Multilinear Maps from Obfuscation

Martin R. Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, Kenneth G. Paterson

Multilinear Maps from Obfuscation

Martin R. Albrecht (RHUL) Pooya Farshim (QUB) Dennis Hofheinz (KIT)
Enrique Larraia (RHUL) Kenneth G. Paterson (RHUL)

October 20, 2016

Abstract

We provide constructions of multilinear groups equipped with natural hard problems from indistinguishability obfuscation, homomorphic encryption, and NIZKs. This complements known results on the constructions of indistinguishability obfuscators from multilinear maps in the reverse direction.

We provide two distinct, but closely related constructions and show that multilinear analogues of the DDH assumption hold for them. Our first construction is *symmetric* and comes with a κ-linear map $\mathbf{e} : \mathbb{G}^\kappa \longrightarrow \mathbb{G}_T$ for prime-order groups $\mathbb{G}$ and $\mathbb{G}_T$. To establish the hardness of the κ-linear DDH problem, we rely on the existence of a base group for which the $(\kappa - 1)$-strong DDH assumption holds. Our second construction is for the *asymmetric* setting, where $\mathbf{e} : \mathbb{G}_1 \times \cdots \times \mathbb{G}_\kappa \longrightarrow \mathbb{G}_T$ for a collection of $\kappa + 1$ prime-order groups $\mathbb{G}_i$ and $\mathbb{G}_T$, and relies only on the standard DDH assumption in its base group. In both constructions the linearity κ can be set to any arbitrary but a priori fixed polynomial value in the security parameter.

We rely on a number of powerful tools in our constructions: (probabilistic) indistinguishability obfuscation, dual-mode NIZK proof systems (with perfect soundness, witness indistinguishability and zero knowledge), and additively homomorphic encryption for the group $\mathbb{Z}_N^+$. At a high level, we enable "bootstrapping" multilinear assumptions from their simpler counterparts in standard cryptographic groups, and show the equivalence of IO and multilinear maps under the existence of the aforementioned primitives.

Keywords. Multilinear map, indistinguishability obfuscation, homomorphic encryption, decisional Diffie–Hellman, Groth–Sahai proofs.

Contents

1 Introduction

1.1 Main contribution

In this paper, we explore the relationship between multilinear maps and obfuscation. Our main contribution is a construction of multilinear maps for groups of prime order equipped with natural hard problems, using indistinguishability obfuscation (IO) in combination with other tools, namely NIZK proofs, homomorphic encryption, and a base group $\mathbb{G}_0$ satisfying a mild cryptographic assumption. This complements known results in the reverse direction, showing that various forms of indistinguishability obfuscation can be constructed from multilinear maps [GGH+13b, CLTV15, Zim15]. The relationship between IO and multilinear maps is a very natural question to study, given the rich diversity of cryptographic constructions that have been obtained from both multilinear maps and obfuscation, and the apparent fragility of current constructions for multilinear maps. More on this below.

We provide two distinct but closely related constructions. One is for multilinear maps in the *symmetric* setting, that is non-degenerate multilinear maps $\mathbf{e} : \mathbb{G}_1{}^\kappa \longrightarrow \mathbb{G}_T$ for groups $\mathbb{G}_1$ and $\mathbb{G}_T$ of prime order N. Our construction relies on the existence of a base group $\mathbb{G}_0$ in which the $(\kappa-1)$-SDDH assumption holds—this states that, given a κ-tuple of $\mathbb{G}_0$-elements $(g, g^\omega, \ldots, g^{\omega^{\kappa-1}})$, we cannot efficiently distinguish g^{ω^κ} from a random element of $\mathbb{G}_0$. Under this assumption, we prove that the κ-MDDH problem, a natural analogue of the DDH problem as stated below, is hard.

> **(The κ-MDDH problem, informal)** Given a generator g_1 of $\mathbb{G}_1$ and $\kappa+1$ group elements $g_1^{a_i}$ in $\mathbb{G}$ with $a_i \hookleftarrow\$\ \mathbb{Z}_N$, distinguish $\mathbf{e}(g_1,\ldots,g_1)^{\prod_{i=1}^{\kappa+1} a_i}$ from a random element of $\mathbb{G}_T$.

This problem can be used as the basis for several cryptographic constructions [BS03] including, as the by now the classic example of multiparty non-interactive key exchange (NIKE) [GGH13a].

Our other construction is for the *asymmetric* setting, that is multilinear maps $\mathbf{e} : \mathbb{G}_1 \times \cdots \times \mathbb{G}_\kappa \longrightarrow \mathbb{G}_T$ for a collection of κ groups $\mathbb{G}_i$ and $\mathbb{G}_T$ all of prime order N. It uses a base group $\mathbb{G}_0$ in which we require only that the standard DDH assumption holds. For this construction, we show that a natural asymmetric analogue of the κ-MDDH assumption holds (wherein all but two of the $\kappa+1$ group elements input to $\mathbf{e}$ come from distinct groups).

In Section 7, we also show the intractability of the *rank problem* for our construction for multilinear maps in the symmetric setting; this is a generalization of DDH-like problems to matrices that has proven to be useful in cryptographic constructions [BHHO08, NS09, GHV12, BLMR13, EHK+13].

At a high level, then, our constructions are able to "bootstrap" from rather mild assumptions in a standard cryptographic group to much stronger multilinear assumptions in a group (or groups, in the asymmetric setting) equipped with a κ-linear map. Here κ is fixed up-front at construction time, but is otherwise unrestricted. Of course, such constructions cannot be expected to come "for free," and we need to make use of powerful tools including probabilistic IO (PIO) for obfuscating randomized circuits [CLTV15], dual-mode NIZK proofs enjoying perfect soundness (for a binding CRS), perfect witness indistinguishability (for a hiding CRS), and perfect zero knowledge, and additive homomorphic encryption for the group $(\mathbb{Z}_N, +)$ (or alternatively, a perfectly correct FHE scheme). It is an important open problem arising from our work to weaken the requirements on, or remove altogether, these additional tools.

1.2 General approach

Our approach to obtaining multilinear maps in the symmetric setting is as follows (with many details to follow in the main body).[1] Let $\mathbb{G}_0$ with generator g_0 be a group of prime order N in which the $(\kappa-1)$-SDDH assumption holds.

We work with redundant encodings of elements h of the base group $\mathbb{G}_0$ of the form $h = g_0^{x_0}(g_0^{\omega})^{x_1}$ where g_0^{ω} comes from a $(\kappa-1)$-SDDH instance; we write $\mathbf{x} = (x_0, x_1)$ for the vector of exponents *representing* h. Then $\mathbb{G}_1$ consists of all strings of the form $(h, \mathbf{c}_1, \mathbf{c}_2, \pi)$ where $h \in \mathbb{G}_0$, ciphertext $\mathbf{c}_1$ is a homomorphic encryption under public key pk_1 of a vector $\mathbf{x}$ representing h, ciphertext $\mathbf{c}_2$ is a homomorphic encryption under a second public key pk_2 of another vector $\mathbf{y}$ also representing h, and π is a NIZK proof showing consistency of the two vectors $\mathbf{x}$ and $\mathbf{y}$, i.e., a proof that the plaintexts $\mathbf{x}$, $\mathbf{y}$ underlying $\mathbf{c}_1$, $\mathbf{c}_2$ encode the *same* group element h. Note that each element of the base group $\mathbb{G}_0$ is multiply represented when forming elements in $\mathbb{G}_1$, but that equality of group elements in $\mathbb{G}_1$ is easy to test. An alternative viewpoint is to consider $(\mathbf{c}_1, \mathbf{c}_2, \pi)$ as being *auxiliary information* accompanying element $h \in \mathbb{G}_0$; we prefer the perspective of redundant encodings, and our abstraction in Section 3 is stated in such terms. When viewed in this way, our approach can be seen as closely related to the Naor–Yung paradigm for constructing CCA-secure PKE [NY90].

Addition of two elements in $\mathbb{G}_1$ is carried out by an obfuscation of a circuit C_{Add} that is published along with the groups. It has the secret keys sk_1, sk_2 hard-coded in; it first checks the respective proofs, then uses the additive homomorphic property of the encryption scheme to combine ciphertexts, and finally uses the secret keys sk_1, sk_2 as witnesses to generate a new NIZK proof showing equality of encodings. Note that the new encoding is as compact as that of the two input elements.

The multilinear map on inputs $(h_i, \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i)$ for $1 \le i \le \kappa$ is computed using the obfuscation of a circuit C_{Map} that has sk_1 and ω hard-coded in. This allows C_{Map} to "extract" full exponents of h_i in the form $(x_{i,1} + \omega \cdot x_{i,2})$ from $\mathbf{c}_{i,1}$, and thereby compute the element $g_0^{\prod_i (x_{i,1}+\omega\cdot x_{i,2})}$. This is defined to be the output of our multilinear map $\mathbf{e}$, and so our target group $\mathbb{G}_T$ is in fact $\mathbb{G}_0$, the base group. The multilinearity of $\mathbf{e}$ follows immediately from the form of the exponent.

In the asymmetric case, the main difference is that we work with different values ω_i in each of our input groups $\mathbb{G}_i$. However, the groups are all constructed via redundant encodings, just as above.

This provides a high-level view of our approach, but no insight into why the approach achieves our aim of building multilinear maps with associated hard problems. Let us give some intuition on why the κ-MDDH problem is hard in our setting. We transform a κ-MDDH tuple $\mathbf{h} = ((g_1^{a_i})_{i\le\kappa+1}, g_T^{d})$, where d is the product of the $a_i \in \mathbb{Z}_N$, g_1 is in the "encoded" form above, thus $g_1 = (h_1, \mathbf{c}_1, \mathbf{c}_2, \pi)$, and g_T is a generator of $\mathbb{G}_T = \mathbb{G}_0$, into another κ-MDDH tuple $\mathbf{h}'$ with exponents $a_i' = a_i + \omega$ for $i \le \kappa$. This means that the exponent of the challenge element in the target group $d' = \prod_1^{\kappa}(a_i + \omega)a_{\kappa+1}$ can be seen as a degree κ polynomial in ω. Therefore, with the knowledge of the a_i and a $(\kappa-1)$-SDDH challenge, with ω implicit in the exponent, we are able to randomize $g_T^{d'}$ replacing $g_T^{\omega^\kappa}$ with a uniform value.

Nevertheless, in the preceding simplistic argument we have made two assumptions. The first is that we are able to provide an obfuscation of a circuit C'_{Map} that has the same functionality as

[1] This version of the paper fixes a flaw that we found in the proof of Theorem 5.3. The construction of Section 4 has been slightly modified, but it does not make use of stronger assumptions and has comparable efficiency.

C_{Map} over $\mathbb{G}_1$ *without* the explicit knowledge of ω. We resolve this by showing a way of evaluating the κ-linear map on any elements of $\mathbb{G}_1$ using only the powers $g_0^{\omega^i}$ for $1 \leq i \leq \kappa - 1$, and vectors extracted from the accompanying ciphertexts, and then applying IO to the two circuits.[2]

The second assumption we made is that we can indeed switch from $\mathbf{h}$ to $\mathbf{h}'$ without being noticed. In other words, that the vectors $\mathbf{x}_i$, $\mathbf{y}_i$ representing g^{a_i} can be replaced (without being noticed) with vectors $\mathbf{h}'_i$ whose second coordinate is always fixed. Intuitively this is based on the IND-CPA security of the FHE scheme, but in order to give a successful reduction we also have to change the circuit C_{Add} (since C_{Add} uses both decryption keys). We show two ways to do this: one is based on probabilistic indistinguishability obfuscation [CLTV15], and the other uses only (deterministic) indistinguishability obfuscation, and additionally exploits the specific structure of a particular (pairing-based) NIZK implementation due to Groth and Sahai [GS08].

We note that in this work we do not construct graded encoding schemes as in [GGH13a]. That is, we do not construct maps from $\mathbb{G}_i \times \mathbb{G}_j$ to $\mathbb{G}_{i+j}$. On the other hand, our construction is noiseless and is closer to multilinear maps as defined by Boneh and Silverberg [BS03].

1.3 Attacks on multilinear maps

Multilinear maps have been in a state of turmoil, with the discovery of attacks [CHL+15, HJ15, CLR15, MF15, Cor15] against the GGH13 [GGH13a], CLT [CLT13] and GGH15 [GGH15] proposals, and a sequence of countermeasures and fixes [CLT15, CGH+15], which since have been broken, too. Hence, our confidence in constructions for graded encoding schemes (and thereby multilinear maps) has been shaken. On the other hand, when IO is constructed from graded encoding schemes via Barrington's theorem or dual-input straddling sets [GGH+13b, AB15, Zim15], then none of the known attacks on graded encoding schemes seem to apply [CGH+15]. Indeed, when building IO from multilinear maps one restricts the pool of available operations to an attacker by fixing a circuit a priori which means that certain "interesting" elements cannot be (easily) constructed. Hence, currently it is perhaps more plausible to assume that IO exists than it is to assume that secure multilinear maps exist. However, we stress that more cryptanalysis of IO constructions is required to investigate what security they provide.

Moreover, even though current constructions for IO rely on graded encoding schemes, it is not implausible that alternative routes to achieving IO without relying on multilinear maps will emerge in due course. And setting aside the novel applications obtained directly from IO, multilinear maps, and more generally graded encoding schemes, have proven to be very fruitful as constructive tools in their own right (cf. [BS03, PTT10], resp., [FHPS13, GGH+13c, HSW13, GGSW13, BWZ14, TLL14, BLR+15]). This rich set of applications coupled with the current uncertainty over the status of graded encoding schemes and multilinear maps provides additional motivation to ask what additional tools are needed in order to upgrade IO to multilinear maps. As an additional benefit, we upgrade (via IO) noisy graded encoding schemes to clean multilinear maps—sometimes now informally called "dream" or "ideal" multilinear maps.

1.4 Related work

The closest related work to ours is that of Yamakawa et al. [YYHK14, YYHK15]; indeed, their work was the starting point for ours. Yamakawa et al. construct a *self-pairing map*, that is a bilinear map

[2] This is not trivial since the new method should not lead to an exponential blow-up in κ.

from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}$; multilinear maps can be obtained by iterating their self-pairing. Their work is limited to the RSA setting. It uses the group of signed quadratic residues modulo a Blum integer N, denoted QR_N^+, to define a pairing function that, on input elements g^x, g^y in QR_N^+, outputs g^{2xy}. In their construction, elements of QR_N^+ are augmented with auxiliary information to enable the pairing computation—in fact, the auxiliary information for an element g^x is simply an obfuscation of a circuit for computing the $2x$th power modulo $\mathrm{ord}(QR_N^+)$, and the pairing is computed by evaluating this circuit on an input g^y (say). The main contribution of [YYHK14] is in showing that these obfuscated circuits leak nothing about x or the group order.

A nice feature of their scheme is that the degree of linearity κ that can be accommodated is not limited up-front in the sense that the pairing output is also a group element to which further pairing operations (derived from auxiliary information for other group elements) can be applied. However, the construction has several drawbacks. First, the element output by the pairing does not come with auxiliary information.[3] Second, the size of the auxiliary information for a product of group elements grows exponentially with the length of the product, as each single product involves computing the obfuscation of a circuit for multiplying, with its inputs already being obfuscated circuits. Third, the main construction in [YYHK14] only builds hard problems for the self-pairing of the computational type (in fact, they show the hardness of the computational version of the κ-MDDH problem in QR_N^+ assuming that factoring is hard). Still, this is sufficient for several cryptographic applications.

In contrast, our construction is *generic* with respect to its platform group. Furthermore, the equivalent of the auxiliary information in our approach does not itself involve any obfuscation. Consequently, the description of a product of group elements stays compact. Indeed, given perfect additive homomorphic encryption for $(\mathbb{Z}_p, +)$, we can perform arbitrary numbers of group operations in each component group $\mathbb{G}_i$. It is an open problem to find a means of augmenting our construction with the equivalent of auxiliary information in the *target* group $\mathbb{G}_T$, to make our multilinear maps amenable to iteration and thereby achieve graded maps as per [GGH13a, CLT13].

2 Preliminaries

2.1 Notation

We denote the security parameter by $\lambda \in \mathbb{N}$ and assume that it is implicitly given to all algorithms in the unary representation 1^λ. By an algorithm we mean a stateless Turing machine. Algorithms are randomized unless stated otherwise and PPT as usual stands for "probabilistic polynomial-time" in the (unary) security parameter. Given a randomized algorithm $\mathcal{A}$ we denote the action of running $\mathcal{A}$ on input(s) $(1^\lambda, x_1, \ldots)$ with fresh random coins r and assigning the output(s) to $y_1, \ldots$ by $(y_1, \ldots) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, x_1, \ldots; r)$. For a finite set X, we denote its cardinality by $|X|$ and the action of sampling a uniformly random element x from X by $x \leftarrow\!\!\$\ X$. Vectors are written in boldface $\mathbf{x}$ and by slight abuse of notation, running algorithms on vectors of elements indicates component-wise operation. Throughout the paper $\perp$ denotes a special error symbol, and $\mathrm{poly}(\cdot)$ stands for a fixed

[3] The authors of [YYHK14] state that such information can be added in their construction, but what would be needed is the obfuscation of a circuit for computing $4xy$th powers. The information available for building this would be obfuscations of circuits for computing $2x$th and $2y$th powers, so an obfuscation of a *composition* of *already* obfuscated circuits would be required. Strictly speaking then, the auxiliary information associated with elements output by their pairing is of a different type to that belonging to the inputs, making it questionable whether "self-pairing" is the right description of what is constructed in [YYHK14].

polynomial. A real-valued function $\mathrm{negl}(\lambda)$ is negligible if $\mathrm{negl}(\lambda) \in \mathcal{O}(\lambda^{-\omega(1)})$. We denote the set of all negligible functions by NEGL, and use $\mathrm{negl}(\lambda)$ to denote an unspecified negligible function.

2.2 Homomorphic public-key encryption

CIRCUITS. A polynomial-sized deterministic circuit family $\mathcal{C} := \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is a sequence of sets of $\mathrm{poly}(\lambda)$-sized circuits for a fixed polynomial poly. We assume that for all $\lambda \in \mathbb{N}$, all circuits $C \in \mathcal{C}_\lambda$ share a common input domain $(\{0,1\}^\lambda)^{a(\lambda)}$, where $a(\lambda)$ is a the arity of the circuit family, and codomain $\{0,1\}^\lambda$. A randomized circuit family is defined similarly except that the circuits now also take random coins $r \in \{0,1\}^{r(\lambda)}$. To make the coins used by a circuit explicit (e.g., to view a randomized circuit as a deterministic one) we write $C(x;r)$.

SYNTAX AND COMPACTNESS. A tuple of PPT algorithms $\Pi := (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Eval})$ is called a homomorphic public-key encryption (HPKE) scheme for deterministic circuit family $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ of arity $a(\lambda)$ if $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ is a conventional public-key encryption scheme with message space $\{0,1\}^\lambda$ and **Eval** is a *deterministic* algorithm that on input a public key pk a circuit $C \in \mathcal{C}_\lambda$ and ciphertexts $c_1, \ldots, c_{a(\lambda)}$ outputs a ciphertext c. We require HPKE schemes to be *compact* in the sense that the outputs of **Eval** have a size that is bounded by a polynomial function of the security parameter (and independent of the size of the circuit). Without loss of generality, we assume that secret keys of an HPKE scheme are the random coins used in key generation. This will allow us to check key pairs for validity.

CORRECTNESS. We require the following *perfect* correctness requirements from a HPKE scheme. (1) Scheme $\Pi := (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ is perfectly correct as a PKE scheme; that is for any $\lambda \in \mathbb{N}$, any $(sk, pk) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)$, any $m \in \{0,1\}^\lambda$, and any $c \leftarrow\!\!\$\ \mathbf{Enc}(m, pk)$ we have that $\mathbf{Dec}(c, sk) = m$. (2) The evaluation algorithm in also perfectly correct in the sense that for any $\lambda \in \mathbb{N}$, any $(sk, pk) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)$, any $m_i \in \{0,1\}^\lambda$ for $i \in [a(\lambda)]$, any $c_i \leftarrow\!\!\$\ \mathbf{Enc}(m_i, pk)$, any $C \in \mathcal{C}_\lambda$ and any $c \leftarrow \mathbf{Eval}(pk, C, c_1, \ldots, c_{a(\lambda)})$ we have that $\mathbf{Dec}(c, sk) = C(m_1, \ldots, m_{a(\lambda)})$.

