CRYPTREC-EX-2603-2016

Cryptographic Multilinear Maps

A Status Report

Mehdi Tibouchi
NTT Secure Platform Laboratories

January 2017

Executive Summary

A few years ago, Garg, Gentry and Halevi (EUROCRYPT 2013) proposed the first candidate construction of *cryptographic multilinear maps*, a primitive first envisioned a decade earlier by Boneh and Silverberg (Contemp. Math. 2003) as a higher-dimensional generalization of bilinear pairings on elliptic curves.

Boneh and Silverberg themselves had pointed out several interesting applications of multilinear maps, such as one-round multiparty key agreement, verifiable pseudorandom functions and efficient broadcast encryption. Furthermore, following the construction of Garg et al., a flurry of new research uncovered even more far-reaching applications, including long-awaited primitives like attribute-based encryption for all circuits and general functional encryption, fruitful new ideas like witness encryption, and the startlingly powerful notion of *indistinguishability obfuscation.*

However, the candidate construction of Garg et al. was not provably secure. As a result, part of the new research focused on clarifying its security, and on exploring alternate techniques to achieve multilinear maps.

This document aims at giving a bird's eye view of the main results so far, in terms of new definitions, candidate constructions and major applications, and to summarize known attacks against existing schemes, discussing their current status as far as security is concerned.

This is a very active and rapidly evolving area of research, so we cannot even come close to an exhaustive survey of existing literature, and although we have strived to take into account some of the most recent published results as of late 2016, significant shifts in our understanding of multilinear maps in the near future are not only impossible to rule out but even likely to occur. Indeed, it has happened on several occasions already that a newly proposed scheme has been broken, fixed and broken again within the span of a few weeks.

With those caveats, here are some notable takeaways from the state of the art at this point in time:

- There are three main constructions proposed for multilinear maps: the original one from Garg, Gentry and Halevi (GGH13), a variant "over the integers" due to

Coron, Lepoint and Tibouchi (CLT13), and a "graph-induced" construction by Gentry, Gorbunov and Halevi (GGH15).

- Although these constructions are conceptually inspired by fully homomorphic encryption schemes that can be proved secure under well-understood hardness assumptions, the multilinear map schemes themselves have no proof of security. (Note that the same is true for bilinear pairings as well).

- Over each of the three constructions, *there exists a polynomial time attack against the basic Diffie–Hellman key multiparty exchange protocol.* In fact, the conceptual counterpart of the CDH assumption fails to hold. As a result, most of the (stronger) assumptions used to prove the existence of more interesting cryptographic notions like witness encryption and indistinguishability obfuscation also fail to hold.

- However, this does not necessarily translate to a direct attack against the actual instantiations of the primitives themselves. For indistinguishability obfuscation, in particular, attacks are known against some instantiations, but countermeasures have been proposed to circumvent them. Thus, *there are constructions of indistinguishability obfuscation* over GGH13, CLT13 and GGH15 *against which no attack is known* at the present time. Whether this will continue to hold is difficult to predict.

- Theoretically speaking, and assuming standard cryptographic hardness assumptions, it is known that indistinguishability obfuscation and secure functional encryption are essentially equivalent, and imply the existence of secure n-linear maps for polynomially large n (both in the original sense of Boneh and Silverberg and in the sense of graded encodings, as introduced by Garg et al.). This means that any alternate method to construct indistinguishability obfuscation or functional encryption would indirectly yield secure multilinear maps. Unfortunately, no such method is known at present.

- Conversely, it has also been shown that 5-linear maps for which the (subexponential) DDH assumption holds are sufficient to obtain indistinguishability obfuscation. This means that one can bootstrap constant-degree multilinear maps to arbitrary polynomial degree, and also that we seem to be tantalizingly close to achieving indistinguishability obfuscation (and hence everything else) from bilinear pairings, a primitive that we are much more confident does exist. Closing the gap from degree 5 to degree 2, however, appears to be an elusive problem.

- There is no prospect of achieving practical levels of efficiency for any of the primitives considered in this document in the foreseeable future.

Contents

Chapter 1

Introduction

1.1 From Diffie–Hellman to multilinear maps

"We stand today on the brink of a revolution in cryptography," wrote Diffie and Hellman in their seminal *New Directions* paper from 1976 [DH76], which introduced the main ideas of public-key cryptography. In particular, they described the well-known key exchange protocol which bears their name: Alice and Bob can derive a common secret by exchanging messages publicly on an insecure channel. To do so, they agree on a group $\mathbb{G}$ (say a cyclic subgroup of large prime order q in the multiplicative group $\mathbb{F}_p^*$ of a finite field $\mathbb{F}_p$) and a generator g of $\mathbb{G}$. Then Alice and Bob choose random exponents $a, b \in \{0, \dots, q-1\}$, and compute the group elements

$$A = g^a \quad \text{and} \quad B = g^b$$

respectively. Alice sends A to Bob and Bob B to Alice, and they can then both compute the common group element $g^{ab} = A^b = B^a$. However, the problem of distinguishing g^{ab} from a random element of $\mathbb{G}$ given g, g^a and g^b is believed to be hard (for the group $\mathbb{G}$ mentioned above, and many other groups like suitably chosen elliptic curves). As a result, an eavesdropper learns no information about the common secret by intercepting the communication between the two parties.

As we well know, that idea, and the corresponding Decisional Diffie–Hellman (DDH) hardness assumption, proved extremely fruitful. It can be used to construct semantically secure homomorphic encryption [ElG85], digital signatures [Sch91], efficient pseudorandom functions [NR04], CCA-secure encryption [CS03] and more. And it is cited as one of the main reason for Diffie and Hellman's Turing award.

Nevertheless, some cryptographic primitives cannot be constructed from DDH. For example, Papakonstantinou et al. were able to obtain a black-box separation result [PRV12] between DDH and identity-based encryption (IBE). To construct IBE, a more powerful setting is necessary, and that setting emerged in the early 2000s, bringing about what would be fair to call a second "revolution in cryptography": the era of bilinear pairings.

The existence of efficiently computable bilinear pairings between certain families of elliptic curve groups was first understood as a cryptanalytic liability [MVO93], but Joux noticed that it could be used constructively to generalize Diffie–Hellman key exchange to three parties in one round [Jou04]. Indeed, if $\mathbb{G}$ is a cyclic group of prime order q endowed with a symmetric non degenerate bilinear pairing $e\colon \mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, Alice, Bob and Charlie can use it to derive a common secret as follows. They choose $a,b,c\in\{0,\dots,q-1\}$ at random and compute

$$A = g^a, \quad B = g^b \quad \text{and} \quad C = g^c$$

respectively, for some agreed upon generator g. They can then all compute the common value

$$e(g,g)^{abc} = e(A,B)^c = e(B,C)^a = e(C,A)^b$$

but an eavesdropper seeing A, B and C cannot distinguish that value from a random element of $\mathbb{G}_T$, assuming the hardness of the *decisional bilinear Diffie–Hellman* (DBDH) problem, which is believed to be hard over well-constructed pairing-friendly elliptic curves.

Using this new bilinear structure, Boneh and Franklin were then able to construct the first IBE scheme [BF03], opening up the path to numerous new cryptographic notions, including public-key encryption with keyword search [BDOP04], attribute-based encryption (for boolean formulas) [GPSW06a] and homomorphic encryption for quadratic polynomials [BGN05]. It also led to more efficient constructions of previous primitives such as signatures [BLS04], group signatures [BBS04], non-interactive zero-knowledge proofs [GS08] and more. In short, the possibility offered by bilinear pairings to carry out not only linear operations in the exponent of group elements but also *one* level of multiplication proved to be particularly fecund. It also earned Joux, Boneh and Franklin the 2013 Gödel prize.

Soon after the cryptographic community realized the power of bilinear maps, Boneh and Silverberg [BS03] asked the natural question of whether this development could be pursued further, in such a way that *several* levels of multiplications could be carried out in the exponent of group elements. This would be possible using what they called *cryptographic multilinear maps*.

1.2 Multilinear maps from geometry?

For cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q, a map $e\colon \mathbb{G}^n\to\mathbb{G}_T$ is said to be a (symmetric) *n-linear map* (or just a *multilinear map* when n is omitted) if for any $a_1,\dots,a_n\in\mathbb{Z}$ and $g_1,\dots,g_n\in\mathbb{G}$, we have

$$e(g_1^{a_1},\dots,g_n^{a_n}) = e(g_1,\dots,g_n)^{a_1\cdots a_n},$$

and furthermore e is non-degenerate in the sense that $e(g,\dots,g)$ is a generator of $\mathbb{G}_T$ for any generator g of $\mathbb{G}$. For such a structure to be of cryptographic interest, one needs to be able to compute efficiently with it (in the sense that e itself and the group operations on $\mathbb{G}$ and $\mathbb{G}_T$ are efficiently computable), and it needs to satisfy some notion of security—the most basic of which would be to ask that the discrete logarithm problem in $\mathbb{G}$ be hard (which implies that it is hard in $\mathbb{G}_T$ as well). This is in essence how Boneh and Silverberg define cryptographic multilinear maps [BS03].

They observed that if one can construct such cryptographic multilinear maps (satisfying slightly stronger security notions that the basic discrete log one), a number of interesting cryptographic consequences follow, beyond what can be done with bilinear pairings. In particular, using an n-linear maps, one can obtain a one-round Diffie–Hellman-like key exchange protocol between $n+1$ parties, as a direct generalization of Joux's protocol. Indeed, if users $U_0, \dots, U_n$ want to derive a common secret, they can simply pick random exponents $a_0, \dots, a_n$, compute the group elements $A_i = g^{a_i} \in \mathbb{G}$ and broadcast them. They are then able to compute the common value $e(g, \dots, g)^{a_0 \cdots a_n}$: user U_j can obtain it as $e(A_{j+1}, \dots, A_n, A_0, \dots, A_{j-1})^{a_j}$. However, under the obvious generalization of the decisional Diffie–Hellman assumption, that value is indistinguishable from a random element of $\mathbb{G}_T$ given only the A_i's, making the protocol secure against eavesdroppers.

Other applications mentioned by Boneh and Silverberg include efficient unique signatures and broadcast encryption with short keys and optimal communication complexity. It turns out that multilinear maps also imply much stronger cryptographic notions, including indistinguishability obfuscation (see §1.4 below).

So do these multilinear maps exist? The question is especially natural in view of the fact that bilinear pairings on elliptic curves are a special case of a type of multilinear structure that exists on very large classes of algebraic geometric objects. Roughly speaking, a geometric object (say a project algebraic variety) gives rises to certain groups called "cohomology groups," together with multilinear maps between them known as cup-products. An object of dimension d has cohomology groups of degrees 0 to $2d$ and degrees add up in cup-products, so one could in principle construct a $2d$-linear map from degree 1 to degree $2d$ from any d-dimensional object; in fact, elliptic curve pairings are essentially of that form. However, it is unclear in general how to compute on those groups efficiently (or what the suitable analogue of pairing-friendly elliptic curves would be).

Boneh and Silverberg carried out a detailed analysis of the most direct generalization of elliptic curves to higher dimensions, namely abelian varieties. As for elliptic curves, their set of points is endowed with an efficiently computable group law, and that group is isomorphic to degree 1 cohomology, so that one can actually compute inside that cohomology. This makes it possible to define multilinear maps in various ways. Unfortunately, Boneh and Silverberg found that, unlike what happens with elliptic curves, the *target group* of those multilinear maps does not appear to lend itself to efficient arithmetic operations: what one gets is essentially a higher tensor power of the multiplicative group. For example, over the finite field $\mathbb{F}_p$, the target group is essentially $\mathbb{F}_p^*$, except that g^a is represented as the tuple $(g^{a_1}, \dots, g^{a_d}) \in (\mathbb{F}_p^*)^d$ for any $(a_1, \dots, a_d)$ such that $a = a_1 \cdots a_d$. Clearly, one cannot even efficiently decide equality in that group without breaking the computational Diffie–Hellman problem.

More generally, their paper shows that, under widely believed assumptions, it is impossible to construct n-linear maps from geometry whose target group is $\mathbb{F}_p^*$ itself (as opposed to a higher tensor power, say) for any $n > 2$. This does not entirely rule out multilinear maps from geometry (e.g. one could still conceivably have multilinear maps whose target group would lie in an elliptic curve or some other group with efficient arithmetic), but makes it implausible enough that the problem has only been revisited on a handful of occasions since then [RH09]. In any case, after Boneh and Silverberg's paper, constructing multilinear maps was considered intractable for at least a decade.

1.3 Fully homomorphic encryption and graded encoding schemes

New ideas to tackle the problem of obtaining multilinear maps only came about after a third "revolution" swept the world of cryptography, mainly from the mid-2000s onwards: lattice-based cryptography, ultimately leading to the construction by Gentry of a *fully-homomorphic encryption scheme* [Gen09], which solved a major, 30-year old open problem [RAD78].

Before Gentry, some encryption schemes like those of ElGamal and Paillier [ElG85, Pai99] had made it possible to carry out *either* additions *or* multiplications on ciphertexts. The pairing-based scheme of Boneh, Goh and Nissim [BGN05] supported arbitrarily many additions and *one* level of multiplications. In contrast, fully-homomorphic encryption (FHE) makes it possible to carry out *both* additions and multiplications on ciphertexts, arbitrarily many times (and as a result, any efficient function can be evaluated homomorphically on ciphertexts).

A few years later, this led to the intuition that FHE ciphertexts behave a bit like the exponents of group elements in a multilinear map. More precisely, they behave similarly to the exponents of group elements in what Garg, Gentry and Halevi call a *graded encoding scheme* [GGH13a]. Roughly speaking, such a scheme is a family of efficient cyclic groups $\mathbb{G}_0, \dots, \mathbb{G}_n$ of the same prime order q together with efficient non-degenerate bilinear pairings $e\colon \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j}$ whenever $i + j \leq n$. In other words, if we fix a family of generators g_i of the $\mathbb{G}_i$'s in such a way that $g_{i+j} = e(g_i, g_j)$, we can *add* exponents within a given group $\mathbb{G}_i$:

$$g_i^a \cdot g_i^b = g_i^{a+b}$$

and *multiply* exponents from two groups $\mathbb{G}_i, \mathbb{G}_j$ as long as $i + j \leq n$:

$$e(g_i^a, g_j^b) = g_{i+j}^{a \cdot b}.$$

This makes g_1^a somewhat similar to an "FHE encryption" of a.

Of course, there are a number of differences. First, FHE ciphertexts should be randomized. This is not a serious difficulty: one can allow for randomized representations of group elements as well, and such representations are in fact permitted in Garg et al.'s definition of a graded encoding scheme. However, one should still make it possible to test the equality of two (randomized representations of) group elements in $\mathbb{G}_n$, say; this cannot be done publicly in an FHE scheme, as it would break semantic security. Nevertheless, this may be doable once some limited information about the FHE secret key is made public. Finally, a third difference is that one should not be able to invert the bilinear pairings, so the representations of g_i^a and g_j^a cannot be of the same form when $i \neq j$. This can be dealt with by introducing some secret multiplicative factor in ciphertexts that will appear at the power i in the ciphertext corresponding to an element of $\mathbb{G}_i$.

