58th Annual IEEE Symposium on Foundations of Computer Science

Lockable Obfuscation

Rishab Goyal
Department of Computer Science
The University of Texas at Austin
Austin, USA
Email: rgoyal@cs.utexas.edu

Venkata Koppula
Department of Computer Science
The University of Texas at Austin
Austin, USA
Email: kvenkata@cs.utexas.edu

Brent Waters
Department of Computer Science
The University of Texas at Austin
Austin, USA
Email: bwaters@cs.utexas.edu

***Abstract*—In this paper we introduce the notion of lockable obfuscation. In a lockable obfuscation scheme there exists an obfuscation algorithm Obf that takes as input a security parameter, a program P, a message msg and "lock value" lck and outputs an obfuscated program oP. One can evaluate the obfuscated program oP on any input x where the output of evaluation is the message msg if P(x) = lck and otherwise receives a rejecting symbol.**

We proceed to provide a construction of lockable obfuscation and prove it secure under the Learning with Errors (LWE) assumption. Notably, our proof only requires LWE with polynomial hardness and does not require complexity leveraging.

We follow this by describing multiple applications of lockable obfuscation. First, we show how to transform any attribute-based encryption (ABE) scheme into one in which the attributes used to encrypt the message are hidden from any user that is not authorized to decrypt the message. (Such a system is also know as predicate encryption with one-sided security.) The only previous construction due to Gorbunov, Vaikuntanathan and Wee is based off of a specific ABE scheme of Boneh et al. By enabling the transformation of any ABE scheme we can inherent different forms and features of the underlying scheme such as: multi-authority, adaptive security from polynomial hardness, regular language policies, etc.

We also show applications of lockable obfuscation to separation and uninstantiability results. We first show how to create new separation results in circular encryption that were previously based on indistinguishability obfuscation. This results in new separation results from learning with error including a public key bit encryption scheme that it IND-CPA secure and not circular secure. The tool of lockable obfuscation allows these constructions to be almost immediately realized by translation from previous indistinguishability obfuscation based constructions.

In a similar vein we provide random oracle uninstantiability results of the Fujisaki-Okamoto transformation (and related transformations) from the lockable obfuscation combined with fully homomorphic encryption. Again, we take advantage that previous work used indistinguishability obfuscation that obfuscated programs in a form that could easily be translated to lockable obfuscation.

I. Introduction

The topic of indistinguishability obfuscation has received an tremendous amount of attention from the cryptographic community over the last several years. Initially, the concept was introduced by Barak et al. [1], [2] as an possible alternative to the notion of virtual black box obfuscation which they showed to be impossible to achieve for some functionalities. However, the concept indistinguishability obfuscation did not receive much immediate attention since (1) there were no such obfuscation candidates at the time and (2) the perceived lack of applications due to the fact that it only guaranteed security between two functionally equivalent circuits.

In 2013, two works in the literature addressed these questions. First, Garg et al. [3] provided the first indistinguishability obfuscation candidate using the Garg, Gentry and Halevi [4] multilinear map candidate. Then Sahai and Waters [5] introduced the "punctured programming" methodology for building cryptographic primitives from indistinguishability obfuscation which was used in their work and several subsequent works to resolve many open problems in cryptography.

When the potential of indistinguishability obfuscation was exposed, attention naturally moved to establishing security of obfuscation candidates since the original work of Garg et al. [3] only provided a heuristic argument of security. Initial work in this line attempted to prove security under certain multilinear map models or assumptions [6], [7], [8], [9], [10]. However, the security guarantees delivered from such proofs could only be as strong as the underlying multilinear map candidates [4], [11], [12], [13] which have been under a steady stream of cryptanalysis (see e.g. [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26] and the references therein). To combat this there have been new multilinear map candidates proposed as well as models meant to capture most existing attacks [27], [28]. While these techniques present progress in defense against currently known cryptanalysis, it is unclear whether they can be connected to standard assumptions. Another set of works [29], [30], [31], [32], [33], [34], [35] have shown connections between certain types of functional encryption schemes and indistinguishability obfuscation with results showing that a constant degree multilnear maps combined with a constant depth PRG give indistinguishability obfuscation.

In this paper we approach the problem of achieving provably secure obfuscation from a different direction. Our philosophy is to anchor ourselves to the Learning with Errors (LWE) assumption and explore what applications and forms

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.62

of obfuscation are achievable. Here we propose and define a new form of obfuscation that we call lockable obfuscation for which we give a construction that is provably secure under the LWE assumption. In addition, we show several applications of lockable obfuscation that were only known to this point under indistinguishability obfuscation.

We begin by informally introducing the notion of lockable obfuscation. A lockable obfuscation scheme consists of an obfuscation and evalution algorithms. The (randomized) obfuscation algorithm Obf takes as input a security parameter λ, a program P, a message msg and lock value α and outputs an obfuscated program $\widetilde{P}$. The evaluation algorithm Eval takes as input an obfuscated program $\widetilde{P}$ and an input x. If $P(x) = \alpha$ then the evaluation algorithm outputs the message msg, where P, msg and α were the program, message and lock values input to create $\widetilde{P}$. Otherwise, if $P(x) \neq \alpha$ the evaluation algorithm (with high probability) outputs the $\perp$ symbol to indicate rejection.

Intuitively, security states that if the lock value is chosen at random and kept from the attacker, then the attacker cannot learn anything about the program and message other than their sizes. That is there exist a simulator Sim such that for all program message pairs P, msg

$$\{\mathsf{Obf}(1^\lambda, P, \mathsf{msg}, \alpha) : \alpha \leftarrow \{0,1\}^{m(\lambda)}\} \approx_c \mathsf{Sim}(1^\lambda, 1^{|P|}, 1^{|\mathsf{msg}|}).$$

We show how to construct a lockable obfuscation scheme for any polynomial sized circuit of sufficient output length from the Learning with Errors assumption. Our construction relies only on the polynomial hardness of the assumption (i.e. unlike witness encryption/indistinguishability obfuscation constructions, there is no sub-exponential hardness or complexity leveraging involved). [1] For this reason lockable obfuscation could be a preferred abstraction for building certain primitives even in a possible future where indistinguishability obfuscation is realizable from the assumption of LWE with subexponential hardness. The reason we don't require subexponential hardness of LWE is that the security of our construction is derived from the hidden lock value α. In particular, our proof of security does *not* step through each possible input like many reductions to witness encryption or indistinguishability obfuscation [9], [10], [29], [30], [31]. Also, note that since the lock value is chosen at random (and independent of the program), therefore lockable obfuscation could also be interpreted as obfuscation for a family of evasive functions [36].

