_q$. We aim to alter that structure such that our group is isomorphic to the additive group of polynomials in $\mathbb{Z}_q[X]$ (of bounded degree). Particularly, we alter the equality relation that is defined on the set of encodings such that two encodings are considered equal only if the thereby defined polynomials are equal as abstract polynomials. For that purpose, we remove the dependency on the point w by altering the point function obfuscation po such that it maps all inputs to 0. Therefore, the zero testing circuit C_{Zero} only treats an encoding that contains the zero polynomial as an encoding of the identity element of the group. As the value w is never used explicitly in the game (as all the proofs of consistency are simulated), this modification is unnoticeable due to the security property of the point function obfuscation po. This is a crucial step paving the way for employing arguments from linear algebra to enable randomization.

The final step requires to randomize the challenge encoding such that there is no detectable difference between a real challenge and a randomly sampled one. First, we recall that encodings do not encrypt polynomials in the plain. The encodings contain the representation of polynomials with respect to some basis $\{a_1, \ldots, a_d\}$. That is, given a polynomial $P(X)$, the encoding corresponding to $g^{P(s)}$ encrypts the vectors

$$\overrightarrow{f} = \overrightarrow{f'} = (a_1 \mid \ldots \mid a_d) \cdot \underbrace{P(c \cdot X)}_{=\overline{P}(X)}, \tag{5}$$

where $P(c \cdot X)$ is interpreted as a vector of coefficients in the natural way. Therefore, the only information about the matrix $(a_1 | \ldots | a_d)$ is given by matrix

[4] More precisely, we again use an obfuscation of C''_{Add} instead of an obfuscation of C_{Add} as described above.

vector products. To avoid trivialities, the challenge polynomial P^* can be assumed not to lie in the span of the queries $P_1, \dots, P_l$, which is why $P^*(c \cdot X)$ does not lie in the span of $P_1(c \cdot X), \dots, P_l(c \cdot X)$. Hence, we may resort to an information-theoretic argument. More precisely, an adversary that is able to adaptively ask for matrix vector products, information-theoretically learns nothing about matrix vector products that are linearly independent of its queries. Therefore, the polynomial that is contained in the real challenge encoding information-theoretically looks like a randomly sampled polynomial (with bounded degree) given that the matrix $(a_1 | \dots | a_d)$ is uniformly distributed.

Obtaining the *multivariate* Interactive Uber assumption The main difficulty that arises from generalizing our results to the multivariate Interactive Uber assumption is that we do not have a polynomial-time algorithm that computes all zeros of a multivariate polynomial. Hence, the zero testing circuit C_{Zero} needs to know the point $\boldsymbol{\omega} := (\omega_1, \dots, \omega_m) \in \mathbb{Z}_q^m$ in the clear to explicitly evaluate the polynomial f that is defined by a given encoding. Our previous proof strategy, however, crucially relies on removing the dependency on w such that C_{Zero} only treats encodings containing the zero polynomial as encodings of the identity element. This is equivalent to altering the group structure such that it is isomorphic to the additive group of polynomials over $\mathbb{Z}_q$ (of bounded degree).

Although the zero testing circuit C_{Zero} knows $\boldsymbol{\omega}$ in the clear, it is nevertheless possible to pursue a similar strategy. Our solution is to gradually alter C_{Zero} such that it "forgets" the components ω_i of $\boldsymbol{\omega}$ one by one. Particularly, we define intermediate circuits $C_{\mathsf{Zero}}^{(i)}$ that test if the polynomial

$$F_i^{(f)}(X_1, \dots, X_i) := f(X_1, \dots, X_i, \omega_{i+1}, \dots, \omega_m) \tag{6}$$

equals the zero polynomial in $\mathbb{Z}_q[X_1, \dots, X_i]$. Observe that the original circuit C_{Zero} tests whether $F_0^{(f)} \equiv 0$. Our goal is to unnoticeably establish $C_{\mathsf{Zero}}^{(m)}$ as zero testing circuit, as it realizes the stricter equality relation we aim for.

In order to unnoticeably replace an obfuscation of $C_{\mathsf{Zero}}^{(i)}$ with an obfuscation of $C_{\mathsf{Zero}}^{(i+1)}$, we first alter the implementation of $C_{\mathsf{Zero}}^{(i)}$ such that it performs the test whether $F_i^{(f)}$ is the zero polynomial by evaluating it at a randomly sampled point $\boldsymbol{r} \in \mathbb{Z}_q^i$. Applying the Schwartz-Zippel lemma upper bounds the statistical distance of the output distributions of the two circuits enabling to reduce this step to the security of the obfuscator.

Furthermore, the condition that $F_i^{(f)}(\boldsymbol{r}) = F_{i+1}^{(f)}(\boldsymbol{r}, \omega_{i+1}) = 0$ is equivalent to the condition that the univariate polynomial $F_{i+1}^{(f)}(\boldsymbol{r}, X_{i+1})$ is zero at the point ω_{i+1}. This can be implemented in a similar manner as in the univariate case using a point function obfuscation of ω_{i+1}. In addition, this circuit contains a conceptional logical or statement testing whether the polynomial $F_{i+1}^{(f)}(\boldsymbol{r}, X_{i+1})$ equals the zero polynomial. Using a similar argument as above we are able to alter the point function obfuscation for ω_{i+1} to a point function obfuscation that never triggers.

Hence, our zero testing circuit effectively only tests whether $F_{i+1}^{(f)}(\boldsymbol{r}, X_{i+1})$ equals the zero polynomial in $\mathbb{Z}_q[X_{i+1}]$. Applying the Schwartz-Zippel lemma again, we are able to unnoticeably alter the implementation of the zero testing circuit such that it tests whether $F_{i+1}^{(f)}$ equals the zero polynomial in $X_1, \ldots, X_{i+1}$ concluding the argument.

Roadmap After fixing notation and recalling some basic definitions in Section 2, we present our main group construction in Section 3. In Supplementary Section A we prove several technical lemmas that facilitate proving our main theorem. Our main theorem, Theorem 1, states the validity of (our variant of) the Interactive Uber assumption relative to the group construction from Section 3. In particular, in Supplementary Section A.1 we prove that it is hard to decide whether public parameters of the group are generated honestly or such that all proofs of consistency are simulated. In Supplementary Section A.2, we prove a "switching lemma" for encodings of group elements and in Supplementary Section A.3 we prove the above mentioned information-theoretic argument that enables the randomization of the challenge in the main proof. Finally, Theorem 1 appears in Supplementary Section B.

2 Preliminaries

2.1 Notation

For $n \in \mathbb{N}$, let 1^n denote the string consisting of n times the digit 1. For a probabilistic algorithm A, let $y \leftarrow A(x)$ denote that y is the output of A on input x. The randomness which A uses during the computation can be made explicit by $y \leftarrow A(x; r)$, where r denotes the randomness. Let λ denote the security parameter. We assume that the security parameter is implicitly given to all algorithms as 1^λ.

Let $\mathcal{G}$ be a group and let h be a fixed generator of $\mathcal{G}$. Then, $[n]$ denotes the group element h^n.

Let $n \in \mathbb{N}$ be a number, let $\mathbb{K}$ be a field, and let $\mathbb{K}^n$ denote the vector space of n-tuples of elements of $\mathbb{K}$. Further, for any $i \in \{1, \ldots, n\}$, let $e_i \in \mathbb{K}^n$ be the vector such that the i-th entry of e_i equals 1 and any remaining entry equals 0. Then, the set $\{e_1, e_2, \ldots, e_n\}$ denotes the *standard basis* of $\mathbb{K}^n$. Let $b_1, \ldots, b_i \in \mathbb{K}^n$, then $\langle b_1, \ldots, b_i \rangle \subseteq \mathbb{K}^n$ denotes the span of those vectors.

2.2 Assumptions

Let $(\mathcal{G}_\lambda)_{\lambda \in \mathbb{N}}$ be a family of finite cyclic groups. If it is clear from the context, we write $\mathcal{G}$ instead of $\mathcal{G}_\lambda$. We assume that the order $q := |\mathcal{G}|$ of the group is known and *prime*. Let $\mathsf{Gens}_\mathcal{G}$ be the set of generators of $\mathcal{G}$. We assume that we can efficiently sample elements uniformly at random from $\mathsf{Gens}_\mathcal{G}$.

A very basic and well-established cryptographic assumption is the decisional Diffie-Hellman (DDH) assumption. The DDH assumption states that the distributions $([x], [y], [x \cdot y])$ and $([x], [y], [z])$ are computationally indistinguishable for $x, y, z \leftarrow \mathbb{Z}_q$.

Definition 1 (Decisional Diffie-Hellman (DDH) assumption). *For any PPT adversary $\mathcal{A}$, the advantage $Adv^{ddh}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible in λ, where*

$$\begin{aligned} Adv^{ddh}_{\mathcal{G},\mathcal{A}}(\lambda) := &\Pr\big[\mathcal{A}(1^\lambda, [x], [y], [x \cdot y]) = 1 \bigm| x, y \leftarrow \mathbb{Z}_q\big] \\ &- \Pr\big[\mathcal{A}(1^\lambda, [x], [y], [z]) = 1 \bigm| x, y, z \leftarrow \mathbb{Z}_q\big] \end{aligned}$$

and q is the order of the group $\mathcal{G}$.

Let $(\mathcal{G}_1, \mathcal{G}_2, e)$ be finite cyclic groups of prime order $|\mathcal{G}_1| = |\mathcal{G}_2|$ and let $e\colon \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ be a pairing (i.e. a non-degenerate and bilinear map). The groups $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_T$, as well as the pairing e depend on the security parameter. For greater clarity, we omit this dependency in this setting.

A natural extension of the DDH assumption to the bilinear setting is the symmetric external Diffie-Hellman (SXDH) assumption. The SXDH assumption states that the DDH assumption holds in both groups $\mathcal{G}_1$ and $\mathcal{G}_2$.

2.3 Point obfuscation

In our construction we employ a cryptographic primitive that is called *point obfuscation* [16, 39]. A point obfuscation serves the purpose to hide a certain point, but to enable a test whether a given value is hidden inside. Equivalently, this notion can be seen as an "obfuscation" of a point-function that evaluates to 1 at exactly this given point and to 0 everywhere else. We require that it is infeasible to distinguish a point obfuscation that triggers at a randomly sampled point from a point obfuscation that never triggers. This security requirement is rather weak compared to similar notions [6].

Definition 2 (Point obfuscation). *A* point obfuscation *for message space $\mathcal{M}_\lambda$ is a PPT algorithm* POBF.

$\mathrm{POBF}(1^\lambda, x) \to \mathsf{po}$ *On input a message $x \in \mathcal{M}_\lambda \cup \{\bot\}$,* POBF *produces a description of the point function*

$$\mathsf{po}\colon \mathcal{M}_\lambda \to \{0,1\},\ y \mapsto \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise} \end{cases}.$$

We require the following two properties to hold:

Correctness: *For any x, $y \in \mathcal{M}_\lambda$ and any $\mathsf{po} \leftarrow \mathrm{POBF}(1^\lambda, x)$, $\mathsf{po}(y) \mapsto 1$ if and only if $x = y$.*

Soundness: *For any PPT adversary $\mathcal{A}$, the advantage $Adv^{po}_{\mathrm{POBF},\mathcal{A}}(\lambda)$ is negligible in λ, where*

$$\begin{aligned} Adv^{po}_{\mathrm{POBF},\mathcal{A}}(\lambda) := &\Pr\big[\mathcal{A}(1^\lambda, \mathsf{po}) = 1 \bigm| \mathsf{po} \leftarrow \mathrm{POBF}(1^\lambda, x),\ x \leftarrow \mathcal{M}_\lambda\big] \\ &- \Pr\big[\mathcal{A}(1^\lambda, \mathsf{po}) = 1 \bigm| \mathsf{po} \leftarrow \mathrm{POBF}(1^\lambda, \bot)\big]. \end{aligned}$$

An adaption of a construction proposed in [16] yields a point obfuscation $\mathrm{PO}\textsc{bf}$ with message space $\mathbb{Z}_p$ based on the DDH assumption. Furthermore, a point obfuscation with message space $\mathbb{Z}_p$ can be used to construct a point obfuscation for message space $\mathbb{Z}_q$, where q is a prime such that $\frac{p}{q}$ is negligible in λ. For further details, we refer the reader to Supplementary Section C.

Remark 1. According to a reviewer of TCC 2017, a point obfuscation with message space $\{0,1\}^{\mathsf{poly}(\lambda)}$ can be constructed from an injective one-way function F together with a corresponding hardcore bit B.

Given a string x, the tuple $(F(x), B(x))$ is the obfuscation of x. The tuple $(F(y), 1 - B(y))$ is an obfuscation of $\perp$, where y is a random element from the message space.

2.4 Subset membership problems

The notion of subset membership problems was introduced in [23]. Informally, a hard subset membership problem specifies a set, such that it is intractable to decide whether a value is inside this set or not. Let $\mathcal{L} = (\mathcal{L}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of families of languages $L \subseteq \mathcal{X}_\lambda$ in a universe $\mathcal{X}_\lambda = \mathcal{X}$. Further, let $\mathcal{R}$ be an efficiently computable witness relation, such that $x \in L$ if and only if there exists a witness $w \in \{0,1\}^{\mathsf{poly}(|x|)}$ with $\mathcal{R}(x, w) = 1$, where poly is a fixed polynomial. We assume that we are able to efficiently and uniformly sample elements from L together with a corresponding witness, and that we are able to efficiently and uniformly sample elements from $\mathcal{X} \setminus L$.

Definition 3 (Hard subset membership problem). *The* subset membership problem *(SMP) $L \subseteq \mathcal{X}$ is hard, if for any PPT adversary $\mathcal{A}$, the advantage*

$$Adv^{smp}_{\mathcal{L},\mathcal{A}}(\lambda) := \Pr\big[\mathcal{A}(1^\lambda, x) = 1 \;\big|\; x \leftarrow L\big] - \Pr\big[\mathcal{A}(1^\lambda, x) = 1 \;\big|\; x \leftarrow \mathcal{X} \setminus L\big]$$

is negligible in λ.

For our construction we need a family $\mathcal{L} = (\mathcal{L}_\lambda)_{\lambda\in\mathbb{N}}$ such that for any $L \in \mathcal{L}_\lambda$ and any $x \in L$, there exists exactly one witness $r \in \{0,1\}^*$ with $\mathcal{R}(x, w) = 1$.

Let $\mathcal{G} = \{\mathcal{G}_\lambda\}$ be a family of finite cyclic groups of prime order such that the DDH assumption holds. A possible instantiation of a hard SMP meeting our requirements is the Diffie-Hellman language $\mathcal{L}^{\mathrm{dh}} := (\mathcal{L}^{\mathrm{dh}}_\lambda)_{\lambda\in\mathbb{N}}$. For any $\lambda \in \mathbb{N}$, $\mathcal{L}^{\mathrm{dh}}_\lambda := \{L_{g,h} \,|\, g, h \in \mathsf{Gens}_\mathcal{G}\}$, $\mathcal{X}_\lambda = \mathsf{Gens}_\mathcal{G} \times \mathsf{Gens}_\mathcal{G}$, and $L_{g,h} := \{(g^r, h^r) \,|\, r \in \mathbb{Z}_q\}$, where $q = |\mathcal{G}_k|$. The SMP $L_{g,h} \subseteq \mathcal{X}$ is hard for randomly chosen generators g, $h \leftarrow \mathsf{Gens}_\mathcal{G}$. Given $(g^r, h^r) \in L_{g,h}$, the corresponding unique witness is $r \in \mathbb{Z}_q$.

2.5 Non-interactive commitments

Non-interactive commitment schemes are a commonly used cryptographic primitive [30]. They enable to commit to a chosen value without revealing this value. Additionally, once committed to a value, this value cannot be changed. In contrast to the notion of point obfuscations, a commitment scheme prevents to test whether a particular value is hidden inside a commitment.

Definition 4 (Perfectly binding non-interactive commitment scheme (syntax and security)). *A* perfectly binding non-interactive commitment scheme *for message space $\mathcal{M}_\lambda$ is a triple of PPT algorithms* $\textsc{Com} = (\textsc{ComSetup}, \textsc{Commit}, \textsc{Open})$.

$\textsc{ComSetup}(1^\lambda) \to ck$ *On input the unary encoded security parameter, the algorithm* $\textsc{ComSetup}$ *outputs a commitment key ck.*

$\textsc{Commit}_{ck}(m) \to (com, op)$ *On input the commitment key ck and a message $m \in \mathcal{M}_\lambda$,* $\textsc{Commit}$ *outputs a tuple (com, op).*

$\textsc{Open}_{ck}(com, op) \to \widetilde{m}$ *On input the commitment key ck and a commitment-opening pair (com, op),* $\textsc{Open}$ *outputs the committed message m if op is a valid opening for com. Otherwise,* $\textsc{Open}$ *outputs $\perp$.*

We require $\textsc{Com}$ *to be* perfectly correct, perfectly binding, *and* computationally hiding.

Correctness $\textsc{Com}$ *is* correct *if for any $\lambda \in \mathbb{N}$, any $ck \leftarrow \textsc{ComSetup}(1^\lambda)$, and any $m \in \mathcal{M}_\lambda$, $\textsc{Open}_{ck}(\textsc{Commit}_{ck}(m)) = m$.*

Perfectly binding $\textsc{Com}$ *is* perfectly binding *if it is not possible to find a commitment that has valid openings for more than one message, i.e. for any (possibly unbounded) adversary $\mathcal{A}$, $Adv^{binding}_{\textsc{Com},\mathcal{A}}(\lambda) = 0$, where*

$$Adv^{binding}_{\textsc{Com},\mathcal{A}}(\lambda) := \Pr\Big[Exp^{binding}_{\textsc{Com},\mathcal{A}}(\lambda) = 1\Big].$$

Computationally hiding $\textsc{Com}$ *is* computationally hiding *if commitments for different messages are computationally indistinguishable, i.e. for any PPT adversary $\mathcal{A}$, $Adv^{hiding}_{\mathcal{A}}(\lambda)$ is negligible, where*

$$Adv^{hiding}_{\textsc{Com},\mathcal{A}}(\lambda) := \Pr\Big[Exp^{hiding}_{\textsc{Com},\mathcal{A}}(\lambda) = 1\Big] - \frac{1}{2}.$$

The games $Exp^{binding}_{\textsc{Com},\mathcal{A}}(\lambda)$ and $Exp^{hiding}_{\textsc{Com},\mathcal{A}}(\lambda)$ are defined in Fig. 1.

$\textsc{Experiment}\ Exp^{\text{binding}}_{\textsc{Com},\mathcal{A}}(\lambda)$	$\textsc{Experiment}\ Exp^{\text{hiding}}_{\textsc{Com},\mathcal{A}}(\lambda)$
$ck \leftarrow \textsc{ComSetup}(1^\lambda)$	$ck \leftarrow \textsc{ComSetup}(1^\lambda)$
$(c, o_1, o_2) \leftarrow \mathcal{A}(1^\lambda, ck)$	$(m_0, m_1, st) \leftarrow \mathcal{A}(1^\lambda, ck, \mathsf{find})$
$m_1 \leftarrow \textsc{Open}_{ck}(c, o_1)$, $m_2 \leftarrow \textsc{Open}_{ck}(c, o_2)$	$b \leftarrow \{0,1\}$, $(c, o) \leftarrow \textsc{Commit}_{ck}(m_b)$
if $m_1 \neq \perp \wedge m_2 \neq \perp \wedge m_1 \neq m_2$ **then**	$b' \leftarrow \mathcal{A}(1^\lambda, c, st, \mathsf{attack})$
return 1	**if** $b = b'$ **then return** 1
return 0	**return** 0

Fig. 1. The description of the Binding game $Exp^{\text{binding}}_{\textsc{Com},\mathcal{A}}(\lambda)$ (left) and the Hiding game $Exp^{\text{hiding}}_{\textsc{Com},\mathcal{A}}(\lambda)$ (right).

Such a commitment scheme can be obtained from a group in which the DDH assumption holds.

2.6 Dual mode NIWI proof system

The notion of dual mode NIWI proof systems abstracts from the NIWI proof system proposed in [31]. A similar abstraction was used in [3].

Definition 5 (Dual mode NIWI proof system (syntax and security)). *A* dual mode non-interactive witness-indistinguishable (NIWI) proof system *for a relation $\mathcal{R}$ is a tuple of PPT algorithms Π = (Setup_Π, K, S, Prove, Verify, $\mathsf{Extract}$).*

$\mathsf{Setup}_\Pi(1^\lambda) \to (gpk, gsk)$ *On input the unary encoded security parameter, Setup_Π outputs a group key gpk and, additionally, may output some related information gsk. The relation $\mathcal{R}$ is an efficiently computable ternary relation consisting of triplets of the form (gpk, x, w) and defines a group-dependent language L. The language L consists of the statements x, such that there exists a witness w with $(gpk,\ x,\ w) \in \mathcal{R}$.*

$\mathsf{K}(gpk, gsk) \to (crs, td_{ext})$ *On input the group keys gpk and gsk, K outputs a* binding *common reference string (CRS) crs and a corresponding extraction trapdoor td_{ext}.*

$\mathsf{S}(gpk, gsk) \to (crs, \perp)$ *On input the group keys gpk and gsk, S outputs a* hiding *CRS crs.*

$\mathsf{Prove}(gpk, crs, x, w) \to \pi$ *On input the public group key gpk, the CRS crs, a statement x, and a corresponding witness w, Prove produces a proof π.*

$\mathsf{Verify}(gpk, crs, x, \pi) \to \{0, 1\}$ *On input the public group key gpk, the CRS crs, a statement x, and a proof π, Verify outputs 1 if the proof is valid and 0 if the proof is rejected.*

$\mathsf{Extract}(td_{ext}, x, \pi) \to w$ *On input the extraction trapdoor td_{ext}, a statement x, and a proof π, $\mathsf{Extract}$ outputs a witness w.*

We require Π to meet the following requirements:

CRS indistinguishability *Common reference strings generated via K(gpk, gsk) and S(gpk, gsk) are computationally indistinguishable, i.e.*

$$Adv^{crs}_{\Pi,\mathcal{A}}(\lambda) := \Pr\big[Exp^{crs}_{\Pi,\mathcal{A}}(\lambda) = 1\big] - \frac{1}{2}$$

is negligible in λ, where $Exp^{crs}_{\Pi,\mathcal{A}}(\lambda)$ is defined as in Fig. 2.

