Multilinear Maps from Obfuscation

Martin R. Albrecht[1(✉)], Pooya Farshim[2], Dennis Hofheinz[3], Enrique Larraia[1], and Kenneth G. Paterson[1]

[1] Royal Holloway, University of London, Egham, UK
martin.albrecht@rhul.ac.uk
[2] Queen's University Belfast, Belfast, UK
[3] Karlsruhe Institute of Technology, Karlsruhe, Germany

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

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,

E. Kushilevitz and T. Malkin (Eds.): TCC 2016-A, Part I, LNCS 9562, pp. 446–473, 2016.
DOI: 10.1007/978-3-662-49096-9_19

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 $g^{\omega^{\kappa}}$ 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 $a_i \leftarrow\!\!\$\ \mathbb{Z}_N$, 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 Sect. 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). 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 Sect. 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 $1 \leq i \leq \kappa$ 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 $\mathbf{h} = ((g_1^{a_i})_{i \leq \kappa+1}, g_T^d)$, 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 $i \leq \kappa$. 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 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.[1]

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, CLT15] and GGH15 [GGH15] proposals. 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 [GGH+13b] or dual-input straddling sets [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

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

in their own right (cf. [BS03,PTT10], resp., [FHPS13,GGH+13c,HSW13] and [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 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.[2] 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

[2] 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].

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 Background

The security parameter is denoted by $\lambda \in \mathbb{N}$. We assume that λ is an implicit input given in unary to all algorithms. Given a randomized algorithm $\mathcal{A}$ we denote the action of running $\mathcal{A}$ on inputs $(x_1, \ldots)$ with fresh random coins r and assigning the output(s) to $y_1, \ldots$ by $(y_1, \ldots) \leftarrow\!\!\$\ \mathcal{A}(x_1, \ldots; r)$, and for a finite set X, we denote 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. A real-valued function $\mu(\lambda)$ is negligible if $\mu(\lambda) \in \mathcal{O}(\lambda^{-\omega(1)})$. The set of all negligible functions is denoted by NEGL.

2.1 Homomorphic Public-Key Encryption

Scheme $\Pi := (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Eval})$ denotes a homomorphic public-key encryption (HPKE) with message space $\{0,1\}^\lambda$, where **Eval** is a *deterministic* algorithm. We require Π to be IND-CPA, perfectly correct, and compact, and also assume that the secret keys are the random coins used in key generation; this will allow to check key pairs for validity.

2.2 Obfuscators

An algorithm **Obf** is an *obfuscator* for circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if for any $m \in \{0,1\}^\lambda$, $C \in \mathcal{C}_\lambda$, and $\overline{C} \leftarrow\!\!\$\ \mathbf{Obf}(C)$ we have that $C(m) = \overline{C}(m)$. The security of **Obf** with respect a class $\mathcal{C}$ requires that no PPT adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ can distinguish the obfuscation of two circuits in $\mathcal{C}$ with noticeable probability. We will consider two notions of obfuscation depending on the class of permissible adversaries. The first notion is *functional equivalence*, whereby the two circuits any sampled circuits C_1, C_2 must satisfy $C(m) = C(m)$ for all m. We will write **IO** for obfuscator whenever this level of security is assumed. The second notion is *X-ind sampling* [CLTV15], which, roughly speaking, requires the existence of a domain subset $\mathcal{X}$ of size at most X such that the two circuits are functionally equivalent outside $\mathcal{X}$ and furthermore within $\mathcal{X}$ the outputs are indistinguishable. We will write **PIO** for this case.

2.3 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. 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. 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 for x, 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[3]; 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.

Security. We require a dual-mode NIZK to meet the following requirements. (1) binding and hiding CRS indistinguishability; (2) perfect completeness under the hiding and binding modes; (3) perfect soundness under the binding mode; (4) perfect extractability under the binding mode; (5) perfect witness-indistinguishability under the hiding mode; and (6) perfect zero-knowledge under the binding mode.

2.4 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 efficiently distinguish between $x \leftarrow\!\!\$\ L$ for $L \leftarrow\!\!\$\ \mathcal{L}_\lambda$, and $x \leftarrow\!\!\$\ U = U_\lambda$.

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 is a direct adaptation of the "cryptographic" MLG setting of [BS03] to a setting where group elements have *non-unique* encodings. In our abstraction, on top of the procedures needed for

[3] 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.).

generating, manipulating and checking group elements, we introduce an *equality-checking* procedure which generalizes that 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). We assume pp is provided to the various algorithms below.