These intuitive ideas essentially describe how Garg, Gentry and Halevi's GGH13 multilinear maps [GGH13a] are obtained based on (the large message space, somewhat homomorphic variant of) Gentry's FHE scheme [Gen09].

More precisely, Gentry's scheme is defined over the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^m + 1)$ for some m, with respect to a certain principal ideal $I = \langle \mathbf{g} \rangle$ with small generator $\mathbf{g}$. The plaintext space consists of small elements in R/I (say polynomials with $0/1$ coefficients) and a message $\mathbf{m}$ is encrypted as $\mathbf{c} = \mathbf{m} + \mathbf{r} \cdot \mathbf{g} \bmod q \in R_q = R/qR$ for some small noise $\mathbf{r}$. In contrast, a GGH13 encoding of $\mathbf{m}$ at level i is of the form:

$$\mathbf{c}_i = \frac{\mathbf{m} + \mathbf{r} \cdot \mathbf{g}}{\mathbf{z}^i} \bmod q,$$

where $\mathbf{z}$ is a secret masking element, and one can see that linear operations at a given level as well as multiplications between levels work as expected (as long as the noise values $\mathbf{r}$ remain appropriately small). Equality tests at level n are carried out using a *zero-testing parameter* $\mathbf{p}_{\text{zt}}$ of the form:

$$\mathbf{p}_{\text{zt}} = \mathbf{h} \cdot \mathbf{z}^n \cdot \mathbf{g}^{-1} \bmod q$$

where $\mathbf{h}$ is small: the idea is that $\mathbf{p}_{\text{zt}} \cdot \mathbf{c}_n \equiv \mathbf{h} \cdot (\mathbf{m}\mathbf{g}^{-1} + \mathbf{r}) \bmod q$ will be small if and only if $\mathbf{m} = 0$, allowing to test for equality to zero, and then usual equality by linearity.

For Gentry's scheme to be secure, the generator $\mathbf{g}$ of I has to be kept secret, although a "bad" basis of I consisting of large vectors can be published. Since $\mathbf{p}_{\text{zt}}$ depends on $\mathbf{g}$, we can see that the zero-testing parameters reveals some information about the FHE secret key as expected. This partial key leakage, which is inherent to the conceptual construction of multilinear maps from FHE, is the reason why we are not able to prove the scheme secure even though the FHE scheme itself has a proof of security.

Soon after Garg et al. published their candidate construction, another FHE-inspired construction was described by Coron, Lepoint and Tibouchi [CLT13a], related this time to the FHE scheme "over the integers" of van Dijk et al. [vDGHV10] (or more precisely, on the batch variant due to Cheon et al. [CCK$^+$13]). The main ingredients of the construction are essentially the same as those of [GGH13a], although a number of technical details are different.

Later on, variants of those two constructions were proposed to address certain technical issues, although with limited success [LSS14, CLT15]. In addition, a substantially different construction (GGH15) was introduced by Gentry, Gorbunov and Halevi [GGH15], inspired by the LWE-based FHE scheme of Gentry, Sahai and Waters [GSW13]. The functionality achieved in GGH15 differs syntactically from that of GGH13 and CLT13: instead of being arranged in a graded structure, the "groups" containing the encodings correspond to edges on a directed acyclic graph, and two encodings can be multiplied together if and only if their associated edges are adjacent. It is not immediately obvious how to use that primitive to construct the same cryptographic objects as with standard multilinear maps, but Gentry et al. showed how it can be done in a number of specific instances, including multiparty key exchange and obfuscation.

1.4 Some applications of multilinear maps

Multiparty Diffie–Hellman key exchange. As we have mentioned, the most direct application of n-linear maps is a one-round protocol for $(n+1)$-way Diffie–Hellman key exchange. This protocol can also be instantiated in the *graded encoding scheme* setting of Garg et al. [GGH13a]. Note however that since encodings are randomized, to derive a common shared key, the parties need to be able to extract some deterministic value depending only on the underlying "group element" that randomized encoding represent. This procedure is an extension of the zero-testing algorithm alluded to above, and is part of the formal definition of a graded encoding scheme. See §2.2 for details, and §2.3 for a formal description of the multiparty Diffie–Hellman key exchange protocol over graded encoding schemes.

Attribute-based encryption for circuits. One of the great successes of pairing-based cryptography is the realization of the notion of *attribute-based encryption* (ABE) [SW05, GPSW06a]. In a (ciphertext-policy) ABE scheme, users have secret keys associated with certain sets of attributes, and messages are encrypted with respect to policies which are Boolean functions of the attributes. Thus, a user with attributes x and y can decrypt ciphertexts associated with the policy $x \wedge y$, or the policy $x \vee z$, but not the policy $x \wedge z$. A major challenge in constructing ABE is the requirement that the scheme should achieve *collusion-resistance*: if Alice has the attributes x, y and Bob has the attributes y, z, they should not be able to decrypt a ciphertext with policy $x \wedge z$ even when colluding together.

There are constructions of ABE based on bilinear pairings that support policies represented by arbitrary Boolean *formulas* of the attributes, or more generally by span programs [GPSW06b], but techniques based on pairings have so far failed to achieve ABE for arbitrary polynomial-size Boolean *circuits*. One seems to encounter a fundamental limitation of bilinearity when trying to obtain collusion-resistance for arbitrary circuits, due to a class of attack known as backtracking [GGH$^+$13c, §1].

On the other hand, over *multilinear maps*, relatively direct generalizations of the classical pairing-based constructions of ABE yield ABE for all circuits right away, as shown by Garg et al. [GGH$^+$13c, GGHZ14]. Later on, Gorbunov, Vaikuntanathan and Wee were able to construct attribute-based encryption for circuits from standard lattice assumptions as well [GVW13], but the problem of a pairing-based realization remains open.

Witness encryption. Shortly after the first multilinear map candidate GGH13 was proposed, Garg, Gentry, Sahai and Waters introduced the intriguing and powerful new notion of *witness encryption*, and showed how it can be realized from multilinear maps [GGSW13]. A witness encryption scheme is defined with respect to a certain NP language L, and consists of two efficient algorithms: $\mathsf{Encrypt}(1^\lambda, x, m)$ takes as input a security parameter, a string x and a message m, and outputs a ciphertext c; $\mathsf{Decrypt}(c, w)$ takes as input a ciphertext c and a string w, and outputs either a message m' or $\perp$. Correctness states that if x is an instance of L and w is a witness of $x \in L$, then $\mathsf{Decrypt}(\mathsf{Encrypt}(1^\lambda, x, m), w)$ outputs the same message m with probability 1. Soundness security states that the encryptions of distinct messages with respect to a string $x \notin L$ are indistinguishable.

In other words, witness encryption makes it possible to encrypt messages with respect to an instance x of the language L, and one can decrypt a ciphertext if one knows a witness to the fact that $x \in L$. Security does *not* imply that knowing a witness is necessary to decrypt in general. But depending on the language L, it may be the case that instances and non-instances are computationally indistinguishable without a witness (consider e.g. the language of Diffie–Hellman pairs over a DDH group), and soundness then implies a form of semantic security with respect to adversary who do not know the witness.

Witness encryption is a powerful notion; it implies strong forms of identity-based encryption, and even ABE for all circuits in an essentially black-box way [GGSW13]. The original instantiation was based on a hardness assumption that mimicked the construction very closely, but constructions based on less ad hoc assumptions were later proposed as well [GLW14, AJN+16].

Functional encryption. The notion of *functional encryption* is a far-reaching generalization of ABE introduced by Boneh, Sahai and Waters [BSW11]. In a functional encryption scheme defined with respect to a functionality $F\colon K \times X \to \{0,1\}^*$, a user secret key sk_k is associated with an element k the set K, and if a ciphertext c is an encryption of $x \in X$, the decryption algorithm applied to sk_k and c returns $F(k,x)$. For example, ciphertext-policy ABE is the special case when elements k of K are sets of attributes, elements of X consist of a pair (m,f) of a message and a predicate, and the functionality $F(k,x)$ evaluates to m if $f(k)$ is true and to $\perp$ otherwise.

A number of special cases of functional encryption have been described in the context of pairing-based cryptography, such as predicate encryption for inner-products [KSW08, OT09], spatial encryption [Ham11] and functional encryption for inner-product functionalities [ABDP15, BJK15], but they tend to be limited to functionalities that are “bilinear” in some sense.[1]

In contrast, one of the first results to emerge as a consequence of multilinear maps was a construction of functional encryption for all circuits [GGH+13b]. That construction is in fact based on the *indistinguishability obfuscator* proposed in the same paper (see below), so it relies on multilinear maps only in an indirect way in some sense. However, other instantiations based directly on multilinear maps (i.e. without obfuscation) have later been described, starting with the scheme of Garg, Gentry, Halevi and Zhandry [GGHZ16].

Indistinguishability obfuscation. Perhaps the most impressive result that followed the GGH13 multilinear map candidate was the description by Garg et al. [GGH+13b] of a possible construction of *indistinguishability obfuscation* for all circuits. Program obfuscation, roughly speaking, aims at making it possible to publish programs whose functionality depends on some secrets in such a way that even the source code of the program will not reveal those secrets. They are, in some sense, hidden in plain sight.

[1] More general notions have also been achieved over lattices, such as leveled predicate encryption for circuits [GVW15], and even some strong forms of functional encryption for circuits with a single-bit output [GKP+13]. Those notions, however, are weaker than the functional encryption schemes achieved from multilinear maps.

Strong forms of obfuscation can be achieved for very limited classes of functionalities using standard cryptographic techniques. For example, one can publish the source code of a program that checks if its input is equal to a secret password (a so-called "point function") without revealing that password: simply put in the program the image of the password under some one-way function. But being able to do the same with much more general classes of function is immensely powerful: for example, it allows to convert any symmetric key encryption scheme to public-key (just publish as "public key" the obfuscated encryption algorithm with an embedded symmetric key).

Unfortunately, as part of their study of various notions of obfuscation, Barak et al. found that the most natural notion of program obfuscation ("black-box obfuscation") is in fact impossible to achieve for general programs [BGI+01, BGI+10]. However, they also introduced weaker notions for which they could not obtain an impossibility result, including indistinguishability obfuscation, which Goldwasser and Rothblum later showed to be, in a precise technical sense, the *best possible* obfuscation [GR07].

An indistinguishability obfuscator $\mathscr{O}$ for a class $\mathscr{C}$ of circuits is a circuit transformation which is functionality-preserving (i.e. for a circuit $C \in \mathscr{C}$, $\mathscr{O}(C)$ is another circuit which agrees with C on all inputs) and guarantees that for two circuits $C_1, C_2 \in \mathscr{C}$ that are functionally equivalent (i.e. agree on all inputs), then $\mathscr{O}(C_1)$ and $\mathscr{O}(C_2)$ are computationally indistinguishable. Note that it is not immediately clear what that notion could be useful for: for example, since point functions associated with distinct passwords are inequivalent, there is no guarantee that applying an indistinguishability obfuscator to such a function will hide the password.

Nevertheless, most readers of Barak et al. and Goldwasser–Rothblum would probably have assumed that an impossibility result for indistinguishability obfuscation to be a lot more likely than an instantiation, so Garg et al.'s construction [GGH+13b] came as a great surprise. Moreover, their paper demonstrated that the notion is in fact actually extremely powerful, since it was sufficient to achieve the long-awaited construction of functional encryption. Following their work, a number of papers, such as [SW14], developed more systematic techniques to use indistinguishability obfuscation, and it is now understood to be powerful enough to construct, in the words of Bitansky and Vaikuntanathan, "almost any known cryptographic object." [BV15]

Relations between some of these notions. It is interesting to note that the more powerful notions described above, namely functional encryption and indistinguishability obfuscation, turn out to be essentially *equivalent*, and also equivalent to multilinear maps.

More precisely, as we have said, indistinguishability obfuscation (together with some standard primitives like PRFs) implies (compact, multibit) functional encryption for all circuits [GGH+13b], even with adaptive security [Wat15] (and in fact, there is a generic conversion from selective to adaptive security [ABSV15]). Conversely, (compact, multibit) functional encryption for all circuits is sufficient to achieve indistinguishability obfuscation [AJ15, BV15]. In fact, recent candidate constructions of indistinguishability obfuscation such as [LV16, Lin16, AS16] have used some form of functional encryption as an intermediate building block.

In addition, it is now known that if n-linear maps satisfying certain DDH-like security notions exist for some sufficiently large constant n (the current record is $n = 5$, obtained by Lin in [Lin16] and Ananth and Sahai in [AS16]), then indistinguishability obfuscation/function encryption exist as well. And conversely, Albrecht et al. have shown that indistinguishability obfuscation (again together with standard primitives like homomorphic encryption and NIZK) is enough to construct multilinear maps [AFH+16].

This means that constructing multilinear maps, functional encryption and indistinguishability obfuscation are equivalent goals, and future constructions could be obtained from any of those primitives.

1.5 Attacks against multilinear map constructions

If secure, we have seen that candidate constructions of multilinear maps have very interesting consequences in cryptography (and we have only touched upon a few among many). The actual security picture is far from clear, however.

Indeed, attacks have been demonstrated against all constructions so far, and we describe a number of them in details in Chapter 3. The current situation is that, due to a long series of attacks [CHL+15, CLT14, CGH+15, CFL+16, HJ16, CLLT16a], multiparty Diffie–Hellman key exchange is broken over all of the proposed candidates. In addition, a number of attacks have been demonstrated against several constructions of indistinguishability obfuscation [MSZ16a, CGH16, ADGM16, CLLT17], but not all schemes are broken yet.

We can also mention that GGH13 and CLT13 are both broken in classical subexponential time and quantum polynomial time. In the case of CLT13, it is because it relies on the hardness of factoring. In the case of GGH13, it is a consequence of recent progress on the cryptanalysis of some ideal lattice assumptions in the presence of very small noise [ABD16].

Chapter 2

Definitions and Constructions

2.1 Multilinear maps

2.1.1 The Boneh–Silverberg setting

Boneh and Silverberg introduced the notion of multilinear maps in a cryptographic setting [BS03]. The definition they adopted for their purposes was touched upon in §1.2. We recall it more formally below.

Definition 1. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups (denoted additively), and $e\colon \mathbb{G}^\kappa \to \mathbb{G}_T$ a mapping for some integer $\kappa \geq 1$. We say that e is a κ-*linear map *(or simply a* multilinear map*) when the following conditions hold:*

1. *$\mathbb{G}$ and $\mathbb{G}_T$ are of the same prime order;*
2. *for any $a_1, \dots, a_\kappa \in \mathbb{Z}$ and $g_1, \dots, g_\kappa \in \mathbb{G}$, we have*
$$e(a_1 \cdot g, \dots, a_\kappa \cdot g) = a_1 \cdots a_\kappa \cdot e(g, \dots, g);$$
3. *if g is a generator of $\mathbb{G}$, then $e(g, \dots, g)$ is a generator of $\mathbb{G}_T$.*

2.1.2 Efficient algorithms

Boneh and Silverberg called a multilinear map as above a *cryptographic multilinear map* when the groups $\mathbb{G}$ and $\mathbb{G}_T$ admit efficient group operations, when the map e itself is efficiently computable, and when the scheme satisfies some notion of security like the hardness of discrete logarithms in $\mathbb{G}$. Since efficiency and security are asymptotic notions, they can only make sense with respect to some instance generation algorithm.

