Maliciously Circuit-Private FHE

Rafail Ostrovsky[1,⋆], Anat Paskin-Cherniavsky[2,⋆⋆], and Beni Paskin-Cherniavsky[3]

[1] Department of Computer Science and Mathematics, UCLA, USA
rafail@cs.ucla.edu
[2] Department of Computer Science, UCLA, USA
anpc@cs.ucla.edu
[3] cben@users.sf.net

Abstract. We present a framework for transforming FHE (fully homomorphic encryption) schemes with no circuit privacy requirements into maliciously circuit-private FHE. That is, even if both maliciously formed public key and ciphertext are used, encrypted outputs only reveal the evaluation of the circuit on some well-formed input x^*. Previous literature on FHE only considered semi-honest circuit privacy. Circuit-private FHE schemes have direct applications to computing on encrypted data. In that setting, one party (a receiver) holding an input x wishes to learn the evaluation of a circuit C held by another party (a sender). The goal is to make receiver's work sublinear (and ideally independent) of $|C|$, using a 2-message protocol. The transformation technique may be of independent interest, and have various additional applications. The framework uses techniques akin to Gentry's bootstrapping and conditional disclosure of secrets (CDS [AIR01]) combining a non circuit private FHE scheme, with a homomorphic encryption (HE) scheme for a smaller class of circuits which is maliciously circuit-private. We devise the first known circuit private FHE, by instantiating our framework by various (standard) FHE schemes from the literature.

Keywords: Fully homomorphic encryption, computing on encrypted data, privacy, malicious setting.

⋆ Work supported in part by NSF grants 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014 -11 -1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

⋆⋆ Work supported in part by NSF grants 09165174, 1065276, 1118126 and 1136174. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014 -11 -1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

J.A. Garay and R. Gennaro (Eds.): CRYPTO 2014, Part I, LNCS 8616, pp. 536–553, 2014.

1 Introduction

In this paper, we devise a first fully homomorphic encryption scheme (FHE) [Gen09] that satisfies (a meaningful form of) circuit privacy in the malicious setting—a setting where the public key and ciphertext input to Eval are not guaranteed to be well-formed. We present a framework for transforming FHE schemes with no circuit privacy requirements into maliciously circuit-private FHE. The transformation technique may be of independent interest, and have various additional applications. The framework uses techniques akin to Gentry's bootstrapping and conditional disclosure of secrets (CDS [AIR01]) combining a non circuit private FHE scheme, with a homomorphic encryption (HE) scheme for a smaller class of circuits which is maliciously circuit-private. We then demonstrate an instantiation of this framework using schemes from the literature.

The notion of FHE does not require circuit privacy even in the semi-honest setting (but rather standard IND-CPA security, the ability to evaluate arbitrary circuits on encrypted inputs and encrypted outputs being "compact"). In [Gen09] and [vDGHV09, appendix C], the authors show how to make their FHE schemes circuit-private in the semi-honest setting.

One natural application of (compact) FHE is the induced 2-message, 2-party protocol, where a receiver holds an input x, and a sender holds a circuit C; the receiver learns $C(x)$, while the sender learns nothing. In the first round the receiver generates a public-key pk, encrypts x to obtain c, and sends (pk, c). The sender evaluates C on (pk, c) using the schemes' homomorphism, and sends back the result. An essential requirement is that receiver's work (and overall communication) is $poly(k, n, o(|C|))$, where k is a security parameter, ideally independent of $|C|$ altogether. This application of homomorphic encryption, termed *computing on encrypted data*, was studied both in several works [IP07, BKOI07] predating Gentry's first fully homomorphic scheme, and mentioned in [Gen09].

The underlying scheme's IND-CPA security translates into the standard simulation-based notion of *privacy* against a malicious sender in the stand-alone model[1] (but not any form of correctness against a malicious sender). The circuit privacy of the scheme translates into a privacy guarantee against a malicious receiver (of the same "flavor"). While standard FHE (without extra requirements) does not imply any security guarantees against malicious receivers, the semi-honestly circuit-private schemes from (e.g.) [vDGHV09] imply standard simulation-based security against semi-honest receivers. Thus, a maliciously circuit-private scheme induces a protocol which is private against malicious corruptions in the stand-alone model.

Let us now define maliciously circuit-privacy of FHE more precisely. We say a scheme is circuit-private if it satisfies the following privacy notion ala [IP07], stating that any (pk, c) pair induces some "effective" encrypted input x^*:

[1] Privacy against a malicious sender comes "for free", as the protocol is 2-round, and the client speaks first.

Definition 1. *(informal). We say a $\mathcal{C}$-homomorphic[2] encryption scheme* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is (maliciously) circuit-private if there exists an unbounded algorithm* Sim*, such that for all security parameters k, and all pk^*, c^* there exists x^*, such that for all circuits $C \in \mathcal{C}$ over $|x^*|$ variables* $\mathsf{Sim}(1^k, C(x^*)) =^s \mathsf{Eval}(1^k, pk^*, C, c^*)$ *(statistically indistinguishable). We say the scheme is semi-honestly circuit-private if the above holds only for* well-formed pk^*, c^* *pairs.*

An FHE satisfying Definition 1 induces a protocol private against a malicious sender (by IND-CPA security of the FHE), but private against unbounded malicious receivers with unbounded simulation.[3]

On one hand, this privacy notion is weaker compared to full security as the simulation is not efficient; on the other hand, it is stronger in the sense that it holds against unbounded adversaries as well.

Due to impossibility results for general 2-round sender-receiver computation in the plain model (e.g. [BLV04]), this notion has become standard in the (non-interactive, plain model) setting of computing on encrypted data [NP01, AIR01, HK12, IP07] as a plausible relaxation.

It is important to note that we only consider the plain model. If preprocessing, such as CRS was allowed, the malicious case could be easily reduced to the semi-honest case. That is, given CRS, Enc could have added a NIZK proving that the key is well-formed, and that the ciphertext is a valid ciphertext under that public key. Then, Eval could explicitly check that the proof is valid, if not return $\perp$, otherwise run Eval as for the semi-honest setting (for that scheme). Some care needs to be taken even in this setting, so that the scheme for the semi-honest setting used has somewhat enhanced privacy. More specifically, it needs to hold assuming (pk, c) are in the support of valid pk and $c \in Enc_{pk}(\cdot)$ respectively, but not that the *distribution* of (pk, c) is identical to the honestly generated one. Indeed, such a semi-honestly circuit-private scheme has been put forward in [GHV10] (when applied to a perfectly correct FHE scheme). On the other hand, the semi-honestly private scheme suggested in [vDGHV09] needs that pk has the proper distribution ($\mathsf{KeyGen}(1^k)$), rather than just being in the support of that distribution.

To summarize, our main "take home" theorem is as follows.

Theorem 1. *(informal) Assume an FHE scheme $\mathcal{F}$ with decryption circuits in* NC^1 *exists. Assume further there exists a maliciously circuit private HE $\mathcal{B}$ that supports bit OT exists. Then, there exists a maliciously circuit private multi-hop FHE scheme.*

There exist several instantiations of the theorem by ingredients from the literature. All known FHE schemes from the literature, such as [Gen09, vDGHV09, BV11] have efficient decryption circuits as required by the Theorem. Some candidates for $\mathcal{B}$ are [NP01, AIR01, HK12].

[2] In particular, for fully homomorphic schemes, $\mathcal{C}$ is the class of all circuits.

[3] Jumping ahead, settling for computational indistinguishability with unbounded simulation would allow for somewhat simplified constructions. However, we shoot for the best achievable privacy notion.