We will defer the explanation of our construction and proof to the technical overview of Section I-A and move on to discussing applications.

[1]Note that we still require subexponential LWE modulus. This translates to a subexponential approximation factor in the worst case hardness of lattice problems.

Hiding Attributes in Attribute-Based Encryption: In a (key-policy) Attribute-Based Encryption (ABE) system [37] a user will encrypt a message msg under an attribute string x. A private key, as issued by an authority, will be associated with a boolean function f. When a user holding a secret key for function f attempts to decrypt a ciphertext he will learn the message if and only if $f(x) = 1$. Otherwise, the message remains hidden. While an ABE ciphertext will securely hide a private message, its security definition provides no guarantees about hiding the attribute string x. This can be problematic in many practical scenarios. Suppose we use ABE to encrypt email messages and use header metadata such as the sender, recipients, time and subject as attributes, where access functions are written over these attributes. In many settings this metadata itself will be very sensitive and disclosing it as part of the ciphertext is undesirable.

Almost all expressive[2] ABE systems built from standard tools[3] allow for an attacker to learn the attribute string associated with a ciphertext. One notable exception is the work of Gorbunov, Vaikuntanathan and Wee [40] that achieves one-sided security in hiding attributes. In that their construction hides the attribute string x as long as $f(x) = 0$ for all functions f that the adversary has a secret key for. This gives a much improved security picture to before as in our example an attacker will not be able to learn the header metadata of the emails so long as they are not authorized to decrypt them. We will call such a scheme a predicate encryption scheme with one-sided security.

The GVW construction was built by adapting the ABE system of Boneh et al. [41] and required an intricate knowledge of the original system in order to utilize certain special mathematical properties. The resulting construction achieves the same functionality of bounded depth circuits as the original as well as maintains selective security under the Learning with Errors assumption.

In this paper we show how to use lockable obfuscations (sometimes in combination with fully homomorphic encryption) to generically transform *any* ABE scheme into a predicate encryption scheme with one-sided security with corresponding functionality. An advantage of using a generic transformation is that it can take advantages of the features or forms of ABE constructions that have been introduced over the last 10+ years. including key-policy [42], ciphertext-policy [43], adaptive security (without complexity leveraging) [44], multi-authority [45], [46], [47], and efficient expression of keys as deterministic finite automata [48], [49] and circuits [50], [41]. Currently, there is no single ABE construction from standard tools that simultaneously

[2]We note that there are constructions of more limited expressivity such as vector matching [38] or inner product testing [39] that achieve such security.

[3]For the purposes of this discussion we roughly consider number theoretic constructions grounded on RSA, bilinear maps and (Ring) LWE to be standard tools and those based on multilinear maps or indistinguishability obfuscation not to be.

delivers all such features. The most desirable ABE form can vary significantly between applications. Our transformations will allow one to inherent the properties of the underlying ABE scheme and at the same obtain one-sided hiding of the ciphertext attributes.

The technique for hiding attributes is fairly simple. To perform a (key-policy) encryption to attributes x and message msg the encryption algorithm first chooses a random lock value α. Next it creates a (sub) ciphertext C which is an encryption of the message α under the attribute string x in the original ABE system. Now consider the program P which takes as input a secret key and then decrypts C and outputs the result. The final ciphertext ct is a lockable obfuscation of this program P, under the lock value α and message msg. Key generation is the same as in the original scheme and decryption is simply using the lockable obfuscation evaluation algorithm on the ct with the secret key as the input.

Correctness can be observed. Suppose a user has a secret key sk_f for function f and applies it to a ciphertext encryption with attributes x where $f(x) = 1$). The evaluation algorithm will output msg since $P(\mathsf{sk}_f) = \alpha$, which is the lock value. On the other hand suppose that the attacker does not have any secret keys for a function f where $f(x) = 1$, then by the message hiding security of the underling ABE scheme, he won't be able to distinguish an encryption of the lock value α from an encryption of the all 0's string. Now that the lock value is hidden, one can take an additional hybrid step to argue that this is indistinguishable from a simulated obfuscation to erase knowledge of the program P including C and the attribute string x.

The above construction works for any scheme with an apriori bounded size decryption circuit. By adding a level of indirection one can leverage leveled homomorphic encryption (which is realizable under LWE [51], [52]) to upgrade this to any scheme with an apriori decryption circuit depth. Or leverage fully homomorphic encryption to work without any depth bounds. The details of the transformation are given in the full version.

Separation and Uninstantiability Results: We now demonstrate the power of lockable obfuscation for achieving negative results in cryptography by focusing on two families of separation and uninstantiability results.

In recent years there has been a significant interest on the problem of circular security [53], [54], [55] perhaps in large part due to Gentry's [56] result showing that a leveled homomorphic encryption scheme that is circular secure implies unbounded fully homomorphic encryption. Roughly, an encryption system is circular secure if it maintains security in the setting where there are n pairs of public keys and ciphertexts arranged such that the i^{th} ciphertext encrypts the $(i+1)^{th}$ secret key.

There have been several separation results [57], [58], [59], [60], [61], [62], [63], [64], [65], [66] showing that such security does not come for free. In particular, they show in different contexts that there exist schemes that are IND-CPA secure, but not circular secure. (We discuss this prior work in detail in the full version). A natural dichotomy of these results is between separations achieved using indistinguishability obfuscation [60], [65] and those built from standard assumptions [57], [58], [62], [63], [64], [66] such as assumptions on bilinear groups or LWE.