Following [GGH12], one can capture these notions (minus the security, which can be definitely independently by a suitable game) by saying that a *multilinear map scheme* is a tuple of algorithms $(\mathsf{InstGen}, \mathsf{add}, \mathsf{neg}, \mathsf{EncTest}, \mathsf{map})$ for instance generation, group operations, membership testing and multilinear pairing, which can be described as follows.

Instance generation. $\mathsf{InstGen}(1^\lambda, 1^\kappa)$ is an efficient randomized algorithm that takes as input the security parameter λ and the multilinearity degree κ, and outputs a description of the groups $\mathbb{G}$ and $\mathbb{G}_T$, their order q, a description of the multilinear map $e\colon \mathbb{G}^\kappa \to \mathbb{G}_T$, and a string $g \in \{0,1\}^*$ encoding a generator of $\mathbb{G}$. The tuple $(\mathbb{G}, \mathbb{G}_T, q, e)$ is denoted by pp.

Membership testing. $\mathsf{EncTest}(\mathsf{pp}, b, x)$ is an efficient, deterministic algorithm that takes as input the parameters pp, a bit b and a string $x \in \{0,1\}^*$ and decides whether x is a valid representation of an element of $\mathbb{G}$ (resp. $\mathbb{G}_T$) when $b = 0$ (resp. $b = 1$). It is assumed that representations are unique, so we simply denote validity by membership: $x \in \mathbb{G}$ (resp. $x \in \mathbb{G}_T$).

Group operations. $\mathsf{add}(\mathsf{pp}, b, x, y)$ is efficient, deterministic, and returns $x + y$ when $b = 0$ and x and y are both elements of $\mathbb{G}$ (resp. $b = 1$ and $x, y \in \mathbb{G}_T$). Note that this is sufficient to efficiently compute $a \cdot x$ for $a \in \mathbb{Z}_q$ by double-and-add. Similarly, $\mathsf{neg}(\mathsf{pp}, b, x)$ computes $-x$.

Multilinear map. $\mathsf{map}(\mathsf{pp}, x_1, \dots, x_\kappa)$ is efficient, deterministic, and returns the target group element $e(x_1, \dots, x_\kappa) \in \mathbb{G}_T$.

2.1.3 Symmetry vs. asymmetry

The multilinear maps describe above are *symmetric* in the sense that all the source group are the same (or equivalently, are efficiently isomorphic). It is straightforward to extend the definition to multilinear maps of the form $e\colon \mathbb{G}_1 \times \cdots \times \mathbb{G}_\kappa \to \mathbb{G}_T$ where the groups $\mathbb{G}_i$ are all of the same prime order, but there does not necessarily exist efficient isomorphisms between them. That setting occurs frequently with elliptic curves (for type II and type III pairings in the sense of [GPS08]), and has been described for multilinear maps by Rothblum [Rot13].

2.2 Graded encoding schemes

As discussed in §1.3, the functionality achieved by constructions such as [GGH13a] and [CLT13a] differ from the Boneh–Silverberg definition above in at least two important aspects:

- contrary to the Boneh–Silverberg setting where elements of the source group are combined in one go to form an element of the target group, encodings are arranged in several levels, and one can pair elements at level i and level j to obtain an element at level $i + j$, and so on several times;
- a single "exponent" can be represented at a given level by many different encodings.

The corresponding notion is captured by the definition of a graded encoding system, and its algorithmic description, as presented below.

2.2.1 Graded encoding system

We recall the formal definition of a κ*-graded encoding system* from [GGH13a]. For simplicity we only consider the symmetric case below. See [GGH12, Appendix A] for the description of a more general framework that can handle asymmetric multilinear maps and gradings with respect to more complicated monoids.

Definition 2. *A κ-*graded encoding system *for a ring R is a system of sets $\mathscr{S} = \{S_v^{(\alpha)} \in \{0,1\}^* : v \in \mathbb{N}, \alpha \in R\}$, with the following properties:*

1. *For every $v \in \mathbb{N}$, the sets $\{S_v^{(\alpha)} : \alpha \in R\}$ are disjoint.*
2. *There are binary operations $+$ and $-$ (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every $v \in \mathbb{N}$, and every $u_1 \in S_v^{(a_1)}$ and $u_2 \in S_v^{(\alpha_2)}$, it holds that $u_1 + u_2 \in S_v^{(\alpha_1+\alpha_2)}$ and $u_1 - u_2 \in S_v^{(\alpha_1-\alpha_2)}$ where $\alpha_1 + \alpha_2$ and $\alpha_1 - \alpha_2$ are addition and subtraction in R.*
3. *There is an associative binary operation $\times$ (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every v_1, v_2 with $0 \leq v_1 + v_2 \leq \kappa$, and every $u_1 \in S_{v_1}^{(\alpha_1)}$ and $u_2 \in S_{v_2}^{(\alpha_2)}$, it holds that $u_1 \times u_2 \in S_{v_1+v_2}^{(\alpha_1 \cdot \alpha_2)}$ where $\alpha_1 \cdot \alpha_2$ is multiplication in R.*

2.2.2 Efficient procedures

We also recall the definition of the procedures for manipulating encodings. As previously we consider only the symmetric case.

Instance generation. The randomized $\mathsf{InstGen}(1^\lambda, 1^\kappa)$ takes as inputs the parameters λ and κ, and outputs $(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}})$, where pp is a description of a κ-Graded Encoding System as above, and $\mathbf{p}_{\mathrm{zt}}$ is a zero-test parameter.

Ring sampler. The randomized $\mathsf{samp}(\mathsf{pp})$ outputs a "level-zero encoding" $a \in S_0^{(\alpha)}$ for a nearly uniform element $\alpha \in_R R$. Note that the encoding a does not need to be uniform in $S_0^{(\alpha)}$.

Encoding. The (possibly randomized) $\mathsf{enc}(\mathsf{pp}, i, a)$ takes as input a level-zero encoding $a \in S_0^{(\alpha)}$ for some $\alpha \in R$ and a level $i \leq \kappa$, and outputs a level-i encoding $u \in S_i^{(\alpha)}$ for the same α.

Rerandomization. The randomized $\mathsf{reRand}(\mathsf{pp}, i, u)$ re-randomizes encodings relative to the same level i. Specifically, given an encoding $u \in S_v^{(\alpha)}$, it outputs another encoding $u' \in S_v^{(\alpha)}$. Moreover for any two $u_1, u_2 \in S_v^{(\alpha)}$, the output distributions of $\mathsf{reRand}(\mathsf{pp}, i, u_1)$ and $\mathsf{reRand}(\mathsf{pp}, i, u_2)$ are nearly the same.

Addition and negation. Given pp and two encodings relative to the same level, $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, we have $\mathsf{add}(\mathsf{pp}, u_1, u_2) \in S_i^{(\alpha_1+\alpha_2)}$ and $\mathsf{neg}(\mathsf{pp}, u_1) \in S_i^{(-\alpha_1)}$. Below we write $u_1 + u_2$ and $-u_1$ as a shorthand for applying these procedures.

Multiplication. For $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_j^{(\alpha_2)}$, we have $\mathsf{mul}(\mathsf{pp}, u_1, u_2) = u_1 \times u_2 \in S_{i+j}^{(\alpha_1 \cdot \alpha_2)}$.

Zero-testing. The procedure $\mathsf{isZero}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, u)$ outputs 1 if $u \in S_\kappa^{(0)}$ and 0 otherwise.

Extraction. The procedure extracts a random function of ring elements from their level-κ encoding. Namely $\mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, u)$ outputs $s \in \{0,1\}^\lambda$, such that:

1. For any $\alpha \in R$ and $u_1, u_2 \in S_\kappa^{(\alpha)}$, $\mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathsf{zt}}, u_1) = \mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathsf{zt}}, u_2)$.

2. The distribution $\{\mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathsf{zt}}, u) : \alpha \in_R R, u \in S_\kappa^{(\alpha)}\}$ is nearly uniform over $\{0,1\}^\lambda$.

2.2.3 Approximate graded encodings

As pointed out in [GGH13a], actual constructions only achieve a slightly relaxed definition of isZero and ext, where isZero can still output 1 even for some non-zero encoding u with negligible probability, and ext can extract different outputs when applied to encodings of the same elements, also with negligible probability. See [GGH12, §2.2.2 and A.2] for the corresponding definitions.

2.3 Security definitions: the example of Diffie–Hellman key exchange

The sheer number of subtly or wildly different hardness assumptions used for security proofs in the field of pairing-based cryptography has been the object of many comments, for better or worse [Boy08, KM10]. Unsurprisingly, the more convoluted setting of multilinear maps and graded encoding schemes has seen the use of an even broader range of potential hard problems (see e.g. the discussion in [LV16, §1] for a discussion of the particular case of obfuscation candidates). It seems difficult, at this stage, to point to a particular security definition that could be singled out as the *correct* desirable security goal when trying to construct multilinear maps.

Nevertheless, one simple security definition has been emphasized in a number of construction papers, including [GGH13a, CLT13a, LSS14, CLT15], namely the graded encoding analogue of the decisional Diffie–Hellman assumption. Since it is so common, we recall it here, and add a few comments afterwards discussing the place of that assumption within the literature.

2.3.1 The graded decisional Diffie–Hellman problem

In their original paper [GGH13a], Garg et al. introduced the *graded decisional Diffie–Hellman* assumption (GDDH) as a security goal for graded encoding scheme. It is defined as follows (this is the definition from [LSS14], which looks slightly different from the one in [GGH13a, CLT13a], but is easily seen to be equivalent as long as reRand behaves correctly).

Consider the following procedure, parametrized by λ and κ:

1. Run $\mathsf{InstGen}(1^\lambda, 1^\kappa)$ to obtain $(\mathsf{pp}, \mathbf{p}_{\mathsf{zt}})$.

2. Sample $a_j \leftarrow \mathsf{samp}(\mathsf{pp})$ for $0 \le j \le \kappa$.

3. Compute $u_j \leftarrow \mathsf{reRand}(\mathsf{pp}, 1, \mathsf{enc}(\mathsf{pp}, 1, a_j))$ for $0 \le j \le \kappa$.

4. Sample $b \leftarrow \mathsf{samp}(\mathsf{pp})$.

5. Compute the product encoding $u^* = a_0 \cdot \prod_{j=1}^{\kappa} u_i$ of the u_j's by repeated application of the mul procedure (encoding at level κ).

6. Set $v^{(0)} = \mathsf{reRand}(\mathsf{pp}, \kappa, u^*)$.

7. Set $v^{(1)} = \mathsf{reRand}(\mathsf{pp}, \kappa, \mathsf{enc}(\mathsf{pp}, \kappa, b))$.

8. Pick a bit β uniformly at random and set $v \leftarrow v^{(\beta)}$.

The GDDH assumption asserts that an efficient adversary receiving as input the values $(p^*_{,}\mathbf{p}_{\mathrm{zt}}, u_0, \dots, u_\kappa, v)$ can only guess the bit β with an advantage negligible in the security parameter λ.

Clearly, the GDDH assumption implies that the N-party key exchange protocol defined below is passively secure.

$\mathsf{Setup}(1^\lambda, 1^N)$. Output $(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$ as the public parameter, with $\kappa = N - 1$.

$\mathsf{Publish}(\mathsf{pp}, i)$. Each party i samples a random $c_i \leftarrow \mathsf{samp}(\mathsf{pp})$ as a secret value, and publishes as the public value the corresponding level-1 encoding, computed as $c_i' \leftarrow \mathsf{reRand}(\mathsf{pp}, 1, \mathsf{enc}(\mathsf{pp}, 1, c_i))$.

$\mathsf{KeyGen}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, i, c_i, \{c_j'\}_{j \neq i})$. Each party i computes $\tilde{c}_i = c_i \cdot \prod_{j \neq i} c_j'$, and uses the extraction routine to locally compute the common secret $s \leftarrow \mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, \tilde{c}_i)$.

2.3.2 Discussion

The GDDH assumption does capture the security of multiparty key exchange (almost tautologically so!), but may not otherwise be a particularly useful security definition. The hardness assumptions under which more interesting primitives like witness encryption (e.g. in [GGSW13]) and indistinguishability obfuscation (e.g. in [PST14]) have been shown to exist are usually considerably more intricate, and not much has been done over multilinear maps with Diffie–Hellman-like assumptions. As a recent result counter to that trend, albeit in the Boneh–Silverberg setting rather than over graded encoding schemes, one can mention the surprising construction by Lin of indistinguishability obfuscation from (subexponential) DDH over 5-linear maps [Lin16].

Another issue with the GDDH assumption is that, unfortunately, proposed multilinear candidates have turned out not to satisfy it: as we will see in the next chapter, attacks have been found against multiparty Diffie–Hellman over the graded encoding schemes from [GGH13a, CLT13a] and their variants! This is of course considered a serious problem. A silver lining, however, is that GDDH is not *as basic* a problem as it sounds.

Indeed, one particular feature of the multiparty Diffie–Hellman scheme as described above is that it relies on the possibility for all users to publicly generate and rerandomize their own encodings. In contrast, in many other schemes, including witness encryption and indistinguishability obfuscation, the ability to generate encodings of new values is only used by the same user that generates the system parameters. In those settings, it is thus possible to require secret information in enc and reRand, whereas only arithmetic operations and zero-testing/extraction remain public procedures. This leads to the definition of what Albrecht et al. call *secret-key graded encoding schemes* [ACLL15], which tend to be much more difficult to attack than Diffie–Hellman key exchange. Nevertheless, attacks have indeed been found in that setting as well (e.g. against constructions of indistinguishability obfuscation [CGH+15, MSZ16a, CLLT17]), as we will see in the next chapter.

2.4 A concrete instantiation: the CLT13 graded encoding scheme

The graded encoding schemes from [GGH13a] and [CLT13a] are very similar to each other, but a thorough presentation of GGH13 requires somewhat more background material, especially on algebraic number theory and Gaussian sampling on lattices. Therefore, for simplicity's sake, we only give a complete description of the CLT13 scheme. We refer back to §1.3 above for a conceptual-level description of GGH13, and to §2.4.3 below for a short rundown of the main differences between GGH13 and CLT13.

