# Amplifying the Security of Functional Encryption, Unconditionally

Aayush Jain(✉), Alexis Korb(✉), Nathan Manohar(✉), and Amit Sahai(✉)

UCLA, Los Angeles, CA, USA
{aayushjain,alexiskorb,nmanohar,sahai}@cs.ucla.edu

**Abstract.** Security amplification is a fundamental problem in cryptography. In this work, we study security amplification for functional encryption (FE). We show two main results:

- For any constant $\epsilon \in (0,1)$, we can amplify any FE scheme for $\mathsf{P/poly}$ which is $\epsilon$-secure against all polynomial sized adversaries to a fully secure FE scheme for $\mathsf{P/poly}$, unconditionally.
- For any constant $\epsilon \in (0,1)$, we can amplify any FE scheme for $\mathsf{P/poly}$ which is $\epsilon$-secure against subexponential sized adversaries to a fully subexponentially secure FE scheme for $\mathsf{P/poly}$, unconditionally.

Furthermore, both of our amplification results preserve compactness of the underlying FE scheme. Previously, amplification results for FE were only known assuming subexponentially secure LWE.

Along the way, we introduce a new form of homomorphic secret sharing called set homomorphic secret sharing that may be of independent interest. Additionally, we introduce a new technique, which allows one to argue security amplification of nested primitives, and prove a general theorem that can be used to analyze the security amplification of parallel repetitions.

## 1 Introduction

Security amplification is a fundamental problem in which one takes a weakly secure cryptographic primitive and transforms it into a fully secure primitive. For instance, suppose $(G, E, D)$ is a public-key encryption (PKE) scheme satisfying standard correctness, but which only satisfies the weak security guarantee that there exists a constant $\epsilon \in (0,1)$ such that for all messages $m_0, m_1 \in \{0,1\}^\lambda$ and for all polynomial-time adversaries $\mathcal{A}$, we have

$$|\Pr[\mathcal{A}(\mathsf{pk}, E(\mathsf{pk}, m_0)) = 1 \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow G(1^\lambda)] - \Pr[\mathcal{A}(\mathsf{pk}, E(\mathsf{pk}, m_1)) = 1 \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow G(1^\lambda)]| \leq \epsilon.$$

Then, the relevant security amplification goal for such an $\epsilon$-secure public-key encryption would be to construct a new PKE $(G', E', D')$ that satisfies standard security, where the constant $\epsilon$ above would be replaced with a negligible function in $\lambda$. It has long been known [22,37] that the security of $\epsilon$-secure PKE can be amplified to achieve fully secure PKE unconditionally. (Remarkably, however,


D. Micciancio and T. Ristenpart (Eds.): CRYPTO 2020, LNCS 12170, pp. 717–746, 2020.
https://doi.org/10.1007/978-3-030-56784-2_24

there are still natural questions about security amplification for $\epsilon$-secure PKE that remain open – see below.)

Aside from being a fundamental question in its own right, security amplification also opens the door to building cryptographic primitives from new intractability assumptions. For instance, in the future, we may discover natural sources of hardness that yield cryptographic primitives with only a weak level of security. Using security amplification, such novel sources of hardness would still yield fully secure cryptographic primitives. This motivation is especially important for cryptographic primitives for which only a few assumptions are known to yield that primitive.

There have been numerous works throughout the years on security amplification for various cryptographic primitives (for example, [5,8,13,19–21,31–33,35–38,41,47,50–52,54,59,62,63]). As with all cryptographic primitives, minimizing assumptions is a major goal in security amplification research. Indeed, unlike many results in cryptography, security amplification results can be *unconditional* (e.g. [13,19,21,33,35–38,47,51,52,54,59,62,63]).

*Security Amplification for Functional Encryption.* The focus of this paper is to study security amplification in the context of functional encryption. Functional encryption (FE), introduced by [56] and first formalized by [18,53], is one of the core primitives in the area of computing on encrypted data. This notion allows an authority to generate and distribute keys associated with functions $f_1, \ldots, f_q$, called *functional keys*, which can be used to learn the values $f_1(x), \ldots, f_q(x)$ given an encryption of $x$. Intuitively, the security notion states that the functional keys associated with $f_1, \ldots, f_q$ and an encryption of $x$ reveal nothing beyond the values $f_1(x), \ldots, f_q(x)$.

Functional encryption has been the subject of intense study [1,3,5,8–10,16,17,24–29,39,45,46,48,49,55,56] and has opened the floodgates to important cryptographic applications that have long remained elusive. These applications include, but are not limited to, multi-party non-interactive key exchange [27], universal samplers [27], reusable garbled circuits [28], verifiable random functions [11,14,30], and adaptive garbling [34]. FE has also helped improve our understanding of important theoretical questions, such as the hardness of Nash equilibrium [26,27]. One of the most important applications of FE is its implication to indistinguishability obfuscation (iO for short) [9,16]. There have also been several recent works on functional encryption combiners [2,7,40] and the related problem of iO combiners [6,23]. While amplifiers allow one to transform a weakly secure candidate into a fully secure one, combiners allow one to take many candidates of which at least one is fully secure (and the others are potentially completely insecure) and transform them into a fully secure scheme.

*Our Results.* Remarkably, although functional encryption was introduced 15 years ago in [56], security amplification for $\epsilon$-secure FE, defined analogously to $\epsilon$-secure PKE above, was first studied only recently in [5,8], which achieved amplification assuming *subexponentially secure LWE*. In fact, no security amplification results for FE are known under any other assumptions. In this paper,

we show that one can obtain amplification for FE *unconditionally*. In particular, we obtain the following:

**Theorem 1 (Informal).** *Assuming an $\epsilon$-secure FE scheme for $\mathsf{P/poly}$ secure against all polynomial sized adversaries for some constant $\epsilon \in (0,1)$, there exists a fully secure FE scheme secure against all polynomial sized adversaries. Furthermore, the transformation preserves compactness.*

Additionally, our amplification result can be generalized to hold against larger adversaries, in particular, adversaries of subexponential size.

**Theorem 2 (Informal).** *Assuming an $\epsilon$-secure FE scheme for $\mathsf{P/poly}$ secure against subexponential sized adversaries for some constant $\epsilon \in (0,1)$, there exists a subexponentially secure FE scheme. Furthermore, the transformation preserves compactness.*

As a consequence of the above theorem and the FE to iO transformations of [9,15,16,42,44], we observe that we can construct iO from an $\epsilon$-secure FE scheme secure against subexponential sized adversaries without the need for any additional assumptions.

*Techniques and additional results.* To achieve our results, we introduce and construct a new form of homomorphic secret sharing called set homomorphic secret sharing (SetHSS), informally defined below in our Technical Overview. This generalizes a recent notion of combiner friendly homomorphic secret sharing introduced in [40] to a probabilistic scenario tailored for security amplification.

Our work also involves an intertwined use of hardcore measures [12,38,43,51,61] and efficient leakage simulation [20,41,57,58,60]. First, we improve upon and simplify a technique introduced in [5,8] and then used in [31] that allows one to argue that some fraction of many parallel repetitions of a weakly secure primitive are likely to be secure. The original technique critically uses the leakage simulation theorems [20,41] in conjunction with a hardcore measure theorem [51], which allows one to escape the computational overhead of sampling from hardcore measures. We simplify their technique by using a different leakage simulation theorem [57] which allows for more direct simulation of the applicable leakage. Moreover, we introduce a new "fine-grained" analysis that is crucial to achieving the parameters we need for unconditional amplification. Finally, we isolate the core of their technique and derive a general and applicable theorem (which we call the probabilistic replacement theorem). This theorem is not specific to any cryptographic primitive and, thus, we believe that it might be useful for future efforts in cryptographic amplification beyond FE.

Our second technique is a new technique which allows one to argue security amplification of nested encryptions. In particular, using this technique, we are able to prove the following:

**Theorem 3 (Informal).** *For any constant $\epsilon \in (0,1)$ and $\epsilon$-secure FE scheme $\mathsf{FE}$, the FE scheme $\mathsf{FE}^*$ obtained by composing $\mathsf{FE}$ with itself is $\epsilon^2 + \mathsf{negl}(\lambda)$ -secure.*

We remark that this technique can also be generalized to argue similar security for public-key encryption (PKE). As such, we also show the following:

**Theorem 4 (Informal).** *For any constant $\epsilon \in (0,1)$ and $\epsilon$-secure PKE scheme* $\mathsf{PKE}$*, the PKE scheme* $\mathsf{PKE}^*$ *obtained by composing* $\mathsf{PKE}$ *with itself is* $\epsilon^2 + \mathsf{negl}(\lambda)$ *-secure.*

Prior to our paper, to the best of our knowledge, it was not known how to prove that a simple nesting provided this amplification even for public-key encryption.

Lastly, we remark that this amplification by nesting technique also critically relies on a combination of leakage simulation and hardcore measures. We believe our results exemplify how potent this combination can be for security amplification of cryptographic primitives.

## 2 Technical Overview

To establish our results, we proceed in two phases:

1. First, we construct an amplifier that converts an $\epsilon$-secure FE scheme for any constant $\epsilon \in (0,1)$ to an $\epsilon'$-secure FE scheme for any arbitrarily small constant $\epsilon' < \epsilon$.
2. Second, we construct an amplifier that converts an $\epsilon$-secure FE scheme for any sufficiently small constant $\epsilon < \frac{1}{6}$ to a fully secure FE scheme.

The above template also works to give an amplifier that is subexponentially secure (Theorem 2). By composing the amplifiers of these two stages, we arrive at our results. We will begin by focusing on the second stage of our amplification procedure, namely, how we amplify an FE scheme that is $\epsilon$-secure for a constant $\epsilon < \frac{1}{6}$ to one that is fully secure.

### 2.1 Amplification via Secret Sharing and Parallel Repetition

Typically, in order to amplify a weakly secure primitive to a fully secure one, one proceeds by constructing a scheme that uses many copies of the weakly secure primitive and is secure if a fraction of these copies are secure. Intuitively, we expect that if these copies of the weakly secure primitive are independent, then at least some fraction should be secure, and the resulting construction will also be secure. This idea of parallel repetitions of the weakly secure primitive is utilized typically in tandem with a secret sharing scheme. For example, the canonical public-key encryption amplifier works by secret sharing the message and then encrypting each of these shares independently in parallel using the weakly-secure public-key encryption scheme [47]. This paradigm has also been used to amplify other primitives such as non-interactive zero-knowledge [31], by constructing a suitable secret sharing scheme.

In order to amplify functional encryption (FE), a natural approach to utilize this framework is via function secret sharing (FSS). Function secret sharing allows one to split a function $f$ into shares $f_1, \ldots, f_n$ such that for any input

$x$, we can also split $x$ into shares $x_1, \ldots, x_n$ such that learning the evaluations $f_1(x_1), \ldots, f_n(x_n)$ allows one to recover $f(x)$. Informally, the security property associated with a function secret sharing scheme is that given all but one of the input shares, the input should remain hidden (beyond what is revealed by $f(x)$) even if one is given all the function shares and their evaluations on the input shares. If we had such a function secret sharing scheme, we could simply encrypt each input share $x_i$ under an instantiation $\mathsf{FE}_i$ of our weakly secure FE scheme to obtain $\mathsf{ct}_i$. A ciphertext in our scheme would be $(\mathsf{ct}_i)_{i\in[n]}$. Similarly, key generation could use $\mathsf{FE}_i$ to generate a key $\mathsf{sk}_i$ for the function $f_i$. The function key in our scheme would then be $(\mathsf{sk}_i)_{i\in[n]}$. From these ciphertexts and function keys, one could learn $(f_i(x_i))_{i\in[n]}$ and recover $f(x)$. For security, one would expect that if the FE scheme is weakly secure, then at least one out of the $n$ instantiations would be secure, in which case, the overall scheme's security would follow by the security of the function secret sharing scheme. This general approach was used in [5,8] to amplify FE assuming subexponentially secure LWE.