Val$_i(h)$**:** This is the validity testing algorithm. On input (the group parameters and) a group index $1 \leq i \leq \kappa + 1$ and a string $h \in \{0,1\}^*$, it returns $b \in \{\top, \bot\}$. We define $\mathbb{G}_i$, which is also parameterized by pp, as the set of all h for which **Val**$_i(h)$ holds. We write $h \in \mathbb{G}_i$ when **Val**$_i(h)$ holds 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^κ and 1^λ, a property that we refer to as *compactness*.

Eq$_i(h_1, h_2)$**:** This is the equality testing algorithm. On input two valid group elements $h_1, h_2 \in \mathbb{G}_i$, it outputs a Boolean value $b \in \{\top, \bot\}$.[4] We require **Eq**$_i$ to define an equivalence relation. We say that the group has unique encodings if **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 **Eq**$_i(h, h') = \top$; 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 **Eq**$_i$ we will henceforth will 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 **Eq**$_i$ partitions $\mathbb{G}_i$, to be finite and equal to N_i (where N_i comes from pp). Note that equality testing algorithms **Eq**$_i$ for $1 \leq i \leq \kappa$ can be derived from one for **Eq**$_{\kappa+1}$ using the multilinear map **e** defined below, provided $N_{\kappa+1}$ is prime.

Op$_i(h_1, h_2)$**:** This algorithm will define our 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 **Op**$_i(h_1, h_2)$ for simplicity. We require that **Op**$_i$ respect the equivalence relations **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 $\mathbb{G}_i$. These 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 **Op**$_i$ and identity element 1_i.

The algorithm **Op** gives rise to an exponentiation algorithm **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 **Exp**$_i(h, z)$ by h^z and define $h^0 := 1_i$. Note that under the definition of N_i for any $h \in \mathbb{G}_i$

[4] We assume, without loss of generality, that all algorithms return $\bot$ when run on invalid group elements.

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).

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 is 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 $h'_j \in \mathbb{G}_j$

$$\mathbf{e}(h_1, \ldots, h_j h'_j, \ldots, h_\kappa) = \mathbf{e}(h_1, \ldots, h_j, \ldots, h_\kappa)\mathbf{e}(h_1, \ldots, h'_j, \ldots, h_\kappa).$$

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 $\mathbb{G}_{\kappa+1}$ not in the equivalence class of $\mathbf{1}_{\kappa+1}$. (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 are independent of the group index for $1 \leq i \leq \kappa$ and the $\mathbf{e}$ algorithm is invariant under permutations of its inputs. That is for any permutation $\pi : [\kappa] \longrightarrow [\kappa]$ we have

$$\mathbf{e}(h_1, \ldots, h_\kappa) = \mathbf{e}(h_{\pi(1)}, \ldots, h_{\pi(\kappa)}).$$

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 $\mathbf{1}_T$ and g_T in place of $\mathbf{1}_{\kappa+1}$ 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 $\mathbf{Sam}_i(a)$.

In some applications, we also rely on the following algorithm, which provides a canonical string for all group elements within an equivalence class.

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

$\mathbf{Ext}_i(h)$: This is the extraction algorithm. On input $h \in \mathbb{G}_i$ it outputs a string $s \in \{0,1\}^{p(\lambda)}$ where $p(\cdot)$ denotes a polynomial function. 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 the distribution of $\mathbf{Ext}_i([z]_i)$ is uniform over $\{0,1\}^{p(\lambda)}$, for $[z]_i \leftarrow\!\!\$\ \mathbf{Sam}_i(\varepsilon)$. For groups with unique encodings this algorithm trivially exists.

In the full version of the paper we provide possible extensions to this syntax.

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 a $\mathbf{e}_{i,j}$ algorithm that takes inputs from $\mathbb{G}_i$ and $\mathbb{G}_j$ and returns an element in $\mathbb{G}_{i+j}$ such that the result is linear in each input. 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.

4 The Construction

We now present our construction of an MLG scheme Γ according to the syntax introduced in Sect. 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 $\mathbf{Obf}$. (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 TDwhich has a hard subset membership problem, and such that all TDhave efficiently computable witness relations with unique witnesses.[6] (See Sect. 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.

[6] 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.