2.4.1 The shape of CLT13 encodings

In the "integer-based" scheme of [CLT13a], a level-k encoding of a short integer vector $\mathbf{m} = (m_i) \in \mathbb{Z}^n$ is an integer c such that for all $1 \leq i \leq n$:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \pmod{p_i} \tag{2.1}$$

where the r_i's are ρ-bit random integers (specific to the encoding c), with the following secret parameters: the p_i's are η-bit prime integers, the g_i's are α-bit primes, and the denominator z is a random (invertible) integer modulo $x_0 = \prod_{i=1}^{n} p_i$. The integer c is therefore well-defined modulo x_0, where x_0 is made public. Since the p_i's must remain secret, the user cannot encode the vectors $\mathbf{m} \in \mathbb{Z}^n$ by CRT directly from (2.1); instead one includes in the public parameters a set of ℓ level-0 encodings x'_j of random vectors $\mathbf{a}_j \in \mathbb{Z}^n$, and the user can generate a random level-0 encoding by computing a random subset sum of those x'_j's.

From (2.1) we see that each integer m_i is actually defined modulo g_i. Therefore, the CLT13 scheme encodes vectors $\mathbf{m}$ from the ring $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$.

2.4.2 Detailed description of CLT13

System parameters. The main parameters are the security parameter λ and the required multilinearity level $\kappa \leq \mathrm{poly}(\lambda)$. Based on λ and κ, we choose the vector dimension n, the bit-size η of the primes p_i, the bit-size α of the primes g_i, the maximum bit-size ρ of the randomness used in encodings, and various other parameters that will be specified later; the constraints that these parameters must satisfy are described in the next section. For integers z, p we denote the reduction of z modulo p by $(z \bmod p)$ or $[z]_p$ with $-p/2 < [z]_p \leq p/2$.

Instance generation. $(\mathsf{pp}, \mathbf{p}_{\mathsf{zt}}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. This algorithm generates n secret random η-bit primes p_i and computes $x_0 = \prod_{i=1}^n p_i$. It then generates a random invertible integer z modulo x_0, n random α-bit prime integers g_i, and a secret matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}^{n\times\ell}$, where each component a_{ij} is randomly generated in $[0, g_i) \cap \mathbb{Z}$. It also generates an integer y, three sets of integers $\{x_j\}_{j=1}^\tau$, $\{x'_j\}_{j=1}^\ell$ and $\{\Pi_j\}_{j=1}^n$, a zero-testing vector $\mathbf{p}_{\mathsf{zt}}$, and a seed s for a strong randomness extractor; the shape of these elements is detailed below in the respective algorithms where they intervene. The parameters $\mathsf{pp} = \big(n, \eta, \alpha, \rho, \beta, \tau, \ell, y, \{x_j\}_{j=1}^\tau, \{x'_j\}_{j=1}^\ell, \{\Pi_j\}_{j=1}^n, s\big)$ and $\mathbf{p}_{\mathsf{zt}}$ are finally output and made public.

Sampling level-zero encodings. $c \leftarrow \mathsf{samp}(\mathsf{pp})$. Recall that the parameters pp contain a set of ℓ integers x'_j, where each x'_j encodes at level-0 the column vector $\mathbf{a}_j \in \mathbb{Z}^n$ of the secret matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}^{n\times\ell}$. More precisely, the integers x'_j are generated by Chinese remaindering, subject to the condition that:

$$x'_j \equiv r'_{ij} \cdot g_i + a_{ij} \pmod{p_i} \quad \text{for } 1 \le j \le \ell, \tag{2.2}$$

where the r'_{ij}'s are randomly generated in $(-2^\rho, 2^\rho) \cap \mathbb{Z}$.

Using those values x'_j, the randomized sampling algorithm $\mathsf{samp}(\mathsf{pp})$ works as follows: it samples a random binary vector $\mathbf{b} = (b_j) \in \{0,1\}^\ell$ and outputs the level-0 encoding

$$c = \sum_{j=1}^{\ell} b_j \cdot x'_j \bmod x_0.$$

From Equation (2.2), this gives $c \equiv \big(\sum_{j=1}^\ell r'_{ij} b_j\big) \cdot g_i + \sum_{j=1}^\ell a_{ij} b_j \pmod{p_i}$. As required, the output c is a level-0 encoding:

$$c \equiv r_i \cdot g_i + m_i \pmod{p_i} \tag{2.3}$$

of some vector $\mathbf{m} = \mathbf{A} \cdot \mathbf{b} \in \mathbb{Z}^n$ which is a random subset-sum of the column vectors $\mathbf{a}_j$. The sizes of the reductions $[c]_{p_i}$ are then well controlled for all i:

$$|r_i \cdot g_i + m_i| \le \ell \cdot 2^{\rho+\alpha}.$$

A left-over hash lemma argument ensures that $\mathbf{m}$ will be statistically close to uniform over $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ for suitably chosen parameters. See [CLT13a] for details.

Encodings at higher levels. $c_k \leftarrow \mathsf{enc}(\mathsf{pp}, k, c)$. To allow encoding at higher levels, a level-one random encoding of the vector $\mathbf{1}$ was published as part of the public parameters pp. That value is an integer y is generated in such a way that:

$$y \equiv \frac{r_i \cdot g_i + 1}{z} \pmod{p_i}$$

for random integers $r_i \in (-2^\rho, 2^\rho) \cap \mathbb{Z}$.

Given a level-0 encoding c of $\mathbf{m} \in \mathbb{Z}^n$ as given by (2.3), a level-1 encoding of the same $\mathbf{m}$ can be obtained by computing $c_1 = c \cdot y \bmod x_0$. Indeed, we then have:

$$c_1 \equiv \frac{r_i^{(1)} \cdot g_i + m_i}{z} \pmod{p_i} \tag{2.4}$$

for some integers $r_i^{(1)}$, and we get $|r_i^{(1)} \cdot g_i + m_i| \leq \ell \cdot 2^{2(\rho+\alpha)}$ for all i. More generally to generate a level-k encoding we compute $c_k = c_0 \cdot y^k \bmod x_0$.

Note however that this element should not be published as is, since it would then be possible to go back to the lower-level encoding c by simply dividing by y, thus inverting the multilinear map. Instead the level-1 encoding c_1 should first be re-randomized into a new level-1 encoding c_1' for the same vector $\mathbf{m}$, but whose distribution is otherwise independent of the original c. This is done with the following algorithm.

Rerandomization. $c' \leftarrow \mathsf{reRand}(\mathsf{pp}, k, c)$. To allow rerandomization of encodings at level $k = 1$, the public parameters pp contain a set of n integers Π_j which are all level-1 random encodings of zero:

$$\Pi_j \equiv \frac{\varpi_{ij} \cdot g_i}{z} \pmod{p_i} \quad \text{for } 1 \leq j \leq n.$$

The matrix $\mathbf{\Pi} = (\varpi_{ij}) \in \mathbb{Z}^{n \times n}$ is a diagonally dominant matrix generated as follows: the non-diagonal entries are randomly and independently generated in $(-2^\rho, 2^\rho) \cap \mathbb{Z}$, while the diagonal entries are randomly generated in $(n2^\rho, n2^\rho + 2^\rho) \cap \mathbb{Z}$.

The parameters pp also contain a set of τ integers x_j, each one of which is a level-1 random encoding of zero:

$$x_j \equiv \frac{r_{ij} \cdot g_i}{z} \pmod{p_i} \quad \text{for } 1 \leq j \leq \tau,$$

and where the column vectors of the matrix $(r_{ij}) \in \mathbb{Z}^{n \times \tau}$ are randomly and independently generated in the half-open parallelepiped spanned by the columns of the previous matrix $\mathbf{\Pi}$. This somewhat complicated choice is made to ensure a proper rerandomization.

Given a level-1 encoding c_1 as given by (2.4), the procedure reRand rerandomizes it by adding a random subset-sum of the x_j's and a linear combination of the Π_j's:

$$c_1' = c_1 + \sum_{j=1}^{\tau} b_j \cdot x_j + \sum_{j=1}^{n} b_j' \cdot \Pi_j \bmod x_0 \tag{2.5}$$

where $b_j \leftarrow \{0, 1\}$, and $b_j' \leftarrow [0, 2^\mu) \cap \mathbb{Z}$. One of the main technical difficulties of the construction of [CLT13a] is the proof that the distribution of c_1' is nearly independent of the input c_1 (aside from the fact that both encodings correspond to the same vector $\mathbf{m}$). This is shown using a "left-over hash lemma over lattices". We refer to the original paper for details.

Adding and multiplying encodings. It is clear that one can homomorphically add encodings. Moreover the product of κ level-1 encodings u_i can be interpreted as an encoding of the product. Namely, given level-one encodings u_j of vectors $\mathbf{m}_j \in \mathbb{Z}^n$ for $1 \le j \le \kappa$, with $u_j \equiv (r_{ij} \cdot g_i + m_{ij})/z \pmod{p_i}$, the product

$$u = \prod_{j=1}^{\kappa} u_j \bmod x_0$$

satisfies:

$$u \equiv \frac{\prod_{j=1}^{\kappa} (r_{ij} \cdot g_i + m_{ij})}{z^\kappa} \equiv \frac{r_i \cdot g_i + \left(\prod_{j=1}^{\kappa} m_{ij}\right) \bmod g_i}{z^\kappa} \pmod{p_i}$$

for some $r_i \in \mathbb{Z}$. This is a level-κ encoding of the vector $\mathbf{m}$ obtained by componentwise product of the vectors $\mathbf{m}_j$, as long as $\prod_{j=1}^{\kappa}(r_{ij} \cdot g_i + m_{ij}) < p_i$ for all i. When computing the product of κ level-1 encodings from reRand and one level-0 encoding from samp as in multiparty Diffie–Hellman key exchange, one can easily check that $|r_i| \le (4n^2 2^{\mu+\rho+\alpha})^\kappa \cdot \ell \cdot 2^{\rho+1}$ for all i.

Zero testing. $\mathsf{isZero}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, u_\kappa) \stackrel{?}{=} 0/1$. Zero testing of top level encodings is carried out with the parameter $\mathbf{p}_{\mathrm{zt}}$ obtained as follows as part of instance generation. First, an integer matrix $\mathbf{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$ is randomly generated in such a way that $\mathbf{H}$ is invertible over $\mathbb{Z}$ and both $\|\mathbf{H}^T\|_\infty \le 2^\beta$ and $\|(\mathbf{H}^{-1})^T\|_\infty \le 2^\beta$, for some parameter β; here $\|\cdot\|_\infty$ is the operator norm on $n \times n$ matrices with respect to the ℓ^∞ norm on $\mathbb{R}^n$. A technique for generating such an $\mathbf{H}$ is presented in the appendix of [CLT13b]. Then, $\mathbf{p}_{\mathrm{zt}} \in \mathbb{Z}^n$ is computed as:

$$(\mathbf{p}_{\mathrm{zt}})_j = \sum_{i=1}^{n} h_{ij} \cdot \left(z^\kappa \cdot g_i^{-1} \bmod p_i\right) \cdot \prod_{i' \ne i} p_{i'} \bmod x_0. \tag{2.6}$$

To determine whether a level-κ encoding c is an encoding of zero or not, one computes the vector $\boldsymbol{\omega} = c \cdot \mathbf{p}_{\mathrm{zt}} \bmod x_0$ and tests whether $\|\boldsymbol{\omega}\|_\infty$ is small: $\mathsf{isZero}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, c)$ returns 1 if $\|\boldsymbol{\omega}\| < x_0 \cdot 2^{-\nu}$ and 0 otherwise, for some parameter ν.

The authors of [CLT13a] show that suitable choices of β and ν can ensure that this zero-testing procedure is then correct for all encodings c whose noise coefficients are appropriately bounded.

Extraction. $sk \leftarrow \mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, c)$. To extract a random value depending only on the vector $\mathbf{m}$ encoded in a level-κ encoding c, one proceeds as follows: multiply it by the zero-testing parameter $\mathbf{p}_{\mathrm{zt}}$ modulo x_0, collect the ν most significant bits of each of the n components of the resulting vector, and apply a strong randomness extractor (using the seed s from pp). More formally:

$$\mathsf{ext}(\mathsf{pp}, \mathbf{p}_{\mathrm{zt}}, c) = \mathsf{Extract}_s\big(\mathsf{msbs}_\nu(c \cdot \mathbf{p}_{\mathrm{zt}} \bmod x_0)\big)$$

where msbs_ν extracts the ν most significant bits of the result. If two encodings c and c' encode the same $\mathbf{m} \in \mathbb{Z}^n$, one can show (using the precise result establishing the correctness of zero-testing) that $\|(c - c') \cdot \mathbf{p}_{\mathrm{zt}} \bmod x_0\|_\infty < x_0 \cdot 2^{-\nu-\lambda}$, and therefore we expect that $\boldsymbol{\omega} = c \cdot \mathbf{p}_{\mathrm{zt}} \bmod x_0$ and $\boldsymbol{\omega}' = c' \cdot \mathbf{p}_{\mathrm{zt}} \bmod x_0$ agree on their ν most significant bits, and therefore extract to the same value. And conversely if the encoded values are distinct. We refer to [CLT13a] for the nitty-gritty details.

2.4.3 Differences with GGH13

Recall from §1.3 that the graded encoding scheme from [GGH13a] is defined over the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^m + 1)$, with respect to a certain principal ideal $I = \langle \mathbf{g} \rangle$ with secret, small generator $\mathbf{g}$. A vector $\mathbf{m} \in R$ with small coefficients is encoded at level k by an element of the form:

$$\mathbf{c}_k = \frac{\mathbf{m} + \mathbf{r} \cdot \mathbf{g}}{\mathbf{z}^k} \bmod q,$$

where $\mathbf{z}$ is the secret masking element. This is essentially the same as (2.1) above.

A crucial difference, however, is the shape of the zero-testing parameter. In GGH13, it is simply of the form:

$$\mathbf{p}_{\mathrm{zt}} = \mathbf{h} \cdot \mathbf{z}^{\kappa} \cdot \mathbf{g}^{-1} \bmod q$$

with $\mathbf{h}$ small. Indeed, multiplying a level-κ encoding $\mathbf{c}_\kappa$ of $\mathbf{m}$ gives:

$$\mathbf{p}_{\mathrm{zt}} \cdot \mathbf{c}_\kappa \equiv \mathbf{h} \cdot (\mathbf{m}\mathbf{g}^{-1} + \mathbf{r}) \bmod q$$

which is small when $\mathbf{m} = 0$, but large otherwise since $\mathbf{g}^{-1}$ is expected to be of full size modulo q.

Adopting a similar zero-testing element $p_{\mathrm{zt}} = hz^{\kappa}/g \bmod x_0$ in the CLT13 setting, however, ***does not work***. This is because multiplying that value with a level-κ encoding c_κ yields an integer modulo x_0 whose ***reductions*** modulo all of the prime factors p_i of x_0 are small. But since those prime factors must be kept secret, there is no way of checking that directly. This is the reason why the vector $\mathbf{p}_{\mathrm{zt}}$ in CLT13 has the different shape (2.6), involving extra factors of the form $\prod_{j \neq i} p_j$.

Other differences between the GGH13 and CLT13 constructions mainly reside in the technical details of how various properties of the schemes (such as correct sampling and rerandomization) are proved in both settings. And of course, they have different properties in terms of security.