In terms of implications to MPC, our result can be interpreted as non-interactive 2PC protocols with asymmetric inputs as follows.

Theorem 2. *(informal) Assume the preconditions of Theorem 1 hold. Then, there exist 2-message client-server MPC protocols where the client holds x, and the server holds $n = 1^{|x|}$ a circuit C with n inputs (which may be much larger then x), and 1^k a security parameter. The client learns $C(x)$ (but nothing else, not even $|C|$), and the server learns nothing about x (but $|x|$). The privacy guarantee for the client is standard simulation-based computational privacy. The privacy guarantee for the server is based on unbounded simulation (against possibly unbounded clients). The protocols's communication is at most $poly(n, k)$ (as the client is efficient in its own input).* [4]

Multi-hop circuit-private FH. It is a desirable property of an FHE scheme that the outputs of Eval applied to any given circuit C mapping $\{0,1\}^n$ to $\{0,1\}^m$ can be fed again into Eval running on a circuit taking $\{0,1\}^m$ as input, and so on - an unbounded number of times. In the terminology of [GHV10], this property of a HE scheme is referred to as multi-hop. If only upto some i such iterations are supported, the scheme is called i-hop. The standard definition of FHE, thus corresponds to 1-hop encryption. However, there exists a simple transformation for any compact FHE into multi-hop. This is done by including an encryption of the secret key in the public key, and homomorphically decrypting the encrypted outputs received using the encrypted key bits (as in Gentry's bootstrapping theorem [Gen09, GHV10]). The maliciously circuit-private FHE resulting from our construction is also designed to be only 1-hop, but the standard transformation does not make it multi-hop, as it does not preserve malicious circuit-privacy.

In Section 3.3, we define multi-hop maliciously circuit-private HE, and sketch a modification to our 1-hop scheme, making it multi-hop. Our transformation starts with the above transformation, and adds some validation of the added key bits.

1.1 Previous Work

Circuit private FHE implicit in work on MPC. As explained above, (compact) HE naturally gives rise to non-interactive client-server protocols for computing on encrypted data (and the privacy level of the protocol depends on the notion of circuit privacy of the FHE). An essential requirement is that client's work in these protocols is sub-linear in $|C|$ (ideally independent of $|C|$).

In the other direction, standard two-message client-server protocols (inputs of similar length, only client learns output), robust against malicious receivers,

[4] In fact, the above result can be interpreted as general "size hiding" 2PC with asymmetric inputs. The case in Theorem 2 is a special case with $F(x, y)$ being the universal function of evaluating a circuit y on input x. In the general case, F is some polynomial-time computable function. The client learns $F(x, y)$ (but not even $|y|$), and the server learns only $|x|$. This can be implemented by letting the server set $C = F_y(x)$, and run the protocol from Theorem with input $C, |x|$ for the server and input x for the client.

induce circuit-private (not necessarily compact) HE, when applied with the universal function [CCKM00,GHV10]. Roughly the round-1 message of the client is viewed as an encryption for his input, and the senders' reply as Eval's output. This is still not an encryption scheme, as in the protocol, the client needs to "remember" the randomness generating a round-1 message in order to "decode" the reply. This is solved by setting KeyGen output a public-key, private-key pair of some public key encryption scheme as (pk, sk) respectively. Enc is augmented concatenate an encryption c_r of its randomness under pk. Eval is augmented to pass c_r as is. Intuitively, if the protocol was private against malicious clients, then so is the encryption scheme (as the randomness was known to the client anyway).

One specific construction of HE from 2PC is by combining an information-theoretic version of Yao's garbled circuits with oblivious transfer (OT) secure against malicious receivers [NP01, AIR01, HK12, IP07]. As information-theoretic Yao is only efficient for NC^1, the resulting HE scheme captures only functions in NC^1. Also, this generic construction, even in the semi-honest setting (for all known 2PC protocols from then literature) has encrypted output size at least $|C|$, while the crux of HE is having compact encrypted outputs—ideally $poly(k)$, as modern FHE schemes achieve.

Another relevant work is a recent work on 2-message 2-party evaluation of a public function $f(x, y)$, where the work of the party holding y (wlog.) is $poly(k, n, \log |f|)$, where $|f|$ is the size of f's circuit representation [DFH12]. Their protocol is maliciously UC-secure [Can01]. Instantiating f with the universal function for evaluating circuits, results in HE with good compactness properties for circuits of certain size. However, this protocol requires CRS, and thus does not translate into (circuit private) HE in the plain model.

Circuit privacy in HE literature. Without the circuit privacy requirement, FHE candidates have been proposed in a line of work following the seminal work of Gentry [Gen09, vDGHV09, BV11], to mention a few. However, these works typically do not have circuit privacy as a goal.

Circuit privacy in the semi-honest setting, for properly generated pk and c has been addressed in [Gen09, vDGHV09]. In both works, the solution method is akin to that used in additively homomorphic cryptosystems [GM84, DJ01]. The idea in the latter is to (homomorphically) add a fresh encryption of 0. In FHE, the situations is a bit more complicated, as the output of Eval typically has a different domain than "fresh" encryptions, so adding a 0 is not straightforward. However, a generalization of this technique often works (see e.g. [vDGHV09]).

Another approach suggested in [vDGHV09] for the semi-honest setting is replacing the encrypted output c with a Yao garbled circuit for decryption (with sk, c as inputs), thereby transforming any scheme into a semi-honestly circuit-private one.

The work of [GHV10] considers a generalization of circuit privacy of HE (referred there as function privacy) to a setting with multiple evaluators and single encryptor (and decryptor), where all but a single evaluator E_i can collude to learn extra information about E_i's circuit. Among other contributions, in [GHV10], the

authors further abstract the Yao-based approach from [vDGHV09] as a combination of two HE scheme, one compact but not private, the other (semi-honestly) private but not compact, so that the result is both compact and (semi-honestly) private. We use this transformation as is as a first step in our transformation (along with some additional ideas formulated in [GHV10]).

As mentioned above, the malicious setting (with compact encrypted outputs) has been addressed in the context of Oblivious Transfer (OT) [NP01, AIR01, HK12] (these works can be viewed as HE for the limited class of Oblivious Transfer functions). For broader classes of functions [IP07] devise maliciously circuit-private HE for depth-bounded *branching programs* (with partial compactness).

All of the above schemes use the Conditional Disclosure of Secrets (CDS) methodology [GIKM98]. CDS is a light-weight alternative to zero-knowledge proofs, that receives a secret string, an encryption c of some x in an HE, and the corresponding pk. It discloses the secret iff. x satisfies a certain condition. CDS was originally defined for well-formed pk, c, leading to semi-honestly circuit private HE constructions [AIR01]. The CDS from [AIR01] works for additive HE with ciphertexts over groups of a prime order, and [Lip05] generalized it to groups of sufficiently "rough" composite order. For specific groups, the technique of [Lip05] turned out to generalize to situations where (pk, c) may not be well-formed. Roughly, the secret luckily remains hidden even if the CDS was obliviously performed on the (possibly malformed) encryptions and pk as if they were proper. Such CDS was used in [HK12, IP07] to obtain maliciously circuit-private HE.