Separations built on indistinguishability obfuscation have the advantage that they can be developed relatively rapidly and are also relatively simple as one can build constructions using "normal" programs without diving into number theory. The disadvantage is that indistinguishability is not currently known from any standard assumptions. On the flip side, the number theoretic constructions are based on much more standard assumptions. However, they developing and understanding such solutions is a much more arduous task.

In the full version, we show how to translate existing two indistinguishability obfuscation separation results due to Koppula, Ramchen and Waters [60] and one result due Goyal, Koppula and Waters [65] to rely on lockable obfuscation. The translations are extremely straightforward and our resulting solutions are almost identical with the exception that we use lockable obfuscation. The main insight is that the programs obfuscated in the above results come in a lockable friendly form. In particular, they perform a sequence of computations on the input that result in a value s. Then the program tests and reports if $\mathrm{PRG}(s) = t$ for a pseudorandom generator PRG and hardwired value t. Using lockable obfuscation one simply uses s as the output (which is a possible lock value).

Our translations of these prior works to lockable obfuscation lead to separations that can be built on the Learning with Errors assumption. Concretely, we obtain new results from LWE that were previously known only from indistinguishability obfuscation. (1) We show how to build a *public key* bit encryption scheme that is not circular secure and (2) we show a separation for unbounded length cycles.

Our second family of negative results relate to a grouping of constructions related to the well known Fujisaki-Okamoto transformation that achieves chosen ciphertext security in the random oracle model from any scheme which is IND-CPA secure. Included in this grouping are: the Bellare et al. [67] transformation from an IND-CPA scheme to an injective trapdoor function, two transformations from IND-CPA to IND-CCA security due to Fujisaki and Okamoto [68], [69] and the deterministic encryption construction of Bellare, Boldyreva and O'Neill [70].

All of the constructions follow a similar paradigm where they encrypt a string x under random coins determined from $H(x)$. (How the string x is construed varies somewhat between the schemes.) The works above show that if H is presented as an oracle access to a random function, the transformation results in a secure scheme under the relevant

definition.

We give a random oracle uninstantiability [71] result where using lockable obfuscation there exists an encryption scheme where for any hash function of up to apriori bounded size the applying the above transformations will result in an insecure encryption scheme — the message will be easily discovereable. If we add the assumption of fully homomorphic encryption we can remove the bounded size restriction.

Brzuska, Farshim and Mittelbach [72] achieved these results using indistinguishability obfuscation.[4] We realize our results by simply translating the BFM result to move from indistinguishability obfuscation to lockable obfuscation. Again, this is possible because the programs obfuscated in the BFM paper follow the same lockable friendly form.

Indistinguishability Obfuscation for Rejecting Programs: For our final application we now consider a new notion of obfucation that we call indistinguishability obfuscation for rejecting programs and show how to construct it from lockable obfuscation and witness encryption [73] for circuit satisfiability.

Obfuscators that meet this notion will be defined over boolean circuits. Like indistinguishability obfuscation our obfuscator will take in any (not necessarily rejecting) boolean circuit C in a class and output an obfuscated program that is functionally equivalent to C. However, the security guarantees given by such an obfuscator are limited to "rejecting" programs. Informally, they state that no PPT adversary can distinguish between circuits C_0 and C_1 so long as for all inputs x $C_0(x) = C_1(x) = 0$. In contrast, standard indistinguishability obfuscation security allows C_0, C_1 to have arbitrary (both 0 and 1) outputs so long as they are functionally equivalent.

Our construction is simple and follows along the same conceptual lines as our techniques for building predicate encryption with one sided security from Attribute-Based Encryption.

Concurrent Work: In an independent and concurrent work, Wichs-Zirdelis proposed a similar notion called *Obfuscation for Compute-and-Compare Programs*. While the notions are very similar, the syntax is different. A compute-and-compare program $CC[f, y]$ is defined by a function f and a value y. Obfuscation of $CC[f, y]$ outputs a program P such that P, on any input x, outputs 1 if $f(x) = y$. For security, [74] require that for a randomly chosen y (from a high-entropy distribution), the obfuscation of $CC[f, y]$ is indistinguishable from simulated obfuscation, where the simulator gets only size of f and y. Wichs-Zirdelis also extend this notion where the obfuscation algorithm also takes as input a message m together with f, y, and the obfuscated program P on input x, outputs m if $f(x) = y$. Security requires that the message m is also hidden. This message-based version of obfuscation of compute-and-compare programs is identical to our notion of lockable obfuscation, modulo the distribution of lock y, which is uniform in our case and can be any high-entropy distribution in their case. However, this gap can be simply bridged by using an PRG for pseudo-entropy seeds instead of a regular PRG.

Both works have a few applications in common. This includes the ABE to predicate encryption transformation, witness encryption to reject-iO and circular security counterexamples. In addition, we show new uninstantiability results in the random oracle model. Applications unique to [74] are transformation from any secure sketch [75] to private secure sketch [76], and obfuscation for conjunctions/affine testers.

A. Overview of our Lockable Obfuscation Construction

We will now describe our lockable obfuscation scheme for a family of poly-depth circuits. The construction is described in detail in the full version. At a high level, our scheme can be divided into three components — (1) A lockable obfuscation scheme for a family of low-depth circuits and 1-bit messages, (2) a bootstrapping mechanism to amplify to lockable obfuscation for a family of poly-depth circuits and 1-bit messages, and (3) extending to lockable obfuscation for a family of poly-depth circuits and multi-bit messages. (Note in our actual construction, we combine the first two components into one for technical reasons.)

Lockable Obfuscation for Low-Depth Circuits and 1-Bit Messages.: The primary ingredients of our construction are low-depth pseudorandom generators (PRGs), lattice trapdoors [77], telescoping products/cascading cancellations [64], [63], [12] and oblivious sequence transformation [66].