We note that although most proposals in the literature for HPKE are not perfectly correct, this is usually assumed in the literature (cf. [GGI+14]). Indeed, it is plausible that perfectly correct HPKE can be achieved from standard HPKE constructions by adapting the probability distribution of the noise to a bounded distribution and by applying worst-case bounds in all steps. Moreover, in this work we will only need a mod-p additively homomorphic scheme of arity 2, traditionally known as a singly homomorphic PKE scheme for $(\mathbb{Z}_p, +)$. Formally, such a scheme corresponds to a family of circuits of arity 2 which add two λ-bit numbers modulo λ-bit primes p.

SECURITY. The IND-CPA security of an HPKE scheme is defined identically to a standard PKE scheme without reference to the **Dec** and **Eval** algorithms. Formally, we require that for any legitimate PPT adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathbf{Adv}^{\text{ind-cpa}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\text{IND-CPA}^{\mathcal{A}}_{\Pi}(\lambda)\right] - 1 \in \text{NEGL}\ ,$$

where game $\text{IND-CPA}^{\mathcal{A}}_{\Pi}(\lambda)$ is shown in Figure 1 (left). Adversary $\mathcal{A}$ is legitimate if it outputs two messages of equal lengths.

2.3 Obfuscators

SYNTAX AND CORRECTNESS. A PPT algorithm **Obf** is called an *obfuscator* for (deterministic or

randomized) circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if **Obf** on input the security parameter 1^λ and the description of a (deterministic or randomized) circuit $C \in \mathcal{C}_\lambda$ outputs a deterministic circuit $\overline{C}$. For deterministic circuits, we require **Obf** to be perfectly correct in the sense the circuits C and $\overline{C}$ are functionally equivalent; that is, that for all $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$, all $\overline{C} \leftarrow\!\!\$\ \mathbf{Obf}(1^\lambda, C)$, and all $m_i \in \{0,1\}^\lambda$ for $i \in [a(\lambda)]$ we have that $C(m_1, \ldots, m_{a(\lambda)}) = \overline{C}(m_1, \ldots, m_{a(\lambda)})$. For randomized circuits, the authors of [CLTV15] define correctness via computational indistinguishability of the outputs of C and $\overline{C}$. For our constructions we do *not* rely on this property and instead require that C and $\overline{C}$ are functionally equivalent up to a change in randomness; that is, for all $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$, all $\overline{C} \leftarrow\!\!\$\ \mathbf{Obf}(1^\lambda, C)$ and all $m_i \in \{0,1\}^\lambda$ for $i \in [a(\lambda)]$ we require there is an r such that $\overline{C}(m_1, \ldots, m_{a(\lambda)}) = C(m_1, \ldots, m_{a(\lambda)}; r)$. In this paper by correctness we refer to this latter property. We note that the construction from [CLTV15] is correct as it relies on a correct (indistinguishability) obfuscator (and a PRF to internally generate the required random coins).

SECURITY. The security of an obfuscator **Obf** requires that for any legitimate PPT adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$

$$\mathbf{Adv}^{\text{ind}}_{\mathbf{Obf},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\text{IND}^{\mathcal{A}}_{\mathbf{Obf}}(\lambda)\right] - 1 \in \text{NEGL}\,,$$

where game IND is shown in Figure 1 (middle). Depending on the notion of legitimacy different security notions for the obfuscator emerge; we consider two such notions below.

FUNCTIONALLY EQUIVALENT SAMPLERS. We call (the first phase of) $\mathcal{A}$ a *functionally equivalent sampler* if for any (possibly unbounded) distinguisher $\mathcal{D}$

$$\mathbf{Adv}^{\text{eq}}_{\mathcal{A},\mathcal{D}}(\lambda) := \Pr\left[C_0(x) \neq C_1(x) : (C_0, C_1, \text{st}) \leftarrow\!\!\$\ \mathcal{A}_1(1^\lambda); x \leftarrow\!\!\$\ \mathcal{D}(C_0, C_1, \text{st})\right] \in \text{NEGL}\,.$$

The security notion associated with equivalent samplers is called *indistinguishability*. We call an obfuscator meeting this level of security an *indistinguishability obfuscator* [GGH+13b] and use **IO** instead of **Obf** to emphasize this.

X-IND SAMPLERS [CLTV15]. Roughly speaking, the first phase of $\mathcal{A}$ is an X-IND sampler if there is a set $\mathcal{X}$ of size at most X such that the circuits output by $\mathcal{A}$ are functionally equivalent outside $\mathcal{X}$ and furthermore within $\mathcal{X}$ the outputs of the two sampled circuits are indistinguishable. Formally, let $X(\cdot)$ be a function such that $X(\lambda) \leq 2^\lambda$ for all $\lambda \in \mathbb{N}$. We call $\mathcal{A}$ an X-IND *sampler* if there is a set $\mathcal{X}_\lambda$ of size at most $X(\lambda)$ such that the following two conditions holds: (1) for all (possibly unbounded) $\mathcal{D}$ the advantage function below is negligible

$$\mathbf{Adv}^{\text{eq\$}}_{\mathcal{A},\mathcal{D}}(\lambda) := \Pr\left[C_0(x;r) \neq C_1(x;r) \wedge x \notin \mathcal{X}_\lambda : (C_0, C_1, \text{st}) \leftarrow\!\!\$\ \mathcal{A}_1(1^\lambda); (x,r) \leftarrow\!\!\$\ \mathcal{D}(C_0, C_1, \text{st})\right]\,.$$

(2) For all non-uniform PPT distinguishers $\mathcal{D} := (\mathcal{D}_1, \mathcal{D}_2)$

$$X(\lambda) \cdot \mathbf{Adv}^{\text{sel-ind}}_{\mathcal{A},\mathcal{D}}(\lambda) := X(\lambda) \cdot \Pr\left[\text{Sel-IND}^{\mathcal{D}}_{\mathcal{A}}(1^\lambda)\right] \in \text{NEGL}\,,$$

where game $\text{Sel-IND}^{\mathcal{D}}_{\mathcal{A}}(1^\lambda)$ is shown in Figure 1 (right). This game has a static (or selective) flavor as $\mathcal{D}_1$ chooses a differing-input x *before* it gets to see the challenge circuit pair. We call an obfuscator meeting this level of security a *probabilistic indistinguishability obfuscator* [CLTV15] and use **PIO** instead of **Obf** to emphasize this.

$\underline{\text{IND-CPA}^{\mathcal{A}}_{\Pi}(\lambda)}$:	$\underline{\text{IND}^{\mathcal{A}}_{\mathbf{Obf}}(\lambda)}$:	$\underline{\text{Sel-IND}^{\mathcal{D}}_{\mathcal{A}}(\lambda)}$:
$(sk, pk) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)$	$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{A}_1(1^\lambda)$	$(x, z) \leftarrow\!\!\$\ \mathcal{D}_1(1^\lambda)$
$(m_1, m_1, st) \leftarrow\!\!\$\ \mathcal{A}_1(pk)$	$b \leftarrow\!\!\$\ \{0,1\}$	$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{A}_1(1^\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$\overline{C} \leftarrow\!\!\$\ \mathbf{Obf}(1^\lambda, C_b)$	$b \leftarrow\!\!\$\ \{0,1\}$; $r \leftarrow\!\!\$\ \{0,1\}^{r(\lambda)}$
$c \leftarrow\!\!\$\ \mathbf{Enc}(m, pk)$	$b' \leftarrow\!\!\$\ \mathcal{A}_2(\overline{C}, st)$	$y \leftarrow C_b(x; r)$
$b' \leftarrow\!\!\$\ \mathcal{A}_2(c, st)$	Return $(b = b')$	$b' \leftarrow\!\!\$\ \mathcal{D}_2(y, C_0, C_1, st, z)$
Return $(b = b')$		Return $(b = b')$

Figure 1: **Left**: IND-CPA security of a (homomorphic) PKE scheme. **Middle**: Indistinguishability security of an obfuscator. **Right**: Static-input (aka. selective) X-IND property of $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$.

2.4 Dual-mode NIZK proof systems

In our constructions we will be relying on special types of non-interactive zero-knowledge proof systems [GS08]. These systems have "dual-mode" common reference string (CRS) generation algorithms that produce indistinguishable CRSs in the "binding" and "hiding" modes. They also enjoy perfect completeness in both modes, are perfectly sound and extractable in the binding mode, and perfectly witness indistinguishable (WI) and zero-knowledge (ZK) in the hiding mode. The standard prototype for such schemes are pairing-based Groth–Sahai proofs [GS08], and using a generic NP reduction to the satisfiability of quadratic equations we can obtain a suitable proof system for any NP language.[4] We formalize the syntax and security of such proof systems next.

SYNTAX. A relation with setup is a pair of PPT algorithms $(\mathbf{S}, \mathbf{R})$ such that $\mathbf{S}(1^\lambda)$ outputs (gpk, gsk) and $\mathbf{R}(gpk, x, w)$ is a ternary relation and outputs a bit $b \in \{0,1\}$. A dual-mode non-interactive zero-knowledge (NIZK) proof system Σ for $(\mathbf{S}, \mathbf{R})$ consists of five algorithms as follows. (1) Algorithm $\mathbf{BCRS}(gpk, gsk)$ outputs a (binding) common reference string crs and an extraction trapdoor td_{ext}; (2) $\mathbf{HCRS}(gpk, gsk)$ outputs a (hiding) common reference string crs and a simulation trapdoor td_{zk}; (3) $\mathbf{Prove}(gpk, crs, x, w)$ on input crs, an instance x, and a witness w, outputs a proof π; (4) $\mathbf{Verify}(gpk, crs, x, \pi)$ on input a bit string crs, an instance x, and a proof π, outputs accept or reject; (5) $\mathbf{WExt}(td_{ext}, x, \pi)$ on input an extraction trapdoor, an instance x, and a proof π, outputs a witness w; and (6) $\mathbf{Sim}(td_{zk}, crs, x)$ on input the simulation trapdoor td_{zk}, the CRS crs, and an instance x, outputs a simulated proof π. We require a dual-mode NIZK to meet the following requirements.

CRS INDISTINGUISHABILITY. The common reference strings generated through $\mathbf{BCRS}(gpk, gsk)$ and $\mathbf{HCRS}(gpk, gsk)$ are computationally indistinguishable. We denote the distinguishing advantage of a PPT adversary $\mathcal{A}$ in the relevant security game by $\mathbf{Adv}^{crs}_{\Sigma,\mathcal{A}}(\lambda)$.

PERFECT COMPLETENESS UNDER **BCRS**/**HCRS**. For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow\!\!\$\ \mathbf{S}(1^\lambda)$, any $crs \leftarrow\!\!\$\ \mathbf{BCRS}(gpk, gsk)$, any (x, w) such that $\mathbf{R}(gpk, x, w) = 1$, and any $\pi \leftarrow\!\!\$\ \mathbf{Prove}(gpk, crs, x, w)$ we have that $\mathbf{Verify}(gpk, crs, x, \pi) = 1$. We require this property to also hold for any choice of $crs \leftarrow\!\!\$\ \mathbf{HCRS}(gpk, gsk)$.

[4] We note that extraction in Groth–Sahai proofs does not for all types of statements recover a witness. (Instead, for some types of statements, only g^{w_i} for a witness variable $w_i \in \mathbb{Z}_p$ can be recovered.) Here, however, we will only be interested in witnesses $w = (w_1, \ldots, w_n) \in \{0,1\}^n$ that are bit strings, in which case extraction always recovers w. (Specifically, extraction will recover g^{w_i} for all i, and thus all w_i.)

PERFECT SOUNDNESS UNDER **BCRS**. For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow\!\!\$\ \mathbf{S}(1^\lambda)$, any common reference string $crs \leftarrow\!\!\$\ \mathbf{BCRS}(gpk, gsk)$, any x for which for all $w \in \{0,1\}^*$, we have $\mathbf{R}(gpk, x, w) = 0$, and any $\pi \in \{0,1\}^*$ we have that $\mathbf{Verify}(gpk, crs, x, \pi) = 0$.

PERFECT EXTRACTABILITY UNDER **BCRS**. For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow\!\!\$\ \mathbf{S}(1^\lambda)$, any $(crs, td_{zk}) \leftarrow\!\!\$\ \mathbf{BCRS}(gpk, td_{ext})$, any (x, π) with $\mathbf{Verify}(gpk, crs, x, \pi) = 1$, and for $w \leftarrow\!\!\$\ \mathbf{WExt}(td_{ext}, x, \pi)$, we always have that $\mathbf{R}(gpk, x, w) = 1$.

PERFECT WI UNDER **HCRS**. For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow\!\!\$\ \mathbf{S}(1^\lambda)$, any $(crs, td_{zk}) \leftarrow\!\!\$\ \mathbf{HCRS}(gpk, gsk)$, any (x, w_b) such that $\mathbf{R}(gpk, x, w_b) = 1$ for $b \in \{0,1\}$, we have that $\pi_b \leftarrow\!\!\$\ \mathbf{Prove}(gpk, crs, x, w_b)$ for $b \in \{0,1\}$ are identically distributed.

PERFECT ZK UNDER **HCRS**. For any $\lambda \in \mathbb{N}$, any $(gpk, gsk) \leftarrow\!\!\$\ \mathbf{S}(1^\lambda)$, any $(crs, td_{zk}) \leftarrow\!\!\$\ \mathbf{HCRS}(gpk, gsk)$, any (x, w) such that $\mathbf{R}(gpk, x, w) = 1$, we have that $\pi_0 \leftarrow\!\!\$\ \mathbf{Prove}(gpk, crs, x, w)$ and $\pi_1 \leftarrow\!\!\$\ \mathbf{Sim}(td_{zk}, x)$ are identically distributed.

2.5 Hard membership problems

Finally, we will use languages with hard membership problems. More specifically, we say that a family $\mathcal{L} = \{\mathcal{L}_\lambda\}$ of families $\mathcal{L}_\lambda = \{L\}$ of languages $L \subseteq U$ in a universe $U = U_\lambda$ has a hard subset membership problem if the following holds. Namely, we require that no PPT algorithm can, given $L \leftarrow\!\!\$\ \mathcal{L}_\lambda$, efficiently distinguish between $x \leftarrow\!\!\$\ L$ and $x \leftarrow\!\!\$\ U$.

3 Multilinear Groups with Non-unique Encodings

Before presenting our constructions, we formally introduce what we mean by a multilinear group (MLG) scheme. Our abstraction differs from that of Garg, Gentry and Halevi [GGH13a] in that our treatment of MLG schemes is a direct adaptation of the "dream" MLG setting (called the "cryptographic" MLG setting in [BS03]) to a setting where group elements have *non-unique* encodings. In our abstraction, on top of the procedures needed for generating, manipulating and checking group elements, we introduce an *equality* procedure which generalizes the equality relation for groups with unique encodings.

SYNTAX. A multilinear group (MLG) scheme Γ consists of six PPT algorithms as follows.

Setup$(1^\lambda, 1^\kappa)$**:** This is the setup algorithm. On input the security parameter 1^λ and the multilinearity 1^κ, it outputs the group parameters pp. These parameters include *generators* $g_1, \ldots, g_{\kappa+1}$, *identity elements* $1_1, \ldots, 1_{\kappa+1}$, and integers $N_1, \ldots, N_{\kappa+1}$, which will represent group orders. (Generators, identity elements and group orders are discussed below.) We assume pp is provided to the various algorithms below.

Val$_i(h)$**:** This is the validity testing algorithm. On input (the group parameters), a group index $1 \le i \le \kappa+1$ and a string $h \in \{0,1\}^*$, it returns $b \in \{0,1\}$. We define $\mathbb{G}_i$, which is also parameterized by pp, as the set of all h for which $\mathbf{Val}_i(h) = 1$. We write $h \in \mathbb{G}_i$ when $\mathbf{Val}_i(h) = 1$ and refer to such strings as *group elements* (since we will soon impose a group structure on $\mathbb{G}_i$). We require that the bit strings in $\mathbb{G}_i$ have lengths that are polynomial in 1^λ and κ, a property that we refer to as *compactness*.

$\mathbf{Eq}_i(h_1, h_2)$: This is the equality algorithm. On input two valid group elements $h_1, h_2 \in \mathbb{G}_i$, it outputs a bit $b \in \{0, 1\}$. We require $\mathbf{Eq}_i$ to define an equivalence relation. We say that the group has unique encodings if $\mathbf{Eq}_i$ simply checks the equality of bit strings. We write $\mathbb{G}_i(h)$ for the set of all $h' \in \mathbb{G}_i$ such that $\mathbf{Eq}_i(h, h') = 1$; for any such h, h' in $\mathbb{G}_i$ we write $h = h'$; sometimes we write $h = h'$ *in* $\mathbb{G}_i$ for clarity. Since "=" refers to equality of bit strings as well as equivalence under $\mathbf{Eq}_i$ we will henceforth write "as bit strings" when we mean equality in that sense. We require $|\mathbb{G}_i/\mathbf{Eq}_i|$, the number of equivalence classes into which $\mathbf{Eq}_i$ partitions $\mathbb{G}_i$, to be finite and equal to N_i (where N_i comes from *pp*). Note that equality algorithms $\mathbf{Eq}_i$ for $1 \leq i \leq \kappa$ can be derived from one for $\mathbf{Eq}_{\kappa+1}$ using the multilinear map $\mathbf{e}$ defined below, provided $N_{\kappa+1}$ is prime. We assume throughout the paper that various algorithms below return $\perp$ when run on invalid group elements.

$\mathbf{Op}_i(h_1, h_2)$: This algorithm defines the group operation. On input two valid group elements $h_1, h_2 \in \mathbb{G}_i$ it outputs $h \in \mathbb{G}_i$. We write $h_1 h_2$ in place of $\mathbf{Op}_i(h_1, h_2)$ for simplicity. We require that $\mathbf{Op}_i$ respect the equivalence relations $\mathbf{Eq}_i$, meaning that if $h_1 = h_2$ in $\mathbb{G}_i$ and $h \in \mathbb{G}_i$, then $h_1 h = h_2 h$ in $\mathbb{G}_i$. We also demand that $h_1 h_2 = h_2 h_1$ in $\mathbb{G}_i$ (commutativity), for any third $h_3 \in \mathbb{G}_i$ we require $h_1(h_2 h_3) = (h_1 h_2)h_3$ in $\mathbb{G}_i$ (associativity) and $h_1 1_i = h_1$ in $\mathbb{G}_i$.

The algorithm $\mathbf{Op}$ gives rise to an exponentiation algorithm $\mathbf{Exp}_i(h, z)$ that on input $h \in \mathbb{G}_i$ and $z \in \mathbb{N}$ outputs an $h' \in \mathbb{G}_i$ such that $h' = h \cdots h$ in $\mathbb{G}_i$ with z occurrences of h. When no h is specified, we assume $h = g_i$. This algorithm runs in polynomial time in the length of z. We denote $\mathbf{Exp}_i(h, z)$ by h^z and define $h^0 := 1_i$. Note that under the definition of N_i for any $h \in \mathbb{G}_i$ we have that $\mathbf{Exp}_i(h, N_i) = 1_i$.[5] This in turn leads to an inversion algorithm $\mathbf{Inv}_i(h)$ that on input $h \in \mathbb{G}_i$ outputs $h^{N_i - 1}$. We insist that g_i in fact has order N_i, so that (the equivalence class containing) g_i generates $\mathbb{G}_i/\mathbf{Eq}_i$. We do not treat the case where the N_i are unknown but the formalism is easily extended to include it by adding an explicit inversion algorithm and by replacing N_i in *pp* with an approximation (which may be needed for sampling purposes). The above requirements ensure that $\mathbb{G}_i/\mathbf{Eq}_i$ acts as an Abelian group of order N_i with respect to the operation induced by $\mathbf{Op}_i$, with identity (the equivalence class containing) 1_i, and inverse operation $\mathbf{Inv}_i$.

