Lockable Obfuscation from Circularly Insecure Fully Homomorphic Encryption

Kamil Kluczniak[1,2]

[1] CISPA Helmholtz Center for Information Security
[2] Stanford University
kamil.kluczniak@{cispa.saarland,stanford.edu}

Abstract. In a lockable obfuscation scheme, a party called the obfuscator takes as input a circuit C, a lock value y, and a message m, and outputs an obfuscated circuit. Given the obfuscated circuit, an evaluator can run it on an input x and learn the message if $C(x) = y$. For security, we require that the obfuscation reveals no information on the circuit as long as the lock y has high entropy even given the circuit C.
The only known constructions of lockable obfuscation schemes require indistinguishability obfuscation ($i\mathcal{O}$) or the learning with errors (LWE) assumption. Furthermore, in terms of technique, all known constructions, excluding $i\mathcal{O}$-based, are build from provably secure variations of graph-induced multilinear maps.
We show a generic construction of a lockable obfuscation scheme built from a (leveled) fully homomorphic encryption scheme that is circularly insecure. Specifically, we need a fully homomorphic encryption scheme that is secure under chosen-plaintext attack (IND-CPA) but for which there is an efficient cycle tester that can detect encrypted key cycles. Our finding sheds new light on how to construct lockable obfuscation schemes and shows why cycle tester constructions were helpful in the design of lockable obfuscation schemes. One of the many use cases for lockable obfuscation schemes are constructions for IND-CPA secure but circularly insecure encryption schemes. Our work shows that there is a connection in both ways between circular insecure encryption and lockable obfuscation.

1 Introduction

In program obfuscation, we want to compile a circuit C to an obscure form $\widehat{C}$ while preserving the functionality of the input circuit. For security, we require that $\widehat{C}$ reveal no information on C, except what is trivially known from inspecting the input/output relations. We refer to this strong security property as virtual black-box (VBB) security. Unfortunately, Barak et al. [BGI+01, BGI+12] showed that it is impossible to achieve virtual black-box security for general functionalities. On the other hand, it turns out that it is possible to realize VBB security for some relaxed classes of functions. One such relatively expressive class consists of compute-and-compare programs, for which Goyal, Koppula, and Waters [GKW17a] and independently Wichs and Zirdelis [WZ17] construct

obfuscators under the learning with errors assumption. Additionally, Wichs and Zirdelis [WZ17] show a simple construction assuming indistinguishability obfuscation [BGI+01, BGI+12]. In short, we call obfuscation for such classes lockable obfuscation as in [GKW17a].

While the functionality of lockable obfuscation is limited to evasive functions, both works [GKW17a, WZ17] show numerous applications. For example, we can compile encryption schemes to their anonymous versions that hide the recipients public key, identity, or attributes, or construct a private sketch [DS05] from a non-private one [DRS04, DORS08]. Importantly, lockable obfuscation implies obfuscators for other important classes of functionalities like point functions [Can97, LPS04, Wee05] or conjunctions [BR13, BVWW16, BKM+18, BW19, BLMZ19].

Both works [GKW17a, WZ17], constructed lockable obfuscation from a variant of the graph induced multilinear maps of Gentry, Gorbunov, and Halevi [GGH15] also known as GGH15 directed encodings. Chen, Vaikuntanathan, and Wee [CVW18b] gave an extension of GGH15 encodings from permutation branching programs to read-once matrix branching program, and along the way, showed a lockable obfuscator for that class of functions. Recently, Goyal et al. [GKVW20] extended the construction from [GKW17a], to offer perfect correctness.

While all current constructions [GKW17a, WZ17, CVW18b, GKVW20] can be proven secure assuming the hardness of the learning with errors (LWE) problem [Reg05] with subexponential modulus-to-noise ratio, all lockable obfuscators, excluding the $i\mathcal{O}$ based, build upon on a variant of the GGH15 encodings technique [GGH15]. Despite recent advancements in constructing $i\mathcal{O}$ [LT17, AJL+19, Agr19, JLMS19, BHJ+19, JLS20, GJLS20, LPST16, BDGM20a, GP20, BDGM20b, WW20], the existing constructions are heavy and require circular or subexponential security of the underlying primitives. We note that even if $i\mathcal{O}$ is realizable from standard assumptions in the near future, lockable obfuscation may actually be the tool of choice in many applications for efficiency reasons or simplicity of the constructions. Nevertheless, to us, the current state of affairs is unsatisfactory. Notably, while the GGH15-based constructions themselves are elegant, the used techniques do not reveal any general design pattern from weaker primitives. Furthermore, current techniques are insufficient to instantiate lockable obfuscation from other assumptions. For instance, it is not clear how to realize lockable obfuscation from the ring version of LWE [LPR10], approximate greatest common divisor [HG01] or NTRU-style [HPS98] assumptions, in a way that exploits the underlying structure of the problems to get more efficient constructions.

1.1 Contribution.

In this paper, we show generic constructions for lockable obfuscation, assuming the existence of a symmetric encryption scheme and a (leveled) fully homomorphic encryption (FHE) scheme that is indistinguishable under chosen-plaintext attack (IND-CPA). Additionally, we assume that the FHE is circularly insecure, in the sense that it is feasible to detect encrypted key cycles or encryptions of

key-dependent plaintexts. We give a thorough study of our main idea and show multiple variations and extensions of our lockable obfuscation schemes.

Base Generic Constructions. Our basic construction assumes that the symmetric key encryption scheme is IND-CPA secure. We show that when we consider cryptosystems with weak keys or, in other words, leakage resilient symmetric encryption, then we can achieve lockable obfuscation where the lock value has high HILL pseudo-entropy [HILL99, HLR07] or is unpredictable given the circuit. An important observation is that when the fully homomorphic encryption scheme is itself leakage resilient for a class of leakage, then we can build the obfuscation scheme only from the FHE scheme. In particular, for the class of uniformly distributed lock values, we need to assume only the existence of the circularly insecure FHE.

Then we show a slight modification that may be of interest for concrete efficiency that assumes that the symmetric key encryption has pseudorandom ciphertexts. That is, the ciphertexts are indistinguishable from pseudorandom given an adaptive encryption oracle. We will call both schemes the base schemes.

Based on the analysis of the base schemes, we note that in the case where the FHE scheme is key-dependent message insecure, i.e., there exists a cycle tester for a key cycle of length one, then we can implement the symmetric encryption scheme as a one time pad.

Extensions. We show how to extend both schemes to lockable obfuscation with multi-bit messages. We note that there are generic methods to build such extensions. In particular, [GKW17a, WZ17] use a method that requires providing an obfuscated program for every bit of the message. Our method is conceptually different and exploits the homomorphism of the underlying FHE scheme to decode an encrypted message. Crucially, we do not need to publish an obfuscated circuit for every single bit of the message.

Finally, we observe that our technique to encode and decode a message in our lockable obfuscator can be used to launch a key recovery attack. Consequently, we show that FHE schemes that are circularly insecure and are capable of binary decomposing an encrypted message are naturally susceptible to key recovery attacks. We note, however, that the result does not influence the security of our lockable obfuscation.

Implications. As our constructions are generic and as we showcase several versions targeting different settings for lockable obfuscation, we believe that the results give us a better understanding of the primitive. Importantly we believe that the overall design paradigm is very simple and can even be used as a classroom example for lockable obfuscation. An important consequence of our work is that we showcase the usefulness of IND-CPA secure but circularly insecure encryption. Furthermore, our results, together with [GKW17a, WZ17], show a two-way connection of such encryption with lockable obfuscation. In summary, the works [GKW17a, WZ17] show that given a lockable obfuscation scheme and an IND-CPA secure encryption scheme, we can build an IND-CPA secure

encryption scheme equipped with a cycle tester. We note that the encryption scheme may be a fully homomorphic encryption scheme. For completeness we give the construction in [Klu21, Appendix A]. In this paper, we show that we can build a lockable obfuscation scheme given a (leveled) fully homomorphic encryption scheme with an efficient cycle tester. We believe that our results explain why GGH15 multilinear maps or similar cascading cancellations techniques [KW16, GKW17c] devised to build cycle testers proved to be so useful to build lockable obfuscation.

Finally, in this paper, we focus solely on the generic construction, its variations, and its extensions. While our result opens the gate to lockable obfuscation schemes, secure under assumptions other than LWE, and may perhaps even admit concretely efficient instantiations, we leave concrete constructions of such to future work.

1.2 Overview of our Techniques

In the following section, we informally discuss our results and techniques.

Our main idea. Let us remind again that in lockable obfuscation, a party can evaluate an obfuscation $\widehat{C}$ of the circuit C on an input x, and learn a message msg if $C(x) = \mathsf{lock}$, where lock is a lock value. We require that $\widehat{C}$ reveals no information on C, assuming that lock has large min-entropy even if the adversary (the evaluator) would be given C and some auxiliary information aux. Let us, for now, focus on the simplified case, where msg is always 1. In other words, if $C(x) = \mathsf{lock}$, then the lockable obfuscator returns 1, and $\perp$ otherwise. Intuitively, we can think of a lockable obfuscation $\widehat{C}$ as an encryption of C that we can evaluate and then test whether $C(x) = \mathsf{lock}$ or not. Note that the concept is very similar to zero testable homomorphic encryption and multilinear maps. However, in the case of lockable obfuscation, we allow testing an element of high min-entropy in contrast to testing zeros.

Encrypting the Circuit and Testing Ciphertexts. To encrypt a circuit, we can use a fully homomorphic encryption (FHE) scheme. That is an encryption scheme in which we can evaluate any polynomial-size circuit over encrypted data. We can also use a somewhat/leveled homomorphic encryption scheme where the circuit's depth is upper-bounded. Still, for simplicity, we refer to the scheme as fully homomorphic. As usual, we require that the fully homomorphic encryption scheme is indistinguishable under chosen plaintext attacks (IND-CPA). Hence an encryption of the circuit is indistinguishable from an encryption of zero and, in particular, reveals no information on the circuit. But to realize the testing part of the obfuscation seems to be rather difficult. This is because, at first glance, the IND-CPA property seems to stand in the way of testing anything about the plaintexts. However, we observe that actually, there already exist encryption schemes that are provably IND-CPA secure but allow to test whether a ciphertext encrypts its secret key or not. A long line of works

[Rot13, BHW15, KRW15, KW16, AP16, GKW17c, GKW17b, GKW17a, WZ17] showed separations between IND-CPA secure encryption and circular secure encryption. Roughly speaking, an encryption scheme is said to be n-circular secure if a vector of encryptions $\mathtt{Enc}(\mathsf{sk}_1, \mathsf{sk}_2), \ldots, \mathtt{Enc}(\mathsf{sk}_n, \mathsf{sk}_1)$ is indistinguishable from encryptions of zero. Previous works were primarily concerned with whether IND-CPA secure encryption is also circular secure. Fortunately, for our work, the answer is negative. That is, there are provably IND-CPA secure encryption schemes that are not circular secure, and in some drastic cases allow to recover the secret key if given a key cycle. We exploit such distinguishing or key recovery attacks to test whether the evaluated obfuscation of C equals the lock or not. In particular, we use the concept of cycle testers first formalized by Bishop, Hohenberger, and Waters [BHW15]. For instance, the folklore[3] circularly insecure encryption does satisfy our needs, as we need cycle testers that work correctly when given a FHE ciphertext that is not necessarily a fresh ciphertext. We note that previous work considered only cycle testers for fresh encryptions. For simplicity, we focus on the special case of 1-cycles in this section, and show a generalized construction in Section 3.