2.5 GGH15 and the graph-induced approach

As mentioned in §1.3, the third main construction of multilinear maps after [GGH13a] and [CLT13a] is due to Gorbunov, Gentry and Halevi [GGH15] and differs substantially from the previous constructions even in syntactic terms. The primitive that the authors achieve is not a graded-encoding scheme, but what they call a *graph-induced encoding scheme*. In what follows, we recall the definition of that primitive, and give a description of the scheme they propose.

2.5.1 Graph-induced encoding scheme

The primitive constructed in [GGH15] is parametrized by a certain directed acyclic graph, and encodings are associated to edges (or more precisely, paths of edges) on that graph. Encodings on the same path can be combined linearly, and multiplication is permitted between encodings if and only if their associated paths are adjacent. These properties are captured by saying that a *graph-induced encoding scheme* is a tuple (PrmGen, InstGen, Sample, Enc, add, neg, mult, ZeroTest, Extract) of efficient algorithms described as follows. These procedures are subject to some technical correctness conditions for which we refer to [GGH15]; the paper also discusses possible variants that we do not address in this document.

Parameter generation. $\mathsf{PrmGen}(1^\lambda, G)$ takes as inputs the security parameter λ and the underlying directed graph G, and outputs the global system parameters gp, including in particular the graph G, a description of the plaintext ring R, and a distribution χ over R from which plaintexts are sampled.

Instance generation. $\mathsf{InstGen}(\mathsf{gp})$ takes as inputs the system parameters and outputs the secret and public parameters sp, pp.

Ring sampler. The randomized $\mathsf{Sample}(\mathsf{pp})$ algorithm outputs an element of the plaintext ring R sampled according to the distribution χ.

Encoding. $\mathsf{Enc}(\mathsf{sp}, p, \alpha)$ takes as input the *secret* parameters, a path $p = u \rightsquigarrow v$ and a ring element $\alpha \in R$ in the range of Sample, and outputs an encoding u_p of α according to the path p.

Addition, negation and multiplication. The arithmetic procedures $\mathsf{add}(\mathsf{pp}, u_p, u'_p)$, $\mathsf{neg}(\mathsf{pp}, u_p)$ and $\mathsf{mult}(\mathsf{pp}, u_p, u'_{p'})$ are deterministic and take as input the public parameters together with some encodings.

Negation takes an encoding u_p of some $\alpha \in R$ with respect to a path p, and returns an encoding of $-\alpha$ relative to the same path p. Addition takes encodings u_p, u'_p of some $\alpha, \alpha' \in R$ with respect to the same path p, and returns an encoding of $\alpha + \alpha'$ relative to p. Finally, multiplication takes encodings u_p, $u'_{p'}$ of $\alpha, \alpha' \in R$ with respect to paths p, p' which are consecutive (i.e. $p = u \rightsquigarrow v$ and $p' = v \rightsquigarrow w$), and returns an encoding of $\alpha \cdot \alpha'$ with respect to the composed path $u \rightsquigarrow w$.

Zero-testing. The procedure $\mathsf{ZeroTest}(\mathsf{pp}, u)$ is deterministic, and decides whether a given encoding u is an encoding of 0 or not.

Extraction. The procedure $\mathsf{Extract}(\mathsf{pp}, u)$ is deterministic, and returns a λ-bit string depending only on the underlying plaintext α of the encoding u.

2.5.2 The GGH15 instantiation

We now describe the candidate graph-induced encoding scheme proposed by Gentry et al. in [GGH15]. Their paper actually describes several variants; we focus here on the commutative, ring based version, which is the one they use to achieve multiparty Diffie–Hellman key exchange.

That scheme is defined over the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n + 1)$. Plaintexts are small elements s in that ring, sampled according to a Gaussian distribution χ. Public parameters consist in particular of row vectors $\mathbf{A}_v \in R_q^m$ (where $R_q = R/qR$) associated to the vertices v of the underlying graph. An encoding of s associated with a path $u \rightsquigarrow v$ in the graph is then a matrix $\mathbf{D} \in R^{m \times m}$ with small coefficients such that:

$$\mathbf{A}_u \cdot \mathbf{D} = s \cdot \mathbf{A}_v + \mathbf{E} \pmod q$$

for some small error vector $\mathbf{E} \in R^m$. Such encodings $\mathbf{D}$ can be generated given some secret trapdoor information generated along with the public vectors $\mathbf{A}_v$, using classical lattice techniques [GPV08, MP12]. Formally speaking, the graph-induced encoding procedures can thus be described as follows.

Parameter generation. $\mathsf{PrmGen}(1^\lambda, G)$ computes the system parameters gp, which consist of the graph G, the description of the cyclotomic ring R, the vector dimension m, the modulus q, the plaintext Gaussian distribution χ, a dispersion parameter σ used in trapdoor sampling, and the number of most significant bits t used for zero-testing and extraction.

Instance generation. $\mathsf{InstGen}(\mathsf{gp})$ uses the trapdoor sampling algorithm of Micciancio and Peikert [MP12] to generate the vectors $\mathbf{A}_v$ for all vertices v in the underlying graph G, together with the corresponding trapdoor information τ_v. The algorithm also samples a seed and some extra information for randomness extraction. The vectors $\mathbf{A}_v$ and the extraction information form the public parameters pp, whereas the trapdoors τ_v form the secret parameters sp.

Ring sampler. The randomized $\mathsf{Sample}(\mathsf{pp})$ simply samples an element $s \in R$ according to the Gaussian distribution χ.

Encoding. $\mathsf{Enc}(\mathsf{sp}, p, s)$: to sample an encoding for $s \in R$ along the path $p = u \rightsquigarrow v$, this algorithm first samples an error vector $\mathbf{E} \in R^m$ according to χ^m, and computes $\mathbf{V} = s \cdot \mathbf{A}_v + \mathbf{E}$. It then uses the trapdoor information τ_u and the Micciancio–Peikert algorithm [MP12] to obtain a small matrix $\mathbf{D} \in R_q^{m \times m}$ such that $\mathbf{D} \cdot \mathbf{A}_u = \mathbf{V}$ over R_q. This matrix $\mathbf{D}$ is the required encoding.

Addition, negation and multiplication. Addition, negation and multiplication are the corresponding operations directly on matrices. It is easy to see that they behave as expected. Indeed, in the case of addition, if $\mathbf{D}_1$ and $\mathbf{D}_2$ are encodings of s_1, s_2 relative to the same path $u \rightsquigarrow v$, so we can write:

$$\begin{aligned} \mathbf{A}_u \cdot \mathbf{D}_1 &= s_1 \cdot \mathbf{A}_v + \mathbf{E}_1 \pmod q \\ \mathbf{A}_u \cdot \mathbf{D}_2 &= s_2 \cdot \mathbf{A}_v + \mathbf{E}_2 \pmod q \end{aligned}$$

we obtain:

$$\mathbf{A}_u \cdot (\mathbf{D}_1 + \mathbf{D}_2) = (s_1 + s_2) \cdot \mathbf{A}_v + \mathbf{E}_1 + \mathbf{E}_2 \pmod q.$$

Similarly, two encodings $\mathbf{D}_1$ and $\mathbf{D}_2$ relative to path $u \rightsquigarrow v$ and $v \rightsquigarrow w$ can be multiplied to get an encoding relative to path $u \rightsquigarrow w$. Namely given:

$$\begin{aligned} \mathbf{A}_u \cdot \mathbf{D}_1 &= s_1 \cdot \mathbf{A}_v + \mathbf{E}_1 \pmod q \\ \mathbf{A}_v \cdot \mathbf{D}_2 &= s_2 \cdot \mathbf{A}_w + \mathbf{E}_2 \pmod q \end{aligned}$$

we obtain by multiplying the matrix encodings $\mathbf{D}_1$ and $\mathbf{D}_2$:

$$\begin{aligned} \mathbf{A}_u \cdot \mathbf{D}_1 \cdot \mathbf{D}_2 &= (s_1 \cdot \mathbf{A}_v + \mathbf{E}_1) \cdot \mathbf{D}_2 \pmod q \\ &= s_1 \cdot s_2 \cdot \mathbf{A}_w + s_1 \cdot \mathbf{E}_2 + \mathbf{E}_1 \cdot \mathbf{D}_2 \pmod q \\ &= s_1 \cdot s_2 \cdot \mathbf{A}_w + \mathbf{E}' \pmod q \end{aligned}$$

for some new error vector $\mathbf{E}'$. Since s_1, $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathbf{D}_2$ have small coefficients, $\mathbf{E}'$ still has small coefficients (compared to q), and therefore the product $\mathbf{D}_1 \cdot \mathbf{D}_2$ is an encoding of $s_1 \cdot s_2$ for the path $u \rightsquigarrow w$.

Zero-testing. The procedure $\mathsf{ZeroTest}(\mathsf{pp}, \mathbf{D})$, for an encoding $\mathbf{D}$ relative to the path $u \rightsquigarrow v$, returns true if and only if $\|\mathbf{A}_u \cdot \mathbf{D}\| < q/2^{t+1}$. The justification of that procedure is that $\mathbf{A}_u \cdot \mathbf{D} = s \cdot \mathbf{A}_v + \mathbf{E}$ is small for $s = 0$, but not otherwise (because the matrix $\mathbf{A}_v$ itself is not small).

Extraction. The correctness of zero-testing shows that the t most significant bits of $\mathbf{D}$ depend only on the underlying plaintext s. Therefore, we can carry out the extraction procedure by applying a randomness extractor to those t bits.

Chapter 3

Overview of Known Attacks

3.1 Zeroizing attacks: breaking Diffie–Hellman key exchange over GGH13 and CLT13

3.1.1 Notation and attack goals

The GGH13 and CLT13 schemes share a very similar structure; here we summarize the common features that are used in the attacks:

- Each encoding is "associated" with the vector of small integers in the numerator. For GGH13 this is a 1-vector consisting of a single algebraic integer,and for CLT13 this is a vector of n integers in $\mathbb{Z}$. Below we write informally $u \sim (a_1, \ldots, a_n)$ to denote the fact that the encoding u is associated with the vector of a_i's. Roughly speaking, the goal of the attacks is to recover the vector $(a_j)_j$ from the encoding u. Recovering this vector (even if not in full) is usually considered a break of the scheme.

- An encoding of zero is associated with a vector divisible by the g_j's, namely $u \sim (g_j r_j)_j$ for some r_j's.

- Addition and multiplication of encodings acts entry-wise on the vector of integers in the numerator. Importantly, the addition and multiplication of these vectors is done *over the integers, with no modular reduction.* This is because a wrap-around in these operations is an error condition, and so the parameters are always set to ensure that it does not happen.

- If $u \sim (g_j r_j)_j$ is an encoding of zero at the top level, then applying the zero-test to u yields the integer $w = \sum_j r_j \rho_j$, where the r_j's are the multipliers from the numerator vector and the ρ_j's are system parameters independent of u.

  In other words, applying the zero-test to an encoding of zero yields the inner-product of the associated vector (without the g_j's) with a fixed secret vector. (In GGH13 this is the 1-vector (h), in CLT13 the vector is $(p_j^* h_j)_j$, where we denote $p_j^* = x_0/p_j = \prod_{i \neq j} p_i$). Importantly, here too the inner product is over the integers, with no modular reduction.

3.1.2 Weak-DL attack on GGH13 and the Hu–Jia attack

The first published attack against the GGH13 scheme appears in the original paper itself [GGH13a]. It considers the following setting. Suppose one gets a level-t encoding of zero $u_0 \sim (gr)$ and many other level-$(\kappa - t)$ encodings $u_m \sim (a_m)$. Multiplying u_0 by any of the u_m's yields a top-level encoding of zero $u_0 u_m \sim (gra_m)$, and applying the zero-test yields the algebraic integer $w_m = hra_m$. Note that this almost recovers the numerators a_m's; indeed we have them up to the common factor $h' = hr$.

If we also knew the ideal $I_g = gR$ that defines the plaintext space, then being able to recover the numerator up to a constant is enough to break many hardness assumptions. For example, given an encoded matrix we could compute its determinant (modulo I_g) up to a constant, which would tell us whether or not the encoded matrix has full rank.

Typically, however, the ideal I_g is not explicitly given. Even in that case, however, Garg et al. described how it can be recovered in certain cases using GCD computations. Roughly, we can use GCD to identify and remove the common factor h', thereby getting the a_m's themselves, except that these are all algebraic integers so we only have GCD in terms of their ideals. Recovering the ideal $I_a = aR$ is not always useful, e.g., if I_a and I_g are co-prime then knowing I_a does not tell us anything about our plaintext coset $a + I_g$. However if some of the u_i's are themselves encodings of zero, namely $a_i = gr_i$, then given enough ideals $I_{a_i} = gr_i R$ we could again use GCD calculations to recover the ideal I_g itself, and then use that knowledge to attack the non-zero encodings among the u_i's. This attack was called a "weak discrete-log attack" in [GGH13a]. It is easily seen to break the multilinear analogue of assumptions like subgroup membership: see [GGH13a, §4.2].

3.1.3 The zeroizing attack of Cheon et al.

In [CHL+15], Cheon, Han, Lee, Ryu and Stehlé describe a major extension of the GGH13 zeroizing attack, which can be used to *completely break* multiparty Diffie–Hellman key agreement over CLT13, and more generally any CLT13-based scheme in which a similar family of low-level encodings of zero are available. The attack recovers the factorization of x_0, and then all secret information.

To mount the zeroizing attack of Cheon et al. [CHL+15], one needs three sets of encoded inputs, which we denote by $\mathscr{A} = \{A_i : i = 1, \ldots, n\}$, $\mathscr{B} = \{B_0, B_1\}$, and $\mathscr{C} = \{C_j : j = 1, \ldots, n\}$ (with n the dimension of the numerator vectors). The A's are all random encoding of zeros, the B's are the target of the attack, and the C's are just helper encodings of random vectors. The levels of these encodings are such that multiplying $A_i \cdot B_\sigma \cdot C_j$ yields a top-level encoding of zero for any i, σ, j. Below we denote the numerator vectors associated with these encodings by

$$A_i \sim (g_1 r_{i,1}, \ldots, g_n r_{i,n}), \; B_\sigma \sim (b_{\sigma,1}, \ldots, b_{\sigma,n}), \text{ and } C_j \sim (c_{j,1}, \ldots, c_{j,n}).$$

Multiplying $A_i \cdot B_\sigma \cdot C_j$ yields a top-level encoding of zero, associated with the vector $A_i \cdot B_\sigma \cdot C_j \sim (g_1 r_{i,1} b_{\sigma,1} c_{j,1}, \ldots, g_n r_{i,n} b_{\sigma,n} c_{j,n})$. Applying the zero-test we get a four-wise inner product, yielding the integer $w_\sigma[i,j] = \sum_{k=1}^{n} \rho_k r_{i,k} b_{\sigma,k} c_{j,k}$. We can write this four-wise inner product in matrix form as

$$w_\sigma[i,j] = (r_{i,1} \ \dots \ r_{i,n}) \times \begin{pmatrix} \rho_1 b_{\sigma,1} & & \\ & \ddots & \\ & & \rho_n b_{\sigma,n} \end{pmatrix} \times \begin{bmatrix} c_{j,1} \\ \vdots \\ c_{j,n} \end{bmatrix},$$

and denote the vector on the left by $\mathbf{a}_i$, the matrix in the middle by B'_σ, and the vector on the right by $\mathbf{c}_j$. For a fixed σ, let i,j range over $1,\dots,n$. This yields an $n \times n$ matrix of integers $W_\sigma = [w_\sigma[i,j]]_{i,j} = A' \times B'_\sigma \times C'$, where A' has the $\mathbf{a}_i$'s for rows and C' has the $\mathbf{c}_j$'s for columns. Since the $r_{i,k}$'s, $b_{\sigma,k}$'s, $c_{j,k}$'s and ρ_k's are all random (small) quantities, then with high probability the matrices are all invertible (over the rationals). Having computed the matrices W_σ, the attacker now sets

$$W = W_0 \times W_1^{-1} = (A'B'_0C') \times (A'B'_1, C')^{-1} = A' \times (B'_0 \times {B'_1}^{-1}) \times {A'}^{-1}.$$

Observe now that $B^* = B'_0 \times {B'_1}^{-1}$ is a diagonal matrix with $b_{0,j}/b_{1,j}$ on the diagonal, and thus the eigenvalues of B^* are all the ratios $b_{0,j}/b_{1,j}$. And since W and B^* are similar matrices, then also the eigenvalues of W are the $b_{0,j}/b_{1,j}$'s. Hence once it computes W, the attacker can find its eigenvalues (over the rationals) and obtain all the ratios $b_{0,j}/b_{1,j}$.