In this work, our goal is to amplify FE *unconditionally*. We first observe that we can assume secure one-way functions and still achieve unconditional amplification since a weakly-secure FE implies a weakly-secure one-way function, which can subsequently be amplified using the result of [38]. Unfortunately, we do not know how to construct function secret sharing schemes of the above form assuming only secure one-way functions. However, we note that the above function secret sharing scheme allows up to $n-1$ of the shares to be corrupted while maintaining security. Yet, if we take many copies of an $\epsilon$-secure FE scheme, we would expect roughly a $(1-\epsilon)$ fraction of copies to be secure, not just one! Thus, the above function secret sharing scheme has a stronger security property than the one we would intuitively expect to require for amplification. All we actually need is a secret sharing scheme that is secure against *typical* corruption patterns (that is, one that is secure with high probability if each share is corrupted independently with some probability $p$). To capitalize on this intuition, we introduce and construct a new type of homomorphic secret sharing scheme, called a *set homomorphic secret sharing scheme*.

*Set Homomorphic Secret Sharing Scheme.* In a set homomorphic secret sharing (SetHSS) scheme, function shares are associated with sets $(T_i)_{i\in[m]}$, where each set $T_i \subset \{1, 2, \ldots, n\}$. The input $x$ is split into $n$ shares $x_1, \ldots, x_n$. A function $f_i$ associated with the set $T_i$ takes as input all $x_j$'s such that $j \in T_i$. Thus, we can think of the $T_i$'s as sets of the indices of the input shares that the function takes as input. The security guarantee is that if the adversary corrupts some of the $T_i$'s and learns all the input shares corresponding to these sets, security still holds provided there is at least one input share $x_{i^*}$ that the adversary does not learn.

Using a SetHSS scheme, it is possible to build (what we expect to be) an FE amplifier. We follow the same approach detailed above for a function secret sharing scheme to build FE, except we instead use SetHSS with respect to sets $(T_i)_{i\in[m]}$. That is, we run $m$ copies of the FE setup algorithm to obtain $m$

master secret keys $(\mathsf{msk}_i)_{i\in[m]}$. To encrypt a message $x$, we $n$-out-of-$n$ secret share $x$ into shares $x_1, \ldots, x_n$. For each $i \in [m]$, we encrypt $(x_j)_{j\in T_i}$ under $\mathsf{msk}_i$ to obtain $\mathsf{ct}_i$ and set the ciphertext $\mathsf{ct}$ as $(\mathsf{ct}_i)_{i\in[m]}$. To generate function keys, we use the SetHSS scheme to obtain function shares $f_1, \ldots, f_m$ and then set $\mathsf{sk}_f = (\mathsf{sk}_i)_{i\in[m]}$, where $\mathsf{sk}_i$ is the function key for $f_i$ generated using $\mathsf{msk}_i$. Observe that by the correctness of the SetHSS scheme and the FE scheme, the above is a correct FE construction. Since the FE scheme is only weakly-secure, if we assume that each encryption becomes corrupted with some probability $p$ (this corresponds to a set $T_i$ becoming corrupted in the SetHSS scheme), we can calculate the probability that the SetHSS scheme remains secure when the corresponding input shares are leaked.

The question that naturally follows is how do we construct such a SetHSS scheme? The first step towards this was taken in the recent work of [40], which introduced a specialized form of function secret sharing, called *combiner-friendly homomorphic secret sharing* (CFHSS), which was constructed assuming only one-way functions. Essentially, a CFHSS is a SetHSS where $m = \binom{n}{3}$, and the sets $T_i$ are all possible size 3 subsets of $\{1, 2, \ldots, n\}$. We observe that unfortunately, such a SetHSS scheme will not suffice for our purposes, because if any constant fraction of the sets $T_i$ are corrupted, then almost certainly every input share $x_j$ would be corrupted.

Instead, for some parameters $n$ and $m$, we generate sets $(T_i)_{i\in[m]}$ by including each element in $[n]$ in each $T_i$ independently at random with some probability $q$. We can then calculate two probabilities: First, we can ensure that the probability that at least one share $x_j$ is not corrupted, is sufficiently high – this should intuitively guarantee security. Second, we can ensure that all sets of size 3 are covered by at least one of the sets $T_i$ – this will allow us to ensure correctness by setting the function share $f_i$ in our SetHSS scheme to be the concatenation of the CFHSS function shares corresponding to each size 3 subset contained in $T_i$.

It turns out that setting the parameters $n, m$, and $q$ above to achieve both properties simultaneously is nontrivial, and, in fact, we iterate this process twice. The first SetHSS scheme lets us amplify from $\epsilon < \frac{1}{6}$ security to $1/\mathsf{poly}(\lambda)$ security. The second SetHSS scheme lets us amplify from $1/\mathsf{poly}(\lambda)$ security to negligible (or sub-exponential) security.

However, our security calculations only give us a sense of what we expect the resulting security level to be. How do we actually prove that the scheme attains this level of security?

### 2.2 Proving Security: Probabilistic Replacement Theorem

Consider the following situation: There are $n \in \mathbb{N}$ independent copies of some primitive that is known to be only weakly secure (over the randomness of the primitive) for some notion of security. Then, one wants to claim that if $n$ is large enough, with high probability, at least one of these $n$ instantiations will be secure. Or as a stronger notion, one might want some fraction of the $n$ instantiations to be secure. This is useful when security of some larger primitive holds provided

that some fraction of these $n$ instantiations are secure. For example, if one were to additively secret share a message and then independently encrypt each share, the message remains hidden as long as at least one of the encryptions cannot be broken.

*Proofs Using Hardcore Lemmas:* Typical proofs of this sort rely on hardcore lemmas that define hardcore measures. First, we review the notion of a hardcore measure. Suppose that a primitive is secure with some low probability over its randomness. Then, Impagliazzo's hardcore lemma [38] states that there exists some "hard core" of the primitive's randomness such that the primitive is secure with high probability (against a somewhat smaller class of adversaries) when its randomness is restricted to this "hard core". In other words, though the primitive may be weakly secure over uniform randomness, there is some "hard core" portion of the randomness on which the primitive is strongly secure. This "hard core" may be defined as a measure over the randomness (which we call a hardcore measure) or as a subset of the randomness (which we call a hardcore set). A more precise specification of the relationship between the security gain and the density of the hardcore measure can be found in various hardcore lemmas (refer to Sect. 3).

Then, typical security amplification proofs proceed as follows: In the scenario above, each of the $n$ instances of the primitive independently samples its randomness from a uniform distribution. However, this is equivalent to having each primitive sample its randomness from its hardcore measure with probability proportional to the density of the hardcore measure and sample from the complement of the hardcore measure with probability proportional to the density of the complement. When considered this way, if the density of the hardcore measure is large enough, with high probability, some of the instances of the primitive will sample randomness from their hardcore measures. Therefore, those primitives are secure by the definition of the hardcore measure.

*Dealing with the Time Complexity of Sampling Hardcore Measures:* Now, this proof technique works whenever it is the final step in a larger proof of security. But what happens when this is not the case? For instance, suppose we independently encrypt secret shares of a message $m$, and then after claiming some fraction of the encryptions are secure, suppose we want to move to an experiment where the secure shares are replaced with shares corresponding to the message 0. A natural idea would be to replace the shares known to be secure (those where the randomness of the encryption was sampled from the hardcore measures) with simulated shares via a reduction to some notion of indistinguishability between the real and simulated shares when the real shares are hidden.

We note that the reduction in this case, upon receiving either the simulated or real shares, would need to encrypt these challenge shares using the secure encryption instances. This means the reduction needs to sample randomness from the hardcore measures of the encryption. This can be problematic because there is no bound on the efficiency of sampling from these hardcore measures. Therefore, there is no bound on the efficiency of the reduction. This would be fine

if the secret sharing satisfied a *statistical* notion of security. Unfortunately, this will not work if the underlying secret sharing scheme achieves only computational security, such as is the case with our SetHSS scheme. In general, the same issue can occur whenever computational assumptions need to be used in the remainder of the proof of security, after applying an appropriate hardcore lemma.

In essence, the issue is that once one uses the fact that one is sampling from the hardcore measures to prove that an instance is secure, then later reductions may also have to sample from the hardcore measures. But this sampling may not be efficient, so the reduction may also be inefficient. To address this problem, we build upon a technique introduced in [5,8]. We first observe that hardcore measures of sufficiently high density also have high min-entropy. Then, we use a leakage simulation theorem from [57] which allows one to simulate sampling from measures with high min-entropy in a manner that is more efficient; by careful choice of parameters, we show that this simulation can be made efficient enough to allow us to perform cryptographic reductions. This allows one to continue performing reductions even after one has invoked the hardcore measures (instead of sampling from the hardcore measure, we can instead run the simulator for the measure). Furthermore, we can ensure that the simulator is independent of some of its inputs through the appropriate use of commitments. We note that instead of using [57] for leakage simulation, [5,8] uses a different leakage simulation lemma [20] that deals with low output length leakage instead of high min-entropy leakage and, therefore, requires the leakage to be first transformed into an appropriate form. Our proof is thus simpler and more direct. Additionally, by considering the output of the simulator as a single joint distribution, we can also get slightly better and more fine-grained parameters, which allows us to get polynomial time simulators for all of the appropriate parameter regimes we use in this paper. We then present the core of this technique in a more abstract and modular way so that it can be applied to other situations and proofs. We note that our abstracted theorem does not refer to hardcore measures at all, but instead refers to the more natural problem of claiming that some fraction of $n$ primitives is secure.

*The Probabilistic Replacement Theorem:* More specifically, suppose there are two randomized functions $E$ and $F$ that are weakly indistinguishable over their randomness. Then, our theorem shows indistinguishability between the following two experiments: In one experiment, the adversary gets $n$ independent evaluations of $E$ on $n$ inputs. In the other experiment, we probabilistically replace some of the instances of $E$ with $F$. Then, we give the adversary evaluations of these instances of $E$ and $F$ using randomness generated by some bounded-time function $h$. Essentially, we show that one can replace some of the instances of $E$ with instances of $F$, while still maintaining overall efficiency. Please refer to Sect. 7 for more details.

Relating this back to the notion of security, we could let $F$ be a "secure" variant of some primitive $E$. For instance, $F$ could be an encryption of 0 and $E$ an encryption of the message $m$. If $E$ is weakly secure in the sense that $E$ is weakly indistinguishable from $F$, then if one has enough independent instances

of $E$, we show that at least some fraction of them will be secure (in the sense that one can replace these instances of $E$ with the secure variant $F$). For more details, please refer to the proof overview in Sect. 7.