1.2 Our Techniques

We devise a framework for transforming FHE schemes with no circuit privacy requirements into maliciously circuit-private FHE. We use 2 ingredients which have implementations in the literature: powerful (evaluate circuits) compact FHE without privacy $\mathcal{F}$, and weak (evaluate formulas) non-compact maliciously circuit-private HE $\mathcal{P}$ (by "compact" we refer to the strong requirement that encrypted outputs have size $poly(k)$, where k is the security parameter). Our construction proceeds in three steps:

Lemma 1 ([GHV10]). *A compact FHE without privacy can be upgraded to (compact) semi-honestly circuit private by decrypting its encrypted output under a (possibly non-compact) enhanced semi-honestly private HE, capable of evaluating the decryption circuit. The resulting scheme has enhanced semi-honest circuit privacy, assuming only that pk, c are in the support of honestly generated pairs, rather then being distributed as honestly generated pairs.*

Lemma 2 (this paper). *An enhanced semi-honestly circuit-private FHE can be upgraded to maliciously circuit-private by homomorphically validating its keys and inputs under a (possibly non-compact) maliciously circuit-private FHE capable of evaluating a circuit validating that (pk, c) are well-formed (provided a suitable witness as additional input).*

The output resulting from composing these two steps is not compact—but fortunately:

Lemma 3 (this paper,same construction as [GHV10] for semi-honest setting). *Any circuit-private FHE can be upgraded to compact by homomorphically decrypting its output under a compact (F)HE (while preserving circuit-privacy).*

Let us now elaborate on each of the steps.

Step 1. The first step transforms a "main" (compact) FHE scheme $\mathcal{M}_1$ into a semi-honestly circuit-private scheme ($\mathcal{M}_2$). An output encrypted via $\mathsf{Eval}_{\mathcal{M}_1}$ may contain extra information about the structure of C (though limited to $poly(k)$ since $\mathcal{M}_1$ is compact). An easy way to strip all information beyond the value of the function is to decrypt it already during Eval under an "auxiliary" scheme $\mathcal{A}_1$ which is semi-honestly circuit-private. To make sure the evaluator learns no secret information (sk_M, x), the decryption is done "blindly", using $\mathcal{A}_1$'s homomorphic properties. Details follow.

KeyGen generates a public key $pk = (pk_{\mathcal{M}_1}, pk_{\mathcal{A}_1}, a_{sk_{\mathcal{M}_1}} = \mathsf{Enc}_{\mathcal{A}_1}(sk_{\mathcal{M}_1}))$ and secret key $sk_{\mathcal{A}_1}$. Enc simply outputs $c_{\mathcal{M}_1} = \mathsf{Enc}_{\mathcal{M}_1}(x)$. Now, Eval first computes $out_{\mathcal{M}_1} = \mathsf{Eval}_{\mathcal{M}_1}(C, c_{\mathcal{M}_1})$, and outputs $\mathsf{Eval}_{\mathcal{A}_1}(\mathsf{Dec}_{\mathcal{M}_1}, (out_{\mathcal{M}_1}, a_{sk_{\mathcal{M}_1}}))$.[5] Dec simply applies $\mathsf{Dec}_{\mathcal{A}_1}$ to Eval's output.

Enhanced semi-honest circuit privacy of the resulting scheme follows by (semi-honest) circuit privacy of $\mathcal{A}_1$, and the correctness of $\mathcal{M}_1$. For enhanced circuit privacy, we assume perfect correctness of $\mathcal{M}_1$ rather then allowing for negligible decryption error, as is common in the FHE literature. The reason is that otherwise, the receiver could pick pairs (pk, c) for which the output of Eval is not $C(x)$ with high probability over Eval's randomness, and potentially reveal a bit about the circuit not consistent with x.

Note that since $\mathcal{M}_1$ is compact $|out_{\mathcal{A}_1}| = \mathsf{poly}(k)$ even if $\mathcal{A}_1$ is not compact. $\mathcal{M}_1, \mathcal{A}_1$ can be instantiated via almost any FHE from the literature, and (non-compact) semi-honestly circuit-private HE obtained from non-interactive 2PC protocols, such as Yao-based protocols (see above). In particular, although most FHE schemes from the literature have (negligible) decryption errors, they can be modified to have perfect correctness, while maintaining security.[6]

[5] The trick of "re-encrypting" under $\mathcal{A}_1$ using $\mathcal{A}_1$'s own Eval procedure to perform the decryption is similar in Gentry's bootstrapping technique in [Gen09] for transforming a "somewhat homomorphic" scheme into (unleveled) FHE. One difference is that here we can use two different schemes. Another difference is that for the purpose of reducing noise via Gentry's bootstrapping theorem, it is important to hardwire c as a string into the decryption circuit, rather then supplying an encryption of it. In step 1, we can afford introducing c into Dec in either way.

[6] Consider for example the [BV11] scheme can be modified to use Gaussian noise truncated to a value which still does not incur decryption errors. It is easy to prove that the new scheme remains secure under the same (LWE) assumption - essentailly because samples larger then some bound hae negligible probability of being sampled. A similar trick can be applied to other LWE-based FHE schemes [BV11, Bra12, GSW13].

Step 2. The above approach generally fails in the malicious setting, even with stronger ingredients. Let $\mathcal{A}_2$ be a maliciously circuit-private scheme, and $\mathcal{M}_2$ be the semi-honestly circuit-private FHE resulting from step 1. An obvious attempt is using $\mathcal{A}_2$ instead of $\mathcal{A}_1$ in the first decryption; another is repeating the construction, taking $\mathcal{M}_2$ and additionally decrypting its output under $\mathcal{A}_2$. Neither is enough.

On a high level, in a maliciously circuit-private $\mathcal{A}_2$, any $pk_{\mathcal{A}_2}, a_{sk_M}$ "induce" an encryption of *some* sk_M^* under $\mathcal{A}_2$, so, we may think of them as being well-formed. However, the following potential attack exists. Assume even that pk_M is well-formed, but c_M is arbitrarily malformed. Thus out_M is not guaranteed to be a valid encryption of some x, and may instead carry some other arbitrary (upto $poly(k)$) bits of information about C. In turn, $\mathsf{Dec}_M(c_M, sk_M^*)$ may leak some of this information (even if sk_M^* is the right key corresponding to pk_M).

To fix the above potential attack (and other similar ones) and achieve malicious circuit privacy, we validate (pk, c) of $\mathcal{M}_2$. (In particular, for $\mathcal{M}_2$ resulting from step 1, we need to also check that $a_{sk_{\mathcal{M}_1}}$ encrypts an $sk_{\mathcal{M}_1}$ that corresponds to $pk_{\mathcal{M}_1}$ as part of validating $pk_{\mathcal{M}_2}$ is well-formed.) Such validation is generally hard, so we augment $\mathsf{KeyGen}_{\mathcal{M}_2}$ and $\mathsf{Enc}_{\mathcal{M}_2}$ to supply a helpful witness (encrypted under $\mathcal{A}_2$). For starters we want the full *randomness* used by them. However known $\mathcal{A}_2$ instantiations with malicious circuit privacy against unbounded adversaries, as we require can only evaluate functions in NC^1 and $\mathsf{KeyGen}_{\mathcal{M}_2}, \mathsf{Enc}_{\mathcal{M}_2}$ need not be in NC^1. [7] But surely they are in P, and we can use the standard transform to validate polynomial work in parallel we require the witness to also include values of *all intermediate wires* of $\mathsf{KeyGen}_{\mathcal{M}_2}, \mathsf{Enc}_{\mathcal{M}_2}$, validate all gates in parallel, and have a log-depth AND tree. A similar issue arises already in step 1, where $\mathcal{A}_1$ needs to evaluate the decryption circuit of $\mathcal{M}_1$. The same trick can not by applied there, as the values to decrypt are not known to the receiver. Thus we need to assume $\mathsf{Dec}_{\mathcal{M}_1}$ is in NC^1, which is fortunately satisfied by all known schemes from the literature.