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 $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_k)^t \in \mathbb{Z}_N^{\kappa \times \ell}$ 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 Sect. 4.2. It also constructs two obfuscated circuits $\overline{C}_{\mathrm{Map}}$ and $\overline{C}_{\mathrm{Add}}$ which we will describe in Sects. 4.3 and 4.4. For $1 \le i \le \kappa$, the identity elements $\mathbf{1}_i$ and group generators g_i are sampled using $\mathbf{Sam}_i(0)$ and $\mathbf{Sam}_i(x_i)$ respectively for algorithm $\mathbf{Sam}_i$ described in Sect. 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 $\mathbf{1}_{\kappa+1} = \mathbf{1}_0$ and $g_{\kappa+1} = g_0$. The scheme parameters are

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

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 $\mathbf{Enc}$ 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 $y \in \mathsf{TD}$.

For $1 \leq i \leq \kappa$, 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 $1 \leq i \leq \kappa$ 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 Fig. 1 (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 $\overline{C}_{\mathrm{Add}}$, the obfuscator directly deals with the needed randomness.[7] 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

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

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. (only h' is invalid) repeat 5a with h'

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

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 $[z_{\kappa+1}]_{\kappa+1}$

Fig. 1. 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 $gpk = (pp_0, pk_1, pk_2, [\mathbf{W}]_0, \mathsf{TD}, y)$.

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 Sect. 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 $\overline{C}_{\text{Map}}$ of the circuit implementing this operation (see Fig. 1, bottom) will be made available through the public parameters and $\mathbf{e}$ is defined to run this circuit on its inputs.

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 Sect. 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.

5 Indistinguishability of Encodings

In this section we will state two theorems that are essential tools in establishing the intractability of the κ-MDDH for our MLG scheme Γ constructed in Sect. 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 Fig. 2. 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

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

Fig. 2. 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.

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 Sect. 2; 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 Sect. 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 $\overline{C}_{\text{Add}}$ 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 Sect. 4). The next lemma, roughly speaking, says that *provided* parameters pp include an instance $y \in \mathsf{TD}$, then there exists an alternative implementation $\widehat{C}_{\text{Add}}$ that does not use the secret keys, and whose obfuscation is indistinguishable to that of C_{Add} of Fig. 1 (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. A formal proof is in the full version, we give an overview of the proof below.

Lemma 1. *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 Sect. 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* $\widehat{C}_{\text{Add}}$ *of Fig. 3. Then, for any* PPT *adversary* $\mathcal{A}$*,*

$$Pr[\mathcal{A}(\widehat{pp}, sk_1, sk_2) = 1 \ :\ (sk_1, sk_2) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)]$$
$$- Pr[\mathcal{A}(\widehat{pp}, sk_1, sk_2) = 1 \ :\ (sk_1, sk_2) \leftarrow\!\!\$\ \mathbf{Gen}(1^\lambda)] \ \in\ \text{NEGL}.$$

CIRCUIT $\widehat{C}_{\text{Add}}[gpk, crs, w_y; r](i, h, h')$:
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. $\pi'' \leftarrow \mathbf{Prove}(gpk, crs, ([z'']_0, \mathbf{c}''_1, \mathbf{c}''_2), w_y; r)$
6. return $([z''], \mathbf{c}''_1, \mathbf{c}''_2, \pi'')$

Fig. 3. Alternative circuit for addition of group elements. Recall that here $\widehat{pp}$ 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 $\widehat{crs}'$). The circuit uses (the) witness w_y to $\widetilde{y} \in \mathsf{TD}$ to produce π''.

Proof (Sketch). 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 $\widehat{C}_{\text{Add}}$ can use w_y instead of sk_1, sk_2 to produce the output proof π''. Below we provide brief descriptions of the transformation from C_{Add} to $\widehat{C}_{\text{Add}}$, as well as some intuition for the justifications of each step.

Game_0: We start with (a PIO obfuscation of) circuit C_{Add} of Fig. 1 and with $\widehat{pp}$ including $\widetilde{y} \in \mathsf{TD}$ and a binding crs'.

Game_1: The circuit has witness w_y to $\widetilde{y} \in \mathsf{TD}$ hardcoded. If some input reaches the "invalid" branches (steps 5a or 5b of C_{Add}; see Fig. 1 (top)), C_{Add} does not extract a witness from the corresponding proof, but instead uses w_y to generate proof π''. Since the witness w_y is unique, and the CRS crs' guarantees perfect soundness, this leads to exactly the same behavior of C_{Add} in Game 0. Hence, this hop is justified by PIO. Note that Game 1 requires no extraction trapdoor td_{ext} anymore.