We use the *bracket* notion [EHK+13] to denote an element $h = g_i^x$ in $\mathbb{G}_i$ with $[x]_i$. When using this notation, we will write the group law additively. This notation will be convenient in the construction and analysis of our MLG schemes. For example, $[z]_i + [z']_i$ succinctly denotes $\mathbf{Op}_i(\mathbf{Exp}(g_i, z), \mathbf{Exp}(g_i, z'))$. Note that when writing $[z]_i$ it is *not* necessarily the case that z is explicitly known.

$\mathbf{e}(h_1, \ldots, h_\kappa)$: This the multilinear map algorithm. For κ group elements $h_i \in \mathbb{G}_i$ as input, it outputs $h_{\kappa+1} \in \mathbb{G}_{\kappa+1}$. We demand that for any $1 \leq j \leq \kappa$ and any $h'_j \in \mathbb{G}_j$

$$\mathbf{e}(h_1, \ldots, h_j h'_j, \ldots, h_\kappa) = \mathbf{e}(h_1, \ldots, h_j, \ldots, h_\kappa)\mathbf{e}(h_1, \ldots, h'_j, \ldots, h_\kappa) \text{ in } \mathbb{G}_{\kappa+1} .$$

We also require the map to be *non-degenerate* in the sense that for some tuple of elements as input the multilinear map outputs an element of $\mathbb{G}_{\kappa+1}$ outside the equivalence class of $1_{\kappa+1}$. (This implies that $\mathbf{e}$ is surjective onto $\mathbb{G}_{\kappa+1}/\mathbf{Eq}_{\kappa+1}$ when N_i is prime, but need not imply surjectivity when $N_{\kappa+1}$ is composite.) We call an MLG scheme *symmetric* if the group algorithms

[5] However, note that N_i need not be the least integer with this property.

are independent of the group index for $1 \leq i \leq \kappa$ and $\mathbf{e}$ is invariant under permutations of its inputs. That is, for any permutation $\pi : [\kappa] \longrightarrow [\kappa]$ we have

$$\mathbf{e}(h_1, \ldots, h_\kappa) = \mathbf{e}(h_{\pi(1)}, \ldots, h_{\pi(\kappa)}) \text{ in } \mathbb{G}_{\kappa+1} .$$

We refer to all the other cases as being *asymmetric*. To distinguish the target group we frequently write $\mathbb{G}_T$ instead of $\mathbb{G}_{\kappa+1}$ (and similarly for 1_T and g_T in place of $1_{\kappa+1}$ and $g_{\kappa+1}$) as its structure in our construction will be different from that of the source groups $\mathbb{G}_1, \ldots, \mathbb{G}_\kappa$.

$\mathbf{Sam}_i(z)$: This is the sampling algorithm. On input $z \in \mathbb{N}$ it outputs $h \in \mathbb{G}_i$ whose distribution is "close" to that of uniform over the equivalence class $\mathbb{G}_i(g_i^z)$. Here "close" is formalized via computational, statistical or perfect indistinguishability. We also allow a special input ε to this algorithm, in which case the sampler is required to output a uniformly distributed $h \in \mathbb{G}_i$ together with a z such that $h \in \mathbb{G}_i(g_i^z)$. When outputting z is not required, we say that $\mathbf{Sam}_i(\varepsilon)$ is *discrete-logarithm oblivious*. Note that for groups with unique encodings these algorithms trivially exist. For notational convenience, for a known a we define $[a]_i$ to be an element sampled via $\mathbf{Sam}_i(a)$.

Some applications also rely on the following algorithm which provides a canonical bit string for the group elements within a single equivalence class.

$\mathbf{Ext}_i(h)$: This is the extraction algorithm. On input $h \in \mathbb{G}_i$ it outputs a string $s \in \{0,1\}^{\mathrm{poly}(\lambda)}$. We demand that for any $h_1, h_2 \in \mathbb{G}_i$ with $h_1 = h_2$ in $\mathbb{G}_i$ we have that $\mathbf{Ext}_i(h_1) = \mathbf{Ext}_i(h_2)$ (as bit strings). We also require that for $[z]_i \hookleftarrow\$ \mathbf{Sam}_i(\varepsilon)$, the distribution of $\mathbf{Ext}_i([z]_i)$ is uniform over $\{0,1\}^{\mathrm{poly}(\lambda)}$. For groups with unique encodings this algorithm trivially exists.

COMPARISON WITH GGH. Our formalization differs from that of [GGH13a] which defines a *graded encoding scheme*. The main difference is that a graded encoding scheme defines bilinear maps $\mathbf{e}_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \longrightarrow \mathbb{G}_{i+j}$. Using this algorithm, one can implement $\mathbf{Eq}_i$ for any $1 \leq i \leq \kappa$ from $\mathbf{Eq}_{\kappa+1}$ as follows (if $\mathbf{e}_{i,j}$ is injective). To check the equality of $h_1, h_2 \in \mathbb{G}_i$ call $\mathbf{e}_{i,\kappa+1-i}(h, g_{\kappa+1-i})$ for $h = h_1, h_2$ to map these elements to the target group and check equality there using $\mathbf{Eq}_{\kappa+1}$. Similarly, $\mathbf{Ext}_i(h)$ can be constructed from $\mathbf{Ext}_{\kappa+1}(h)$ and 1_j for all $\mathbb{G}_j$. (Note that for extraction we need a canonical *string* rather than a canonical group element.) Moreover, the abstraction and construction of graded encodings schemes in [GGH13a] do not provide any validity algorithms; these are useful in certain adversarial situations such as CCA security and signature verification. Further, all known candidate constructions of graded encoding schemes are noisy and only permit a limited number of operations. Finally, the known candidate graded encoding schemes do not permit sampling for specific values of z, but rather only permit sampling elements with a z that is only known up to its equivalence class.

SYNTACTIC EXTENSIONS. Although our syntax does not treat the cases of graded [GGH13a, CLT13], exponentially multilinear, or self-pairing [YYHK14] maps, it can be modified to capture these variants. We briefly outline the required modifications. For graded maps, we require the existence of a map that on input $h_i \in \mathbb{G}_i$ for indices $i = i_1, \ldots, i_\ell$ with $t := \sum_{i=1}^{\ell} i_j \leq \kappa$ outputs a group element in $\mathbb{G}_t$. This map is required to be multilinear in each component. For exponential (aka. unbounded) linearity, we provide the linearity κ in its binary representation to the **Setup** algorithm. We also include procedures for generator and identity element generation.[6] Proper

[6] It is also more natural to work with unbounded maps in the graded setting as otherwise we would have to provide an exponential number of inputs and hence assume "default" values rather than being able to include them all in *pp*.

self-pairing maps correspond to a setting where the group algorithms are independent of the group index for $1 \le i \le \kappa + 1$ (including the target index $\kappa + 1$), and the group generators and identity elements are all identical. Observe that a proper self-pairing would induce a graded encoding scheme of unbounded linearity; recall from the introduction that the scheme of Yamakawa et al. [YYHK14] does not meet this definition because of the growth in the size of its auxiliary information.

4 The Construction

We now present our construction of an MLG scheme Γ according to the syntax introduced in Section 3. In the later sections we will consider special cases of the construction and prove the hardness of analogues of the multilinear DDH problem under various assumptions.

We rely on the following building blocks in our MLG scheme. (1) A cyclic group $\mathbb{G}_0$ of some order N_0 with generator g_0 and identity 1_0; formally we think of this as a 1-linear MLG scheme Γ_0 with unique encodings in which $\mathbf{e}$ is trivial; the algorithm $\mathbf{Val}_0$ implies that elements of $\mathbb{G}_0$ are efficiently recognizable. (2) A general-purpose obfuscator **Obf**. (3) An additively homomorphic public-key encryption scheme $\Pi := (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Eval})$ with plaintext space $\mathbb{Z}_N$ (alternatively, a perfectly correct HPKE scheme). (4) A dual-mode NIZK proof system. (5) A family $\mathcal{TD}$ of (families of) languages TD which has a hard subset membership problem, and such that all TD have efficiently computable witness relations with unique witnesses.[7] (See Section 2 for more formal definitions.)

We reserve variables and algorithms with index 0 for the base scheme Γ_0; we also write $N = N_0$. We require that the algorithms of Γ_0 except for $\mathbf{Setup}_0$ and $\mathbf{Sam}_0$ are deterministic. We will also use the bracket notation to denote the group elements in $\mathbb{G}_0$. For example, we write $[z]_0, [z']_0 \in \mathbb{G}_0$ for two valid elements of the base group and $[z]_0 + [z']_0 \in \mathbb{G}_0$ for $\mathbf{Op}_0([z]_0, [z']_0)$. Variables with nonzero indices correspond to various source and target groups. Given all of the above components, our MLG scheme Γ consists of algorithms as detailed in the sections that follow.

4.1 Setup

The setup algorithm for Γ samples parameters $pp_0 \leftarrow\!\!\$\ \mathbf{Setup}_0(1^\lambda)$ for the base MLG scheme, generates two encryption key pairs $(pk_j, sk_j) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)$ $(j = 1, 2)$, and a matrix $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\kappa)^t \in \mathbb{Z}_N^{\kappa \times \ell}$ where κ is the linearity and $\ell \in \{2, 3\}$ is a parameter of our construction. It sets

$$gpk := (pp_0, pk_1, pk_2, [\mathbf{W}]_0, \mathsf{TD}, y) ,$$

where $[\mathbf{W}]_0$ denotes a matrix of $\mathbb{G}_0$ elements that entry-wise is written in the bracket notation, $\mathsf{TD} \leftarrow\!\!\$\ \mathcal{TD}$, and y is *not* in TD. In our MLG scheme we set $N_1 = \cdots = N_{\kappa+1} := N$, where N is the group order implicit in pp_0. The setup algorithm then generates a common reference string $crs = (crs', y)$ where $crs' \leftarrow\!\!\$\ \mathbf{BCRS}(gpk, gsk)$ for a relation $(\mathbf{S}, \mathbf{R})$ that will be defined in Section 4.2. It also constructs two obfuscated circuits $\overline{C}_{\mathrm{Map}}$ and $\overline{C}_{\mathrm{Add}}$ which we will describe in Sections 4.3 and 4.4. For $1 \le i \le \kappa$, the identity elements 1_i and group generators g_i are sampled using

[7] An example of such a language is the Diffie–Hellman language $\mathsf{TD} = \{(g_1^r, g_2^r) \mid r \in \mathbb{N}\}$ in a DDH group with generators g_1, g_2. In particular, a suitable trapdoor language imposes no additional computational assumption in our upcoming security proof.

$\mathbf{Sam}_i(0)$ and $\mathbf{Sam}_i(x_i)$ respectively for algorithm $\mathbf{Sam}_i$ described in Section 4.5 with $x_i \in [N]$ that is co-prime to N. We emphasize that this approach is well defined since the operation of $\mathbf{Sam}_i$ is defined independently of the generators and the identity elements and depends only on *gpk* and *crs*. We set $1_{\kappa+1} = 1_0$ and $g_{\kappa+1} = g_0$. The scheme parameters are

$$pp := (gpk, crs, \overline{C}_{\mathrm{Map}}, \overline{C}_{\mathrm{Add}}, g_1, \ldots, g_{\kappa+1}, 1_1, \ldots, 1_{\kappa+1}) .$$

We note that this algorithm runs in polynomial time in λ as long as κ is polynomial in λ.

4.2 Validity and equality

The elements of $\mathbb{G}_i$ for $1 \le i \le \kappa$ are tuples of the form $h = ([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi)$ where $\mathbf{c}_1, \mathbf{c}_2$ are encryptions of vectors from $\mathbb{Z}_N^\ell$ under , pk_1, pk_2, respectively (encryption algorithm **Enc** extends from plaintext space $\mathbb{Z}_N$ to $\mathbb{Z}_N^\ell$ in the obvious way) and where π is a NIZK to be defined below. We refer to $(\mathbf{c}_1, \mathbf{c}_2, \pi)$ as the *auxiliary information* for $[z]_0$. The elements of $\mathbb{G}_{\kappa+1}$ are just those of $\mathbb{G}_0$.

The NIZK proof system that we use corresponds to the following inclusive disjunctive relation $(\mathbf{S}, \mathbf{R} := \mathbf{R}_1 \vee \mathbf{R}_2)$. Algorithm $\mathbf{S}(1^\lambda)$ outputs $gpk = (pp_0, pk_1, pk_2, [\mathbf{W}]_0, \mathsf{TD})$ as defined above and sets $gsk = (sk_1, sk_2)$. Relation $\mathbf{R}_1$ on input *gpk*, tuple $([z]_0, \mathbf{c}_1, \mathbf{c}_2)$, and witness $(\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2, sk_1, sk_2)$ accepts iff $[z]_0 \in \mathbb{G}_0$, the *representations* of $[z]_0$ as $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_N^\ell$ are valid with respect to $[\mathbf{W}]_0$ in the sense that

$$[z]_0 = [\langle \mathbf{x}, \boldsymbol{\omega}_i \rangle]_0 \wedge [z]_0 = [\langle \mathbf{y}, \boldsymbol{\omega}_i \rangle]_0 ,$$

(where $\langle \cdot, \cdot \rangle$ denotes inner product) and the following ciphertext validity condition (with respect to the inputs to the relation) is met:

$$\begin{gathered}
(\mathbf{c}_1 = \mathbf{Enc}(\mathbf{x}, pk_1; \mathbf{r}_1) \wedge \mathbf{c}_2 = \mathbf{Enc}(\mathbf{x}, pk_2; \mathbf{r}_2)) \\
\vee \\
(pk_1, sk_1) = \mathbf{Gen}(sk_1) \wedge (pk_2, sk_2) = \mathbf{Gen}(sk_2) \\
\wedge \mathbf{x} = \mathbf{Dec}(\mathbf{c}_1, sk_1) \wedge \mathbf{y} = \mathbf{Dec}(\mathbf{c}_2, sk_2))
\end{gathered}$$

Recall that we have assumed the secret key of the encryption scheme to be the random coins used in **Gen**. Note that the representation validity check can be efficiently performed "in the exponent" using $[\mathbf{W}]_0$ and the explicit knowledge of $\mathbf{x}$ and $\mathbf{y}$. Note also that for honestly generated keys and ciphertexts the two checks in the expression above are equivalent (although this not generally the case when ciphertexts are malformed).

Relation $\mathbf{R}_2$ depends on the language TD, and on input *gpk*, tuple $([z]_0, \mathbf{c}_1, \mathbf{c}_2)$, and witness w_y accepts iff w_y is a valid witness to $y \in \mathsf{TD}$. (Note that $\mathbf{R}_2$ completely ignores $([z]_0, \mathbf{c}_1, \mathbf{c}_2)$.)

For $1 \le i \le \kappa$, the $\mathbf{Val}_i$ algorithm for Γ, on input $([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi)$, first checks that the first component is in $\mathbb{G}_0$ using $\mathbf{Val}_0$ and then checks the proof π; if both tests pass, it then returns $\top$, else $\bot$. Observe that for an honest choice of $crs = (crs', y)$, the perfect completeness and the perfect soundness of the proof system ensure that only those elements which pass relation $\mathbf{R}_1$ are accepted. Algorithm $\mathbf{Val}_{\kappa+1}$ just uses $\mathbf{Val}_0$.

The equality algorithm $\mathbf{Eq}_i$ of Γ for $1 \le i \le \kappa$ first checks the validity of the two group elements passed to it and then returns true iff their first components match, according to $\mathbf{Eq}_0$, the equality algorithm from the base scheme Γ_0. Algorithm $\mathbf{Eq}_{\kappa+1}$ just uses $\mathbf{Eq}_0$. The correctness of this algorithm follows from the perfect completeness of Σ.

4.3 Group operations

We provide a procedure that, given as inputs $h = ([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi)$ and $h' = ([z']_0, \mathbf{c}'_1, \mathbf{c}'_2, \pi') \in \mathbb{G}_i$, generates a tuple representing the product $h \cdot h'$. This, in particular, will enable our multilinear map to be run on the additions of group elements whose explicit representations are not necessarily known. We exploit the structure of the base group as well as the homomorphic properties of the encryption scheme to "add together" the first three components. We then use (sk_1, sk_2) as a witness to generate a proof π'' that the new tuple is well formed. (For technical reasons we check the validity of h and h' in two different ways: using proofs π, π', and also explicitly using (sk_1, sk_2). Note that, although useful in the analysis, the explicit check is redundant by the perfect soundness of the proof system under a binding crs'.)

In pp we include an obfuscation of the C_{Add} circuit shown in Figure 2 (top), and again we emphasize that steps 5a or 5b are never reached with a binding crs' (but they may be reached with a hiding crs' later in the analysis). Either an **IO** or a **PIO** will be used to obfuscate this circuit. Note that although we have assumed the evaluation algorithm to be deterministic, algorithm **Prove** is randomized and we need to address how we deal with its coins. When using PIO to obfuscate $\overline{C}_{Add}$, the obfuscator directly deals with the needed randomness.[8] When using IO, a random (but fixed) set of coins will be hardwired into the circuit and hence the same set of coins will be used for all inputs. (As we shall see, when using IO the proof system has to satisfy extra structural requirements; these ensure that using the same coins throughout does not compromise security.) The $\mathbf{Op}_i$ algorithm for $1 \leq i \leq \kappa$ runs the obfuscated circuit on i, the input group elements. Algorithm $\mathbf{Op}_{\kappa+1}$ just uses $\mathbf{Op}_0$ as usual. The correctness of this algorithm follows from those of Γ_0 and Π, the completeness of Σ and the correctness, in our sense of, (the possibly probabilistic) obfuscator **Obf**; see Section 2 for the definitions.

4.4 The multilinear map

The multilinear map for Γ, on input κ group elements $h_i = [z_i]_i = ([z_i]_0, \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i)$, uses sk_1 to recover the representation $\mathbf{x}_i$. It then uses the explicit knowledge of the matrix $\mathbf{W}$ to compute the output of the map as

$$\mathbf{e}([z_1]_1, \ldots, [z_\kappa]_\kappa) := \left[\prod_{i=1}^{k} \langle \mathbf{x}_i, \boldsymbol{\omega}_i \rangle \right]_{\kappa+1} .$$

Recalling that $\mathbb{G}_{\kappa+1}$ is nothing other than $\mathbb{G}_0$, and $g_{\kappa+1} = g_0$, the output of the map is just the $\mathbb{G}_0$-element $(g_0)^{\prod_{i=1}^{k} \langle \mathbf{x}_i, \boldsymbol{\omega}_i \rangle}$. The product in the exponent can be efficiently computed over $\mathbb{Z}_N$ for *any* polynomial level of linearity κ and any ℓ as it uses $\mathbf{x}_i$ and $\boldsymbol{\omega}_i$ explicitly. The multilinearity of the map follows from the linearity of each of the multiplicands in the above product (and the completeness of Σ, the correctness of Π, and the correctness of the (possibly probabilistic) obfuscator **Obf**). An obfuscation $\overline{C}_{Map}$ of the circuit implementing this operation (see Figure 2, bottom) will be made available through the public parameters and $\mathbf{e}$ is defined to run this circuit on its inputs.

[8] Typically, the obfuscated circuit will have a PRF key hardwired in and derives the required randomness by applying the PRF to the circuit inputs.