Perfect completeness under K and S *For any $\lambda \in \mathbb{N}$, any $(gpk,\ gsk) \leftarrow \mathsf{Setup}_\Pi(1^\lambda)$, any CRS $(crs, \cdot) \leftarrow \mathsf{K}(gpk,\ gsk)$, any $(x,\ w)$ such that $(gpk,\ x,\ w) \in \mathcal{R}$, and any $\pi \leftarrow \mathsf{Prove}(gpk,\ crs,\ x,\ w)$, $\mathsf{Verify}(gpk,\ crs,\ x,\ \pi) \to 1$. The same holds for any $(crs, \cdot) \leftarrow \mathsf{S}(gpk,\ gsk)$.*

Perfect soundness under K *For any $\lambda \in \mathbb{N}$, any $(gpk,\ gsk) \leftarrow \mathsf{Setup}_\Pi(1^\lambda)$, any $(crs, \cdot) \leftarrow \mathsf{K}(gpk,\ gsk)$, any statement x such that there exists no witness w with $(gpk,\ x,\ w) \in \mathcal{R}$, and any $\pi \in \{0,1\}^*$, $\mathsf{Verify}(gpk,\ crs,\ x,\ \pi) \to 0$.*

Perfect extractability under K *For any $\lambda \in \mathbb{N}$, any key pair $(gpk,\ gsk) \leftarrow \mathsf{Setup}_\Pi(1^\lambda)$, any $(crs,\ td_{ext}) \leftarrow \mathsf{K}(gpk,\ gsk)$, any $(x,\ \pi)$ such that $\mathsf{Verify}(gpk,\ crs,\ x,\ \pi) \to 1$, and for any $w \leftarrow \mathsf{Extract}(td_{ext},\ x,\ \pi)$, w is a satisfying witness for the statement x, i.e. $(gpk, x, w) \in \mathcal{R}$.*

Perfect witness-indistinguishability under S *For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow \mathsf{Setup}_{\Pi}(1^{\lambda})$, any $(crs, \cdot) \leftarrow \mathsf{S}(gpk, gsk)$, any (x, w_0) and (x, w_1) with (gpk, x, w_0), $(gpk, x, w_1) \in \mathcal{R}$, the output of $\mathsf{Prove}(gpk, crs, x, w_0)$ and the output of $\mathsf{Prove}(gpk, crs, x, w_1)$ are identically distributed.*

Experiment $Exp^{\mathrm{crs}}_{\Pi,\mathcal{A}}(\lambda)$

$(gpk, gsk) \leftarrow \mathsf{Setup}_{\Pi}(1^{\lambda})$
$(crs_0, \cdot) \leftarrow \mathsf{K}(gpk, gsk)$, $(crs_1, \cdot) \leftarrow \mathsf{S}(gpk, gsk)$
$b \leftarrow \{0, 1\}$, $b' \leftarrow \mathcal{A}(1^{\lambda}, gpk, crs_b)$
if $b' = b$ **then return** 1
return 0

Fig. 2. The description of the CRS inistinguishability game $Exp^{\mathrm{crs}}_{\Pi,\mathcal{A}}(\lambda)$.

An exemplary dual mode NIWI proof system satisfying computational CRS indistinguishability, perfect completeness, perfect soundness, perfect extractability, and perfect witness-indistinguishability is the proof system proposed by Groth and Sahai in [31]. The soundness, in particular the indistinguishability of common reference strings, of this construction can for instance be based on the SXDH assumption. The Groth-Sahai proof system allows perfect extractability for group elements, however, does not provide a natural way to extract scalars. Nevertheless, perfect extractability can be achieved by using the proof system for the bit representation of the particular scalars [35].

2.7 Probabilistic indistinguishability obfuscation

The notion of probabilistic circuit obfuscation was proposed in [17]. Informally, probabilistic circuit obfuscation enables to conceal the implementation of probabilistic circuits while preserving their functionality. Let $\mathcal{C} = (\mathcal{C}_{\lambda})_{\lambda \in \mathbb{N}}$ be a family of sets $\mathcal{C}_{\lambda}$ of probabilistic circuits. The set $\mathcal{C}_{\lambda}$ contains circuits of polynomial size in λ. A *circuit sampler* for $\mathcal{C}$ is defined as a set of (efficiently samplable) distributions $S = (S_{\lambda})_{\lambda \in \mathbb{N}}$, where S_{λ} is a distribution over triplets (C_0, C_1, z) with $C_0, C_1 \in \mathcal{C}_{\lambda}$ such that C_0 and C_1 take inputs of the same length and $z \in \{0, 1\}^{\mathsf{poly}(\lambda)}$.

Definition 6 (Probabilistic indistinguishability obfuscation for a class of samplers $\mathcal{S}$, [3, 17]). *A* probabilistic indistinguishability obfuscator *(pIO) for a class of samplers $\mathcal{S}$ over the probabilistic circuit family $\mathcal{C} = (\mathcal{C}_{\lambda})_{\lambda \in \mathbb{N}}$ is a uniform PPT algorithm $pi\mathcal{O}$, such that the following properties hold:*

Correctness *On input the unary encoded security parameter 1^{λ} and a circuit $C \in \mathcal{C}_{\lambda}$, $pi\mathcal{O}$ outputs a deterministic circuit Λ of polynomial size in $|C|$ and λ. For any $\lambda \in \mathbb{N}$, any $C \in \mathcal{C}_{\lambda}$, any $\Lambda \leftarrow pi\mathcal{O}(1^{\lambda}, C)$, and any inputs $m \in \{0, 1\}^*$ (of matching length), there exists a randomness r, such that*

$C(m;r) = \Lambda(m)$.
Furthermore, for every non-uniform PPT distinguisher $\mathcal{D}$, every $\lambda \in \mathbb{N}$, every $C \in \mathcal{C}_\lambda$, and every auxiliary input $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$, the advantage

$$Adv^{pio\text{-}c}_{C,z,\mathcal{D}}(\lambda) := \Pr\Big[Exp^{pio\text{-}c}_{C,z,\mathcal{D}}(\lambda) = 1\Big] - \frac{1}{2}$$

is negligible in λ, where $Exp^{pio\text{-}c}_{C,z,\mathcal{D}}(\lambda)$ is defined as in Fig. 3.

Security with respect to $\mathcal{S}$ *For any circuit sampler $S = \{S_\lambda\}_{\lambda\in\mathbb{N}}$, for any non-uniform PPT adversary $\mathcal{A}$, the advantage*

$$Adv^{pio\text{-}ind}_{pi\mathcal{O},S,\mathcal{A}}(\lambda) := \Pr\Big[Exp^{pio\text{-}ind}_{pi\mathcal{O},S\mathcal{A}}(\lambda) = 1\Big] - \frac{1}{2}$$

is negligible in λ, where $Exp^{pio\text{-}ind}_{pi\mathcal{O},S\mathcal{A}}(\lambda)$ is defined as in Fig. 3.

Experiment $Exp^{\text{pio-c}}_{C,z,\mathcal{D}}(\lambda)$	Experiment $Exp^{\text{pio-ind}}_{pi\mathcal{O},S,\mathcal{A}}(\lambda)$	Experiment $Exp^{\text{sel-ind}}_{S,\mathcal{A}}(\lambda)$
$C_0 := C$	$(C_0, C_1, z) \leftarrow S_\lambda$	$(x, st) \leftarrow \mathcal{A}_1(1^\lambda)$
$C_1 := pi\mathcal{O}(1^\lambda, C)$	$b \leftarrow \{0,1\}$	$(C_0, C_1, z) \leftarrow S_\lambda$, $b \leftarrow \{0,1\}$
$b \leftarrow \{0,1\}$	$\Lambda \leftarrow pi\mathcal{O}(1^\lambda, C_b)$	$y \leftarrow C_b(x;r)$// for fresh randomness r
$b' \leftarrow \mathcal{A}^{C_b(\cdot)}(1^\lambda, C, z)$	$b' \leftarrow \mathcal{A}(1^\lambda, C_0, C_1, \Lambda, z)$	$b' \leftarrow \mathcal{A}_2(1^\lambda, C_0, C_1, z, y, st)$
if $b' = b$ **then return** 1	**if** $b' = b$ **then return** 1	**if** $b' = b$ **then return** 1
return 0	**return** 0	**return** 0

Fig. 3. The descriptions of the games $Exp^{\text{pio-c}}_{C,z,\mathcal{D}}(\lambda)$ (left), $Exp^{\text{pio-ind}}_{pi\mathcal{O},S,\mathcal{A}}(\lambda)$ (middle), and $Exp^{\text{sel-ind}}_{S,\mathcal{A}}(\lambda)$ (right). In $Exp^{\text{pio-c}}_{C,z,\mathcal{D}}(\lambda)$, $\mathcal{D}$ has oracle access to either a probabilistic circuit C_0 using fresh randomness for every oracle query or to a deterministic circuit C_1. $\mathcal{D}$ can make an unbounded number of oracle queries with the restriction that no input is queried twice.

We remark that the construction proposed in [17] also satisfies our definition of correctness.

Let $X\colon \mathbb{N} \to \mathbb{N}$ be a function. For our purposes we use a class of circuit samplers, such that the sampled circuits are functionally equivalent for all inputs outside of a set $\mathcal{X}$, and the outputs of the circuits are indistinguishable for inputs inside of this set $\mathcal{X}$. The set $\mathcal{X}$ is a subset of the circuits' domain of cardinality at most $X(\lambda)$. Two circuits C_0 and C_1 are functionally equivalent if for any input x of matching length and any randomness r, $C_0(x;r) = C_1(x;r)$.

Definition 7 (X-Ind sampler, [3, 17]). *Let $X\colon \mathbb{N} \to \mathbb{N}$ be a function with $X(\lambda) \le 2^\lambda$, for all $\lambda \in \mathbb{N}$. The class $\mathcal{S}^{X\text{-}ind}$ of* X-Ind samplers *for a circuit family $\mathcal{C}$ contains all circuit samplers S for $\mathcal{C}$ satisfying, that for any $\lambda \in \mathbb{N}$, there exists a set $\mathcal{X} = \mathcal{X}_\lambda \subseteq \{0,1\}^*$ with $|\mathcal{X}| \le X(\lambda)$, such that the following two properties hold:*

X-differing inputs *For any (possibly unbounded) deterministic adversary $\mathcal{A}$, the advantage*

$$Adv^{eq\$}_{S,\mathcal{A}}(\lambda) := \Pr\left[C_0(x;r) \neq C_1(x;r) \wedge x \notin \mathcal{X} \;\middle|\; \begin{array}{l}(C_0, C_1, z) \leftarrow S_\lambda,\\ (x,r) \leftarrow \mathcal{A}(C_0, C_1, z)\end{array}\right]$$

is negligible in λ.

X-indistinguishability *For any non-uniform PPT distinguisher $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage*

$$X(\lambda) \cdot Adv^{sel\text{-}ind}_{S,\mathcal{A}}(\lambda) := X(\lambda) \cdot \left(\Pr\left[Exp^{sel\text{-}ind}_{S,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right)$$

is negligible in λ, where $Exp^{sel\text{-}ind}_{S,\mathcal{A}}(\lambda)$ is defined as in Fig. 3.

For our construction we use an obfuscator for the class $\mathcal{S}^{X\text{-ind}}$.

According to Theorem 2 in the proceedings of [17], a pIO which is secure with respect to $\mathcal{S}^{X\text{-ind}}$ for a circuit family $\mathcal{C}$ that only contains circuits of size at most λ can be obtained from sub-exponentially secure indistinguishability obfuscation (IO) for deterministic circuits in conjunction with sub-exponentially secure puncturable PRF. The construction given in [17] satisfies this security requirement even if the circuit family $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ contains circuits with polynomial size in λ as long as the input length of those circuits is at most λ.

2.8 Fully homomorphic encryption scheme

Let $\mathcal{C} = (\mathcal{C}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of sets of polynomial sized circuits of arity $a(\lambda)$, i.e. the set $\mathcal{C}_\lambda$ contains circuits of polynomial size in λ. We assume that for any $\lambda \in \mathbb{N}$ the circuits in $\mathcal{C}_\lambda$ share the common input domain $(\{0,1\}^{\mathsf{poly}(\lambda)})^{a(\lambda)}$ for a fixed polynomial $\mathsf{poly}(\lambda)$. A homomorphic encryption scheme enables evaluation of circuits on encrypted data. The first fully homomorphic encryption scheme was proposed in [29]. In this paper, we abide by the notation used in [3].

Definition 8 (Homomorphic public-key encryption (HPKE) scheme (syntax and security)). *A* homomorphic public-key encryption scheme *with message space $\mathcal{M} \subseteq \{0,1\}^*$ for a deterministic circuit family $\mathcal{C} = (\mathcal{C}_\lambda)_{\lambda\in\mathbb{N}}$ of arity $a(\lambda)$ and input domain $(\{0,1\}^{\mathsf{poly}(\lambda)})^{a(\lambda)}$ is a tuple of PPT algorithms* $\textsc{Hpke} = (\textsc{Gen}, \textsc{Enc}, \textsc{Dec}, \textsc{Eval})$.

$\textsc{Gen}(1^\lambda) \to (pk, sk)$ *On input the unary encoded security parameter 1^λ,* $\textsc{Gen}$ *outputs a public key pk and a secret key sk.*

$\textsc{Enc}(pk, m) \to c$ *On input the public key pk and a message $m \in \mathcal{M}$,* $\textsc{Enc}$ *outputs a ciphertext $c \in \{0,1\}^{\mathsf{poly}(\lambda)}$ for message m.*

$\textsc{Dec}(sk, c) \to m$ *On input the secret key sk and a ciphertext $c \in \{0,1\}^{\mathsf{poly}(\lambda)}$,* $\textsc{Dec}$ *outputs the corresponding message $m \in \mathcal{M}$ (or $\perp$, if the ciphertext is not valid).*

$\textsc{Eval}(pk, C, c_1, \ldots, c_{a(\lambda)}) \to c$ *On input the public key pk, a deterministic circuit $C \in \mathcal{C}_\lambda$, and ciphertexts $(c_1, \ldots, c_{a(\lambda)}) \in (\{0,1\}^{\mathsf{poly}(\lambda)})^{a(\lambda)}$,* $\textsc{Eval}$ *outputs a ciphertext $c \in \{0,1\}^{\mathsf{poly}(\lambda)}$.*

We require HPKE *to meet the following requirements:*

Perfect correctness *The triple* (GEN, ENC, DEC) *is perfectly correct as a PKE scheme, i.e. for any* $\lambda \in \mathbb{N}$*, any* $(pk,\, sk) \leftarrow \textsc{Gen}(1^\lambda)$*, any* $m \in \mathcal{M}$*, and any* $c \leftarrow \textsc{Enc}(pk,\, m)$*,* $\textsc{Dec}(sk,\, c) = m$*. Furthermore, the evaluation algorithm* EVAL *is perfectly correct in the sense that for any* $\lambda \in \mathbb{N}$*, any* $(pk,\, sk) \leftarrow \textsc{Gen}(1^\lambda)$*, any* $m_1, \ldots, m_{a(\lambda)} \in \mathcal{M}$*, any* $c_i \leftarrow \textsc{Enc}(pk,\, m_i)$*, any* $C \in \mathcal{C}_\lambda$*, and any* $c \leftarrow \textsc{Eval}(pk,\, C,\, c_1, \ldots, c_{a(\lambda)})$*,* $\textsc{Dec}(sk,\, c) = C(m_1, \ldots, m_{a(\lambda)})$*.*

Compactness *The size of the output of* EVAL *is polynomial in* λ *and independent of the size of the circuit* C*.*

Security *For any legitimate PPT adversary* $\mathcal{A}$*, the advantage*

$$Adv^{\textit{ind-cpa}}_{\textsc{Hpke},\mathcal{A}}(\lambda) := Exp^{\textit{ind-cpa}}_{\textsc{Hpke},\mathcal{A}}(\lambda) - \frac{1}{2}$$

is negligible in λ*, where* $Exp^{\textit{ind-cpa}}_{\textsc{Hpke},\mathcal{A}}$ *is defined as in Fig. 4. An adversary* $\mathcal{A}$ *is* legitimate *if it outputs two messages* m_0*,* m_1 *of identical length.*

EXPERIMENT $Exp^{\text{ind-cpa}}_{\textsc{Hpke},\mathcal{A}}(\lambda)$

$(pk, sk) \leftarrow \textsc{Gen}(1^\lambda)$, $(m_0, m_1, st) \leftarrow \mathcal{A}(1^\lambda, pk, \mathsf{find})$
$b \leftarrow \{0,1\}$, $c \leftarrow \textsc{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}(1^\lambda, c, st, \mathsf{attack})$
if $b' = b$ **then return** 1
return 0

Fig. 4. The description of the IND-CPA game $Exp^{\text{ind-cpa}}_{\textsc{Hpke},\mathcal{A}}(\lambda)$.

Without loss of generality, we assume that the secret key is the randomness that was used during the key generation. This enables to test whether key pairs are valid.

3 Construction

3.1 Group scheme

A group scheme is an abstraction from the properties of groups formalized via a tuple of PPT algorithms. For our purposes, we further abstract this notion to suit groups where group elements do not necessarily have unique encodings. We adapt the notion described in [3] which in turn generalizes the notion introduced in [11]. As demonstrated in [3], such group schemes benefit from the fact that group elements can be represented with many different encodings. This allows to add auxiliary information inside encodings of group elements in order to add more structure to the group. In our case, however, we exploit that group schemes with non-unique encodings can be used to conceal the structure of the group.

Definition 9 (Group scheme with non-unique encodings).
A group scheme with non-unique encodings Γ *is a tuple of PPT algorithms* $\Gamma = (\textsc{Setup}, \textsc{Val}, \textsc{Sam}, \textsc{Add}, \textsc{Equal})$.

$\textsc{Setup}(1^\lambda) \to pp$ *On input the unary encoded security parameter* 1^λ, Setup *outputs public parameters pp. In particular, pp contains the group order* q. *We assume that pp is given implicitly to the following algorithms.*

We assume that any encoding is represented as a bit string. In order to decide, whether a given bit string is a valid encoding of a group element, Γ *provides a validation algorithm* Val. *We refer to bit strings causing* Val *to output* 1 *as (valid) encodings of group elements.*

$\textsc{Val}(h) \to \{0,1\}$ *On input a bit string* $h \in \{0,1\}^*$, Val *outputs* 1 *if* h *is a valid encoding with respect to pp, otherwise* Val *outputs* 0.

In general, it is not sufficient to compare encodings as bit strings in order to decide whether they represent the same group element. Hence, a group scheme needs to define an algorithm that provides this functionality. This algorithm is called Equal. *We require* Equal *to realize an equivalence relation on the set of valid encodings. For any valid encoding* $h \in \{0,1\}^*$, *let* $\mathcal{G}(h)$ *denote the equivalence class of this encoding. In other words,* $\mathcal{G}(h)$ *contains all encodings that correspond to the same group element as the encoding* h. *For any valid encoding* h, *we require that* $|\{a \in \{0,1\}^* \mid \textsc{Val}(a) = 1\}/\mathcal{G}(h)| = q$ *is the order of the group. We refer to the equivalence classes in* $\{a \in \{0,1\}^* \mid \textsc{Val}(a) = 1\}/\mathcal{G}(h)$ *as group elements.*

$\textsc{Equal}(a,b) \to \{0,1,\perp\}$ *On input two valid encodings* a *and* b, Equal *outputs* 1 *if* a *and* b *represent the same group element, otherwise* Equal *outputs* 0. *If either* a *or* b *is invalid,* Equal *outputs* $\perp$.

In order to perform the group operation on two given encodings, we define an addition algorithm Add.

$\textsc{Add}(a,b)$ *On input two valid encodings* a *and* b, Add *outputs an encoding corresponding to the group element that results from the addition of the group elements represented by* a *and* b. *If either* a *or* b *is invalid,* Add *outputs* $\perp$.

The sampling algorithm Sam *enables to produce an encoding of a group element and only uses information that is part of the public parameters pp. Let* h *be a bit string produced via* $\textsc{Sam}(1)$. *For any* $z \in \mathbb{N}$, *let* $[z]$ *denote the group element corresponding to the equivalence class* $\mathcal{G}(h^z)$, *where the group operation is performed using* Add. *We require the distribution of* $\textsc{Sam}(z)$ *to be computationally indistinguishable from uniform distribution over* $[z]$.

$\textsc{Sam}(z) \to a$ *On input an exponent* $z \in \mathbb{N}$, Sam *outputs an encoding* a *from the equivalence class* $\mathcal{G}(h^z)$.

Given the order q of the group, it is sufficient to provide an addition algorithm to enable inversion of group elements. To invert a given group element, we use the square-and-multiply approach to add the given encoding $q-1$ times to itself. Further, it suffices to define an algorithm Zero that tests whether a given encoding corresponds to the identity element of the group instead of an algorithm Equal as above. To implement the algorithm Equal on input two encodings a and b, we invert b, add the result to a and test whether the result corresponds to the identity element using Zero.