First, let us recall the notion of permutation branching programs, lattice trapdoors and oblivious sequence transformation. A permutation branching program of length L and width w can be represented using w states, $2L$ permutations $\sigma_{j,b}$ over states for each level $j \leq L$, an input-selector function $\mathsf{inp}(\cdot)$ which determines the input read at each level, and an accepting and rejecting state. The program execution starts at state 1 of level 0. Suppose the branching program reads first input bit (say b) at level 1 (i.e., $\mathsf{inp}(1) = 1$). Then, the state of the program changes to $\sigma_{1,b}(\mathsf{st})$. Such a process can be carried out (iteratively) to compute the final program state at level L. Depending upon the final state, the program either accepts or rejects.

A lattice trapdoor generation algorithm can be used to sample a (uniformly looking) matrix $\mathbf{A}$ together with a trapdoor $T_{\mathbf{A}}$. The trapdoor can be used to compute, for any

[4] Technically, their result with no bounds on the hash function required indistinguishability obfuscation for Turing Machines with unbounded input. However, this could have been replaced with indistinguishability obfuscation for circuits and fully homomorphic encryption.

matrix $\mathbf{U}$, a low norm matrix $\mathbf{S}$ such that $\mathbf{A} \cdot \mathbf{S} = \mathbf{U}$.[5] As a result, the matrix $\mathbf{S}$ can be used to 'transform' any matrix $\widetilde{\mathbf{A}} \approx \mathbf{A}$ to another matrix $\widetilde{\mathbf{U}} \approx \mathbf{U}$. Oblivious sequence transformation is a technique that enables sampling a sequence of matrices $\mathbf{B}_1, \ldots, \mathbf{B}_w$ along with some trapdoor such that for any sequence of matrices $\mathbf{U}_1, \ldots, \mathbf{U}_w$, one can construct a short matrix $\mathbf{X}$ such that $\mathbf{B}_i \cdot \mathbf{X} = \mathbf{U}_i$ for all i (and $\mathbf{X}$ is oblivious of i).

Moving on to our construction, at a high level the obfuscator starts by first generating a sequence of permutation branching programs corresponding to the circuit C (where each branching program computes one output bit), and then encoding the state transition permutations for each level for every branching program using the technique of oblivious sequence transformation. Let ℓ be the output length of C. In other words, for obfuscating circuit C under lock α with message msg, the obfuscator first expresses C as a set of width 5 permutation branching programs $\{\mathsf{BP}^{(i)}\}_i$ of polynomial length L, where for each $i \in [\ell]$ $\mathsf{BP}^{(i)}$ computes the i^{th} output bit of circuit C.[6] Without loss of generality, we can assume that all branching programs have a common input selector function $\mathsf{inp}(\cdot)$ such that $\mathsf{inp}(j)$ bit of the input is read at level j. The obfuscation algorithm continues by choosing $5(L+1)\ell$ matrices, one matrix for each (level, state) of each branching program.[7] Let $\mathbf{B}^{(i)}_{j,w}$ be the matrix corresponding to state $w \in [5]$ at level $j \leq L$ for branching program $\mathsf{BP}^{(i)}$, $i \in [\ell]$. For every $i \in [\ell]$, the matrices $\{\mathbf{B}^{(i)}_{j,1}, \ldots, \mathbf{B}^{(i)}_{j,5}\}$ for the first L levels (i.e., all but top-level matrices) are sampled such that they have a common trapdoor $T^{(i)}_j$, i.e. using oblivious sequence transformation. The top-level matrices, however, are sampled uniformly at random without a trapdoor subject to the constraint that the top-level matrices corresponding to lock string α sum to a special fixed matrix depending upon the message msg. More formally, for each $i \in [\ell]$, let q be the LWE modulus, and $\mathsf{acc}^{(i)}$ and $\mathsf{rej}^{(i)}$ be the accepting and rejecting states for $\mathsf{BP}^{(i)}$, then the obfuscator chooses the matrices $\mathbf{B}^{(i)}_{L,\mathsf{acc}^{(i)}}, \mathbf{B}^{(i)}_{L,\mathsf{rej}^{(i)}}$ such that

$$\sum_{i \,:\, \alpha_i = 0} \mathbf{B}^{(i)}_{L,\mathsf{rej}^{(i)}} + \sum_{i \,:\, \alpha_i = 1} \mathbf{B}^{(i)}_{L,\mathsf{acc}^{(i)}} = \begin{cases} \mathbf{0}^{n \times m} & \text{if } \mathsf{msg} = 0. \\ \sqrt{q} \cdot \left[\mathbf{I}_n \,||\, \mathbf{0}^{n \times (m-n)}\right] & \text{if } \mathsf{msg} = 1. \end{cases}$$

For each level $j \in [L]$, the obfuscation algorithm also chooses two low-norm matrices $\mathbf{S}^{(0)}_j$ and $\mathbf{S}^{(1)}_j$ (these are shared across all branching programs), and computes 2ℓ low-norm matrices $\{\mathbf{C}^{(i,0)}_j, \mathbf{C}^{(i,1)}_j\}_{i,j}$ such that for every state $w \in [5]$, $\mathbf{B}^{(i)}_{j-1,w} \cdot \mathbf{C}^{(i,0)}_j \approx \mathbf{S}^{(0)}_j \cdot \mathbf{B}^{(i)}_{j,\sigma^{(i)}_{j,0}(w)}$ and $\mathbf{B}^{(i)}_{j-1,w} \cdot \mathbf{C}^{(i,1)}_j \approx \mathbf{S}^{(1)}_j \cdot \mathbf{B}^{(i)}_{j,\sigma^{(i)}_{j,1}(w)}$. That is, the matrices $\mathbf{C}^{(i,0)}_j$ and $\mathbf{C}^{(i,1)}_j$ represent the state transition from level $j-1$ to j when bit 0 or 1 is read at step j of branching program execution. For each $i \in [\ell], j \in [L]$, the $\mathbf{C}^{(i,b)}_j$ matrices can be generated using the lattice trapdoors $T^{(i)}_j$. The obfuscation algorithm outputs these matrices $\{\mathbf{C}^{(i,0)}_j, \mathbf{C}^{(i,1)}_j\}_{i,j}$ together with the base-level matrices $\{\mathbf{B}^{(i)}_{0,1}\}_i$ as the final obfuscated program.