Somewhat more precisely, we transform $\mathcal{M}_2$ in the following non-blackbox way.

- $\mathsf{Enc}(x)$ outputs $c_{\mathcal{M}_2} = \mathsf{Enc}_{\mathcal{M}_2}(r', pk_{\mathcal{M}_2}, x)$ along with $a_{r_{\mathcal{M}_2}} = \mathsf{Enc}_{\mathcal{A}_2}(r)$ - a witness that $c_{\mathcal{M}_2}$ is a proper encryption (derived from r').
- $\mathsf{Eval}(C, c_{\mathcal{M}_2})$: Let $Validate(pk_{\mathcal{M}_2}, c_{\mathcal{M}_2}, out, rk_{\mathcal{M}_2}, r_{\mathcal{M}_2})$ denote a circuit where $rk_{\mathcal{M}_2}, r_{\mathcal{M}_2}$ are purported witnesses for the well-formedness of $pk_{\mathcal{M}_2}, c_{\mathcal{M}_2}$ respectively. It outputs out if $rk_{\mathcal{M}_2}, r_{\mathcal{M}_2}$ certify well-formedness of $pk_{\mathcal{M}_2}, c_{\mathcal{M}_2}$, and the all-zero vector otherwise.
  - Compute $out_{\mathcal{M}_2} = \mathsf{Eval}_{\mathcal{M}_2}(pk_{\mathcal{M}_2}, C, c_{\mathcal{M}_2})$.
  - Output $out = \mathsf{Eval}_{\mathcal{A}_2}(Validate_{pk_{\mathcal{M}_2}, c_{\mathcal{M}_2}, out_{\mathcal{M}_2}}(a_{rk_{\mathcal{M}_2}}, a_{r_{\mathcal{M}_2}}))$ (that is, fixing the suitable variables in $Validate$ to the values at the subscript).
- Dec outputs $\mathsf{Dec}_{\mathcal{M}_2}(sk_{\mathcal{M}_2}, \mathsf{Dec}_{\mathcal{A}_2, out}(sk_{\mathcal{A}_2}))$.

[7] Settling for unbounded simulation against bounded distinguishers, would allow to evaluate arbitrary circuits under the scheme without further complications, however, we are shooting for the strongest possible privacy guarantee.

Note that $pk_{\mathcal{M}_2}, c_{\mathcal{M}_2}, out_{\mathcal{M}_2}$ are hardwired into $Validate$, rather then encrypted via $\mathcal{A}_2$ (in Eval). The subtle reason for this, is that if $pk_{\mathcal{M}_2}$ is malformed, "encrypting" values under it can yield effective encryptions of different values, possibly making us perform the "wrong" validation.

In a nutshell, the construction works by enhanced semi-honest circuit privacy of $\mathcal{M}_2$ if $(pk_{\mathcal{M}_2}, c_{\mathcal{M}_2})$ is well-formed, and the validation procedure takes care of the fact that it is indeed well-formed (otherwise, no information whatsoever is revealed about C).

Merging steps 1 and 2 (Section 3.1). Steps 1+2 provide a clear blueprint for transforming a (compact) FHE $\mathcal{F}$ into maliciously circuit-private FHE by combining it with a maliciously circuit-private HE $\mathcal{P}$ capable of evaluating $\mathcal{F}$'s decryption and validation circuits. That is, the same maliciously circuit-private $\mathcal{P}$ may instantiate both $\mathcal{A}_1$ and $\mathcal{A}_2$. $\mathcal{M}_2$ resulting from step 1 is fed into step 2. In section 3.1 we describe the natural composition of the two steps in a single protocol, making a small shortcut that exploits the structure of $\mathcal{M}_2$ output by step 1, and the fact that $\mathcal{A}_1 = \mathcal{A}_2$. Namely, we do not have to check the well-formedness of $a_{sk_{\mathcal{M}_2}}$ as an encryption under $\mathcal{A}_2$.

Step 3. Let us look more closely at the compactness of the scheme achieved from steps 1+2. The validate circuit that needs to be evaluated has input of size $m = poly(k, n)$ (and polynomial size $|C|$). Thus, if $\mathcal{P}$ is not compact, the encrypted output size is some $poly(|C|, m, k)$ – also $poly(k, n)$.

This is acceptable for our main application of computing on encrypted data, as receiver's input is of size n. However, it would be nice to meet the current standard for FHE where encrypted output size is independent of n.

A more complicated setting is that of leveled FHE. In such schemes, $\mathsf{KeyGen}(1^k, 1^d)$ generates an additional key pk_{Eval} received by Eval (all the rest remains the same), which may grow with the bound d on depth of circuits to be evaluated. In a nutshell, such schemes are often considered in the FHE literature in order to make the underlying assumptions more plausible, avoiding so called (weak) circular security assumptions.

The encrypted output size is $poly(k, n, d)$—quite undesirable!

The idea is to combine the circuit-private (non-compact) HE $\mathcal{M}_3$ resulting from steps 1+2 with a compact FHE $\mathcal{A}_3$ with no circuit privacy "in the opposite direction" from step 1. That is, use $\mathcal{A}_3$ to homomorphically decrypt the output of $\mathcal{M}_3$ to "compress" it. Intuitively, even though the FHE $\mathcal{A}_3$ used for decrypting is not circuit-private, the resulting scheme is, because $\mathsf{Eval}_{\mathcal{A}_3}$ merely acts upon a string that we originally were willing to output "in the plain", so there is no need to protect it.

2 Preliminaries

Notation. We use $\overrightarrow{\text{arrow}}$ to denote vectors, though not always—we tend to use it to stress element-wise handling, e.g. bit-by-bit encryptions. For a function

$f(a,b,c,\ldots)$, $\overrightarrow{f}(\overrightarrow{a},b,\overrightarrow{c},\ldots)$ is a shorthand for $(f(a_1,b,c_1,\ldots),f(a_2,b,c_2,\ldots),\ldots)$. When considering function vectors, all inputs which are the same in all executions appear without an arrow (even if they are vectors by themselves).

For a pair of vectors u,v (u,v) denotes the vector resulting from concatenating u,v. For vectors u,v over some U^t,V^t $(u;v)$ denotes $((u_1,v_1),\ldots,(u_t,v_t))$.

For a function $f(a,b,c,\ldots)$, we denote the set of functions fixing some of its parameters (here b,c) as follows $f|_{b,c}(a,\ldots)$. $f|_{b=B,c=C}$ denotes a function fixing the parameters to particular values B,C respectively.

For randomized algorithms $A(x,r)$, we sometimes write $out \in A(x)$ as a shorthand for $out \in support(A(x,r))$. By $\mathrm{negl}(k)$ we refer to a function that for all polynomials $p(k)$, $\mathrm{negl}(k) < \frac{1}{p(k)}$ for all $k > K$, where K is a constant determined by p. We use the standard notions of statistical and computational indistinguishability of distribution ensembles. Usually an encryption of x under scheme $\mathcal{Y}$ will be named y_x.

Representation Models When we say a HE scheme is $\mathcal{C}$-homomorphic for a class of functions, we actually mean functions having programs C from the set $\mathcal{C}$ of programs. By a program C, we mean a string representing a function $f:\{0,1\}^n \to \{0,1\}$. The correspondence between programs C and the function it represents is determined by an underlying representation model U. A *representation model* $U:\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is an efficient algorithm taking an input (C,x), and returning $f(x)$, where f is the function represented by C. By $|C|$ we simply refer to the length of the string C (as opposed to size(C), which is a related measure depending on U, such as the number of gates in a boolean circuit). For completeness, for circuits and other models we let $U(C,x)=0$ whenever the input (C,x) is syntactically malformed.