According to [3], a group scheme with non-unique encodings, in addition to the algorithms defined above, provides an extraction algorithm. The extraction algorithm, given a valid encoding, produces a bit string such that all encodings that represent the same group element lead to the same bit string. However, we omit this algorithm, as our construction does not provide one. It remains an open problem to extend our construction with an extraction algorithm such that the validity of the (m, n)-Interactive Uber assumption (see Definition 10) can still be proven.

3.2 Interactive Uber assumption

The Uber assumption is a very strong cryptographic assumption in bilinear groups first proposed in [9] and refined in [12]. It provides a natural framework that enables to assess the plausibility of cryptographic assumptions in bilinear groups.

In contrast to the original definition, we consider adaptive attacks (in which an adversary may ask adaptively for more information about the game secrets and choose his challenge).

Definition 10 ((m, n)-Interactive Uber assumption for group schemes). *Let $m = m(\lambda)$ and $n = n(\lambda)$ such that $d := \binom{n+m}{m}$ is a polynomial*[5] *in λ, and let Γ be a group scheme. The (m, n)-*Interactive Uber assumption *holds for Γ if for any legitimate PPT adversary $\mathcal{A}$, the advantage $Adv^{uber}_{\Gamma,\mathcal{A}}(\lambda)$ is negligible in λ, where*

$$Adv^{uber}_{\Gamma,\mathcal{A}}(\lambda) := \Pr\big[Exp^{uber}_{\Gamma,\mathcal{A}}(\lambda) = 1\big] - \frac{1}{2}.$$

The game $Exp^{uber}_{\Gamma,\mathcal{A}}(\lambda)$ is described in Fig. 5. An adversary $\mathcal{A}$ is legitimate*, if and only if it always guarantees $P^*(\boldsymbol{X}) \notin \langle 1,\ P_1(\boldsymbol{X}),\ \ldots,\ P_l(\boldsymbol{X})\rangle$ and for any $P(\boldsymbol{X}) \in \{P^*(\boldsymbol{X}),\ P_1(\boldsymbol{X}),\ \ldots,\ P_l(\boldsymbol{X})\}$, $\deg(P(\boldsymbol{X})) \leq n$ in $Exp^{uber}_{\Gamma,\mathcal{A}}(\lambda)$, where $\{P_1(\boldsymbol{X}),\ \ldots,\ P_l(\boldsymbol{X})\}$ are the polynomials that $\mathcal{A}$ requests from its oracle $\mathcal{O}$.*

For technical reasons, we need the maximum total degree n of the polynomials appearing in $Exp^{\mathrm{uber}}_{\Gamma,\mathcal{A}}(\lambda)$ and the number of unknowns m to be bounded a priori.

3.3 Our construction

Inspired by the construction in [3], an encoding of a group element includes two ciphertexts each encrypting a vector determining an m-variate polynomial over $\mathbb{Z}_q$ of maximum total degree n with respect to some randomly sampled basis $\{a_1,\ \ldots,\ a_d\}$. That basis is hidden inside the public parameters of the group scheme via a perfectly binding commitment. An encoding corresponds to the group element whose discrete logarithm equals the evaluation of the thus

[5] If the parameters m and n both grow at most logarithmically in λ or one of them grows polynomially in λ while the other one is a constant, the binomial coefficient $d = \binom{n+m}{m}$ grows polynomially in λ.

Experiment $Exp^{\mathrm{uber}}_{\Gamma,\mathcal{A}}(\lambda)$	Oracle $\mathcal{O}(P(\boldsymbol{X}))$
$pp \leftarrow \textsc{Setup}(1^\lambda)$, $\boldsymbol{s} \leftarrow (\mathbb{Z}_q)^m$	**return** $\textsc{Sam}(P(\boldsymbol{s}))$
$(P^*(\boldsymbol{X}), st) \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(1^\lambda, pp, \mathsf{find})$	
$b \leftarrow \{0,1\}$, $r \leftarrow \mathbb{Z}_q$	
$z_0 \leftarrow \textsc{Sam}(P^*(\boldsymbol{s}))$, $z_1 \leftarrow \textsc{Sam}(r)$	
$b' \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(1^\lambda, z_b, st, \mathsf{attack})$	
if $b = b'$ **then return** 1	
return 0	

Fig. 5. The description of the (m,n)-Interactive Uber game $Exp^{\mathrm{uber}}_{\Gamma,\mathcal{A}}(\lambda)$. The oracle $\mathcal{O}$ on input a polynomial $P(\boldsymbol{X})$, returns an encoding of the group element $[P(\boldsymbol{s})]$. We refer to $P^*(\boldsymbol{X})$ as "challenge polynomial" and to z_b as "challenge encoding". Further, we call the polynomials that $\mathcal{A}$ requests from the oracle $\mathcal{O}$ "query polynomials".

determined polynomial at a random point $\boldsymbol{\omega} \in \mathbb{Z}_q^m$. That random point $\boldsymbol{\omega}$ is fixed in the public parameters via a point obfuscation po.

For our construction we employ the following building blocks: **(i)** a dual mode NIWI proof system Π, **(ii)** a homomorphic encryption scheme Hpke with message space $\mathcal{M} = \mathbb{Z}_q^d$ for a family of circuits of arity $a(\lambda) = 2$ adding two tuples in $\mathbb{Z}_q^d$ component-by-component modulo q, **(iii)** a point obfuscation POBF for message space $\mathcal{M}_k = \mathbb{Z}_q$, **(iv)** a family $\mathcal{TD} = (\mathcal{TD}_\lambda)_{\lambda\in\mathbb{N}}$ of families $\mathcal{TD}_\lambda$ of languages TD in a universe $\mathcal{X} = \mathcal{X}_\lambda$ with unique witnesses for $y \in \mathrm{TD}$ such that the subset membership problem $\mathrm{TD} \subseteq \mathcal{X}$ is hard, **(v)** a perfectly binding non-interactive commitment scheme Com for message space $\mathbb{Z}_q^{d\times d}$, and **(vi)** a general purpose X-Ind pIO $pi\mathcal{O}$ (i.e. a pIO that is secure with respect to $\mathcal{S}^{X\text{-ind}}$ for a circuit family that only contains circuits with input length at most l, where l is the security parameter used for $pi\mathcal{O}$). Let $n = n(\lambda)$ and let $m = m(\lambda)$ such that $\binom{n+m}{m}$ is a polynomial in λ. The group scheme we construct depends on n and m. We emphasize this fact by calling it $\Gamma_{m,n} := (\textsc{Setup}, \textsc{Val}, \textsc{Sam}, \textsc{Add}, \textsc{Equal})$. As mentioned above, we provide an algorithm that tests if a given encoding is an encoding of the identity group element, instead of implementing Equal.

In Fig. 6 we describe the algorithm Setup of our construction. The number q is a prime number that is greater than $2^{p(\lambda)}$ and will serve as the order of our group scheme. We require p to be a polynomial such that $p(\lambda) \geq poly(\lambda)$, where $poly$ is used to scale the security parameter of $pi\mathcal{O}$. We emphasize that our construction allows to arbitrarily choose the group order q as long as q is greater than $2^{p(\lambda)}$ and prime. Therefore, q can be understood as an input of the algorithm Setup. For the sake of simplicity, we do not write q as input and assume that Setup generates a suitable group order.

We remark that the circuits C_{Add} and $C^{(0)}_{\mathsf{Zero}}$ that appear in the algorithm Setup implement the addition of two group elements and a test for the identity element respectively. For a description of these circuits we refer the reader to Fig. 7. The polynomial $poly(\lambda) \geq \lambda$ that is used to scale the security parameter

Algorithm $\textsc{Setup}(1^\lambda)$

$(gpk, gsk) \leftarrow \mathsf{Setup}_\Pi(1^\lambda)$
$(pk, sk) \leftarrow \textsc{Gen}(1^\lambda)$, $(pk', sk') \leftarrow \textsc{Gen}(1^\lambda)$
$\boldsymbol{\omega} \leftarrow (\mathbb{Z}_q)^m$, $\mathsf{po}_i \leftarrow \textsc{PObf}(1^\lambda, \omega_i)$ for $1 \le i \le m$, $\mathsf{po} := (\mathsf{po}_1, \dots, \mathsf{po}_m)$
$\mathrm{TD} \leftarrow \mathcal{TD}_\lambda$, $y \leftarrow \mathcal{X} \setminus \mathrm{TD}$
$A \leftarrow \{B \in \mathrm{GL}_d(\mathbb{Z}_q) \mid B \cdot e_1 = e_1\}$
$ck \leftarrow \textsc{ComSetup}(1^\lambda)$, $(com, op) \leftarrow \textsc{Commit}_{ck}(A)$
$(crs, td_{\mathrm{ext}}) \leftarrow \mathsf{K}(gpk, gsk)$
$\Lambda_{\mathrm{add}} \leftarrow pi\mathcal{O}(1^{poly(\lambda)}, C_{\mathsf{Add}})$, $\Lambda_{\mathrm{zero}} \leftarrow pi\mathcal{O}(1^{poly(\lambda)}, C^{(0)}_{\mathsf{Zero}})$
return $pp := (q,\, gpk,\, crs,\, y,\, \mathrm{TD},\, pk,\, pk',\, \Lambda_{\mathrm{add}},\, \Lambda_{\mathrm{zero}},\, \mathsf{po},\, ck,\, com)$

Fig. 6. The implementation of the $\textsc{Setup}$ algorithm producing public parameters pp.

for the obfuscator $pi\mathcal{O}$ upper bounds the input length of these circuits C_{Add} and $C^{(0)}_{\mathsf{Zero}}$. All versions of addition circuits and all versions zero testing circuits that appear during the proofs are padded to the same length respectively. We emphasize that it is necessary to scale the used security parameter as the pIO $pi\mathcal{O}$ we rely on is secure with respect to $\mathcal{S}^{X\text{-ind}}$ for a circuit family that only contains circuits with input length at most λ', where λ' denotes the security parameter that is used to invoke $pi\mathcal{O}$.

Encodings of group elements Encodings of group elements are of the form $h = (C, C', \pi)$. The first two entries C and C' are ciphertexts encrypting vectors $\overrightarrow{f} \in \mathbb{Z}_q^d$ and $\overrightarrow{f'} \in \mathbb{Z}_q^d$ respectively under the public keys pk and pk' respectively, where d is the dimension of the $\mathbb{Z}_q$ vector space of m-variate polynomials over $\mathbb{Z}_q$ with total degree at most n, i.e. $d = \binom{n+m}{m}$. We require the dimension d of the vector space to grow at most polynomially in λ. The last entry π is the so-called consistency proof. We refer to the vectors $\overrightarrow{f}$ and $\overrightarrow{f'}$ as *representation vectors* of the group element and to the tuple $(\overrightarrow{f}, \overrightarrow{f'})$ as *representation* of the group element. Let $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{N}^m$ denote tuples with $\sum_{i=1}^m \alpha_i \le n$ and let

$$\varphi_{\mathrm{pol}} \colon \mathbb{Z}_q^d \to \mathbb{Z}_q[\boldsymbol{X}],\ (\dots, v_\alpha, \dots)^T \mapsto \sum_\alpha v_\alpha \cdot X_1^{\alpha_1} \cdots X_m^{\alpha_m}$$

be the vector space homomorphism mapping the standard basis of $\mathbb{Z}_q^d$ to a natural basis of the vector space of m-variate polynomials of degree at most n. For well-definedness we use the lexicographical order on the tuples $(\alpha_1, \dots, \alpha_m) \in \mathbb{N}^m$, particularly, the first vector of the standard basis of $\mathbb{Z}_q^d$ is mapped to the constant polynomial 1. The image of φ_{pol} is $\mathrm{Im}(\varphi_{\mathrm{pol}}) = \{p \in \mathbb{Z}_q[\boldsymbol{X}] \mid \deg(p) \le n\}$ and the kernel is $\ker(\varphi_{\mathrm{pol}}) = \{0\}$. Hence, φ_{pol} is an isomorphism between the vector spaces $\mathbb{Z}_q^d$ and $\mathrm{Im}(\varphi_{\mathrm{pol}})$.

We recall that $\textsc{Setup}(1^\lambda)$ samples the matrix A uniformly at random from $\mathrm{GL}_d(\mathbb{Z}_q)$ such that the first column equals e_1. Hence, the matrix A^{-1} exists and has the form $A^{-1} = (a_1 \mid a_2 \mid \dots \mid a_d)$ such that $a_1 = e_1$. The columns $a_1, \dots, a_d \in \mathbb{Z}_q^d$ form a basis of the vector space $\mathbb{Z}_q^d$.

The coefficients of the representation vectors $\overrightarrow{f} = (f_1, \ldots, f_d)^T$ and $\overrightarrow{f'} = (f'_1, \ldots, f'_d)^T$ of a group element define the polynomials $f(\boldsymbol{X}),\ f'(\boldsymbol{X}) \in \mathrm{Im}(\varphi_{\mathrm{pol}})$ via

$$\begin{aligned} f(\boldsymbol{X}) :=& \sum_{i=1}^{d} f_i \cdot \varphi_{\mathrm{pol}}(a_i) \qquad & f'(\boldsymbol{X}) :=& \sum_{i=1}^{d} f'_i \cdot \varphi_{\mathrm{pol}}(a_i) \\ =& \varphi_{\mathrm{pol}}\left(A^{-1} \cdot \overrightarrow{f}\right) & =& \varphi_{\mathrm{pol}}\left(A^{-1} \cdot \overrightarrow{f'}\right) \end{aligned}$$

In other words, the representation vectors $\overrightarrow{f}$ and $\overrightarrow{f'}$ are the representations of the abstract polynomials $f(\boldsymbol{X})$ and $f'(\boldsymbol{X})$ respective to the basis $\{\varphi_{\mathrm{pol}}(a_1) = \varphi_{\mathrm{pol}}(e_1), \varphi_{\mathrm{pol}}(a_2), \ldots, \varphi_{\mathrm{pol}}(a_d)\}$. Intuitively, a valid encoding that contains the representation vector $\overrightarrow{f} \in \mathbb{Z}_q^d$ corresponds to the group element $[f(\boldsymbol{\omega})]$, where $\boldsymbol{\omega}$ is the value that is fixed in the public parameters of the group scheme via po. The same holds for the representation vector $\overrightarrow{f'}$ resulting in a redundant encoding. This approach is similar to the Naor-Yung paradigm [36].

We call the representation $(\overrightarrow{f}, \overrightarrow{f'})$ *consistent* if both representation vectors correspond to the same group element, i.e. the evaluation of the corresponding polynomials $f(\boldsymbol{X})$ and $f'(\boldsymbol{X})$ at $\boldsymbol{\omega}$ are equal. Otherwise, we call such a representation *inconsistent*. If the representation $(\overrightarrow{f}, \overrightarrow{f'})$ is consistent, we call this representation *constant* if the corresponding polynomials $f(\boldsymbol{X})$ and $f'(\boldsymbol{X})$ are constant (i.e. are of total degree at most 0). If a consistent representation is not constant we call this representation *non-constant*. The purpose of the so-called consistency proof is to ensure consistency of encodings, i.e. to ensure that the corresponding representation is consistent. Further, we use the terms constant, non-constant, consistent, and inconsistent to characterize encodings if the associated representation has the respective properties.

Consistency proof and validation algorithm The above mentioned *consistency proof* ensures that the representations, that are encrypted inside of encodings, are consistent. In other words, the consistency proof ensures that both representation vectors $\overrightarrow{f}$ and $\overrightarrow{f'}$ used for an encoding lead to the same group element. We realize this by using the dual mode NIWI proof system $\mathit{\Pi}$ to produce the consistency proof π for a relation $\mathcal{R}$. The relation $\mathcal{R}$ is a disjunction of three main statements $\mathcal{R} = \mathcal{R}_1 \vee \mathcal{R}_2 \vee \mathcal{R}_3$:

The relation $\mathcal{R}_1$ is satisfied for representations that are constant and consistent. We formalize this via relation $\mathcal{R}_{1.a}$:

$$\mathcal{R}_{1.a} := \left[\overrightarrow{f} = \overrightarrow{f'} \wedge \deg\left(\varphi_{\mathrm{pol}}(\overrightarrow{f})\right) \leq 0\right]$$

We recall the convention that the degree of the zero polynomial is defined to be $-\infty$. For technical reasons, we need to make sure that the knowledge of the secret decryption keys (sk, sk') and the knowledge of the used encryption randomness are both sufficient as witnesses. Thus, additionally to $\mathcal{R}_{1.a}$ we define the two relations $\mathcal{R}_b$ and $\mathcal{R}_c$. The relations $\mathcal{R}_b$ and $\mathcal{R}_c$ connect the ciphertexts C, C' of the encoding with the corresponding representation vectors $\vec{f}$, $\vec{f'}$ appearing in relation $\mathcal{R}_{1.a}$.

$$\mathcal{R}_b := \left[\quad C = \textsc{Enc}(pk, \vec{f}; R) \wedge C' = \textsc{Enc}(pk', \vec{f'}; R') \quad \right]$$
$$\mathcal{R}_c := \left[\begin{array}{c} (pk, sk) = \textsc{Gen}(sk) \ \wedge \ \vec{f} = \textsc{Dec}(sk, C) \quad \wedge \\ (pk', sk') = \textsc{Gen}(sk') \wedge \vec{f'} = \textsc{Dec}(sk', C') \end{array} \right]$$

At this point we make use of the assumption that a secret decryption key equals the randomness that was used to produce the corresponding public encryption key. The relation $\mathcal{R}_1$ is defined as follows:

$$\mathcal{R}_1 := \mathcal{R}_{1.a} \wedge (\mathcal{R}_b \vee \mathcal{R}_c)\,. \tag{7}$$

Given a consistent and constant representation $(\vec{f}, \vec{f'})$ and resulting ciphertexts C and C', there are two possible witnesses to produce the consistency proof for the relation $\mathcal{R}_1$: using the secret decryption keys $(sk,\ sk',\ \vec{f},\ \vec{f'})$ and using the encryption randomness $((\vec{f},\ R),\ (\vec{f'},\ R'))$.

The relation $\mathcal{R}_2$ is satisfied for representations that are consistent. Again, we formalize this via a relation $\mathcal{R}_{2.a}$:

$$\mathcal{R}_{2.a} := \left[\begin{array}{lll} \varphi_{\mathrm{pol}}\left(A^{-1} \cdot \vec{f}\right)(\boldsymbol{\omega}) & = \varphi_{\mathrm{pol}}\left(A^{-1} \cdot \vec{f'}\right)(\boldsymbol{\omega}) & \wedge \\ \forall i \in \{1, \dots, m\} \colon \mathsf{po}_i(\omega_i) = 1 & & \wedge \\ \textsc{Open}_{ck}(com, op) & = A \ \wedge \ A \neq \bot & \end{array} \right]$$

The relation $\mathcal{R}_2$ is defined as follows:

$$\mathcal{R}_2 := \mathcal{R}_{2.a} \wedge (\mathcal{R}_b \vee \mathcal{R}_c)\,. \tag{8}$$

Given a consistent representation $(\vec{f}, \vec{f'})$ and resulting ciphertexts C and C', there are two possible witnesses to produce the consistency proof for the relation $\mathcal{R}_2$: using the secret decryption keys $(sk,\ sk',\ \vec{f},\ \vec{f'},\ \boldsymbol{\omega},\ op)$ and using the encryption randomness $((\vec{f},\ R),\ (\vec{f'},\ R'),\ \boldsymbol{\omega},\ op)$. To be precise, the matrix A also is part of these witnesses. However, as we can assume that A is a part of op, we omit this fact in our notation.

The relation $\mathcal{R}_3$ introduces a trapdoor enabling production of consistency proofs for inconsistent encodings.

$$\mathcal{R}_3 := \big[y \in \mathrm{TD}\big]\,. \tag{9}$$

This relation only depends on the instance (TD, y) of the subset membership problem $\mathrm{TD} \subseteq \mathcal{X}$ defined in the public parameters. We recall that if $y \in \mathrm{TD}$, there exists a unique witness w_y satisfying the witness relation for the SMP. Hence, the witness for the relation $\mathcal{R}_3$ is (w_y). Given public parameters pp that are generated via $\textsc{Setup}(1^\lambda)$, y is not in TD. Therefore, there exists no trapdoor if pp is generated honestly.

Let rp denote the parts of the public parameters that are necessary to produce consistency proofs, i.e. $rp := (q,\ pk,\ pk',\ \mathsf{po},\ ck,\ com,\ \mathrm{TD},\ y)$. To be precise, the corresponding language L has the following form:

$$
\begin{aligned}
L :=& \{x = (\underbrace{q, pk, pk', \mathsf{po}, ck, com, \mathrm{TD}, y}_{=rp}, C, C') \mid \exists w : (x, w) \in \mathcal{R}\} \\
=& L_1 \cup L_2 \cup L_3,
\end{aligned}
$$

where $L_i := \{x = (rp, C, C') \mid \exists w\colon (x, w) \in \mathcal{R}_i\}$. For the sake of clarity, we henceforth omit the parameters rp and treat the tuple $(C,\ C')$ as the statement.

The validation algorithm VAL, on input a bit string $h \in \{0,1\}^*$, parses h into $(C,\ C',\ \pi)$ and executes $\mathsf{Verify}(gpk,\ crs,\ x,\ \pi)$ of the underlying NIWI proof system Π for the relation $\mathcal{R}$.

Addition and Zero Algorithm The implementations of the algorithms ADD and ZERO need to know secret information that is associated with the public parameters, for instance the secret decryption keys. Therefore, we implement these algorithms as probabilistic circuits and "hard-code" the necessary secret parameters inside. The security requirement of the employed obfuscator $pi\mathcal{O}$ enables to conceal the implementation of these circuits and, hence, conceals the secret parameters that are hard-coded. The PPT algorithms ADD and ZERO simply execute the respective obfuscated circuit Λ_{add} and Λ_{zero}.

