Electronic Colloquium on Computational Complexity, Report No. 110 (2026)

Security Amplification via Robust Indistinguishability Combiners*

Benny Applebaum† Nir Bitansky‡ Nathan Geier§

Abstract

A robust combiner for a cryptographic primitive P takes multiple candidate constructions of P and produces a secure construction of P provided that sufficiently many of the candidates are secure. A closely related notion is that of a security amplifier, where given a weakly secure construction of P, we aim to obtain a (strongly) secure one. Intuitively, one may expect that any robust combiner should act as an amplifier by thinking of "good randomness" as inducing secure instances, and of "bad randomness" as inducing insecure instances. Formalizing this intuition, however, has turned out to be challenging. Despite significant progress, general results remain limited and confined either to specific primitives or only to the statistical setting.

We establish a new framework of *robust indistinguishability combiners*, which greatly extends the class of combiners covered by prior work, and prove that they inherently act as security amplifiers. Our results extend to the computational setting, provided that the combiner makes a *single query* to each candidate. The new framework allows us to rederive previously known amplification results in a simplified manner, as well as prove new amplification results that have so far been out of reach.

As our main application, we present the first *security amplifier for functional encryption*, resolving an open question that first arose in constructions of indistinguishability obfuscation, and for which a gap was discovered in previous proofs. Our amplifier transforms a weak scheme with any constant indistinguishability error into one with full negligible security.

*B. A. and N. G. are supported by Israel Science Foundation (ISF) grant no. 2805/21 and by the European Union (ERC-2022-ADG) under grant agreement no.101097959 NFITSC.

†Tel Aviv University, Israel, benny.applebaum@gmail.com

‡New York University, USA, nbitansky@gmail.com

§Tel Aviv University, Israel, nathangeier.cs@gmail.com

ISSN 1433-8092

Contents

1 Introduction

Security amplification is a fundamental goal in cryptography: given a cryptographic primitive that is weakly secure in some sense, we would like to make it strongly secure. It provides a method to guarantee constructions remain secure even if their underlying components exhibit weaknesses, for example due to physical side-channel leakage, an unproven mathematical assumption that turns out to only partially hold, or advances in technology and the computational power of adversaries. It also allows researchers to simplify the task of constructing advanced primitives by letting them aim merely for weak security. Foundational examples of security amplification include Yao's amplification of weak one-way functions (OWFs) and Yao's XOR lemma for amplifying unpredictability [Yao82, GNW11]. This line of research has been extended to public-key encryption (PKE) [DNR04, HR05, LT13] and key-agreement (KE) [Hol05], weakly verifiable puzzles [CHS05], oblivious transfer (OT) [CK90, DKS99, Wul07], non-interactive zero knowledge (NIZK) [GJS19, BKP+24, BG24, AK25b], and many others.

The notion of *robust combiners* [HKN+05, Her05] is closely related, yet distinct. A (k, n)-robust combiner is a construction that takes n candidate schemes and guarantees security as long as at least k of them are secure. Such combiners are typically built using simple secret-sharing techniques, or more sophisticated tools such as secure multiparty computation (MPC) [HIKN08, AK25a].

The existence of an amplifier usually implies the existence of a combiner, as one can construct a weakly secure scheme by executing a randomly selected candidate, and then amplify this resulting scheme. Indeed, this is true for all the amplifiers mentioned above. But *does a combiner imply an amplifier*? Such a theorem would be extremely valuable, as amplifiers are typically significantly more challenging to construct and analyze, and often incur substantial additional computational overhead. Moreover, for some primitives only a combiner is known to exist, most notably indistinguishability obfuscation [FHNS16, AJN+16] and functional encryption [JMS20].

To illustrate the relation between combiners and amplifiers, let us review two classical examples that will be used across the introduction.

Direct Product. Given n candidate one-way functions $f_1, \ldots, f_n$, it is straightforward to show that the mapping $(x_1, \ldots, x_n) \mapsto (f_1(x_1), \ldots, f_n(x_n))$ is one-way provided that at least one of the functions is one-way. Thus, the direct product yields a robust combiner. Demonstrating that the same construction also serves as an amplifier is considerably more subtle (see Section 2.3 of [Gol01]).