*Applying the Probabilistic Replacement Theorem:* Having shown the probabilistic replacement theorem (Sect. 7), it is now possible to prove the security of our FE amplifier described above fairly easily. Roughly, we will use the probabilistic replacement theorem to replace FE encryptions of SetHSS shares with simulated FE encryptions. Once this has been done, we can use the security of the underlying SetHSS scheme to argue security of our FE amplifier.

*Setting the Parameters:* By appropriately setting the parameters $n$ (number of input shares), $m$ (number of sets in the SetHSS scheme), and $q$ (the probability of an element in $[n]$ being included in any set), we are able to show that our construction indeed amplifies security. We will have to apply the construction twice. First, we are able to amplify from a constant $\epsilon < \frac{1}{6}$ secure FE scheme to one that is $1/\mathsf{poly}(\lambda)$ secure. Then, we are able to amplify a $1/\mathsf{poly}(\lambda)$ scheme to one that is fully secure. An astute reader may have noticed that at each invocation of our amplifier, we also lose some correctness. However, in between applications of our amplifier, we can easily amplify correctness by parallel repetition. This is because we only need one of our repetitions to be correct. This approach does lose a factor of security proportional to the number of repetitions, but the parameters can be set so that overall we gain in security while preserving correctness. Please refer to Sect. 8 for more details.

### 2.3 Amplifying Security via Nesting

The above FE amplifier was already sufficient to amplify an $\epsilon$-secure FE scheme with $\epsilon < \frac{1}{6}$ to a fully secure one. However, we would like to be able to amplify an $\epsilon$-secure FE scheme for any constant $\epsilon \in (0, 1)$. Here, we show how to amplify an $\epsilon$-secure FE scheme for any $\epsilon \in (0, 1)$ to an $\epsilon'$-secure FE scheme for any $\epsilon' \in (0, 1)$. To do this, we first show how to amplify an $\epsilon$-secure FE scheme to a (roughly) $\epsilon^2$-secure one. By repeatedly applying this transformation a constant number of times, we can amplify to any smaller constant. The construction itself is to simply nest two independent copies of the underlying $\epsilon$-secure FE scheme. Namely, first encrypt the message under $\mathsf{FE}_1$ to compute $\mathsf{ct}_1$ and then encrypt $\mathsf{ct}_1$ under $\mathsf{FE}_2$ to obtain the final ciphertext $\mathsf{ct}$, with appropriate functional secret keys. Intuitively, since there are two layers of encryption, where each layer is secure with probability $(1 - \epsilon)$, we would expect the double encryption to be secure with probability $(1 - \epsilon^2)$. However, proving this requires some care. Indeed, to the best of our knowledge, such a security amplification result, even for nested public-key encryption, was not previously known.

*Proof Overview:* As noted above, we expect our nested scheme to be secure if one of the encryption layers is secure. Now, if we could prove that each layer is *independently* insecure with probability at most $\epsilon$, then we could show that the

amplified $\mathsf{FE}^*$ scheme is only insecure with probability at most $\epsilon^2$. Unfortunately, the security of the two layers is not independent; in general the hard core sets of randomness which lead to secure encryptions could depend on the message being encrypted. Instead, we will achieve similar amplification by in some sense "simulating" the security of the outer $\mathsf{FE}$ in a way that is independent of the security of the inner $\mathsf{FE}$.

First, we quantify the security of the outer $\mathsf{FE}$ using hardcore measures. If we have an $\epsilon$-secure $\mathsf{FE}$, then for any fixed output of the inner $\mathsf{FE}$, the outer $\mathsf{FE}$ is secure with probability at least $1 - \epsilon$. Therefore, by [51], there exist hardcore measures (of density $1 - \epsilon$) of the randomness of the outer $\mathsf{FE}$ such that the outer $\mathsf{FE}$ is strongly secure when its randomness is sampled from these hardcore measures. So, with probability at least $1 - \epsilon$, we sample randomness from the hardcore measures of the outer $\mathsf{FE}$ and achieve security via these hardcore measures. But with probability $\epsilon$, we have no guarantee that the outer $\mathsf{FE}$ is secure, so we must rely on the security of the inner $\mathsf{FE}$.

Now, we want to show that conditioned on the outer $\mathsf{FE}$ being potentially insecure (i.e. when we do not sample from these hardcore measures), then the inner $\mathsf{FE}$ is still only insecure with probability close to $\epsilon$. In other words, we want to show that the security of the inner and outer $\mathsf{FE}$ schemes are close to independent. To do so, we need to perform a reduction to the $\epsilon$-security of the inner $\mathsf{FE}$. At this point, we run into two issues. First, in order to perform our reduction to the security of the inner $\mathsf{FE}$, we will need to sample from the complement hardcore measures of the outer $\mathsf{FE}$. (Recall that we first conditioned on the outer $\mathsf{FE}$ being potentially insecure.) However, this is problematic because we have no bound on the efficiency of computing or sampling from these hardcore measures. Secondly, the hardcore measures of the outer $\mathsf{FE}$ depend implicitly on the randomness used by the inner $\mathsf{FE}$. Or, in other words, the security of the outer $\mathsf{FE}$, as quantified by these measures, is not independent of the security of the inner $\mathsf{FE}$.

To resolve these issues, we need to find a way to give an efficient reduction to the security of the inner $\mathsf{FE}$, despite the inefficiencies and dependencies outlined above. Intuitively, we proceed as follows: Our reduction takes as input the ciphertext produced by the inner $\mathsf{FE}$. The reduction then uses the fact that the complement of the hard core measure of the outer $\mathsf{FE}$ has density $\epsilon$ to efficiently simulate randomness that is indistinguishable from hardcore randomness; this simulation uses the leakage simulation theorem of [57]. This allows our reduction to create the outer $\mathsf{FE}$ ciphertext that the adversary expects. Please refer to Sect. 9 for more details.

### 2.4 Organization

In Sect. 3, we recall necessary preliminaries. In Sect. 4, we define functional encryption notions with partial security. In Sects. 5 and 6, we define and instantiate set homomorphic secret sharing schemes and analyze their correctness and security when the underlying sets are sampled in a probabilistic manner. In Sect. 7, we state and prove the Probabilistic Replacement Theorem. In Sect. 8,

we show our parallel repetition amplification theorem. In Sect. 9, we show our nesting amplification theorem. In Sect. 10, we show that nesting amplifies the security of public-key encryption. Finally, in Sect. 11, we combine our nesting and parallel repetition amplification results.

## 3 Preliminaries

**Notation.** Let $\lambda \in \mathbb{N}$ be the security parameter. Throughout, we define various size and advantage parameters as functions of $\lambda$. We say that a function $f(\lambda)$ is negligible, denoted $f(\lambda) = \mathsf{negl}(\lambda)$, if $f(\lambda) = \lambda^{-\omega(1)}$. We say that a function $f(\lambda)$ is polynomial, denoted $f(\lambda) = \mathsf{poly}(\lambda)$, if $f(\lambda) = p(\lambda)$ for some fixed polynomial $p$. Throughout, when we write inequalities in terms of functions of $\lambda$, we mean that these inequalities hold for sufficiently large $\lambda$. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \ldots, n\}$. For a set $S$, let $x \leftarrow S$ denote the process of sampling $x$ from the uniform distribution over $S$. For a distribution $\mathcal{D}$, let $x \leftarrow \mathcal{D}$ denote the process of sampling $x$ from $\mathcal{D}$.

**Definition 1 ($(s, \epsilon)$-Indistinguishability).** *We say that two ensembles $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$ are $(s, \epsilon)$-indistinguishable if for any adversary $\mathcal{A}$ of size $s$,*

$$\left| \Pr_{x \leftarrow \mathcal{X}_\lambda}[\mathcal{A}(1^\lambda, x)] - \Pr_{y \leftarrow \mathcal{Y}_\lambda}[\mathcal{A}(1^\lambda, y)] \right| \leq \epsilon$$

*for sufficiently large $\lambda \in \mathbb{N}$.*

**Notation.** We will say that ensembles satisfy $(\mathsf{poly}(\lambda) \cdot s, \epsilon)$-indistinguishability if the ensembles satisfy $(p(\lambda) \cdot s, \epsilon)$-indistinguishability for every polynomial $p(\lambda)$.

We will make use of the following Chernoff bound in our analysis.

**Definition 2 (Chernoff Bound).** *Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed Boolean random variables. Let $X = \sum_{i \in [n]} X_i$ and let $\mu = \mathbb{E}[X]$. Then, for $\delta \geq 1$,*

$$\Pr[X \geq (1 + \delta)\mu] \leq e^{-\frac{\delta\mu}{3}}.$$

We define a measure.

**Definition 3.** *A measure is a function $\mathcal{M} : \{0,1\}^k \to [0,1]$.*

- *The size of a measure is $|\mathcal{M}| = \sum_{x \in \{0,1\}^k} \mathcal{M}(x)$.*
- *The density of a measure is $\mu(\mathcal{M}) = |\mathcal{M}|2^{-k}$.*
- *The distribution defined by a measure (denoted by $\mathcal{D}_\mathcal{M}$) is a distribution over $\{0,1\}^k$, where for every $x \in \{0,1\}^k$, $\Pr_{X \leftarrow \mathcal{D}_\mathcal{M}}[X = x] = \mathcal{M}(x)/|\mathcal{M}|$.*
- *A scaled version of a measure for a constant $0 < c < 1$ is $\mathcal{M}_c = c\mathcal{M}$. Note that $\mathcal{M}_c$ induces the same distribution as $\mathcal{M}$.*
- *The complement of a measure is $\overline{\mathcal{M}} = 1 - \mathcal{M}$.*

### 3.1 Useful Lemmas

We defer this section to the full version.

## 4 Functional Encryption

We define the notion of a (secret key) functional encryption scheme.

*Syntax of a Functional Encryption Scheme.* A functional encryption (FE) scheme $\mathsf{FE}$ for a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ consists of four polynomial time algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ defined as follows. Let $\mathcal{X}_\lambda$ be the input space of the circuit class $\mathcal{C}_\lambda$, and let $\mathcal{Y}_\lambda$ be the output space of $\mathcal{C}_\lambda$. We refer to $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ as the input and output space of the scheme, respectively.

- **Setup,** $\mathsf{msk} \leftarrow \mathsf{FE.Setup}(1^\lambda)$: It takes as input the security parameter $\lambda$ and outputs the master secret key $\mathsf{msk}$.
- **Encryption,** $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, m)$: It takes as input the master secret key $\mathsf{msk}$ and a message $m \in \mathcal{X}_\lambda$ and outputs $\mathsf{ct}$, an encryption of $m$.
- **Key Generation,** $\mathsf{sk}_C \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, C)$: It takes as input the master secret key $\mathsf{msk}$ and a circuit $C \in \mathcal{C}_\lambda$ and outputs a function key $\mathsf{sk}_C$.
- **Decryption,** $y \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct})$: It takes as input a function secret key $\mathsf{sk}_C$, a ciphertext $\mathsf{ct}$ and outputs a value $y \in \mathcal{Y}_\lambda$.

We can similarly define the notion of a public key FE scheme, and our results in this work also hold for public key FE. However, we choose to focus on secret key FE, as this is a weaker primitive.

We describe the properties associated with an FE scheme.