At a high level, one could visualize the obfuscated program which consists of the base-level matrices $\{\mathbf{B}^{(i)}_{0,1}\}_i$ and matrices $\{\mathbf{C}^{(i,0)}_j, \mathbf{C}^{(i,1)}_j\}_{i,j}$ as "encodings" of the branching program starting states and state transition permutations, respectively. Therefore, evaluating an obfuscated program on some input x will be analogous to evaluating the branching programs $\mathsf{BP}^{(i)}$ on input x directly. Fix some $i \in [\ell]$. Suppose the first input bit x_1 is read at level 1. Then evaluation of $\mathsf{BP}^{(i)}$ at level 1 would map the state 1 at level 0 to state $\sigma^{(i)}_{1,x_1}$ at level 1. Analogously, the obfuscation evaluator can compute $\mathbf{B}^{(i)}_{0,1} \cdot \mathbf{C}^{(i,x_1)}_1 \approx \mathbf{S}^{(x_1)}_1 \cdot \mathbf{B}^{(i)}_{1,\sigma^{(i)}_{1,x_1}}$. In general, if the program state at level $j-1$ during execution is w, then the evaluator will accumulate the product of the form $\mathbf{\Gamma}_{j-1} \cdot \mathbf{B}^{(i)}_{j,w}$, where $\mathbf{\Gamma}_{j-1}$ is a product of $j-1$ low-norm matrices. This can be easily verified as follows. Suppose the next bit read is b, then the new state at level j will be $\sigma^{(i)}_{j,b}(w)$, thus the new accumulated product during obfuscation evaluation will be $\mathbf{\Gamma}_{j-1} \cdot \mathbf{B}^{(i)}_{j,w} \cdot \mathbf{C}^{(i,b)}_j \approx \mathbf{\Gamma}_j \cdot \mathbf{B}^{(i)}_{j+1,\sigma^{(i)}_{j,b}(w)}$, where $\mathbf{\Gamma}_j = \mathbf{\Gamma}_{j-1} \cdot \mathbf{S}^{(b)}_j$. Therefore, the invariant is maintained. Note that the matrix $\mathbf{\Gamma}_L$ will be same for all branching programs since the low-norm matrices $\mathbf{S}^{(0)}_j$ and $\mathbf{S}^{(1)}_j$ are shared across all branching programs.

Continuing this way, the evaluator can iteratively compute the matrix product at the top. Thus, for each branching program, the accumulated product at the top will either be $\approx \mathbf{\Gamma}_L \cdot \mathbf{B}^{(i)}_{L,\mathsf{acc}^{(i)}}$ or $\approx \mathbf{\Gamma}_L \cdot \mathbf{B}^{(i)}_{L,\mathsf{rej}^{(i)}}$, depending on whether $C(x)_i = 0$ or 1. Let $\mathbf{\Delta}^{(i)} = \mathbf{\Gamma}_L \cdot \mathbf{B}^{(i)}_{L,\mathsf{st}^{(i)}}$, where $\mathsf{st}^{(i)} = \mathsf{acc}^{(i)}$ or $\mathsf{rej}^{(i)}$ depending on $C(x)_i$. Finally, the evaluator simply sums the top-level accumulated products ($\approx \mathbf{\Delta}^{(i)}$) and checks whether the norm of the final summed matrix lies in appropriate range. More concretely, consider the case when $C(x) = \alpha$ and $\mathsf{msg} = 0$, then $\sum_i \mathbf{\Delta}^{(i)} = \Gamma_L \cdot \sum_i \mathbf{B}^{(i)}_{L,\mathsf{st}^{(i)}} = \mathbf{0}^{n \times m}$. Since the final top-level matrix sum is close to $\sum_i \mathbf{\Delta}^{(i)}$, thus it will have norm close to 0, and hence the evaluator can simply test this and output 0 as the message.

[5] For ease of exposition, we will use the notation $\mathbf{A}^{-1}(\cdot)$ to represent the pre-image operation. In the formal description of our algorithms later, we use the pre-image sampling algorithm $\mathsf{SamplePre}$.

[6] From Barrington's Theorem [78], we know that for every $\mathbf{NC}^1$ circuits there exists a width 5 permutation branching program of polynomial length.

[7] Note that if a branching program has length L, then it has $L+1$ levels.

Similarly we could argue correctness for the cases when $\mathsf{msg} = 1$ or $C(x) \neq \alpha$. However, our current proof techniques do not seem sufficient for proving the security of above construction. The reason is that there is an inherent tension in setting scheme parameters while basing security on LWE. This is discussed in detail later in the full version. We were able to bypass this problem by obfuscating an "expanded" circuit, which evaluates a low-depth pairwise independent hash function h and a low-depth pseudorandom generator PRG with a large enough polynomial stretch (in succession) on the output of circuit C, instead of directly obfuscating circuit C using the above matrix encoding procedure.

In other words, let Q be the circuit that on input x, outputs $\mathrm{PRG}(h(C(x)))$. Observe that if h and PRG can be computed by low-depth circuits ($\mathbf{NC}^1$), then Q also can be computed by an $\mathbf{NC}^1$ circuit (since C is assumed to be a log-depth circuit). Therefore, Q can be expressed by a set of width 5 permutation branching programs of polynomial length L as well. Additionally, now the matrix component generation procedure will use $\beta = \mathrm{PRG}(h(\alpha))$ as the lock instead of α. This modification is sufficient to avoid the tension between scheme parameters, and also allows us to prove security of our scheme under LWE with only polynomial hardness. This completes the description of our lockable obfuscation scheme for low-depth circuits and 1-bit messages.

Bootstrapping Lockable Obfuscation.: Let $\mathcal{O}_{\mathsf{NC}^1}$ be a lockable obfuscator for log-depth circuits. We will use leveled homomorphic encryption (LHE) with an $\mathbf{NC}^1$ decryption circuit to bootstrap $\mathcal{O}_{\mathsf{NC}^1}$ to an obfuscator that works for any depth d. The obfuscator gets as input a circuit C of depth d, a string α and a message msg. It first chooses the LHE secret-evaluation keys and encrypts the circuit C. Let Q be the circuit which takes as input an LHE ciphertext and decrypts it using the hardwired LHE secret key and outputs the decrypted string. Note that the circuit Q is a logarithmic depth circuit. The obfuscator outputs $\mathcal{O} \leftarrow \mathcal{O}_{\mathsf{NC}^1}(Q, \alpha, \mathsf{msg})$ together with the encryption of C and the LHE evaluation key.