As typical in the FHE literature, our default representation model is boolean circuits, unless stated otherwise. We use circuits over some complete set of gates, such as $\{AND, XOR, NOT\}$. Another model we will consider is boolean formulas, which are circuits with fanout 1. We assume the underlying DAGs of circuits are connected. For formulas, we assume wlog. that $depth(C) \leq c \log size(C)$ for a global constant c (that is, that they are "balanced"). For a circuit C, $size(C)$ denotes the number of wires in C's underlying graph, and $depth(C)$ the number of gates on the longest path between an input wire and the output wire of the circuit. By NC^1 we refer to the class of function (families) with uniform formulas of size $poly(n)$.

2.1 Homomorphic Encryption

Throughout the paper k denotes the security parameter taken by HE schemes. A (public-key) homomorphic encryption scheme (HE) $\mathcal{E} = (\mathsf{KeyGen}_E, \mathsf{Enc}_E, \mathsf{Eval}_E, \mathsf{Dec}_E)$ is a quadruple of PPT algorithms as follows.

$\mathsf{KeyGen}(1^k)$**:** Outputs a public key, secret key pair (pk, sk).
$\mathsf{Enc}(pk, b)$**:** Takes a public key and a bit b to encrypt, and returns an encryption c of the bit under pk.

$\mathsf{Eval}(pk, C, c = (c_1, \ldots, c_n))$**:** Takes a public key pk, a bit-by-bit encryption c of a bit vector $x \in \{0,1\}^n$, a function represented by a program C (encoded in a pre-fixed representation model U) and outputs an encryption *out* of bit $U(C, x)$. We assume wlog. that pk includes 1^k (intuitively, this is intended to handle maliciously generated public keys). We refer to outputs of Eval as "encrypted outputs".

$\mathsf{Dec}(sk, out)$**:** Takes a secret key sk, and a purported output *out* of Eval, outputs a bit.

Throughout the paper, HE is semantically secure if $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies standard IND-CPA security for public key encryption schemes as in [GM84]. An HE scheme is *weakly circular-secure* if even knowing a bit-by-bit encryption of the schemes' secret key sk, the adversary still has negligible advantage in the IND-CPA experiment.In this paper, we consider a general notion of homomorphism, under various program classes, rather then just circuits (to express weaker homomorphism properties then FHE).

Definition 2 ($(U,\mathcal{C})$-homomorphic encryption). *Let $\mathcal{C} = \bigcup \mathcal{C}_k$. We say a scheme $\mathcal{E}$ is $(U,\mathcal{C})$-homomorphic if for every $k > 0$ and every program $C \in \mathcal{C}_k$ on inputs $x \in \{0,1\}^n$, the experiment*

$$(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$$
$$out \xleftarrow{\$} \mathsf{Dec}(sk, \mathsf{Eval}(pk, C, \overrightarrow{\mathsf{Enc}}(pk, \overrightarrow{x})))$$

outputs $out = U(C, x)$ with probability 1 for all $x \in \{0,1\}$ and all random choices of the algorithms involved. We say the scheme is k-independently homomorphic if $\mathcal{C}_k = \mathcal{C}$ for all k.[8]

By default our schemes are k-independently homomorphic (in particular the $\mathcal{C}_k$'s are not explicitly defined).

If a scheme satisfies that $\mathcal{C}$ equals the set of all circuits, and is k-independetly homomorphic, we refer to it as "fully homomorphic" (FHE).

Definition 3. *We say a $(U,\mathcal{C})$-homomorphic scheme $\mathcal{E}$ is* compact *if there exists an output bound $B(k, n, |C|) = \mathsf{poly}(k)$ on the output of* Eval *on all $1^k, n$ and $C \in \mathcal{C}_k$ on n bits.*

Another standard variant of HE we consider is *leveled* HE. In this variant, KeyGen is modified to take another parameter 1^d. KeyGen outputs keys (pk, sk), where pk includes a fixed-size part pk_{Enc}, which depends only on k; likewise, sk depends only on k. Enc is modified to accept pk_{Enc} as the public key, and only Eval receives the entire public key pk. In particular, Enc is the same for all d. The notions of compact HE is as for non-leveled schemes ($B(k, n, |C|) =$

[8] For instance, the notion of "somewhat homomorphic" schemes in the FHE literature corresponds to non k-independent schemes, where C_k is a set of circuits with depth bounded by some function of k.

$poly(k)$).For compact schemes, the algorithm Dec is also independent of d. We say such a leveled compact scheme is an FHE, if for all D, the (standard) HE $\mathcal{E}^D$ induced by fixing $d = D$ when calling $\mathsf{KeyGen}(1^d, \cdot)$ induces a k-independently $\mathcal{C}$-homomorphic scheme $\mathcal{E}^D$, where C is the set of all depth-D circuits. The encrypted outputs' size is still $poly(k)$ (for a global polynomial independent of d).

Standard FHE schemes can be thought of as a special case of leveled FHE schemes where KeyGen simply ignores d. Thus, all schemes $\mathcal{E}^d$ are the same (standard) FHE scheme. We refer to this special case as *unleveled* FHE. A HE scheme is maliciously circuit private if every (pk, c) (even arbitrarily malformed) induce some "effective" input x^*.

Definition 4. *Let* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *denote a* $(U, \mathcal{C})$*-homomorphic scheme. We say* $\mathcal{E}$ *is (maliciously)* circuit private *if there exist unbounded algorithms* $\mathsf{Sim}(1^k, pk^*, c^*, b)$*, and deterministic* $\mathsf{Ext}(1^k, pk^*, b)$*, such that for all* k*, and all* $pk^*, c^* = (c_1^*, \ldots, c_n^*)$ *and all programs* $C : \{0,1\}^n \to \{0,1\} \in (U, \mathcal{C})$ *in* $\mathcal{C}_k$ *the following holds:*

- $\overrightarrow{x^*} = \overrightarrow{\mathsf{Ext}}(1^k, pk^*, \overrightarrow{c^*})$.
- $\mathsf{Sim}(1^k, pk^*, c^*, U(C, x^*)) =^s \mathsf{Eval}(1^k, pk^*, C, c^*)$.

In particular, for circuits $C(x_1, \ldots, x_n) \in \mathcal{C}_k$ *the output distribution of* Eval *(including length) depends only on* n, k*. For leveled schemes,* Sim *and* Ext *also take a depth parameter* 1^d*. We say a scheme is semi-honestly circuit-private if the above holds, where* pk^*, c^* *belong to the set of* well-formed *public-key, ciphertext pairs.*

3 Framework

In this section we spell out the construction outlined in the introduction in more detail, including a certain simplification. All security proofs are deferred to the full version. We will need the representation model $(U_{SI}, \mathcal{C})$ (from "split-input") and the weaker notion of "input-privacy" for $\mathcal{P}$. The purpose of introducing this seemingly unnatural representation model and relaxed circuit privacy definition is to allow for simpler implementations of auxiliary HE schemes we use, and overall presentation of our result. The scenario in Theorem 3 is that a function f known to the adversary is to be homomorphically evaluated on a (partially) secret input y, together with the adversary's input x, so hiding f would be an overkill. More specifically, the implementation we use for $\mathcal{P}$ is based on Yao's garbled circuits, which works with a pair of private inputs x, y, and a public circuit C. It outputs $C(x, y)$, but nothing else about x, y. It also leaks quite a lot about C itself (there is no goal of protecting it). While we could use Yao to get circuit privacy for $f|_y$ scheme by evaluating a universal function, there is no need to. The straightforward use of Yao with $C = f$ provides exactly what we need.