*Correctness.*

**Definition 4 (Approximate Correctness).** *A functional encryption scheme* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be* $\mu$*-correct if it satisfies the following property: for every* $C : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda \in \mathcal{C}_\lambda, m \in \mathcal{X}_\lambda$ *it holds that:*

$$\Pr\left[\begin{array}{c} \mathsf{msk} \leftarrow \mathsf{FE.Setup}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, m) \\ \mathsf{sk}_C \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, C) \\ C(m) \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct}) \end{array}\right] \geq \mu,$$

*where the probability is taken over the coins of the algorithms.*

*We refer to FE schemes that satisfy the above definition of correctness with* $\mu = 1 - \mathsf{negl}(\lambda)$ *for a negligible function* $\mathsf{negl}(\cdot)$ *as correct.*

*Efficiency: Sublinearity and Compactness.*

**Definition 5 (Sublinearity and Compactness).** *A functional encryption scheme* $\mathsf{FE}$ *for a circuit class* $\mathcal{C}$ *containing circuits of size at most* $s$ *that take inputs of length* $\ell$ *is said to be sublinear if there exists some constant* $\epsilon > 0$ *such that the size of the encryption circuit is bounded by* $s^{1-\epsilon} \cdot \mathsf{poly}(\lambda, \ell)$ *for some fixed polynomial* $\mathsf{poly}$*. If the above holds for* $\epsilon = 1$*, then the FE scheme is said to be* compact*.*

In this work, we will focus on FE schemes that are sublinear (and possibly compact).

*Security.* We recall indistinguishability-based super-selective security for FE. This security notion is modeled as a game between a challenger $\mathsf{Chal}$ and an adversary $\mathcal{A}$. The game begins with $\mathcal{A}$ submitting message queries $(x_i)_{i\in[\Gamma]}$, a challenge message query $(x_0^*, x_1^*)$, and a function query $C$. $\mathsf{Chal}$ samples a bit $b$ and responds with ciphertexts corresponding to $(x_i)_{i\in[\Gamma]}$ and $x_b^*$ along with a function key $\mathsf{sk}_C$ corresponding to $C$. $\mathcal{A}$ wins the game if she can guess $b$ with probability significantly more than $1/2$ and if $C(x_0^*) = C(x_1^*)$. That is to say, the function evaluation computable by $\mathcal{A}$ on the challenge ciphertext gives the same value regardless of $b$. We can define our security notion in terms of the size $s = s(\lambda)$ of adversaries against which security holds and an advantage $\epsilon = \epsilon(\lambda)$ that such adversaries can achieve. We say such a scheme is $(s, \epsilon)-$secure.

**Definition 6 ($(s,\epsilon)$-secure FE).** *A secret-key FE scheme* $\mathsf{FE}$ *for a class of circuits* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *and message space* $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ *is* $(s, \epsilon)$*-secure if for any adversary* $\mathcal{A}$ *of size* $s$*, the advantage of* $\mathcal{A}$ *is*

$$\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}} = \left|\mathsf{Pr}[\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \mathsf{Pr}[\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 1) = 1]\right| \leq \epsilon,$$

*where for each* $b \in \{0, 1\}$ *and* $\lambda \in \mathbb{N}$*, the experiment* $\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, b)$ *is defined below:*

1. ***Challenge queries**: $\mathcal{A}$ submits message queries $(x_i)_{i\in[\Gamma]}$, a challenge message query $(x_0^*, x_1^*)$, and a function query $C$ to the challenger $\mathsf{Chal}$, with $x_i \in \mathcal{X}_\lambda$ for all $i \in [\Gamma]$, $x_0^*, x_1^* \in \mathcal{X}_\lambda$, and $C \in \mathcal{C}_\lambda$ such that $C(x_0^*) = C(x_1^*)$. Here, $\Gamma$ is an arbitrary (a priori unbounded) polynomial in $\lambda$.*
2. *$\mathsf{Chal}$ computes $\mathsf{msk} \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and then computes $\mathsf{ct}_i \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, x_i)$ for all $i \in [\Gamma]$. It then computes $\mathsf{ct}^* \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, x_b^*)$ and $\mathsf{sk}_C \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, C)$. It sends $((\mathsf{ct}_i)_{i\in[\Gamma]}, \mathsf{ct}^*, \mathsf{sk}_C)$ to $\mathcal{A}$.*
3. *The output of the experiment is set to $b'$, where $b'$ is the output of $\mathcal{A}$.*

*Adaptive Security and Collusions.* The above security notion is referred to as super-selective security in the literature. One can consider a stronger notion of security, called *adaptive security with unbounded collusions*, where the adversary can make an unbounded (polynomial) number of function secret key queries and can interleave the challenge messages and the function queries in any arbitrary

order. In this paper, we only deal with super-selectively secure FE schemes. However, it holds for any fully-secure sublinear FE scheme that these notions are equivalent [4,42], and therefore, we only focus on super-selective security in this work, as it is a simpler starting place.

### 4.1 Semi-functional FE

In this work, to simplify some constructions and proofs, we will consider the notion of semi-functional FE (sFE). Semi-functional FE is simply a functional encryption scheme with two auxiliary algorithms. We defer the definition to the full version.

## 5 Set Homomorphic Secret Sharing Schemes

In [40], as an intermediate step in their construction of an FE combiner, they define and construct what they call a combiner-friendly homomorphic secret sharing scheme (CFHSS). We defer the definition to the full version. [40] show the following.

**Theorem 5 ([40]).** *Assuming one-way functions, there exists a combiner-friendly homomorphic secret sharing scheme for* $\mathsf{P/poly}$ *for* $n = O(\mathsf{poly}(\lambda))$ *candidates.*

Moreover, [40] also show the following extension of the above theorem, when the underlying OWF is $(O(s), O(s^{-1}))$-secure for $s = \omega(\mathsf{poly}(\lambda))$.

**Theorem 6 ([40]).** *Assuming an* $(O(s), O(s^{-1}))$*-secure one-way function, there exists an* $(O(s), \mathsf{poly}(\lambda) \cdot O(s^{-1}))$*-secure combiner-friendly homomorphic secret sharing scheme for* $\mathsf{P/poly}$ *for* $n = O(\mathsf{poly}(\lambda))$ *candidates. Moreover, the size of* $\mathsf{InpEncode}$ *is independent of the size of the circuit class and the size of any* $C_{i,j,k}$ *is bounded by* $|C| \cdot \mathsf{poly}(\lambda, n)$ *for some fixed polynomial.*

In this work, we extend the notion of a combiner-friendly homomorphic secret sharing scheme [40] to a more general setting, which will be useful for amplification. The CFHSS scheme of [40] implicitly restricts the shares to correspond to all subsets $T \subseteq [n]$ with $|T| = 3$. This is clear by simply noting that we can think of the share $s_{i,j,k}$ as corresponding to the set $T = \{i, j, k\}$ (the construction in [40] does not care about the ordering of $i, j, k$, so there are only $\binom{n}{3}$ shares in their construction, not $n^3$). For amplification, we will need to use a more general approach, where we allow the sets to be arbitrary and given as input to the scheme.

**Definition 7.** *A set homomorphic secret sharing scheme,* $\mathsf{SetHSS} = (\mathsf{InpEncode}, \mathsf{FuncEncode}, \mathsf{Decode})$*, for* $n \in \mathbb{N}$ *candidates,* $m \in \mathbb{N}$ *sets* $\{T_i\}_{i \in [m]}$*, where each set* $T_i \subseteq [n]$*, and a class of circuits* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ *with input space* $\mathcal{X}_\lambda$ *and output space* $\mathcal{Y}_\lambda$ *consists of the following polynomial time algorithms:*

- **Input Encoding,** $\mathsf{InpEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, x)$*: It takes as input the security parameter* $\lambda$*, the number of candidates* $n$*, a collection of* $m$ *sets* $\{T_i\}_{i\in[m]}$*, where each set* $T_i \subseteq [n]$*, and an input* $x \in \mathcal{X}_\lambda$ *and outputs a set of input shares* $\{s_i\}_{i\in[m]}$*.*
- **Function Encoding,** $\mathsf{FuncEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, C)$*: It takes as input the security parameter* $\lambda$*, the number of candidates* $n$*, a collection of* $m$ *sets* $\{T_i\}_{i\in[m]}$*, where each set* $T_i \subseteq [n]$*, and a circuit* $C \in \mathcal{C}$ *and outputs a set of function shares* $\{C_i\}_{i\in[m]}$*.*
- **Decoding,** $\mathsf{Decode}(\{C_i(s_i)\}_{i\in[m]}, \{T_i\}_{i\in[m]})$*: It takes as input a set of evaluations of function shares on their respective input shares and* $m$ *sets and outputs a value* $y \in \mathcal{Y}_\lambda \cup \{\perp\}$*.*

*A set homomorphic secret sharing scheme,* SetHSS*, for sets* $\{T_i\}_{i\in[m]}$ *has the following properties:*

- **Correctness***: For every* $\lambda \in \mathbb{N}$*, circuit* $C \in \mathcal{C}_\lambda$*, and input* $x \in \mathcal{X}_\lambda$*, it holds that:*

$$\Pr\left[\begin{array}{c}\{s_i\}_{i\in[m]} \leftarrow \mathsf{InpEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, x)\\ \{C_i\}_{i\in[m]} \leftarrow \mathsf{FuncEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, C)\\ C(x) \leftarrow \mathsf{Decode}(\{C_i(s_i)\}_{i\in[m]}, \{T_i\}_{i\in[m]})\end{array}\right] \geq 1 - \mathsf{negl}(\lambda),$$

  *where the probability is taken over the coins of the algorithms and* $\mathsf{negl}(\lambda)$ *is a negligible function in* $\lambda$*.*
- **Security***:*

**Definition 8** ***(IND-secure*** SetHSS***).*** *A set homomorphic secret sharing scheme* SetHSS *for a class of circuits* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in[\mathbb{N}]}$ *with input space* $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in[\mathbb{N}]}$ *and sets* $\{T_i\}_{i\in[m]}$ *is selectively secure if for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\cdot)$ *such that for all sufficiently large* $\lambda \in \mathbb{N}$*, the advantage of* $\mathcal{A}$ *is*

$$\mathsf{Adv}^{\mathsf{SetHSS}}_{\mathcal{A}} = \left|\mathsf{Pr}[\mathsf{Expt}^{\mathsf{SetHSS}}_{\mathcal{A}}(1^\lambda, 1^n, 0) = 1] - \mathsf{Pr}[\mathsf{Expt}^{\mathsf{SetHSS}}_{\mathcal{A}}(1^\lambda, 1^n, 1) = 1]\right| \leq \mu(\lambda),$$

*where for each* $b \in \{0,1\}$ *and* $\lambda \in \mathbb{N}$ *and* $n \in \mathbb{N}$*, the experiment* $\mathsf{Expt}^{\mathsf{SetHSS}}_{\mathcal{A}}(1^\lambda, 1^n, b)$ *is defined below:*

$\mathsf{Expt}^{\mathsf{SetHSS}}_{\mathcal{A}}(1^\lambda, 1^n, b)$

1. ***Secure share****:* $\mathcal{A}$ *submits an index* $i^* \in [n]$ *that it will not learn the input shares for.*
2. ***Challenge input queries****:* $\mathcal{A}$ *submits input queries,*

$$\left(x_0^\ell, x_1^\ell\right)_{\ell\in[L]}$$

   *with* $x_0^\ell, x_1^\ell \in \mathcal{X}_\lambda$ *to the challenger* Chal*, where* $L = \mathsf{poly}(\lambda)$ *is chosen by* $\mathcal{A}$*.*
3. *For all* $\ell$*,* Chal *computes* $\{s_i^\ell\}_{i\in[m]} \leftarrow \mathsf{InpEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, x_b^\ell)$*. For all* $\ell$*, the challenger* Chal *then sends* $\{s_i^\ell\}_{i\in[m], i^*\notin T_i}$*, the input shares that do not correspond to a set containing* $i^*$*, to the adversary* $\mathcal{A}$*.*

4. ***Function queries***: *The following is repeated an at most polynomial number of times:* $\mathcal{A}$ *submits a function query* $C \in \mathcal{C}_\lambda$ *to* $\mathsf{Chal}$. *The challenger* $\mathsf{Chal}$ *computes function shares* $\{C_i\}_{i\in[m]} \leftarrow \mathsf{FuncEncode}(1^\lambda, 1^n, \{T_i\}_{i\in[m]}, C)$ *and sends them to* $\mathcal{A}$ *along with all evaluations* $\{C_i(s_i^\ell)\}_{i\in[m]}$ *for all* $\ell \in [L]$.
5. *If there exists a function query* $C$ *and challenge message queries* $(x_0^\ell, x_1^\ell)$ *such that* $C(x_0^\ell) \neq C(x_1^\ell)$, *then the output of the experiment is set to* $\perp$. *Otherwise, the output of the experiment is set to* $b'$, *where* $b'$ *is the output of* $\mathcal{A}$.