These ratios may be enough by themselves to break some hardness assumptions, but for CLT13 it is possible to use them to factor x_0, thereby getting a complete break. Specifically, since each ratio is rational it can be written as $u/v = b_{0,j}/b_{1,j}$ with u,v co-prime integers. Recalling now that B_0, B_1 are two encodings at the same level (say, level t) with numerator vectors $(b_{0,1},\dots,b_{0,n})$ and $(b_{1,1},\dots,b_{1,n})$, respectively, we get that

$$uB_1 - vB_0 = [\mathrm{CRT}\left(ub_{1,1} - vb_{0,1},\dots,ub_{1,n} - vb_{0,n}\right)/z^t]_{x_0}.$$

This means that the j-th CRT component is $ub_{1,j} - vb_{0,j} = 0$, and with high probability the others are not, so we get $\gcd(x_0, uB_1 - vB_0) = p_j$.

3.1.4 The attack of Hu and Jia

The attack of Cheon et al. [CHL+15] relies crucially on the fact that CLT13 is defined over the integers, and on the fact that finding the factorization of x_0 suffices to break the scheme. These aspects have no counterpart in GGH13 setting, and therefore the attack does not apply (although it does apply to a *matrix variant* of GGH13: see [CGH+15]).

However, it turns out that the GGH13 version of multiparty Diffie–Hellman key exchange is *also* insecure. This was shown by Hu and Jia [HJ16], using another attack that expands upon the weak-DL attack above. One can sum up the attack as follows.

An eavesdropper in Diffie–Hellman key exchange sees encodings $u_i = e_i y + \rho_{i0} x_0 + \rho_{i1} x_1$, $0 \le i \le \kappa$, where x_0, x_0, y are level-1 encodings of $0, 0, 1$ respectively, and the e_i and ρ_{ij} are small. The secret derived by the parties is obtained from the most significant bits of $p_{\mathrm{zt}} \cdot \prod u_i$, or equivalently $h/g \cdot \prod e_i$.

The first step of Hu and Jia's attack is the weak-DL computation. Applying zero-testing to $u_i \cdot x_0 \cdot y^{\kappa-2}$, one gets:

$$v_i = p_{\mathrm{zt}} \cdot u_i \cdot x_0 \cdot y^{\kappa-2} \bmod q = e_i b_0 h + \xi_i g$$

for some small ξ_i (without modular reduction). Similarly, applying zero-testing to $x_0 \cdot y^{\kappa-1}$, one obtains:

$$\tilde{v} = p_{\text{zt}} \cdot x_0 \cdot y^{\kappa-1} \bmod q = b_0 h + \tilde{\xi} g$$

again without modular reduction. As a result, $\tilde{v}^{-1} \cdot v_i \equiv e_i \pmod{I_g}$, where I_g is the principal ideal generated by g, as above. If we denote by w_i a representative of $\tilde{v}^{-1} \cdot v_i \bmod I_g$, then the product:

$$\eta := \prod_{i=0}^{\kappa} w_i \equiv \prod_{i=0}^{\kappa} e_i \pmod{I_g}.$$

Thus, there exists some $\zeta_0 \in R$ such that $\eta = \prod e_i + \zeta g$. This element ζ, however, is not a priori small, so we cannot directly solve the problem by taking the most significant bits of η. Instead, Hu and Jia introduce some auxiliary zero-test values as follows:

$$\begin{aligned} X_i &= p_{\text{zt}} \cdot x_1 x_i y^{\kappa-2} \mod q = h(1+ag)^{\kappa-2} b_1 b_i g \\ Y &= p_{\text{zt}} \cdot x_1 y^{\kappa-1} \mod q = h(1+ag)^{\kappa-1} b_1 g. \end{aligned}$$

Then $Y \cdot \eta$ is congruent to $Y \cdot \prod e_i$ modulo $b_1 g$, and since X_1 is a multiple of $b_1 g$, $\eta' = Y \cdot \eta \bmod X_0$ is also congruent to $Y \cdot \prod e_i$ modulo $b_1 g$. Thus:

$$y/x_1 \cdot \eta' \bmod q = p_{\text{zt}} \cdot y^{\kappa} \cdot \prod e_i + \zeta' \cdot (1+ag) \bmod q$$

for some small ζ'. Thus, we have computed $h/g \cdot \prod e_i +$ small error $\bmod q$, which breaks the Diffie–Hellman key exchange as required.

3.1.5 Other zeroizing attacks

Following the attack of Cheon et al. [CHL+15], several papers attempted to modify the CLT13 construction in order to protect against the attack. However, the modified variants turned out to be vulnerable to extensions of the same attack.

This includes in particular the "immunization technique" suggested by Boneh, Wu and Zimmerman [BWZ14] and the countermeasure proposed by Garg, Gentry, Halevi and Zhandry in [GGHZ14, §7], both of which can be broken by essentially extending the dimension of the matrices involved in Cheon et al.'s attack by a small factor, as described in [CLT14, CGH+15]. This also includes the CLT15 graded encoded scheme, proposed by Coron, Lepoint and Tibouchi in [CLT15], which was broken soon after it was published by Cheon, Fouque, Lee, Minaud and Ryu [CFL+16], again using a simple extension of Cheon et al.'s attack.

3.2 Graph-induced cryptanalysis: breaking GGH15 key exchange

Diffie–Hellman key exchange is also insecure over GGH15 multilinear maps. The protocol was broken by Coron, Lee, Lepoint and Tibouchi [CLLT16a], both in the basic case and when additional security defenses are implemented. Their attack also breaks the graph-induced variant of GGH13. We give a description of the basic attack below.

3.2.1 GGH15-based multiparty Diffie–Hellman

We first recall the structure of the multiparty Diffie–Hellman key exchange protocol over GGH15 [GGH15]. We consider the protocol with k users. As illustrated in Figure 3.1 for $k = 3$ users, each user i for $1 \leq i \leq k$ has a directed path of vectors $\mathbf{A}_{i,1}, \ldots, \mathbf{A}_{i,k+1}$, all sharing the same end-point $\mathbf{A}_0 = \mathbf{A}_{i,k+1}$. The i-th user will use the resulting chain to extract the session key. Each user i has a secret exponent s_i. Each secret exponent s_i will be encoded in each of the k chains; the encoding of s_i on the j-th chain for $j \neq i$ will be published, while the encoding of s_i on the i-th chain will be kept private by user i. Therefore on the i-th chain only user i will be able to compute the session key. The exponents s_i are encoded in a "round robin" fashion; namely the i-th secret s_i is encoded on the chain of user j at edge $\ell = i + j - 1$, with index arithmetic modulo k. Only the vectors $\mathbf{A}_{i,1}$ for $1 \leq i \leq k$ are made public to enable extraction of the session-key; the others are kept private.

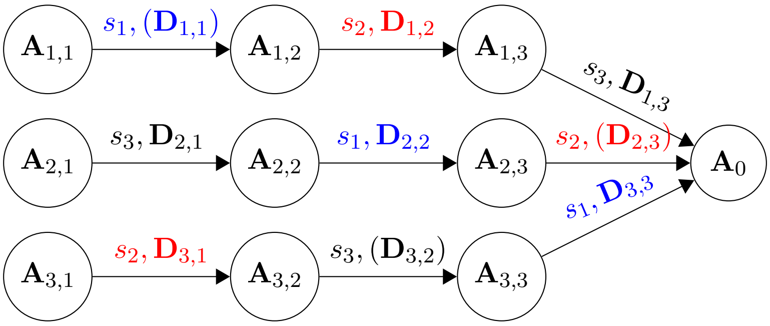

Figure 3.1: Graph of a key agreement between 3 parties for GGH15. The vertices contain random vectors $\mathbf{A}_{ij}$, and encodings are represented on the edges. Each party is represented by a different color, keeps the encoding in parenthesis secret and publishes the two other encodings.

3.2.2 The attack of Coron et al.

In [CLLT16a], Coron et al. show how an eavesdropper can recover the secret key derived by the parties in the previous protocol in polynomial time. The attack proceeds in two steps:

1. As a first step, the attacker will express one secret exponent s_1 as a linear combination of the other secret exponents $t_{1,\ell}$, using a variant of the attack of Cheon et al. [CHL+15]. However this does not immediately break the protocol, because the coefficients of the linear combination are not small.

2. In the second step, which can be seen as a generalization of the techniques of Hu and Jia [HJ16], the attacker will compute an equivalent of the private encoding of

User 1 from the previous linear combination, by correcting the error due to the large coefficients. This breaks the key-exchange protocol.

We now describe the first step in the particular case of 3 users, which illustrates the main ideas of the attack while avoiding some technical complications. We refer to [CLLT16a] for the whole details. In the 3-user case, we have the following relations:

$$\begin{aligned}
\mathbf{A}_{1,1}\cdot\mathbf{D}_{1,1} &= s_1\cdot\mathbf{A}_{1,2}+\mathbf{F}_{1,1} \pmod q & \mathbf{A}_{1,1}\cdot\mathbf{C}_{1,1,\ell} &= t_{1,\ell}\cdot\mathbf{A}_{1,2}+\mathbf{E}_{1,1,\ell} \pmod q\\
\mathbf{A}_{1,2}\cdot\mathbf{D}_{1,2} &= s_2\cdot\mathbf{A}_{1,3}+\mathbf{F}_{1,2} \pmod q & \mathbf{A}_{1,2}\cdot\mathbf{C}_{1,2,\ell} &= t_{2,\ell}\cdot\mathbf{A}_{1,3}+\mathbf{E}_{1,2,\ell} \pmod q\\
\mathbf{A}_{1,3}\cdot\mathbf{D}_{1,3} &= s_3\cdot\mathbf{A}_{0}+\mathbf{F}_{1,3} \pmod q & \mathbf{A}_{1,3}\cdot\mathbf{C}_{1,3,\ell} &= t_{3,\ell}\cdot\mathbf{A}_{0}+\mathbf{E}_{1,3,\ell} \pmod q\\[1ex]
\mathbf{A}_{2,1}\cdot\mathbf{D}_{2,1} &= s_3\cdot\mathbf{A}_{2,2}+\mathbf{F}_{2,1} \pmod q & \mathbf{A}_{2,1}\cdot\mathbf{C}_{2,1,\ell} &= t_{3,\ell}\cdot\mathbf{A}_{2,2}+\mathbf{E}_{2,1,\ell} \pmod q\\
\mathbf{A}_{2,2}\cdot\mathbf{D}_{2,2} &= s_1\cdot\mathbf{A}_{2,3}+\mathbf{F}_{2,2} \pmod q & \mathbf{A}_{2,2}\cdot\mathbf{C}_{2,2,\ell} &= t_{1,\ell}\cdot\mathbf{A}_{2,3}+\mathbf{E}_{2,2,\ell} \pmod q\\
\mathbf{A}_{2,3}\cdot\mathbf{D}_{2,3} &= s_2\cdot\mathbf{A}_{0}+\mathbf{F}_{2,3} \pmod q & \mathbf{A}_{2,3}\cdot\mathbf{C}_{2,3,\ell} &= t_{2,\ell}\cdot\mathbf{A}_{0}+\mathbf{E}_{2,3,\ell} \pmod q\\[1ex]
\mathbf{A}_{3,1}\cdot\mathbf{D}_{3,1} &= s_2\cdot\mathbf{A}_{3,2}+\mathbf{F}_{3,1} \pmod q & \mathbf{A}_{3,1}\cdot\mathbf{C}_{3,1,\ell} &= t_{2,\ell}\cdot\mathbf{A}_{3,2}+\mathbf{E}_{3,1,\ell} \pmod q\\
\mathbf{A}_{3,2}\cdot\mathbf{D}_{3,2} &= s_3\cdot\mathbf{A}_{3,3}+\mathbf{F}_{3,2} \pmod q & \mathbf{A}_{3,2}\cdot\mathbf{C}_{3,2,\ell} &= t_{3,\ell}\cdot\mathbf{A}_{3,3}+\mathbf{E}_{3,2,\ell} \pmod q\\
\mathbf{A}_{3,3}\cdot\mathbf{D}_{3,3} &= s_1\cdot\mathbf{A}_{0}+\mathbf{F}_{3,3} \pmod q & \mathbf{A}_{3,3}\cdot\mathbf{C}_{3,3,\ell} &= t_{1,\ell}\cdot\mathbf{A}_{0}+\mathbf{E}_{3,3,\ell} \pmod q
\end{aligned}$$

where all encodings $\mathbf{C}_{i,j,\ell}$ and $\mathbf{D}_{i,j}$ are public, except $\mathbf{D}_{1,1}$ which is private on Row 1, $\mathbf{D}_{2,3}$ is private on Row 2, and $\mathbf{D}_{3,2}$ is private on Row 3. The corresponding graph is illustrated in Figure 3.1. Note that on each row we have used the same index ℓ for $t_{1,\ell}$, $t_{2,\ell}$ and $t_{3,\ell}$, but on a given row one can obviously compute product of encodings for different indices.

In the first step of the attack, we show that we can express s_1 as a linear combinations of the $t_{1,\ell}$'s. For this we consider the rows 2 and 3, for which the encodings $\mathbf{D}_{2,2}$ and $\mathbf{D}_{3,3}$ corresponding to s_1 are public. In the remaining of the attack, we always consider a fixed index $\ell = 1$ for the encodings corresponding to $t_{3,\ell}$, and for simplicity we write $t_3 := t_{3,1}$, $\mathbf{C}_{1,3} := \mathbf{C}_{1,3,1}$, $\mathbf{C}_{2,1} := \mathbf{C}_{2,1,1}$ and $\mathbf{C}_{3,2} := \mathbf{C}_{3,2,1}$.