The XOR Lemma. Suppose we are given n weakly unpredictable bits $b_1, \ldots, b_n$. Trivially, XOR is a robust combiner: if at least one bit is uniformly distributed, then so is their XOR. However, analyzing the amplification properties of this combiner in the setting where the bits are only computationally weakly unpredictable turns out to be highly nontrivial [Yao82, GNW11]. Furthermore, extending the XOR lemma to bit-strings reveals an inherent barrier to any general combiner-is-amplifier theorem. While the combiner guarantee remains unchanged, amplification may fail for certain parameter regimes. In particular, if all strings are "almost random" except that their first bit is fixed to 0, XOR-based amplification fails dramatically. This underscores the distinction between the inherently "all-or-nothing" nature of combiners and the "consistent small leakage" phenomenon that may arise for weak candidates.

Secret-Sharing-Based Amplification for Encryption Schemes. Suppose we are given a public-key encryption scheme with both privacy and correctness error. Concretely, assume that any two encryptions cannot be efficiently distinguished with advantage better than 0.1, and that

(honest) decryption fails with probability 0.1 for every message. The problem of amplifying privacy and correctness was studied by [DNR04, HR05], who obtained amplification results for bit encryption.

For large message spaces, a natural approach is to secret-share the message m into n shares $m_1, \ldots, m_n$ using a t-out-of-n threshold scheme, say with $t = 0.5n$, that tolerates a 0.2-fraction of corrupted shares (e.g., Shamir's scheme based on Reed-Solomon codes). It is straightforward to show that this yields negligible decryption error. Intuitively, security should follow from the "combining" properties of the scheme. However, no formal analysis of this approach appears to be available in the literature. Indeed, [DNR04] mention a similar idea and note that while the argument goes through for weak *information-theoretic* security, "it is not clear how to carry it through in the computational setting".

Given the above discussion, we ask:

> Is it possible to prove a general theorem that allows us to argue that a given combiner is also an amplifier?

Ideally, such a theorem would lead to new amplifiers, or, at the very least, provide a unified proof for all the above examples, possibly with sub-optimal parameters.

Related Work: Neutralizing Constructions. The potential for combiners to function as amplifiers was explored in [MPR07, MT09, LM20] through the framework of *neutralizing constructions*. In this framework, a *specification* of a cryptographic primitive is modeled as an ideal "random system" I, which can be viewed as a randomized stateful oracle. (For example, in the context of pseudorandom functions, I may be taken to be a truly random function.) Syntactically, a neutralizing construction C is another system that is given oracle access to n candidate constructions $\vec{S} = (S_1, \ldots, S_n)$ over some input/output space.

Such a vector of candidates is said to be *k-ideal* with respect to a vector of ideal systems $\vec{I} = (I_1, \ldots, I_n)$ if at least k out of the n candidates $\vec{S}$ are ideal. The neutralizing property asserts that for every k-ideal vector of candidates, the combined construction $C^{\vec{S}}$ behaves *exactly as if all n candidates were ideal*; that is, $C^{\vec{S}} \equiv C^{\vec{I}}$.

It is shown in [MPR07, MT09, LM20] that *neutralization implies amplification*: if each candidate is "close" to ideal, then the combined outcome is "close" to the ideal combined distribution $C^{\vec{I}}$. The notion of closeness is defined in [MPR07, LM20] in an information-theoretic sense (namely, via statistical distance), and is extended in [MT09] to computational indistinguishability in the special case $k = 1$.

The latter extension yields several elegant amplification theorems. For example, in the case of pseudorandom functions, one can take the ideal oracle to be a truly random function and observe that the XOR construction provides a neutralizing combiner for $k = 1$. More generally, this framework is well suited to "random primitives," such as pseudorandom generators (PRGs), pseudorandom functions (PRFs), and pseudorandom permutations (PRPs). However, it becomes overly restrictive when applied to other cryptographic primitives.