We refer to a $\mathsf{SetHSS}$ scheme that satisfies the correctness and security properties as a correct and secure $\mathsf{SetHSS}$ scheme, respectively.

### 5.1 SetHSS from CFHSS

Given the $\mathsf{CFHSS}$ scheme from [40], we can construct a correct $\mathsf{SetHSS}$ scheme for sets $T_1, T_2, \ldots, T_m$ provided that $\{T_i\}_{i\in[m]}$ covers all subsets of size 3 (formally defined in Definition 9). Looking ahead, our $\mathsf{SetHSS}$ scheme will remain secure if the corruption pattern on the $T_i$'s is such that some element $j \in [n]$ is not in any corrupted set. This is exactly the unmarked element condition in Sect. 6.

Formally, we show the following.

**Theorem 7.** *Assuming one-way functions, there exists a set homomorphic secret sharing scheme for* $\mathsf{P/poly}$ *for* $n = O(\mathsf{poly}(\lambda))$ *candidates for sets* $T_1, T_2, \ldots, T_m$ *that cover all subsets of size* 3. *Moreover, security holds regardless of the sets* $T_1, T_2, \ldots, T_m$.

We simultaneously also show the following for $s = \omega(\mathsf{poly}(\lambda))$.

**Theorem 8.** *Assuming an* $(O(s), O(s^{-1}))$*-secure one-way function, there exists an* $(O(s), \mathsf{poly}(\lambda) \cdot O(s^{-1}))$*-secure set homomorphic secret sharing scheme for* $\mathsf{P/poly}$ *for* $n = O(\mathsf{poly}(\lambda))$ *candidates for sets* $T_1, T_2, \ldots, T_m$ *that cover all subsets of size* 3. *Security holds regardless of the sets* $T_1, T_2, \ldots, T_m$. *Moreover, the size of the circuit* $\mathsf{InpEncode}(\cdot)$ *is independent of the size of the circuit class and the size of any function encoding* $C_i$ *has size bounded by* $|C| \cdot \mathsf{poly}(\lambda, n, m)$ *for some fixed polynomial.*

We defer the proofs of these theorems to the full version.

## 6 Covering Sets

In this section, we will define some properties of covering sets that will be useful in our FE construction. Informally, covering sets are a collection of sets $(X_i)$ such that some other collection of sets $(Y_j)$ are covered by the $X_i$'s. By this, we mean that every $Y_j$ is a subset of some $X_i$. As discussed previously, our overall plan for constructing an amplified FE is to use a set homomorphic secret sharing scheme, which will allow us to secret share the message into $n$ shares

and then encrypt $m$ sets, each which contains some of the $n$ shares. Thus, we can think of the $X_i$'s as subsets of $[n]$. However, we only know how to construct such set homomorphic secret sharing schemes if the sets cover all subsets of size 3. Furthermore, these set homomorphic secret sharing schemes have a specific security property defined in Sect. 5. In this section, we analyze the probability that randomly sampled sets will cover all size $t$ subsets and the probability that the security property is satisfied when the sets are randomly corrupted. These probabilities will be instrumental in analyzing the correctness and security properties of our amplified FE construction in Sect. 8.1.

**Definition 9 (Set $t$-Covering).** *We say that a collection of sets $T_1, T_2, \ldots, T_m$ over $[n]$ covers all subsets of size $t$ if for every $T' \subseteq [n]$ with $|T'| = t$, there exists some $i \in [m]$ such that $T' \subseteq T_i$.*

**Definition 10 (Unmarked Element).** *Let $f : [m] \to \{0,1\}$ be a marking function, where we say an index $i \in [n]$ is "marked" if $f(i) = 1$ and "unmarked" if $f(i) = 0$. A collection of sets $T_1, T_2, \ldots, T_m$ over $[n]$ has an unmarked element with respect to $f$ if there exists an index $i \in [n]$ such that for all sets $T_j$ with $i \in T_j$, $f(j) = 0$.*

**Lemma 1.** *Consider sampling $m$ sets $T_1, T_2, \ldots, T_m$, where each set is chosen by independently including each element in $[n]$ with probability $q$. Then, with probability $\geq 1 - n^t(1-q^t)^m$, $T_1, T_2, \ldots, T_m$ is a $t-$covering.*

*Proof.* Let $S_1, \ldots, S_{\binom{n}{t}}$ be all subsets of $[n]$ of size $t$. For any $i \in [\binom{n}{t}]$ and $j \in [m]$, then

$$\Pr[S_i \not\subseteq T_j] = (1 - q^t).$$

Therefore,

$$\Pr[\forall j \in [m], S_i \not\subseteq T_j] = (1 - q^t)^m.$$

By the union bound,

$$\Pr\left[\exists i \in \left[\binom{n}{t}\right], \forall j \in [m], S_i \not\subseteq T_j\right] \leq n^t(1-q^t)^m,$$

giving the desired result.

**Lemma 2.** *Consider sampling $m$ sets $T_1, T_2, \ldots, T_m$, where each set is chosen by independently including each element in $[n]$ with probability $q$. Define the marking function $f : [m] \to \{0,1\}$ by setting, independently at random for each $i \in [m]$, $f(i) = 1$ with probability $p$. Then, for any $\delta \geq 1$, with probability at least $(1 - e^{-\frac{\delta pm}{3}})(1 - (1 - (1-q)^{(1+\delta)pm})^n)$, the sets have an unmarked element with respect to $f$.*

*Proof.* Let $S \subseteq [m]$. Define $B_S$ to be the event that $\forall u \in S$, $f(u) = 1$, and $\forall v \notin S$, $f(v) = 0$. Since any distinct $i, j \in [n]$ are independently included in each set, observe that for any $S \subseteq [m]$, the event that $i$ is unmarked given $B_S$ is independent of the event that $j$ is unmarked given $B_S$. Therefore, since $i$ is

included in each marked set (a set $T_u$ with $f(u) = 1$) with probability $1 - q$, then

$$\begin{aligned}\Pr\ [i \text{ unmarked } \mid B_S] &= (1-q)^{|S|} \\ \Pr[\forall i \in [n], i \text{ marked } \mid B_S] &= (1-(1-q)^{|S|})^n \\ \Pr[\exists i \in [n], i \text{ unmarked } \mid B_S] &= 1-(1-(1-q)^{|S|})^n.\end{aligned}$$

Then,

$$\begin{aligned}\Pr\ [\exists i \in [n], i \text{ unmarked}] &= \sum_{S_j \subseteq [m]} \Pr[B_{S_j}](1-(1-(1-q)^{|S_j|})^n) \\ &= \sum_{k=0}^{n} \sum_{S_j, |S_j|=k} \Pr\left[B_{S_j}\right](1-(1-(1-q)^k)^n) \\ &= \sum_{k=0}^{n} \Pr[k \text{ sets are marked}](1-(1-(1-q)^k)^n) \\ &\geq \Pr[\text{at most } k \text{ sets are marked}](1-(1-(1-q)^k)^n).\end{aligned}$$

for every $k \in [n]$. Let $X_i$ be the event that set $T_i$ is marked (in other words, $f(i) = 1$). Let $X = \sum_{i \in [m]} X_i$. Note that $\mathbb{E}[X] = pm$. Then, by the Chernoff bound (Definition 2) for any $\delta \geq 1$,

$$\Pr[X \geq (1+\delta)pm] \leq e^{-\frac{\delta pm}{3}}.$$

Therefore,

$$\Pr[\exists i \in [n], i \text{ unmarked}] \geq (1-e^{-\frac{\delta pm}{3}})(1-(1-(1-q)^{(1+\delta)pm})^n).$$

## 7 Probabilistic Replacement Theorem

Please refer to the technical overview (Sect. 2.2) for the high level overview and motivation of this theorem as well as an introduction to hardcore measures.

*Our Theorem:* Suppose there are two randomized functions $E$ and $F$ that are weakly indistinguishable over their randomness and the randomness of the distinguisher. Then, our theorem below shows indistinguishability between the following two experiments: In one experiment, the adversary gets $n$ independent evaluations of $E$ on $n$ inputs. In the other experiment, we probabilistically replace some of the instances of $E$ with $F$. Then, we give the adversary evaluations of these instances of $E$ and $F$ using randomness generated by some bounded time function $h$. Essentially, we show that one can replace some of the instances of $E$ with instances of $F$ while still maintaining overall efficiency.

We also include some other details. First, we need to determine which inputs to evaluate $E$ and $F$ on. As such, we define $\mathsf{Gen}$ to be any randomized circuit that

produces these inputs, and evaluate $E$ and $F$ on the output of $\mathsf{Gen}$. Second, we also allow for the adversary to receive additionally auxiliary input, which can also be output by $\mathsf{Gen}$. Lastly, we allow some control over which inputs of $E$ and $F$ the bounded time function $h$ will depend upon. We can achieve this by modifying our first experiment to also output a commitment $Z$ of the inputs we wish to remain hidden. Then, the simulator $h$ produced in the second experiment will only depend on some of the hidden values, namely the values needed to compute the instances of $E$ and $F$ that are actually output. (In contrast, $h$ could have been dependent upon on all of the potential inputs of both $E$ and $F$ in every instance.)

Finally, we note that our introduction of a commitment into the theorem is not a significant problem when using this theorem to prove the security of some game that did not originally contain commitments. Rather than proving directly that an adversary cannot break a security game, one can instead prove a stronger notion of security in which the adversary is unable to break the security game even when additionally given a commitment of some secret information. Since, an adversary can only have a smaller advantage in differentiating these experiments when this commitment is not given (an adversary that can break security without the commitment can break security with the commitment by ignoring the commitment), regular security trivially follows. In fact, we use this exact technique in our FE amplification. Note that if the adversary is not strong enough to break the commitment, then giving them a commitment of the secret information will not significantly impact security.

*Remark 1.* We wrote our theorem in a very general form in order to facilitate potential reuse in other research. As such, the security parameters in the theorem statement are quite complex. However, we have also included three corollaries that use much simpler and more natural parameters. We refer the reader to these corollaries rather than the actual theorem when fine-grained tuning of the parameters is not necessary.