Since we always work with the same t_3, on Row 2 we define the product encodings $\hat{\mathbf{C}}_{2,2,\ell} := \mathbf{C}_{2,1}\cdot\mathbf{C}_{2,2,\ell}$, and on Row 3 we define the product encodings $\hat{\mathbf{C}}_{3,2,\ell} := \mathbf{C}_{3,1,\ell}\cdot\mathbf{C}_{3,2}$; recall that we use a fixed index for t_3. Therefore we can write:

$$\begin{aligned}
\mathbf{A}_{2,1}\cdot\hat{\mathbf{C}}_{2,2,\ell} &= t_{1,\ell}\cdot t_3\cdot\mathbf{A}_{2,3}+\hat{\mathbf{E}}_{2,2,\ell} \pmod q \qquad (3.1)\\
\mathbf{A}_{2,3}\cdot\mathbf{C}_{2,3,\ell} &= t_{2,\ell}\cdot\mathbf{A}_{0}+\mathbf{E}_{2,3,\ell} \pmod q\\
\mathbf{A}_{3,1}\cdot\hat{\mathbf{C}}_{3,2,\ell} &= t_{2,\ell}\cdot t_3\cdot\mathbf{A}_{3,3}+\hat{\mathbf{E}}_{3,2,\ell} \pmod q\\
\mathbf{A}_{3,3}\cdot\mathbf{C}_{3,3,\ell} &= t_{1,\ell}\cdot\mathbf{A}_{0}+\mathbf{E}_{3,3,\ell} \pmod q
\end{aligned}$$

for some small error vectors $\hat{\mathbf{E}}_{2,2,\ell}$ and $\hat{\mathbf{E}}_{3,2,\ell}$.

For simplicity of notations, we first consider a fixed index i for the encodings corresponding to $t_{1,i}$, and we write $t_1 := t_{1,i}$, $\hat{\mathbf{C}}_{2,2} := \hat{\mathbf{C}}_{2,2,i}$ and $\mathbf{C}_{3,3} := \mathbf{C}_{3,3,i}$. Similarly we consider a fixed index j for the encodings corresponding to $t_{2,j}$ and we write $t_2 := t_{2,j}$, $\mathbf{C}_{2,3} := \mathbf{C}_{2,3,j}$ and $\hat{\mathbf{C}}_{3,2} := \hat{\mathbf{C}}_{3,2,j}$. We use similar notations for the corresponding error vectors.

All previous equations hold modulo q only. To get a result over R instead of only modulo q, we compute the difference between two rows, for the same product of secret exponents. More precisely, we compute:

$$\begin{aligned}\omega &= \mathbf{A}_{2,1} \cdot \hat{\mathbf{C}}_{2,2} \cdot \mathbf{C}_{2,3} - \mathbf{A}_{3,1} \cdot \hat{\mathbf{C}}_{3,2} \cdot \mathbf{C}_{3,3} \qquad (3.2)\\ &= t_1 \cdot t_3 \cdot t_2 \cdot \mathbf{A}_0 + t_1 \cdot t_3 \cdot \mathbf{E}_{2,3} + \hat{\mathbf{E}}_{2,2} \cdot \mathbf{C}_{2,3}\\ &\quad - t_2 \cdot t_3 \cdot t_1 \cdot \mathbf{A}_0 - t_2 \cdot t_3 \cdot \mathbf{E}_{3,3} - \hat{\mathbf{E}}_{3,2} \cdot \mathbf{C}_{3,3}\\ &= t_1 \cdot t_3 \cdot \mathbf{E}_{2,3} + \hat{\mathbf{E}}_{2,2} \cdot \mathbf{C}_{2,3} - t_2 \cdot t_3 \cdot \mathbf{E}_{3,3} - \hat{\mathbf{E}}_{3,2} \cdot \mathbf{C}_{3,3}\,. \qquad (3.3)\end{aligned}$$

Namely the latter equation holds over R (and not only modulo q) because all the terms in (3.3) have small coefficients; namely the only term $t_1 \cdot t_2 \cdot t_3 \cdot \mathbf{A}_0$ with large coefficients modulo q is canceled when doing the subtraction.

We have that ω is a vector of dimension m. Now an important step is to restrict ourselves to the first component of ω. Namely in order to apply the same technique as in Cheon et al.'s attack, we would like to express ω as the product of two vectors, where the left vector corresponds to User 1 and the right vector corresponds to User 2. However due to the "round-robin" fashion of exponent encodings, for this we would need to swap the product $\hat{\mathbf{E}}_{3,2} \cdot \mathbf{C}_{3,3}$ appearing in (3.3), since $\hat{\mathbf{E}}_{3,2}$ corresponds to User 2 while $\mathbf{C}_{3,3}$ corresponds to User 1; this cannot be done if we consider the full vector ω. By restricting ourselves to the first component of ω, the product $\hat{\mathbf{E}}_{3,2} \cdot \mathbf{C}_{3,3}$ becomes a simple scalar product that can be swapped; namely the scalar product of $\hat{\mathbf{E}}_{3,2}$ by the first column vector $\mathbf{C}'_{3,3}$ of the matrix $\mathbf{C}_{3,3}$. We obtain the scalar:

$$\omega = t_1 \cdot t_3 \cdot E_{2,3} + \hat{\mathbf{E}}_{2,2} \cdot \mathbf{C}'_{2,3} - t_2 \cdot t_3 \cdot E_{3,3} - \mathbf{C}'_{3,3} \cdot \hat{\mathbf{E}}_{3,2}$$

where $\mathbf{C}'_{2,3}$ and $\mathbf{C}'_{3,3}$ are the first column vectors of $\mathbf{C}_{2,3}$ and $\mathbf{C}_{3,3}$ respectively, both of dimension m; similarly $E_{2,3}$ and $E_{3,3}$ are the first components of $\mathbf{E}_{2,3}$ and $\mathbf{E}_{3,3}$ respectively.

We can now write ω as the scalar product of 2 vectors, the left one corresponding only to User 1, and the right one corresponding only to User 2:

$$\omega = \begin{bmatrix} t_1 & \hat{\mathbf{E}}_{2,2} & E_{3,3} & \mathbf{C}'_{3,3} \end{bmatrix} \cdot \begin{bmatrix} t_3 \cdot E_{2,3} \\ \mathbf{C}'_{2,3} \\ -t_2 \cdot t_3 \\ -\hat{\mathbf{E}}_{3,2} \end{bmatrix}.$$

Note that the two vectors in the product have dimension $2m + 2$.

As in the attack of Cheon et al. [CHL+15], we can now extend ω to a matrix by considering many left row vectors and many right column vectors. However instead of a square matrix as in Cheon et al.'s attack, we consider a rectangular matrix with $2m + 3$ rows and $2m + 2$ columns. In (3.2), this is done by considering $2m + 3$ public encodings $\hat{\mathbf{C}}_{2,2,i}$ and $\mathbf{C}_{3,3,i}$ corresponding to User 1, and similarly $2m+2$ encodings $\mathbf{C}_{2,3,j}$ and $\hat{\mathbf{C}}_{3,2,j}$ corresponding to User 2, for $1 \le i \le 2m + 3$ and $1 \le j \le 2m + 2$. More precisely we compute as previously over R the following matrix elements, restricting ourselves to the first component:

$$(\mathbf{W})_{ij} = \mathbf{A}_{2,1} \cdot \hat{\mathbf{C}}_{2,2,i} \cdot \mathbf{C}'_{2,3,j} - \mathbf{A}_{3,1} \cdot \hat{\mathbf{C}}_{3,2,j} \cdot \mathbf{C}'_{3,3,i} \qquad (3.4)$$

and as previously we can write:

$$(\mathbf{W})_{ij} = \begin{bmatrix} t_{1,i} & \hat{\mathbf{E}}_{2,2,i} & E_{3,3,i} & \mathbf{C}'_{3,3,i} \end{bmatrix} \cdot \begin{bmatrix} t_3 \cdot E_{2,3,j} \\ \mathbf{C}'_{2,3,j} \\ -t_{2,j} \cdot t_3 \\ -\hat{\mathbf{E}}_{3,2,j} \end{bmatrix} .$$

We obtain a $(2m+3) \times (2m+2)$ matrix $\mathbf{W}$ with:

$$\mathbf{W} = \underbrace{\begin{bmatrix} & & \dots & \\ t_{1,i} & \hat{\mathbf{E}}_{2,2,i} & E_{3,3,i} & \mathbf{C}'_{3,3,i} \\ & & \dots & \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} & t_3 \cdot E_{2,3,j} & \\ \vdots & \mathbf{C}'_{2,3,j} & \vdots \\ & -t_{2,j} \cdot t_3 & \\ & -\hat{\mathbf{E}}_{3,2,j} & \end{bmatrix}}_{\mathbf{B}}$$

where the matrix $\mathbf{A}$ has $2m+3$ rows vectors, each of dimension $2m+2$, and the matrix $\mathbf{B}$ has $2m+2$ column vectors, each of dimension $2m+2$; hence $\mathbf{B}$ is a square matrix.

By doing linear algebra, we can find a vector $\mathbf{u}$ over R of dimension $2m+3$ such that $\mathbf{u} \cdot \mathbf{W} = 0$, which gives:

$$(\mathbf{u} \cdot \mathbf{A}) \cdot \mathbf{B} = 0 .$$

Heuristically with good probability the matrix $\mathbf{B}$ is invertible, which implies:

$$\mathbf{u} \cdot \mathbf{A} = 0 .$$

Since the first column of the matrix $\mathbf{A}$ is the column vector given by the $t_{1,i}$'s, such vector $\mathbf{u}$ gives a linear relation among the secret exponents $t_{1,i}$.

Moreover, since the encodings $\mathbf{D}_{2,2}$ and $\mathbf{D}_{3,3}$ corresponding to s_1 are public, we can express s_1 as a linear combination of the $t_{1,i}$'s, over R. Namely we can define as previously the product encoding $\hat{\mathbf{D}}_{2,2} := \mathbf{C}_{2,1} \cdot \mathbf{D}_{2,2}$, with:

$$\mathbf{A}_{2,1} \cdot \hat{\mathbf{D}}_{2,2} = s_1 \cdot t_3 \cdot \mathbf{A}_{2,3} + \hat{\mathbf{F}}_{2,2} \pmod q$$

for some small error vector $\hat{\mathbf{F}}_{2,2}$, and we can now compute the same $(\mathbf{W})_{ij}$ as in (3.4) but with $\hat{\mathbf{D}}_{2,2}$ and $\mathbf{D}'_{3,3}$ instead of $\hat{\mathbf{C}}_{2,2,i}$ and $\mathbf{C}'_{3,3,i}$, where $\mathbf{D}'_{3,3}$ is the first column of $\mathbf{D}_{3,3}$. More precisely, we compute for all $1 \leq j \leq 2m+2$:

$$\omega_j = \mathbf{A}_{2,1} \cdot \hat{\mathbf{D}}_{2,2} \cdot \mathbf{C}'_{2,3,j} - \mathbf{A}_{3,1} \cdot \hat{\mathbf{C}}_{3,2,j} \cdot \mathbf{D}'_{3,3}$$

which gives as previously:

$$\omega_j = \begin{bmatrix} s_1 & \hat{\mathbf{F}}_{2,2} & F_{3,3} & \mathbf{D}'_{3,3} \end{bmatrix} \cdot \begin{bmatrix} t_3 \cdot E_{2,3,j} \\ \mathbf{C}'_{2,3,j} \\ -t_{2,j} \cdot t_3 \\ -\hat{\mathbf{E}}_{3,2,j} \end{bmatrix} .$$

This implies that we can replace any row vector $[t_{1,i}\ \hat{\mathbf{E}}_{2,2,i}\ E_{3,3,i}\ \mathbf{C}'_{3,3,i}]$ in the matrix $\mathbf{A}$ by the row vector:

$$[s_1\ \ \hat{\mathbf{F}}_{2,2}\ \ F_{3,3}\ \ \mathbf{D}'_{3,3}] \tag{3.5}$$

where $\mathbf{D}'_{3,3}$ is the first column of $\mathbf{D}_{3,3}$, and $F_{3,3}$ is the first component of $\mathbf{F}_{3,3}$. Using the previous technique, we can therefore obtain a linear relation between s_1 and the $t_{1,i}$'s over R. More precisely, with overwhelming probability, such a relation can be put in the form:

$$\mu \cdot s_1 = \sum_{i=1}^{2m+2} \lambda_i \cdot t_{1,i} \tag{3.6}$$

with $\mu \in \mathbb{Z}$ and $\lambda_1, \dots, \lambda_{2m+2} \in R$. Indeed, we obtain such a relation by computing the kernel of the matrix analogous to $\mathbf{W}$ above in echelon form over the fraction field of R, which gives the kernel of the corresponding matrix $\mathbf{A}$ (assuming that $\mathbf{B}$ is invertible). Unless a minor of that matrix vanishes, which happens with only negligible probability, this gives a relation where the coefficient of s_1 is 1 and the other coefficients are in the fraction field $R \otimes_{\mathbb{Z}} \mathbb{Q}$ of R. By clearing denominators, we get an expression of the form (3.6).

Then, by considering exactly one additional $t_{1,i}$ (say $t_{1,2m+3}$) and carrying out the same computations with indices $i = 2, \dots, 2m+3$ instead of $i = 1, \dots, 2m+2$, we get a second relation:

$$\nu \cdot s_1 = \sum_{i=2}^{2m+3} \lambda'_i \cdot t_{1,i}\,.$$

If the integers μ and ν are relatively prime, which happens with significant probability, we can apply Bézout's identity to obtain a linear relation in R where the coefficient of s_1 is 1:

$$s_1 = \sum_{i=1}^{2m+3} \alpha_i \cdot t_{1,i}, \tag{3.7}$$

which completes this first attack step and our description.

3.3 Attacks on obfuscation

We conclude this chapter by a short discussion of proposed attacks against constructions of indistinguishability obfuscation. Since a precise description of the constructions themselves exceeds the scope of this document, we simply give a very rough idea of what the attacks can achieve and of their limitations, without any attempt to provide technical details, for which we refer to the corresponding papers.

3.3.1 Attacks against obfuscation over CLT13

Candidate constructions of indistinguishability obfuscation from multilinear maps (aside from more recent techniques via functional encryption) can be broadly divided into two types: one the one hand, obfuscation for *branching programs*, that rely on Barrington's theorem to obfuscate circuits, and on the other hand, circuit-obfuscation constructions, that work directly on circuits. Limited attacks exist on both types of schemes when instantiated over CLT13.

Attacks on branching program obfuscation. Recall that a branching program is a collection of pairs $\mathbf{A}_{i,0}, \mathbf{A}_{i,1}$, $1 \leq i \leq t$, of $d \times d$ square matrices together with some input assignment function $\mathsf{inp}\colon \{1,\dots,t\} \to \{1,\dots,n\}$. It computes the Boolean function on n inputs bits given by:

$$f(x_1,\dots,x_n) = \begin{cases} 1 & \text{if } \prod_{i=1}^{t} \mathbf{A}_{i,x_{\mathsf{inp}(i)}} = \mathbf{I}_d \\ 0 & \text{otherwise} \end{cases}.$$

Roughly speaking, branching program obfuscation candidates such as the one described by Garg et al. in [GGH+13b] work by taking such a branching program, randomizing it using Kilian's technique, increasing the dimension of the matrices with some diagonal padding, and then encoding the expanded randomized matrices element-wise using a multilinear map.