Evaluating on input x. Let $U_x(\cdot)$ be the universal circuit with input x hardwired (that is, it takes a circuit C as input and outputs $C(x)$). The evaluation algorithm first homomorphically evaluates the circuit U_x on the encryption of C. This results in an LHE ciphertext ct, which is an encryption of $C(x)$. It then evaluates the obfuscation $\mathcal{O}$ on input ct, and outputs the resulting string. Using the correctness of $\mathcal{O}$ and the LHE scheme, we can argue that if $C(x) = \alpha$, then the evaluation outputs msg, and it outputs $\perp$ otherwise.

The security proof here is fairly simple. Using the security of the underlying obfuscator $\mathcal{O}_{\mathsf{NC}^1}$, we first switch the obfuscation $\mathcal{O}$ to be a simulated obfuscation. Once the obfuscator $\mathcal{O}$ is simulated, the obfuscator no longer needs the LHE secret key. Therefore, we can now replace the LHE encryption of C with encryption of zeros, thereby erasing all the information about circuit C except its size. Therefore, the final simulator simply outputs an encryption of zeros, together with a simulated obfuscation, and this is indistinguishable from the honestly computed obfuscation.

Extending Lockable Obfuscation for 1-Bit Messages to Multi-Bit Messages.: We would like point out that the standard repitition method for extending message space does not work for lockable obfuscation schemes because the security is only guaranteed when the adversary does not know the lock string α. However, we observe there are still multiple ways to extend its message space. We briefly discuss two possible extensions. One option could be to encode a multi-bit message directly in the top-level matrices. Currently, the top-level matrices are set to sum to $\mathbf{0}^{n\times m}$ if $\mathsf{msg} = 0$, otherwise to $\sqrt{q} \cdot \left[\mathbf{I}_n \,||\, \mathbf{0}^{n\times(m-n)}\right]$. However, if we interpret the mesage msg as an integer $v < \sqrt{q}/2$, then we could simply set the sum to be $v \cdot \sqrt{q} \cdot \left[\mathbf{I}_n \,||\, \mathbf{0}^{n\times(m-n)}\right]$.

The second extension could be carried more generally without exploiting the mathematical structure of the underlying obfuscation scheme. The high level idea is to again "expand" the circuit C using a pairwise independent hash function and a pseudorandom generator before obfuscation. Suppose the lock α be a string of length k. Let $\beta = \mathrm{PRG}(h(\alpha))$ and $|\beta| = \ell \cdot k$. To obfuscate circuit C under lock α for an ℓ-bit message msg, the multi-bit obfuscator (for each $i < \ell$) independently obfuscates the circuit $Q[i]$ under lock $\beta[i]$ for message msg_i using the 1-bit obfuscation scheme, where circuit $Q[i]$ denotes the circuit that outputs the $i \cdot k + 1, \ldots, (i+1) \cdot k$ output bits of circuit $\mathrm{PRG}(h(C(\cdot)))$ and $\beta[i] = \beta_{i\cdot k+1}, \ldots, \beta_{(i+1)\cdot k}$. The security proof follows from a simple hybrid argument. This transformation is described later in the full version.

This completes the technical overview of our lockable obfuscation scheme.

B. More on Related Encoding Works

The idea of using lattice trapdoors for constructing multilinear maps was first seen in the work of Gentry, Gorbunov and Halevi (GGH) where they proposed a candidate for graph-induced multilinear maps [12]. Their work builds upon the homomorphic encryption scheme of Gentry, Sahai and Waters [79] which could be considered as the starting point of cascading cancellations technique. In [12], there is a fixed (directed acyclic) graph $G = (V, E)$, and plaintext messages are associated with edges. Each vertex u has an associated matrix $\mathbf{A}_u$ and a secret parameter T_u which is the trapdoor for $\mathbf{A}_u$. The plaintext space consists of matrices, and the encoding of a matrix $\mathbf{M}$ along an edge (u, v) is $\mathbf{A}_u^{-1}(\mathbf{M} \cdot \mathbf{A}_v + \mathsf{noise})$. Note that given an encoding of matrix $\mathbf{M}_1$ for edge (u, v) and an encoding of matrix

$\mathbf{M}_2$ for edge (v, w), one can compute an encoding of $\mathbf{M}_1 \cdot \mathbf{M}_2$ along path (u, v, w).Informally, their intuition was that given all the encodings it should be easy to compute an encoding for any path in graph G, however it should remain hard for anyone to generate encodings over any non-path combination of vertices. Using these multilinear maps, GGH gave candidate constructions for obfuscation and multipartite key-agreement. However, there were no security proofs for these candidates, and their key-agreement protocol was later shown to be broken [18]. In a later work, Brakerski et al. [80] gave a construction for obfuscating conjunctions based on an entropic variant of Ring-LWE. In that work, they observed that if the underlying graph was a straight-line graph (i.e., an ordered sequence of vertices), then the corresponding GGH multilinear maps provide some provable security properties.

In a more recent work, Koppula and Waters [64] used what they called cascading cancellations technique for constructing k-circular security separations. Their construction deviates from the GGH/BVWW paradigm in two crucial aspects. First, their construction involves multiple *strands* (say ℓ) of length k, instead of just a single strand. Unlike the previous works, which involved directly comparing two distinct encodings (or, components) along the same path for equality, they first combined (using matrix multiplications) all the k components for each different strand, and then summed the ℓ final components (one for each strand) to perform an equality check with a fixed value. Second, what is mechanically close to the "plaintext" encoded values in GGH is viewed here as simply random matrices (and just part of the overall randomness) and not a value to be encoded. This differs from GGH in which the the elements being encoded were labeled as "plaintext" values. For instance, the elements that were encoded in the GGH obfuscation candidate were the state transition matrices of the actual branching program beign obfuscated and thus can reflect some stronger semantics. Concurrently, Alamati and Peikert [63] also provided circular security separations, that had a cancellation type effect although with a different technical approach.