**Theorem 9 (Probabilistic Replacement Theorem).** *Let $\lambda$ be a parameter. Let $E : \mathcal{S} \times \mathcal{X} \times \{0,1\}^\ell \to \mathcal{W}$ and $F : \mathcal{T} \times \mathcal{Y} \times \{0,1\}^\ell \to \mathcal{W}$ be deterministic $O(\mathsf{poly}(\lambda))$-time computable functions, with $\ell = O(\mathsf{poly}(\lambda))$. Let $n = O(\mathsf{poly}(\lambda))$. Then, if*

- $\mathsf{Com}$ *is any commitment with $(\mathsf{size}_{\mathsf{HIDE}}, \mathsf{adv}_{\mathsf{HIDE}})$-computational hiding and $(\mathsf{stat}_{\mathsf{BIND}})$-statistical binding,*
- $\mathsf{Gen}$ *is any randomized circuit of size $O(\mathsf{poly}(\lambda))$ with range $(\mathcal{S} \times \mathcal{X} \times \mathcal{T} \times \mathcal{Y})^n \times \mathsf{AUX}$ such that for all $((s_i, x_i, t_i, y_i)_{i\in[n]}, \mathsf{aux})$ output by $\mathsf{Gen}(1^\lambda, 1^n)$ for all $i \in [n]$ and for all $\mathsf{size}_{EF}$ algorithms $\mathcal{A}$,*

$$\left| \Pr_{r_i \leftarrow \{0,1\}^\ell}[\mathcal{A}(E(s_i, x_i, r_i)) = 1] - \Pr_{r_i \leftarrow \{0,1\}^\ell}[\mathcal{A}(F(t_i, y_i, r_i)) = 1] \right| \leq \mathsf{adv}_{EF},$$

*there exists a randomized function $h$ of size $\mathsf{size}_h$ such that for all algorithms $\mathcal{A}'$ of size $\mathsf{size}^*$,*

$$|\Pr[\mathcal{A}'(\mathsf{EXP}_0) = 1] - \Pr[\mathcal{A}'(\mathsf{EXP}_1) = 1]| \leq \mathsf{adv}^*,$$

*where we define*

$\mathsf{EXP}_0$*:*

1. *Compute* $((s_i, x_i, t_i, y_i)_{i\in[n]}, \mathsf{aux}) \leftarrow \mathsf{Gen}(1^\lambda, 1^n)$.
2. *Compute* $Z \leftarrow \mathsf{Com}((s_i, t_i)_{i\in[n]})$.
3. *Sample* $r_i$ *from* $\{0,1\}^\ell$ *for* $i \in [n]$.
4. *Compute* $w_i = E(s_i, x_i, r_i)$ *for* $i \in [n]$.
5. *Output* $(Z, (w_i)_{i\in[n]}, \mathsf{aux})$.

$\mathsf{EXP}_1$*:*

1. *Compute* $((s_i, x_i, t_i, y_i)_{i\in[n]}, \mathsf{aux}) \leftarrow \mathsf{Gen}(1^\lambda, 1^n)$.
2. *Compute* $Z \leftarrow \mathsf{Com}(0^{\ell_Z})$ *where* $\ell_Z = |(s_i, t_i)_{i\in[n]}|$.
3. *Sample a string* $\alpha \in \{0,1\}^n$ *such that for each* $i \in [n]$*, we set* $\alpha_i = 1$ *with probability* $(1 - \mathsf{adv}_{EF})$ *and set* $\alpha_i = 0$ *with probability* $\mathsf{adv}_{EF}$.
4. *Compute* $(r_i)_{i\in[n]} \leftarrow h(\alpha, Z, (s_i)_{i\in A_0}, (t_i)_{i\in A_1}, (x_i, y_i)_{i\in[n]}, \mathsf{aux})$ *where* $A_0 = \{i \mid \alpha_i = 0\}$ *and* $A_1 = \{i \mid \alpha_i = 1\}$.
5. *For every* $i \in [n]$*, if* $\alpha_i = 1$*, compute* $w_i = F(t_i, y_i, r_i)$*; otherwise, compute* $w_i = E(s_i, x_i, r_i)$.
6. *Output* $(Z, (w_i)_{i\in[n]}, \mathsf{aux})$.

*and for any parameters* $\mathsf{size}_{\mathsf{SIM}} > 0$ *and* $\mathsf{adv}_{\mathsf{SIM}}, \mathsf{adv}_{\mathsf{HCM}} \in (0,1)$ *and for* $\mathsf{adv}_{min} = \min(\mathsf{adv}_{EF}, 1 - \mathsf{adv}_{EF})$,

- $\mathsf{size}_h = O(\mathsf{poly}(\lambda) \cdot \mathsf{size}_{\mathsf{SIM}} 2^{2n \log(\mathsf{adv}_{min}^{-1})} \mathsf{adv}_{\mathsf{SIM}}^{-5})$.
- $\mathsf{size}^*$ *is the minimum of the following:*
  - $\frac{\mathsf{size}_{EF} \mathsf{adv}_{\mathsf{HCM}}^2}{128(2\ell+1)} - \mathsf{poly}(\lambda)$
  - $\mathsf{size}_{\mathsf{SIM}} - \mathsf{poly}(\lambda)$
  - $\mathsf{size}_{\mathsf{HIDE}} - \mathsf{size}_h - \mathsf{poly}(\lambda)$
- $\mathsf{adv}^* \leq n \cdot \mathsf{adv}_{\mathsf{HCM}} + \mathsf{stat}_{\mathsf{BIND}} + \mathsf{adv}_{\mathsf{SIM}} + \mathsf{adv}_{\mathsf{HIDE}}$.

Theorem 9 immediately gives rise to two corollaries: one where we assume that $E$ and $F$ are weakly indistinguishable against polynomial sized adversaries, and one where they are weakly indistinguishable against subexponential sized adversaries. These corollaries are deferred to the full version.

Furthermore, using a more fine-grained approach, it is possible to prove a variant of the probabilistic replacement theorem that allows us to lower the size of $h$ at the cost of increasing the distinguishing advantage of the adversary. We will need to use this fine-grained approach when proving security against polynomial time adversaries. We state the resulting corollary here and provide a proof after the proof of the main theorem at the end of this section. We defer this corollary to the full version.

We defer the proof of this theorem and the corollaries to the full version.

# 8 Amplification via Secret Sharing and Parallel Repetition

In this section, we prove our main amplification results. As discussed previously, this is done by building an FE scheme using our set homomorphic secret sharing scheme $\mathsf{SetHSS}$. In our construction, we encrypt each share in our set homomorphic secret sharing scheme under an instantiation of a weakly secure FE scheme. (To simplify the proof, we will actually use a weakly secure semi-functional FE scheme, which can be built from a weakly secure FE scheme assuming OWFs). For key generation, we first generate function encodings corresponding to each share using $\mathsf{SetHSS.FuncEncode}$ and then generate function keys for each of these function encodings using the appropriate weakly secure FE instantiation. Recall from Sect. 5 that $\mathsf{SetHSS}$ is parameterized by $n$ "elements" and $m$ sets $(T_i)_{i\in[m]}$ that are subsets of $[n]$. We will let $n$ and $m$ be parameters of our FE construction. To generate the sets $(T_i)_{i\in[m]}$ used by $\mathsf{SetHSS}$, we will sample each set by including each element in $[n]$ independently with probability $q$, where $q$ is a parameter of our construction. Recall that in Sect. 6, we proved various properties of such sets when sampled in this manner. These lemmas will come in handy when analyzing the correctness and security of our FE construction. Once we have analyzed correctness and security as functions of the parameters $n, m$, and $q$, we will set these parameters to obtain our results. We will apply our construction twice. The first application will amplify a weakly secure FE where an adversary has advantage $\epsilon = c$ for some small constant $c$ to one where an adversary has advantage $\epsilon = 1/\mathsf{poly}(\lambda)$. On the second application, we amplify an FE scheme with $\epsilon = 1/\mathsf{poly}(\lambda)$ to one with $\epsilon = \mathsf{negl}(\lambda)$ (or $2^{-\lambda^c}$ for some constant $c > 0$ when dealing with subexponential adversaries).

Recall the following notation:

**Notation.** We say that ensembles satisfy $(\mathsf{poly}(\lambda) \cdot s, \epsilon)$-indistinguishability if the ensembles satisfy $(p(\lambda) \cdot s, \epsilon)$-indistinguishability for every polynomial $p(\lambda)$.

Our main results in this section are the following.

**Theorem 10.** *Assuming a $(\mathsf{poly}(\lambda), \epsilon)$-secure FE scheme for $\mathsf{P/poly}$ for some constant $\epsilon < 1/6$, there exists a $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$-secure FE scheme for $\mathsf{P/poly}$. Moreover, this transformation preserves sublinearity/compactness.*

**Theorem 11.** *Assuming a $(2^{O(\lambda^c)}, \epsilon)$-secure FE scheme for $\mathsf{P/poly}$ for some constant $\epsilon < 1/6$ and some constant $c > 0$, there exists a $(2^{O(\lambda^{c'})}, 2^{-O(\lambda^{c'})})$-secure FE scheme for $\mathsf{P/poly}$ for some constant $0 < c' < c$. Moreover, this transformation preserves sublinearity/compactness.*

## 8.1 Construction

Our FE construction makes use of the following primitives.

- Let $\mathsf{sFE} = (\mathsf{sFE.Setup}, \mathsf{sFE.Enc}, \mathsf{sFE.KeyGen}, \mathsf{sFE.Dec}, \mathsf{sFE.SFEnc}, \mathsf{sFE.SFKeyGen})$ be an $(s, \nu, \epsilon)$-secure semi-functional encryption scheme, where $\frac{1}{p(\lambda)} \leq \epsilon < 1 - \frac{1}{p(\lambda)}$ for some polynomial $p(\lambda)$. Such a scheme is implied by an $(s, \epsilon)$-secure FE scheme assuming an $(s, \nu)$-secure one-way function and this transformation preserves sublinearity/compactness.
- Let $\mathsf{SetGen}(1^n, 1^m, q)$ be an algorithm that outputs $(T_i)_{i\in[m]}$, where for each $T_i \subseteq [n]$, we include each element of $[n]$ in $T_i$ independently with probability $q$.
- Let $\mathsf{SetHSS} = (\mathsf{SetHSS.InpEncode}, \mathsf{SetHSS.FuncEncode}, \mathsf{SetHSS.Decode})$ be a set homomorphic secret sharing scheme.
- Let $\mathsf{Com}$ be a statistically binding, computationally hiding commitment scheme. ($\mathsf{Com}$ does not show up in the construction and is only used in the security proof.)