In Fig. 7 we present the implementation of the circuit C_{Add} and the implementation of the circuit $C^{(0)}_{\mathsf{Zero}}$. We remark that C_{Zero} only uses the representation vector $\vec{f}$ and ignores the representation vector $\vec{f'}$. This enables to exploit the Naor-Yung like double encryption.

The addition circuit C_{Add} is similar to the one constructed in [3]. The difference is limited to the fact that in our case C_{Add} differentiates between three instead of two different possibilities to produce the new consistency proof. The encodings of group elements in the construction of [3] are of the form $(h,\ C,\ C',\ \pi)$, where C and C' are some ciphertexts and π is a corresponding consistency proof. The value h is the group element in an underlying group that is represented by the encoding. As h uniquely identifies the represented group element, the equality test simply compares these values of the given encodings. In our case, however, the encodings do not contain a similar entry. Therefore, the implementation of the equality test, or rather the zero test, needs to decrypt the ciphertext C in order to be able to make a statement about the represented group element.

Sampling Algorithm The sampling algorithm SAM, on input an exponent $z \in \mathbb{N}$, uses the representation $(\vec{f},\ \vec{f'}) := ((z,\ 0,\ \dots,\ 0)^T,\ (z,\ 0,\ \dots,\ 0)^T)$ to produce an encoding of the requested group element. The consistency proof is produced for relation $\mathcal{R}_1$ using the witness $((\vec{f},\ R),\ (\vec{f'},\ R'))$, where R and R' are the randomnesses that are used to encrypt $\vec{f}$ and $\vec{f'}$ respectively. If the sampling algorithm does not receive any input, it samples the exponent z from

$\textsc{Circuit}\ C_{\mathsf{Add}}[gpk, rp, sk, sk', \boldsymbol{\omega}, op, td_{\text{ext}}](a, b)$	$\textsc{Circuit}\ C^{(0)}_{\mathsf{Zero}}[q, sk, \boldsymbol{\omega}, A](a)$
if $\neg\textsc{Val}(a) \vee \neg\textsc{Val}(b)$ **then return** $\perp$	**if** $\neg\textsc{Val}(a)$ **then**
parse $a =: (C^{(a)}, C'^{(a)}, \pi^{(a)})$	**return** $\perp$
parse $b =: (C^{(b)}, C'^{(b)}, \pi^{(b)})$	parse $a =: (C, C', \pi)$
$C^{(c)} := \textsc{Eval}(pk, \oplus, C^{(a)}, C^{(b)})$	$\vec{f} \leftarrow \textsc{Dec}(sk, C)$
$C'^{(c)} := \textsc{Eval}(pk', \oplus, C'^{(a)}, C'^{(b)})$	$f(\boldsymbol{X}) := \varphi_{\text{pol}}(A^{-1} \cdot \vec{f})$
$\vec{f}^{(a)} := \textsc{Dec}(sk, C^{(a)}), \vec{f}'^{(a)} := \textsc{Dec}(sk', C'^{(a)})$	**if** $f(\boldsymbol{\omega}) = 0$ **then**
$\vec{f}^{(b)} := \textsc{Dec}(sk, C^{(b)}), \vec{f}'^{(b)} := \textsc{Dec}(sk', C'^{(b)})$	**return** 1
$\vec{f}^{(c)} := \oplus(\vec{f}^{(a)}, \vec{f}^{(b)}), \vec{f}'^{(c)} := \oplus(\vec{f}'^{(a)}, \vec{f}'^{(b)})$	**return** 0
if $(C^{(a)}, C'^{(a)}), (C^{(b)}, C'^{(b)}) \in L_1$ **then**	
$\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}), (sk, sk', \vec{f}^{(c)}, \vec{f}'^{(c)}))$	
elseif $(C^{(a)}, C'^{(a)}), (C^{(b)}, C'^{(b)}) \in L_2$ **then**	
$\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}), (sk, sk', \vec{f}^{(c)}, \vec{f}'^{(c)}, \boldsymbol{\omega}, op))$	
else	
let $\alpha \in \{a, b\} :\ (C^{(\alpha)}, C'^{(\alpha)}) \notin L_1 \cup L_2$	
$w_y \leftarrow \mathsf{Extract}(td_{\text{ext}}, (C^{(\alpha)}, C'^{(\alpha)}), \pi^{(\alpha)})$	
$\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}), (w_y))$	
return $c := (C^{(c)}, C'^{(c)}, \pi^{(c)})$	

Fig. 7. Circuit C_{Add} (left) for addition of two group elements, and circuit $C^{(0)}_{\mathsf{Zero}}$ (right) for testing whether a given encoding is an encoding of the identity element. Additionally to the publicly available parameters gpk and rp, C_{Add} has the secret decryption keys sk, sk', the values $\boldsymbol{\omega}$, the opening op, and the extraction trapdoor td_{ext} hard-coded. The circuit $C^{(0)}_{\mathsf{Zero}}$ knows the publicly available parameter q and additionally has the secret parameters sk, $\boldsymbol{\omega}$, and A hard-coded. The circuit $\oplus$ realizes addition in $\mathbb{Z}_q^d$.

$\{0, \ldots, q-1\}$ uniformly at random and proceeds as above. Due to the IND-CPA security of Hpke, the distribution of the output of $\textsc{Sam}(z)$ is computationally indistinguishable from uniform distribution over the equivalence class $\mathcal{G}(\textsc{Sam}(z))$ (see Lemma 2).

We remark that our group scheme allows for re-randomization of encodings. To re-randomize a given encoding, we sample an encoding of the identity element and use the addition algorithm to add it to the encoding to be randomized. We require the employed homomorphic encryption scheme to satisfy an additional natural property. Namely, we require that ciphertexts can be re-randomized by homomorphically adding a fresh ciphertext of 0. This property is also known as circuit privacy.

3.4 Main theorem

Theorem 1. *Let $\Gamma_{m,n}$ be the group scheme constructed in Section 3.3. Further, let $pi\mathcal{O}$ be a probabilistic indistinguishability obfuscator with respect to $\mathcal{S}^{X\text{-}ind}$ for a circuit family containing circuits with input length at most $poly(\lambda)$, let*

$\mathcal{TD} = (\mathcal{TD}_\lambda)_{\lambda\in\mathbb{N}}$ *be a family of families* $\mathcal{TD}_\lambda = \{TD\}$ *of languages* $TD \subseteq \mathcal{X}_\lambda$ *such that the subset membership problem is hard, let* Π *be a dual mode NIWI proof system, let* HPKE *be an IND-CPA secure HPKE scheme, let* COM *be a perfectly binding non-interactive commitment scheme, and let* POBF *be a point obfuscation. Then, the* (m, n)*-Interactive Uber assumption (cf. Definition 10) holds for* $\Gamma_{m,n}$.

In Table 1 we give an overview on the proof of Theorem 1. The distribution $\widetilde{pp}$ denotes the distribution of public parameters that are sampled according to SETUP with the difference that y is sampled from within the trapdoor language TD. The distribution $\widehat{pp}$ denotes the same distribution as $\widetilde{pp}$ with the difference that the CRS is sampled in hiding mode and Λ_{add} is computed for an addition circuit that simulates consistency proofs and, hence, does not need to know the matrix A or the value $\boldsymbol{\omega}$. The distribution $\overline{pp}^{(i)}$ (for $i \in \{0, \ldots, m\}$) denotes the same distribution as $\widehat{pp}$ with the difference that Λ_{zero} is an obfuscation of a zero testing circuit that tests whether the polynomial $f(X_1, \ldots, X_i, \omega_{i+1}, \ldots, \omega_m)$ equals the zero polynomial. Furthermore, the point obfuscations in $\widehat{pp}$ obfuscate $\perp$ whereas the point obfuscations in $\overline{pp}^{(i)}$ obfuscate the values $\omega_{i+1}, \ldots, \omega_m$. The distribution $\underline{pp}$ is the same as $\overline{pp}^{(m)}$ with the difference that Λ_{zero} is produced for a zero testing circuit that simply tests whether the representation vector $\overrightarrow{f}$ equals zero in $\mathbb{Z}_q^d$ and, hence, does not need to know the matrix A and $\boldsymbol{\omega}$ anymore. For the formal definitions of pp, $\widetilde{pp}$, $\widehat{pp}$, $\overline{pp}^{(i)}$ (for $i \in \{0, \ldots, m\}$), and $\underline{pp}$ we refer the reader to Supplementary Sections A and B.

Table 1. An overview on the steps of the proof of Theorem 1. The $\boxed{\text{boxes}}$ emphasize changes compared to the previous game. Let W_i denote the witness that is used to prove relation $\mathcal{R}_i$ for $i \in \{1, 2, 3\}$. The witnesses W_1 and W_2 contain the used encryption randomness. Further, for a polynomial $P(\boldsymbol{X})$, let $R_P := A \cdot \varphi_{\mathrm{pol}}^{-1}(P(\boldsymbol{c} \circ \boldsymbol{X}))$, and for a vector $v^* \in \mathbb{Z}_q^d$, let $\overline{R}_{v^*} := \varphi_{\mathrm{pol}}\left(A^{-1} \cdot v^*\right)(\boldsymbol{\omega}) \cdot e_1$.

	Publ. param.	**Secret** s	**Representations for queries** P	**/ challenge** P^*	**Witness for** π	**Remark**
Game_0	pp	$\boldsymbol{s} \leftarrow \mathbb{Z}_q^m$	$P(\boldsymbol{s}) \cdot e_1$	$P^*(\boldsymbol{s}) \cdot e_1$	W_1	the real Uber game
Game_1	pp	$\boxed{\begin{array}{l}\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega} \\ \boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m\end{array}}$	$P(\boldsymbol{s}) \cdot e_1$	$P^*(\boldsymbol{s}) \cdot e_1$	W_1	negl. statistical distance
Game_2	pp	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	$\boxed{R_P}$	$\boxed{R_{P^*}}$	W_1, W_2	Switching lemma (Lemma 2)
Game_3	$\boxed{\widetilde{pp}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	W_1, W_2	SMP TD $\subseteq \mathcal{X}$
Game_4	$\boxed{\widehat{pp}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	W_1, W_2	Swap lemma (Lemma 1)
Game_5	$\widehat{pp}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	$\boxed{W_3}$	perfect WI of Π
Game_6	$\boxed{\overline{pp}^{(0)}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	W_3	hiding property of Com
Game_7	$\boxed{\overline{pp}^{(m)}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	W_3	Lemma 5
Game_8	$\boxed{\underline{pp}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	R_{P^*}	W_3	security of $pi\mathcal{O}$
Game_9	$\underline{pp}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$\boxed{v^* \leftarrow \mathbb{Z}_q^d}$	W_3	Rand. lemma (Lemma 3)
Game_{10}	$\boxed{\overline{pp}^{(m)}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	W_3	security of $pi\mathcal{O}$
Game_{11}	$\boxed{\overline{pp}^{(0)}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	W_3	Lemma 5
Game_{12}	$\boxed{\widehat{pp}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	W_3	hiding property of Com
Game_{13}	$\widehat{pp}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	$\boxed{W_1, W_2}$	perfect WI of Π
Game_{14}	$\boxed{\widetilde{pp}}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	W_1, W_2	Swap lemma (Lemma 1)
Game_{15}	$\boxed{pp}$	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	R_P	$v^* \leftarrow \mathbb{Z}_q^d$	W_1, W_2	SMP TD $\subseteq \mathcal{X}$
Game_{16}	pp	$\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$ $\boldsymbol{c} \leftarrow \left(\mathbb{Z}_q^\times\right)^m$	$\boxed{P(\boldsymbol{s}) \cdot e_1}$	$\boxed{\begin{array}{l}\overline{R}_{v^*}, \\ v^* \leftarrow \mathbb{Z}_q^d\end{array}}$	W_1	Switching lemma (Lemma 2)
Game_{17}	pp	$\boxed{\boldsymbol{s} \leftarrow \mathbb{Z}_q^m}$	$P(\boldsymbol{s}) \cdot e_1$	$\overline{R}_{v^*}$, $v^* \leftarrow \mathbb{Z}_q^d$	W_1	negl. statistical distance
Game_{18}	pp	$\boldsymbol{s} \leftarrow \mathbb{Z}_q^m$	$P(\boldsymbol{s}) \cdot e_1$	$\boxed{\begin{array}{l}r \cdot e_1, \\ r \leftarrow \mathbb{Z}_q\end{array}}$	W_1	identically distributed

References

[1] Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. "Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups". In: *Proc. CRYPTO 2011*. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 649–666.

[2] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. "Unified, Minimal and Selectively Randomizable Structure-Preserving Signatures". In: *Proc. TCC 2014*. Vol. 8349. Lecture Notes in Computer Science. Springer, 2014, pp. 688–712.

[3] Martin R Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, and Kenneth G Paterson. "Multilinear maps from obfuscation". In: *Theory of Cryptography*. Springer, 2016, pp. 446–473.

[4] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. "Protecting Obfuscation against Algebraic Attacks". In: *Proc. EUROCRYPT 2014*. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 221–238.

[5] Mihir Bellare and Adriana Palacio. "The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols". In: *Proc. CRYPTO 2004*. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 273–289.

[6] Mihir Bellare and Igors Stepanovs. "Point-function obfuscation: a framework and generic constructions". In: *Theory of Cryptography Conference*. Springer. 2016, pp. 565–594.

[7] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. "On virtual grey box obfuscation for general circuits". In: *International Cryptology Conference*. Springer. 2014, pp. 108–125.

[8] Dan Boneh and Xavier Boyen. "Short Signatures Without Random Oracles". In: *Proc. EUROCRYPT 2004*. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 56–73.

[9] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. "Hierarchical identity based encryption with constant size ciphertext". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2005, pp. 440–456.

[10] Dan Boneh and Alice Silverberg. "Applications of Multilinear Forms to Cryptography". In: *Contemporary Mathematics* 324 (2003), pp. 71–90.

[11] Dan Boneh and Alice Silverberg. "Applications of multilinear forms to cryptography". In: *Contemporary Mathematics* 324.1 (2003), pp. 71–90.

[12] Xavier Boyen. "The uber-assumption family". In: *International Conference on Pairing-Based Cryptography*. Springer. 2008, pp. 39–56.

[13] Zvika Brakerski and Guy N. Rothblum. "Virtual Black-Box Obfuscation for All Circuits via Generic Graded Encoding". In: *Proc. TCC 2014*. Vol. 8349. Lecture Notes in Computer Science. Springer, 2014, pp. 1–25.

[14] Daniel R. L. Brown. "Generic Groups, Collision Resistance, and ECDSA". In: *Des. Codes Cryptography* 35.1 (2005), pp. 119–152.

[15] Daniel R. L. Brown. *The Encrypted Elliptic Curve Hash*. IACR ePrint Archive, report 2008/12. `http://eprint.iacr.org/2008/12`. 2008.

[16] Ran Canetti. “Towards realizing random oracles: Hash functions that hide all partial information”. In: *Annual International Cryptology Conference*. Springer. 1997, pp. 455–469.

[17] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. “Obfuscation of probabilistic circuits and applications”. In: *Theory of Cryptography*. Springer, 2015, pp. 468–497.

[18] David G Cantor and Hans Zassenhaus. “A new algorithm for factoring polynomials over finite fields”. In: *Mathematics of Computation* (1981), pp. 587–592.

[19] Nicholas Cavanna. “Polynomial Factoring Algorithms and their Computational Complexity”. In: (2014).

[20] Melissa Chase, Mary Maller, and Sarah Meiklejohn. “Déjà Q All Over Again: Tighter and Broader Reductions of q-Type Assumptions”. In: *Proc. ASIACRYPT (2) 2016*. Vol. 10032. Lecture Notes in Computer Science. 2016, pp. 655–681.

[21] Melissa Chase and Sarah Meiklejohn. “Déjà Q: Using Dual Systems to Revisit q-Type Assumptions”. In: *Proc. EUROCRYPT 2014*. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 622–639.

[22] Jung Hee Cheon. “Security Analysis of the Strong Diffie-Hellman Problem”. In: *Proc. EUROCRYPT 2006*. Vol. 4004. Lecture Notes in Computer Science. Springer, 2006, pp. 1–11.

[23] Ronald Cramer and Victor Shoup. “Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption”. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2002, pp. 45–64.

[24] Ivan Damgård. “Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks”. In: *Proc. CRYPTO 1991*. Vol. 576. Lecture Notes in Computer Science. Springer, 1992, pp. 445–456.

[25] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. “Square Span Programs with Applications to Succinct NIZK Arguments”. In: *Proc. ASIACRYPT (1) 2014*. Vol. 8873. Lecture Notes in Computer Science. Springer, 2014, pp. 532–550.

[26] Whitfield Diffie and Martin E. Hellman. “New directions in cryptography”. In: *IEEE Trans. Information Theory* 22.6 (1976), pp. 644–654.

[27] Sanjam Garg, Craig Gentry, and Shai Halevi. “Candidate Multilinear Maps from Ideal Lattices”. In: *Proc. EUROCRYPT 2013*. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 1–17.

[28] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. “Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits”. In: *Proc. FOCS 2013*. IEEE Computer Society, 2013, pp. 40–49.

[29] Craig Gentry. “A fully homomorphic encryption scheme”. PhD thesis. Stanford University, 2009.

[30] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2001.

[31] Jens Groth and Amit Sahai. “Efficient non-interactive proof systems for bilinear groups”. In: *Advances in Cryptology–EUROCRYPT 2008*. Springer, 2008, pp. 415–432.

[32] Satoshi Hada and Toshiaki Tanaka. “On the Existence of 3-Round Zero-Knowledge Protocols”. In: *Proc. CRYPTO 1998*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 408–423.

[33] Neal Koblitz and Alfred Menezes. “The Brave New World of Bodacious Assumptions in Cryptography”. In: *Notices of the AMS* 57 (2010), pp. 357–365.

[34] Helger Lipmaa. “On the CCA1-Security of Elgamal and Damgård’s Elgamal”. In: *Proc. Inscrypt 2010*. Ed. by Xuejia Lai, Moti Yung, and Dongdai Lin. Vol. 6584. Lecture Notes in Computer Science. Springer, 2010, pp. 18–35.

[35] Sarah Meiklejohn. “An extension of the Groth-Sahai proof system”. PhD thesis. Brown University, 2009.

[36] Moni Naor and Moti Yung. “Public-key cryptosystems provably secure against chosen ciphertext attacks”. In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. ACM. 1990, pp. 427–437.

[37] Rafael Pass, Karn Seth, and Sidharth Telang. “Indistinguishability obfuscation from semantically-secure multilinear encodings”. In: *International Cryptology Conference*. Springer. 2014, pp. 500–517.

[38] Dominique Unruh. “Relations amount Statistical Security Notions-or-Why Exponential Adversaries are Unlimited.” In: *IACR Cryptology ePrint Archive* (2005). `https://eprint.iacr.org/2005/406`, p. 406.

[39] Hoeteck Wee. “On obfuscating point functions”. In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. ACM. 2005, pp. 523–532.

[40] Joe Zimmerman. “How to Obfuscate Programs Directly”. In: *Proc. EUROCRYPT (2) 2015*. Vol. 9057. Lecture Notes in Computer Science. Springer, 2015, pp. 439–467.

Supplementary Material

A Preparations for the main theorem

In this chapter we prove that if the building blocks we use to construct the group scheme $\Gamma_{m,n}$ satisfy their respective security requirements, the (m,n)-Interactive Uber assumption holds for the group scheme $\Gamma_{m,n}$ constructed in Section 3.3. Preliminary, we prove three lemmas that facilitate and modularize the proof of this statement.

A.1 Interchangeable ways to sample the public parameters

During the proofs in this chapter we manipulate the way public parameters for the group scheme are sampled. For greater clarity, we refer to the following distributions of public parameters as follows:

$$\widetilde{pp} : \text{sampled as } pp \text{ but with } y \in \mathrm{TD} \tag{10}$$

$$\widehat{pp} : \text{sampled as } \widetilde{pp} \text{ but with } \Lambda_{\mathrm{add}} \leftarrow pi\mathcal{O}(1^{poly(\lambda)}, C''_{\mathsf{Add}}), \text{ and } crs \text{ in hiding mode} \tag{11}$$

See Fig. 8 for the implementation of the circuit C''_{Add}.

Circuit $C''_{\mathsf{Add}}[gpk, (\mathrm{TD}, y), w_y](a, b)$

if $\neg\textsc{Val}(a) \vee \neg\textsc{Val}(b)$ **then return** $\perp$

parse $a =: (C^{(a)}, C'^{(a)}, \pi^{(a)})$, parse $b =: (C^{(b)}, C'^{(b)}, \pi^{(b)})$

$C^{(c)} := \textsc{Eval}(pk, \oplus, C^{(a)}, C^{(b)})$, $C'^{(c)} := \textsc{Eval}(pk', \oplus, C'^{(a)}, C'^{(b)})$

$\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}), (w_y))$

return $c := (C^{(c)}, C'^{(c)}, \pi^{(c)})$

Fig. 8. The circuit C''_{Add} for addition of two group elements that always produces the consistency proof $\pi^{(c)}$ for the relation $\mathcal{R}_3$ and does not require any secret information except for the witness w_y. The circuit $\oplus$ realizes addition in $\mathbb{Z}_q^d$.

These two distributions are computationally indistinguishable even if an adversary knows the corresponding secret decryption keys sk and sk', the point $\boldsymbol{\omega}$, and the opening op. A very similar statement was stated in Lemma 1 in the proceedings version of [3]. In contrast to that statement we need this indistinguishability to hold even if the adversary knows the values $\boldsymbol{\omega}$ that are hidden inside the point obfuscations $\mathsf{po} = (\mathsf{po}_1, \ldots, \mathsf{po}_m)$ and the opening op for the commitment com.