Programs in the model are represented by a pair (C_p, C_s), where C_p is a circuit on some m variables, and $C_s \in \{0,1\}^t$ for some $t \leq m$. A program (C_p, C_s) is

interpreted as a function f over $n = m - |C_s|$ variables via $U_{SI}((C_p, C_s), x) = C_p(C_s, x)$. Typically, we will consider $(U_{SI}, \mathcal{C})$-homomorphic schemes where for each (C_p, C_s), all (C_p, Z) for $Z \in \{0,1\}^*$ of length $t \leq m$ are in $\mathcal{C}$. In this case, we specify $\mathcal{C}$ as just a set of circuits.

We say a $(U_{SI}, \mathcal{C})$-homomorphic scheme is *input-private* if it satisfies Definition 4, with the only modification that Sim receives C_p as an input. (This is exactly the guarantee from converting any 2PC protocol where both parties have private input but the function is public to an HE.)

3.1 From Compact FHE to Circuit-Private (Somewhat Compact) FHE

In this section we spell out the combination of steps 1 and 2 as described in the introduction. The schemes $\mathcal{F}, \mathcal{P}$ are a compact FHE, and maliciously circuit private HE respectively. Here $\mathcal{P}$ is for the split input model, and is input-private rather then circuit-private. Although the construction would go through with standard circuit privacy of $\mathcal{P}$, this simplifies the presentation and instantiation of the framework.

Given a leveled FHE $\mathcal{F}$ we define a set of programs $\mathcal{C}_{\mathcal{F}}$ (to be interpreted via U_{SI}) as follows.

1. Let $Dec_{\mathcal{F},k}(sk_{\mathcal{F}}, out_{\mathcal{F}})$ denote the decryption circuit of $\mathcal{F}$ instantiated with security parameter k (recall $\mathsf{Dec}, \mathsf{Enc}$ are independent of d).
2. Let $Validate_{k,d,n}(pk_{\mathcal{F}}, c_{\mathcal{F}}, sk'_{\mathcal{F}}, r_{FE}, out_{\mathcal{F}})$ be the circuit computing

$$Validate_{k,d,n}(\ldots) = \begin{cases} out_{\mathcal{F}} & \text{if } (pk_{\mathcal{F}}, sk_{\mathcal{F}}) \in \mathsf{KeyGen}_{\mathcal{F}}(1^k, 1^d), \text{and} \\ & \forall i \ (c_{\mathcal{F},i} \in \mathsf{Enc}_{\mathcal{F}}(b_i, r_{FE,i}) \text{ for some bit } b_i \in \{0,1\}) \\ \overrightarrow{0} & \text{otherwise.} \end{cases}$$

   where:
   - $(pk_{\mathcal{F}}, sk_{\mathcal{F}})$ is a purported public-key private-key pair output by $\mathsf{KeyGen}_{\mathcal{F}}(1^k, 1^d)$. $sk'_{\mathcal{F}} = (sk_{\mathcal{F}}, r_{FK})$ where r_{FK} is the purported random string used in the generation of $(pk_{\mathcal{F}}, sk_{\mathcal{F}})$, along with all the intermediate outputs of gates in the circuit for KeyGen on input r_{FK}.[9]
   - $c_{\mathcal{F}} = (c_{\mathcal{F},1}, \ldots, c_{\mathcal{F},n})$ is a purported encryption under $pk_{\mathcal{F}}$ of the input bit vector and r_{FE} is a purported vector of randomness used when generating $c_{\mathcal{F}}$, along with intermediate values for the circuit (where $r_{FK,i}$ corresponds to bit x_i).
3. Let $\mathcal{C}_{\mathcal{F}}$ include all pairs of the forms $C = (Validate_{k,d,n}(\ldots), (pk_{\mathcal{F}}, c_{\mathcal{F}}, out_{\mathcal{F}}))$ and $(\mathsf{Dec}_{\mathcal{F},k}(sk_{\mathcal{F}}), out_{\mathcal{F}})$.

Construction 5. *Let $\mathcal{F}$,$\mathcal{P}$ be schemes as above. We construct the following scheme $\mathcal{M}_3$.*

[9] As explained in the intro, the goal of the intermediate values is to put the validation in NC^1, making it implementable by known circuit-private $\mathcal{P}$ with the required notion of privacy.

$\mathsf{KeyGen}_{\mathcal{M}_3}(1^k)$**:** *let* $(pk_{\mathcal{P}}, sk_{\mathcal{P}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathcal{P}}(1^k)$, $(pk_{\mathcal{F}}, sk_{\mathcal{F}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathcal{F}}(1^k, 1^d)$, *and let* $sk'_{\mathcal{F}} = (sk_{\mathcal{F}}, r_{KF})$, *where* r_{KF} *is induced by the randomness used by* $\mathsf{KeyGen}_{\mathcal{F}}$ *as specified in* $Validate$; $\overrightarrow{p_{sk'_{\mathcal{F}}}} = \overrightarrow{\mathsf{Enc}_{\mathcal{P}}}(pk_{\mathcal{P}}, \overrightarrow{sk'_{\mathcal{F}}})$. *Return* $(pk_{\mathcal{M}_3}, sk_{\mathcal{M}_3}) = ((pk_{\mathcal{P}}, pk_{\mathcal{F}}, \overrightarrow{p_{sk'_{\mathcal{F}}}}), (sk_{\mathcal{P}}, sk_{\mathcal{F}}))$. *Here* $pk_{\mathcal{M}_3,\mathsf{Enc}} = (pk_{\mathcal{P}}, pk_{\mathcal{F},\mathsf{Enc}}, p_{sk_{\mathcal{F}}})$.

$\mathsf{Enc}_{\mathcal{M}_3}(pk_{\mathcal{M}_3} = (pk_{\mathcal{P}}, pk_{\mathcal{F},\mathsf{Enc}}, p_{sk_{\mathcal{F}}}), b \in \{0,1\})$**:** *Return* $(c, \overrightarrow{p_{r_{FE}}}) = (\mathsf{Enc}_{\mathcal{F}}(pk_{\mathcal{F}}, b), \overrightarrow{\mathsf{Enc}_{\mathcal{P}}}(pk_{\mathcal{P}}, \overrightarrow{r_{FE}}))$, *where* r_{FE} *is derived from the randomness used by* $\mathsf{Enc}_{\mathcal{F}}$ *as in* $Validate$.

$\mathsf{Eval}_{\mathcal{M}_3}(1^k, pk_P = (pk_{\mathcal{P}}, pk_{\mathcal{F}}, p_{sk'_{\mathcal{F}}}), C, c = (c_{\mathcal{P}}; p_{r_{FE}}))$**:**

1. *If* C *is syntactically malformed, or* $|x|$ *does not match the number of inputs to* C, *replace* C *with the circuit returning* $x_1 \wedge \overline{x_1}$.
2. *Set* $out_{\mathcal{F}} = \mathsf{Eval}_{\mathcal{F}}(pk_{\mathcal{F}}, C, c_{\mathcal{F}})$, $out_{\mathcal{P}} = \mathsf{Eval}_{\mathcal{P}}(pk_{\mathcal{P}}, (\mathsf{Dec}_{\mathcal{F}}, out_{\mathcal{F}}), p_{sk_{\mathcal{F}}})$
3. *Let* $(C_p, C_s) = (Validate_{k,d,n}, (out_{\mathcal{P}}, pk_{\mathcal{F}}, c_{\mathcal{F}}))$
4. *Compute and output* $out = \mathsf{Eval}_{\mathcal{P}}(pk_{\mathcal{P}}, (C_p, C_s), p_{sk'_{\mathcal{F}}}, p_{r_{FE}})$.