The Lockable Obfuscation. At first, it seems that our job is done. We set lock to the secret key sk of the FHE scheme equipped with a cycle tester, encrypt the circuit C, and we have a lockable obfuscation of C. There is but one more problem to overcome. Namely, we need to be able to choose the lock independently from the FHE parameters. Let SKE be a symmetric key encryption scheme. In the final obfuscation scheme, we give an encryption of the FHE secret key using the lock as a secret key for SKE. Concretely, we compute $\overline{\mathsf{ct}}^{(\mathsf{lock})} \leftarrow \mathsf{SKE}.\mathtt{Enc}(\mathsf{lock}, \mathsf{sk})$, $\mathsf{ct}^{(\mathsf{lock})} \leftarrow \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}, \overline{\mathsf{ct}}^{(\mathsf{lock})})$ and $\mathsf{ct} \leftarrow \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}, C)$. Then we set the obfuscated circuit as $\widehat{C} = (\mathsf{ct}^{(\mathsf{lock})}, \mathsf{ct})$. To evaluate on x, we homomorphically evaluate the universal circuit U_x that takes a circuit f and outputs $f(x)$. Specifically, we evaluate U_x on ct, obtaining as a result $\mathsf{ct}^{(C)}$ such that $\mathsf{FHE}.\mathtt{Dec}(\mathsf{sk}, \mathsf{ct}^{(C)}) = C(x)$ with high probability. Then we homomorphically evaluate the SKE decryption circuit on $\mathsf{ct}^{(\mathsf{lock})}$ using the plaintext in $\mathsf{ct}^{(C)}$ as the secret key. Precisely, we compute $\mathsf{ct}^{(\mathsf{Test})} \leftarrow \mathsf{FHE}.\mathtt{Eval}([\mathsf{ct}^{(C)}, \mathsf{ct}^{(\mathsf{lock})}], \mathsf{SKE}.\mathtt{Dec}(.,.))$. If $C(x) = \mathsf{lock}$, then $\mathsf{SKE}.\mathtt{Dec}(\mathsf{lock}, \overline{\mathsf{ct}}^{(\mathsf{lock})}) = \mathsf{sk}$ and $\mathsf{FHE}.\mathtt{Dec}(\mathsf{sk}, \mathsf{ct}^{(\mathsf{Test})}) = \mathsf{sk}$. In other words, $\mathsf{ct}^{(\mathsf{Test})}$ encrypts its secret key what we can test with a cycle tester. Otherwise, with overwhelming probability, we end up with an FHE encryption of something different from the FHE secret key.

Proving Security. To prove security, we need to construct a simulator and show that the real obfuscation is computationally indistinguishable from a simulated obfuscation. The simulator gets as input only the dimensions of the circuit and the security parameter and outputs FHE encryptions of zero of the same quantity

[3] The folklore counterexample for 1-cycles is an augmented construction of any IND-CPA secure encryption. In short, we append $y \leftarrow F(\mathsf{sk})$ to the public key, where F is a one-way function. Encryption of a message m is as in the original encryption scheme, except we return m if $F(m) = y$.

and with the same parameters as in the real obfuscation algorithm. Now we give a hybrid argument showing that a real obfuscation is indistinguishable from a simulated one.

Hybrid 0: This is the real obfuscation algorithm.

Hybrid 1: Instead of $\overline{\mathsf{ct}}^{(\mathsf{lock})} \leftarrow \mathsf{SKE.Enc}(\mathsf{lock}, \mathsf{sk})$, we compute $\mathsf{SKE.Enc}(\mathsf{lock}, 0)$. Indistinguishability of the hybrids follows from IND-CPA security of the SKE scheme.

Hybrid 2: Instead of $\mathsf{ct}^{(\mathsf{lock})} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, \overline{\mathsf{ct}}^{(\mathsf{lock})})$, we compute $\mathsf{FHE.Enc}(\mathsf{sk}, 0)$. Indistinguishability of the hybrids follows from the IND-CPA security of the FHE scheme. Note that from Hybrid 1, $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ is independent of any parameter of the FHE scheme. In particular, $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ does not depend on sk anymore. Hence we can use IND-CPA of the FHE scheme, even if the adversary would know/chose lock.

Hybrid 3: Instead of $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, C)$, we compute $\mathsf{FHE.Enc}(\mathsf{sk}, 0)$. Indistinguishability of the hybrids follows again from the IND-CPA security of the FHE scheme.

Finally, after Hybrid 3, we end up with an obfuscation that is equivalent to a simulated one.

Lock ciphertext in the Plain. Note that for the simulator to work, we need to encrypt the ciphertext $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ with the FHE key. In many concrete instantiations, this requirement may pose a significant problem for concrete efficiency, and especially the size of the obfuscated circuit. Technically, if $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ would be given in the clear, then the simulator still needs to know the lock value, and IND-CPA security is insufficient to get rid of lock. To overcome the problem, we need to assume that ciphertexts of the SKE scheme are indistinguishable from uniformly random strings. We also need to redefine the simulator, to choose $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ uniformly at random. Then, in Hybrid 1, we choose $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ uniformly at random, and we set Hybrid 3 in place of Hybrid 2. That is, after we change the obfuscation to choose $\overline{\mathsf{ct}}^{(\mathsf{lock})}$ uniformly in Hybrid 1, we compute $\mathsf{FHE.Enc}(\mathsf{sk}, 0)$ instead of $\mathsf{FHE.Enc}(\mathsf{sk}, C)$ in Hybrid 2.

Extending the Message Space. Finally, we show an extension of both the above obfuscation methods to the general case, where the obfuscation returns a message $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$, instead of just indicating whether $C(x) = \mathsf{lock}$ or not.

Previous Approaches. Similarly, as in previous work [GKW17a, WZ17], we could encode the message by building an obfuscation for each bit of the message. To encode a 1-bit, the obfuscation is as given by the specification. To encode a 0-bit, the obfuscation is created as in a simulation. There are some additional problems with the above solution that we can resolve using pseudorandom generators as

in [GKW17a, WZ17]. The obvious problem with this repetition approach is efficiency, as every single bit of the message requires publishing and evaluating an obfuscated circuit. Additionally, Goyal, Kopppula, and Waters [GKW17a] show an extension that is specific to their lockable obfuscation construction. In particular, it is not a generic construction. We show how to exploit the homomorphism of the FHE scheme in the presence of a cycle tester to encode a large message. Consequently, we obtain a generic construction that does not require publishing an obfuscated program for every bit in the circuit. Furthermore, the evaluator only need to perform a small constant number of homomorphic operation per bit of the message, in contrast to evaluating an entire obfuscated circuit.

Decoding Messages via Homomorphism and Cycle Testing. The main observation is as follows. Suppose that along with the obfuscated circuit, we publish $\mathsf{FHE.Enc}(\mathsf{msg}_i)$, where msg_i is the ith bit of the message msg. Then assuming the FHE scheme is multiplicatively homomorphic, we have with high probability

$$\mathsf{FHE.Dec}\big(\mathsf{sk}, \mathsf{FHE.Enc}(\mathsf{sk}, \mathsf{sk}) \cdot \mathsf{FHE.Enc}(\mathsf{sk}, \mathsf{msg}_i)\big) = \mathsf{msg}_i \cdot \mathsf{sk}.$$

Now, it is easy to see that if $\mathsf{msg}_i = 0$, then we have an encryption of zero, and the cycle tester will output that the ciphertext does not encode the FHE secret key. Otherwise, if $\mathsf{msg}_i = 1$, then the cycle tester will output 1 with high probability. This way, we can restore all bits of msg. Security follows immediately from our base lockable obfuscators' security and IND-CPA security of the FHE scheme.

Key Recovery Attack. The way we test the message as described above gave us a simple idea of constructing a key recovery attack against a fully homomorphic encryption scheme that has a cycle tester. The attack requires an encrypted secret key (or key cycle) and assumes the FHE scheme is capable of binary decomposing an encrypted message. Fully homomorphic encryption schemes with a binary plaintext space satisfy the later requirement immediately. Then it is easy to see that we can use the decoding technique from the previous paragraph to decode the secret keys from the key cycle.

A consequence of this observation is that a party capable of evaluating a lockable obfuscation to output a message may also be able to decrypt the obfuscated circuit. Note that this does not contradict the security notion for lockable obfuscation. However, we note that our lockable obfuscation schemes, together with the key recovery attack, tightly exemplify the security guarantees that a lockable obfuscator may offer. For instance, in constructions based on the GGH15 directed encodings technique [GGH15], it is not immediately clear whether one can easily decrypt the circuit upon successful evaluation.

1.3 Related Work and Applications

As mentioned in the introduction, all current constructions [GKW17a, WZ17, CVW18b, GKVW20] rely on the GGH15 directed encoding technique [GGH15],

or indistinguishability obfuscation[4] [BGI+01, BGI+12] to build lockable obfuscation. The original works [GKW17a, WZ17] showed the first applications of lockable obfuscation. Both works show how to use lockable obfuscation to build one-sided predicate encryption assuming, additionally, attribute-based encryption, or anonymous broadcast encryption from non-anonymous broadcast encryption. A similar technique can be used to build indistinguishability obfuscation for evasive functions assuming, additionally, witness encryption. Finally, we can also compile a public key or identity-based encryption to their anonymous counterparts where ciphertexts do not reveal the receiver's public key or its identity. An important use for lockable obfuscation is to show separations between IND-CPA security and circular security. Additionally, Goyal, Koppula, and Waters [GKW17a] show random oracle uninstantiability results. Wichs and Zirdelis [WZ17], show how to use lockable obfuscation to obfuscate affine functions and conjunctions. It is worth noting that there is only a handful of conjunction obfuscator constructions. In particular, Brakerski and Rothblum [BR13] show such obfuscators from multilinear maps, Brakerski et al. [BVWW16] assume entropic LWE. Bishop et al. [BKM+18] followed by Beullens and Wee [BW19] show conjunction obfuscators in the generic group model or from new knowledge assumptions. Recently, Bartusek et al. [BLMZ19], building upon [BKM+18], showed conjunction obfuscators for exponential alphabets in the generic group model and for binary alphabets from learning parity with noise. Notably, lockable obfuscation gives the only conjunction obfuscator for exponential alphabets from standard LWE with subexponential modulus-to-noise ratio. Our work shows the first way of building such schemes generically. Furthermore, lockable obfuscation trivially implies point function and hyperplane obfuscation [Can97, LPS04, Wee05, CD08, DKL09, GKPV10, CRV10, YZ16, BS16, KY18]. Finally, [WZ17] show how to build private secure sketches [DS05] from lockable obfuscation and non-private secure sketches [DRS04, DORS08].

As discussed, the core of our technique relies on IND-CPA secure encryption that is breakable/testable in the presence of a key cycle. We explicitly use the terminology of cycle testers introduced by Bishop, Hohenberger, and Waters [BHW15]. The first separations for IND-CPA secure and circular secure encryption are due to Haitner and Holenstein [HH09] who show that there is no black-box reduction from circular secure encryption to one-way functions, or any cryptographic assumption if the adversary can obtain encryption of an arbitrarily chosen function of the secret key. Acar et al. [ABBC10] and later Cash, Green and Hohenberger [CGH12] construct encryption schemes that are testable in the presence of a key cycle of the length of 2. Rothblum [Rot13] showed encryption schemes that allow to recover the secret key given a key cycle for bit encryption. Koppula, Ramchen, and Waters [KRW15] show a IND-CPA secure encryption scheme that allows testing n-length cycles assuming indistin-

[4] Specifically, Wichs and Zirdelis show a lockable obfuscator from null-$i\mathcal{O}$, that is, $i\mathcal{O}$ for evasive functions. However, the only known realization requires lockable obfuscation and witness encryption which we know how to build from $i\mathcal{O}$ or multilinear maps that imply $i\mathcal{O}$.

guishability obfuscation. Later Koppula and Waters [KW16], and independently Alamati and Peikert [AP16] achieve a similar result from LWE and ring-LWE. Goyal, Koppula, and Waters [GKW17b] showed 1-circular insecure bit encryption from $i\mathcal{O}$. Finally, [GKW17a, WZ17] used lockable obfuscation to construct cycle testers for bit encryption of unbounded cycle length.

We note that the idea of exploiting circular insecure encryption to build useful cryptographic algorithms is borrowed from a very recent paper by Kluczniak [Klu20], who shows a witness encryption scheme from a variant of fully homomorphic encryption with a cycle tester.