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

Game_3: Here, output proofs π'' for those inputs entering the "valid" branch (step 5c; see Fig. 1) 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. This hop is justified by PIO, where the perfect witness indistinguishability of $\widehat{crs}'$ (when constructed as a hiding CRS) guarantees that the C_{Add} circuits in Games 2 and 3 have identically distributed outputs.

With the above lemma we can invoke IND-CPA security, and via a sequence of games obtain the result stated below. The proof can be found in the full version; here we give a high-level overview of the proof (see also Fig. 4).

Theorem 1 (Switching encodings using PIO). *Let Γ be the MLG scheme constructed in Sect. 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 $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{A}}(\lambda) \in \text{NEGL}.$$

Proof (Sketch). The strategy of the proof is as follows. We start replacing parameters pp as described in Sect. 4 with parameters $\widetilde{pp}$ of Lemma 1, the latter include an instance $\widetilde{y} \in \mathsf{TD}$, this hop is justified by the hardness of deciding membership in TD; then we apply Lemma 1 to replace parameters $\widetilde{pp}$ with $\widehat{pp}$, including an obfuscation of circuit $\widehat{C}_{\mathrm{Add}}$ of Fig. 3; at this point we invoke the IND-CPA security of the encryption scheme to change the representation vector encrypted under pk_2 of the challenge encoding (the challenge proof π^* is generated using simulator trapdoor td_{zk}, and hence is identically distributed to a real proof); next, we revert back to parameters pp, including a no-instance $y \notin \mathsf{TD}$ and an obfuscation of circuit C_{Add} of Fig. 1, which is justified again by the hardness of $\mathsf{TD}$and Lemma 1; note that now it is possible to use sk_2 in C_{Map}, instead of sk_1, invoking the security of **PIO** (functional equivalence follows from the perfect soundness of the NIZK with a binding CRS); last, we repeat the same steps to change the representation vector encrypted under pk_1. This completes the proof. (See Fig. 4 for a sketch of the hybrids.)

5.2 Doing Without Probabilistic Obfuscation

In contrast to the PIO-based approach from Sect. 5.1, we can also 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 (when the CRS is generated using **BCRS**), perfect WI (when the CRS is generated by **HCRS**), and meets a structural requirement we explain below. This requirement is fulfilled by Groth–Sahai proofs [GS08] based on the DDH or k-Linear assumption.

A STRUCTURAL PROPERTY. To explain the required structural property, recall first that perfect WI guarantees that proofs that are honestly generated (under a hiding CRS) have a distribution that is independent of the used witness. For our purposes, we require a slightly more specialized property: we require that a change of the used witness (in **Prove**) can be compensated with a change of random coins. In other words, we require that for every hiding CRS crs, and for every statement x and pair of witnesses w, w' for x, there is a value Δ such that

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

where "+" is a suitable homomorphic operation on random coins. Note that Δ may depend on w and w', but not on r. Furthermore, we require that Δ can be efficiently computed from x, w, w', and the zero-knowledge CRS trapdoor td_{zk} output by **HCRS**.

Again, we stress that Groth–Sahai proofs have the desired property (when restricting to statements with witnesses $w \in \{0,1\}^*$ that are bit strings). We give more details in the full version of this paper.

THE DETERMINISTIC CIRCUIT C_{Add}. We now comment on a necessary slight tweak to the multilinear map construction itself. Namely, we have to view both C_{Add} and C_{Map} as deterministic circuits (so they can be obfuscated using an