$\mathsf{Dec}_{\mathcal{M}_3}(sk_{\mathcal{M}_3}, out_A)$**:** *Output* $y = \mathsf{Dec}_{\mathcal{F}}(sk_{\mathcal{F}}, \overrightarrow{\mathsf{Dec}_{\mathcal{P}}}(sk_{\mathcal{P}}, \overrightarrow{out}))$.

Theorem 3. *Assume a compact leveled FHE scheme* $\mathcal{F} = (\mathsf{KeyGen}_{\mathcal{F}}, \mathsf{Enc}_{\mathcal{F}}, \mathsf{Eval}_{\mathcal{F}}, \mathsf{Dec}_{\mathcal{F}})$ *and* $\mathcal{P}$ *a* $(U_{SI}, \mathcal{C}_{\mathcal{F}})$*-homomorphic, input-private scheme, exist. Consider the resulting scheme* $\mathcal{M}_3$ *as specified in Construction 5 above when instantiating with* $\mathcal{F}, \mathcal{P}$. *Then* $\mathcal{M}_3$ *is a circuit-private FHE. It is unleveled iff* $\mathcal{M}_3$ *is unleveled, and is compact iff* $\mathcal{P}$ *is compact. If* $\mathcal{P}$ *is not compact,* $\mathcal{M}_3$*'s output complexity is* $poly(k, d, n)$ *(*$poly(k, n)$ *if* $\mathcal{F}$ *is unleveled).*

3.2 Compactization of Circuit-Private FHE

When instantiated by the best known constructions from the literature, Theorem 3 only yields $poly(n, k), poly(n, d, k)$ encrypted output complexity for unleveled and leveled $\mathcal{F}$ respectively. This is so, since all circuit-private $\mathcal{C}_{\mathcal{F}}$-homomorphic $\mathcal{P}$ for some FHE $\mathcal{F}$ we know of are not compact.

In this section, we devise a simple transformation (corresponding to Lemma 3 in the introduction) for making a (leveled) scheme's output compact (only $poly(k)$), while preserving circuit privacy. This will yield leveled circuit-private FHE with optimal ($poly(k)$) compactness.

The idea is to use bootstrapping similar to that of step 1 but "in reverse" order. Namely, we take a main scheme $\mathcal{M}_3$ which is circuit-private but not compact, and "decrypt" it under a scheme $\mathcal{A}_3$ which is compact but has no circuit privacy guarantees.

Theorem 4 (Compaction theorem). *Assume a leveled* $\mathcal{C}$*-homomorphic circuit-private scheme* $\mathcal{M}_3$ *and a compact FHE scheme* $\mathcal{A}_3$ *exist.*[10] *Then the scheme* $\mathcal{M}_4$ *in the following construction is a compact* $\mathcal{C}$*-homomorphic circuit-private scheme.*

[10] In fact, $\mathcal{A}_3$ should only be compact and homomorphic for the circuit family it is used for. It does not need to be an FHE.

Construction 6. *Let $\mathcal{M}_3, \mathcal{A}_3$ be HE schemes as in Theorem 4.*

$\mathsf{KeyGen}_{\mathcal{M}_4}(1^k, 1^d)$**:** *Sample* $(pk_{\mathcal{M}_3}, sk_{\mathcal{M}_3}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathcal{M}_3}(1^k, 1^d, r_k)$; *let* $sk'_{\mathcal{M}_3} = (sk_{\mathcal{M}_3}, r_k)$; $(pk_{\mathcal{A}_3}, sk_{\mathcal{A}_3}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathcal{A}_3}(1^k)$; $\overrightarrow{a_{sk'_P}} \xleftarrow{\$} \overrightarrow{\mathsf{Enc}_{\mathcal{A}_3}}(pk_{\mathcal{A}_3}, \overrightarrow{(sk_{\mathcal{M}_3}, r_k)})$. *Output* $(pk, sk) = ((pk_{\mathcal{M}_3}, pk_{\mathcal{A}_3}, a_{sk_{\mathcal{M}_3}}), sk_{\mathcal{A}_3})$. *Here* $pk_{\mathsf{Enc}} = (pk_{\mathcal{M}_3,\mathsf{Enc}}, pk_{\mathcal{A}_3}, a_{sk_{\mathcal{M}_3}})$.

$\mathsf{Enc}_{\mathcal{M}_4}(pk, b)$**:** *Output* $\mathsf{Enc}_{\mathcal{M}_3}(pk_{\mathcal{M}_3,\mathsf{Enc}}, b)$. [11]

$\mathsf{Eval}_{\mathcal{M}_4}(1^k, pk, C, c)$**:**

- $out_{\mathcal{M}_3} \xleftarrow{\$} \mathsf{Eval}_{\mathcal{M}_3}(1^k, pk_{\mathcal{M}_3}, C, c)$.
- *Let* $Dec_{\mathcal{M}_3,k}$ *denote the decryption circuit of* $\mathcal{M}_3$ *with parameter* k. *Then* $Dec_{\mathcal{M}_3,k}|_{out=out_{\mathcal{M}_3}}(sk_{\mathcal{M}_3})$ *is a circuit for decrypting (hard-wired)* $out_{\mathcal{M}_3}$ *under secret keys generated by* $\mathsf{KeyGen}_{\mathcal{M}_3}$.
- *Compute and output* $out_{\mathcal{A}_3} = \mathsf{Eval}_{\mathcal{A}_3}(1^k, pk_{\mathcal{A}_3}, Dec_{\mathcal{M}_3,k}, a_{sk_{\mathcal{M}_3}})$.

$\mathsf{Dec}_{\mathcal{M}_4}(sk = sk_{\mathcal{A}_3}, out)$**:** *Output* $\mathsf{Dec}_{\mathcal{A}_3}(sk_{\mathcal{A}_3}, out)$.

Combining Theorem 3 and Theorem 4, we get:

Theorem 5. *Assume a compact unleveled FHE scheme $\mathcal{F}$ and a $(U_{SI}, \mathcal{C}_{\mathcal{M}})$-homomorphic maliciously input-private scheme $\mathcal{P}$ exist. Then there exists a maliciously circuit-private compact unleveled scheme $\mathcal{M}_4$.*

Getting rid of circular security Theorem 4 still leaves open the question of obtaining compact leveled circuit-private FHE. We show that posing some mild additional efficiency requirements on $\mathcal{M}_3, \mathcal{A}_3$ in Theorem 4, we are able to modify Construction 6 to allow for a leveled *Athree*.

Theorem 6. *Assume schemes $\mathcal{F}, \mathcal{P}$ as Theorem 5 exist. Additionally, assume $\mathsf{Dec}_{\mathcal{P},k}(out, sk)$ has depth $poly(depth(C_p), k)$, where $out = \mathsf{Eval}_{\mathcal{P}}(C_p, C_s)$ for some C_s.*[12] *Assume also that $\{Dec_{\mathcal{F},k}\}$ induced by $\mathcal{F}$ is in NC^1. Then there exists a compact leveled circuit-private scheme $\mathcal{M}_4$.*

3.3 Multi-hop Circuit-Private FHE

In this chapter, we focus on unleveled FHE schemes and show how to upgrade Theorem 5 to yield a multi-hop $\mathcal{M}_4$ (under the same assumptions).