Other Applications. Chen et al. [CVW+18a] used lockable obfuscation to build traitor tracing schemes. Badrinarayanan et al. [BKSW18] showed separations for encryption secure under chosen ciphertext attack and Functional Encryption compatible encryption using lockable obfuscation. Chen et al. [CVW+18a] use lockable obfuscation to build mixed functional encryption [GKW18]. Lockable obfuscation was also used by Bitansky, Khurana, and Paneth [BKP19] to construct zero-knowledge arguments with low round complexity. Recently Ananth and La Placa [AL20], and Bitansky and Shmueli [BS20] constructed constant-round post-quantum secure zero-knowledge arguments using lockable obfuscation.

2 Preliminaries

Notation. We denote as $[i]_{i=1}^{n}$ the vector $[1, 2 \ldots, n]$. For brevity, we denote as $[n]$ the vector $[i]_{i=1}^{n}$ and as $[n, m]$ the vector $[n, n+1, \ldots, m]$. We sample a variable a from a distribution S as $a \leftarrow_{\mathsf{D}} S$. We sample a variable a from the uniform distribution over S as $a \leftarrow_{\mathsf{R}} S$. By default, we sample from the uniform distribution unless said otherwise. We denote as $x \leftarrow \mathsf{A}^{\mathcal{O}(.)}(y)$ an execution of the algorithm A on input y that gets access to an oracle $\mathcal{O}$ and treats it as its subroutine. In general, we mark unassigned variables when calling an algorithm with a ".".

We denote any positive polynomial as $\mathsf{poly}(.)$. Finally, we denote as $\mathsf{negl}(.)$ any negligible function. That is, for any positive polynomial $\mathsf{poly}(.)$ there exists $c \in \mathbb{N}$ such that for all $\lambda \geq c$ we have $|\mathsf{negl}(\lambda)| \leq \frac{1}{\mathsf{poly}(\lambda)}$.

Entropy. The min-entropy of a random variable A is defined as $\mathbf{H}_{\infty}(A) = -\log(\max_a \Pr[A = a])$. Let $\mathbf{E}$ denote the expectation of a random variable. The average conditional min-entropy of a random variable X conditioned on a possibly correlated variable Y is defined as

$$\widetilde{\mathbf{H}}_{\infty}(X|Y) = -\log\left(\mathbf{E}_{y \leftarrow Y}\left[2^{-\mathbf{H}_{\infty}(X|Y=y)}\right]\right).$$

Definition 1 (Conditional (HILL) Pseudo-Entropy [HILL99, HLR07]). *Let λ be a security parameter. Let $X = \{X_\lambda\}$, $Y = \{Y_\lambda\}$ be ensembles of jointly*

distributed random variables. We define the conditional pseudo-entropy of X conditioned on Y to be at least $\alpha(\lambda)$, denoted $\mathbf{H}_{\mathsf{HILL}}(X|Y) \geq \alpha(\lambda)$ if there exist some $X' = \{X'_\lambda\}$ possibly jointly distributed with Y such that $\widetilde{\mathbf{H}}_\infty(X'_\lambda|Y_\lambda) \geq \alpha(\lambda)$, and for all **PPT** *adversaries we have*

$$\big|\Pr[\mathsf{A}(X,Y)=1] - \Pr[\mathsf{A}(X',Y)=0]\big| = \mathsf{negl}(\lambda).$$

Symmetric Encryption. Below we give a generalized definition of symmetric key encryption. Our correctness definition states explicitly that decryption with a wrong key should result in an incorrect message with high probability. We define indistinguishability under chosen-plaintext attack and pseudorandom ciphertexts of symmetric-key ciphers. We define the security properties for secret keys sampled from a given class of distributions. Later we recall popular classes of distributions from the literature, but we stress that our results are shown generically, without relying on any particular class.

Definition 2 (Symmetric Key Encryption). *An encryption scheme* $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *consists of an encryption algorithm* $\mathtt{Enc}$ *and decryption algorithm* $\mathtt{Dec}$ *with the following syntax.*

$\mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg})$**:** *Takes as input a security parameter λ, a secret key $\overline{\mathsf{sk}} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ and a message $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$ where $\ell_{\mathsf{sk}}, \ell_{\mathsf{msg}} = \mathsf{poly}(\lambda)$, and outputs a ciphertext $\overline{\mathsf{ct}} \in \{0,1\}^{\ell_{\mathsf{ct}}}$ where $\ell_{\mathsf{ct}} = \mathsf{poly}(\lambda)$.*

$\mathtt{Dec}(\overline{\mathsf{sk}}, \overline{\mathsf{ct}})$**:** *This deterministic algorithm takes as input a secret key $\overline{\mathsf{sk}} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ and a ciphertext $\overline{\mathsf{ct}} \in \{0,1\}^{\ell_{\mathsf{ct}}}$, and outputs $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$.*

Correctness: *We say that $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ is correct, if for all security parameters $\lambda \in \mathbb{N}$, $\overline{\mathsf{sk}} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ and $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$, where $\ell_{\mathsf{sk}}, \ell_{\mathsf{msg}} = \mathsf{poly}(\lambda)$ we have*

$$\mathtt{Dec}(\overline{\mathsf{sk}}, \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg})) = \mathsf{msg},$$

and for all $\overline{\mathsf{sk}}' \in \{0,1\}^{\ell_{\mathsf{sk}}}$ such that $\overline{\mathsf{sk}}' \neq \overline{\mathsf{sk}}$ we have

$$\Pr\left[\mathtt{Dec}\big(\overline{\mathsf{sk}}', \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg})\big) = \mathsf{msg}\right] = \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda),$$

where $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda) = \mathsf{negl}(\lambda)$.

$\mathcal{D}$-Indistinguishability Under Chosen Plaintext Attack: *Let $\lambda \in \mathbb{N}$ be a security parameter and $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ be a* **PPT** *adversary. Let $\mathcal{D}$ be a class of distribution ensembles $\{D_k\}_{k\in\mathbb{N}}$ that sample $(\overline{\mathsf{sk}}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_k$ with $\overline{\mathsf{sk}} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ where $\ell_{\mathsf{sk}} = \mathsf{poly}(k)$. We define the advantage of the adversary A against a $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ encryption scheme in the $\mathcal{D}$-IND-CPA game as*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{A},\mathsf{SKE}}(\lambda) = \Pr\left[\mathsf{A}_1(\overline{\mathsf{ct}}_b, \mathsf{st}) = b\text{: } \begin{array}{c} (\overline{\mathsf{sk}}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_\lambda, \\ (\mathsf{st}, \mathsf{msg}_0, \mathsf{msg}_1) \leftarrow \mathsf{A}_0^{\mathcal{O}(\overline{\mathsf{sk}},\cdot)}(\lambda, \mathsf{aux}), \\ b \leftarrow_{\mathsf{R}} \{0,1\}, \\ \overline{\mathsf{ct}}_b \leftarrow \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg}_b) \end{array}\right],$$

where the oracle $\mathcal{O}$ on input a message msg *outputs* $\overline{\mathsf{ct}} \leftarrow \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg})$.
We say that $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *is* $\mathcal{D}$-IND-CPA*-secure if for all* **PPT** *adversaries* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ *we have* $\mathsf{Adv}^{\mathcal{D}\text{-IND-CPA}}_{\mathsf{A},\mathsf{SKE}}(\lambda) = \mathsf{negl}(\lambda)$.
We say that a cryptosystem $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *is* $\mathcal{D}$*-semantically secure if the above holds but* A *has no access to the oracle* $\mathcal{O}$.

$\mathcal{D}$-Pseudorandom Ciphertexts: *Let* $\lambda \in \mathbb{N}$ *be a security parameter and* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ *be a* **PPT** *adversary. Let* $\mathcal{D}$ *be a class of distribution ensembles* $\{D_k\}_{k\in\mathbb{N}}$ *that sample* $(\overline{\mathsf{sk}}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_k$ *with* $\overline{\mathsf{sk}} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ *where* $\ell_{\mathsf{sk}} = \mathsf{poly}(k)$.
We define the advantage of A *against a* $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *encryption scheme in the pseudorandom ciphertexts game as*

$$\mathsf{Adv}^{\mathsf{RandCt}}_{\mathsf{A},\mathsf{SKE}}(\lambda) = \Pr\left[\mathsf{A}_1(\overline{\mathsf{ct}}_b, \mathsf{st}) = b: \begin{array}{c} (\overline{\mathsf{sk}}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_\lambda, \\ (\mathsf{st}, \mathsf{msg}) \leftarrow \mathsf{A}_0^{\mathcal{O}(\overline{\mathsf{sk}},.)}(\lambda, \mathsf{aux}), \\ b \leftarrow_{\mathsf{R}} \{0,1\}, \\ \overline{\mathsf{ct}}_0 \leftarrow \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg}), \overline{\mathsf{ct}}_1 \leftarrow_{\mathsf{R}} \{0,1\}^{\ell_{\mathsf{ct}}} \end{array}\right],$$

where the oracle $\mathcal{O}$ on input a message msg *outputs* $\overline{\mathsf{ct}} \leftarrow \mathtt{Enc}(\lambda, \overline{\mathsf{sk}}, \mathsf{msg})$.
We say that $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *has* $\mathcal{D}$*-pseudorandom ciphertexts if for all* **PPT** *adversaries* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ *we have* $\mathsf{Adv}^{\mathcal{D}\text{-RandCt}}_{\mathsf{A},\mathsf{SKE}}(\lambda) = \mathsf{negl}(\lambda)$.
Analogously to semantic security, we say that a cryptosystem $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *has weakly* $\mathcal{D}$*-pseudorandom ciphertexts if the above holds but* A *has no access to the oracle* $\mathcal{O}$.

Classes of Distributions. Let us recall popular classes of distributions. The following classes were also considered by Wichs, and Zirdelis [WZ17] for their lockable obfuscation scheme.

Uniform: The variable x is chosen uniformly at random. This is the standard definition of IND-CPA.

Unpredictable: Informally, it is hard to predict x given aux. Formally, a class $\mathcal{D}$ is unpredictable if for all **PPT** adversaries A, security parameters $\lambda \in \mathbb{N}$, and distribution ensembles $\{D_k\}_{k\in\mathbb{N}} \in \mathcal{D}$ we have

$$\Pr[x \leftarrow \mathsf{A}(\mathsf{aux}) : (x, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_\lambda] = \mathsf{negl}(\lambda).$$

Pseudo-Entropy: For a function $\alpha(\lambda)$ in the security parameter λ the class of α-pseudo-entropy distributions consists of ensembles $\{D_k\}_{k\in\mathbb{N}}$ such that $(x, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_\lambda$ satisfies $\mathbf{H}_{\mathsf{HILL}}(x|\mathsf{aux}) \geq \alpha(\lambda)$.

Symmetric or public-key encryption schemes secure for the class of unpredictable distribution can be constructed from learning parity with noise [DKL09], decisional Diffie-Hellman and learning with errors [DGK+10] assumptions and from point function obfuscators satisfying some special properties [CKVW10]. For the class of pseudo-entropy distributions we know constructions from learning with errors [AGV09, GKPV10] hash proof systems [NS09, ADN+10], assumptions in bilinear groups [DHLW10], computational Diffie-Hellman and subgroup indistinguishability assumptions [BG10, BLSV18]. It is worth mentioning

that we might realize leakage resilient encryption from pseudorandom functions with weak seeds [Pie09, AKPW13] and encryption schemes with semantic security and weakly pseudorandom ciphertexts from leakage resilient pseudorandom generators [DP08, Zha16].

Fully Homomorphic Encryption. We recall the definition of fully homomorphic encryption [RAD+78, Gen09]. In the definition, the $\texttt{Setup}$ algorithm takes as input a depth of the circuit reflecting leveled/somewhat homomorphic schemes capable of evaluating the circuit of the given depth. We note, however, that our results apply to unbounded fully homomorphic encryption schemes as well. For brevity, we will omit "leveled/somewhat" and refer to the schemes as fully homomorphic. Additionally, we note that usually, we define a public key or an evaluation key in fully homomorphic encryption schemes. In this paper, we do not use such keys explicitly. Therefore, we assume that such a public/evaluation key is part of the secret key or ciphertext.