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 1
3	$\widehat{pp}$	w_y	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$\boxed{(\mathbf{x}_1, \mathbf{y}_1)}$	IND-CPA
4	$\boxed{\widetilde{pp}}$	$\boxed{sk_1, sk_2, td_{ext}}$	sk_1	$(\mathbf{x}_0, \mathbf{y}_0)$	$(\mathbf{x}_1, \mathbf{y}_1)$	Lemma 1
5	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 1
9	$\widehat{pp}$	w_y	sk_2	$\boxed{(\mathbf{x}_1, \mathbf{y}_1)}$	$(\mathbf{x}_1, \mathbf{y}_1)$	IND-CPA
10	$\boxed{\widetilde{pp}}$	$\boxed{sk_1, sk_2, td_{ext}}$	sk_2	$(\mathbf{x}_1, \mathbf{y}_1)$	$(\mathbf{x}_1, \mathbf{y}_1)$	Lemma 1
11	$\boxed{pp}$	sk_1, sk_2, td_{ext}	sk_2	$(\mathbf{x}_1, \mathbf{y}_1)$	$(\mathbf{x}_1, \mathbf{y}_1)$	TD indist.
12	pp	sk_1, sk_2, td_{ext}	$\boxed{sk_1}$	$(\mathbf{x}_1, \mathbf{y}_1)$	$(\mathbf{x}_1, \mathbf{y}_1)$	PIO

Fig. 4. Outline of the proof steps of Theorem 1. b is the random bit of the κ-Switch game (see Fig. 2). 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 1. The hops relying on PIO use the perfect soundness under binding crs' to argue function equivalence.

indistinguishability obfuscator **IO**). For C_{Map}, this is trivial, since it already is deterministic. Furthermore, we can view C_{Add} as a deterministic circuit that takes as input (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 hard-wired 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* (upon creation time) into C_{Add}. When we want to make the choice of R explicit, we also write C^R_{Add}.

For this slight variation of our construction, we claim:

Theorem 2 (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 Sect. 4 obtained using these primitives. Then, for any* PPT *adversary* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma,\mathcal{A}}(\lambda) \in \text{NEGL}.$$

Here, we only give a brief intuition for the proof. A more detailed proof is given in the full version.

In a nutshell, the proof of Theorem 2 proceeds like that of Theorem 1, 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 1, and in the hops from Game_5 to Game_6 and from Game_{11} to Game $_{12}$ in the proof of Theorem 1. Since the corresponding deterministic circuits are functionally equivalent (in case of $C_{\mathrm{Add}} = C^R_{\mathrm{Add}}$: when the same value of R is used), the security of **IO** can be directly utilized.

The second type of steps lets C_{Add} use a different witness (e.g., w_y instead of (sk_1, sk_2), or vice versa) to generate consistency proofs π''. This type of proof step occurs in the hop from Game_2 to Game_3 in the proof of Lemma 1. Note that at this point, the generated CRS is hiding, and $C_{\mathrm{Add}} = C^R_{\mathrm{Add}}$ uses a single hardcoded random string R as random coins to generate such proofs. By property $(\star)$ above, we have that

$$C^R_{\mathrm{Add},1} \;\equiv\; C^{R+\Delta}_{\mathrm{Add},2},$$

where $C_{\mathrm{Add},1}$ and $C_{\mathrm{Add},2}$ denote the C_{Add} variants before and after the step, and Δ denotes the randomness shift value from $(\star)$.

Hence, this change can be justified with a reduction to the (deterministic) indistinguishability property of **IO**. Specifically, a suitable circuit sampler would sample circuits $C_1 := C^R_{\mathrm{Add},1}$ and $C_2 := C^{R+\Delta}_{\mathrm{Add},2}$ for a uniform R, and a Δ generated from the corresponding witnesses. (We note that *during* this reduction, we can of course assume both relevant witnesses (sk_1, sk_2) and w_y to be known.)

The remaining parts of the proof of Theorem 2 (including the proof of Lemma 1) apply unchanged.

6 The Multilinear DDH Problem

In the full version we show that natural multilinear analogues of the decisional Diffie–Hellman (DDH) problem are hard for our MLG scheme Γ from Sect. 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 Sect. 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 Sect. 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 Fig. 5 (left).

$\underline{\mathrm{DDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)}$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}_0(1^\lambda, 1^0)$
$b \leftarrow\!\!\$\ \{0,1\}$
$x, y, z \leftarrow\!\!\$\ \mathbb{Z}_N$
if $b = 1$ then
  $z \leftarrow x \cdot y$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [x]_0, [y]_0, [z]_0)$
Return $(b = b')$

$\underline{q\text{-SDDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)}$:
$pp \leftarrow\!\!\$\ \mathbf{Setup}_0(1^\lambda, 1^0)$
$q \leftarrow q(\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$
$x, z \leftarrow\!\!\$\ \mathbb{Z}_N$
if $b = 1$ then
  $z \leftarrow x^{q+1}$
$b' \leftarrow\!\!\$\ \mathcal{A}(pp, [x]_0, \ldots, [x^q]_0, [z]_0)$
Return $(b = b')$

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

Fig. 5. 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.