Lemma 1 (Swap lemma, [3]). *Let $pi\mathcal{O}$ be a probabilistic indistinguishability obfuscator with respect to $\mathcal{S}^{X\text{-}ind}$ for a circuit family that only contains circuits with input length upper bounded by $poly(\lambda)$, and let Π be a dual mode NIWI proof system. Then, for any PPT distinguisher $\mathcal{A}$, the advantage*

$$\begin{aligned} Adv_{\mathcal{A}}^{swap}(\lambda) := & \Pr\big[\mathcal{A}(1^{\lambda}, pp, sk, sk', \boldsymbol{\omega}, op) = 1 \;\big|\; pp \leftarrow \widetilde{pp}\big] \\ & - \Pr\big[\mathcal{A}(1^{\lambda}, pp, sk, sk', \boldsymbol{\omega}, op) = 1 \;\big|\; pp \leftarrow \widehat{pp}\big] \end{aligned} \tag{12}$$

is negligible in λ.

Table 2. An overview on the proof steps of Lemma 1, [3]. The $\boxed{\text{boxes}}$ emphasize changes compared to the previous game.

	C_{Add} **knows**	**CRS**	**Remark**
Game_0	$sk,\ sk',\ \boldsymbol{\omega},\ op,\ td_{\text{ext}}$	binding	
Game_1	$\boxed{sk,\ sk',\ \boldsymbol{\omega},\ op,\ w_y}$	binding	security of $pi\mathcal{O}$
Game_2	$sk,\ sk',\ \boldsymbol{\omega},\ op,\ w_y$	$\boxed{\text{hiding}}$	CRS indistinguishability of Π
Game_3	$\boxed{w_y}$	hiding	security of $pi\mathcal{O}$

Proof. To prove this statement, we proceed over a series of games using similar arguments as in the proof of Lemma 1 in the proceedings version of [3]. Let out_i denote the output of Game_i. For an overview on the proof steps we refer the reader to Table 2.

Game_0. This game samples public parameters pp as $\widetilde{pp}$ (see Eq. (10)), calls the adversary $\mathcal{A}$ on input $(1^{\lambda},\ pp,\ sk,\ sk',\ \boldsymbol{\omega},\ op)$, and outputs $\mathcal{A}$'s output.

Game_1. Is the same as Game_0 with the difference that Game_1 produces the obfuscation Λ_{add} via $pi\mathcal{O}(1^{poly(\lambda)},\ C'_{\mathsf{Add}})$ for the circuit C'_{Add} (see Fig. 9 for the implementation of C'_{Add}). Due to the perfect extractability of Π and the fact that w_y is the unique witness for the statement $y \in \text{TD}$, the two circuits C_{Add} and C'_{Add} are functionally equivalent. Furthermore, $poly(\lambda)$ is an upper bound for the input length of the two circuits. Hence, this game hop is justified by the security property of $pi\mathcal{O}$. In particular, there exists a circuit sampler $S_1 \in \mathcal{S}^{X\text{-ind}}$ and a PPT adversary $\mathcal{B}_1$, such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq 2 \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O},S_1,\mathcal{B}_1}(poly(\lambda)) \right|$. As the extraction trapdoor td_{ext} is no longer necessary in Game_1, we are able to switch over to use a hiding CRS without any further changes to the game.

Game_2. Is the same as Game_1 except for the fact that Game_2 produces the CRS crs in hiding mode via $(crs,\ \cdot) \leftarrow \mathsf{S}(gpk,\ gsk)$. This game hop is justified by the

Circuit $C'_{\mathsf{Add}}[gpk, rp, sk, sk', \boldsymbol{\omega}, op, w_y](a, b)$

if $\neg\textsc{Val}(a) \vee \neg\textsc{Val}(b)$ **then return** $\perp$
parse $a =: (C^{(a)}, C'^{(a)}, \pi^{(a)})$
parse $b =: (C^{(b)}, C'^{(b)}, \pi^{(b)})$
$C^{(c)} := \textsc{Eval}(pk, \oplus, C^{(a)}, C^{(b)})$
$C'^{(c)} := \textsc{Eval}(pk', \oplus, C'^{(a)}, C'^{(b)})$
$\vec{f}^{(a)} := \textsc{Dec}(sk, C^{(a)}), \vec{f'}^{(a)} := \textsc{Dec}(sk', C'^{(a)})$
$\vec{f}^{(b)} := \textsc{Dec}(sk, C^{(b)}), \vec{f'}^{(b)} := \textsc{Dec}(sk', C'^{(b)})$
$\vec{f}^{(c)} := \oplus(\vec{f}^{(a)}, \vec{f}^{(b)}), \vec{f'}^{(c)} := \oplus(\vec{f'}^{(a)}, \vec{f'}^{(b)})$
if $(C^{(a)}, C'^{(a)}), (C^{(b)}, C'^{(b)}) \in L_1$ **then**
  $\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}),$
    $(sk, sk', \vec{f}^{(c)}, \vec{f'}^{(c)}))$
elseif $(C^{(a)}, C'^{(a)}), (C^{(b)}, C'^{(b)}) \in L_2$ **then**
  $\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}),$
    $(sk, sk', \vec{f}^{(c)}, \vec{f'}^{(c)}, w, op))$
else
  $\cancel{w_y \leftarrow \mathsf{Extract}(td_{\mathrm{ext}}, (C^{(\alpha)}, C'^{(\alpha)}), \pi^{(\alpha)})}$
  $\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}), (w_y))$
return $c := (C^{(c)}, C'^{(c)}, \pi^{(c)})$

Circuit $C''_{\mathsf{Add}}[gpk, (\mathrm{TD}, y), w_y](a, b)$

if $\neg\textsc{Val}(a) \vee \neg\textsc{Val}(b)$ **then return** $\perp$
parse $a =: (C^{(a)}, C'^{(a)}, \pi^{(a)})$
parse $b =: (C^{(b)}, C'^{(b)}, \pi^{(b)})$
$C^{(c)} := \textsc{Eval}(pk, \oplus, C^{(a)}, C^{(b)})$
$C'^{(c)} := \textsc{Eval}(pk', \oplus, C'^{(a)}, C'^{(b)})$
$\pi^{(c)} \leftarrow \mathsf{Prove}(gpk, crs, (C^{(c)}, C'^{(c)}),$
  $(w_y))$
return $c := (C^{(c)}, C'^{(c)}, \pi^{(c)})$

Fig. 9. The circuit C'_{Add} (left) for addition of two group elements that does not use the extraction trapdoor td_{ext}. The circuit C''_{Add} (right) for addition of two group elements that always produces the consistency proof $\pi^{(c)}$ for the relation $\mathcal{R}_3$ and, hence, does not need any secret information except for w_y. In contrast to C_{Add}, the code line in C'_{Add} that is ~~highlighted~~ is not necessary anymore as C'_{Add} has the witness w_y hard-coded. The differing sections of the implementations of C'_{Add} and C''_{Add} are highlighted. The circuit $\oplus$ realizes addition in $\mathbb{Z}_q^d$.

CRS indistinguishability of Π. In other words, there exists a PPT adversary $\mathcal{B}_2$, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq 2 \cdot \left|Adv^{\mathrm{crs}}_{\Pi,\mathcal{B}_2}(\lambda)\right|$.
Game_3. Is identical to Game_2 except for the fact that Game_3 produces the obfuscation Λ_{add} via $pi\mathcal{O}(1^{poly(\lambda)}, C''_{\mathsf{Add}})$ for the circuit C''_{Add} (see Fig. 9 for the implementation of C''_{Add}). However, the two circuits C'_{Add} and C''_{Add} are not functionally equivalent.

Claim. For any PPT distinguisher $\mathcal{A}$, there exists a circuit sampler $S_3 \in \mathcal{S}^{X\text{-ind}}$ and a PPT adversary $\mathcal{B}_3$, such that $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq 2 \cdot \left|Adv^{\text{pio-ind}}_{pi\mathcal{O},S_3,\mathcal{B}_3}(poly(\lambda))\right|$.

Proof (sketch). The difference between the circuits C'_{Add} and C''_{Add} is limited to the fact that C''_{Add} always produces the consistency proof $\pi^{(c)}$ for the relation $\mathcal{R}_3$ using w_y as a witness, even if the representations of its inputs a and b are consistent. However, due to the perfect witness-indistinguishability of Π under a hiding CRS, the consistency proofs $\pi^{(c)}$ produced by C'_{Add} and C''_{Add} are identically distributed. We remark that an adversary $\mathcal{B}_3$ in this reduction is invoked using $poly(\lambda)$ as security parameter. Hence, $poly(\lambda)$ is an upper bound for the input length of the circuits C'_{Add} and C''_{Add}. Furthermore, the cardinality of the domain of the circuits C'_{Add} and C''_{Add} is less or equal to $2^{poly(\lambda)}$. Let $X \colon \mathbb{N} \to \mathbb{N}$ be a function such that $X(l) = 2^l$ for any $l \in \mathbb{N}$. Therefore, $X(poly(\lambda))$ is greater or equal to the cardinality of the domain of the circuits C'_{Add} and C''_{Add}.

We construct a circuit sampler S_3 that samples public parameters as in Game_2 omitting the obfuscated circuit Λ_{add} and outputs the implementations of the two circuits C'_{Add} and C''_{Add}. To prove that $S_3 \in \mathcal{S}^{X\text{-ind}}$, we define the set $\mathcal{X}$ to span the entire domain of the circuits C'_{Add} and C''_{Add}. Thus, for any possibly unbounded adversary $\mathcal{D}$, the advantage $Adv^{\text{eq\$}}_{S_3,\mathcal{D}}(\lambda) = 0$. Furthermore, for any non-uniform PPT distinguisher $\mathcal{D}'$, the advantage $Adv^{\text{sel-ind}}_{S_3,\mathcal{D}'}(\lambda) = 0$ as for any input, the resulting output of the two circuits is identically distributed. Therefore, S_3 is an X-Ind sampler.

To complete the proof, we construct $\mathcal{B}_3$ such that it simulates Game_2 if $Exp^{\text{pio-ind}}_{\mathcal{B}_3,pi\mathcal{O}}$ provides an obfuscation of C'_{Add}, and Game_3 otherwise. □

Hence, for any PPT distinguisher $\mathcal{A}$, there exists an X-Ind sampler S and PPT adversaries $\mathcal{D}_1$ and $\mathcal{D}_2$ such that

$$|Adv^{\text{swap}}_{\mathcal{A}}(\lambda)| \leq 4 \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O},S,\mathcal{D}_1}(poly(\lambda)) \right| + 2 \cdot \left| Adv^{\text{crs}}_{\Pi,\mathcal{D}_2}(\lambda) \right|. \tag{13}$$

□

Later in the proof we will consider a zero testing circuit that uses sk' to decrypt the second part C' of encodings instead of the first part C using sk. We refer to this circuit as $\overline{C}_{\mathsf{Zero}}$ and refer the reader to Fig. 11 for more details. An adaption of Lemma 1 such that both $\widetilde{pp}$ and $\widehat{pp}$ contain an obfuscation of the circuit $\overline{C}_{\mathsf{Zero}}$ instead of an obfuscation of the circuit $C^{(0)}_{\mathsf{Zero}}$ will turn out to be useful. We refer to these distributions as $\widetilde{pp}'$ and $\widehat{pp}'$ respectively. Using a similar argument as before, we can see that Swap lemma also holds for these two distributions. We refer to the corresponding advantage of a PPT distinguisher $\mathcal{A}$ as $Adv^{\text{swap}'}_{\mathcal{A}}(\lambda)$.

A.2 Switching of encodings

In this section, we observe that encodings of the same group element are computationally indistinguishable. The notion of indistinguishability is formalized via the Switch game defined in Fig. 10.

An adversary $\mathcal{A}$ for the Switch game $Exp^{\text{switch}}_{\mathcal{A}}$ is *legitimate*, if and only if it always guarantees that the representations $(\overrightarrow{f}^{(0)}, \overrightarrow{f'}^{(0)})$ and $(\overrightarrow{f}^{(1)}, \overrightarrow{f'}^{(1)})$ in $Exp^{\text{switch}}_{\mathcal{A}}$ are consistent and represent the same group element. In other words, any legitimate adversary for the Switch game always chooses $(\overrightarrow{f}^{(0)}, \overrightarrow{f'}^{(0)})$ and $(\overrightarrow{f}^{(1)}, \overrightarrow{f'}^{(1)})$ such that $f^{(0)}(\boldsymbol{\omega}) = f'^{(0)}(\boldsymbol{\omega}) = f^{(1)}(\boldsymbol{\omega}) = f'^{(1)}(\boldsymbol{\omega})$, where $f^{(b)} := \varphi_{\text{pol}}(A^{-1} \cdot \overrightarrow{f}^{(b)})$ and $f'^{(b)} := \varphi_{\text{pol}}(A^{-1} \cdot \overrightarrow{f'}^{(b)})$ for $b \in \{0, 1\}$.

$$
\begin{array}{l}
\textsc{Experiment } Exp_{\mathcal{A}}^{\text{switch}}(\lambda) \\
\hline
pp \leftarrow \textsc{Setup}(1^\lambda) \\
((\vec{f}^{(0)}, \vec{f'}^{(0)}), (\vec{f}^{(1)}, \vec{f'}^{(1)}), st) \leftarrow \mathcal{A}(1^\lambda, pp, \boldsymbol{\omega}, op, \mathsf{find}) \\
b \leftarrow \{0,1\},\ C \leftarrow \textsc{Enc}(pk, \vec{f}^{(b)}; R),\ C' \leftarrow \textsc{Enc}(pk', \vec{f'}^{(b)}; R') \\
\textbf{if } (C, C') \in L_1 \textbf{ then} \\
\quad \pi \leftarrow \mathsf{Prove}(gpk, crs, (C, C'), ((\vec{f}^{(b)}, R), (\vec{f'}^{(b)}, R'))) \\
\textbf{else} \\
\quad \pi \leftarrow \mathsf{Prove}(gpk, crs, (C, C'), ((\vec{f}^{(b)}, R), (\vec{f'}^{(b)}, R'), \boldsymbol{\omega}, op)) \\
b' \leftarrow \mathcal{A}(1^\lambda, (C, C', \pi), st, \mathsf{attack}) \\
\textbf{if } b = b' \textbf{ then return } 1 \\
\textbf{return } 0
\end{array}
$$

Fig. 10. The description of the Switch game $Exp_{\mathcal{A}}^{\text{switch}}(\lambda)$.

A similar statement of indistinguishability was stated in Theorem 1 in the proceedings version of [3]. In contrast to the game formalized in [3], the Switch game needs to explicitly decide whether to produce the consistency proof π of the challenge encoding (C, C', π) for the relation $\mathcal{R}_1$ or for the relation $\mathcal{R}_2$.

We remark that in the Switch game, the consistency proof is produced using the encryption randomness as part of the witness, whereas in the addition circuit the secret decryption keys sk and sk' are used as part of the witness.

Lemma 2 (Switching lemma, [3]). *Let $\Gamma_{m,n}$ be the group scheme constructed in Section 3.3. Further, let pi$\mathcal{O}$ be a probabilistic indistinguishability obfuscator with respect to $\mathcal{S}^{X\text{-}ind}$ for a circuit family that only contains circuits with input length upper bounded by poly(λ), let $\mathcal{TD} = (\mathcal{TD}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of families $\mathcal{TD}_\lambda = \{TD\}$ of languages $TD \subseteq \mathcal{X}_\lambda$ such that the subset membership problem is hard, let Π be a dual mode NIWI proof system, and let* Hpke *be an IND-CPA secure HPKE scheme. Then, for any legitimate PPT adversary $\mathcal{A}$, the advantage*

$$
Adv_{\mathcal{A}}^{switch}(\lambda) := \Pr\left[Exp_{\mathcal{A}}^{switch}(\lambda) = 1\right] - \frac{1}{2} \tag{14}
$$

is negligible in λ.

An important observation to adapt the proof strategy of [3] is that consistency proofs that are produced for either $\mathcal{R}_1$ or $\mathcal{R}_2$ depending on whether the chosen representation is constant or not, and consistency proofs that are produced for $\mathcal{R}_3$, are identically distributed under a hiding CRS. Therefore, the proof strategy is similar as in the proof of Theorem 1 in the proceedings version of [3].

Proof. To prove this statement, we proceed over a series of games. We start with the original Switch game and stop in a game that is independent of the bit b. Let out_i denote the output of Game_i. Further, let $\mathcal{A}$ be a legitimate PPT adversary

Table 3. An overview on the steps of the proof of Lemma 2, [3]. The $\boxed{\text{boxes}}$ emphasize changes compared to the previous game. Let pp' denote public parameters that are sampled like in SETUP(1^λ) but contain an obfuscation of the circuit $\overline{C}_{\mathsf{Zero}}$ instead of an obfuscation of the circuit C_{Zero}. Further, let W_1 denote the witness that is used to prove relation $\mathcal{R}_1$, i.e. $W_1 := ((\vec{f}, R), (\vec{f'}, R'))$, let W_2 denote the witness that is used to prove relation $\mathcal{R}_2$, i.e. $W_2 := ((\vec{f}, R), (\vec{f'}, R'), \boldsymbol{\omega}, op)$, and let W_3 denote the witness (w_y).

	Public parameters pp	C **encrypts**	C' **encrypts**	**Witness for consistency proof**	**Remark**
Game_0	pp	$\vec{f}^{(b)}$	$\vec{f'}^{(b)}$	W_1 resp. W_2	Switch game
Game_1	$\boxed{\widetilde{pp}}$	$\vec{f}^{(b)}$	$\vec{f'}^{(b)}$	W_1 resp. W_2	SMP TD $\subseteq \mathcal{X}$
Game_2	$\boxed{\widehat{pp}}$	$\vec{f}^{(b)}$	$\vec{f'}^{(b)}$	W_1 resp. W_2	Swap lemma (Lemma 1)
Game_3	$\widehat{pp}$	$\vec{f}^{(b)}$	$\vec{f'}^{(b)}$	$\boxed{W_3}$	perfect WI of Π
Game_4	$\widehat{pp}$	$\vec{f}^{(b)}$	$\boxed{\vec{f'}^{(1)}}$	W_3	IND-CPA security of HPKE
Game_5	$\widehat{pp}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	$\boxed{W_1 \text{ resp. } W_2}$	perfect WI of Π
Game_6	$\boxed{\widetilde{pp}}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	W_1 resp. W_2	Swap lemma (Lemma 1)
Game_7	$\boxed{pp}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	W_1 resp. W_2	SMP TD $\subseteq \mathcal{X}$
Game_8	$\boxed{pp'}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	W_1 resp. W_2	security of $pi\mathcal{O}$
Game_9	$\boxed{\widetilde{pp}'}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	W_1 resp. W_2	SMP TD $\subseteq \mathcal{X}$
Game_{10}	$\boxed{\widehat{pp}'}$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	W_1 resp. W_2	Swap lemma for $\widetilde{pp}'$, $\widehat{pp}'$
Game_{11}	$\widehat{pp}'$	$\vec{f}^{(b)}$	$\vec{f'}^{(1)}$	$\boxed{W_3}$	perfect WI of Π
Game_{12}	$\widehat{pp}'$	$\boxed{\vec{f}^{(1)}}$	$\vec{f'}^{(1)}$	W_3	IND-CPA security of HPKE independent of b

for the Switch game. For an overview on the proof steps we refer the reader to Table 3.

Game_0. Is the original Switch game $Exp_{\mathcal{A}}^{\text{switch}}(\lambda)$.

Game_1. Is the same as Game_0 with the difference that the public parameters are sampled with a YES-instance $y \leftarrow$ TD of the subset membership problem TD $\subseteq \mathcal{X}$. In other words, in Game_1 pp is distributed as $\widetilde{pp}$. This game hop is justified by the hardness of the subset membership problem TD $\subseteq \mathcal{X}$. Particularly, there exists a PPT adversary $\mathcal{B}_1$, such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq \left| Adv_{\mathcal{TD},\mathcal{B}_1}^{\text{smp}}(\lambda) \right|$. This enables to produce valid but inconsistent encodings of group elements.

Game_2. Is the same as Game_1 except for the fact that the public parameters pp are sampled like $\widehat{pp}$ instead of being sampled like $\widetilde{pp}$. This hop is justified by the Swap lemma (Lemma 1). In other words, $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq$

$\left|Adv^{\mathrm{swap}}_{\mathcal{B}_2}(\lambda)\right|$ for a suitable PPT distinguisher $\mathcal{B}_2$. We emphasize that the secret decryption key sk' is never used in this game.

Game_3. Is identical to Game_2 except for the generation of the consistency proof. In this game the consistency proof π for the challenge encoding is produced for relation $\mathcal{R}_3$ instead of being produced for either relation $\mathcal{R}_1$ or relation $\mathcal{R}_2$. The corresponding witness is w_y. As the CRS *crs* is in hiding mode and Π is perfectly witness-indistinguishable under a hiding CRS, $\Pr[out_3 = 1] - \Pr[out_2 = 1] = 0$.

Game_4. Is the same as Game_3 with the difference that the ciphertext C' for the challenge encoding is produced as $C' \leftarrow \textsc{Enc}(pk', \overrightarrow{f'}^{(1)}; R')$. In other words, the ciphertext C' in this game always encrypts $\overrightarrow{f'}^{(1)}$ instead of $\overrightarrow{f'}^{(b)}$. This hop is justified by the IND-CPA security of $\textsc{Hpke}$.