Definition 3 (Fully Homomorphic Encryption). *A fully homomorphic encryption* FHE *consists of algorithms* $(\texttt{Setup}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ *with the following syntax.*

$\texttt{Setup}(\lambda, \delta)$**:** *This* **PPT** *algorithm takes as input a security parameter* λ *and bound on the circuit depth* δ*. The algorithm outputs a secret key* sk*. Sometimes we omit the circuit depth in the input when it is not needed in the given context.*

$\texttt{Enc}(\mathsf{sk}, \mathsf{msg})$**:** *This* **PPT** *algorithm takes as input a secret key* sk*, and a message* msg*, and returns a ciphertext* ct*.*

$\texttt{Eval}([\mathsf{ct}_i]_{i=1}^{\kappa}, C)$**:** *Given as input a set of ciphertexts* $[\mathsf{ct}_i]_{i=1}^{\kappa}$*, and a circuit* C*, the algorithm outputs a ciphertext* ct*.*

$\texttt{Dec}(\mathsf{sk}, \mathsf{ct})$**:** *This deterministic algorithm given a secret key* sk *and a ciphertext* ct*, outputs a message* msg*.*

Correctness: *We say that* $\mathsf{FHE} = (\texttt{Setup}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ *is correct, if for all security parameters* $\lambda \in \mathbb{N}$*, circuits* $C : \mathcal{M}^{\kappa} \mapsto \mathcal{M}$ *over the message space* $\mathcal{M}$ *of depth* $\delta = \mathsf{poly}(\lambda)$*, and messages* $[\mathsf{msg}_i \in \mathcal{M}]_{i=1}^{\kappa}$ *we have*

$$\Pr\left[\texttt{Dec}(\mathsf{sk}, \mathsf{ct}_{\mathsf{out}}) = C([\mathsf{msg}_i]_{i=1}^{\kappa}) : \begin{array}{r} \mathsf{sk} \leftarrow \texttt{Setup}(\lambda, \delta), \\ [\mathsf{ct}_i \leftarrow \texttt{Enc}(\mathsf{sk}, \mathsf{msg}_i)]_{i=1}^{\kappa} \\ \mathsf{ct}_{\mathsf{out}} \leftarrow \texttt{Eval}([\mathsf{ct}_i]_{i=1}^{\kappa}, C) \end{array}\right] = 1 - \mathsf{Err}^{\mathsf{corr}}_{\texttt{Eval}}(\lambda),$$

where $\mathsf{Err}^{\mathsf{corr}}_{\texttt{Eval}}(\lambda) = \mathsf{negl}(\lambda)$*. We call* $\mathsf{Err}^{\mathsf{corr}}_{\texttt{Eval}}(\lambda)$ *the correctness error.*

The distribution of evaluated ciphertexts and fresh ciphertexts may differ. In our correctness analysis, we need conveniently denote to what message a given ciphertext decrypts. Therefore, we denote as $\mathsf{ct}_{\mathsf{out}} \approx \texttt{Enc}(\mathsf{sk}, \mathsf{msg})$ the fact that $\texttt{Dec}(\mathsf{sk}, \mathsf{ct}_{\mathsf{out}}) = \mathsf{msg}$ with some correctness error $\mathsf{Err}^{\mathsf{corr}}_{\texttt{Eval}}(\lambda)$.

Efficiency: We require that $\texttt{Setup}$, $\texttt{Enc}$ and $\texttt{Dec}$ run in $\mathsf{poly}(\lambda, \delta)$ time, and $\texttt{Eval}$ runs in $\mathsf{poly}(\lambda, |C|)$ time.

Indistinguishability Under Chosen Plaintext Attack: We define indistinguishability under chosen plaintext attack as in Definition 2, with the exception that the $\mathtt{Setup}$ algorithm generates the secret key. Furthermore, we note that, while it is possible to define fully homomorphic encryption with weak keys, it does not play a special role in our paper. Therefore, we consider the secret keys' distribution to be uniform (the $\mathtt{Setup}$ algorithm works on a uniformly random seed), and we use $\mathsf{IND\text{-}CPA}$ as the acronym instead of $\mathcal{D}\text{-}\mathsf{IND\text{-}CPA}$.

We use the concept of cycle testers introduced by Bishop, Hohenberger, and Waters [BHW15]. However, we use the definition by Kluczniak [Klu20], as it is easier to use for our purposes[5]. We give the construction from [GKW17a] of a fully homomorphic encryption with a cycle tester in [Klu21, Appendix A]. Furthermore, we note that the the scheme can be instantiated from LWE with subexponential modulus-to-noise ratio.

Definition 4 (Cycle Testing). *We define an additional algorithm* $\mathtt{Test}$ *with the following syntax.*

$\mathtt{Test}([\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m})$**:** *The algorithm on input a vector of ciphertexts* $[\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m}$ *outputs a bit* $b \in \{0,1\}$.

Efficiency: *We require that* $\mathtt{Test}$ *runs in time* $\mathsf{poly}(\lambda)$.

Correctness: *Let* $\mathsf{FHE} = (\mathtt{Setup}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{Dec}, \mathtt{Test})$ *be a fully homomorphic encryption scheme with an* n*-cycle tester* $\mathtt{Test}$ *for functions* $\mathsf{F}_j : \mathcal{S} \mapsto \mathcal{M}$*, where* $\mathcal{M}$ *is the message space and* $\mathcal{S}$ *is the secret key space and* $j \in [m]$*. We say that the cycle tester is correct if for all security parameters* $\lambda \in \mathbb{N}$*, and all executions* $\left[\mathsf{sk}_i \leftarrow \mathtt{Setup}(\lambda)\right]_{i=1}^{n}$*, we have* $\mathsf{Err}^{\mathsf{corr}}_{\mathtt{Test}}(\lambda) = \mathsf{negl}(\lambda)$*, where*

$$\Pr\left[\mathtt{Test}([\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m}) \neq 1\right] \leq \mathsf{Err}^{\mathsf{corr}}_{\mathtt{Test}}(\lambda)$$

given that $[\mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})]_{i=1,j=1}^{n,m} = [\mathtt{Dec}(\mathsf{sk}_i, \mathsf{ct}_{i,j})]_{i=1,j=1}^{n,m}$*, and*

$$\Pr\left[\mathtt{Test}([\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m}) \neq 0\right] \leq \mathsf{Err}^{\mathsf{corr}}_{\mathtt{Test}}(\lambda)$$

given that $[\mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})]_{i=1,j=1}^{n,m} \neq [\mathtt{Dec}(\mathsf{sk}_i, \mathsf{ct}_{i,j})]_{i=1,j=1}^{n,m}$.

Lockable Obfuscation. Now we recall lockable obfuscation introduced by Goyal, Koppula, and Waters [GKW17a], and independently by Wichs and Zirdelis [WZ17].

Definition 5 (Lockable Obfuscation). *A lockable obfuscation scheme* $\mathsf{LObf} = (\mathtt{Obf}, \mathtt{Eval})$ *consists of an obfuscation algorithm* $\mathtt{Obf}$ *and an evaluation algorithm* $\mathtt{Eval}$ *with the following syntax.*

[5] As pointed by Kluczniak [Klu20], the definition by Bishop, Hohenberger, and Waters [BHW15] does not make a distinction between a cycle tester and an encryption scheme with an efficient zero tester.

$\mathtt{Obf}(\lambda, C, \mathsf{lock}, \mathsf{msg})$**:** *This algorithm takes as input a security parameter* $\lambda \in \mathbb{N}$*, a circuit* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*, a lock string* $\mathsf{lock} \in \{0,1\}^\eta$*, and a message* $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$*. The algorithm outputs an obfuscated circuit* $\widehat{C}$*.*

$\mathtt{Eval}(\widehat{C}, x)$: *This deterministic algorithm takes as input an obfuscated circuit* $\widehat{C}$ *and input* $x \in \{0,1\}^\kappa$*, and outputs* msg *or* $\perp$*.*

Efficiency: *We say that the lockable obfuscation scheme is polynomially efficient, if* Obf *and* Eval *run in time* $\mathsf{poly}(\lambda, |C|)$*.*

Correctness: *We say that a lockable obfuscator* $\mathsf{LObf} = (\mathtt{Obf}, \mathtt{Eval})$ *is correct if for all* $\lambda \in \mathbb{N}$*,* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*,* $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$*,* $\mathsf{lock} \in \{0,1\}^\eta$*, and* $x \in \{0,1\}^\kappa$*, given that* $\widehat{C} \leftarrow \mathtt{Obf}(\lambda, C, \mathsf{lock}, \mathsf{msg})$ *and* $C(x) = \mathsf{lock}$*, we have*

$$\Pr[\mathtt{Eval}(\widehat{C}, x) \neq \mathsf{msg}] \leq \mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}.\mathtt{Eval}}(\lambda),$$

and given that $C(x) \neq \mathsf{lock}$ *we have that*

$$\Pr[\mathtt{Eval}(\widehat{C}, x) \neq \perp] \leq \mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}.\mathtt{Eval}}(\lambda),$$

where $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}.\mathtt{Eval}}(\lambda) = \mathsf{negl}(\lambda)$ *and the probability is over random coins of the obfuscation algorithm* Obf*.*

We consider also a limited version, where the lockable obfuscation has the message set to $\mathsf{msg} = 1$ for $C(x) = \mathsf{lock}$ and outputs 0 instead of $\perp$. In particular, our first construction given in Section 3 follows the limited functionality. Later in Section 4.2, we show how to extend the scheme to handle any polynomial-size messages.

Distributional Virtual Black-Box ($\mathcal{D}$-DVBB) Security: *Let* $\mathcal{C}_k = \{C_{\kappa,\eta,\upsilon}\}$ *be the set of all circuits with* κ *input variables,* η *output variables and size* υ*, where* $\kappa, \eta, \upsilon = \mathsf{poly}(k)$*. Let* $\mathcal{D}$ *be a class of distribution ensembles* $\{D_k\}_{k\in\mathbb{N}}$ *that sample* $(\mathsf{lock}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_k$ *with* $\mathsf{lock} \in \{0,1\}^\eta$*.*
We say that the lockable obfuscation is distributional virtual black-box secure for the distribution class $\mathcal{D}$ *if for all* **PPT** *adversaries* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$*, there exists a* **PPT** *simulator* Sim*, such that* $\mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}}(\lambda) = \mathsf{negl}(\lambda)$*, where*

$$\mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}}(\lambda) = \left| \Pr\left[\mathsf{A}_2(\widehat{C}_b, \mathsf{st}) = b : \begin{array}{c} (\mathsf{lock}, \mathsf{aux}) \leftarrow_{\mathsf{D}} D_\lambda, \\ b \leftarrow_{\mathsf{R}} \{0,1\}, \\ (C, \mathsf{msg}, \mathsf{st}) \leftarrow \mathsf{A}_1(\lambda, \mathsf{aux}), \\ \mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}, C \in \mathcal{C}_\lambda \\ \widehat{C}_0 \leftarrow \mathtt{Obf}(\lambda, C, \mathsf{lock}, \mathsf{msg}), \\ \widehat{C}_1 \leftarrow \mathtt{Sim}(\lambda, \kappa, \eta, \upsilon, \ell_{\mathsf{msg}}) \end{array} \right] - \frac{1}{2} \right|,$$

We call $\mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}}(\lambda)$ *the advantage of the adversary* A *against* DVBB *security.*

3 Lockable Obfuscation from Circular Insecure FHE

In this section, we show the basic construction of lockable obfuscation from fully homomorphic encryption with an efficient cycle tester. The lockable obfuscation returns a single bit that is set to 1 when the outcome of the obfuscated function is equal to the lock, and $\perp$ otherwise.