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

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

where game $q\text{-SDDH}^{\mathcal{A}}_{\Gamma_0}(\lambda)$ is shown in Fig. 5 (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 Fig. 5 (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 Sect. 4 which gives rise to a *symmetric* MLG scheme as defined in Sect. 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 2 (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 $\mathcal{O}(\kappa^\ell)$.

A formal statement and proof of the following result is in the full version of the paper, here we give a high level overview. Below $I = I^*$ denotes the index set with all the second components being 1.

Theorem 3 ($(\kappa-1)$-SDDH hard $\implies$ symmetric (κ, I^*)-MDDH hard). *Let* Γ^* *denote scheme* Γ *of Sect. 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* $\mathcal{B}_1$, $\mathcal{B}_2$ *of essentially the same complexity as* $\mathcal{A}$ *such that*

$$\mathbf{Adv}^{(\kappa,I^*)\text{-mddh}}_{\Gamma^*,\mathcal{A}}(\lambda) \leq\ 2 \cdot \mathbf{Adv}^{(\kappa-1)\text{-sddh}}_{\Gamma_0,\mathcal{B}_1}(\lambda) + (\kappa+1) \cdot \mathbf{Adv}^{\kappa\text{-switch}}_{\Gamma^*,\mathcal{B}_2}(\lambda) + \mu(\lambda),$$

for all $\lambda \in \mathbb{N}$ *and a suitable negligible function* μ.

Proof (Sketch). 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 2 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 (Fig. 5, 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 $[a_{\kappa+1}]_1$.)

$\text{Game}_{\kappa+1}$: 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.

$\text{Game}_{\kappa+2}$: 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.

$\text{Game}_{\kappa+3}$: 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}.

$\text{Game}_{\kappa+4}$: 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.

$\text{Game}_{\kappa+5}$: This game introduces a conceptual change analogous to that in $\text{Game}_{\kappa+1}$ 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.

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

In $\text{Game}_{\kappa+6}$, 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 Sect. 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 $I = \{(i, I(i)) : 1 \le i \le \kappa + 1\}$ 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 Sect. 5. We present the proof intuition here and leave the details to the full version.

Theorem 4 (DDH hard $\implies$ asymmetric (κ, I^*)-MDDH hard). *Let Γ^* denote scheme Γ of Sect. 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 $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

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

for a all $\lambda \in \mathbb{N}$ and suitable negligible function μ.

Proof (Sketch). 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.

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

(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 $(b = b')$

Fig. 6. The RANK problem parameterized by integers κ, m, n, r_0 and r_1.

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 Sect. 4 follows from that of the q-SDDH problem for Γ_0.

7.1 Formalization of the Problem

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

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

where game (κ, m, n, r_0, r_1)-RANK$^{\mathcal{A}}_{\Gamma}(\lambda)$ is shown in Fig. 6.

In the presence of a κ-linear map the (κ, m, n, r_0, r_1)-RANK$^{\mathcal{A}}_{\Gamma}(\lambda)$ 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 problems [BHHO08, NS09].

7.2 The RANK Problem with Our MLG Scheme

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 Sect. 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 $m_{i,j} \in \mathbb{Z}_N$.

We show that for our construction in Sect. 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 proof is in the full version of the paper, here we give only give some intuition.

Theorem 5 (SDDH $\implies$ RANK)**.** *Let Γ denote scheme Γ of Sect. 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 $r \geq \kappa$. 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 $\lambda \in \mathbb{N}$ and a suitable negligible function μ*

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

7.3 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 Sect. 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 dependent 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 2 (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] Applebaum, B., Brakerski, Z.: Obfuscating circuits via composite-order graded encoding. In: Dodis and Nielsen [DN15], pp. 528–556

[BBS04] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)

[BHHO08] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)

[BLMR13] Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti and Garay [CG13a], pp. 410–428

[BLR+15] Boneh, D., Lewi, K., Raykova, M., Sahai, A., Zhandry, M., Zimmerman, J.: Semantically secure order-revealing encryption: multi-input functional encryption without obfuscation. In: Oswald and Fischlin [OF15], pp. 563–594

[BS03] Boneh, D., Silverberg, A.: Applications of multilinear forms to cryptography. Contemp. Math. **324**, 71–90 (2003)