CIRCUIT $C_{Add}[gpk, crs, sk_1, sk_2, td_{ext}; r](i, h, h')$:

1. if $\neg\mathbf{Val}_i(h) \vee \neg\mathbf{Val}_i(h')$ return $\perp$
2. parse $([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi) \leftarrow h$ and $([z']_0, \mathbf{c}'_1, \mathbf{c}'_2, \pi') \leftarrow h'$
3. $[z'']_0 \leftarrow [z]_0 + [z']_0$; $\mathbf{c}''_1 \leftarrow \mathbf{c}_1 + \mathbf{c}'_1$; $\mathbf{c}''_2 \leftarrow \mathbf{c}_2 + \mathbf{c}'_2$
4. // explicit validity check of h, h'
   - 4.1 $\mathbf{x} \leftarrow \mathbf{Dec}(\mathbf{c}_1, sk_1)$, $\mathbf{y} \leftarrow \mathbf{Dec}(\mathbf{c}_2, sk_2)$
     $\mathbf{x}' \leftarrow \mathbf{Dec}(\mathbf{c}'_1, sk_1)$, $\mathbf{y}' \leftarrow \mathbf{Dec}(\mathbf{c}'_2, sk_2)$
   - 4.2a if $([z]_0 \neq [\langle \mathbf{x}, \boldsymbol{\omega}_i \rangle]_0) \vee ([z]_0 \neq [\langle \mathbf{y}, \boldsymbol{\omega}_i \rangle]_0)$ goto 5a
   - 4.2b else if $([z']_0 \neq [\langle \mathbf{x}', \boldsymbol{\omega}_i \rangle]_0) \vee ([z']_0 \neq [\langle \mathbf{y}', \boldsymbol{\omega}_i \rangle]_0)$ goto 5b
   - 4.2c else goto 5c // h, h' are valid

5a. // h is invalid

- 5a.1 $w'_y \leftarrow_\$ \mathbf{WExt}(td_{ext}, ([z]_0, \mathbf{c}_1, \mathbf{c}_2), \pi)$
- 5a.2 if $\neg\mathbf{R}_2(gpk, (([z]_0, \mathbf{c}_1, \mathbf{c}_2)), w'_y)$ return $\perp$
- 5a.3 $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}''_1, \mathbf{c}''_2), w'_y; r)$

5b. repeat 5a with h' // only h' is invalid

5c. $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}''_1, \mathbf{c}''_2), (sk_1, sk_2); r)$

6. return $([z''], \mathbf{c}''_1, \mathbf{c}''_2, \pi'')$

CIRCUIT $C_{Map}[gpk, crs, \mathbf{W}, sk_1](h_1, \ldots, h_\kappa)$:

1. for $i = 1 \ldots \kappa$
   - 1.1 if $\neg\mathbf{Val}_i(h_i)$ return $\perp$
   - 1.2 $([z_i]_0, \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i) \leftarrow h_i$
   - 1.3 $\mathbf{x}_i \leftarrow \mathbf{Dec}(\mathbf{c}_{i,1}, sk_1)$
2. $z_{\kappa+1} \leftarrow \prod_{i=1}^{k} \langle \mathbf{x}_i, \boldsymbol{\omega}_i \rangle \pmod N$
3. return $[z_{\kappa+1}]_{\kappa+1}$

Figure 2: **Top**: Circuit for addition of group elements. Explicit randomness r is used with an **IO** and is internally generated when using a **PIO**. **Bottom**: Circuit implementing the multilinear map. Recall that here $gpk = (pp_0, pk_1, pk_2, [\mathbf{W}]_0, \mathsf{TD}, y)$.

4.5 Sampling and extraction

Given vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{Z}_N^\ell$ satisfying $\langle \mathbf{x}, \boldsymbol{\omega}_i \rangle = \langle \mathbf{y}, \boldsymbol{\omega}_i \rangle$, we set $[z]_0 := [\langle \mathbf{y}, \boldsymbol{\omega}_i \rangle]_0$ (which can be computed using $[\mathbf{W}]_0$ and explicit knowledge of $\mathbf{x}$) and

$$[z]_i \leftarrow \big([z]_0, \mathbf{c}_1 = \mathbf{Enc}(\mathbf{x}, pk_1; \mathbf{r}_1), \mathbf{c}_2 = \mathbf{Enc}(\mathbf{y}, pk_2; \mathbf{r}_2), \\ \pi = \mathbf{Prove}(gpk, crs, ([z]_i, \mathbf{c}_1, \mathbf{c}_2), (\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2))\big) .$$

If $\mathbf{W}$ is explicitly known the vectors $\mathbf{x}$ and $\mathbf{y}$ can take arbitrary forms subject to validity. This matrix, however, is only implicitly known, and in our sampling procedure we set $\mathbf{x} = \mathbf{y} = (z, 0)$ when $\ell = 2$ and $\mathbf{x} = \mathbf{y} = (z, 0, 0)$ when $\ell = 3$. (We call these the canonical representations.) Note that the outputs of the sampler are *not* statistically uniform within $\mathbb{G}_i([z]_i)$. Despite this, under the IND-CPA security of the encryption scheme it can be shown that the outputs are computationally close to uniform.

Since the target group has unique encodings, as noted in Section 3, an extraction algorithm for all groups can be derived from one for the target group. The latter can be implemented by applying a universal hash function to the group elements in $\mathbb{G}_T$, for example.

$\kappa\text{-Switch}^{\mathcal{A}}_{\Gamma}(\lambda)$:

$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^\lambda, 1^\kappa)$
$((\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_1), i, st) \leftarrow\!\!\$\ \mathcal{A}_1(pp, \mathbf{W})$
$b \leftarrow\!\!\$ \{0,1\}; \mathbf{r}_1, \mathbf{r}_2 \leftarrow\!\!\$ (\{0,1\}^{r(\lambda)})^{|\mathbf{x}_0|}$
$\mathbf{c}_1 \leftarrow \mathbf{Enc}(\mathbf{x}_b, pk_1; \mathbf{r}_1);\ \mathbf{c}_2 \leftarrow \mathbf{Enc}(\mathbf{y}_b, pk_2; \mathbf{r}_2)$
$\pi \leftarrow\!\!\$\ \mathbf{Prove}(gpk, crs, ([z]_0, \mathbf{c}_1, \mathbf{c}_2), (\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2, \perp, \perp))$
$b' \leftarrow\!\!\$\ \mathcal{A}_2\big(([\langle \mathbf{x}_b, \boldsymbol{\omega}_i \rangle]_0, \mathbf{c}_1, \mathbf{c}_2, \pi), st\big)$
Return $(b = b')$

Figure 3: Game formalizing the indistinguishability of encodings with an equivalence class. This game is specific to our construction Γ. An adversary is legitimate if $z = \langle \mathbf{x}_b, \boldsymbol{\omega}_i \rangle = \langle \mathbf{y}_b, \boldsymbol{\omega}_i \rangle$ for $b \in \{0,1\}$. We note that $\mathcal{A}$ gets explicit access to matrix $\mathbf{W}$ generated during setup.

5 Indistinguishability of Encodings

In this section we will prove two theorems that are essential tools in establishing the intractability of the κ-MDDH for our MLG scheme Γ constructed in Section 4. These theorems, roughly speaking, state that valid encodings of elements within a single equivalence class are computationally indistinguishable. We formalize this property via the κ-Switch game shown in Figure 3. This game lets an adversary $\mathcal{A}$ choose an element $[z]_i \in \mathbb{G}_i$ by producing two valid representations $(\mathbf{x}_0, \mathbf{y}_0)$ and $(\mathbf{x}_1, \mathbf{y}_1)$ for it. The adversary is given an encoding of $[z]_i$ generated using $(\mathbf{x}_b, \mathbf{y}_b)$ for a random b, and has to guess the bit b. In this game, besides access to pp, which contains the obfuscated circuits for the group operation and the multilinear map, we also provide the matrix $\mathbf{W}$ in the clear to the adversary. This strengthens the κ-Switch game and is needed for our later analysis.

To prove that the advantage of $\mathcal{A}$ in the κ-Switch game is negligible we rely on the security of the obfuscator, the IND-CPA security of the encryption scheme, and the security of the NIZK proof system. Depending on the type of the obfuscator and proof system used, we show indistinguishability of encodings in two incomparable ways: (1) using a *probabilistic* obfuscator that is secure against X-IND adversaries and a dual-mode NIZK as defined in Section 2.4; and (2) using a (standard) indistinguishability obfuscator for deterministic circuits and a dual-mode NIZK that is required to satisfy a "witness-translation" property that we formalize in Section 5.2.

5.1 Using probabilistic indistinguishability obfuscation

The indistinguishability of encodings using the first set of assumptions above is conceptually simpler to prove and we start with this case. Intuitively, the IND-CPA security of the encryption scheme will ensure that the encryptions of the two representations are indistinguishable. This argument, however, does not immediately work as the parameters pp contain component $\overline{\mathsf{C}}_{\mathrm{Add}}$ that depends on *both* decryption keys. We deal with this by finding an alternative implementation of this circuit without the knowledge of the secret keys, in the presence of a slightly different public parameters (which are computationally indistinguishable to those described in Section 4). The next lemma, roughly speaking, says that *provided* parameters pp include an instance $y \in \mathsf{TD}$, then there exists an alternative implementation $\widehat{\mathsf{C}}_{\mathrm{Add}}$ that does not use the secret keys, and whose obfuscation is indistinguishable to that of $\mathsf{C}_{\mathrm{Add}}$ of Figure 2 (top) for an adversary that *knows* the secret keys. It relies on the security of the obfuscator and the security of the NIZK proof system.

Lemma 5.1 ($\mathsf{C}_{\mathrm{Add}}$ without decryption keys)**.** *Let* **PIO** *be a secure obfuscator for* X-IND *samplers, and* Σ *be a dual-mode NIZK proof system. Additionally, let parameters* $\widetilde{pp}$ *sampled as in Section 4 but with* $\widetilde{y} \in \mathsf{TD}$*, and let* $\widehat{pp}$ *sampled as* $\widetilde{pp}$ *but with a hiding CRS* $\widehat{crs}'$*, and an obfuscation of circuit* $\widehat{\mathsf{C}}_{\mathrm{Add}}$ *of Fig. 4 (bottom). Then, for any* PPT *adversary* $\mathcal{A}$ *there are* ppt *adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *of essentially the same complexity as* $\mathcal{A}$ *such that for all* $\lambda \in \mathbb{N}$

$$\begin{aligned}&\Pr[\mathcal{A}(\widetilde{pp}, sk_1, sk_2) = 1 \ : \ (sk_1, sk_2) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)] - \Pr[\mathcal{A}(\widehat{pp}, sk_1, sk_2) = 1 \ : \ (sk_1, sk_2) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)] \\ &\leq 2 \cdot \mathbf{Adv}^{\mathrm{ind}}_{\mathbf{PIO},\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\mathrm{crs}}_{\Sigma,\mathcal{B}_2}(\lambda).\end{aligned}$$

Proof. The crucial observation is that a witness w_y to $\widetilde{y} \in \mathsf{TD}$ is also a witness to $x \in \mathbf{R}$, and therefore $\widehat{\mathsf{C}}_{\mathrm{Add}}$ can use w_y instead of sk_1, sk_2 to produce the output proof π''. Below we provide descriptions of the transformation from $\mathsf{C}_{\mathrm{Add}}$ to $\widehat{\mathsf{C}}_{\mathrm{Add}}$, and let W_i denote the event that $\mathcal{A}$ in Game_i outputs 1.

Game_0 : We start with (a PIO obfuscation of) circuit $\mathsf{C}_{\mathrm{Add}}$ of Fig. 2 and with $\widetilde{pp}$ including $\widetilde{y} \in \mathsf{TD}$ and a binding crs'.

Game_1 : The circuit has witness w_y to $\widetilde{y} \in \mathsf{TD}$ hard-coded. If some input reaches the "invalid" branches, $\mathsf{C}_{\mathrm{Add}}$ does not extract a witness from the corresponding proof, but instead uses w_y to generate proof π'' (see Fig. 4 (top)). Note that Game_1 requires no extraction trapdoor td_{ext} anymore.

We claim that $|\Pr[\mathsf{W}_0(\lambda)] - \Pr[\mathsf{W}_1(\lambda)]| \leq \mathbf{Adv}^{\mathrm{ind}}_{\mathbf{PIO},\mathcal{B}_1}(\lambda)$.

By construction, the only difference between the games is that in Game_1 proof π'', with respect to invalid (input) encodings, is generated using hard-coded witness w_y to $y \in \mathsf{TD}$. Since w_y is unique, and the CRS crs' guarantees perfect soundness, this leads to *identical* behavior of $\mathsf{C}_{\mathrm{Add}}$ in Game 0. Hence this hop is justified by PIO.

Game_2 : The CRS $\widehat{crs}'$ included in the public parameters is now hiding (such that the generated proofs are perfectly witness-indistinguishable). We have that

$$|\Pr[\mathsf{W}_1(\lambda)] - \Pr[\mathsf{W}_2(\lambda)]| \leq \mathbf{Adv}^{\mathrm{crs}}_{\Sigma,\mathcal{B}_2}(\lambda),$$

where $\mathcal{B}_2$ is a PPT algorithm against the indistinguishability of binding and hiding CRS's.

Game_3 : Here, output proofs π'' for those inputs entering the "valid" branch (step 5b of Fig. 4 (top)) use w_y (and not sk_1, sk_2) as witness. In particular, this game does not need to perform a explicit validity check (using sk_1, sk_2) anymore, and therefore the addition circuit can be described as in Fig. 4 (bottom).

We claim that $|\Pr[\mathsf{W}_2(\lambda)] - \Pr[\mathsf{W}_3(\lambda)]| \leq \mathbf{Adv}^{\mathrm{ind}}_{\mathbf{PIO},\mathcal{B}_1}(\lambda)$.

By constrution the only difference between both games is that, the public parameters in Game_2 contain a PIO obfuscation of $\mathsf{C}_{\mathrm{Add}}$, and in Game_3 contain a PIO obfuscation of $\widehat{\mathsf{C}}_{\mathrm{Add}}$ of Fig. 4. In Lemma 5.2 we prove that these circuit variants are given by an X-IND sampler, and therefore their PIO obfuscations are indistinguishable.

□

CIRCUIT $\mathsf{C}_{\mathrm{Add}}[gpk, crs, sk_1, sk_2, w_y; r](i, h, h')$:

1. if $\neg\mathbf{Val}_i(h) \vee \neg\mathbf{Val}_i(h')$ return $\perp$
2. parse $([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi) \leftarrow h$ and $([z']_0, \mathbf{c}_1', \mathbf{c}_2', \pi') \leftarrow h'$
3. $[z'']_0 \leftarrow [z]_0 + [z']_0$; $\mathbf{c}_1'' \leftarrow \mathbf{c}_1 + \mathbf{c}_1'$; $\mathbf{c}_2'' \leftarrow \mathbf{c}_2 + \mathbf{c}_2'$
4. // explicit validity check of h, h'
   - 4.1 $\mathbf{x} \leftarrow \mathbf{Dec}(\mathbf{c}_1, sk_1)$, $\mathbf{y} \leftarrow \mathbf{Dec}(\mathbf{c}_2, sk_2)$
     $\mathbf{x}' \leftarrow \mathbf{Dec}(\mathbf{c}_1', sk_1)$, $\mathbf{y}' \leftarrow \mathbf{Dec}(\mathbf{c}_2', sk_2)$
   - 4.2a if $([z]_0 \neq [\langle \mathbf{x}, \boldsymbol{\omega}_i \rangle]_0) \vee ([z]_0 \neq [\langle \mathbf{y}, \boldsymbol{\omega}_i \rangle]_0)$ or $([z']_0 \neq [\langle \mathbf{x}', \boldsymbol{\omega}_i \rangle]_0) \vee ([z']_0 \neq [\langle \mathbf{y}', \boldsymbol{\omega}_i \rangle]_0)$ goto 5a
   - 4.2c else goto 5b

5a. // h or h' invalid

- 5a.1 $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}_1'', \mathbf{c}_2''), w_y; r)$

5b. // h and h' valid

- 5b.1 $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}_1'', \mathbf{c}_2''), (sk_1, sk_2); r)$

6. return $([z''], \mathbf{c}_1'', \mathbf{c}_2'', \pi'')$

CIRCUIT $\widehat{\mathsf{C}}_{\mathrm{Add}}[gpk, crs, w_y; r](i, h, h')$:

1. if $\neg\mathbf{Val}_i(h) \vee \neg\mathbf{Val}_i(h')$ return $\perp$
2. parse $([z]_0, \mathbf{c}_1, \mathbf{c}_2, \pi) \leftarrow h$, and $([z']_0, \mathbf{c}_1', \mathbf{c}_2', \pi') \leftarrow h'$
3. $[z'']_0 \leftarrow [z]_0 + [z']_0$; $\mathbf{c}_1'' \leftarrow \mathbf{c}_1 + \mathbf{c}_1'$; $\mathbf{c}_2'' \leftarrow \mathbf{c}_2 + \mathbf{c}_2'$
4. $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}_1'', \mathbf{c}_2''), w_y; r)$
5. return $([z''], \mathbf{c}_1'', \mathbf{c}_2'', \pi'')$

Figure 4: Circuits for addition of group elements used in Lemma 5.1. $\widehat{pp}$ includes $gpk = (pp_0, pk_1, pk_2, [\mathbf{W}]_0, \mathsf{TD}, \widetilde{y})$ where $\widetilde{y} \in \mathsf{TD}$ (also includes a hiding CRS $\widehat{crs}'$). Both circuits also have hard-coded (the) witness w_y to $\widetilde{y} \in \mathsf{TD}$. **Top:** sk_1, sk_2 are used to produce π'' on valid inputs. **Bottom:** w_y is always used to produce π''.

Lemma 5.2 (X-IND sampling)**.** *Let Σ be a dual-mode NIZK proof system for the relation $(\mathbf{S}, \mathbf{R})$ defined in Section 4.2. Suppose Σ is perfectly witness-indistinguishable under a hiding CRS. Let $\mathcal{A}_1$ be a sampler which outputs circuits $(\mathsf{C}_{\mathrm{Add}}, \widehat{\mathsf{C}}_{\mathrm{Add}})$ of Fig. 4. (Both circuits have the system parameters hard-coded in.) Then any $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ for a* PPT *$\mathcal{A}_2$ is* X-IND *for (the optimal) X, the size of the domain of the circuits. More precisely, for any (possibly unbounded) distinguisher $\mathcal{D}'$ and for any* PPT *distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ and any $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}^{\mathrm{eq\$}}_{\mathcal{A},\mathcal{D}'}(\lambda) = 0 \quad \textit{and} \quad \mathbf{Adv}^{\text{sel-ind}}_{\mathcal{A},\mathcal{D}}(\lambda) = 0\,.$$

Proof. The first equality is immediate as $\mathcal{X}$ is set to be the entire domain of the circuits. The second equality follows from the perfect witness-indistinguishability property of the proof system. Indeed, the only difference between the two circuits is that, for those inputs that are valid encodings, $\mathsf{C}_{\mathrm{Add}}$ uses decryption keys sk_1,sk_2 as witness to generate the output proof $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}_1'', \mathbf{c}_2''), (sk_1, sk_2); r)$, and $\widehat{\mathsf{C}}_{\mathrm{Add}}$ uses witness w_y to $y \in \mathsf{TD}$ (with y in the public parameters) to generate the proof $\widehat{\pi}'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}_1'', \mathbf{c}_2''), w_y; r)$. The WI property with a hiding $\widehat{crs}$ guarantees that π'' and $\widehat{\pi}''$ are *identically* distributed, and hence so are the outputs of $\mathsf{C}_{\mathrm{Add}}$ and $\widehat{\mathsf{C}}_{\mathrm{Add}}$. Note that no random coins are hardwired into these circuits—we are in the PIO setting—and fresh coins are used to compute the circuits' outputs. □

With Lemma 5.1 we can invoke IND-CPA security, and via a sequence of games obtain the result stated below. The proof can be found in Appendix A.1; we will give a high-level overview

of the proof below (see also Fig. 5).

Theorem 5.3 (Switching encodings using PIO)**.** *Let Γ be the MLG scheme constructed in Section 4, where* **PIO** *is secure for* X-IND *samplers, Π is an* IND-CPA*-secure encryption scheme, and Σ is a dual-mode NIZK proof system. Then, encodings of equivalent group elements are indistinguishable. More precisely, for any* PPT *adversary $\mathcal{A}$ and all $\lambda \in \mathbb{N}$, there are* ppt *adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ and $\mathcal{B}_4$ of essentially the same complexity as $\mathcal{A}$ such that for all $\lambda \in \mathbb{N}$*

$$\mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{A}}(\lambda) \leq 3 \cdot \mathbf{Adv}^{\mathrm{sm}}_{\mathsf{TD},\mathcal{B}_1} + 7 \cdot \mathbf{Adv}^{\mathrm{ind}}_{\mathbf{PIO},\mathcal{B}_2}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{crs}}_{\Sigma,\mathcal{B}_3}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\Pi,\mathcal{B}_4}(\lambda) \ .$$

Furthermore $\mathcal{B}_2$ is an X-IND *sampler for any function* $X(\lambda)$*.*

Proof sketch. The proof of this theorem proceeds via a sequence of 9 games as follows.