Construction 1 (Our Lockable Obfuscation Construction) *Let* $\mathsf{FHE} = (\mathtt{Setup}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Test})$ *be a fully homomorphic encryption scheme with a cycle tester detecting* n*-length key cycles for* F_j *where* $j \in [m]$*. Let* $\mathsf{SKE} = (\mathtt{Enc}, \mathtt{Dec})$ *be a symmetric encryption scheme with secret key space* $\{0,1\}^\eta$ *and message space* $\{0,1\}^{\ell_{\mathsf{sk}}}$*. Denote as* $U_x(.)$ *the universal circuit that on input a circuit* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*, outputs* $C(x)$*, where* $x \in \{0,1\}^\kappa$*. Let* $\delta \in \mathbb{N}$ *be the depth of the circuit* $\mathsf{SKE}.\mathtt{Dec}(U_x(.),.)$*. We define the lockable obfuscation* $\mathsf{LObf} = (\mathtt{Obf}, \mathtt{Eval})$ *as follows.*

$\mathtt{Obf}(\lambda, C, \mathsf{lock})$**:** *Takes as input a security parameter* λ*, a circuit* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*, and a lock string* $\mathsf{lock} \in \{0,1\}^\eta$*.*

1. *For* $i \in [n]$ *do*
   - *Run* $\mathsf{sk}_i \leftarrow \mathsf{FHE}.\mathtt{Setup}(\lambda, \delta)$*.*
   - *Run* $\mathsf{ct}_i \leftarrow \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, C)$*.*
   - *For* $j \in [m]$ *do*
     - *Run* $\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})} \leftarrow \mathsf{SKE}.\mathtt{Enc}\big(\lambda, \mathsf{lock}, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})\big)$*.*
     - *Run* $\mathsf{ct}_{i,j}^{(\mathsf{lock})} \leftarrow \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})})$*.*
2. *Return* $\widehat{C} \leftarrow \big([\mathsf{ct}_i]_{i=1}^n, [\mathsf{ct}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}\big)$*.*

$\mathtt{Eval}(\widehat{C}, x)$**:** *Takes as input an obfuscated circuit* $\widehat{C} = \big([\mathsf{ct}_i]_{i=1}^n, [\mathsf{ct}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}\big)$*, and an input* $x \in \{0,1\}^\kappa$*.*

1. *For* $i \in [n]$ *do*
   - *Compute* $\mathsf{ct}_i^{(C)} \leftarrow \mathsf{FHE}.\mathtt{Eval}(\mathsf{ct}_i, U_x)$*.*
   - *For* $j \in [m]$ *compute*

$$\mathsf{ct}_{i,j}^{(\mathsf{Test})} \leftarrow \mathsf{FHE}.\mathtt{Eval}\big([\mathsf{ct}_i^{(C)}, \mathsf{ct}_{i,j}^{(\mathsf{lock})}], \mathsf{SKE}.\mathtt{Dec}(.,.)\big).$$

2. *If* $\mathsf{FHE}.\mathtt{Test}([\mathsf{ct}_{i,j}^{(\mathsf{Test})}]_{i=1,j=1}^{n,m}) = 1$*, then output* 1*, and output* $\perp$ *otherwise.*

Theorem 1 (Correctness). *For all* λ*,* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*, all* $\mathsf{lock} \in \{0,1\}^\eta$*,* LObf *as given by Construction 1 is a polynomially efficient and correct lockable obfuscation with correctness error*

$$\mathsf{Err}_{\mathsf{LObf}}^{\mathsf{corr}}(\lambda) \leq n \cdot m \cdot \mathsf{Err}_{\mathsf{FHE}.\mathtt{Eval}}^{\mathsf{corr}}(\lambda, \delta) + \mathsf{Err}_{\mathsf{SKE}}^{\mathsf{corr}}(\lambda) + \mathsf{Err}_{\mathsf{FHE}.\mathtt{Test}}^{\mathsf{corr}}(\lambda).$$

Proof. Polynomial efficiency follows directly from the efficiency of the underlying encryption schemes. Thus we focus on analyzing the correctness. From correctness of the FHE scheme we have $\mathsf{ct}_i^{(C)} = \mathsf{FHE}.\mathtt{Eval}(\mathsf{ct}_i, U_x) \approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, C(x))$ and

$$\begin{aligned}\mathsf{ct}_{i,j}^{(\mathsf{Test})} &= \mathsf{FHE}.\mathtt{Eval}\big([\mathsf{ct}_i^{(C)}, \mathsf{ct}_{i,j}^{(\mathsf{lock})}], \mathsf{SKE}.\mathtt{Dec}(.,.)\big) \\ &\approx \mathsf{FHE}.\mathtt{Enc}\big(\mathsf{sk}_i, \mathsf{SKE}.\mathtt{Dec}(C(x), \overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})})\big)\end{aligned}$$

with probability of failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Eval}}(\lambda,\delta)$ for each $i \in [n]$ and $j \in [m]$.

If $C(x) = \mathsf{lock}$ we have $\mathsf{ct}^{(\mathsf{Test})}_{i,j} \approx \mathsf{FHE.Enc}\big(\mathsf{sk}_i, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})\big)$. Then we have $\mathtt{Test}([\mathsf{ct}^{(\mathsf{Test})}_{i,j}]^{n,m}_{i=1,j=1}) = 1$ with probability failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda)$.

If $C(x) \neq \mathsf{lock}$, then we have $\mathsf{ct}^{(\mathsf{Test})}_{i,j} \approx \mathsf{FHE.Enc}(\mathsf{sk}_i, \widetilde{\mathsf{msg}}_{i,j})$, where the plaintext is $\widetilde{\mathsf{msg}}_{i,j} = \mathsf{SKE.Dec}(C(x), \overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j})$. From correctness of the SKE scheme we have that there exists $i \in [n]$ and all $j \in [m]$ such that $\widetilde{\mathsf{msg}}_{i,j} \neq \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ with probability at least $1 - \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda)$. Therefore, the ciphertexts $\mathsf{ct}^{(\mathsf{Test})}_{i,j}$ does not encode a proper cycle. Consequently, we have $\mathtt{Test}([\mathsf{ct}^{(\mathsf{Test})}_{i,j}]^{n,m}_{i=1,j=1}) = 0$ with probability of failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda)$.

To summarize we have the probability of failure of the lockable obfuscation $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}}(\lambda) \leq n \cdot m \cdot \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Eval}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda)$.

Theorem 2 (Security). *Let $\mathcal{D}$ be a class of distribution ensembles $\{D_\lambda\}_{\lambda\in\mathbb{N}}$ that sample $(\mathsf{lock}, C) \leftarrow_{\mathsf{D}} D_\lambda$, with $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$, $\mathsf{lock} \in \{0,1\}^\eta$ and $\kappa, \eta = \mathsf{poly}(\lambda)$. Let SKE be a $\mathcal{D}$-IND-CPA secure symmetric key encryption scheme, and FHE be a IND-CPA secure fully homomorphic encryption scheme. Then, LObf given by Construction 1, is $\mathcal{D}$-DVBB secure.*

Proof. Let us first describe the simulator. The simulator $\mathtt{Sim}$ takes as input λ, κ, η, υ and ℓ_{msg}. Then $\mathtt{Sim}$ runs $\mathsf{sk}_i \leftarrow \mathsf{FHE.Setup}(\lambda,\delta)$ as in the real scheme, and computes $\widehat{C} \leftarrow \big([\mathsf{ct}_i]^n_{i=1}, [\mathsf{ct}^{(\mathsf{lock})}_{i,j}]^{n,m}_{i=1,j=1}\big)$, where $\mathsf{ct}_i \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, 0)$ and $\mathsf{ct}^{(\mathsf{lock})}_{i,j} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, 0)$, for all $i \in [n]$ and $j \in [m]$.

Via the following hybrid argument, we show that a simulated program is computationally indistinguishable from an obfuscated program. We denote as $\mathcal{H}_j$ the event that an adversary guesses the bit b in Hybrid j.

Hybrid 0**:** This is the DVBB game with the bit $b = 0$. That is we compute $\widehat{C}_b \leftarrow \mathtt{Obf}(\lambda, C, \mathsf{lock}, \mathsf{msg})$. We have $\mathsf{Adv}^{\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}} = |\Pr[\mathcal{H}_0] - \frac{1}{2}|$.

Hybrid $(i-1)\cdot m + j$**:** For $i \in [n]$ and $j \in [m]$ we compute the ciphertext $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j} \leftarrow \mathsf{SKE.Enc}(\lambda, \mathsf{lock}, 0)$ instead of the ciphertext $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j} \leftarrow \mathsf{SKE.Enc}\big(\lambda, \mathsf{lock}, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})\big)$.

Claim. If an adversary A distinguishes between Hybrid $(i-1)\cdot m + j$ and Hybrid $(i-1)\cdot m + j - 1$, then there exists a distinguisher D, that uses A to break $\mathcal{D}$-IND-CPA security of SKE. We have

$$\big|\Pr[\mathcal{H}_{(i-1)\cdot m+j}] - \Pr[\mathcal{H}_{(i-1)\cdot m+j-1}]\big| = \mathsf{Adv}^{\mathcal{D}\text{-IND-CPA}}_{\mathsf{D},\mathsf{SKE}}(\lambda).$$

Proof. First, the solver generates all secret keys of the FHE scheme. For $i' \in [n]$ and $j' \in [m]$ such that $(i'-1)\cdot m + j' < (i-1)\cdot m + j$ the solver queries the $\mathcal{O}$ for $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i',j'} \leftarrow \mathcal{O}(\mathsf{lock}, 0)$. For $(i'-1)\cdot m + j' > (i-1)\cdot m + j$, the solver queries $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i',j'} \leftarrow \mathcal{O}(\mathsf{lock}, \mathsf{F}_{j'}(\mathsf{sk}_{(i' \mod n)+1})$. Finally, the solver submits $\mathsf{msg}_0 = \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ and $\mathsf{msg}_1 = 0$ as the challenge, and obtains

$\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})}$. The rest of the obfuscated program is computed as given by the specification. Then if the adversary outputs that it is Hybrid $(i-1)\cdot m+j-1$, then the solver answers that the encrypted message is msg_0. Otherwise, the solver answers that the message is msg_1.

Hybrid $n \cdot m + (i-1) \cdot m + j$**:** For $i \in [n]$ and $j \in [m]$ we compute $\mathsf{ct}_{i,j}^{(\mathsf{lock})} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, 0)$ instead of $\mathsf{ct}_{i,j}^{(\mathsf{lock})} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, \overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})})$.

Claim. If an adversary A distinguishes between Hybrid $n \cdot m + (i-1) \cdot m + j$ and Hybrid $n \cdot m + (i-1) \cdot m + j - 1$, then there exists a distinguisher D, that uses A to break IND-CPA security of FHE.
We have

$$\left| \Pr[\mathcal{H}_{n \cdot m + (i-1) \cdot m + j}] - \Pr[\mathcal{H}_{n \cdot m + (i-1) \cdot m + j - 1}] \right| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda).$$

Proof. First, the solver generates all secret keys of the FHE scheme except sk_i. For all $i' \in [n]$ such that $i' \neq i$ and all $j' \in [m]$, the solver generates $\mathsf{ct}_{i',j}^{(\mathsf{lock})}$ and $\mathsf{ct}_{i'}$ as in the previous hybrid. To obtain ct_i the solver queries $\mathcal{O}(sk_i, .)$ on input C. To obtain $\mathsf{ct}_{i,j'}^{(\mathsf{lock})}$ the solver submits 0 for all $j' < j$, and $\overline{\mathsf{ct}}_{i,j'}^{(\mathsf{lock})}$ for $j' > j$. Finally, to obtain $\mathsf{ct}_{i,j}^{(\mathsf{lock})}$ the solver sets the challenge query as $\mathsf{msg}_0 = \overline{\mathsf{ct}}_{i,j'}^{(\mathsf{lock})}$ and $\mathsf{msg}_1 = 0$.
If the adversary outputs that it is Hybrid $n \cdot m + (i-1) \cdot m + j - 1$, then the solver answers that the encrypted message is msg_0. Otherwise, the solver answers that the message is msg_1.