Our FE scheme is defined, with respect to parameters $n, m \in \mathbb{N}$ where $n, m = O(\mathsf{poly}(\lambda))$ and a probability $q \in [0, 1]$, as follows:

- $\mathsf{FE.Setup}(1^\lambda)$ : Setup proceeds as follows:
  1. Compute $(T_i)_{i\in[m]} \leftarrow \mathsf{SetGen}(1^n, 1^m, q)$
  2. For each $i \in [m]$, generate $\mathsf{msk}_i \leftarrow \mathsf{sFE.Setup}(1^\lambda)$.
  3. Output $\mathsf{MSK} = ((\mathsf{msk}_i)_{i\in[m]}, (T_i)_{i\in[m]})$.
- $\mathsf{FE.Enc}(\mathsf{MSK}, \mathsf{msg})$ : Encryption proceeds as follows:
  1. Parse $\mathsf{MSK}$ as $((\mathsf{msk}_i)_{i\in[m]}, (T_i)_{i\in[m]})$.
  2. Compute $(s_i)_{i\in[m]} \leftarrow \mathsf{SetHSS.InpEncode}(1^\lambda, 1^n, (T_i)_{i\in[m]}, \mathsf{msg})$.
  3. For $i \in [m]$, compute $\mathsf{ct}_i \leftarrow \mathsf{sFE.Enc}(\mathsf{msk}_i, s_i)$.
  4. Output $\mathsf{CT} = (\mathsf{ct}_i)_{i\in[m]}$.
- $\mathsf{FE.KeyGen}(\mathsf{MSK}, C)$ : Key generation proceeds as follows:
  1. Parse $\mathsf{MSK}$ as $((\mathsf{msk}_i)_{i\in[m]}, (T_i)_{i\in[m]})$.
  2. Compute $(C_i)_{i\in[m]} \leftarrow \mathsf{SetHSS.FuncEncode}(1^\lambda, 1^n, (T_i)_{i\in[m]}, C)$.
  3. For $i \in [m]$, compute $\mathsf{sk}_{C_i} \leftarrow \mathsf{sFE.KeyGen}(\mathsf{msk}_i, C_i)$.
  4. Output $\mathsf{sk}_C = (\mathsf{sk}_{C_i})_{i\in[m]}$.
- $\mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{CT})$ : Decryption proceeds as follows:
  1. Parse $\mathsf{sk}_C$ as $(\mathsf{sk}_{C_i})_{i\in[m]}$ and $\mathsf{CT}$ as $(\mathsf{ct}_i)_{i\in[m]}$.
  2. For $i \in [m]$, compute $y_i = \mathsf{sFE.Dec}(\mathsf{sk}_{C_i}, \mathsf{ct}_i)$.
  3. Output $\mathsf{SetHSS.Decode}((y_i)_{i\in[m]})$.

*Correctness.* Correctness holds provided that $\mathsf{sFE}$ is correct and that $\mathsf{SetHSS}$ is a correct set homomorphic secret sharing scheme with respect to the sets $(T_i)_{i\in[m]}$ sampled by the setup algorithm. To see this, observe that $\mathsf{sFE.Dec}(\mathsf{sk}_{C_i}, \mathsf{ct}_i) = C_i(s_i)$ since $\mathsf{ct}_i$ is an encryption of $s_i$. Thus, the output of decryption is $\mathsf{SetHSS.Decode}((C_i(s_i))_{i\in[m]}) = C(\mathsf{msg})$ by correctness of $\mathsf{SetHSS}$.

If we instantiate $\mathsf{SetHSS}$ with the scheme constructed in Sect. 5, we see that $\mathsf{SetHSS}$ is correct provided that $(T_i)_{i\in[m]}$ cover all subsets of $[n]$ of size 3 (Theorem 7). For parameters $n, m \in \mathbb{N}$ and probability $q \in [0, 1]$, the probability of $(T_i)_{i\in[m]}$ covering all subsets of size 3 when sampled in this manner was calculated in Lemma 1 to be

$$\geq 1 - n^3(1 - q^3)^m.$$

By a union bound and the correctness of $\mathsf{sFE}$, the probability that one of the $m$ copies of $\mathsf{sFE}$ is incorrect is $\leq m \cdot \mathsf{negl}(\lambda)$. Therefore, the constructed scheme is correct with probability

$$\geq 1 - n^3(1-q^3)^m - m \cdot \mathsf{negl}(\lambda).$$

*Sublinearity/Compactness.* Let $\beta \in (0,1]$ denote the sublinearity/compactness parameter of $\mathsf{sFE}$. Sublinearity/compactness follows from observing that the size of the encryption circuit is bounded by $\mathsf{poly}(\lambda, n, m) + |\mathsf{SetHSS.InpEncode}| + m \cdot |C_i|^{1-\beta} \cdot \mathsf{poly}(\lambda, |s_i|)$ for fixed polynomials independent of the size of the circuit class. Since each $|C_i| \leq |C| \cdot \mathsf{poly}(\lambda, n, m)$ and $n, m = \mathsf{poly}(\lambda)$, and $|s_i|$ and $|\mathsf{SetHSS.InpEncode}|$ are both $\mathsf{poly}(\lambda, n, m)$, it follows that the size of the encryption circuit is $\leq |C|^{1-\beta} \cdot \mathsf{poly}(\lambda)$ for some fixed polynomial independent of $C$.

Please refer to the full version for the proof of security. We also defer the instantiation of parameters to prove Theorems 10 and 11.

## 9 Amplification via Nesting

In this section, we amplify a secret key FE scheme that is secure with some constant probability $(1-\epsilon)$ to another secret key FE scheme that is secure with some larger constant probability (in the neighborhood of $(1-\epsilon^2)$). In this way, we can create an $\epsilon'$ secure FE scheme for any arbitrarily small constant $\epsilon'$ from any constantly secure FE scheme by repeating this transformation a constant number of times. We show that this amplification preserves compactness and note that although we consider the secret key variant, our proofs extend to the case of public key FE.

Our main results in this section are the following:

**Theorem 12.** *If there exists a $(\mathsf{poly}(\lambda), \epsilon)$-secure functional encryption scheme for $\mathsf{P/poly}$ for some constant $\epsilon \in (0,1)$, then there exists a $(\mathsf{poly}(\lambda), \epsilon')$-secure functional encryption scheme for $\mathsf{P/poly}$ for any constant $\epsilon' \in (0,1)$. Moreover, the transformation preserves compactness.*

**Theorem 13.** *If there exists a $(2^{\lambda^c}, \epsilon)$-secure functional encryption scheme for $\mathsf{P/poly}$ for some constant $\epsilon \in (0,1)$ and some constant $c > 0$, then there exists a $(2^{\lambda^{c'}}, \epsilon')$-secure functional encryption scheme for $\mathsf{P/poly}$ for any constant $\epsilon' \in (0,1)$ and any constant $c' < c$. Moreover, the transformation preserves compactness.*

### 9.1 Construction

Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.KeyGen}, \mathsf{FE.Dec})$ be a secret key functional encryption scheme for $\mathsf{P/poly}$ that satisfies $(s, \epsilon)-$security (as described in Definition 6) for some constant $\epsilon \in (0, 1)$.

We now construct an amplified functional encryption scheme $\mathsf{FE}^*$ as described below. Essentially, $\mathsf{FE}^*$ works by nesting the original functional encryption $\mathsf{FE}$. Intuitively, the idea is that as long as one layer of $\mathsf{FE}$ is secure, then the nested $\mathsf{FE}^*$ is secure. Therefore, we can get amplification since our nested $\mathsf{FE}^*$ is broken only when all layers of $\mathsf{FE}$ are broken. We formalize this notion in the security proof.

We will use a two-layer nesting where we have an "inner" and "outer" $\mathsf{FE}$. To encrypt a message, we first encrypt using the "inner" $\mathsf{FE}$ and then encrypt the result using the "outer" $\mathsf{FE}$. To create a function key for $C$, we first create a normal function key for $C$ using the "inner" $\mathsf{FE}$. Then, our final function key for $C$ is the function key for the "outer" $\mathsf{FE}$ of the function that decrypts the input with the "inner" function key.

$\mathsf{FE}^*$ (Amplified Functional Encryption)

- $\mathsf{Setup}(1^\lambda)$:
  1. Generate $\mathsf{msk}_1 \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $\mathsf{msk}_2 \leftarrow \mathsf{FE.Setup}(1^\lambda)$.
  2. Output $\mathsf{MSK} = (\mathsf{msk}_1, \mathsf{msk}_2)$.
- $\mathsf{Enc}(\mathsf{MSK}, m)$:
  1. Parse $\mathsf{MSK}$ as $(\mathsf{msk}_1, \mathsf{msk}_2)$.
  2. Compute $\mathsf{ct}_1 \leftarrow \mathsf{FE.Enc}(\mathsf{msk}_1, m)$.
  3. Compute $\mathsf{ct}_2 \leftarrow \mathsf{FE.Enc}(\mathsf{msk}_2, \mathsf{ct}_1)$.
  4. Output $\mathsf{CT} = \mathsf{ct}_2$.
- $\mathsf{KeyGen}(\mathsf{MSK}, C)$:
  1. Parse $\mathsf{MSK}$ as $(\mathsf{msk}_1, \mathsf{msk}_2)$.
  2. Compute $\mathsf{sk}_{C,1} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}_1, C)$.
  3. Compute $\mathsf{sk}_{C,2} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}_2, G)$ where $G(x) = \mathsf{FE.Dec}(\mathsf{sk}_{C,1}, x)$.
  4. Output $\mathsf{sk}_C = \mathsf{sk}_{C,2}$.
- $\mathsf{Dec}(\mathsf{sk}_C, \mathsf{CT})$:
  1. Output $y = \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{CT})$.

*Correctness:* If the underlying $\mathsf{FE}$ is correct, then so is the scheme $\mathsf{FE}^*$. This is because for any function $C$, message $m$, honestly generated ciphertext $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}_2, \mathsf{FE.Enc}(\mathsf{msk}_1, m))$ and key $\mathsf{sk}_C \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}_2, G)$ where $G(x) = \mathsf{FE.Dec}(\mathsf{FE.KeyGen}(\mathsf{msk}_1, C), x)$, then $\mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{CT}) = G(\mathsf{FE.Enc}(\mathsf{msk}_1, m)) = \mathsf{FE.Dec}(\mathsf{FE.KeyGen}(\mathsf{msk}_1, C), \mathsf{FE.Enc}(\mathsf{msk}_1, m)) = C(m)$. Thus, correctness holds with probability 1.

*Preserving Compactness:* It follows immediately that if $\mathsf{FE}$ satisfies compactness, then so does $\mathsf{FE}^*$. If the running time needed to compute an $\mathsf{FE}$ ciphertext is independent of the function size, then so is the running time needed to compute an $\mathsf{FE}^*$ encryption of a message.

### 9.2 Security

We will prove the following two lemmas.

**Lemma 3.** *For any constant* $\epsilon \in (0,1)$ *if*

- $\mathsf{FE}$ *is a* $(\mathsf{poly}(\lambda), \epsilon)$*-secure functional encryption scheme for* $\mathsf{P/poly}$*,*
- $\mathsf{Com}$ *is any commitment with* $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$*-computational hiding and* $\mathsf{negl}(\lambda)$*-statistical binding,*

*then* $\mathsf{FE}^*$ *is a* $(\mathsf{poly}(\lambda), \epsilon^2 + \mathsf{negl}(\lambda))$*-secure functional encryption scheme.*

**Lemma 4.** *For any constant* $\epsilon \in (0,1)$*, any constant* $c' > 0$*, and any constant* $c > c'$*, if*

- $\mathsf{FE}$ *is a* $(2^{\lambda^c}, \epsilon)$*-secure functional encryption scheme for* $\mathsf{P/poly}$*,*
- $\mathsf{Com}$ *is any commitment with* $(2^{\lambda^c}, \mathsf{negl}(\lambda))$*-computational hiding and* $\mathsf{negl}(\lambda)$*-statistical binding,*

*then* $\mathsf{FE}^*$ *is a* $(2^{\lambda^{c'}}, \epsilon^2 + \mathsf{negl}(\lambda))$*-secure functional encryption scheme.*

Since weakly-secure FE implies a weakly-secure OWF (which can then be amplified to a fully secure OWF via [38]), Theorems 12 and 13 immediately follow from Lemmas 3 and 4 by instantiating $\mathsf{Com}$ using this OWF and repeating the transformation a constant number of times.