Game_0 : This is the κ-Switch game. The public parameters pp contain a no-instance $y \notin \mathsf{TD}$, a binding crs', $\mathsf{C}_{\mathrm{Add}}$ is constructed using (sk_1, sk_2) and $\mathsf{C}_{\mathrm{Map}}$ using sk_1 (see Fig. 2). The ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$ contain $\mathbf{x}_b$ and $\mathbf{y}_b$ for a random bit b.

Game_1 : This game generates the public parameters $\widetilde{pp}$ so that include a yes-instance $y \in \mathsf{TD}$. The difference to the previous game can be bounded via the hardness of deciding membership to TD.

Game_2 : The public parameters $\widehat{pp}$ change so that include a hiding $\widehat{crs}'$, and a (PIO) obfuscation of circuit $\widehat{\mathsf{C}}_{\mathrm{Add}}$, see Fig. 4. (Recall that this circuit uses the witness w_y to $y \in \mathsf{TD}$ to produce the output proofs $\widetilde{\pi}''$, and therefore the *simultaneous* knowledge of decryption keys sk_1,sk_2 is not needed anymore.) By Lemma 5.1 the difference with the previous game can be bounded by PIO and CRS indistinguishability.

Game_3 : This games generates $\mathbf{c}_2$ by encrypting $\mathbf{y}_1$, even when $b = 0$. We can bound the difference in any adversary's success probability via the IND-CPA advantage of Π with respect to pk_2 (the reduction will know (pk_1, sk_2) so as to be able to construct $\mathsf{C}_{\mathrm{Map}}$.)

Game_4**:** The public parameters are changed back to $\widetilde{pp}$, so that include a binding crs', and a (PIO) obfuscation of circuit $\mathsf{C}_{\mathrm{Add}}$ of Figure 2 (top). The difference with the previous game is bounded again with Lemma 5.1.

Game_5 : Now a no-instance $y \notin \mathsf{TD}$ is included in the public parameters pp. This game is justified by the hardness of deciding membership to TD.

Game_6 : This game uses sk_2 (in place of sk_1) in the generation of $\mathsf{C}_{\mathrm{Map}}$ circuit. In this transition we rely on the security of **Obf** and the perfect soundness of Σ. Perfect soundness implies consistency of the two representations underlying $\mathbf{c}_1$, $\mathbf{c}_2$ (recall that this means they represent the same group element with respect to $\mathbf{W}$). We then get that the two circuits (using sk_1, and sk_2, respectively) are funcitonally equivalent. We can then use the IO security of **Obf** to justify the switch from using sk_1 to using sk_2. (Note tht for any function X, any obfuscator that is for X-IND samplers is also secure as an indistinguishability obfuscator.) Note that in this game it is crucial that the crs' is in the binding mode.

Game_7 : This game, similarly to Game_1 switches to public parameters $\widetilde{pp}$ with a yes-instance $y \in \mathsf{TD}$. The analysis is as before.

Game_8 : This game, similarly to Game_2, includes in $\widehat{\mathsf{pp}}$ a hiding $\widehat{\mathsf{crs}}'$, and a (PIO) obfuscation of circuit $\widehat{\mathsf{C}}_{\text{Add}}$ (see Fig. 4). The analysis is as before.

Game_9 : This game generates $\mathbf{c}_1$ by encrypting $\mathbf{x}_1$, even when $b = 0$. The analysis is as in Game_3.

Observe that the challenge encoding in Game_9 is independent of the random bit b and the advantage of any (possibly unbounded) adversary $\mathcal{A}$ is 0. Collecting bounds on the probabilities involved in the various game hops concludes the proof.

G.	public parameters	C_{Add} knows	C_{Map} knows	$\mathbf{c}_1$ ($b = 0$) contains	$\mathbf{c}_2$ ($b = 0$) contains	**remark**
0	pp	sk_1,sk_2, td_{ext}	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_0, \mathbf{y}_0)$	
1	$\boxed{\widetilde{pp}}$	sk_1,sk_2, td_{ext}	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_0, \mathbf{y}_0)$	TD indist.
2	$\boxed{\widehat{pp}}$	$\boxed{w_y}$	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_0, \mathbf{y}_0)$	Lemma 5.1
3	$\widehat{\mathsf{pp}}$	w_y	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$\boxed{(\mathbf{x}_1, \mathbf{y}_1)}$	IND-CPA
4	$\boxed{\widetilde{pp}}$	$\boxed{sk_1,sk_2, td_{\text{ext}}}$	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	Lemma 5.1
5	$\boxed{pp}$	sk_1,sk_2, td_{ext}	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	TD indist.
6	pp	sk_1,sk_2, td_{ext}	$\boxed{sk_2}$	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	PIO
7	$\boxed{\widetilde{pp}}$	sk_1,sk_2, td_{ext}	sk_2	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	TD indist.
8	$\boxed{\widehat{pp}}$	$\boxed{w_y}$	sk_2	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	Lemma 5.1
9	$\widehat{pp}$	w_y	sk_2	$\boxed{(\mathbf{x}_1, \mathbf{y}_1)}$	$(\mathbf{x}_1, \mathbf{y}_1)$	IND-CPA

Figure 5: Outline of the proof steps of Theorem 5.3. b is the random bit of the κ-Switch game (see Figure 3). Changing between pp and $\widetilde{pp}$ is justified by the hardness of deciding membership of TD, and changing between $\widetilde{pp}$ and $\widehat{pp}$ by Lemma 5.1. The hops relying on PIO use the perfect completness and the perfect soundness under binding crs' to argue function equivalence of C_{Map}.

□

5.2 Doing without probabilistic obfuscation

In contrast to the PIO-based approach from Section 5.1, here we will only use (deterministic) indistinguishability obfuscation, but a stronger notion of NIZK proof system. Concretely, our proof works for any dual-mode NIZK proof system that enjoys perfect completeness, perfect soundness and extraction under **BCRS**, perfect WI under **HCRS** (as defined in Section 2.4), and meets a number of extra structural requirements as we detail below. These requirements are fulfilled by Groth–Sahai proofs [GS08] based on the DDH or k-Linear assumption, and we are in fact not aware of other suitable proof systems. Still, we find it convenient to state only the abstract properties that we require for the our result in this section.

The proof system $\Sigma = (\mathbf{HCRS}, \mathbf{BCRS}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{Sim})$ is required to satisfy the following structural properties.

Proof structure : Proofs π output by **Prove** or **Sim** have the form $\pi = (\pi_{\mathbf{com}}, \pi_{\mathbf{open}})$. We call $\pi_{\mathbf{com}}$ the *commitment part* and $\pi_{\mathbf{open}}$ the *opening part* of π.

Commitment parts output by Prove : The commitment part $\pi_{\mathbf{com}}$ of a proof generated by **Prove** is a (probabilistic) commitment to the respective witness $w \in \{0,1\}^\ell$. (Recall that we can restrict

ourselves to witnesses that are bit strings of a length $\ell = p(|x|)$ for a fixed and public polynomial p.) That is, we can write

$$\pi_{\mathbf{com}} \;=\; \mathbf{Com}(gpk, crs, w; r)$$

for a fixed commitment algorithm **Com** and **Prove**'s random coins r. Furthermore, **Prove** uses no random coins beyond those used for **Com**.

Homomorphic property of commitment parts : The commitment algorithm **Com** used by **Prove** and **Sim** is homomorphic in the following sense: for every *gpk*, every hiding *crs* (generated by **HCRS**), and all $w, w' \in \{0,1\}^\ell$, there exists a value $\Delta = \Delta(w, w')$, such that

$$\forall r: \qquad \mathbf{Com}(gpk, crs, w; r) = \mathbf{Com}(gpk, crs, w'; r + \Delta)\ , \tag{$\star$}$$

where we assume that the random coins of **Com** form a (finite) group whose operation is denoted by "+". In other words, w-commitments and 0^ℓ-commitments not only have the same distribution, they also only differ by a fixed shift in the random coins of **Com**. We furthermore require that Δ can be efficiently computed from w, w', and the simulation trapdoor td_{zk}.

SUITABILITY OF GROTH–SAHAI PROOFS. We briefly comment on the suitability of the NIZK proof system of Groth and Sahai [GS08].[9] The dual nature of the proof system, and its perfect completeness, perfect soundness, perfect extractability, and perfect witness-indistinguishability and perfect zero-knowledge are already proven in [GS08]. (However, cf. Footnote 4.) Moreover, it is easy to verify the syntactic requirements above.

To see ($\star$), it is necessary to take a look at the specific structure of the commitment algorithm **Com** from [GS08]. Specifically, when committing to a single bit b, commitments from [GS08] have the form

$$\mathbf{Com}(gpk, crs, b; r) = \left(b \cdot \mathbf{u}_0 + \sum_{i=1}^{n} r_i \cdot \mathbf{u}_i\right)\ ,$$

for publicly known vectors $\mathbf{u}_i$ of group elements (contained in *crs*), and random coins $r_i \in \mathbb{Z}_q$ for the order q of the underlying group. (Commitments to bit strings w can be generated in a bitwise fashion.) When *crs* is produced by **HCRS**, then $\mathbf{u}_0 \in \langle \mathbf{u}_1, \ldots, \mathbf{u}_n \rangle$. Then, a commitment to $b = 1$ differs from a commitment to $b = 0$ by an additive shift $\Delta = (\Delta_1, \ldots, \Delta_n) \in \mathbb{Z}_q^n$ of random coins with $\mathbf{u}_0 = \sum_i \Delta_i \mathbf{u}_i$. Note that this shift does not depend on r. This shows ($\star$).

THE DETERMINISTIC CIRCUIT C_{Add}. We now comment on a necessary slight tweak to the multilinear map construction itself. Namely, in order to facilitate a later use of ($\star$), we adapt the way **IO** generates obfuscations of C_{Add}. Recall that we can view C_{Add} as a deterministic circuit that takes as inputs (among other things) random coins r, and outputs (among other things) a NIZK proof $\pi = \mathbf{Prove}(gpk, crs, x, w; r)$ for a fixed witness w hardwired into C_{Add}. For our purposes, we use a slight variation of C_{Add} that instead generates π as $\mathbf{Prove}(gpk, crs, x, w; R)$, where R is a *uniformly random* value that is hardwired into C_{Add}. When we want to make the choice of R explicit, we also write C_{Add}^R.

Theorem 5.4 (Switching encodings using IO)**.** *Let* **IO** *be an indistinguishability obfuscator,* Π *an* IND-CPA *encryption scheme, and* Σ *the specific dual-mode NIZK proof system of Groth and Sahai (see [GS08]). Let* Γ *be the MLG scheme of Section 4 obtained using these primitives. Then, for any* PPT *adversary* $\mathcal{A}$*, there exist* PPT *adversaries* $\mathcal{B}_1$*,* $\mathcal{B}_2$*,* $\mathcal{B}_3$ *and* $\mathcal{B}_4$ *of essentially the same complexity as* $\mathcal{A}$ *such that for all* $\lambda \in \mathbb{N}$

[9] In fact [GS08] presents different variants of their proof system in prime-order and composite-order groups. Here we refer to the prime-order variants.

$$\mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{A}}(\lambda) \leq 3 \cdot \mathbf{Adv}^{\text{sm}}_{\mathsf{TD},\mathcal{B}_1}(\lambda) + 7 \cdot \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}_2}(\lambda) + 3 \cdot \mathbf{Adv}^{\text{crs}}_{\Sigma,\mathcal{B}_3}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\Pi,\mathcal{B}_4}(\lambda) \ .$$

Proof. The proof of Theorem 5.4 proceeds like that of Theorem 5.3, except of course in those steps that use the security of the probabilistic indistinguishability obfuscator **PIO**.

There are two types of such steps (resp. changes of C_{Map} or C_{Add}): in the first type, functional equivalence is fully preserved (even when viewing C_{Add} as a deterministic circuit). This type of change occurs in the hop from Game_0 to Game_1 in the proof of Lemma 5.1, and in the hop from Game_5 to Game_6 in the proof of Theorem 5.3. Since the corresponding deterministic circuits are functionally equivalent (in case of $\mathsf{C}_{\text{Add}} = \mathsf{C}^{\mathsf{R}}_{\text{Add}}$: when the same value of R is used), the security of **IO** can be directly utilized.

The second type of steps occurs in the hop from Game_2 to Game_3 in the proof of Lemma 5.1. More concretely, these games, which in the IO setting are slightly different, are as described below.

Game'_2 : The public parameters include, among other things, an IO obfuscation of the top circuit of Fig. 4, (albeit with the change we impose to C_{Add} in the IO setting). This circuit generates outputs proofs (if the input encodings are valid) using witness sk_1,sk_2.

Game'_3 : The public parameters include, among other things, an IO obfuscation of the bottom circuit of Fig. 4 (albeit with the change we impose to C_{Add} in the IO setting). This circuit generates proofs using always witness w_{y}.

We now argue that

$$|\Pr[\mathsf{W}'_2(\lambda)] - \Pr[\mathsf{W}'_3(\lambda)]| \leq \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}_1}(\lambda),$$

where W'_i denotes the event that $\mathcal{A}$ in Game'_i outputs 1. Using $(\star)$, the change results in a circuit that is functionally equivalent to the circuit from Game'_3 *when run with a suitably shifted random input* $\mathsf{R} + \Delta$. By our setup of C_{Add}, we can express this shift through the hardwired coins as

$$\mathsf{C}^{\mathsf{R}}_{\text{Add},3} \ \equiv \ \mathsf{C}^{\mathsf{R}+\Delta}_{\text{Add},2} \ ,$$

where $\mathsf{C}_{\text{Add},i}$ denotes the circuit C_{Add} from Game'_i, $\Delta = \Delta((sk_1, sk_2), w_{\mathsf{y}})$ is the appropriate shift vector from $(\star)$, and $\equiv$ denotes functional equivalence. Hence, our change in Game'_3 can be justified with a reduction to the (deterministic) indistinguishability property of **IO**. Specifically, a suitable circuit sampler $\mathcal{A}_1$ (as in Section 2.3) would sample circuits $\mathsf{C}_1 := \mathsf{C}^{\mathsf{R}}_{\text{Add},1}$ and $\mathsf{C}_2 := \mathsf{C}^{\mathsf{R}+\Delta}_{\text{Add},0}$ for a uniform R, and a Δ generated from the corresponding witnesses (sk_1, sk_2) and w_{y}. (We note that *during* this reduction, we can of course assume (sk_1, sk_2) and w_{y} to be known.)

We would like to highlight that Game'_3 itself still chooses a uniform R to prepare proofs. In particular, Game'_3 does not explicitly compute any Δ value as in $(\star)$, and hence does not make use of the corresponding witness (sk_1, sk_2). (The value Δ is only explicitly computed during the reduction to the indistinguishability property of **IO**.)

The remaining parts of the proof of Theorem 5.3 (including the proof of Lemma 5.1) apply unchanged.

□

$\underline{\mathrm{DDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)}$:	$\underline{\text{q-SDDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)}$:	$\underline{(\kappa, I)\text{-MDDH}^{\mathcal{A}}_{\Gamma}(\lambda)}$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}_0(1^\lambda, 1^0)$	$pp \leftarrow\!\!\$\ \mathbf{Setup}_0(1^\lambda, 1^0)$	$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^\lambda, 1^\kappa)$
$b \leftarrow\!\!\$ \{0,1\}$	$q \leftarrow q(\lambda)$; $b \leftarrow\!\!\$ \{0,1\}$	$b \leftarrow\!\!\$ \{0,1\}$
$x, y, z \leftarrow\!\!\$\ \mathbb{Z}_N$	$x, z \leftarrow\!\!\$\ \mathbb{Z}_N$	$a_1, \ldots, a_T, z \leftarrow\!\!\$\ \mathbb{Z}_N$
if $b = 1$ then	if $b = 1$ then	if $b = 1$ then
$\quad z \leftarrow x \cdot y$	$\quad z \leftarrow x^{q+1}$	$\quad [z]_T \leftarrow \mathbf{e}([a_1]_1, \ldots, [a_i]_i)^{a_T}$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [x]_0, [y]_0, [z]_0)$	$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [x]_0, \ldots, [x^q]_0, [z]_0)$	$b' \leftarrow\!\!\$\ \mathcal{A}(pp, \{[a_i]_j\}_{(i,j)\in I}, [z]_T)$
Return $(b = b')$	Return $(b = b')$	Return $(b = b')$

Figure 6: **Left**: The DDH problem. **Middle**: The strong DDH problem. **Right**: The multilinear DDH problem, where I specifies the available group elements. By slight abuse of notation, repeated use of $[a_i]_i$ denotes the same sample.

6 The Multilinear DDH Problem

In this section we show that natural multilinear analogues of the decisional Diffie–Hellman (DDH) problem are hard for our MLG scheme Γ from Section 4. We will establish this for two specific **Setup** algorithms which give rise to symmetric and asymmetric multilinear maps in groups of prime order N. (See Section 3 for the formal definition.) In the symmetric case, we will base hardness on the q-strong DDH problem [BBS04] and in the asymmetric case on the standard DDH problem.

6.1 Intractable problems

We start by formalizing the hard problems that we will be relying on and those whose hardness we will be proving. We do this in a uniform way using the language of group schemes of Section 3. Informally, the DDH problem requires the indistinguishability of g^{xy} from a random element given (g^x, g^y) for random x and y, the q-SDDH problem requires this for $g^{x^{q+1}}$ given $(g^x, g^{x^2}, \ldots, g^{x^q})$ and the κ-MDDH problem, whose hardness we will be establishing, generalizes the standard bilinear DDH problem (and its variants) and requires this for $g_T^{a_1 \cdots a_{\kappa+1}}$ in the presence of $(g^{a_1}, \ldots, g^{a_{\kappa+1}})$.

THE DDH PROBLEM. We say that a group scheme Γ_0 is DDH intractable if

$$\mathbf{Adv}^{\mathrm{ddh}}_{\Gamma_0,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathrm{DDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)\right] - 1 \in \mathrm{NEGL}\,,$$

where game $\mathrm{DDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)$ is shown in Figure 6 (left).

THE q-SDDH PROBLEM. For $q \in \mathbb{N}$ we say that a group scheme Γ_0 is q-SDDH intractable if

$$\mathbf{Adv}^{\text{q-sddh}}_{\Gamma_0,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\text{q-SDDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)\right] - 1 \in \mathrm{NEGL}\,,$$

where game $\text{q-SDDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)$ is shown in Figure 6 (middle).

THE (κ, I)-MDDH PROBLEM. For $\kappa \in \mathbb{N}$ we say that an MLG scheme Γ is κ-MDDH intractable with respect to the index set I if

$$\mathbf{Adv}^{(\kappa,I)\text{-mddh}}_{\Gamma,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[(\kappa, I)\text{-MDDH}^{\mathcal{A}}_{\Gamma}(\lambda)\right] - 1 \in \mathrm{NEGL}\,,$$

where game $(\kappa, I)\text{-MDDH}^{\mathcal{A}}_{\Gamma}(\lambda)$ is shown in Figure 6 (right). Here I is a set of ordered pairs of integers (i, j) with $1 \le i \le \kappa + 1$, $1 \le j \le \kappa$. The adversary is provided with challenge group

elements $[a_i]_j$ for $(i,j) \in I$, so that its challenge elements may lie in any combination of the groups. The standard MDDH problem corresponds to the case where

$$I = I^* := \{(1,1), \ldots, (\kappa,\kappa), (\kappa+1,\kappa)\}\,.$$

6.2 The symmetric setting

We describe a special variant of our general construction in Section 4 which gives rise to a *symmetric* MLG scheme as defined in Section 3. Recall that in the construction a matrix $\mathbf{W}$ was chosen uniformly at random in $\mathbb{Z}_N^{\kappa\times\ell}$. We set $\ell := 2$ and sample $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\kappa)^t$ by setting $\boldsymbol{\omega}_i = (1, \omega)$ for a random $\omega \in \mathbb{Z}_N$. The generators and identity elements for all groups are set to be a single value generated for the first group. These modifications ensure that the scheme algorithms are independent of the index for $1 \le i \le \kappa$ and that $\mathbf{e}$ is invariant under all permutations of its inputs.