Claim. For any legitimate PPT adversary $\mathcal{A}$, there exists a legitimate PPT adversary $\mathcal{B}_4$ for the IND-CPA security of the HPKE scheme $\textsc{Hpke}$, such that $|\Pr[out_4 = 1] - \Pr[out_3 = 1]| \leq 2 \cdot \left|Adv^{\text{ind-cpa}}_{\textsc{Hpke},\mathcal{B}_4}(\lambda)\right|$.

Proof (sketch). We construct a legitimate PPT adversary $\mathcal{B}_4$ for the IND-CPA game with $\textsc{Hpke}$ that samples public parameters as in Game_3 embedding its input pk' and simulates Game_3 for $\mathcal{A}$. Given the output of $\mathcal{A}$'s find-phase, $\mathcal{B}_4$ outputs the tuple $(m_0, m_1) := (\overrightarrow{f'}^{(b)}, \overrightarrow{f'}^{(1)})$ to the IND-CPA game and uses the resulting ciphertext as C' to produce the encoding (C, C', π). As the consistency proof π is produced for relation $\mathcal{R}_3$, $\mathcal{B}_4$ does not need to know the encrypted vector or the used encryption randomness R' to produce π. $\square$

Game_5. Is the same as Game_4, but in this game the consistency proof π of the challenge is again produced for relation $\mathcal{R}_1$ or relation $\mathcal{R}_2$ depending on whether the representation $(\vec{f}^{(b)}, \overrightarrow{f'}^{(1)})$ is constant or non-constant. As $\mathcal{A}$ is legitimate, both representations $(\vec{f}^{(0)}, \overrightarrow{f'}^{(1)})$ and $(\vec{f}^{(1)}, \overrightarrow{f'}^{(1)})$ are consistent and represent the same group element, i.e. $\varphi_{\mathrm{pol}}(A^{-1} \cdot \vec{f}^{(b)})(\boldsymbol{\omega}) = \varphi_{\mathrm{pol}}(A^{-1} \cdot \overrightarrow{f'}^{(1)})(\boldsymbol{\omega})$. As the CRS *crs* is in hiding mode and Π is perfectly witness-indistinguishable under a hiding CRS, $\Pr[out_5 = 1] - \Pr[out_4 = 1] = 0$.

Game_6. Is the same as Game_5 except for the fact that the public parameters are again sampled as $\widetilde{pp}$, i.e. containing a CRS in binding mode and an obfuscation of the circuit C_{Add}. This game hop is justified by Lemma 1. The analysis is similar to the analysis of the game hop from Game_1 to Game_2. Particularly, there exists a PPT distinguisher $\mathcal{B}_6$, such that $|\Pr[out_6 = 1] - \Pr[out_5 = 1]| \leq \left|Adv^{\mathrm{swap}}_{\mathcal{B}_6}(\lambda)\right|$. We emphasize that w_y is not used in this game.

Game_7. Is identical to Game_6 with the difference that the public parameters are sampled with a NO-instance $y \leftarrow \mathcal{X} \setminus \mathrm{TD}$ of the subset membership problem $\mathrm{TD} \subseteq \mathcal{X}$. Hence, the public parameters in this game are again distributed as the output of $\textsc{Setup}(1^\lambda)$. This hop is justified by the hardness of the subset membership problem $\mathrm{TD} \subseteq \mathcal{X}$. In other words, there exists a PPT adversary $\mathcal{B}_7$, such that $|\Pr[out_7 = 1] - \Pr[out_6 = 1]| \leq \left|Adv^{\mathrm{smp}}_{\mathcal{TD},\mathcal{B}_7}(\lambda)\right|$.

Game_8. Is the same as Game_7 except for the fact that the public parameters contain an obfuscation of the circuit $\overline{C}_{\mathsf{Zero}}$ (see Fig. 11 for the implementation of the circuit $\overline{C}_{\mathsf{Zero}}$) instead of an obfuscation of the circuit $C^{(0)}_{\mathsf{Zero}}$. In other words, Λ_{zero} is produced via $pi\mathcal{O}(1^{poly(\lambda)}, \overline{C}_{\mathsf{Zero}})$. This hop is justified by the security of $pi\mathcal{O}$.

Claim. For any legitimate PPT adversary $\mathcal{A}$, there exists an X-Ind sampler S_8 and a PPT adversary $\mathcal{B}_8$, such that $|\Pr[out_8 = 1] - \Pr[out_7 = 1]| \leq 2 \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O},S_8,\mathcal{B}_8}(poly(\lambda)) \right|$.

Proof (sketch). The two circuits $C^{(0)}_{\mathsf{Zero}}$ and $\overline{C}_{\mathsf{Zero}}$ differ only in the fact that $C^{(0)}_{\mathsf{Zero}}$ tests whether the equality $f(\boldsymbol{\omega}) = \varphi_{\text{pol}}(A^{-1} \cdot \vec{f})(\boldsymbol{\omega}) = 0$ holds and $\overline{C}_{\mathsf{Zero}}$ tests whether the equality $f'(\boldsymbol{\omega}) = \varphi_{\text{pol}}(A^{-1} \cdot \vec{f'})(\boldsymbol{\omega}) = 0$ holds. We observe that the public parameters in Game_7 and Game_8 contain a NO-instance of the SMP TD $\subseteq \mathcal{X}$ and the CRS *crs* is in binding mode. As Π is perfectly binding under *crs* and $y \notin$ TD, the representation of any valid encoding necessarily is consistent. Therefore, the circuits $C^{(0)}_{\mathsf{Zero}}$ and $\overline{C}_{\mathsf{Zero}}$ are functionally equivalent and a circuit sampler S_8 that samples public parameters as $\textsc{Setup}(1^\lambda)$ and outputs the implementations of the circuits $C^{(0)}_{\mathsf{Zero}}$ and $\overline{C}_{\mathsf{Zero}}$ is in the sampler class $\mathcal{S}^{X\text{-ind}}$. Besides, $poly(\lambda)$ upper bounds the input length of the circuits $C^{(0)}_{\mathsf{Zero}}$ and $\overline{C}_{\mathsf{Zero}}$. □

$\textsc{Circuit}\ \overline{C}_{\mathsf{Zero}}[q, sk', \boldsymbol{\omega}, A](a)$

if $\neg\textsc{Val}(a)$ **then return** $\perp$
parse $a =: (C, C', \pi)$
$\vec{f'} \leftarrow \textsc{Dec}(sk', C'),\ f'(\boldsymbol{X}) := \varphi_{\text{pol}}(A^{-1} \cdot \vec{f'})$
if $f'(\boldsymbol{\omega}) = 0$ **then return** 1
return 0

Fig. 11. Circuit for testing whether a given encoding corresponds to the identity element. In contrast to $C^{(0)}_{\mathsf{Zero}}$, this circuit uses the decryption key sk' to obtain the coefficients $\vec{f'}$.

Game_9. Is identical to Game_8 with the difference that the public parameters are sampled with a YES-instance $y \leftarrow$ TD of the subset membership problem TD $\subseteq \mathcal{X}$, i.e. the public parameters are distributed as $\widetilde{pp}'$ defined in Supplementary Section A.1. This hop is justified by the hardness of the subset membership problem TD $\subseteq \mathcal{X}$. The reduction is similar to the reduction for hop from Game_0 to Game_1. Hence, there exists a PPT adversary $\mathcal{B}_9$, such that $|\Pr[out_9 = 1] - \Pr[out_8 = 1]| \leq \left| Adv^{\text{smp}}_{\mathcal{TD},\mathcal{B}_9}(\lambda) \right|$.

Game_{10}. Is the same as Game_9 except for the fact that the public parameters pp are sampled like $\widehat{pp}'$. This game hop is justified by the Swap lemma for the distributions $\widetilde{pp}'$ and $\widehat{pp}'$. The analysis is analogous to the analysis of the game hop from Game_1 to Game_2. There exists a PPT distinguisher $\mathcal{B}_{10}$ such that $|\Pr[out_{10} = 1] - \Pr[out_9 = 1]| \leq \left| Adv^{\text{swap}'}_{\mathcal{B}_{10}}(\lambda) \right|$. We remark that the secret decryption key sk is never used in this game.

Game_{11}. Is the same as Game_{10} with the difference that the consistency proof π for the challenge is produced for relation $\mathcal{R}_3$ using w_y as a witness. As the CRS crs is in hiding mode and Π is perfectly witness-indistinguishable under a hiding CRS, $\Pr[out_{11} = 1] - \Pr[out_{10} = 1] = 0$.

Game_{12}. Is identical to Game_{11} with the difference that C always encrypts $\vec{f}^{(1)}$ instead of $\vec{f}^{(b)}$, i.e. C is produced via $C \leftarrow \textsc{Enc}(pk, \vec{f}^{(1)}; R)$. This game hop is justified by the IND-CPA security of $\textsc{Hpke}$. The analysis is analogous to the analysis of the game hop from Game_3 to Game_4. Hence, there exists a legitimate PPT adversary $\mathcal{B}_{12}$ for the IND-CPA security of $\textsc{Hpke}$, such that $|\Pr[out_{12} = 1] - \Pr[out_{11} = 1]| \leq 2 \cdot \left| Adv^{\text{ind-cpa}}_{\textsc{Hpke},\mathcal{B}_{12}}(\lambda) \right|$. Due to the fact that Game_{12} does not depend on b, $\Pr[out_{12} = 1] = \frac{1}{2}$.

Therefore, for any legitimate PPT adversary $\mathcal{A}$, there exists an X-Ind sampler S and (legitimate) PPT adversaries $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$ such that

$$\begin{aligned} \left| Adv^{\text{switch}}_{\mathcal{A}}(\lambda) \right| \;\leq\; & 3 \cdot \left| Adv^{\text{smp}}_{\mathcal{D}_1}(\lambda) \right| + 14 \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O},S,\mathcal{D}_2}(poly(\lambda)) \right| \\ & + 6 \cdot \left| Adv^{\text{crs}}_{\Pi,\mathcal{D}_3}(\lambda) \right| + 4 \cdot \left| Adv^{\text{ind-cpa}}_{\textsc{Hpke},\mathcal{D}_4}(\lambda) \right|. \end{aligned} \tag{15}$$

□

A.3 Randomization

To prove security of our construction, we need a technical lemma that enables to randomize the challenge in the proof of our main theorem (in Supplementary Section B).

$\textsc{Experiment}\ Exp^{\text{rand}}_{(q,d),\mathcal{A}}(\lambda)$	$\textsc{Oracle}\ \mathcal{O}^{\text{rand}}(t)$
$A \leftarrow \{B \in \mathrm{GL}_d(\mathbb{Z}_q) \mid B \cdot e_1 = e_1\}$	**return** $A \cdot t$
$(t^*, st) \leftarrow \mathcal{A}^{\mathcal{O}^{\text{rand}}(\cdot)}(\mathsf{find})$	
$b \leftarrow \{0,1\}$, $v_0^* := A \cdot t^*$, $v_1^* \leftarrow \mathbb{Z}_q^d$	
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\text{rand}}(\cdot)}(v_b^*, st, \mathsf{attack})$	
if $b = b'$ **then return** 1	
return 0	

Fig. 12. The description of the Randomization game $Exp^{\text{rand}}_{(q,d),\mathcal{A}}(\lambda)$.

Lemma 3 (Randomization lemma). *Let $d \in \mathbb{N}$ be a natural number, and let q be a prime number. An adversary $\mathcal{A}$ is* legitimate *if and only if it always guarantees that $t^* \notin \langle e_1, t_1, \ldots, t_l \rangle$ in $Exp^{rand}_{(q,d),\mathcal{A}}(\lambda)$, where $t_1, \ldots, t_l \in \mathbb{Z}_q^d$ are*

$\mathcal{A}$'s oracle queries. Then, for any possibly unbounded legitimate adversary $\mathcal{A}$, the advantage

$$Adv^{rand}_{(q,d),\mathcal{A}}(\lambda) := \Pr\Big[Exp^{rand}_{(q,d),\mathcal{A}}(\lambda) = 1\Big] - \frac{1}{2} \tag{16}$$

is at most $\frac{d}{q}$.

The proof of Lemma 3 is mainly technical.

Proof. To prove this technical lemma, we first observe that any matrix A, that is chosen uniformly at random from $\mathbb{Z}_q^{d\times d}$, such that $A \cdot t_j = v_j$ for any $j \in \{1, \ldots, i\}$, is uniformly distributed over the set of all matrices satisfying these equations. In other words, a matrix A that is sampled uniformly at random from $\mathbb{Z}_q^{d\times d}$ until the condition described above holds is uniformly distributed over the set of all matrices that satisfy this condition.

Without loss of generality, we consider an adversary $\mathcal{A}$ that queries $d-1$ vectors that are linearly independent of the set of all previous queries, only uses the oracle $\mathcal{O}^{\text{rand}}$ in its find-phase, and always queries the vector e_1 at first. This is justified by the fact that given an arbitrary legitimate adversary $\widetilde{\mathcal{A}}$, we are able to construct such an adversary $\mathcal{A}$ that has the same probability of success in the Randomization game as $\widetilde{\mathcal{A}}$. Particularly, $\mathcal{A}$ initially queries the vector e_1, invokes the find-phase of $\widetilde{\mathcal{A}}$ simulating the oracle $\mathcal{O}^{\text{rand}}$ for $\widetilde{\mathcal{A}}$, and only uses its own oracle to answer queries that can not be computed as a linear combination of the previous oracle queries. Given $\widetilde{\mathcal{A}}$'s challenge t^*, $\mathcal{A}$ extends the set of its previous oracle queries together with the vector t^* to a basis of $\mathbb{Z}_q^d$ and queries the resulting vectors from its oracle. Thus, $\mathcal{A}$ is able to simulate the oracle $\mathcal{O}^{\text{rand}}$ for the attack-phase of $\widetilde{\mathcal{A}}$ without using its own oracle.

To prove this statement, we proceed over a series of games. Let out_i denote the output of Game_i.

$\mathsf{Game_0}$. Is the Randomization game as described in Fig. 12.

$\mathsf{Game_1}$. Is the same as Game_0 except for the fact that the matrix A is chosen uniformly at random from all $d \times d$-matrices over $\mathbb{Z}_q$ and not only from $\mathrm{GL}_d(\mathbb{Z}_q)$. As the fraction of non-invertible matrices in the set of all matrices in $\mathbb{Z}_q^{d\times d}$ is at most $\frac{d}{q}$, $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq \frac{d}{q}$.

$\mathsf{Game_2}$. Is the same as Game_1 with the difference that the internal state of the game, i.e. the matrix A, is freshly sampled after the find-phase of $\mathcal{A}$ conditioned by the output the game already made. We refer to the matrix that is freshly sampled as $\widetilde{A}$. The adversary $\mathcal{A}$ does not make oracle queries after its find-phase has terminated. We remark that Game_2 is not necessarily efficient anymore. Using a similar argument as [38], this hop is conceptional and leads to a statistical distance of 0, hence, $\Pr[out_2 = 1] - \Pr[out_1 = 1] = 0$.

Consider the point in time after $\mathcal{A}$ has output the challenge t^*. The view of $\mathcal{A}$ only depends on the answers $v_0, \ldots, v_{d-2}$ to its oracle queries $t_0 = e_1, \ldots, t_{d-2}$. We observe that $t_1, \ldots, t_{d-2}$ and $v_0, \ldots, v_{d-2}$ are random variables depending on the random variable A. However, these vectors do not depend on the freshly sampled matrix $\widetilde{A}$. The matrix $\widetilde{A}$ is uniformly distributed over the set

$M := \{A \in \mathbb{Z}_q^{d\times d} \,|\, A \cdot t_i = v_i$ for any $i \in \{0, \ldots, d-2\}\}$. As $\{t_0, \ldots, t_{d-2}, t^*\}$ is a basis of $\mathbb{Z}_q^d$, $\widetilde{A}$ is of the form

$$\widetilde{A} = B \cdot D \text{ with } B = \left(v_0 \middle| \, v_1 \middle| \ldots \middle| \, v_{d-2} \middle| \, v^* \right) \text{ and } D = \left(t_0 \middle| \, t_1 \middle| \ldots \middle| \, t_{d-2} \middle| \, t^* \right)^{-1}.$$

The matrix D is independent of $\widetilde{A}$ and multiplication with D is a bijection. Hence, the column v^* is uniformly distributed over $\mathbb{Z}_q^d$, and $\Pr[out_2 = 1] = \frac{1}{2}$.

□

B The Interactive Uber assumption

The proof starts with the game $Exp^{\text{uber}}_{\Gamma_{m,n},\mathcal{A}}$ described in Fig. 5. In this game, the adversary requests the evaluation of selected polynomials at a secret point $\boldsymbol{s}$ as encodings in the group scheme $\Gamma_{m,n}$. Originally, the encodings of those group elements simply encrypt those evaluated values directly. For that reason, the game needs to know the secret point $\boldsymbol{s}$. As a first step to bypass that, we represent the secret $\boldsymbol{s}$ as $\boldsymbol{c} \circ \boldsymbol{\omega} := (c_i \cdot \omega_i)_{i=1}^m$ for a random point $\boldsymbol{c}$ and exploit the fact that the discrete logarithm that corresponds to an encoding equals the evaluation of the thereby determined m-variate polynomial at $\boldsymbol{\omega}$. Consequently, we are able to employ the Switching lemma (Lemma 2) to use non-constant polynomials in $\boldsymbol{\omega}$ to produce encodings. As a next step, we need to remove any information about the matrix A from the public parameters. To achieve that, we gradually alter the group structure such that encodings are treated as equal if and only if they determine the same abstract polynomial. This paves the way for employing the Randomization lemma (Lemma 3) to randomize the challenge.

Theorem 1. *Let $\Gamma_{m,n}$ be the group scheme constructed in Section 3.3. Further, let $pi\mathcal{O}$ be a probabilistic indistinguishability obfuscator with respect to $\mathcal{S}^{X\text{-}ind}$ for a circuit family containing circuits with input length at most $poly(\lambda)$, let $\mathcal{TD} = (\mathcal{TD}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of families $\mathcal{TD}_\lambda = \{TD\}$ of languages $TD \subseteq \mathcal{X}_\lambda$ such that the subset membership problem is hard, let Π be a dual mode NIWI proof system, let* Hpke *be an IND-CPA secure HPKE scheme, let* Com *be a perfectly binding non-interactive commitment scheme, and let* PObf *be a point obfuscation. Then, the (m, n)-Interactive Uber assumption (cf. Definition 10) holds for $\Gamma_{m,n}$.*

Before we prove this theorem, we state a technical lemma that helps to argue that the Randomization lemma can be applied.

Lemma 4. *Let $l, n, m \in \mathbb{N}$ be natural numbers, let $\mathbb{K}$ be a field, and let $\{Q_1(\boldsymbol{X}), \ldots, Q_l(\boldsymbol{X})\}$ be a set of m-variate polynomials over $\mathbb{K}$ of total degree at most n. Then the set $\{Q_1(\boldsymbol{X}), \ldots, Q_l(\boldsymbol{X})\}$ is linearly independent over $\mathbb{K}$ if and only if for any $\boldsymbol{c} \in (\mathbb{K}^\times)^m$, the set $\{Q_1(\boldsymbol{c} \circ \boldsymbol{X}), \ldots, Q_l(\boldsymbol{c} \circ \boldsymbol{X})\}$ is linearly independent over $\mathbb{K}$, where $\boldsymbol{c} \circ \boldsymbol{X} = (c_i \cdot X_i)_{i=1}^m$ is the Hadamard product.*

Proof (of Theorem 1). Let $\mathcal{A}$ be a legitimate adversary for the (m, n)-Interactive Uber game. To prove this theorem, we proceed over a series of games. We start with the *real* (m, n)-Interactive Uber game and stop with the *ideal* (m, n)-Interactive Uber game. Let out_i denote the output of Game_i. For an overview on the proof steps we refer the reader to Table 1.

Game_0. Is the **real** interactive Uber game $Exp^{\text{uber}}_{\Gamma_{m,n},\mathcal{A}}(\lambda)$, i.e. the bit b is set to 0. We emphasize that Game_0 produces encodings of group elements in a uniform manner as the sampling algorithm SAM defined in Section 3.3.

Game_1. Is identical to Game_0 with the difference that the secret value $\boldsymbol{s}$ is sampled as $\boldsymbol{s} := \boldsymbol{c} \circ \boldsymbol{\omega}$, where $\boldsymbol{c}$ is sampled uniformly at random from $\left(\mathbb{Z}_q^\times\right)^m$. We recall that the algorithm SETUP samples the value $\boldsymbol{\omega}$ from $(\mathbb{Z}_q)^m$ and includes point obfuscations for these values in *pp*. In Game_0 $\boldsymbol{s}$ and $\boldsymbol{\omega}$ are distributed uniformly and independently over $(\mathbb{Z}_q)^m$. In Game_1 $\boldsymbol{s}$ and $\boldsymbol{\omega}$ are distributed exactly as in Game_0 given that $\boldsymbol{\omega}$ does not contain any zero entries. As the probability that at least one entry of $\boldsymbol{\omega}$ equals zero is at most $\frac{m}{q}$, we have that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq \frac{m}{q}$.

Game_2. Is the same as Game_1 with the difference that Game_2 uses non-constant representations to produce group element encodings. In particular, Game_2 uses the representation vectors $\vec{f} := \vec{f'} := A \cdot \varphi_{\text{pol}}^{-1}(P(\boldsymbol{c} \circ \boldsymbol{X}))$ to produce an encoding of the group element $[P(\boldsymbol{s})]$. We recall that these representation vectors describe the polynomial $P(\boldsymbol{c} \circ \boldsymbol{X})$ with respect to the basis $\{\varphi_{\text{pol}}(a_1), \ldots, \varphi_{\text{pol}}(a_d)\}$. The corresponding consistency proofs are produced for either relation $\mathcal{R}_1$ or for relation $\mathcal{R}_2$ depending on whether the representation $(\vec{f}, \vec{f'})$ is constant or not. This game hop is justified by the Switching lemma (Lemma 2). Particularly, there exists a legitimate PPT adversary $\mathcal{B}_2$ for the Switch game, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq 2 \cdot (l + 1) \cdot \left|Adv^{\text{switch}}_{\mathcal{B}_2}(\lambda)\right|$, where l denotes the number of $\mathcal{A}$'s oracle queries.