Limitations of Neutralizing Constructions. We illustrate this point using secret-sharing based amplification for weak encryption. Ideally, one would like to analyze this construction via the neutralizing-to-amplifier theorem, but it is unclear how to define the appropriate ideal oracle. A natural candidate is a "simulator" that encrypts a fixed message, e.g., the all-zero message. Then, a k-ideal vector includes k oracles that ignore their input message and use the weak scheme to encrypt zeros instead. However, this approach fails for two reasons. First, the real encryption

is only computationally indistinguishable from the simulation, and the existing theorems do not accommodate this type of approximation when $k > 1$. Second, and more importantly: even in the case $k = 1$, the resulting combiner does not qualify as a "neutralizing construction" with respect to this ideal oracle, despite being a valid combiner. To see this, suppose one of the bad candidates outputs some "recognizable" ciphertext, e.g., reveals its entire input or outputs some fixed string. Then it becomes easy to distinguish between the setting in which all candidates are ideal and the setting in which only one is ideal, violating the neutralizing property.

Indeed, the neutralizing property implicitly requires hiding both which candidate fails and the nature of the failure. In the context of encryption, this is an overkill: we do not need an encryption of x using a k-ideal vector $\vec{S}$ to be indistinguishable from an encryption of x using n ideal systems, that is, $C^{\vec{S}}(x) \approx C^{\vec{I}}(x)$. Rather, it suffices to guarantee indistinguishability between encryptions of x and x' whenever a k-ideal vector $\vec{S}$ is used, that is, $C^{\vec{S}}(x) \approx C^{\vec{S}}(x')$. For more advanced primitives with simulation-based security, such as NIZK, this distinction becomes even more apparent, since a weak simulator may simply output $\perp$ with some failure probability. In this case, distinguishing between an execution using n (weak) simulators and, say, $n/2$ becomes overwhelmingly easy for most natural combiners.

Indistinguishability Combiners Are Amplifiers. The above discussion motivates our new generalized "indistinguishability combiner" notion, requiring indistinguishability between inputs/modes to hold whenever a sufficient number of candidates is ideal. This notion extends the notion of neutralizing constructions and captures all the examples discussed so far, as well as a broad class of known combiners for cryptographic primitives with indistinguishability-based security such as PKE, KA, OT, NIZK, and even OWFs (see Example 4). We show that indistinguishability combiners inherently act as amplifiers for statistically weak candidates. Furthermore, we extend this result to the computational setting under the restriction that the combiner makes a single query to each candidate, and guarantees indistinguishability even if the non-ideal candidates are inefficient.

Security Amplification for Functional Encryption. As a main application, we employ our theorem to achieve unconditional security amplification for functional encryption (FE), a central cryptographic tool for enabling computation over encrypted inputs, first introduced by [SW05] and formalized by [BSW11, O'N10]. FE amplification was first studied by [AJS18, AJL+19] under the assumption of subexponentially secure LWE, and was later extended to the unconditional setting by Jain et al. [JKMS20]. However, a gap was subsequently discovered in their proofs [Jai26], rendering all existing amplification results invalid. We resolve this open problem by taking a fundamentally different approach to the security analysis, using our combiner-is-amplifier theorem. Our construction follows a similar design, refining and improving their implicit FE combiner.

We proceed with a detailed overview of our results.

1.1 Our Results

Our first contribution is definitional and conceptual: we introduce a new framework for *indistinguishability combiners* (iC) that overcomes the limitations of existing notions. Roughly speaking, (s, ε)-indistinguishability (i.e., ε-indistinguishability against circuits of size s) is guaranteed between inputs/modes whenever at least k out of the n candidates are "ideal". (By taking $s = \infty$, this definition also captures statistical indistinguishability.) The actual definition may at first appear somewhat technical and non-intuitive, but we will later explain how to adapt it to different settings (see Example 1 below).

An indistinguishability combiner consists of two n-oracle-aided algorithms $(C_0^{(\cdot)}, C_1^{(\cdot)})$. Its security guarantee, parameterized by (s, ε, k, n) and a vector of "ideal" oracles $\vec{I} = (I_1, \ldots, I_n)$, is the following: For every k-ideal vector of n candidates $\vec{S} = (S_1, \ldots, S_n)$, the distributions

$$C_0^{\vec{S}} \quad \text{and} \quad C_1^{\vec{S}} \tag{1}$$

are (s, ε)-indistinguishable. (Recall that a vector is k-ideal if there exists a subset $T \subseteq [n]$ of size at least k for which S_i behaves identically to I_i for all $i \in T$.) Importantly, the indistinguishability requirement (1) uses the same k-ideal oracles $\vec{S}$ over different input distributions, and does not compare directly to the all-ideal oracles as in the case of neutralizing constructions ($C^{\vec{S}} \equiv C^{\vec{I}}$). Nevertheless, neutralizing constructions fit naturally within this framework: one may let $C_0^{\vec{S}}$ execute $C^{\vec{S}}$ normally, while $C_1^{\vec{S}}$ ignores its oracle access and directly implements the ideal functionality $C^{\vec{I}}$. Under this choice, indistinguishability of the two modes is precisely the standard neutralization requirement, showing that our formulation strictly generalizes the traditional one. As we shall see, this seemingly minor reformulation turns out to be surprisingly powerful.