The following lemma, which provides a mechanism to compute polynomial values "in the exponent," will be helpful in the security analysis of our constructions.

Lemma 6.1 (Horner in the exponent)**.** *Let $\boldsymbol{\omega} = (\omega_0, \omega_1, \omega_2) \in \mathbb{Z}_N$, and $\mathbf{x}_i = (x_{i,0}, x_{i,1}, x_{i,2}) \in \mathbb{Z}_N^3$ for $i = 1 \ldots \kappa$. Define $z_i := \langle \mathbf{x}_i, \boldsymbol{\omega}\rangle$. Then given only the implicit values $[\omega_0^i \omega_1^j \omega_2^k]_T$, for all i,j,k such that $i+j+k = \kappa$ and the explicit values $\mathbf{x}_i$ the element $[z_1 \cdots z_n]_T$ can be efficiently computed.*

Proof. Let

$$P(\omega_0,\omega_1,\omega_2) := \prod_{i=1}^{\kappa} (x_{i,0}\cdot\omega_0 + x_{i,1}\cdot\omega_1 + x_{i,2}\cdot\omega_2) = \sum_{i+j+k=\kappa} p_{ijk}\cdot\omega_0^i\omega_1^j\omega_2^k\,,$$

Clearly, if all p_{ijk} are known then $[P(\omega)]_T$ can be computed using $[\omega_0^i\omega_1^j\omega_2^k]_T$ with polynomially many operations. (There are $\mathcal{O}(\kappa^2)$ summands above.) To obtain these values we apply Horner's rule. Define

$$P_i(\omega_0,\omega_1,\omega_2) := \begin{cases} 1 & \text{if } i = 0\,; \\ (x_{i,0}\cdot\omega_0 + x_{i,1}\cdot\omega_1 + x_{i,2}\cdot\omega_2)\cdot P_{i-1}(\omega_0,\omega_1,\omega_2) & \text{otherwise.}\end{cases}$$

The coefficients of P_κ are the required p_{ijk} values. Let t_i denote the number of terms in P_i. It takes at most $3t_i$ multiplications and $t_i - 1$ additions in $\mathbb{Z}_N$ to compute the coefficients of P_i from P_{i-1} and $\mathbf{x}_i$. Since $t_i \in \mathcal{O}(\kappa^2)$, at most $\mathcal{O}(\kappa^3)$ many operations in total are performed. We note that the lemma generalizes to any (constant) ℓ with computational complexity $\mathcal{O}(\kappa^\ell)$. □

We prove the following result formally in Appendix A.2 and give an overview of the proof here. Below $I = I^*$ denotes the index set with all the second components being 1.

Theorem 6.2 ($(\kappa-1)$-SDDH hard $\implies$ symmetric (κ, I^*)-MDDH hard)**.** *Let Γ^* denote scheme Γ of Section 4 constructed using base group Γ_0 and an indistinguishability obfuscator* **IO** *with modifications as described above, and let $\kappa \in \mathbb{N}$. Then for any* PPT *adversary $\mathcal{A}$ there are* ppt *adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ of essentially the same complexity as $\mathcal{A}$ such that for all $\lambda \in \mathbb{N}$*

$$\mathbf{Adv}^{(\kappa,I^*)\text{-mddh}}_{\Gamma^*,\mathcal{A}}(\lambda) \le 2\cdot\mathbf{Adv}^{(\kappa-1)\text{-sddh}}_{\Gamma_0,\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}_1}(\lambda) + (\kappa+1)\cdot\mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma^*,\mathcal{B}_3}(\lambda) + \frac{\kappa-1}{N(\lambda)}\,.$$

Proof. In our reduction, the value ω used to generate $\mathbf{W}$ will play the role of the implicit value in the SDDH problem instance. We therefore change the implementation of C_{Map} to one that *does not know* ω in the clear and only uses the implicit values $[\omega^i]_0$ (recall that in our construction $\mathbb{G}_T$ is just $\mathbb{G}_0$, so these elements come from the SDDH instance). Such a circuit C^*_{Map} can be efficiently implemented using Horner's rule above. In more detail, C^*_{Map} has $[\omega^i]_T$ hard-coded in, recovers $\mathbf{x}_i$ from its inputs using sk_1, and then applies Lemma 6.1 with $(\omega_0, \omega_1, \omega_2) := (1, \omega, 0)$ to evaluate the multilinear map.

The proof proceeds along a sequence of $\kappa + 6$ games as follows.

Game_0 : This is the κ-MDDH problem (Figure 6, right). We use $\mathbf{x}_i$ and $\mathbf{y}_i$ to denote the representation vectors of a_i generated within the sampler $\mathbf{Sam}_{I(i)}(a_i)$, where $(i, I(i)) \in I$.

Game_1–Game_κ : In these games we gradually switch the representations of $[a_i]_1$ for $i \in [\kappa]$ so that they are of the form $(a_i - \omega, 1)$. Each hop can be bounded via the Switch game. (We have not (yet) changed the representation of $[a_{\kappa+1}]_1$.)

$\mathrm{Game}_{\kappa+1}$: This game introduces a conceptual change: the a_i for $i \in [\kappa]$ are generated as $a_i + \omega$. Note that the distributions of these values are still uniform and that the exponent of the MDDH challenge when $b = 1$ is

$$a_{\kappa+1} \cdot \prod_{i=1}^{\kappa} (a_i + \omega) \,.$$

This game prepares us for embedding a $(\kappa-1)$-SDDH challenge and then to stepwise randomize the exponent above.

$\mathrm{Game}_{\kappa+2}$: This game switches C_{Map} to C^*_{Map} as defined above. We use indistinguishability obfuscation and the fact that these circuits are functionally equivalent to bound this hop. We are now in a setting where ω is only implicitly known.

$\mathrm{Game}_{\kappa+3}$: This game replaces $[\omega^\kappa]_0$ with a random value $[\tau]_0$ in C^*_{Map} and the computation of the challenge exponent. This hop can be bounded via the $(\kappa - 1)$-SDDH game. Note that at this point the exponent is not information-theoretically randomized as τ is used within C^*_{Map}.

$\mathrm{Game}_{\kappa+4}$: This game sets the representation of $[a_{\kappa+1}]_1$ to $(a_{\kappa+1} - \omega, 1)$. Once again, this hop can be bounded by the Switch game.

$\mathrm{Game}_{\kappa+5}$: This game introduces a conceptual change analogous to that in $\mathrm{Game}_{\kappa+1}$ for $a_{\kappa+1}$. Note that a linear factor $(a_{\kappa+1} + \omega)$ is introduced in this game. This will help to fully randomize the exponent next.

$\mathrm{Game}_{\kappa+6}$: Analogously to $\mathrm{Game}_{\kappa+3}$, this game replaces $[\omega^\kappa]_0$ with a random value $[\sigma]_0$. We bound this hop using the $(\kappa - 1)$-SDDH game.

In $\mathrm{Game}_{\kappa+6}$, irrespective of the value of $b \in \{0, 1\}$, the challenge is uniformly and independently distributed as σ remains outside the view of the adversary. Hence the advantage of any (unbounded) adversary in this game is 0. This concludes the sketch proof. □

6.3 The asymmetric setting

We describe a second variant of the construction in Section 4 that results in an asymmetric MLG scheme. We set $\ell := 2$ and choose the matrix $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\kappa)^t$ by setting $\boldsymbol{\omega}_i := (1, \omega_i)$ for random $\omega_i \in \mathbb{Z}_N$.

The following theorem shows that for index set $I = \{(i, I(i)) : 1 \leq i \leq \kappa + 1\}$ given by an arbitrary function $I : [\kappa + 1] \longrightarrow [\kappa]$ of range at least 3, this construction is (κ, I)-MDDH intractable under the standard DDH assumption in the base group, the security of the obfuscator, and the κ-Switch game in Section 5. We present the proof intuition here and leave the details to Appendix A.3.

Theorem 6.3 (DDH hard $\implies$ asymmetric (κ, I^*)-MDDH hard)**.** *Let Γ^* denote scheme Γ of Section 4 constructed using base group Γ_0 and an indistinguishability obfuscator* **IO** *with modifications as described above. Let $\kappa \geq 3$ be a polynomial and I^* as above. Then for any* PPT *adversary $\mathcal{A}$ there are* ppt *adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ such that for all λ*

$$\mathbf{Adv}^{(\kappa,I^*)\text{-mddh}}_{\Gamma^*,\mathcal{A}}(\lambda) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{\Gamma_0,\mathcal{B}_1}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}_2}(\lambda) + 3 \cdot \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma^*,\mathcal{B}_3}(\lambda) + \frac{\kappa+1}{N(\lambda)} \,.$$

Proof. The general proof strategy is similar to that of the symmetric case, and proceeds along a sequence of 8 games as follows.

Game_0 : This is the (κ, I)-MDDH problem. Without loss of generality we assume that $I(i) = i$ for $i \in [3]$.

Game_1–Game_3 : In these games we gradually switch the representation vectors of $[a_i]_i$ for $i = 1, 2, 3$ to those of the form $(a_i - \omega_i, 1)$. Each of these hops can be bounded via the Switch game.

Game_4 : This game introduces a conceptual change and generates a_i as $a_i + \omega_i$. The exponent of the MDDH challenge when $b = 1$ is

$$(a_1 + \omega_1)(a_2 + \omega_2)(a_3 + \omega_3) \cdot \prod_{j \geq 4}^{\kappa+1} a_j \,.$$

Game_5 : In this game we change the implementation of C_{Map} to one which uses all but two of the ω_i explicitly, the remaining two implicitly, and additionally $[\omega_1\omega_2]_0$, i.e., $\omega_1\omega_2$ given implicitly in the exponent. The new circuit C^*_{Map} will be implemented using Horner's rule and is functionally equivalent to the original circuit used in the scheme. We invoke the IO security of the obfuscator to conclude the hop. This game prepares us to embed a DDH challenge next.

Game_6 : In this game we replace all the occurrences of $[\omega_1\omega_2]_0$ with a random $[\tau]_0$ and the corresponding implicit values. We bound the distinguishing advantage in this hop down to the DDH game.

Game_7 : Similarly to Game_5, we change the implementation of C^*_{Map} using $[\tau\omega_3]_0$ and argue via indistinguishability of obfuscations for functionally equivalent circuits.

Game_8 : Finally, using the hardness of DDH, we replace all the occurrences of $[\tau\omega_3]_0$ with a random $[\sigma]_0$.

In Game_8, irrespective of the value of $b \in \{0, 1\}$, the challenge is uniformly and independently distributed as σ remains outside the view of the adversary. Hence the advantage of any (possibly unbounded) adversary in this game is 0. □

(κ, m, n, r_0, r_1)-RANK$^{\mathcal{A}}_{\Gamma}(\lambda)$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^\lambda, 1^\kappa)$
$b \leftarrow\!\!\$ \{0,1\}$
$\mathbf{M}_0 \leftarrow\!\!\$\ \mathrm{Rk}_{r_0}(\mathbb{Z}_N^{m\times n})$; $\mathbf{M}_1 \leftarrow\!\!\$\ \mathrm{Rk}_{r_1}(\mathbb{Z}_N^{m\times n})$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [\mathbf{M}_b])$
Return $(b = b')$

Figure 7: The RANK problem parameterized by integers κ, m, n, r_0 and r_1.

7 The Rank Problem

The RANK problem is a generalization of DDH-like problems to matrices and has proven to be very useful in cryptographic constructions [BHHO08, NS09, GHV12, BLMR13, EHK+13]. Here we consider the problem in groups with non-unique encodings equipped with a multilinear map. Our main result is to show that, subject to certain restrictions, the intractability of the rank problem for our construction of an MLG scheme Γ from Section 4 follows from that of the q-SDDH problem for Γ_0.

7.1 Formalization of the problem

Let *pp* denote the public parameters of such an MLG scheme, obtained by running **Setup** with input $(1^\lambda, 1^\kappa)$. For simplicity, we focus on the case where N is prime. Let $\mathrm{Rk}_r(\mathbb{Z}_N^{m\times n})$ denote the set of $m \times n$ matrices over $\mathbb{Z}_N$ of rank r, where necessarily $r \le \min(m, n)$. We use a variant of our construction in Section 4, setting $\ell := 3$ and sampling $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\kappa)^t \in \mathbb{Z}_N^{\kappa\times 3}$ where $\boldsymbol{\omega}_i = (1, \omega, \omega^2)$ for $\omega \leftarrow\!\!\$\ \mathbb{Z}_N$. Note that this results in a symmetric pairing and henceforth we omit subscripts from source group elements. Let $[\mathbf{M}]$ denote a matrix whose (i,j)th entry contains an encoding of the form $[m_{i,j}] = ([m_{i,j}]_0, \mathbf{c}_{i,j,1}, \mathbf{c}_{i,j,2}, \pi_{i,j})$, with $m_{i,j} \in \mathbb{Z}_N$.

THE (κ, m, n, r_0, r_1)-RANK PROBLEM. For $\kappa, m, n, r_0, r_0 \in \mathbb{N}$ we say that an MLG scheme Γ is (κ, m, n, r_0, r_1)-RANK intractable if

$$\mathbf{Adv}^{(\kappa,m,n,r_0,r_1)\text{-rank}}_{\Gamma,\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[(\kappa, m, n, r_0, r_1)\text{-RANK}^{\mathcal{A}}_{\Gamma}(\lambda)\right] - 1 \in \mathrm{NEGL}\,,$$

where game (κ, m, n, r_0, r_1)-RANK$^{\mathcal{A}}_{\Gamma}(\lambda)$ is shown in Figure 7.

In the presence of a κ-linear map the rank problem is easy for any $r_0 < r_1 < \kappa$, since the determinants of all the r_b-minors can be expressed as forms of degree at most κ, and the multilinear map can be used to distinguish their images in the target group. However, this does not invalidate the plausibility of the rank problem for $\kappa \le r_0 < r_1$; indeed there are known reductions to the DDH, the decision linear and the 2-MDDH problems [BHHO08, NS09, GHV12].

We show that for our construction in Section 4, with the modification introduced above, the rank problem is indeed hard provided $\kappa \le r_0 < r_1$. A standard hybrid argument shows that it is sufficient to establish this for $r_1 := r_0 + 1$, with a polynomial loss in the security. Our main result is stated below. The full proof can be found in Appendix A.4.

Theorem 7.1 (SDDH $\implies$ RANK)**.** *Let Γ denote scheme Γ of Section 3 with $\ell := 3$ and with respect to the base group Γ_0 and an indistinguishability obfuscator* **IO***. Let κ, m, n, r be integers with $r \ge \kappa$. Then for*

any PPT *adversary* $\mathcal{A}$ *there are* ppt *adversaries* $\mathcal{B}_1$, $\mathcal{B}_2$ *and* $\mathcal{B}_3$ *of essentially the same complexity as* $\mathcal{A}$ *such that for all* $\lambda \in \mathbb{N}$

$$\mathbf{Adv}^{(\kappa,m,n,r,r+1)\text{-RANK}}_{\Gamma,\mathcal{A}}(\lambda) \leq \sum_{q=1}^{2\kappa-1} \mathbf{Adv}^{q\text{-sddh}}_{\Gamma_0,\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}_2}(\lambda) + (mn) \cdot \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{B}_3}(\lambda) + \frac{1}{N(\lambda)} .$$

7.2 Proof intuition

The main difficulty comes in generating consistent encodings of a rank r challenge matrix $[\mathbf{M}]$ throughout its gradual transformation into a rank $r+1$ challenge matrix. Contrast this with the MDDH reduction of Section 6, where the challenge that is transformed lives in the target group —a group with *unique* encodings. As we will see below, having encodings that are represented *also* with respect to ω^2 will help to overcome this problem and embed a 1-SDDH tuple.

EMBEDDING THE SDDH CHALLENGE. To reduce the rank problem to 1-SDDH, consider the following matrix

$$[\overline{\mathbf{W}}]_0 = \begin{bmatrix} [1]_0 & [\omega]_0 \\ [\omega]_0 & [\tau]_0 \end{bmatrix},$$

which is formed from an 1-SDDH challenge. We will exploit the fact that if $\tau = \omega^2$ then $\overline{\mathbf{W}}$ has rank 2, and if τ is uniform then it has rank 2 with overwhelming probability in λ.

LIFTING.To obtain an $m \times n$ matrix $\mathbf{M}$ of rank $r \geq \kappa$ or $r+1$ we can use the standard trick of embedding the identity matrix $\mathbf{I}_{r-1}$ in the diagonal:

$$\mathbf{M} = \begin{bmatrix} \mathbf{S} & & \\ & \mathbf{I}_{r-1} & \\ & & \mathbf{0} \end{bmatrix},$$

where $\mathbf{0}$ denotes padding with zeroes from $\mathbb{Z}_N$ to bring the matrix up to the required size. Moreover, via the random self-reducibility of the rank problem the structure in $\mathbf{M}$ can be removed. An important point worth mentioning is that after the randomization we are still able to generate an encoded matrix $[\mathbf{M}]$ even when ω and τ are only known in the exponent.

BREAKING CORRELATION WITH C_{Map}. We follow a similar strategy to break the dependency between C_{Map} and ω. Using the powers $[\mathbf{h}]_0 = ([1]_0, [\omega]_0, \ldots, [\omega^{2\kappa}]_0)$ we build circuit functionally equivalent to C_{Map}, indeed a circuit that outputs

$$\left[\prod_i^{\kappa} (x_{i,0} + x_{i,1}\omega + x_{i,2}\omega^2)\right]_T$$

via Lemma 6.1 (recall that $\mathbb{G}_T = \mathbb{G}_0$), and invoke the security of the obfuscator. We then use the q-SDDH assumptions for $2 \leq q \leq 2\kappa - 1$ in $\mathbb{G}_0$ to gradually transform $[\mathbf{h}]_0$ into $[\mathbf{q}]_0 = ([1]_0, [\omega]_0, [\omega^2]_0, [\tau_3]_0, \ldots, [\tau_{2\kappa}]_0)$ and embed a 1-SDDH tuple in the challenge matrix $[\mathbf{M}]$ as explained above.

Acknowledgements

Albrecht, Larraia and Paterson were supported by EPSRC grant EP/L018543/1. Hofheinz was supported by DFG grants HO 4534/2-2 and HO 4534/4-1.

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Dodis and Nielsen [DN15], pages 528–556.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Heidelberg, August 2008.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Canetti and Garay [CG13a], pages 410–428.

[BLR+15] Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In Oswald and Fischlin [OF15], pages 563–594.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 206–223. Springer, Heidelberg, August 2014.

[CG13a] Ran Canetti and Juan A. Garay, editors. *CRYPTO 2013, Part I*, volume 8042 of *LNCS*. Springer, Heidelberg, August 2013.

[CG13b] Ran Canetti and Juan A. Garay, editors. *CRYPTO 2013, Part II*, volume 8043 of *LNCS*. Springer, Heidelberg, August 2013.

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. Cryptology ePrint Archive, Report 2015/596, 2015. `http://eprint.iacr.org/2015/596`.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, April 2015.

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. Cryptology ePrint Archive, Report 2015/934, 2015. http://eprint.iacr.org/.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Canetti and Garay [CG13a], pages 476–493.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Gennaro and Robshaw [GR15], pages 267–286.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Dodis and Nielsen [DN15], pages 468–497.

[Cor15] Jean-Sebastien Coron. Cryptanalysis of GGH15 multilinear maps. Cryptology ePrint Archive, Report 2015/1037, 2015. http://eprint.iacr.org/.

[DN15] Yevgeniy Dodis and Jesper Buus Nielsen, editors. *TCC 2015, Part II*, volume 9015 of *LNCS*. Springer, Heidelberg, March 2015.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Canetti and Garay [CG13b], pages 129–147.