Hybrid $2 \cdot n \cdot m + i$**:** For $i \in [n]$ we compute $\mathsf{ct}_i \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, 0)$ instead of $\mathsf{ct}_i \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, C)$.

Claim. If an adversary A distinguishes between Hybrid $2 \cdot n \cdot m + i$ and Hybrid $2 \cdot n \cdot m + i - 1$, then there exists a distinguisher D, that uses A to break IND-CPA security of FHE.
We have

$$\left| \Pr[\mathcal{H}_{2 \cdot n \cdot m + i}] - \Pr[\mathcal{H}_{2 \cdot n \cdot m + i - 1}] \right| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda).$$

Proof. The proof is a standard reduction to IND-CPA of the FHE scheme analogous to the proof of Hybrids $n \cdot m + (i-1) \cdot m + j$ for $i \in [n]$ and $j \in [m]$.

In Hybrid $2 \cdot n \cdot m + n$ the obfuscated program is equivalent to a simulated program. In particular all encryptions that constitute the obfuscated program are encryptions of 0. To summarize, we have that the advantage to distinguish between Hybrid 0 and Hybrid $2 \cdot n \cdot m + n$ is $\mathsf{Adv}^{\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}} \leq n \cdot m \cdot \mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{SKE}}(\lambda) + (n \cdot m + n) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda)$.

4 Extensions and Variants of the Lockable Obfuscation Scheme

In this section, we show a variant of the lockable obfuscation scheme and an extension that allows the obfuscator to output polynomial-size messages.

4.1 Lock Ciphertext in the Plain

We show a slight modification of Construction 1, where instead of encrypting the lock ciphertexts $[\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}$, with the FHE encyption algorithm, we include these ciphertexts into the obfuscated program. However, for the security proof to work, we need to assume that SKE has pseudorandom ciphertexts. To not restate the construction from Section 3, we only give the changes.

Construction 2 (Lock Ciphertexts in the Plain) *Let* LObf *be as in Construction 1, except we do not compute* $\mathsf{ct}_{i,j}^{(\mathsf{lock})}$*, and* $\mathtt{Obf}$ *returns the obfuscated circuit* $\widehat{C} = \big([(\mathsf{ct}_i]_{i=1}^{n}, [\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}\big)$*. Furthermore, in the* $\mathtt{Eval}$ *algorithm we compute* $\mathsf{ct}_{i,j}^{(\mathsf{Test})} \leftarrow \mathsf{FHE}.\mathtt{Eval}\big(\mathsf{ct}_i^{(C)}, \mathsf{SKE}.\mathtt{Dec}(.,\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})})\big)$.

Theorem 3 (Correctness). *For all* λ, $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$, *all* $\mathsf{lock} \in \{0,1\}^\eta$, LObf *as given by Construction 2 is a polynomially efficient and correct lockable obfuscation with correctness error*

$$\mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}}(\lambda) \leq n \cdot m \cdot \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Eval}}(\lambda,\delta) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Test}}(\lambda).$$

Proof. From correctness of the FHE scheme we have $\mathsf{ct}_i^{(C)} = \mathsf{FHE}.\mathtt{Eval}(\mathsf{ct}_i, U_x) \approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, C(x))$ and

$$\begin{aligned}\mathsf{ct}_{i,j}^{(\mathsf{Test})} &= \mathsf{FHE}.\mathtt{Eval}\big(\mathsf{ct}_i^{(C)}, \mathsf{SKE}.\mathtt{Dec}(.,\mathsf{ct}_{i,j}^{(\mathsf{lock})})\big)\\ &\approx \mathsf{FHE}.\mathtt{Enc}\big(\mathsf{sk}_i, \mathsf{SKE}.\mathtt{Dec}(C(x),\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})})\big)\end{aligned}$$

with probability of failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Eval}}(\lambda)$ for each $i \in [n]$ and $j \in [m]$.

If $C(x) = \mathsf{lock}$, then we have $\mathsf{ct}_{i,j}^{(\mathsf{Test})} \approx \mathsf{FHE}.\mathtt{Enc}\big(\mathsf{sk}_i, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})\big)$ and $\mathtt{Test}([\mathsf{ct}_{i,j}^{(\mathsf{Test})}]_{i=1,j=1}^{n,m}) = 1$ with probability failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Test}}(\lambda)$.

If $C(x) \neq \mathsf{lock}$, then we have $\mathsf{ct}_{i,j}^{(\mathsf{Test})} \approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \widetilde{\mathsf{msg}}_{i,j})$, where the plaintext is $\widetilde{\mathsf{msg}}_{i,j} = \mathsf{SKE}.\mathtt{Dec}(C(x), \overline{\mathsf{ct}}_{i,j}^{\mathsf{lock}})$. From correctness of the SKE scheme we have that there exists $i \in [n]$ and all $j \in [m]$ such that $\widetilde{\mathsf{msg}}_{i,j} \neq \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ with probability at least $1 - \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda)$. Consequently, we have $\mathtt{Test}([\mathsf{ct}_{i,j}^{\mathsf{Test}}]_{i=1,j=1}^{n,m}) = 0$ with probability of failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Test}}(\lambda)$.

To summarize we have the probability of failure from the lockable obfuscation $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{LObf}}(\lambda) \leq n \cdot m \cdot \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Eval}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Test}}(\lambda)$.

Theorem 4 (Security). *Let* $\mathcal{D}$ *be a class of distribution ensembles* $\{D_\lambda\}_{\lambda\in\mathbb{N}}$ *that sample* $(\mathsf{lock}, C) \leftarrow_{\mathsf{D}} \mathsf{D}_\lambda$*, with* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$, $\mathsf{lock} \in \{0,1\}^\eta$ *and*

$\kappa, \eta = \mathsf{poly}(\lambda)$. *Let* SKE *be a* $\mathcal{D}$-RandCt *secure symmetric key encryption scheme, and* FHE *be a* $\mathsf{IND\text{-}CPA}$ *secure fully homomorphic encryption scheme. Then,* LObf *given by Construction 2, is* $\mathcal{D}$-DVBB *secure.*

Proof. To prove security, we define the simulator to compute the FHE ciphertexts as encryptions of 0, and choose the SKE ciphertexts uniformly at random. Note that the simulator requires only the circuit's dimensions and the size of the lock key.

The hybrid argument is the same as in the proof of Theorem 2, except with the following changes. The hybrids $(i-1)\cdot m + j$ for $i \in [n]$ and $j \in [m]$ is as we define below. After hybrid $n \cdot m$ come hybrids $n \cdot m + i$ for $i \in [n]$ that are the same as the hybrids $2 \cdot n \cdot m + i$ in the proof of Theorem 2. Note that the hybrids $n \cdot m + (i-1) \cdot m + j$ from the proof of Theorem 2, are missing as we no longer use those encryptions.

Now the hybrids $(i-1)\cdot m + j$ that we need to redefine are as follows.

Hybrid $(i-1)\cdot m + j$**:** For $i \in [n]$ and $j \in [m]$ we choose the ciphertext $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j} \leftarrow_{\mathsf{R}} \{0,1\}^{\ell_{\mathsf{sk}}}$ from the uniform distribution instead of computing it as $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j} \leftarrow \mathsf{SKE.Enc}\big(\lambda, \mathsf{lock}, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})\big)$.

Claim. If an adversary A distinguishes between Hybrid $(i-1)\cdot m + j$ and Hybrid $(i-1)\cdot m + j - 1$, then there exists a distinguisher D, that uses A to break $\mathcal{D}$-RandCt security of SKE. We have

$$\big|\Pr[\mathcal{H}_{(i-1)\cdot m+j}] - \Pr[\mathcal{H}_{(i-1)\cdot m+j-1}]\big| = \mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{RandCt}}_{\mathsf{D},\mathsf{SKE}}(\lambda).$$

Proof. First, the solver generates all secret keys of the FHE scheme. For $i' \in [n]$ and $j' \in [m]$ such that $(i'-1)\cdot m + j' < (i-1)\cdot m + j$ the solver chooses $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i',j'} \leftarrow_{\mathsf{R}} \{0,1\}^{\ell_{\mathsf{sk}}}$ uniformly at random. For $(i'-1)\cdot m + j' > (i-1)\cdot m + j$, the solver queries $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i',j'} \leftarrow \mathcal{O}(\mathsf{lock}, \mathsf{F}_{j'}(\mathsf{sk}_{(i' \mod n)+1})$. Finally, the solver submits $\mathsf{msg} = \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ as the challenge, and obtains $\overline{\mathsf{ct}}^{(\mathsf{lock})}_{i,j}$. The rest of the obfuscated program is computed as given by the specification. Then if the adversary outputs that it is in Hybrid $(i-1)\cdot m + j - 1$, then the solver answers that the encrypted message is msg. Otherwise, the solver answers that the ciphertext is uniformly random.

In summary we have that the advantage to distinguish between Hybrid 0 and Hybrid $n \cdot m + n$ is $\mathsf{Adv}^{\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}} \leq n \cdot m \cdot \mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{RandCt}}_{\mathsf{D},\mathsf{SKE}}(\lambda) + n \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda)$.

Remark 1 (Relaxing the Security Requirement on the SKE *Scheme).* From the proof of Hybrids $[1, n\cdot m]$ in the proofs of Theorem 2 and Theorem 4, we observe that we need $\mathcal{D}$-$\mathsf{IND\text{-}CPA}$ (resp. $\mathcal{D}$-RandCt) because we need to encrypt multiple FHE secret keys using the same lock key. Note that in the special case of key dependent message insecure fully homomorphic encryption where $n = 1$ and $m = 1$, we can relax the requirement on SKE to $\mathcal{D}$-semantic security (resp. weak $\mathcal{D}$-pseudorandom ciphertext). Furthermore, for $\mathcal{D}$ being the class of uniform distributions, we can efficiently implement SKE as a one-time pad.

4.2 Extending to Multi-Bit Messages

In this section, we show variants of our lockable obfuscation scheme capable of returning larger messages instead of only a single bit. We show that it is enough to publish encryptions of the bits of the message, and then use the multiplicative homomorphism and the cycle tester to test which bit is encrypted. We extend the idea and show that circular insecure, fully homomorphic encryption schemes are naturally susceptible to key recovery attacks. Finally, we note that we can exploit a full key recovery attack to reduce further the number of ciphertexts that constitute the obfuscated program.

Construction 3 (Multibit Lockable Obfuscation) *Let* $\mathsf{LObf} = (\mathtt{Obf}, \mathtt{Eval})$ *be the lockable obfuscation as given by Construction 1 or Construction 2. Let* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$ *be a circuit and let* $\widehat{C} \leftarrow \mathsf{LObf}.\mathtt{Setup}(\lambda, C, \mathsf{lock})$ *for* $\mathsf{lock} \in \{0,1\}^\eta$*. Denote as* $\mathsf{msg}[l] \in \{0,1\}$ *for* $l \in [\ell_{\mathsf{msg}}]$ *the* l*th bit of a message* msg*. For the message* $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$ *where* $\ell_{\mathsf{msg}} = \mathsf{poly}(\lambda)$*, we extend the obfuscated circuit as follows.*

- *The* $\mathtt{Obf}$ *algorithm additionally computes* $\mathsf{ct}_{i,l}^{(\mathsf{msg})} \leftarrow \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \mathsf{msg}[l])$ *for all* $i \in [n]$ *and* $l \in [\ell_{\mathsf{msg}}]$*. The extended program is*
  - $\widehat{C} = \big([\mathsf{ct}_i]_{i=1}^n, [\mathsf{ct}_{i,l}^{(\mathsf{msg})}]_{i=1,l=1}^{n,\ell_{\mathsf{msg}}}, [\mathsf{ct}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}\big)$ *for base Construction 1, and*
  - $\widehat{C} = \big([\mathsf{ct}_i]_{i=1}^n, [\mathsf{ct}_{i,l}^{(\mathsf{msg})}]_{i=1,l=1}^{n,\ell_{\mathsf{msg}}}, [\overline{\mathsf{ct}}_{i,j}^{(\mathsf{lock})}]_{i=1,j=1}^{n,m}\big)$ *for base Construction 2.*
- *The* $\mathtt{Eval}$ *algorithm upon computing* $\mathsf{ct}_{i,j}^{(\mathsf{Test})}$ *as in Construction 1 or Construction 2, restores the message* msg *as follows.*
  1. *If* $\mathsf{FHE}.\mathtt{Test}([\mathsf{ct}_{i,j}^{(\mathsf{Test})}]_{i=1,j=1}^{n,m}) = 0$*, then return* $\perp$*.*
  2. *For all* $l \in [\ell_{\mathsf{msg}}]$ *do*
     - *For* $i \in [n]$ *and* $j \in [m]$ *compute*

$$\mathsf{ct}_{i,j,l}^{(\mathsf{Test},\mathsf{msg})} \leftarrow \mathsf{FHE}.\mathtt{Eval}\big([\mathsf{ct}_{i,j}^{(\mathsf{Test})}, \mathsf{ct}_{i,l}^{(\mathsf{msg})}], \mathtt{Mul}(.,.)\big).$$

     - *Set* $\mathsf{msg}[l] = \mathsf{FHE}.\mathtt{Test}\big([\mathsf{ct}_{i,j,l}^{(\mathsf{Test},\mathsf{msg})}]_{i=1,j=1}^{n,m}\big)$.