[BWZ14] Boneh, D., Waters, B., Zhandry, M.: Low overhead broadcast encryption from multilinear maps. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 206–223. Springer, Heidelberg (2014)

[CG13a] Canetti, R., Garay, J.A. (eds.): CRYPTO 2013, Part I. LNCS, vol. 8042. Springer, Heidelberg (2013)

[CG13b] Canetti, R., Garay, J.A. (eds.): CRYPTO 2013, Part II. LNCS, vol. 8043. Springer, Heidelberg (2013)

[CGH+15] Coron, J.-S., Gentry, C., Halevi, S., Lepoint, T., Maji, H.K., Miles, E., Raykova, M., Sahai, A., Tibouchi, M.: Zeroizing without low-level zeroes: new MMAP attacks and their limitations. In: Gennaro and Robshaw [GR15], pp. 247–266

[CHL+15] Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (2015)

[CLR15] Cheon, J.H., Lee, C., Ryu, H.: Cryptanalysis of the new CLT multilinear maps. Cryptology ePrint Archive, Report 2015/934 (2015). http://eprint.iacr.org/

[CLT13] Coron, J.-S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Canetti and Garay [CG13a], pp. 476–493

[CLT15] Coron, J.-S., Lepoint, T., Tibouchi, M.: New multilinear maps over the integers. In: Gennaro and Robshaw [GR15], pp. 267–286

[CLTV15] Canetti, R., Lin, H., Tessaro, S., Vaikuntanathan, V.: Obfuscation of probabilistic circuits and applications. In: Dodis and Nielsen [DN15], pp. 468–497

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

[DN15] Dodis, Y., Nielsen, J.B. (eds.): TCC 2015, Part II. LNCS, vol. 9015. Springer, Heidelberg (2015)

[EHK+13] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti and Garay [CG13b], pp. 129–147

[FHPS13] Freire, E.S.V., Hofheinz, D., Paterson, K.G., Striecks, C.: Programmable hash functions in the multilinear setting. In: Canetti and Garay [CG13a], pp. 513–530

[GGH13a] Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)

[GGH+13b] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press, October 2013

[GGH+13c] Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti and Garay [CG13b], pp. 479–499

[GGH15] Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. In: Dodis and Nielsen [DN15], pp. 498–527

[GGSW13] Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 467–476. ACM Press, June 2013

[GHV12] Galindo, D., Herranz, J., Villar, J.: Identity-based encryption with master key-dependent message security and leakage-resilience. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 627–642. Springer, Heidelberg (2012)

[GR15] Gennaro, R., Robshaw, M. (eds.): CRYPTO 2015, Part I. LNCS, vol. 9215. Springer, Heidelberg (2015)

[GS08] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

[HJ15] Hu, Y., Jia, H.: Cryptanalysis of GGH map. Cryptology ePrint Archive, Report 2015/301 (2015). http://eprint.iacr.org/2015/301

[HSW13] Hohenberger, S., Sahai, A., Waters, B.: Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In: Canetti and Garay [CG13a], pp. 494–512

[MF15] Minaud, B., Fouque, P.-A.: Cryptanalysis of the new multilinear map over the integers. Cryptology ePrint Archive, Report 2015/941 (2015). http://eprint.iacr.org/

[NS09] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC, pp. 427–437. ACM Press, May 1990

[OF15] Oswald, E., Fischlin, M. (eds.): EUROCRYPT 2015, Part II. LNCS, vol. 9057. Springer, Heidelberg (2015)

[PTT10] Papamanthou, C., Tamassia, R., Triandopoulos, N.: Optimal authenticated data structures with multilinear forms. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 246–264. Springer, Heidelberg (2010)

[TLL14] Tang, F., Li, H., Liang, B.: Attribute-based signatures for circuits from multilinear maps. In: Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S.M. (eds.) ISC 2014. LNCS, vol. 8783, pp. 54–71. Springer, Heidelberg (2014)

[YYHK14] Yamakawa, T., Yamada, S., Hanaoka, G., Kunihiro, N.: Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 90–107. Springer, Heidelberg (2014)

[YYHK15] Yamakawa, T., Yamada, S., Hanaoka, G., Kunihiro, N.: 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] Zimmerman, J.: How to obfuscate programs directly. In: Oswald and Fischlin [OF15], pp. 439–467