Most recently, Goyal, Koppula and Waters [66] further advanced the existing techniques for constructing bit-circular security separations. One of their most important contributions was an alternative mechanism to encode and *hide* (branching) programs using lattice trapdoors. To this end, they introduced a novel technique which they call oblivious sequence transformation. This is a significant departure from prior works as previously most works generated the matrix components (or encodings) independently, however it was essential in their construction that the components be jointly generated. At a high level, they provided new techniques to encode and hide a permutation between a sequence of nodes. Informally, they gave a mechanism to encode a permutation between nodes $u_1, \ldots, u_5$ and $w_1, \ldots, w_5$ such that given a node u_i one could obliviously go to its corresponding node w_j. Another important aspect of their work was to encode a log-depth Pseudo Random Generator (PRG) using the oblivious sequence transformation technique such that the PRG could be publicly evaluated and if the output of computation is some fixed (but unknown) value, then it could be efficiently tested. This seems to be one of the most important technical aspect of their work since in order to prove security from the LWE assumption as well as guarantee efficient testing, encoding a log-depth PRG with a large polynomial stretch is essential. We finally remark that Canetti and Chen [81] recently used the LWE assumption to achieve 1-collusion secure constrained PRFs for **NC1** with constraint hiding. Their construction involves a single strand like BVWW and their embedding of a branching program follows the GGH style of putting branching programs into the encoded plaintext values much more closely.

ACKNOWLEDGMENT

We thank Mihir Bellare and Joseph Jaeger for their detailed and informative feedback on our definitions and proofs.

REFERENCES

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *CRYPTO*, 2001, pp. 1–18.

[2] ——, "On the (im)possibility of obfuscating programs," *J. ACM*, vol. 59, no. 2, p. 6, 2012.

[3] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indstinguishability obfuscation and functional encryption for all circuits," in *FOCS*, 2013.

[4] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *EUROCRYPT*, 2013.

[5] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 475–484.

[6] B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai, "Protecting obfuscation against algebraic attacks," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014.

[7] Z. Brakerski and G. N. Rothblum, "Virtual black-box obfuscation for all circuits via generic graded encoding," in *Theory of Cryptography Conference*, 2014.

[8] R. Pass, K. Seth, and S. Telang, "Indistinguishability obfuscation from semantically-secure multilinear encodings," in *International Cryptology Conference*, 2014.

[9] C. Gentry, A. B. Lewko, and B. Waters, "Witness encryption from instance independent assumptions," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, 2014, pp. 426–443. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44371-2_24

[10] C. Gentry, A. B. Lewko, A. Sahai, and B. Waters, "Indistinguishability obfuscation from the multilinear subgroup elimination assumption," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 151–170. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2015.19

[11] J. Coron, T. Lepoint, and M. Tibouchi, "Practical multilinear maps over the integers," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, 2013, pp. 476–493.

[12] C. Gentry, S. Gorbunov, and S. Halevi, "Graph-induced multilinear maps from lattices," in *TCC*, 2015.

[13] J. Coron, T. Lepoint, and M. Tibouchi, "New multilinear maps over the integers," in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, 2015.

[14] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Cryptanalysis of two candidate fixes of multilinear maps over the integers," Cryptology ePrint Archive, Report 2014/975, 2014.

[15] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé, "Cryptanalysis of the multilinear map over the integers," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, 2015, pp. 3–12.

[16] J. Coron, C. Gentry, S. Halevi, T. Lepoint, H. K. Maji, E. Miles, M. Raykova, A. Sahai, and M. Tibouchi, "Zeroizing without low-level zeroes: New MMAP attacks and their limitations," in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, 2015.

[17] Z. Brakerski, C. Gentry, S. Halevi, T. Lepoint, A. Sahai, and M. Tibouchi, "Cryptanalysis of the quadratic zero-testing of GGH," *IACR Cryptology ePrint Archive*, 2015.

[18] J. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi, "Cryptanalysis of GGH15 multilinear maps," in *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, 2016.

[19] ——, "Zeroizing attacks on indistinguishability obfuscation over CLT13," in *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, 2017.

[20] D. Boneh, D. J. Wu, and J. Zimmerman, "Immunizing multilinear maps against zeroizing attacks," Cryptology ePrint Archive, Report 2014/930, 2014.

[21] Y. Hu and H. Jia, "Cryptanalysis of ggh map," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2016.

[22] S. Halevi, "Graded encoding, variations on a scheme," Cryptology ePrint Archive, Report 2015/866, 2015.

[23] J. H. Cheon, P.-A. Fouque, C. Lee, B. Minaud, and H. Ryu, "Cryptanalysis of the new clt multilinear map over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2016.

[24] E. Miles, A. Sahai, and M. Zhandry, "Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13," in *Annual Cryptology Conference*, 2016.

[25] J. H. Cheon, J. Jeong, and C. Lee, "An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low-level encoding of zero," *LMS Journal of Computation and Mathematics*, 2016.

[26] D. Apon, N. Döttling, S. Garg, and P. Mukherjee, "Cryptanalysis of indistinguishability obfuscations of circuits over ggh13," Cryptology ePrint Archive, Report 2016/1003, 2016.

[27] S. Badrinarayanan, E. Miles, A. Sahai, and M. Zhandry, "Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits," in *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, 2016.

[28] S. Garg, E. Miles, P. Mukherjee, A. Sahai, A. Srinivasan, and M. Zhandry, "Secure obfuscation in a weak multilinear map model," in *Theory of Cryptography Conference*. Springer, 2016, pp. 241–268.

[29] N. Bitansky and V. Vaikuntanathan, "Indistinguishability obfuscation from functional encryption," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 171–190.

[30] P. Ananth and A. Jain, "Indistinguishability obfuscation from compact functional encryption," in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, 2015, pp. 308–326.