In [CGH+15], Coron et al. showed that such a construction provides the necessary data to apply the attack of Cheon et al. [CHL+15] in the particular case when the branching program has a *decomposable structure*, i.e. when the successive matrices can be divided into three groups, each depending on a different subset of input bits. This attack does not apply to actual obfuscation candidates, however, because the branching programs produced by Barrigton's theorem never have the required decomposable structure.

However, Coron, Lee, Lepoint and Tibouchi [CLLT16b, CLLT17] later showed how to dramatically expand the scope of this attack, and make it practically relevant to an actual obfuscation candidates when applied to a large class of functions. The key idea of their attack is the observation that the order of matrices in a branching program can be rearranged in an essentially arbitrary way by taking tensor products, at the cost of increasing the dimension. This can be used to force branching programs into a decomposition suitable to apply the previous attack, at least if the blow-up in matrix dimension is not too large.

As a result, they were able to break the original candidate obfuscator from [GGH+13b] when instantiated over CLT13 multilinear maps, as well as the so-called *single-input* versions of many subsequent candidate obfuscators, including [MSW14, AGIS14, PST14, BGK+14, BMSZ16], again over CLT13. A surprising feature of the attack of [CLLT16b, CLLT17] is that, assuming the existence of certain classes of pseudorandom functions computed by branching programs of short length, it can also break the obfuscator described in [GMM+16], which is proved secure in the *weak multilinear map model*, a model that was believed to capture all known classes of attacks on multilinear map constructions.

The attack of [CLLT16b, CLLT17] can only target functions satisfying a property called input-partionability, however. And soon after the attack was made public, Fernando, Rasmussen and Sahai proposed a generic countermeasure to protect against all attacks of that nature [FRS16]. It works by adding to the input of all functions a "signature structure" that prevents input-partionability.

Attacks on circuit-obfuscation. Coron et al. [CGH+15] also showed that the attack of Cheon et al. can be extended to partially break the circuit-obfuscation schemes of Zimmerman [Zim15] and Applebaum–Brakerski [AB15]. More precisely, the two papers present a "simple" scheme (which is essentially the same in both papers) and more advanced variants (which differ between the two papers). Coron et al. target the simple scheme, and show that an attack similar to Cheon at al.'s can be applied to that scheme when obfuscating simple enough circuits, such as point functions.

Note that this simple scheme uses so-called "composite-order" multilinear maps, which cannot be instantiated over GGH13, so a CLT13-based instantiation is the only possible concrete instantiation of that scheme known so far, and it is partially broken. However, the more advanced versions are not shown to be vulnerable.

3.3.2 Attacks against obfuscation over GGH13 and GGH15

Setting aside obfuscation candidates relying on composite order multilinear maps (which cannot be instantiated over GGH13), the first attack against indistinguishability obfuscators over GGH13 was the *annihilation attack* introduced by Miles, Sahai and Zhandry in [MSZ16a]. It is conceptually different from zeroizing attacks.

The attack targets a family of obfuscator that Miles et al. describe axiomatically, and that captures in particular the constructions from [MSW14, AGIS14, PST14, BGK+14, BMSZ16]. One way of describing the general idea of the attack is to note that the zero-testing values ω_i arising from the evaluation a given branching program are ring elements that can be expressed as polynomials $\sum_j f_{ij}(r_1, \dots, r_\ell) g^j$ on the error factors r_k involved in encodings. And for a different but functionally equivalent branching program, one will find polynomials $f'_{ij} \neq f_{ij}$ in general. One can then for instance distinguish between two different but functionally equivalent branching programs by finding a polynomial relation Q between the $f_{i,0}$'s (i.e. $Q(f_{1,0}, \dots, f_{m,0}) = 0$). Indeed, such a relation will ensure that $Q(\omega_1, \dots, \omega_m)$ is always in the ideal I_g for the first branching program, whereas this will typically happen with only negligible probability for the second branching program, for which the polynomial relation does not hold.

The attack of [MSZ16a] even applies to the dual-input versions of the schemes mentioned above over GGH13. However, several subsequent multilinear map constructions have been proved to be secure against this class of attack [GMS16, MSZ16b].

More recently, several extensions of the attack of [MSZ16a] have been proposed. Chen, Gentry and Halevi [CGH16] show how to break the original obfuscation candidate of Garg et al. [GGH+13b] over GGH13 using annihilation attacks. They also combine the annihilation technique with the attack of [CLLT16a] to break the construction of obfuscation over GGH15 multilinear maps [GGH15]. Like [CLLT17], however, these attacks are limited to input partitionable functions, and can thus be thwarted using the techniques of [FRS16]. One can also mention the work of Apon et al. [ADGM16], which introduces an efficiently testable condition for breaking obfuscation over GGH13 using annihilation attacks, and uses it to attack a larger class of branching programs than [MSZ16a]. The constructions that are provably secure against annihilation attacks [GMS16, MSZ16b] remain unaffected, however.

Chapter 4

Conclusions and Perspectives

4.1 Status of multilinear map-based primitives

We present a snapshot of the current security status of major primitives based on multilinear maps at the time of this writing. The situation is likely to evolve rapidly, however. For an up-to-date overview of current results, we refer the reader to Martin Albrecht's excellent resource entitled "Are Graded Encoding Schemes Broken Yet?" [Alb16].

Multiparty Diffie–Hellman key exchange. Over all proposed multilinear map candidates, multiparty Diffie–Hellman key exchange is broken. Over CLT13 multilinear maps, it was broken by Cheon et al. [CHL⁺15], and later attempts to protect against the attack [BWZ14, GGHZ14, CLT15] were also broken by extensions of that attack [CLT14, CGH⁺15, CFL⁺16]. Over GGH13 multilinear maps, it was broken by Hu and Jia [HJ16], and that attack also applies to the optimized versions proposed in [LSS14, ACLL15]. Finally, over GGH15 multilinear maps, it was broken by Coron et al. [CLLT16a].

Indistinguishability obfuscation. Attacks have been demonstrated against *some* candidate constructions of indistinguishability obfuscation over each of GGH13, CLT13 and GGH15, but *not everything* is broken. More precisely, the annihilation attack of Miles, Sahai and Zhandry [MSZ16a] and its extensions [CGH16, ADGM16] broke almost all indistinguishability obfuscators over GGH13 existing at the time, but later on, constructions were proposed that are provably secure against it [GMS16, MSZ16b]. The zeroizing attack of Coron et al. [CLLT16b, CLLT17] broke almost all indistinguishability obfuscators over CLT13 existing at the time in the *single-input* setting. However, dual-input constructions are unaffected. Moreover, the attack only affects input-partitionable functionalities, and can thus be thwarted using the generic countermeasure of Fernando, Rasmussen and Sahai [FRS16]. Finally, Chen, Gentry and Halevi [CGH16] obtained an attack indistinguishability obfuscation over GGH15 [GGH15]. However, this attack also applies to input-partitionable functionalities only, and is thus thwarted by [FRS16].

Other primitives. No specific cryptanalytic work so far has examined the security of other multilinear map-based primitives like witness encryption. However, a reasonable expectation is that constructions relying on *secret-key* graded encodings, like witness encryption, are likely to behave similarly to indistinguishability obfuscation, whereas constructions relying on *public-key* encodings, like some constructions of ABE, are likely to fall prey to the same kind of attacks as Diffie–Hellman key exchange.

Note also that indistinguishability obfuscation is sufficient to obtain provably secure multilinear maps [AFH+16], so theoretically speaking, as long as indistinguishability obfuscation exists, everything, including multiparty Diffie–Hellman, can be instantiated securely. Of course, the efficiency of such a giant pyramid construction is guaranteed to be atrocious.

4.2 Future prospects

As we have seen, the whole multilinear map edifice is standing on shaky ground, and its security situation is quite precarious. Further progress on the cryptanalytic side is likely, and could easily bring about the unravelling of the last few remaining candidate constructions of indistinguishability obfuscation. And even if one believes that those schemes will stay secure, it is fair to say that the current situation, in which we have to rely on multilinear map constructions that were found to not even satisfy their original, basic security definition, is unsatisfactory. Progress is also being made on the construction side, however, and it could ultimately yield to much more solid foundations. This could come from several directions.

Indistinguishability obfuscation. The conditions needed to obtain indistinguishability obfuscation are becoming less and less stringent. Although early candidate constructions required multilinear maps with polynomially large degrees satisfying very strong security assumptions, this has recently been reduced to conservative assumptions over n-linear maps for n as low as 5 [Lin16, AS16]. If n could be reduced further, one might eventually be able to dispense with multilinear maps altogether and obtain everything from pairings.

Functional encryption. Compact functional encryption for relatively limited classes of functions would also suffice to obtain indistinguishability obfuscation and hence everything else. And current techniques are not very far off from achieving it from LWE [GKP+13, GVW15].

New multilinear maps. Since low-degree multilinear maps are now known to suffice for indistinguishability obfuscation, and hence essentially all applications, geometry-based techniques, which were originally ruled out by Boneh and Silverberg, might be usefully revisited, as has been done on a few occasions [RH09].

In any event, the field of multilinear maps can certainly expect many interesting developments in the months and years to come.

Bibliography

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 528–556, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[ABD16] Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[ABDP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 733–751, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 657–677, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[ACLL15] Martin R. Albrecht, Catalin Cocis, Fabien Laguillaumie, and Adeline Langlois. Implementing candidate graded encoding schemes from ideal lattices. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 752–775, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany.

[ADGM16] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. Cryptology ePrint Archive, Report 2016/1003, 2016. http://eprint.iacr.org/2016/1003.

[AFH+16] Martin R. Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, and Kenneth G. Paterson. Multilinear maps from obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 446–473, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

[AGIS14] Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 14*, pages 646–658, Scottsdale, AZ, USA, November 3–7, 2014. ACM Press.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[AJN+16] Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 491–520, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[Alb16] Martin Albrecht. Are graded encoding schemes broken yet? Regularly updated webpage, 2016. http://malb.io/are-graded-encoding-schemes-broken-yet.html.

[AS16] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. Cryptology ePrint Archive, Report 2016/1097, 2016. http://eprint.iacr.org/2016/1097.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.

[BDOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[BF03] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BGI+10] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6:1–6:48, 2010.

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 221–238, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer, Heidelberg, Germany.

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 470–491, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[BMSZ16] Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 764–791, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Boy08] Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56, Egham, UK, September 1–3, 2008. Springer, Heidelberg, Germany.

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

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer, Heidelberg, Germany.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. Cryptology ePrint Archive, Report 2014/930, 2014. http://eprint.iacr.org/2014/930.

[CCK$^+$13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 315–335, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[CFL$^+$16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 509–536, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[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. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 247–266, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[CGH16] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. Cryptology ePrint Archive, Report 2016/998, 2016. http://eprint.iacr.org/2016/998.

[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, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[CLLT16a] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 607–628, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[CLLT16b] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. Cryptology ePrint Archive, Report 2016/1011, 2016. http://eprint.iacr.org/2016/1011.

[CLLT17] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. In *PKC 2017*, LNCS. Springer, Heidelberg, Germany, 2017. To appear.

[CLT13a] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[CLT13b] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. Cryptology ePrint Archive, Report 2013/183, 2013. http://eprint.iacr.org/2013/183.

[CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. Cryptology ePrint Archive, Report 2014/975, 2014. http://eprint.iacr.org/2014/975.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 267–286, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[FRS16] Rex Fernando, Peter M. R. Rasmussen, and Amit Sahai. Preventing CLT zeroizing attacks on obfuscation. Cryptology ePrint Archive, Report 2016/1070, 2016. http://eprint.iacr.org/2016/1070.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[GGH12] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. Cryptology ePrint Archive, Report 2012/610, 2012. http://eprint.iacr.org/2012/610.

[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, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[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, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.

[GGH+13c] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure attribute based encryption from multilinear maps. Cryptology ePrint Archive, Report 2014/622, 2014. http://eprint.iacr.org/2014/622.

[GGHZ16] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

[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, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[GLW14] Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 426–443, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[GMM+16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 241–268, Beijing, China, October 31 – November 3, 2016. Springer, Heidelberg, Germany.

[GMS16] Sanjam Garg, Pratyay Mukherjee, and Akshayaram Srinivasan. Obfuscation without the vulnerabilities of multilinear maps. Cryptology ePrint Archive, Report 2016/390, 2016. http://eprint.iacr.org/2016/390.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GPSW06a] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.

[GPSW06b] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. Cryptology ePrint Archive, Report 2006/309, 2006. http://eprint.iacr.org/2006/309.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 194–213, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.

[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, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[Ham11] Mike Hamburg. Spatial encryption. Cryptology ePrint Archive, Report 2011/389, 2011. http://eprint.iacr.org/2011/389.

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 537–565, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004.

[KM10] Neal Koblitz and Alfred Menezes. The brave new world of bodacious assumptions in cryptography. *Notices of the American Mathematical Society*, 57:357–365, 2010.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.

[Lin16] Huijia Lin. Indistinguishability obfuscation from DDH on 5-linear maps and locality-5 PRGs. Cryptology ePrint Archive, Report 2016/1096, 2016. http://eprint.iacr.org/2016/1096.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *57th FOCS*, pages 11–20, New Brunswick, NJ, USA, October 9–11, 2016. IEEE Computer Society Press.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[MSW14] Eric Miles, Amit Sahai, and Mor Weiss. Protecting obfuscation against arithmetic attacks. Cryptology ePrint Archive, Report 2014/878, 2014. http://eprint.iacr.org/2014/878.

[MSZ16a] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[MSZ16b] Eric Miles, Amit Sahai, and Mark Zhandry. Secure obfuscation in a weak multilinear map model: A simple construction secure against all known attacks. Cryptology ePrint Archive, Report 2016/588, 2016. http://eprint.iacr.org/2016/588.

[MVO93] Alfred Menezes, Scott A. Vanstone, and Tatsuaki Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004.

[OT09] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer, Heidelberg, Germany.

[PRV12] Periklis A. Papakonstantinou, Charles W. Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? Cryptology ePrint Archive, Report 2012/653, 2012. http://eprint.iacr.org/2012/653.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 500–517, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[RAD78] Ron L. Rivest, Leonard M. Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, editor, *Foundations of Secure Computation*, pages 169–180. Academic Press, 1978.

[RH09] Wayne Raskind and Ming-Deh Huang. A multilinear generalization of the Tate pairing. Talk at the Fq9 conference, University College Dublin, July 2009. https://maths.ucd.ie/~gmg/Fq9Talks/Raskind.pdf.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 579–598, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 24–43, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[Wat15] Brent Waters. A punctured programming approach to adaptively secure functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 678–697, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 439–467, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.