We defer the proofs to the full version.

## 10 Amplification of Nested Public-Key Encryption

Our amplification techniques for nested functional encryption can also be easily extended to prove amplification for nested public-key encryption. We assume familiarity with public-key encryption (PKE). Our main results in this section are the following:

**Theorem 14.** *If there exists a* $(\mathsf{poly}(\lambda), \epsilon)$*- indistinguishability of encryption secure public-key encryption scheme* $\mathsf{PKE}$ *for message space* $\{0,1\}^\lambda$ *and for some constant* $\epsilon \in (0,1)$*, then there exists a* $(\mathsf{poly}(\lambda), \epsilon')$*-indistinguishability of encryption secure public-key encryption scheme* $\mathsf{PKE}^*$ *for any constant* $\epsilon' \in (0,1)$*, where* $\mathsf{PKE}^*$ *is obtained by nesting* $\mathsf{PKE}$ *a constant number of times.*

**Theorem 15.** *If there exists a* $(2^{\lambda^c}, \epsilon)$*- indistinguishability of encryption secure public-key encryption scheme* $\mathsf{PKE}$ *for message space* $\{0,1\}^\lambda$ *and for some constants* $\epsilon \in (0,1)$ *and* $c > 0$*, then there exists a* $(2^{\lambda^{c'}}, \epsilon')$*-indistinguishability of encryption secure public-key encryption scheme* $\mathsf{PKE}^*$ *for any constants* $\epsilon' \in (0,1)$ *and* $c' < c$*, where* $\mathsf{PKE}^*$ *is obtained by nesting* $\mathsf{PKE}$ *a constant number of times.*

We defer this section to the full version.

## 11 Final Amplification Results

By combining the main results of Sects. 8 and 9, we immediately obtain our final amplification results.

**Theorem 16.** *Assuming a* $(\mathsf{poly}(\lambda), \epsilon)$*-secure FE scheme for* $\mathsf{P/poly}$ *for some constant* $\epsilon \in (0,1)$*, there exists a* $(\mathsf{poly}(\lambda), \mathsf{negl}(\lambda))$*-secure FE scheme for* $\mathsf{P/poly}$*. Moreover, this transformation preserves compactness.*

**Theorem 17.** *Assuming a* $(2^{O(\lambda^c)}, \epsilon)$*-secure FE scheme for* $\mathsf{P/poly}$ *for some constant* $\epsilon \in (0,1)$ *and some constant* $c > 0$*, there exists a* $(2^{O(\lambda^{c'})}, 2^{-O(\lambda^{c'})})$*-secure FE scheme for* $\mathsf{P/poly}$ *for some constant* $0 < c' < c$*. Moreover, this transformation preserves compactness.*

**Acknowledgements.** We thank the anonymous CRYPTO reviewers for their helpful feedback regarding this work. We also thank Maciej Skórski for useful discussions about [57,58].

This research is supported in part from DARPA SAFEWARE and SIEVE awards, NTT Research, NSF Frontier Award 1413955, and NSF grant 1619348, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through Award HR00112020024 and the ARL under Contract W911NF-15-C- 0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, NTT Research, or the U.S. Government.

We would also like to thank A.K.'s cat, Mr. Floof, for emotional support during the research process, despite his complete apathy towards the research process and our existence in general.

## References

1. Agrawal, S.: Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 191–225. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_7
2. Ananth, P., Badrinarayanan, S., Jain, A., Manohar, N., Sahai, A.: From FE combiners to secure MPC and back. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11891, pp. 199–228. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_9
3. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 657–677. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_32
4. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 657–677. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_32

5. Ananth, P., Jain, A., Lin, H., Matt, C., Sahai, A.: Indistinguishability obfuscation without multilinear maps: new paradigms via low degree weak pseudorandomness and security amplification. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 284–332. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_10
6. Ananth, P., Jain, A., Naor, M., Sahai, A., Yogev, E.: Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 491–520. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_17
7. Ananth, P., Jain, A., Sahai, A.: Robust transforming combiners from indistinguishability obfuscation to functional encryption. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 91–121. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_4
8. Ananth, P., Jain, A., Sahai, A.: Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. IACR Cryptology ePrint Archive 2018/615 (2018)
9. Ananth, P., Jain, A.: Indistinguishability obfuscation from compact functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 308–326. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_15
10. Ananth, P., Sahai, A.: Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 152–181. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_6
11. Badrinarayanan, S., Goyal, V., Jain, A., Sahai, A.: A note on VRFs from verifiable functional encryption. IACR Cryptology ePrint Archive 2017/51 (2017)
12. Barak, B., Hardt, M., Kale, S.: The uniform hardcore lemma via approximate Bregman projections. In: SODA, pp. 1193–1200 (2009)
13. Bellare, M., Impagliazzo, R., Naor, M.: Does parallel repetition lower the error in computationally sound protocols? In: FOCS, pp. 374–383 (1997)
14. Bitansky, N.: Verifiable random functions from non-interactive witness-indistinguishable proofs. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part II. LNCS, vol. 10678, pp. 567–594. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70503-3_19
15. Bitansky, N., Nishimaki, R., Passelègue, A., Wichs, D.: From cryptomania to obfustopia through secret-key functional encryption. In: Hirt, M., Smith, A. (eds.) TCC 2016, Part II. LNCS, vol. 9986, pp. 391–418. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_15
16. Bitansky, N., Vaikuntanathan, V.: Indistinguishability obfuscation from functional encryption. In: Guruswami, V. (ed.) 56th FOCS, pp. 171–190. IEEE Computer Society Press, October 2015
17. Boneh, D., et al.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
18. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_16
19. Canetti, R., Halevi, S., Steiner, M.: Hardness amplification of weakly verifiable puzzles. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 17–33. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_2

20. Chen, Y.-H., Chung, K.-M., Liao, J.-J.: On the complexity of simulating auxiliary input. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 371–390. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_12
21. Damgård, I., Kilian, J., Salvail, L.: On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 56–73. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_5
22. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 342–360. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_21
23. Fischlin, M., Herzberg, A., Bin-Noon, H., Shulman, H.: Obfuscation combiners. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 521–550. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_18
24. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press, October 2013
25. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Functional encryption without obfuscation. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 480–511. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_18
26. Garg, S., Pandey, O., Srinivasan, A.: Revisiting the cryptographic hardness of finding a nash equilibrium. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 579–604. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_20
27. Garg, S., Pandey, O., Srinivasan, A., Zhandry, M.: Breaking the sub-exponential barrier in obfustopia. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 156–181. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_6
28. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 555–564. ACM Press, June 2013
29. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 503–523. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_25
30. Goyal, R., Hohenberger, S., Koppula, V., Waters, B.: A generic approach to constructing and proving verifiable random functions. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part II. LNCS, vol. 10678, pp. 537–566. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70503-3_18
31. Goyal, V., Jain, A., Sahai, A.: Simultaneous amplification: the case of non-interactive zero-knowledge. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 608–637. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26951-7_21
32. Harnik, D., Ishai, Y., Kushilevitz, E., Nielsen, J.B.: OT-combiners via secure computation. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 393–411. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78524-8_22

33. Håstad, J., Pass, R., Wikström, D., Pietrzak, K.: An efficient parallel repetition theorem. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 1–18. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_1
34. Hemenway, B., Jafargholi, Z., Ostrovsky, R., Scafuro, A., Wichs, D.: Adaptively secure garbled circuits from one-way functions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 149–178. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_6
35. Holenstein, T.: Key agreement from weak bit agreement. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 664–673. ACM Press, May 2005
36. Holenstein, T.: Strengthening key agreement using hard-core sets. Ph.D. thesis, ETH Zurich (2006)
37. Holenstein, T., Renner, R.: One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 478–493. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_29
38. Impagliazzo, R.: Hard-core distributions for somewhat hard problems. In: FOCS, pp. 538–545 (1995)
39. Jain, A., Lin, H., Matt, C., Sahai, A.: How to leverage hardness of constant-degree expanding polynomials over $\mathbb{R}$ to build $i\mathcal{O}$. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 251–281. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_9
40. Jain, A., Manohar, N., Sahai, A.: Combiners for functional encryption, unconditionally. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12105, pp. 141–168. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45721-1_6
41. Jetchev, D., Pietrzak, K.: How to fake auxiliary input. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 566–590. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_24
42. Kitagawa, F., Nishimaki, R., Tanaka, K.: Obfustopia built on secret-key functional encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 603–648. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_20
43. Klivans, A., Servedio, R.: Boosting and hard-core set construction. Mach. Learn. **51**, 217–238 (2003)
44. Komargodski, I., Segev, G.: From minicrypt to obfustopia via private-key functional encryption. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 122–151. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_5
45. Lin, H.: Indistinguishability obfuscation from constant-degree graded encoding schemes. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 28–57. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_2
46. Lin, H.: Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 599–629. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_20
47. Lin, H., Tessaro, S.: Amplification of chosen-ciphertext security. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 503–519. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_30

48. Lin, H., Tessaro, S.: Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 630–660. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_21
49. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: Dinur, I. (ed.) 57th FOCS, pp. 11–20. IEEE Computer Society Press, October 2016
50. Maurer, U., Tessaro, S.: Computational indistinguishability amplification: tight product theorems for system composition. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 355–373. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_21
51. Maurer, U., Tessaro, S.: A hardcore lemma for computational indistinguishability: security amplification for arbitrarily weak PRGs with optimal stretch. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 237–254. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_15
52. Meier, R., Przydatek, B., Wullschleger, J.: Robuster combiners for oblivious transfer. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 404–418. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_22
53. O'Neill, A.: Definitional issues in functional encryption. IACR Cryptology ePrint Archive 2010/556 (2010). http://eprint.iacr.org/2010/556
54. Pass, R., Venkitasubramaniam, M.: An efficient parallel repetition theorem for Arthur-Merlin games. In: STOC, pp. 420–429 (2007)
55. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 475–484. ACM Press, May/Jun 2014
56. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
57. Skórski, M.: Efficiently simulating high min-entropy sources in the presence of side information. In: Biryukov, A., Goyal, V. (eds.) INDOCRYPT 2015. LNCS, vol. 9462, pp. 312–325. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26617-6_17
58. Skórski, M.: A subgradient algorithm for computational distances and applications to cryptography. IACR Cryptology ePrint Archive 2016/158 (2016). http://eprint.iacr.org/2016/158
59. Tessaro, S.: Security amplification for the cascade of arbitrarily weak PRPs: tight bounds via the interactive hardcore lemma. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 37–54. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_3
60. Trevisan, L., Tulsiani, M., Vadhan, S.: Regularity, boosting, and efficiently simulating every high-entropy distribution. In: CCC, pp. 126–136 (2009)
61. Vadhan, S., Zheng, C.J.: A uniform min-max theorem with applications in cryptography. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 93–110. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_6
62. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 555–572. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_32
63. Wullschleger, J.: Oblivious transfer from weak noisy channels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 332–349. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_20