[31] P. Ananth, A. Jain, and A. Sahai, "Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption," *IACR Cryptology ePrint Archive*, 2015.

[32] H. Lin, "Indistinguishability obfuscation from constant-degree graded encoding schemes," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2016.

[33] H. Lin and V. Vaikuntanathan, "Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, 2016.

[34] H. Lin, "Indistinguishability obfuscation from ddh on 5-linear maps and locality-5 prgs," Cryptology ePrint Archive, Report 2016/1096, 2016.

[35] P. Ananth and A. Sahai, "Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps," in *EUROCRYPT*, 2016.

[36] B. Barak, N. Bitansky, R. Canetti, Y. T. Kalai, O. Paneth, and A. Sahai, "Obfuscation for evasive functions," in *Theory of Cryptography Conference*, 2014.

[37] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT*, 2005, pp. 457–473.

[38] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of the 4th conference on Theory of cryptography*, ser. TCC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 535–554. [Online]. Available: http://dl.acm.org/citation.cfm?id=1760749.1760788

[39] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, ser. EUROCRYPT'08, 2008.

[40] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Predicate encryption for circuits from lwe," in *Annual Cryptology Conference*, 2015.

[41] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, pp. 533–556.

[42] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06, 2006.

[43] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, 2007, pp. 321–334.

[44] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT*, 2010, pp. 62–91.

[45] M. Chase, "Multi-authority attribute based encryption," in *TCC*, 2007, pp. 515–534.

[46] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *ACM Conference on Computer and Communications Security*, 2009, pp. 121–130.

[47] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *EUROCRYPT*, 2011, pp. 568–588.

[48] B. Waters, "Functional encryption for regular languages," in *CRYPTO*, 2012.

[49] N. Attrapadung, "Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, pp. 557–577.

[50] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," in *STOC*, 2013.

[51] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," in *FOCS*, 2011, pp. 97–106.

[52] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in *ITCS*, 2012.

[53] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," *IACR Cryptology ePrint Archive*, vol. 2001, p. 19, 2001.

[54] P. Laud, "Encryption cycles and two views of cryptography," in *NORDSEC 2002 - Proceedings of the 7th Nordic Workshop on Secure IT Systems (Karlstad University Studies 2002:31*, 2002, pp. 85–100.

[55] P. Adão, G. Bana, J. Herzog, and A. Scedrov, "Soundness and completeness of formal encryption: The cases of key cycles and partial information leakage," *Journal of Computer Security*, vol. 17, no. 5, pp. 737–797, 2009.

[56] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009.

[57] T. Acar, M. Belenkiy, M. Bellare, and D. Cash, "Cryptographic agility and its relation to circular encryption," in *EUROCRYPT '10*, vol. 6110 of LNCS. Springer, 2010, pp. 403–422.

[58] D. Cash, M. Green, and S. Hohenberger, "New definitions and separations for circular security," in *Public Key Cryptography - PKC*, 2012, pp. 540–557.

[59] R. Rothblum, "On the circular security of bit-encryption," in *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, 2013, pp. 579–598. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36594-2_32

[60] V. Koppula, K. Ramchen, and B. Waters, "Separations in circular security for arbitrary length key cycles," in *Theory of Cryptography Conference (TCC)*, 2015.

[61] A. Marcedone and C. Orlandi, "Obfuscation $\Rightarrow$ (IND-CPA security $!\Rightarrow$ circular security)," in *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, 2014, pp. 77–90.

[62] A. Bishop, S. Hohenberger, and B. Waters, "New circular security counterexamples from decision linear and learning with errors," in *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, 2015, pp. 776–800.

[63] N. Alamati and C. Peikert, "Three's compromised too: Circular insecurity for any cycle length from (ring-)lwe," in *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, 2016, pp. 659–680. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-53008-5_23

[64] V. Koppula and B. Waters, "Circular security counterexamples for arbitrary length cycles from LWE," in *CRYPTO*, 2016.

[65] R. Goyal, V. Koppula, and B. Waters, "Separating IND-CPA and circular security for unbounded length key cycles," in *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, 2017.

[66] ——, "Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption," in *EUROCRYPT*, 2017.

[67] M. Bellare, S. Halevi, A. Sahai, and S. Vadhan, "Many-to-one trapdoor functions and their relation to public-key cryptosystems," in *Annual International Cryptology Conference*, 1998.

[68] E. Fujisaki and T. Okamoto, "How to enhance the security of public-key encryption at minimum cost," in *International Workshop on Public Key Cryptography*. Springer, 1999, pp. 53–68.

[69] ——, "Secure integration of asymmetric and symmetric encryption schemes," in *CRYPTO '99*, vol. 1666 of LNCS. Springer, 1999, pp. 537–554.

[70] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in *Annual International Cryptology Conference*, 2007.

[71] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited (preliminary version)," in *STOC*, 1998, pp. 209–218.

[72] C. Brzuska, P. Farshim, and A. Mittelbach, "Random-oracle uninstantiability from indistinguishability obfuscation," in *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, 2015.

[73] S. Garg, C. Gentry, A. Sahai, and B. Waters, "Witness encryption and its applications," in *STOC*, 2013.

[74] D. Wichs and G. Zirdelis, "Obfuscating compute-and-compare programs under lwe," Cryptology ePrint Archive, Report 2017/276, 2017.

[75] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[76] Y. Dodis and A. D. Smith, "Correcting errors without leaking partial information," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, 2005, pp. 654–663.

[77] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *STOC*, 2008, pp. 197–206.

[78] D. A. Barrington, "Bounded-width polynomial-size branching programs recognize exactly those languages in nc1," in *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, ser. STOC '86, 1986.

[79] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *CRYPTO*, 2013.

[80] Z. Brakerski, V. Vaikuntanathan, H. Wee, and D. Wichs, "Obfuscating conjunctions under entropic ring lwe," in *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 2016.

[81] R. Canetti and Y. Chen, "Constraint-hiding constrained prfs for nc1 from lwe," in *EUROCRYPT*, 2017.