[FHPS13] Eduarda S. V. Freire, Dennis Hofheinz, Kenneth G. Paterson, and Christoph Striecks. Programmable hash functions in the multilinear setting. In Canetti and Garay [CG13a], pages 513–530.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH+13c] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Canetti and Garay [CG13b], pages 479–499.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Dodis and Nielsen [DN15], pages 498–527.

[GGI+14] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, pages 1–24, 2014.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GHV12] David Galindo, Javier Herranz, and Jorge L. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 627–642. Springer, Heidelberg, September 2012.

[GR15] Rosario Gennaro and Matthew J. B. Robshaw, editors. *CRYPTO 2015, Part I*, volume 9215 of *LNCS*. Springer, Heidelberg, August 2015.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

[HJ15] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. Cryptology ePrint Archive, Report 2015/301, 2015. `http://eprint.iacr.org/2015/301`.

[HSW13] Susan Hohenberger, Amit Sahai, and Brent Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In Canetti and Garay [CG13a], pages 494–512.

[MF15] Brice Minaud and Pierre-Alain Fouque. Cryptanalysis of the new multilinear map over the integers. Cryptology ePrint Archive, Report 2015/941, 2015. `http://eprint.iacr.org/`.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer, Heidelberg, August 2009.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[OF15] Elisabeth Oswald and Marc Fischlin, editors. *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*. Springer, Heidelberg, April 2015.

[PTT10] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Optimal authenticated data structures with multilinear forms. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *PAIRING 2010*, volume 6487 of *LNCS*, pages 246–264. Springer, Heidelberg, December 2010.

[TLL14] Fei Tang, Hongda Li, and Bei Liang. Attribute-based signatures for circuits from multilinear maps. In Sherman S. M. Chow, Jan Camenisch, Lucas Chi Kwong Hui, and Siu-Ming Yiu, editors, *ISC 2014*, volume 8783 of *LNCS*, pages 54–71. Springer, Heidelberg, October 2014.

[YYHK14] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 90–107. Springer, Heidelberg, August 2014.

[YYHK15] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. Cryptology ePrint Archive, Report 2015/128, 2015. http://eprint.iacr.org/2015/128.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In Oswald and Fischlin [OF15], pages 439–467.

A Full Proofs from the Main Body

A.1 Proof of Theorem 5.3: Indistinguishability of encodings using PIO

Proof. We consider a chain of 10 games, with Game_0 the κ-Switch game, such that in the last game the challenge encoding is drawn independently of the bit b. Below we let W_i denote the event that Game_i outputs 1.

Game_0 : The original Switch game.

Game_1 : As Game_0 but now the public parameters $\widetilde{pp}$ are changed so that include a yes-instance $y \in \mathsf{TD}$. We have that

$$|\Pr[W_0(\lambda)] - \Pr[W_1(\lambda)]| \leq \mathbf{Adv}^{\text{sm}}_{\mathsf{TD},\mathcal{B}_1}(\lambda),$$

where TD is a language where is hard to decide membership.

Game_2 : The public parameters $\widehat{pp}$ change so that include a hiding $\widehat{crs}'$, and a (PIO) obfuscation of circuit $\widehat{C}_{\text{Add}}$ (see Fig. 4 (bottom)). Recall that this circuit uses the witness w_y to $y \in \mathsf{TD}$ to produce the output proofs $\widetilde{\pi}''$. Therefore the *simultaneous* knowledge of decryption keys sk_1,sk_2 is not needed anymore. By Lemma 5.1 we have that

$$|\Pr[W_1(\lambda)] - \Pr[W_2(\lambda)]| \leq 2 \cdot \mathbf{Adv}^{\text{ind}}_{\mathbf{PIO},\mathcal{B}_2}(\lambda) + \mathbf{Adv}^{\text{crs}}_{\Sigma,\mathcal{B}_3}$$

Game_3 : As Game_2, but, if $b = 0$ the challenge encoding is generated by mixing the representation vectors w.r.t public key pk_2. Thus, on $\mathcal{A}$'s response $(z, (\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_1))$, in this game we set $\mathbf{c}_0 \leftarrow \mathbf{Enc}(\mathbf{x}_0, pk_1; \mathbf{r}_1)$, and $\mathbf{c}_1 \leftarrow \mathbf{Enc}(\mathbf{y}_1, pk_2; \mathbf{r}_2)$.

Claim A.1. $|\Pr[W_2(\lambda)] - \Pr[W_3(\lambda)]| \leq \mathbf{Adv}^{\text{ind-cpa}}_{\Pi,\mathcal{B}_4}(\lambda)$.

Proof Claim A.1. Consider the following PPT distinguisher $\mathcal{B}_4$ against the IND-CPA security of the encryption scheme Π, with respect to key pair (pk_2, sk_2). The distinguisher runs experiment Game_2 using $\mathcal{A}$ as a subroutine with the following differences: when it receives $\mathcal{A}$'s vectors $(\mathbf{x}_j, \mathbf{y}_j)$ (in $\mathbb{Z}_p^\ell$ for $j = 0, 1$) it submits $(\mathbf{y}_0, \mathbf{y}_1)$ to the IND-CPA challenger. It gets back $\mathbf{c}^* = \mathbf{Enc}(\mathbf{y}_{r^*}, pk_2)$. Next, $\mathcal{B}_4$ generates $\mathbf{c}_1 \leftarrow \mathbf{Enc}(\mathbf{x}_0, pk_1)$, and sets $\mathbf{c}_2 = \mathbf{c}^*$; the proof π on instance $x = ([z]_i, \mathbf{c}_1, \mathbf{c}_2)$ is generated using the simulation trapdoor of the proof system. Namely, $\pi \leftarrow_\$ \mathbf{Sim}(crs, x, td_{zk})$. Finally, $\mathcal{B}_4$ outputs what $\mathcal{A}$ outputs.

Algorithm $\mathcal{B}_4$ perfectly simulates the challenger in experiment Game_2 if $r^* = 0$ and in experiment Game_3 if $r^* = 1$. This follows from (1) (x, π) is a valid encoding, indeed ciphertext $\mathbf{c}^*$ contains an encryption of $\mathbf{y}_{r^*}$, such that $[z]_i = [\langle \mathbf{y}_{r^*}, \boldsymbol{\omega}_i \rangle]_i$; and (2) real and simulated proofs are identically distributed under (the hiding) $\widehat{crs}'$ included in $\widehat{pp}$. □

Game_4**:** The public parameters are changed back to $\widetilde{pp}$, so that include a binding crs', and a (PIO) obfuscation of circuit C_{Add} of Fig. 2 (top). ($\widetilde{\text{pp}}$ also include a yes-instance $y \in \mathsf{TD}$.) Again by Lemma 5.1 we have that

$$|\Pr[W_3(\lambda)] - \Pr[W_4(\lambda)]| \leq 2 \cdot \mathbf{Adv}^{\text{ind}}_{\mathbf{PIO},\mathcal{B}_2}(\lambda) + \mathbf{Adv}^{\text{crs}}_{\Sigma,\mathcal{B}_3}.$$

Game_5 : As Game_4 but now the public parameters pp are changed back to the original one described in Section 4 so that include a no-instance $y \notin \mathsf{TD}$. We have that

$$|\Pr[W_4(\lambda)] - \Pr[W_5(\lambda)]| \leq \mathbf{Adv}^{\text{sm}}_{\mathsf{TD},\mathcal{B}_1}(\lambda),$$

where TD is a language where is hard to decide membership.

Game_6 : As Game_5, but now the challenger constructs a different circuit C_{Map} with the second encryption secret key hard-coded. Thus, the extracted vector is set to $\mathbf{y}_i \leftarrow \mathbf{Dec}(\mathbf{c}_{i,1}, sk_2)$. We claim that

$$|\Pr[W_5(\lambda)] - \Pr[W_6(\lambda)]| \leq \mathbf{Adv}^{\text{ind}}_{\mathbf{PIO},\mathcal{B}_1}(\lambda).$$

The variants of the C_{Map} circuit described in the games extract (possibly different) encoding vectors $\mathbf{x}^*_i$, $\mathbf{y}^*_i$, respectively, for any adversarial input $\mathbf{x}^* = (x^*_1, \ldots, x^*_k)$. Observe that the i-th argument $x^*_i = (i, [z_i]_0, \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i)$ has a non-rejecting proof π_i iff $([z_i]_0, \mathbf{c}_{i,1}, \mathbf{c}_{i,2})$ passes relation $\mathbf{R}_1$. (In other words, the ciphertexts encrypt representation vectors of the same $[z_i]_0$.) It follows, from the perfect completeness and perfect soundness of the proof system with a binding CRS, that these variants behave identically on any (possibly malformed) input $\mathbf{x}^*$. Therefore the variants are functionally equivalent and hence trivially drawn by an X-IND sampler, so that their PIO obfuscations are indistinguishable.

Game_7 : As Game_6 but now the public parameters $\widetilde{pp}$ are changed so that include a yes-instance $y \in \mathsf{TD}$. We have that

$$|\Pr[W_6(\lambda)] - \Pr[W_7(\lambda)]| \leq \mathbf{Adv}^{\text{sm}}_{\mathsf{TD},\mathcal{B}_1}(\lambda),$$

where TD is a language where is hard to decide membership.

Game_8 : The public parameters $\widehat{\text{pp}}$ change so that include a hiding $\widehat{crs}'$, and a (PIO) obfuscation of circuit $\widehat{\mathsf{C}}_{\text{Add}}$ (see Fig. 4 (bottom)). By Lemma 5.1 we have that

$$|\Pr[W_7(\lambda)] - \Pr[W_8(\lambda)]| \leq 2 \cdot \mathbf{Adv}^{\text{ind}}_{\mathbf{PIO},\mathcal{B}_2}(\lambda) + \mathbf{Adv}^{\text{crs}}_{\Sigma,\mathcal{B}_3}$$

Game_9 : As Game_8, but, if $b = 0$ the challenge encoding is generated by mixing the representation vectors w.r.t public key pk_1. Thus, on $\mathcal{A}$'s response $(z, (\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_1))$, in this game we set $\mathbf{c}_0 \leftarrow \mathbf{Enc}(\mathbf{x}_1, pk_1; \mathbf{r}_1)$, and $\mathbf{c}_1 \leftarrow \mathbf{Enc}(\mathbf{y}_1, pk_2; \mathbf{r}_2)$. Using a similar argument as in Claim A.1 we have that

$$|\Pr[W_8(\lambda)] - \Pr[W_9(\lambda)]| \leq \mathbf{Adv}^{\text{ind-cpa}}_{\Pi,\mathcal{B}_4}(\lambda).$$

Finally, $\Pr[W_9(\lambda)] = 1/2$ because the challenge encoding is generated using the same pair of representation vectors $(\mathbf{x}_1, \mathbf{y}_1)$ regardless of the bit b. The proof of the theorem is concluded by collecting the terms above.

□

(κ, I^*)-MDDH$^{\mathcal{A}}_{\Gamma}(\lambda)$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^\lambda, 1^\kappa)$
$b \leftarrow\!\!\$\ \{0,1\}$; $z \leftarrow\!\!\$\ \mathbb{Z}_N$
$a_1, \ldots, a_{\kappa+1} \leftarrow\!\!\$\ \mathbb{Z}_N$
if $b = 1$ then $z \leftarrow a_1 \cdots a_{\kappa+1}$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, \{[a_i]_j\}_{(i,j)\in I^*}, [z]_T)$
Return $(b = b')$

Figure 8: The symmetric multilinear DDH problem for our MLG scheme. Here $I^* = \{(1,1), \ldots, (\kappa+1, 1)\}$.

A.2 Proof of Theorem 6.2: Hardness of symmetric MDDH

Proof. We show via a chain of games, starting with the symmetric κ-MDDH problem, such that the last game chooses the challenge at random and independently of the guess bit b. Below we let W_i denote the event that Game_i outputs 1.

Game_0 : The κ-MDDH problem as shown in Figure 8. Here there is only one source group.

Game_s **for** $1 \le s \le \kappa$: As Game_{s-1}, the difference is that the representation vectors $(\mathbf{x}_s, \mathbf{y}_s)$ of the s-th challenge encoding $[a_s]$ are given by

$$x_{s,0} = y_{s,0} = a_s - \omega \quad \text{and} \quad x_{s,1} = y_{s,1} = 1 \,.$$

Thus, in game $s' \ge s$ the second coordinate of the s-th encoding vectors are *always* fixed. Using a similar argument as in Claim A.2 (see the proof of Theorem 6.3) we have that

$$|\Pr[W_{s-1}(\lambda)] - \Pr[W_s(\lambda)]| \le \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{B}}(\lambda) \text{ for } 1 \le s \le \kappa \,.$$

$\text{Game}_{\kappa+1}$: The i-th source exponent is changed to $a'_i = a_i + \omega$ for randomly chosen $a_i \in \mathbb{Z}_N$ and $i \le \kappa$. This means that the target exponent for $b = 1$ is

$$d = (a_1 + \omega) \cdots (a_\kappa + \omega) \cdot a_{\kappa+1} \tag{1}$$

The distribution from which the first κ exponents a'_i are drawn has not changed, and indeed is the uniform distribution. Therefore $\Pr[W_\kappa(\lambda)] = \Pr[W_{\kappa+1}(\lambda)]$.

$\text{Game}_{\kappa+2}$: The differences with the previous game are two-fold.

First, for case $b = 1$ the challenge group element $[d]_T$ is generated as in Lemma 6.1. Thus, we first write Equation (1) as

$$d = P(\omega) \cdot a_{\kappa+1} \,, \tag{2}$$

where P is a degree κ polynomial whose coefficients $\mathbf{p} = (p_0, \ldots, p_\kappa)$ are computed using the iterative rule of Lemma 6.1, with $(x_{i,0}, x_{i,1}, 0) = (a_i, 1, 0)$. Then $[d]_T$ is obtained by evaluating P at point ω in the exponent using group elements $([1]_T, [\omega]_T, \ldots, [\omega^\kappa]_T)$.

The other difference is that we obfuscate a different circuit C^*_{Map} which has the powers $[\omega^i]_T$ hard-coded, for $1 \le i \le \kappa$. This new circuit extracts the encoding vectors $\mathbf{x}_i$ from the inputs, as usual, then it computes the coefficients of P as explained above, and evaluates it at ω in the exponent.

Lemma 6.1 implies that (1) both circuits are functionally equivalent, and (2) C^*_{Map} is of size $\mathtt{poly}(\lambda)$. We conclude that obfuscations of these two variants are indistinguishable. Or putting it differently:

$$|\Pr[W_{\kappa+1}(\lambda)] - \Pr[W_{\kappa+2}(\lambda)]| \leq \mathbf{Adv}^{\mathrm{ind}}_{\mathbf{IO},\mathcal{B}}(\lambda) \ .$$

$\mathrm{Game}_{\kappa+3}$: Here a different challenge $[d]_T$ is generated. We now regard the $(\kappa+1)$-vector $\mathbf{p}$ of Equation (2) as a multivariate $\mathbb{Z}_N$-polynomial P in κ unknowns. The challenger samples random ω, τ in $\mathbb{Z}_N$ and sets

$$[d]_T = a_{\kappa+1} \cdot [P(\omega, \omega^2, \ldots \omega^{\kappa-1}, \tau)]_T \ , \tag{3}$$

where P is evaluated in the exponent. Also, the circuit C^*_{Map} has hard-coded $[\tau]_T$ and $[\omega^i]_T$ for $1 \leq i \leq \kappa - 1$. We emphasize that in this game, for challenge bit $b = 1$, the random variables sampling d and (an obfuscation of) C_{Map} are not independent (in case $b = 0$ they are).

We claim that

$$|\Pr[W_{\kappa+2}(\lambda)] - \Pr[W_{\kappa+3}(\lambda)]| \leq \mathbf{Adv}^{(\kappa-1)\text{-sddh}}_{\Gamma_0,\mathcal{B}}(\lambda) \ .$$

This follows because to simulate both games it suffices to know $[\tau]_T$ and $[\omega^i]_T$ in the exponent. Thus, an adversary $\mathcal{B}$ against $(\kappa-1)$-SDDH on receiving challenge $([\omega^i]_0)_{i\leq\kappa-1}, [\tau]_0)$ can simulate experiment $\mathrm{Game}_{\kappa+2}$ if $\tau = \omega^\kappa$, or experiment $\mathrm{Game}_{\kappa+3}$ if τ is random. (Recall that $\mathbb{G}_T = \mathbb{G}_0$.)

$\mathrm{Game}_{\kappa+4}$: The last source encoding $a_{\kappa+1}$ is generated with representation vectors $\mathbf{x}_{\kappa+1} = \mathbf{y}_{\kappa+1} = (a_{\kappa+1} - \omega, 1)$. Using a similar argument as Claim A.2 we have that

$$|\Pr[W_{\kappa+3}(\lambda)] - \Pr[W_{\kappa+4}(\lambda)]| \leq \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma^*,\mathcal{B}}(\lambda) \ .$$

(Technically, the circuit C_{Map} in game Switch is a κ-linear map, and at this point of the transformation C^*_{Map} is not necessarily linear. However, a closer look to the proof of Theorem 5.3 shows that no linearity assumption is made on the C_{Map} circuit.)

$\mathrm{Game}_{\kappa+5}$: The last source exponent is sampled as $a'_{\kappa+1} = a_{\kappa+1} + \omega$ for a randomly chosen $a_{\kappa+1}$. This means that, for a known τ in $\mathbb{Z}_N$, the challenge d in Equation (3) can be written as

$$\begin{aligned}[d]_T &= (a_{\kappa+1} + \omega) \cdot [P(\omega, \omega^2, \ldots \omega^{\kappa-1}, \tau)]_T \\ &= a_{\kappa+1} \cdot [P(\omega, \omega^2, \ldots \omega^{\kappa-1}, \tau)]_T + [Q_\tau(\omega, \omega^2, \ldots, \omega^\kappa)]_T\end{aligned}$$

where the coefficients $\mathbf{p}$ of polynomial P are computed as explained in $\mathrm{Game}_{\kappa+2}$, and polynomial Q_τ is given by coefficients $\mathbf{q} = (0, p_0 + \tau p_\kappa, p_1, \ldots, p_{\kappa-1})$.

Since $a'_{\kappa+1}$ and d (for case $b = 1$) are distributed as in the previous game we have that $\Pr[W_{\kappa+4}(\lambda)] = \Pr[W_{\kappa+5}(\lambda)]$.

$\mathrm{Game}_{\kappa+6}$: The last game samples the challenge $[d]_T$ for case $b = 1$ as

$$[d]_T = a_{\kappa+1} \cdot [P(\omega, \omega^2, \ldots \omega^{\kappa-1}, \tau)]_T + [Q_\tau(\omega, \omega^2, \ldots, \sigma)]_T \ .$$

Here σ is a random fresh value in $\mathbb{Z}_N$. A $(\kappa-1)$-SDDH challenge $([\omega^i]_0)_{i\leq\kappa-1}, [\sigma]_0)$ can be used to emulate the challenger in $\mathrm{Game}_{\kappa+4}$ if $\sigma = \omega^\kappa$, or in $\mathrm{Game}_{\kappa+5}$ if σ is random. The

(κ, I^*)-MDDH$^{\mathcal{A}}_{\Gamma}(\lambda)$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^\lambda, 1^\kappa)$
$b \leftarrow\!\!\$\ \{0,1\}$; $z \leftarrow\!\!\$\ \mathbb{Z}_N$
$a_1, \ldots, a_{\kappa+1} \leftarrow\!\!\$\ \mathbb{Z}_N$ $a_1, \ldots, a_{\kappa+1} \leftarrow\!\!\$\ \mathbb{Z}_N$
if $b = 1$ then $z \leftarrow a_1 \cdots a_{\kappa+1}$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, \{[a_i]_j\}_{(i,I(i))}, [z]_T)$
Return $(b = b')$

Figure 9: The asymmetric multilinear DDH problem for our MLG scheme. Here I is a function defining the index set $I = (i, I(i))$.

latter follows again from the fact that, if τ is given in the clear, knowing ω^i, σ in the exponent suffices to generate $[d]_T$, since P and Q_τ are evaluated in the exponent. (Recall that $\mathbb{G}_T = \mathbb{G}_0$.) This shows:

$$|\Pr[W_{\kappa+5}(\lambda)] - \Pr[W_{\kappa+6}(\lambda)]| \leq \mathbf{Adv}^{(\kappa-1)\text{-sddh}}_{\Gamma_0,\mathcal{B}}(\lambda)\ .$$

To conclude, to see that $\Pr[W_{\kappa+6}] \leq 1/2 + \mathrm{negl}(\lambda)$ it suffices to show that if the challenge bit is $b = 1$, the exponent target challenge d is randomly distributed. This follows because the last coefficient of Q_τ, namely $p_{\kappa-1}$ given by

$$p_{\kappa-1} = \sum_{i=1}^{\kappa} \left(\prod_{1 \leq j \neq i \leq \kappa} a_j \right)$$

has no inverse in $\mathbb{Z}_N$ with probability $\mathrm{negl}(\lambda) := (\kappa - 1)/N(\lambda)$ provided N is prime. Now, for any fixed τ and $(\omega^i)_{1 \leq i \leq \kappa}$, the map $Q_\tau + a_{\kappa+1} P(\omega, \ldots, \omega^{\kappa-1}, \tau)$ defines a bijection over $\mathbb{Z}_N$ which acts on uniform σ. It follows that $[d]_T$ is distributed uniformly in $\mathbb{G}_T$.