A multi-hop scheme is an FHE scheme, where Eval is modified to support (pk, C, c), where the c_i's are either outputs of Enc or of previous Evals. More formally, if c_i is a purported output of Enc we label it as a level-0 execution of Eval. We recursively define an execution of Eval to be of level l, if the highest level input c_i to it is of level $l-1$. Formally defining (perfectly correct) multi-hop is a natural recursive extension of Definition 2. The base case of level-0 executions is correct decryption for chiphertexts generated by Enc as in Definition 2. We

[11] Here and elsewhere, we do not distinguish between the parts of pk used in Eval and Enc, and refer to both as pk. The distinction is implied by the context.

[12] The circuit for *Dec* can be efficiently computed from *out*.

further require that the evaluation of a level-i execution for every $i > 1$ is correct in the sense that applying Dec to its output recovers the value induced by appropriately combining the circuits involved in the graph of Evals. Similarly, a scheme is i-hop if it satisfies the above correctness requirements only for level-j executions of Eval, where $j \leq i$. Thus standard FHE correspond to 1-hop. The definitions of IND-CPA security and compactness extend for multi-hop schemes in the natural way. The definition of maliciously circuit-private multi-hop FHE is precisely Definition 4.

The construction induced by Theorem 5 is only 1-hop, but not multi-hop. There exists a straightforward transformation from compact FHE schemes into multi-hop schemes (see discussion in Section 1), but it does not necessarily preserve malicious circuit privacy. However, it can be modified to work here. We start from the scheme $\mathcal{M}_4$ resulting from our construction, instantiated with an FHE $\mathcal{F}$, and a maliciously circuit private $\mathcal{P}$, where $\mathcal{M}_1 = \mathcal{A}_3 = \mathcal{F}$, and $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{P}$, reusing keys for the same scheme. Thus, both encrypted inputs and encrypted output of $\mathcal{M}_4$ are encryptions under $\mathcal{F}$ and same key. We can thus include encryptions of the bits of $sk_{\mathcal{F}}$ in pk_{MH}. In subsequent executions of Eval using this one as input, one can first decrypt the out_i's using these key bits, and plug the decrypted out_i's into the original scheme $\mathcal{M}_4$ as the c_i's input to Eval. The main caveat is that in our construction the well-formedness of the c_i's fed to a subsequent instance of Eval needs to be certified (under $\mathcal{A}$) as part of the output of the previous Eval. The key observation is that there is no need to certify the well-formedness of the out_i's, but rather only that of the secret key bits used for decryption![13] Moreover, we only need to prove that $sk_{\mathcal{F}}$ constitute valid encryptions of some bits under $pk_{\mathcal{F}}$ specified in $pk_{\mathcal{M}_4}$ (not even correspondence to the $pk_{\mathcal{F}}$ published as part of pk!). This is ok because $sk_{\mathcal{F}}$ is short and independent of the circuit being evaluated. As these are decrypted under the specified key bits in subsequent Eval's, the result would be an enryption of some value independent of the (subsequent) C, which is what we need (as in $\mathcal{M}_4$, if validation fails, nothing is learned about C).

We defer an explicit description and full analysis of our multi-hop construction, as well as implications to MPC and comparison to [GHV10] to the full version.

4 Instantiations of the Framework

We devise instantiations of schemes $\mathcal{F}, \mathcal{P}$ as required in Theorem 6. As these requirements are strictly stronger then the requirements in Theorem 5, they immediately yield an instantiation of Theorem 5 as well. The component $\mathcal{F}$ has many instantiations from the literature.

[13] If we had to certify them, seemingly, we would need to give a proof on the validity of the execution of Eval, referring to its inputs. It is not clear how to make it short and protect the privacy of that Eval's circuit.

For $\mathcal{P}$ we use the following instantiation, induced by the specific construction combining an information theoretic variant of Yao's garbled circuits [IK02] with maliciously circuit-private OT-homomorphic schemes.

Lemma 4. *Assume the existence of circuit-private schemes which are homomorphic for (bit) OT. In particular, the DDH,QR,Paillier or the DCRA assumptions yield such OT schemes [AIR01, HK12]. Then there exists a circuit-private $(U_{SI}, \mathcal{C})$-homomorphic scheme $\mathcal{P}$ where $\mathcal{C}$ consists of all (balanced) formulas. Furthermore, $\mathcal{P}$ has decryption circuits $\mathsf{Dec}_{\mathcal{P},k}(sk, out)$ of depth $\mathrm{depth}(C_p)poly(k)$, where $out = \mathsf{Eval}_{\mathcal{P}}(C_p, C_s)$.*

The following corollary of Theorem 6 (5) and Lemma 4 is our working, "take-home", instantiation of the framework.

Corollary 1. *Assume a leveled FHE $\mathcal{F}$ with decryption circuits in NC^1 exists. Assume further that there exist (bit) OT-homomorphic circuit-private HE. Then there exists a circuit-private compact FHE $\mathcal{M}_4$. $\mathcal{M}_4$ is unleveled if $\mathcal{M}$ is.*

As mentioned above, $\mathcal{M}$ has many "efficient enough" instantiations. See the full version for some examples and proofs of Lemma 4 and Corollary 1 (almost immediate).

5 Future Work

The "work horse" of our bootstrapping-based transformation 1 for transforming FHE into circuit-private is a circuit-private bit-OT-homomorphic HE. The known constructions from the literature we are aware of can not base circuit-private OT on some assumption the implies FHE (such as LWE, approximate GCD etc.). We do not see good reasons, except for historical ones to why this is the case. Such a construction would give an example of compact FHE which can be made circuit-private without additional assumptions.

References

[AIR01] Boneh, D., Kushilevitz, E., Ostrovsky, R., Skeith III, W.E.: Public key encryption that allows PIR queries. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 50–67. Springer, Heidelberg (2007)

[BKOI07] Boneh, D., Kushilevitz, E., Ostrovsky, R., Skeith III, W.E.: Public Key Encryption That Allows PIR Queries. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 50–67. Springer, Heidelberg (2007)

[BLV04] Barak, B., Lindell, Y., Vadhan, S.P.: Lower bounds for non-black-box zero knowledge. In: Electronic Colloquium on Computational Complexity (ECCC), vol. (83) (2004)

[Bra12] Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012)

[BV11] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) lwe. Cryptology ePrint Archive, Report 2011/344 (2011), http://eprint.iacr.org/2011/344

[Can01] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145. IEEE Computer Society (2001)

[DFH12] Damgård, I., Faust, S., Hazay, C.: Secure Two-Party Computation with Low Communication. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 54–74. Springer, Heidelberg (2012)

[DJ01] Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)

[Gen09] Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009), http://crypto.stanford.edu/craig

[GHV10] Gentry, C., Halevi, S., Vaikuntanathan, V.: i-Hop Homomorphic Encryption and Rerandomizable Yao Circuits. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 155–172. Springer, Heidelberg (2010)

[GIKM98] Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. In: Vitter, J.S. (ed.) STOC, pp. 151–160. ACM (1998)

[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)

[GSW13] Gentry, C., Sahai, A., Waters, B.: Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013)

[HK12] Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptology 25(1), 158–193 (2012)

[IK02] Ishai, Y., Kushilevitz, E.: Perfect constant-round secure computation via perfect randomizing polynomials (2002)

[IP07] Ishai, Y., Paskin, A.: Evaluating Branching Programs on Encrypted Data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007), Full version in, http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2012/PHD/PHD-2012-16

[Lip05] Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 314–328. Springer, Heidelberg (2005)

[NP01] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Rao Kosaraju, S. (ed.) SODA, pp. 448–457. ACM/SIAM (2001)

[vDGHV09] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616 (2009), http://eprint.iacr.org/2009/616