Theorem 5 (Correctness). *For all* $\lambda \in \mathbb{N}$*, all* $C : \{0,1\}^\kappa \mapsto \{0,1\}^\eta$*, all* $\mathsf{lock} \in \{0,1\}^\eta$ *and* $\mathsf{msg} \in \{0,1\}^{\ell_{\mathsf{msg}}}$*,* LObf *given by Construction 3 is a polynomially efficient and correct lockable obfuscation with correctness error*

$$\mathsf{Err}_{\mathsf{LObf}}^{\mathsf{corr}}(\lambda) \leq n \cdot m \cdot \ell_{\mathsf{msg}} \cdot \mathsf{Err}_{\mathsf{FHE}.\mathtt{Eval}}^{\mathsf{corr}}(\lambda, \delta) + \ell_{\mathsf{msg}} \cdot \mathsf{Err}_{\mathsf{FHE}.\mathtt{Test}}^{\mathsf{corr}}(\lambda) + \mathsf{Err}_{\mathsf{SKE}}^{\mathsf{corr}}(\lambda).$$

Proof. Again, polynomial efficiency follows from polynomial efficiency of the underlying primitives. The proofs of correctness for both versions that are based on Construction 1 and Construction 2, follow the proofs of Theorem 2 and Theorem 4, respectively, until the ciphertexts $\mathsf{ct}_{i,j}^{(\mathsf{Test})}$ are computed. Remind that for $C(x) = \mathsf{lock}$ we have $\mathsf{ct}_{i,j}^{(\mathsf{Test})} \approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1}))$. Then from Construction 3 we have

$$\begin{aligned}\mathsf{ct}_{i,j,l}^{(\mathsf{Test},\mathsf{msg})} &= \mathsf{FHE}.\mathtt{Eval}\big([\mathsf{ct}_{i,j}^{(\mathsf{Test})}, \mathsf{ct}_{i,l}^{(\mathsf{msg})}], \mathtt{Mul}(.,.)\big) \\ &\approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1}) \cdot \mathsf{msg}[l])\end{aligned}$$

with probability failure $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Eval}}(\lambda,\delta)$ for all $i \in [n]$, $j \in [m]$ and $l \in [\ell_{\mathsf{msg}}]$. Therefore, if $\mathsf{msg}[l] = 0$, then $\mathtt{Test}([\mathsf{ct}^{(\mathsf{Test},\mathsf{msg})}_{i,j,l}]^{n,m}_{i=1,j=1}) = 0$, and if $\mathsf{msg}[j] = 1$, then $\mathsf{FHE.Test}\big([\mathsf{ct}^{(\mathsf{Test},\mathsf{msg})}_{i,j,l}]^{n,m}_{i=1,j=1}\big) = 1$, with probability of failure $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda)$.

If $C(x) \neq \mathsf{lock}$, then we have $\mathsf{ct}^{(\mathsf{Test},\mathsf{msg})}_{i,j,l} \approx \mathsf{FHE.Enc}(\mathsf{sk}_i, \widetilde{\mathsf{msg}}_{i,j})$, where the plaintext is $\widetilde{\mathsf{msg}}_{i,j} = \mathsf{SKE.Dec}(C(x), \overline{\mathsf{ct}}^{\mathsf{lock}}_{i,j})$. From correctness of the SKE scheme we have that there exists $i \in [n]$ and all $j \in [m]$ such that $\widetilde{\mathsf{msg}}_{i,j} \neq \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ with probability at least $1 - \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda)$. Consequently, we have that, even if $\mathsf{msg}[l] = 1$, the vector $[\mathsf{ct}^{(\mathsf{Test},\mathsf{msg})}_{i,j,l}]^{n,m}_{i=1,j=1}$ does not encode a cycle, and the tester returns 0 with probability failure bounded by $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda)$ and the $\mathsf{LObf.Eval}$ algorithm outputs $\perp$.

Note that in the case $C(x) \neq \mathsf{lock}$, the circuit evaluated by the FHE is smaller, however we upperbound the error with $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Eval}}(\lambda,\delta)$. Furthermore, in the case $C(x) = \mathsf{lock}$, SKE always returns the correct message, but we upperbound the probability of failure with $\mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda)$.

To summarize, we have that the message extraction may fail with probability at least $n \cdot m \cdot \ell_{\mathsf{msg}} \cdot \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Eval}}(\lambda) + \ell_{\mathsf{msg}} \cdot \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE.Test}}(\lambda) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{SKE}}(\lambda)$.

Theorem 6 (Security). *Let* SKE *be a* $\mathcal{D}$-$\mathsf{IND\text{-}CPA}$ *secure symmetric key encryption scheme when using Construction 1 as base, or* $\mathcal{D}$-RandCt *secure when using Construction 2 as base. Let* FHE *be a* $\mathsf{IND\text{-}CPA}$ *secure fully homomorphic encryption scheme. Then,* LObf *given by Construction 1, is* $\mathcal{D}$-DVBB *secure.*

Proof. The proof of Theorem 6 follows the proofs of Theorem 2 and Theorem 4 depending which base construction is used, except with the following changes.

The simulator works as in Theorem 6 or Theorem 4 but it additionally computes $\mathsf{ct}^{(\mathsf{msg})}_{i,l}$ as encryptions of zero. Let L be the number of the last hybrid in the proof of Theorem 2 or Theorem Theorem 4. We additionally define the following sequence of hybrids.

Hybrid $L + n \cdot (i-1) + l$**:** For $i \in [n]$ and $l \in [\ell_{\mathsf{msg}}]$, instead of computing $\mathsf{ct}^{(\mathsf{msg})}_{i,l} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, \mathsf{msg}[l])$, we compute $\mathsf{ct}^{(\mathsf{msg})}_{i,l} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_i, 0)$.

Claim. If an adversary A distinguishes between Hybrid $L+n\cdot(i-1)+l$ and Hybrid $L + n \cdot (i-1) + l - 1$, then there exists a distinguisher D, that uses A to break $\mathsf{IND\text{-}CPA}$ security of FHE. We have

$$\big|\Pr[\mathcal{H}_{L+n\cdot(i-1)+l}] - \Pr[\mathcal{H}_{L+n\cdot(i-1)+l-1}]\big| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda).$$

Proof. First, the solver generates all secret keys of the FHE scheme except sk_i. The solver generates all ciphertexts as in Hybrid $L+n\cdot(i-1)+l-1$, except for the ciphertext $\mathsf{ct}^{(\mathsf{msg})}_{i,l}$. To obtain $\mathsf{ct}^{(\mathsf{msg})}_{i,l}$ the solver sets the challenge query as $\mathsf{msg}_0 = \mathsf{msg}[l]$ and $\mathsf{msg}_1 = 0$. All other ciphertexts for the secret key are obtained by querying $\mathcal{O}$ on messages as in Hybrid $L + n \cdot (i-1) + l - 1$.
If the adversary outputs that it is Hybrid $L+n\cdot(i-1)+l-1$, then the solver answers that the encrypted message is msg_0. Otherwise, the solver answers that the message is msg_1.

Finally, we have that for

- the base Construction 1, the adversary's advantage of distinguish between hybrid 0 and hybrid $L + n \cdot \ell_{\mathsf{msg}}$ is

$$\mathsf{Adv}^{\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}} \leq n \cdot m \cdot \mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{D},\mathsf{SKE}}(\lambda) + n \cdot (m + \ell_{\mathsf{msg}}) \cdot \mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda)$$

and for

- the base Construction 2, the adversary's advantage of distinguish between hybrid 0 and hybrid $L + n \cdot \ell_{\mathsf{msg}}$ is

$$\mathsf{Adv}^{\mathsf{DVBB}}_{\mathsf{A},\mathsf{LObf}} \leq n \cdot m \cdot \mathsf{Adv}^{\mathcal{D}\text{-}\mathsf{RandCt}}_{\mathsf{D},\mathsf{SKE}}(\lambda) + n \cdot (1 + \ell_{\mathsf{msg}}) \cdot \mathsf{Adv}^{\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{D},\mathsf{FHE}}(\lambda)$$

Key Recovery Attack. We show that given a key cycle for any circular insecure fully homomorphic encryption, it is possible to decode the key material. The idea follows from Construction 3.

Construction 4 (The Key Recovery Attack) *Let* $\mathsf{FHE} = (\mathtt{Setup}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Eval}, \mathtt{Test})$ *be a fully homomorphic encryption scheme with a cycle tester. We build the algorithm* `KeyRecovery` *as follows:*

$\mathtt{KeyRecovery}([\mathsf{ct}_i]_{i=1,j=1}^{n,m})$**:** *Takes as input a vector of ciphertexts* $[\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m}$ *and returns a vector* $[\widetilde{\mathsf{sk}}_{i,j}]_{i=1,j=1}^{n,m}$.

1. *Let* $\ell \geq \lceil \log_2 \mathsf{F}_j(.) \rceil$ *for all* $j \in [m]$.
2. *For* $i \in [n]$, $j \in [m]$ *and* $l \in \ell$
   - *Compute* $\mathsf{ct}_l^{(\mathsf{Bit},i,j)} \leftarrow \mathsf{FHE}.\mathtt{Eval}\big(\mathsf{ct}_{i,j}, \mathtt{GetBit}(.,l)\big)$, *where* $\mathtt{GetBit}(x,l)$ *is a circuit that returns the* l*th bit of* x.
   - *For* $i' \in [n]$ *and* $j' \in [m]$
     - *If* $i' = i$, *then set*

$$\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} \leftarrow \mathsf{FHE}.\mathtt{Eval}\big([\mathsf{ct}_{l,i'}^{(\mathsf{Bit},i,j)}, \mathsf{ct}_{i',j'}], \mathtt{Mul}(.,.)\big).$$

     - *Otherwise set* $\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} \leftarrow \mathsf{ct}_{i',j'}$.
   - *Set* $\widetilde{b}_{i,j,l} \leftarrow \mathtt{Test}\big([\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)}]_{i'=1,j'=1}^{n,m}\big)$.
3. *For* $i \in [n]$ *and* $j \in [m]$ *compute* $\widetilde{\mathsf{sk}}_{i,j} \leftarrow \sum_{l=1}^{\ell} \widetilde{b}_{i,j,l} \cdot 2^{l-1}$.
4. *Return* $[\widetilde{\mathsf{sk}}_{i,j}]_{i=1,j=1}^{n,m}$.