□

A.3 Proof of Theorem 6.3: Hardness of asymmetric MDDH

Proof. Let $\kappa \geq 3$ and $I : [\kappa+1] \longrightarrow [\kappa]$ be any function of range with size at least 3. Slightly abusing notation, we set $I = (i, I(i))$ for $1 \leq i \leq \kappa + 1$.

We show a chain of games, starting with the asymmetric (κ, I)-MDDH problem, such that the last game chooses the challenge encoding at random and independently of the challenge bit b. Below we let W_i denote the event that Game_i outputs 1. For simplicity, and without loss of generality, we assume that $I(i) = i$ for $i = 1, 2, 3$.

Game_0 : The asymmetric (κ, I)-MDDH problem as shown in Figure 9.

Game_s **for** $s = 1, 2, 3$: Similar to Game_{s-1} with the difference that the representation vectors $(\mathbf{x}_s, \mathbf{y}_s)$ of the source encoding $[a_s]_s$ are given by

$$x_{s,0} = y_{s,0} = a_s - \omega_s \quad \text{and} \quad x_{s,1} = y_{s,1} = 1\ .$$

Thus, in game $s' \geq s$ the second coordinates of the s-th encoding vectors are *always* fixed.

Claim A.2. $|\Pr[W_{s-1}(\lambda)] - \Pr[W_s(\lambda)]| \leq \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{B}}(\lambda)$.

Proof. Consider the following PPT adversary $(\mathcal{B}_1, \mathcal{B}_2)$ against game κ-Switch of Figure 3. $\mathcal{B}_1$ outputs $((\mathbf{x}_{s-1}, \mathbf{y}_{s-1}), (\mathbf{x}_s, \mathbf{y}_s), s, st)$ representing a uniform value a_s in $\mathbb{Z}_N$, where $(\mathbf{x}_{s-1}, \mathbf{y}_{s-1})$ is as in Game_{s-1} and $(\mathbf{x}_s, \mathbf{y}_s)$ as in Game_s. $\mathcal{B}_1$ can form these vectors because it knows matrix $\mathbf{W}$ and a_s explicitly. Next $\mathcal{B}_2$ receives an encoding $[a_s]_s$ that has embedded in it vector $(\mathbf{x}_{s+b-1}, \mathbf{y}_{s+b-1})$ for a random bit b, and uses $[a_s]_s$ to simulate Game_{s+b-1}. Last, $\mathcal{B}_2$ outputs what $\mathcal{A}$ outputs. □

Game_4 : We change the first three source exponents to $a'_i = a_i + \omega_i$ for randomly chosen $a_i \in \mathbb{Z}_N$. This means that the target exponent for $b = 1$ is

$$d = (a_1 + \omega_1)(a_2 + \omega_2) \cdot (a_3 + \omega_3) \cdot a_{I(4)} \cdots a_{I(\kappa+1)} .$$

Here we use the fact that $|\text{Rng}(I)| \geq 3$. The first three elements a'_i are drawn from the uniform distribution, and their respective representation vectors are $(a_i, 1)$ so $\Pr[W_3(\lambda)] = \Pr[W_4(\lambda)]$.

Game_5 : The implementation of C_{Map} is changed. Now it has hard-coded

$$[\omega_1]_0, [\omega_2]_0, [\tau]_0, \omega_3, \omega_{I(4)} \ldots, \omega_{I(\kappa)} \quad \text{where} \quad \tau = \omega_1\omega_2 .$$

The polynomial $P(w_1, \ldots, w_\kappa) = \prod_{i=1}^{\kappa}(x_{i,0} + x_{i,1}w_i)$ on point $(\omega_{I(1)}, \ldots, \omega_{I(\kappa)})$ can be evaluated in the exponent knowing $[\omega_1]_0$, $[\omega_2]_0$, $[\omega_1\omega_2]_0$, and explicit $\omega_{I(i)}$ for $i \geq 3$ with $I(i) \neq 1, 2$. Since the output of the original C_{Map} is exactly $[P(\omega_{(I(1)}, \ldots, \omega_{I(\kappa)})]_T$ we conclude that

$$|\Pr[W_4(\lambda)] - \Pr[W_5(\lambda)]| \leq \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}}(\lambda) .$$

Game_6 : The challenge target d is set to

$$d = (a_1a_2 + \omega_1a_2 + \omega_2a_1 + \tau)(a_3 + \omega_3)a_{I(4)} \ldots a_{I(\kappa+1)} ,$$

where τ is a fresh random value in $\mathbb{Z}_N$, and C_{Map} has hard-coded

$$[\omega_1]_0,\ [\omega_2]_0,\ [\tau]_0,\ \omega_3,\ \omega_{I(4)},\ \ldots,\ \omega_{I(\kappa)} .$$

We note that the circuit is different to the previous one. More concretely, if $\mathbf{x}_i = (x_{i,0}, x_{i,1})$ is the first representation vector of the ith input $[z]_i$, the new C^*_{Map} outputs the evaluation of the following function at point $(\tau, \omega_{I(1)}, \ldots, \omega_{I(\kappa)})$:

$$f(\tau, w_1, \ldots, w_\kappa) = (x_{1,1}x_{2,0}w_1 + x_{1,0}x_{2,1}w_2 + x_{1,1}x_{2,1}\tau + x_{1,0}x_{2,0})(x_{3,0} + x_{3,1}w_3) \prod_{i\geq 4}^{\kappa}(x_{i,0} + x_{i,1}w_i) .$$

Here it is enough to know $[\omega_1]_0$, $[\omega_2]_0$, and $[\tau]_0$ to compute $[f(\tau, \omega_{I(1)}, \ldots, \omega_{I(\kappa)})]_T$. Also, note that if $\tau = \omega_1\omega_2$ then this is precisely the output of C_{Map} in the previous game. Thus, a DDH challenge $([\omega_1]_0, [\omega_2]_0, [\tau]_0)$ can be used to generate the pair $([d]_T, \overline{C^*}_{\text{Map}})$ [10] as in Game_5 if $\tau = \omega_1\omega_2$, or as in Game_6 if τ is random. This shows:

$$|\Pr[W_5(\lambda)] - \Pr[W_6(\lambda)]| \leq \mathbf{Adv}^{\text{ddh}}_{\Gamma_0,\mathcal{B}}(\lambda) .$$

[10] Here, $[d]_T$ and $\overline{C^*}_{\text{Map}}$ are still correlated via $[\tau]_0$. This is why we cannot stop at Game_6.

Game_7 : Circuit C^*_{Map} has hard-coded

$$\omega_1,\ \omega_2,\ [\tau]_0,\ [\sigma]_0,\ [\omega_3]_0,\ \omega_{I(4)},\ \ldots,\ \omega_{I(\kappa)},\ \text{where } \sigma = \tau\omega_3\ .$$

Here it suffices to know $[\tau]_0$, $[\sigma]_0$, and $[\omega_3]_0$ to evaluate $[f(\tau, \omega_{I(1)}, \ldots, \omega_{I(\kappa)})]_T$, so we have

$$|\Pr[W_6(\lambda)] - \Pr[W_7(\lambda)]| \leq \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}}(\lambda)\ .$$

Game_8 : Random $\sigma \in \mathbb{Z}_N$ is sampled and the challenge target exponent d is set to

$$(\sigma + a_1a_2a_3 + a_1a_2\omega_3 + a_2a_3\omega_1 + a_2\omega_1\omega_3 + a_1a_3\omega_2 + a_1\omega_2\omega_3 + a_3\tau) \cdot a_{I(4)} \cdots a_{I(\kappa+1)} \quad (4)$$

The circuit C^*_{Map} has hard-coded the same values as in the previous game. (The functionality of the circuit changes since now σ is arbitrary, but the algorithm remains the same.) Once again, a DDH challenge $([\tau]_0, [\omega_3]_0, [\sigma]_0)$ can be used to generate the pair $([d]_T, \overline{C^*}_{\text{Map}})$ as in Game_7 if $\sigma = \tau\omega_3$, or as in Game_8 if σ is random, therefore

$$|\Pr[W_7(\lambda)] - \Pr[W_8(\lambda)]| \leq \mathbf{Adv}^{\text{ddh}}_{\Gamma_0,\mathcal{B}}(\lambda)\ .$$

To conclude, we have $\Pr[W_8(\lambda)] \leq 1/2 + \text{negl}(\lambda)$. To see this, we argue that d is randomly distributed in $\mathbb{Z}_N$ for challenge bit $b = 1$ with overwhelming probability in λ as follows: if N is prime, then $\prod_{j=1}^{\kappa+1} a_{I(j)}$ has an inverse in $\mathbb{Z}_N$, and therefore d in Equation 4 seen as a function of σ and parametrized by τ and $a_{I(j)}$ defines a bijection in $\mathbb{Z}_N$ with overwhelming probability. Thus, if σ is uniform so is d.

□

A.4 Proof of Theorem 7.1: Hardness of RANK

Proof. Let $\overline{\mathbf{S}} = ((a, b)(c, d))^t$ and $\overline{\mathbf{W}} = ((\alpha, \beta), (\beta, \gamma))^t$ be two matrices in $\mathbb{Z}_N^{2\times 2}$, where N is prime. Then if a, b, and c are nonzero, it is easy to verify that the matrix

$$\mathbf{S} = \begin{bmatrix} a & b - \frac{\beta}{c} \\ c + \frac{\beta}{b} & d - \frac{\alpha\gamma}{abc} \end{bmatrix}$$

has determinant

$$\det(\mathbf{S}) = \det(\overline{\mathbf{S}}) + \frac{1}{bc}\det(\overline{\mathbf{W}})\ . \quad (5)$$

Now, we give a sequence of games starting with the RANK game, and finishing with a game sampling matrices of rank $r + 1$ with overwhelming probability in λ, independently of the guess bit b. Below we let W_i the event that Game_i outputs 1.

Game_0 : The original $(\kappa, m, n, r, r{+}1)$-RANK problem for $r \geq \kappa$, the game is as shown in Figure 10.

Game_s **for** $1 \leq s \leq mn$: To ease notation let us index these games with (i, j), where $1 \leq i \leq m$, and $1 \leq j \leq n$. The (i, j)th game, if $b = 0$, encodes a random matrix $[\mathbf{M}]$ of rank r with a variant of **Sam** algorithm that uses the following representation vectors

$$\mathbf{x}_{i,j} = \mathbf{y}_{i,j} = \begin{cases} (m_{i,j}, 0, 0) & \text{if } (i, j) = (1, 1) \text{ or } i > 2 \text{ or } j > 2\ ; \\ (m_{i,j} - \omega, 1, 0) & \text{if } (i, j) = (1, 2) \text{ or } (i, j) = (2, 1)\ ; \\ (m_{i,j} - \omega^2, 0, 1) & \text{if } (i, j) = (2, 2)\ . \end{cases}$$

$(\kappa, m, n, r, r+1)$-RANK$^{\mathcal{A}}_{\Gamma}(\lambda)$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}(1^{\lambda}, 1^{\kappa})$
$b \leftarrow\!\!\$\ \{0,1\}$
$\mathbf{M} \leftarrow\!\!\$\ \mathrm{Rk}_{r+b}(\mathbb{Z}_N^{m\times n})$
parse $m_{i,j} \leftarrow \mathbf{M}$
for $(i,j) \in [m] \times [n]$
  $[m_{i,j}] \leftarrow\!\!\$\ \mathbf{Sam}(m_{i,j})$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [\mathbf{M}])$
return $(b' = b)$

Figure 10: The rank problem in the symmetric setting.

Here ω is the random $\mathbb{Z}_N$-value defining the public matrix $[\mathbf{W}]_0$ of pp. What changes in these games is that the representation vectors are sampled with the last two coordinates set to 0 or 1. Using a similar argument as in Claim A.2 (see the proof of Theorem 6.2) we have that

$$|\Pr[W_{s-1}(\lambda)] - \Pr[W_s(\lambda)]| \leq \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{B}}(\lambda)\ .$$

Game$_{mn+1}$: In this game, if $b = 0$, the encoded matrix $[\mathbf{M}]$ is generated as follows. Let $\overline{\mathbf{S}} = ((a,b),(c,d))^t$ be a matrix of rank 1 with $a, b, c \in \mathbb{Z}_N^*$, and let

$$\mathbf{S} = \begin{bmatrix} a & b - \frac{\omega}{c} \\ c + \frac{\omega}{b} & d - \frac{\omega^2}{abc} \end{bmatrix}$$

We use $\mathbf{S}$ to form the following matrix in $\mathbb{Z}_N^{m\times n}$

$$\overline{\mathbf{M}} = \begin{bmatrix} \mathbf{S} & & \\ & \mathbf{I}_{r-1} & \\ & & \mathbf{0} \end{bmatrix} \tag{6}$$

where $\mathbf{I}_{r-1}$ is the identity matrix of order $r-1$. Next, we obtain the encoded matrix $[\overline{\mathbf{M}}]$ as specified in the previous games, and finally we set $[\mathbf{M}] = \mathbf{L}[\overline{\mathbf{M}}]\mathbf{R}$, where $\mathbf{L}$, $\mathbf{R}$ are two random invertible matrices of dimensions $m \times m$ and $n \times n$, respectively; observe that the last step can be done using **Op** and the explicit knowledge of the invertible matrices.

We have that

$$\Pr[W_{mn}(\lambda)] = \Pr[W_{mn+1}(\lambda)]\ .$$

To see this is enough to show that $\mathbf{M}$ is randomly distributed over the set of $m \times n$ matrices of rank r.

To this end we first argue that $\overline{\mathbf{M}}$ has rank r. This follows from two observations: (1) matrix $\mathbf{S}$ has rank 1 applying Equation 5 on matrices $\overline{\mathbf{S}}$ and $\overline{\mathbf{W}} = ((1,\omega),(\omega,\omega^2))^t$; observe that $\det(\overline{\mathbf{W}}) = 0$, and (2) by construction, $\overline{\mathbf{M}}$ has rank r if $\mathbf{S}$ has rank 1.

Next we use the random self-reducibility of the rank problem. Concretely, $[\mathbf{M}]$ is distributed as in Game$_{mn}$ because the left (resp., right) action of invertible matrices $\mathrm{GL}_m(\mathbb{Z}_N)$ (resp., $\mathrm{GL}_n(\mathbb{Z}_N)$) is transitive in the set of $\mathbb{Z}_N$-matrices of dimension $m \times n$ and rank r.

Game_{mn+2} : Here (the obfuscation of) circuit C^*_{Map} included in pp has hard-coded the tuple $[\mathbf{h}]_0 = ([1]_0, [\omega]_0, \ldots [\omega^{2\kappa}]_0)$, and it uses the algorithm of Lemma 6.1 to evaluate the κ-linear map. We have that

$$|\Pr[W_{mn+1}(\lambda)] - \Pr[W_{mn+2}(\lambda)]| \leq \mathbf{Adv}^{\text{ind}}_{\mathbf{IO},\mathcal{B}}(\lambda) \,.$$

This follows from the fact that circuits C_{Map}, knowing ω in the clear, and C^*_{Map}, knowing the powers $[\omega^i]_0$, are functionally equivalent. The other crucial observation is that to generate an encoding of minor $[\mathbf{S}]$ it suffices to know $[1]_0$, $[\omega]_0$, $[\omega^2]_0$ and matrix $\overline{\mathbf{S}}$. For example, the representation vectors of $[s_{2,1}]_0 = c[1]_0 + \frac{1}{b}[\omega]_0$ are $\mathbf{x}_{1,2} = \mathbf{y}_{1,2} = (c, -b, 0)$.

Game_{mn+s+2} **for** $1 \leq s \leq 2(\kappa-1)$: In Game_{mn+2+s} we hard-code into C^*_{Map} the tuple

$$[\mathbf{h}_s]_0 = ([1]_0, [\omega]_0, \ldots, [\omega^{2\kappa-s}]_0, [\tau_{2\kappa-s+1}]_0, [\tau_{2\kappa-s+2}]_0, \ldots, [\tau_{2\kappa}]_0) \,,$$

where $\tau_{2\kappa-s+j}$ are fresh random values in $\mathbb{Z}_N$. Observe that now C^*_{Map} does not implement a κ-linear map. An attacker against $(2\kappa - s)$-SDDH can embed a challenge tuple $[\mathbf{q}]_0 = ([1]_0, [\omega]_0, \ldots, [\omega^{2\kappa-s}]_0, [\tau_{2\kappa-s+1}]_0)$ in the first $2\kappa - s + 1$ positions of $[\mathbf{h}_s]_0$. Then, if $\tau_{2\kappa-s+1} = \omega^{2\kappa-s+1}$ this simulates Game_{mn+s+1}, otherwise it simulates Game_{mn+s+2}. This shows

$$|\Pr[W_{mn+s+1}(\lambda)] - \Pr[W_{mn+s+2}(\lambda)]| \leq \mathbf{Adv}^{(2\kappa-s)\text{-sddh}}_{\Gamma_0,\mathcal{B}}(\lambda) \quad \text{for} \quad 1 \leq s \leq 2(\kappa-1) \,.$$

$\text{Game}_{mn+2\kappa+1}$: Here we hard-code $[\mathbf{h}_{2\kappa-1}]_0$ in C^*_{Map} and the minor $\mathbf{S}$ is set to

$$\mathbf{S} = \begin{bmatrix} a & b - \frac{\omega}{c} \\ c + \frac{\omega}{b} & d - \frac{\tau_2}{abc} \end{bmatrix}$$

for a random fresh value τ_2 in $\mathbb{Z}_N$ (with $[\tau_2]_0$ included in $[\mathbf{h}_{2\kappa-1}]_0$). As observed in the description of Game_{mn+2}, an encoding $[\mathbf{S}]$ (with respect to ω and τ_2) can be generated only with the knowledge of $[1]_0$, $[\omega]_0$, $[\tau_2]_0$ and matrix $\overline{\mathbf{S}}$. So we can embed a 1-SDDH challenge to simulate $\text{Game}_{mn+2\kappa}$ or $\text{Game}_{mn+2\kappa+1}$. Thus

$$|\Pr[W_{mn+2\kappa}(\lambda)] - \Pr[W_{mn+2\kappa+1}(\lambda)]| \leq \mathbf{Adv}^{1\text{-sddh}}_{\Gamma_0,\mathcal{B}}(\lambda) \,.$$

To conclude, to see that $\Pr[W_{mn+2\kappa+1}(\lambda)] \leq 1/2 + \text{negl}(\lambda)$ we apply Equation 5 again to matrices $\overline{\mathbf{S}}$ and $\overline{\mathbf{W}} = ((1, \omega), (\omega, \tau_2))$ to argue that minor $\mathbf{S}$ has rank 2 with overwhelming probability over the choice of τ_2 (concretely $\tau = \omega^2$ with probability $\text{negl}(\lambda) := 1/N(\lambda)$). Therefore $\mathbf{M}$ has rank $r + 1$ for guess bit $b = 0$ also with overwhelming probability in λ. □