To realize this, we use a standard hybrid argument. The hybrid game $\mathsf{Game}_{1.i}$ for $i \in \{0, \ldots, l\}$ is identical to Game_1 with the difference that the first i oracle queries are answered as in Game_2. The hybrid game $\mathsf{Game}_{1.(l+1)}$ is identical to Game_2. The adversary $\mathcal{B}_2$ receives the secret information $\boldsymbol{\omega}$ and *op* as input and, hence, is able to simulate each hybrid. $\mathcal{B}_2$ guesses an index $j \in \{1, \ldots, l+1\}$. If $j \in \{1, \ldots, l\}$, let $P(\boldsymbol{X})$ denote the j-th query polynomial, and if $j = l + 1$, let $P(\boldsymbol{X})$ denote the challenge polynomial. The adversary $\mathcal{B}_2$ outputs the two representations $\vec{f}^{(0)} := \vec{f'}^{(0)} := (P(\boldsymbol{c} \circ \boldsymbol{\omega}), 0, \ldots, 0)^T$ and $\vec{f}^{(1)} := \vec{f'}^{(1)} := A \cdot \varphi_{\text{pol}}^{-1}(P(\boldsymbol{c} \circ \boldsymbol{X}))$ to the Switch game and uses the resulting answer as answer to the j-th oracle query if $j \in \{1, \ldots, l\}$ or as challenge encoding otherwise. We recall that the first column of the matrix A is e_1 which is why $A^{-1} \cdot \vec{f}^{(0)} = P(\boldsymbol{c} \circ \boldsymbol{\omega}) \cdot e_1$. As $\varphi_{\text{pol}}\left(A^{-1} \cdot \vec{f}^{(0)}\right)(\boldsymbol{\omega}) = P(\boldsymbol{c} \circ \boldsymbol{\omega}) = \varphi_{\text{pol}}\left(A^{-1} \cdot \vec{f}^{(1)}\right)(\boldsymbol{\omega})$, $\mathcal{B}_2$ is a legitimate adversary for the Switch game.

Game_3. Is identical to Game_2 with the difference that *pp* is distributed as $\widetilde{pp}$ (see Eq. (10)). This hop is justified by the hardness of the SMP TD $\subseteq \mathcal{X}$, i.e. there exists a PPT adversary $\mathcal{B}_3$, such that $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq \left|Adv^{\text{smp}}_{\mathcal{TD},\mathcal{B}_3}(\lambda)\right|$.

Game_4. Is the same as Game_3 with the difference that the public parameters *pp* are distributed as $\widehat{pp}$ (see Eq. (11)). This hop is justified by Lemma 1. In particular, there exists a PPT distinguisher $\mathcal{B}_4$, such that $|\Pr[out_4 = 1] - \Pr[out_3 = 1]| \leq \left|Adv^{\text{swap}}_{\mathcal{B}_4}(\lambda)\right|$. To realize this, it is important to observe that $\mathcal{B}_4$ is able to simulate Game_3 and Game_4 as the necessary secret information $\boldsymbol{\omega}$ and op are part of its input.

Game_5. Is identical to Game_4 except for the fact that this game produces any consistency proof for relation $\mathcal{R}_3$. As crs is in hiding mode and Π satisfies perfect witness-indistinguishability under a hiding CRS, $\Pr[out_5 = 1] - \Pr[out_4 = 1] = 0$. This step allows to produce consistency proofs even if the commitment com contains $\perp$ and the point obfuscations $\mathsf{po}_1, \ldots, \mathsf{po}_m$ are produced for $\perp$. We emphasize that in this game the opening op is not used anymore.

Game_6. Is the same as Game_5 except for the fact that the commitment com is produced via $\textsc{Commit}_{ck}(\perp)$. In other words, in Game_5 com is a commitment for the matrix A, and in Game_6 com is a commitment for $\perp$. This game hop is justified by the computational hiding property of $\textsc{Com}$, i.e. there exists a PPT adversary $\mathcal{B}_6$, such that $|\Pr[out_6 = 1] - \Pr[out_5 = 1]| \leq 2 \cdot \left|Adv^{\text{hiding}}_{\textsc{Com},\mathcal{B}_6}(\lambda)\right|$.

For notational convenience, let $\mathsf{po}^{(i)} := (\mathsf{po}_1, \ldots, \mathsf{po}_m)$ denote the following distribution

$$\begin{aligned}\mathsf{po}^{(i)} := \big(\, & \mathsf{po}_1 \leftarrow \textsc{PObf}(\perp), \ldots, \mathsf{po}_i \leftarrow \textsc{PObf}(\perp), \\ & \mathsf{po}_{i+1} \leftarrow \textsc{PObf}(\omega_{i+1}), \ldots, \mathsf{po}_m \leftarrow \textsc{PObf}(\omega_m)\,\big).\end{aligned} \tag{17}$$

Hence, the tuple of point obfuscations po as defined in Game_6 is distributed as $\mathsf{po}^{(0)}$. Further, for any multivariate polynomial $f(\boldsymbol{X})$ and any $i \in \{0, \ldots, m\}$ let

$$F_i^{(f)}(X_1, \ldots, X_i) := f(X_1, \ldots, X_i, \omega_{i+1}, \ldots, \omega_m). \tag{18}$$

Let $\overline{pp}^{(i)}$ denote the distribution of public parameters as in Game_6 containing a tuple of point obfuscations distributed as $\mathsf{po}^{(i)}$ and an obfuscation Λ_{zero} for the circuit $C^{(i)}_{\mathsf{Zero}}$.

Game_7. Is the same as Game_6 with the difference that the m-tuple po of point obfuscations is distributed as $\mathsf{po}^{(m)}$ (see Eq. (17)) and the obfuscation of the zero testing circuit Λ_{zero} is produced via $pi\mathcal{O}(1^{poly(\lambda)}, C^{(m)}_{\mathsf{Zero}})$ (see Fig. 13 for the implementation of the circuit $C^{(i)}_{\mathsf{Zero}}$ for $i \in \{0, \ldots, m\}$). In other words, in Game_6 pp is distributed as $\overline{pp}^{(0)}$ and in Game_7 pp is distributed as $\overline{pp}^{(m)}$.

Lemma 5. *For any PPT adversary $\mathcal{A}$, there exists an X-Ind sampler S, and PPT adversaries $\mathcal{D}_1$ and $\mathcal{D}_2$, such that*

$$\begin{aligned}|\Pr[out_7 = 1] - \Pr[out_6 = 1]| \;\leq\; & 8m \cdot \left|Adv^{pio\text{-}ind}_{pi\mathcal{O},S,\mathcal{D}_1}(poly(\lambda))\right| \\ & + m \cdot \left|Adv^{po}_{\textsc{PObf},\mathcal{D}_2}(\lambda)\right|.\end{aligned}$$

Proof (of Lemma 5). We define a series of hybrid games. The hybrid game $\mathsf{Game}_{6.i}$ for $i \in \{0, \ldots, m\}$ is identical to Game_6 with the difference, that

Circuit $C^{(i)}_{\mathsf{Zero}}(a)$

if $\neg\textsc{Val}(a)$ **then return** $\bot$
parse $a =: (C, C', \pi)$
$\vec{f} \leftarrow \textsc{Dec}(sk, C)$
$f(\boldsymbol{X}) := \varphi_{\mathrm{pol}}(A^{-1} \cdot \vec{f})$
if $F_i^{(f)}(X_1, \ldots, X_i) \equiv 0$ **then**
  return 1
return 0

Circuit $C'^{(i)}_{\mathsf{Zero}}(a)$

if $\neg\textsc{Val}(a)$ **then return** $\bot$
parse $a =: (C, C', \pi)$
$\vec{f} \leftarrow \textsc{Dec}(sk, C)$
$f(\boldsymbol{X}) := \varphi_{\mathrm{pol}}(A^{-1} \cdot \vec{f})$
$\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)} \leftarrow (\mathbb{Z}_q)^i$
if $\forall j \in \{1, \ldots, \nu\}$:
  $\Big(F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, X_{i+1}) \equiv 0 \vee$
  $F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, \omega_{i+1}) = 0\Big)$ **then**
  return 1
return 0

Circuit $C''^{(i)}_{\mathsf{Zero}}(a)$

if $\neg\textsc{Val}(a)$ **then return** $\bot$
parse $a =: (C, C', \pi)$
$\vec{f} \leftarrow \textsc{Dec}(sk, C)$
$f(\boldsymbol{X}) := \varphi_{\mathrm{pol}}(A^{-1} \cdot \vec{f})$
$\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)} \leftarrow (\mathbb{Z}_q)^i$
$V_j :=$ zero set of $F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, X_{i+1})$
if $\forall j \in \{1, \ldots, \nu\}$:
  $\Big(F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, X_{i+1}) \equiv 0 \vee$
  $\exists v \in V_j : \mathsf{po}_{i+1}(v) = 1\Big)$ **then**
  return 1
return 0

Circuit $C'''^{(i)}_{\mathsf{Zero}}(a)$

if $\neg\textsc{Val}(a)$ **then return** $\bot$
parse $a =: (C, C', \pi)$
$\vec{f} \leftarrow \textsc{Dec}(sk, C)$
$f(\boldsymbol{X}) := \varphi_{\mathrm{pol}}(A^{-1} \cdot \vec{f})$
$\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)} \leftarrow (\mathbb{Z}_q)^{i+1}$
if $\forall j \in \{1, \ldots, \nu\}$:
  $\Big(F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, r_{i+1}^{(j)}) = 0\Big)$ **then**
  return 1
return 0

Fig. 13. Zero testing circuits that appear in the proof. $C^{(i)}_{\mathsf{Zero}}$, $C'^{(i)}_{\mathsf{Zero}}$ know q, sk, $(\omega_{i+1}, \ldots, \omega_m)$, and A. $C''^{(i)}_{\mathsf{Zero}}$, however, only needs to know po_{i+1} instead of ω_{i+1}. $C'''^{(i)}_{\mathsf{Zero}}$ does not need to know ω_{i+1} at all. Changes to the previous circuit are highlighted. We remark that testing whether an m-variate polynomial of total degree at most n is the zero polynomial can be done in polynomial time as $d = \binom{n+m}{m}$ is polynomial in λ.

the public parameters pp contain $\mathsf{po}^{(i)}$ and an obfuscation Λ_{zero} of the circuit $C^{(i)}_{\mathsf{Zero}}$. That is, the public parameters produced in $\mathsf{Game}_{6.i}$ are distributed as $\overline{pp}^{(i)}$. $\mathsf{Game}_{6.0}$ is identical to Game_6 and $\mathsf{Game}_{6.m}$ is identical to Game_7. For an overview on the proof steps we refer the reader to Table 4.

$\mathsf{Game}_{6.i.0}$. Is the same as $\mathsf{Game}_{6.i}$. That is, pp is distributed as $\overline{pp}^{(i)}$.

$\mathsf{Game}_{6.i.1}$. Is identical to $\mathsf{Game}_{6.i.0}$ with the difference that the obfuscation Λ_{zero} is produced for the circuit $C'^{(i)}_{\mathsf{Zero}}$ (see Fig. 13 for an implementation of the circuit $C'^{(i)}_{\mathsf{Zero}}$). This game hop is justified by the security property of $pi\mathcal{O}$ and the Schwartz-Zippel lemma.

Claim. For any legitimate PPT adversary $\mathcal{A}$, there exists a circuit sampler $S_1 \in \mathcal{S}^{X\text{-ind}}$ and a PPT adversary $\mathcal{B}_1$, such that $|\Pr[out_{6.i.1} = 1] - \Pr[out_{6.i.0} = 1]| \leq 2 \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O}, S_1, \mathcal{B}_1}(poly(\lambda)) \right|$.

Table 4. An overview on the steps of the proof of Lemma 5. The $\boxed{\text{boxes}}$ emphasize changes compared to the previous game. We recall that $F_i^{(f)}(X_1, \ldots, X_i) := f(X_1, \ldots, X_i, \omega_{i+1}, \ldots, \omega_m)$ and V_j is the zero set of the polynomial $F_{i+1}^{(f)}(\boldsymbol{r}^{(j)}, X_{i+1})$, where $\boldsymbol{r}^{(j)} \leftarrow \mathbb{Z}_q^i$.

	Point obfuscations	Zero circuit	Performed test	Remark
$\mathsf{Game}_{6.i.0}$	$\mathsf{po}^{(i)}$	$C_{\mathsf{Zero}}^{(i)}$	$F_i^{(f)}(X_1, \ldots, X_i) \equiv 0$	
$\mathsf{Game}_{6.i.1}$	$\mathsf{po}^{(i)}$	$\boxed{C_{\mathsf{Zero}}^{\prime(i)}}$	$\forall j\colon \Big(F_{i+1}^{(f)}(\boldsymbol{r}^{(j)}, X_{i+1}) \equiv 0 \vee F_{i+1}^{(f)}(\boldsymbol{r}^{(j)}, \omega_{i+1}) = 0\Big)$	security of $pi\mathcal{O}$
$\mathsf{Game}_{6.i.2}$	$\mathsf{po}^{(i)}$	$\boxed{C_{\mathsf{Zero}}^{\prime\prime(i)}}$	$\forall j\colon \Big(F_{i+1}^{(f)}(\boldsymbol{r}^{(j)}, X_{i+1}) \equiv 0 \vee \exists v \in V_j\colon \mathsf{po}_{i+1}(v) = 1\Big)$	security of $pi\mathcal{O}$
$\mathsf{Game}_{6.i.3}$	$\boxed{\mathsf{po}^{(i+1)}}$	$C_{\mathsf{Zero}}^{\prime\prime(i)}$	$\forall j\colon \Big(F_{i+1}^{(f)}(\boldsymbol{r}^{(j)}, X_{i+1}) \equiv 0 \vee \exists v \in V_j\colon \mathsf{po}_{i+1}(v) = 1\Big)$	security of PObf
$\mathsf{Game}_{6.i.4}$	$\mathsf{po}^{(i+1)}$	$\boxed{C_{\mathsf{Zero}}^{\prime\prime\prime(i)}}$	$\forall j\colon \Big(F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, r_{i+1}^{(j)}) = 0\Big)$	security of $pi\mathcal{O}$
$\mathsf{Game}_{6.i+1.0}$	$\mathsf{po}^{(i+1)}$	$\boxed{C_{\mathsf{Zero}}^{(i+1)}}$	$F_{i+1}^{(f)}(X_1, \ldots, X_{i+1}) \equiv 0$	security of $pi\mathcal{O}$

Proof (sketch). The condition of acceptance of $C_{\mathsf{Zero}}^{\prime(i)}$ is a logical or statement such that the left-hand side of the or-statement implies the right-hand side. Thus, the left-hand side of that or-statement is only conceptional. Hence, the difference between the circuits' behavior on input a valid encoding is limited to the fact that $C_{\mathsf{Zero}}^{(i)}$ only outputs 1 if $F_i^{(f)}(X_1, \ldots, X_i) \equiv 0$ as abstract polynomials, whereas $C_{\mathsf{Zero}}^{\prime(i)}$ outputs 1 if

$$\forall j \in \{1, \ldots, \nu\}\colon F_{i+1}^{(f)}(r_1^{(j)}, \ldots, r_i^{(j)}, \omega_{i+1}) = F_i^{(f)}(\boldsymbol{r}^{(j)}) = 0$$

for randomly sampled values $\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)} \leftarrow (\mathbb{Z}_q)^i$. The only event causing the two circuits to produce different outputs is that $F_i^{(f)}$ is a non-zero polynomial and $F_i^{(f)}(\boldsymbol{r}^{(j)}) = 0$ for every $j \in \{1, \ldots, \nu\}$. Applying the Schwartz-Zippel lemma upper bounds the probability for that event by $\frac{n^\nu}{q^\nu}$.

We construct a circuit sampler S_1 that on input of the security parameter $1^{poly(\lambda)}$ produces public parameters as in $\mathsf{Game}_{6.i.0}$ omitting the obfuscated circuit Λ_{zero} and outputs the implementations of the circuits $C_{\mathsf{Zero}}^{(i)}$ and $C_{\mathsf{Zero}}^{\prime(i)}$. As $poly(\lambda)$ upper bounds the input length of the two circuits, we may choose the differing domain $\mathcal{X}$ to span the entire domain of the two circuits using the map $X\colon \mathbb{N} \to \mathbb{N}$, $l \mapsto 2^l$. Therefore, for any possibly unbounded adversary $\mathcal{D}$, the advantage $Adv_{S_1,\mathcal{D}}^{\text{eq\$}}(poly(\lambda)) = 0$. Furthermore, we need to verify that for any non-uniform PPT adversary $\mathcal{D}'$, $X(poly(\lambda)) \cdot Adv_{S_1,\mathcal{D}'}^{\text{sel-ind}}(poly(\lambda))$ is negligible in λ. As the statistical distance between the outputs of $C_{\mathsf{Zero}}^{(i)}$ and $C_{\mathsf{Zero}}^{\prime(i)}$ is upper bounded by $\frac{n^\nu}{q^\nu}$ and $q \geq 2^{p(\lambda)}$, we can easily choose ν such that $X(poly(\lambda)) \cdot Adv_{S_1,\mathcal{D}'}^{\text{sel-ind}}(poly(\lambda))$

is negligible in λ. Hence, S_1 is an X-Ind Sampler. Then, we are able to construct the adversary $\mathcal{B}_1$ such that it simulates $\mathsf{Game}_{6.i.0}$ if it receives an obfuscation of $C^{(i)}_{\mathsf{Zero}}$ from $Exp^{\text{pio-ind}}_{pi\mathcal{O},S_1,\mathcal{B}_1}$ and $\mathsf{Game}_{6.i.1}$ otherwise. □

$\mathsf{Game}_{6.i.2}$. Is the same as $\mathsf{Game}_{6.i.1}$ with the difference that the obfuscation Λ_{zero} is produced for the circuit $C''^{(i)}_{\mathsf{Zero}}$ (see Fig. 13 for an implementation of the circuit $C''^{(i)}_{\mathsf{Zero}}$). Again, this game hop is justified by the security of $pi\mathcal{O}$. As po_{i+1} contains the value ω_{i+1}, the following equivalence holds

$$\exists v \in V_j : \mathsf{po}_{i+1}(v) \iff F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_i, \omega_{i+1}) = 0,$$

where V_j is the zero set of the polynomial $F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_i, X_{i+1})$. The zero set of a univariate polynomial can be computed using the Cantor-Zassenhaus (CS) algorithm [18]. The algorithm CS is a randomized Las Vegas algorithm with expected computational complexity in $O(n^3 \cdot \log(q))$ [19]. As the running time has no upper bound, we define an algorithm CS' that simulates CS for $2 \cdot T$ steps, where T is the expected running time of CS. If CS outputs the zero set during that time, CS' succeeds, otherwise CS' outputs $\perp$. Exploiting Markov's inequality, the probability that CS' succeeds if at least $\frac{1}{2}$. We define the algorithm CS'' that calls CS' $p'(\lambda)$ times. If at least one execution of CS' succeeds, CS'' outputs the (unique) zero set, otherwise CS'' outputs $\emptyset$. Hence, the probability that CS'' succeeds is at least $1 - \frac{1}{2^{p'(\lambda)}}$. The circuit $C''^{(i)}_{\mathsf{Zero}}$ uses the algorithm CS'' to compute V_j. Hence, the two circuits behave differently only if the algorithm CS'' fails to compute V_j for some $j \in \{1, \ldots, \nu\}$. Employing a union bound, the statistical difference between the outputs of $C'^{(i)}_{\mathsf{Zero}}$ and $C''^{(i)}_{\mathsf{Zero}}$ is upper bounded by $\nu \cdot \frac{1}{2^{p'(\lambda)}}$.

We construct a circuit sampler S_2 that produces public parameters as in $\mathsf{Game}_{6.i.1}$ omitting the obfuscated circuit Λ_{zero} and outputs the implementations of the circuits $C'^{(i)}_{\mathsf{Zero}}$ and $C''^{(i)}_{\mathsf{Zero}}$. By the same argument as above, we only need to verify that $X(poly(\lambda)) \cdot Adv^{\text{sel-ind}}_{S_2,\mathcal{D}'}(poly(\lambda))$ is negligible in λ for any PPT adversary $\mathcal{D}'$. As $X(poly(\lambda)) \leq 2^{poly(\lambda)}$ and $Adv^{\text{sel-ind}}_{S_2,\mathcal{D}'}(poly(\lambda)) \leq \nu \cdot \frac{1}{2^{p'(\lambda)}}$, we can easily choose $p'(\lambda)$ such that $2^{poly(\lambda)} \cdot \nu \cdot \frac{1}{2^{p'(\lambda)}}$ is negligible and, hence, $S_2 \in \mathcal{S}^{X\text{-ind}}$. We remark that at this point we make use of the fact that $pi\mathcal{O}$ is secure even if for circuit families that contain circuits that are bigger than the security parameter used to instantiate $pi\mathcal{O}$.

A crucial observation is that the value ω_{i+1} is never used explicitly in $\mathsf{Game}_{6.i.2}$. This enables to utilize the security property of the point obfuscation POBF.