Theorem 7 (Correctness). *For* $i \in [n]$ *and* $j \in [m]$ *let* $\mathsf{ct}_{i,j} \approx \mathsf{FHE}.\mathtt{Enc}(\mathsf{sk}_i, \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1}))$. *Let* C_{KR} *be the circuit that* `KeyRecovery` *homomorphically computes on each ciphertext* $\mathsf{ct}_{i,j}$ *until it obtains* $\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)}$. *Denote as* δ *the depth of* C_{KR}. *Let* $[\widetilde{\mathsf{sk}}_{i,j}]_{i=1,j=1}^{n,m} \leftarrow \mathtt{KeyRecovery}([\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m})$. *Then the equation* $\widetilde{\mathsf{sk}}_{i,j} = \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$ *holds for all* $i \in [n]$ *and all* $j \in [m]$ *with probability* $1 - \big(\mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathtt{Eval}}(\lambda, \delta) + \mathsf{Err}^{\mathsf{corr}}_{\mathsf{FHE}.\mathsf{Test}}(\lambda)\big)$.

Proof. Let us denote as $b_{i,j,l} \in \{0,1\}$ the bits which satisfy $\mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1}) = \sum_{l=1}^{\ell} b_{(i \mod n)+1,j,l} \cdot 2^{l-1}$. From correctness of the FHE we have that $\mathsf{ct}_l^{(\mathsf{Bit},i,j)} = \mathsf{FHE.Eval}(\mathsf{ct}_{i,j}, \mathtt{GetBit}(.,l)) \approx \mathsf{FHE.Enc}(\mathsf{sk}_i, b_{i,j,l})$.

Again from correctness of the FHE we have

$$\begin{aligned}\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} &= \mathsf{FHE.Eval}\big([\mathsf{ct}_{l,i'}^{(\mathsf{Bit},i,j)}, \mathsf{ct}_{i',j'}], \mathtt{Mul}(.,.)\big) \\ &\approx \mathsf{FHE.Enc}\big(\mathsf{sk}_{i'}, b_{i,j,l} \cdot \mathsf{F}_{j'}(\mathsf{sk}_{(i' \mod n)+1})\big),\end{aligned}$$

for $i' = i$. For $i' \neq i$, we set $\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} = \mathsf{ct}_{i,j}$. Now observe that the vector $[\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)}]_{i'=1,j'=1}^{n,m}$ decrypts to the same messages as the vector $[\mathsf{ct}_{i,j}]_{i=1,j=1}^{n,m}$ if $b_{i,j,l} = 1$. If $b_{i,j,l} = 0$, then the ciphertexts $\mathsf{ct}_{i,j'}^{(\mathsf{Test},i,j)}$ are ciphertexts of 0, and the cycle is broken. Hence from correctness of the cycle tester we have $\widetilde{b}_{i,j,l} = 0$ if $b_{i,j,l} = 0$, since $\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} \approx \mathsf{FHE.Enc}(\mathsf{sk}_{i'}, 0)$, and $\widetilde{b}_{i,j,l} = 1$ if $b_{i,j,l} = 1$, since $\mathsf{ct}_{i',j'}^{(\mathsf{Test},i,j)} \approx \mathsf{FHE.Enc}(\mathsf{sk}_{i'}, \mathsf{F}_{j'}(\mathsf{sk}_{(i' \mod n)+1}))$. Finally, from the definition we have that $\widetilde{\mathsf{sk}}_{i,j} = \sum_{l=1}^{\ell} \widetilde{b}_{i,j,l} \cdot 2^{l-1} = \mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})$.

Remark 2 (Further simplification of Multibit Lockable Obfuscation). At this point we believe it is easy to see, that we can reduce the size of the obfuscated program given by Construction 3, by publishing $\mathsf{ct}_{\mathsf{msg}} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}_1, \mathsf{msg})$ instead of $[\mathsf{ct}_{\mathsf{msg}_i,j}]_{j=1}^{\ell_{\mathsf{msg}}})]_{i=1}^{n}$. The idea to decrypt the message from $\mathsf{ct}_{\mathsf{msg}}$, is to run the attack given by Construction 3, i.e., recover all secret keys for the FHE scheme, including sk_1. Note that we assume that it is feasible to recover the secret keys given $[\mathsf{F}_j(\mathsf{sk}_{(i \mod n)+1})]_{i=1,j=1}^{n,m}$. Finally, we compute $\mathsf{msg} \leftarrow \mathsf{FHE.Dec}(\mathsf{sk}_1, \mathsf{ct}_{\mathsf{msg}})$.

5 Conclusions

We believe that our lockable obfuscators are intuitive and easy to understand. Our algorithms exemplify, alongside the work from Kluczniak [Klu20], that circular insecure encryption is a useful building block for advanced cryptographic primitives. It is worth noting that circular insecure encryption was previously constructed solely out of theoretical curiosity.

As mentioned in the introduction, the main aim of this paper is to introduce and analyze a general methodology of building lockable obfuscators. In particular, we leave concrete instantiations of our methods to future work. An exciting direction would be whether, for instance, we can use existing fully homomorphic encryption schemes together with the cycle testers in [BHW15, KW16, AP16] to build more efficient lockable obfuscation without the need to obfuscate branching programs.

Funding. This work has been partially funded/supported by the German Ministry for Education and Research through funding for the project CISPA-Stanford Center for Cybersecurity (Funding number 16KIS0927).

References

ABBC10. Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 403–422. Springer, Heidelberg, May / June 2010.

ADN+10. Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 113–134. Springer, Heidelberg, May / June 2010.

Agr19. Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 191–225. Springer, Heidelberg, May 2019.

AGV09. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, Heidelberg, March 2009.

AJL+19. Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 284–332. Springer, Heidelberg, August 2019.

AKPW13. Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer, Heidelberg, August 2013.

AL20. Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 123–152. Springer, Heidelberg, November 2020.

AP16. Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (ring-)LWE. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 659–680. Springer, Heidelberg, August 2016.

BDGM20a. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 79–109. Springer, Heidelberg, May 2020.

BDGM20b. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE

suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. https://eprint.iacr.org/2020/1024.

BG10. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 1–20. Springer, Heidelberg, August 2010.

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, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, August 2001.

BGI+12. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2), May 2012.

BHJ+19. Boaz Barak, Samuel B. Hopkins, Aayush Jain, Pravesh Kothari, and Amit Sahai. Sum-of-squares meets program obfuscation, revisited. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 226–250. Springer, Heidelberg, May 2019.

BHW15. Allison Bishop, Susan Hohenberger, and Brent Waters. New circular security counterexamples from decision linear and learning with errors. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 776–800. Springer, Heidelberg, November / December 2015.

BKM+18. Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 731–752. Springer, Heidelberg, August 2018.

BKP19. Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 1091–1102. ACM Press, June 2019.

BKSW18. Saikrishna Badrinarayanan, Dakshita Khurana, Amit Sahai, and Brent Waters. Upgrading to functional encryption. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 629–658. Springer, Heidelberg, November 2018.

BLMZ19. James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New techniques for obfuscating conjunctions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 636–666. Springer, Heidelberg, May 2019.

BLSV18. Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes*

in Computer Science, pages 535–564. Springer, Heidelberg, April / May 2018.

BR13. Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 416–434. Springer, Heidelberg, August 2013.

BS16. Mihir Bellare and Igors Stepanovs. Point-function obfuscation: A framework and generic constructions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 565–594. Springer, Heidelberg, January 2016.

BS20. Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd Annual ACM Symposium on Theory of Computing*, pages 269–279. ACM Press, June 2020.

BVWW16. Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In Madhu Sudan, editor, *ITCS 2016: 7th Conference on Innovations in Theoretical Computer Science*, pages 147–156. Association for Computing Machinery, January 2016.

BW19. Ward Beullens and Hoeteck Wee. Obfuscating simple functionalities from knowledge assumptions. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 254–283. Springer, Heidelberg, April 2019.

Can97. Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, Heidelberg, August 1997.

CD08. Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 489–508. Springer, Heidelberg, April 2008.

CGH12. David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 540–557. Springer, Heidelberg, May 2012.

CKVW10. Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 52–71. Springer, Heidelberg, February 2010.

CRV10. Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, Heidelberg, February 2010.

CVW$^+$18a. Yilei Chen, Vinod Vaikuntanathan, Brent Waters, Hoeteck Wee, and Daniel Wichs. Traitor-tracing from LWE made simple and attribute-based.

In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 341–369. Springer, Heidelberg, November 2018.

CVW18b. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 577–607. Springer, Heidelberg, August 2018.

DGK+10. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, Heidelberg, February 2010.

DHLW10. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 613–631. Springer, Heidelberg, December 2010.

DKL09. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 621–630. ACM Press, May / June 2009.

DORS08. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, March 2008.

DP08. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual Symposium on Foundations of Computer Science*, pages 293–302. IEEE Computer Society Press, October 2008.

DRS04. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, Heidelberg, May 2004.

DS05. Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 654–663. ACM Press, May 2005.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, May / June 2009.

GGH15. Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527. Springer, Heidelberg, March 2015.

GJLS20. Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. Cryptology ePrint Archive, Report 2020/764, 2020. https://eprint.iacr.org/2020/764.

GKPV10. Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240. Tsinghua University Press, 2010.

GKVW20. Rishab Goyal, Venkata Koppula, Satyanarayana Vusirikala, and Brent Waters. On perfect correctness in (lockable) obfuscation. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 229–259. Springer, Heidelberg, November 2020.

GKW17a. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 612–621. IEEE Computer Society Press, October 2017.

GKW17b. Rishab Goyal, Venkata Koppula, and Brent Waters. Separating IND-CPA and circular security for unbounded length key cycles. In Serge Fehr, editor, *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 10174 of *Lecture Notes in Computer Science*, pages 232–246. Springer, Heidelberg, March 2017.

GKW17c. Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 528–557. Springer, Heidelberg, April / May 2017.

GKW18. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 660–670. ACM Press, June 2018.

GP20. Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. Cryptology ePrint Archive, Report 2020/1010, 2020. https://eprint.iacr.org/2020/1010.

HG01. Nick Howgrave-Graham. Approximate integer common divisors. In Joseph H. Silverman, editor, *Cryptography and Lattices*, pages 51–66, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

HH09. Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 202–219. Springer, Heidelberg, March 2009.

HILL99. Johan HÅstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, March 1999.

HLR07. Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 169–186. Springer, Heidelberg, May 2007.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon,*

USA, June 21-25, 1998, Proceedings, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

JLMS19. Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 251–281. Springer, Heidelberg, May 2019.

JLS20. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003, 2020. https://eprint.iacr.org/2020/1003.

Klu20. Kamil Kluczniak. Witness encryption from garbled circuit and multikey fully homomorphic encryption techniques. Cryptology ePrint Archive, Report 2020/1502, 2020. https://eprint.iacr.org/2020/1502.

Klu21. Kamil Kluczniak. Lockable obfuscation from circularly insecure fully homomorphic encryption. Cryptology ePrint Archive, Report 2021/1324, 2021. https://ia.cr/2021/1324.

KRW15. Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 378–400. Springer, Heidelberg, March 2015.

KW16. Venkata Koppula and Brent Waters. Circular security separations for arbitrary length cycles from LWE. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 681–700. Springer, Heidelberg, August 2016.

KY18. Ilan Komargodski and Eylon Yogev. Another step towards realizing random oracles: Non-malleable point obfuscation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 259–279. Springer, Heidelberg, April / May 2018.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, Heidelberg, May / June 2010.

LPS04. Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 20–39. Springer, Heidelberg, May 2004.

LPST16. Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 9615 of *Lecture Notes in Computer Science*, pages 447–462. Springer, Heidelberg, March 2016.

LT17. Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 630–660. Springer, Heidelberg, August 2017.

NS09. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer, Heidelberg, August 2009.

Pie09. Krzysztof Pietrzak. A leakage-resilient mode of operation. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 462–482. Springer, Heidelberg, April 2009.

RAD+78. Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.

Rot13. Ron Rothblum. On the circular security of bit-encryption. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 579–598. Springer, Heidelberg, March 2013.

Wee05. Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 523–532. ACM Press, May 2005.

WW20. Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. Cryptology ePrint Archive, Report 2020/1042, 2020. https://eprint.iacr.org/2020/1042.

WZ17. Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 600–611. IEEE Computer Society Press, October 2017.

YZ16. Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 214–243. Springer, Heidelberg, August 2016.

Zha16. Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 479–508. Springer, Heidelberg, August 2016.