$\mathsf{Game}_{6.i.3}$. Is the same as $\mathsf{Game}_{6.i.2}$ with the difference that the m-tuple po of point obfuscations is distributed as $\mathsf{po}^{(i+1)}$. That is, in $\mathsf{Game}_{6.i.2}$ po_{i+1} contains the uniformly distributed value ω_{i+1} and in $\mathsf{Game}_{6.i.3}$ po_{i+1} is produced via POBF($\perp$). This hop is justified by the security property of the point obfuscation POBF. In other words, there exists a PPT adversary $\mathcal{B}_3$, such that $|\Pr[out_{6.i.3} = 1] - \Pr[out_{6.i.2} = 1]| \leq \left| Adv^{\text{po}}_{\text{POBF},\mathcal{B}_3} \right|(\lambda)$.

We observe that the condition $\exists v \in V_j \colon \mathsf{po}_{i+1}(v)$ that is used in $C''^{(i)}_{\mathsf{Zero}}$ can not hold anymore. Thus, $C''^{(i)}_{\mathsf{Zero}}$ on input a valid encoding outputs 1 if and only if for every $j \in \{1, \ldots, m\}$, $F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_i, X_{i+1}) \equiv 0$ as abstract polynomials.

$\mathsf{Game}_{6.i.4}$. Is identical to $\mathsf{Game}_{6.i.3}$ with the difference, that Λ_{zero} is produced for the circuit $C'''^{(i)}_{\mathsf{Zero}}$ (see Fig. 13 for the implementation of the circuit $C'''^{(i)}_{\mathsf{Zero}}$). The difference between the circuits $C''^{(i)}_{\mathsf{Zero}}$ and $C'''^{(i)}_{\mathsf{Zero}}$ is limited to the fact that $C''^{(i)}_{\mathsf{Zero}}$ only outputs 1 if

$$\forall j \in \{1, \ldots, \nu\} \colon F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_i, X_{i+1}) \equiv 0,$$

whereas $C'''^{(i)}_{\mathsf{Zero}}$ uses a weaker condition and outputs 1 if

$$\forall j \in \{1, \ldots, \nu\} \colon F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_{i+1}) = 0,$$

where $\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)}$ are randomly sampled points from $(\mathbb{Z}_q)^{i+1}$. Again, the Schwartz-Zippel lemma upper bounds the probability that the circuits $C''^{(i)}_{\mathsf{Zero}}$ and $C'''^{(i)}_{\mathsf{Zero}}$ behave differently by $\frac{n^\nu}{q^\nu}$. Thus, using a similar argument as for the game hop between $\mathsf{Game}_{6.i.0}$ and $\mathsf{Game}_{6.i.1}$, there exists an X-Ind sampler S_4 and a PPT adversary $\mathcal{B}_4$, such that $|\Pr[out_{6.i.4} = 1] - \Pr[out_{6.i.3} = 1]| \leq 2 \cdot \left|Adv^{\text{pio-ind}}_{pi\mathcal{O}, S_4, \mathcal{B}_4}(poly(\lambda))\right|$.

$\mathsf{Game}_{6.i.5}$. Is identical to $\mathsf{Game}_{6.i.4}$ with the difference, that Λ_{zero} is produced for the circuit $C^{(i+1)}_{\mathsf{Zero}}$. The circuit $C''^{(i)}_{\mathsf{Zero}}$ outputs 1 if

$$\forall j \in \{1, \ldots, \nu\} \colon F^{(f)}_{i+1}(r^{(j)}_1, \ldots, r^{(j)}_{i+1}) = 0$$

for $\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\nu)} \leftarrow (\mathbb{Z}_q)^{i+1}$. The circuit $C^{(i+1)}_{\mathsf{Zero}}$ only outputs 1 if $F^{(f)}_{i+1}(X_1, \ldots, X_{i+1}) \equiv 0$ as abstract polynomials. The only event causing the two circuits to produce different outputs occurs if the polynomial $F^{(f)}_{i+1}$ is a non-zero polynomial and for all $j \in \{1, \ldots, \nu\}$, $F^{(f)}_{i+1}(\boldsymbol{r}^{(j)})$ evaluates to 0. Again, the Schwartz-Zippel lemma upper bounds the probability for that to happen by $\frac{n^\nu}{q^\nu}$. Hence, this game hop is justified by the security property of the employed obfuscator using a similar argument as above. Furthermore, $\Pr[out_{6.i.5} = 1] = \Pr[out_{6.i+1} = 1]$.

Hence, we obtain

$$\begin{aligned} |\Pr[out_7 = 1] - \Pr[out_6 = 1]| &\leq \left| \sum_{i=0}^{m-1} \Pr[out_{6.i+1} = 1] - \Pr[out_{6.i} = 1] \right| \\ &\leq 8m \cdot \left| Adv^{\text{pio-ind}}_{pi\mathcal{O}, S, \mathcal{D}_1}(poly(\lambda)) \right| \\ &\quad + m \cdot \left| Adv^{\text{po}}_{\mathrm{POBF}, \mathcal{D}_2}(\lambda) \right| \end{aligned}$$

for a suitable circuit sampler $S \in \mathcal{S}^{X\text{-ind}}$ and suitable PPT adversaries $\mathcal{D}_1$ and $\mathcal{D}_2$ concluding the proof of Lemma 5. $\square$

In $C^{(m)}_{\mathsf{Zero}}$ the matrix A is not necessary to perform the test whether $F^{(f)}_m(\boldsymbol{X})$ equals the zero polynomial. This enables to employ the security of the obfuscator to unnoticeably switch to a zero testing circuit that does not know the matrix A.

Game_8. Is identical to Game_7 except for the fact that the public parameters are sampled containing an obfuscation of the circuit $\underline{C}_{\mathsf{Zero}}$ (cf. Fig. 14). We refer to this distribution of public parameters as $\underline{pp}$. This game hop is justified by the security of $pi\mathcal{O}$ as the circuits $C^{(m)}_{\mathsf{Zero}}$ and $\underline{C}_{\mathsf{Zero}}$ are functionally equivalent.

```
CIRCUIT C_Zero[q, sk](a)
if ¬VAL(a) then return ⊥
parse a =: (C, C′, π)
f⃗ ← DEC(sk, C)
if f⃗ = (0, ..., 0)^T then
  return 1
return 0
```

Fig. 14. Zero testing circuit that does not know the matrix A.

In Game_8, the matrix A is not necessary to produce the implementation of $\underline{C}_{\mathsf{Zero}}$ as $F^{(f)}_m(\boldsymbol{X}) \equiv 0 \Leftrightarrow \vec{f} = (0, \ldots, 0)^T$ exploiting the fact that multiplication with A is an isomorphism of vector spaces. Thus, the public parameters pp in Game_8 do not contain any information about the matrix A, i.e. the only source of information about A is the oracle $\mathcal{O}$ and the challenge. This enables to apply the Randomization lemma (Lemma 3).

Game_9. Is identical to Game_8 except for the fact that the representation vectors $\vec{f} := \vec{f'}$ for the challenge encoding are sampled uniformly at random from $(\mathbb{Z}_q)^d$. This game hop is justified by the Randomization lemma (Lemma 3).

Claim. For any legitimate PPT adversary $\mathcal{A}$, there exists a legitimate (possibly unbounded) adversary $\mathcal{B}_9$ for the Randomization game $Exp^{\mathrm{rand}}_{(q,d),\mathcal{B}_9}(\lambda)$, such that $|\Pr[out_9 = 1] - \Pr[out_8 = 1]| \leq 2 \cdot \frac{d}{q}$.

Proof (sketch). We construct an adversary $\mathcal{B}_9$ for $Exp^{\mathrm{rand}}_{(q,d),\mathcal{B}_9}(\lambda)$ that simulates either Game_8 or Game_9 depending on whether $\mathcal{B}_9$ receives the real challenge from $Exp^{\mathrm{rand}}_{(q,d),\mathcal{B}_9}(\lambda)$ or not. The public parameters pp that are sampled in Game_8 and Game_9 are identically distributed, and $\mathcal{B}_9$ is able to sample pp exactly like in these games, as pp does not depend on the matrix A. In order to answer $\mathcal{A}$'s oracle queries, $\mathcal{B}_9$ uses its oracle $\mathcal{O}^{\mathrm{rand}}$. Particularly, $\mathcal{B}_9$ obtains the representation vectors that are necessary to answer an oracle query for the polynomial $P(\boldsymbol{X})$ by requesting the vector $t := \varphi^{-1}_{\mathrm{pol}}(P(\boldsymbol{c} \circ \boldsymbol{X}))$ from its own oracle $\mathcal{O}^{\mathrm{rand}}$. To obtain the representation vectors for the challenge encoding, $\mathcal{B}_9$ outputs the vector $t^* := \varphi^{-1}_{\mathrm{pol}}(P^*(\boldsymbol{c} \circ \boldsymbol{X}))$ to the Randomization game. Hence, $\mathcal{B}_9$ simulates Game_8 if the Randomization game provides the real challenge. Otherwise, $\mathcal{B}_9$ simulates Game_9. By premise, the adversary $\mathcal{A}$ is legitimate with respect to the

$(m,\, n)$-Interactive Uber assumption, i.e. $P^*(\boldsymbol{X}) \notin \langle 1,\, P_1(\boldsymbol{X}),\, \dots,\, P_l(\boldsymbol{X})\rangle$ and for any $P(\boldsymbol{X}) \in \{P^*(\boldsymbol{X}),\, P_1(\boldsymbol{X}),\, \dots,\, P_l(\boldsymbol{X})\}$, $\deg(P(\boldsymbol{X})) \le n$. Hence, due to Lemma 4 and exploiting the fact that φ_{pol} is an isomorphism of vector spaces, $\varphi_{\mathrm{pol}}^{-1}(P^*(\boldsymbol{c} \circ \boldsymbol{X})) \notin \langle e_1,\, \varphi_{\mathrm{pol}}^{-1}(P_1(\boldsymbol{c} \circ \boldsymbol{X})),\, \dots,\, \varphi_{\mathrm{pol}}^{-1}(P_l(\boldsymbol{c} \circ \boldsymbol{X}))\rangle$, where $P_1(\boldsymbol{X}),\, \dots,\, P_l(\boldsymbol{X})$ are the polynomials $\mathcal{A}$ queries from its oracle. Therefore, $\mathcal{B}_9$ is legitimate. $\square$

Game_{10}. Is identical to Game_9 with the difference that the public parameters pp are sampled with an obfuscation of $C_{\mathsf{Zero}}^{(m)}$ instead of an obfuscation of $\underline{C}_{\mathsf{Zero}}$. The analysis of this game hop is analogous to the analysis of the game hop from Game_7 to Game_8. Hence, in Game_{10} the public parameters are distributed as $\overline{pp}^{(m)}$.

Game_{11}. Is the same as Game_{10} except for the fact that the public parameters are again distributed as $\overline{pp}^{(0)}$ instead of being distributed as $\overline{pp}^{(m)}$. This game hop is justified by the security property of $pi\mathcal{O}$ and the point obfuscation $\textsc{POBF}$. The analysis is analogous to the proof of Lemma 5.

Game_{12}. Is the same as Game_{11} except for the fact that com is produced via $\textsc{Commit}_{ck}(A)$. The analysis is analogous to the analysis of the game hop from Game_5 to Game_6.

Game_{13}. Is the same as Game_{12} with the difference that Game_{13} produces consistency proofs for relation $\mathcal{R}_1$ or relation $\mathcal{R}_2$ depending on whether the corresponding representation is constant or not. The representation vectors $\vec{f} := \vec{f'}$ for the challenge encoding are sampled uniformly at random from $(\mathbb{Z}_q)^d$. Nevertheless, the resulting representation is consistent. As Π is perfectly witness-indistinguishable under a hiding CRS, $\Pr[out_{13} = 1] - \Pr[out_{12} = 1] = 0$.

Game_{14}. Is the same as Game_{13} with the difference that in Game_{13} pp is distributed as $\widehat{pp}$, whereas in Game_{14} pp is distributed as $\widetilde{pp}$. Hence, this game hop is justified by the Swap lemma (Lemma 1). We remark that the witness w_y for the statement $y \in \mathrm{TD}$ is never used in Game_{14}.

Game_{15}. Is identical to Game_{14} except for the fact that the public parameters pp are sampled according to $\textsc{Setup}(1^\lambda)$. The analysis is similar to the analysis of the game hop form Game_2 to Game_3.

Game_{16}. Is the same as Game_{15} with the difference that group element encodings are sampled in a uniform manner as in the sampling algorithm $\textsc{Sam}$, i.e. using constant representations. In particular, the representation vectors for the challenge encoding are computed via

$$\vec{f} := \vec{f'} := \left(\varphi_{\mathrm{pol}}\left(A^{-1} \cdot v^*\right)(\boldsymbol{\omega}), 0, \dots, 0\right)^T, \text{ for } v^* \leftarrow (\mathbb{Z}_q)^d .$$

The analysis of this game hop is similar to the analysis of the game hop form Game_1 to Game_2 using a hybrid argument and Lemma 2. We observe that Game_{16} does not use the value $\boldsymbol{c}$ explicitly.

Game_{17}. Is identical to Game_{16} with the difference that $\boldsymbol{s}$ is directly sampled uniformly at random from $(\mathbb{Z}_q)^m$. In both games Game_{16} and Game_{17}, $\boldsymbol{s}$ is uniformly

and independently distributed over $(\mathbb{Z}_q)^m$, if $\boldsymbol{\omega}$ does not contain a zero entry. As $\boldsymbol{\omega}$ is chosen uniformly at random from $(\mathbb{Z}_q)^m$, $|\Pr[out_{17} = 1] - \Pr[out_{16} = 1]| \leq \frac{m}{q}$.
$\mathsf{Game}_{\mathbf{18}}$**.** Is identical to Game_{17} except for the fact that the challenge encoding is sampled using the representation $\overrightarrow{f} := \overrightarrow{f'} := (r, 0, \ldots, 0)^T$, where r is sampled uniformly at random from $\mathbb{Z}_q$. In Game_{17}, the vector v^* is sampled uniformly at random from $(\mathbb{Z}_q)^d$, and multiplication with the matrix A defines a bijection on $(\mathbb{Z}_q)^d$ that does not depend on v^*. Thus, the vector $A^{-1} \cdot v^*$ is uniformly distributed over $(\mathbb{Z}_q)^d$. Therefore, Game_{17} and Game_{18} are identically distributed, which is why $\Pr[out_{18} = 1] - \Pr[out_{17} = 1] = 0$.

Therefore, for any legitimate PPT adversary $\mathcal{A}$, there exist legitimate PPT adversaries $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$, and $\mathcal{D}_6$ and a polynomial $l = l(\lambda)$ such that

$$\begin{aligned}\left|Adv^{\text{uber}}_{\Gamma_{m,n},\mathcal{A}}(\lambda)\right| \;\leq\; & \frac{d+m}{q} + (6l+7)\cdot\left|Adv^{\text{smp}}_{\mathcal{TD},\mathcal{D}_1}(\lambda)\right| + m\cdot\left|Adv^{\text{po}}_{\text{POBF},\mathcal{D}_5}(\lambda)\right| \\ & + (12l+14)\cdot\left|Adv^{\text{crs}}_{\Pi,\mathcal{D}_3}(\lambda)\right| + (8l+8)\cdot\left|Adv^{\text{ind-cpa}}_{\text{HPKE},\mathcal{D}_4}(\lambda)\right| \\ & + (28l+8m+34)\cdot\left|Adv^{\text{pio-ind}}_{pi\mathcal{O},\mathcal{D}_2}(poly(\lambda))\right| \\ & + 2\cdot\left|Adv^{\text{hiding}}_{\text{COM},\mathcal{D}_6}(\lambda)\right|. \end{aligned} \tag{19}$$

As m and l grow at most polynomially in λ, the advantage $Adv^{\text{uber}}_{\Gamma_{m,n},\mathcal{A}}(\lambda)$ is negligible in λ concluding the proof. $\square$

C Point obfuscation from DDH

Let $\mathcal{G} = \{\mathcal{G}_\lambda\}$ be a family of finite cyclic groups of prime order p such that the DDH assumption holds. On input the value $x \in \mathbb{Z}_p$, $\text{POBF}(x)$ samples a random generator $g \leftarrow \mathsf{Gens}_{\mathcal{G}}$ and two values $r, r' \leftarrow \mathbb{Z}_p^\times$, and outputs the tuple $(g^r, g^{r\cdot x}, g^{r'}, g^{r'\cdot x})$. On input $\perp$, $\text{POBF}(\perp)$ samples a random generator $g \leftarrow \mathsf{Gens}_{\mathcal{G}}$ and four values $x, x', r, r' \leftarrow \mathbb{Z}_p^\times$ such that $x \neq x'$, and outputs the tuple $(g^r, g^{r\cdot x}, g^{r'}, g^{r'\cdot x'})$. Given the description $\mathsf{po} =: (A, B, A', B')$ of a point function, $\mathsf{po}(y)$ evaluates to 1 if and only if $A^y = B$ and $A'^y = B'$ for some point $y \in \mathbb{Z}_p$.

A point obfuscation with message space $\mathcal{M}_\lambda = \mathbb{Z}_q$ for a prime q that is much larger than p can be constructed using the above point obfuscation with message space $\mathbb{Z}_p$. Let $l \in \mathbb{N}$ be the smallest natural number such that $p^l \geq q$. Essentially, the idea is to produce a point obfuscation for the first component of the p-adic representation of elements in $\mathbb{Z}_q$. Particularly, on input $y \in \mathbb{Z}_q$, POBF' produces po_0 for $y \pmod p$ and appends the remaining p-adic representation in the clear. On input $\perp$, POBF' produces po_0 for $\perp$ and generates a random value $y \leftarrow \mathbb{Z}_q$ and appends the p-adic representation of q omitting the first component. The proof of security relies on the security of the underlying point obfuscation POBF and the fact that uniform distribution over $\mathbb{Z}_p$ and uniform distribution over $\mathbb{Z}_q$ modulo p are statistically close if the quotient $\frac{p}{q}$ is negligible in λ.

Lemma 6. *Let* POBF *be a point obfuscation with message space* $\mathbb{Z}_p$ *for a prime* p *and let* $q \geq p^l$ *be a prime such that* $\frac{p}{q}$ *is negligible in* λ*. Then,* POBF$'$ *as described in Fig. 15 is a point obfuscation for message space* $\mathbb{Z}_q$*.*

ALGORITHM POBF$'(y)$	$\mathsf{po}(x)$
if $y = \bot$ **then** $x \leftarrow \mathbb{Z}_q$	let $(x_0, \ldots, x_{l-1})$ s. t.
else $x := y$	$x = \sum_{i=0}^{l-1} x_i \cdot p^i$ and
let $(x_0, \ldots, x_{l-1})$ s. t.	$0 \leq x_i < p$ for any $0 \leq i < l$
$x = \sum_{i=0}^{l-1} x_i \cdot p^i$ and	**return** $(\mathsf{po}_0(x_0) = 1) \wedge$
$0 \leq x_i < p$ for any $0 \leq i < l$	$(\mathsf{po}_i = x_i$ for $1 \leq i < l)$
if $y = \bot$ **then** $\mathsf{po}_0 \leftarrow \mathrm{POBF}(\bot)$	
else $\mathsf{po}_0 \leftarrow \mathrm{POBF}(x_0)$	
return $\mathsf{po} := (\mathsf{po}_0, x_1, \ldots, x_{l-1})$	

Fig. 15. The description of POBF$'$ (left) and the map defined by *po* (right).

Proof (sketch). Let $\mathcal{A}$ be a PPT adversary. To prove this statement, we proceed over a series of games. Let out_i denote the output of Game_i.

$\mathsf{Game_0}$. This game produces $\mathsf{po} \leftarrow \mathrm{POBF}'(x)$ for $x \leftarrow \mathbb{Z}_q$, calls the adversary $\mathcal{A}$ on input $(1^\lambda, \mathsf{po})$, and outputs $\mathcal{A}$'s output. We have that $\Pr[out_0 = 1] = \Pr[\mathcal{A}(\mathsf{po}) = 1 \mid \mathsf{po} \leftarrow \mathrm{POBF}(x), x \leftarrow \mathbb{Z}_q]$.

$\mathsf{Game_1}$. This game samples a value $\widetilde{x}_0 \leftarrow \mathbb{Z}_p$ and produces $\mathsf{po}_0 \leftarrow \mathrm{POBF}(\widetilde{x}_0)$. The value x is sampled from $\mathbb{Z}_q$ conditioned on $x = \widetilde{x}_0 \pmod p$. Further, $x_0, \ldots, x_{l-1}$ are produced as in Game_0. This game invokes $\mathcal{A}$ with $\mathsf{po} := (\widetilde{x}_0, x_1, \ldots, x_{l-1})$. The statistical distance between uniform distribution over $\mathbb{Z}_p$ and uniform distribution over $\mathbb{Z}_q$ reduced modulo p is upper bounded by $\frac{p}{2q}$, which is why $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq \frac{p}{2q}$.

As the message used to produce the point obfuscation po_0 is uniformly distributed over the message space $\mathbb{Z}_p$, we are able to exploit the security property of POBF.

$\mathsf{Game_2}$. Is identical to Game_1 with the difference that po_0 is produced for $\bot$. Hence, $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq Adv^{\mathrm{po}}_{\mathrm{POBF},\mathcal{B}_2}(\lambda)$.

$\mathsf{Game_3}$. Is the same as Game_2 except for the fact that x is sampled uniformly at random from $\mathbb{Z}_q$, i.e. $x_0 = x \pmod p$ is not distributed uniformly anymore. Applying a similar argument as above, we obtain $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq \frac{p}{2q}$. Furthermore, we have that $\Pr[out_3 = 1] = \Pr\big[\mathcal{A}(\mathsf{po}) = 1 \bigm| \mathsf{po} \leftarrow \mathrm{POBF}'(\bot)\big]$. $\square$