We utilize this definition dynamically: in the analysis of a proposed amplifier, after fixing a weak candidate, a security game and an adversary, the combiner's definition is tailored to them. For example, a (non-adaptive) challenge chosen by the adversary can be embedded into the combiner as hardcoded non-uniform advice. Moreover, as we will later see, the definition of the ideal oracle is often based on the concrete (weak) implementation of the primitive and even on a fixed adversary.

We first show that in the setting where the underlying candidates are statistically close to ideal (to some secure construction), any indistinguishability combiner acts as an amplifier, even if the guaranteed indistinguishability is merely computational. (See Theorem 1 for formal statement.)

Informal Theorem 1 (iCombiner Is Statistical Amplifier)**.** *Let $(C_0^{(\cdot)}, C_1^{(\cdot)})$ be an (s, ε, k, n) indistinguishability combiner with respect to ideal oracles $\vec{I}$. Let $\vec{R} = (R_1, \ldots, R_n)$ be a vector of candidates where each R_i is* statistically *δ-close to ideal I_i for $\delta < 0.5$. Then, $C_0^{\vec{R}}$ and $C_1^{\vec{R}}$ are $(s, \delta' + \varepsilon')$-indistinguishable for*

$$\delta' := \binom{n}{k-1}(2\delta)^{n-k+1}, \quad \textit{and} \quad \varepsilon' := \varepsilon \cdot \min\left(\binom{n}{k}, (1-2\delta)^{-k}\right).$$

Some remarks are in order.

- Recall that $\delta < 0.5$ is an inherent limitation of any general combiner-is-amplifier theorem, as demonstrated by the XOR example for weakly random strings whose first bit is fixed to 0.
- Note that δ' can be upper-bounded by $\delta' \leq (2\delta \cdot en/(n-k+1))^{n-k+1}$; this looser bound is often useful for a large k. For example, for $k = n/2$, we get that $\delta' \leq (4e\delta)^{n/2}$.
- If the combiner is imperfect with $\varepsilon > 0$, in order to ensure the error term ε' does not blow up, we usually want $\binom{n}{k}$ to be polynomial in the security parameter λ (e.g., $k < n = O(\log \lambda)$ or $k = O(1), n = \lambda$) or δ to be small enough (e.g., $\delta < 0.4/k$).

Next, we extend the above result to the setting where candidates are merely computationally close to ideal, provided the combiner makes a single, non-adaptive query to each candidate. By doing so, we implicitly limit the security notion to being "non-adaptive". (See Remark 4 for a detailed discussion of the model's limitations and Section 5.3 for some extensions.) We note that even if the combiner makes multiple queries, as long as they are non-adaptive, we can usually view it as a single-query combiner with batch queries (potentially increasing the indistinguishability error).

Informal Theorem 2 (iCombiner Is Computational Amplifier)**.** *Given an* (s, ε, k, n) single-query *indistinguishability combiner* $(C_0^{(\cdot)}, C_1^{(\cdot)})$ *and candidates* $\vec{R} = (R_1, \ldots, R_n)$ *where each* R_i *is* computationally (s, δ)*-close to ideal for* $\delta < 0.5$*,* $C_0^{\vec{R}}$ *and* $C_1^{\vec{R}}$ *are* $(s \cdot \gamma^2 - \max_{b\in\{0,1\}} \text{size}(C_b^{\vec{R}}),\ \delta' + \varepsilon' + n\gamma)$*-indistinguishable for any "slackness" parameter* $\gamma \in (0, 1)$*, with*

$$\delta' := \binom{n}{k-1}(2\delta)^{n-k+1}, \quad \varepsilon' := \varepsilon \cdot \min\left(\binom{n}{k}, (1-2\delta)^{-k}\right).$$

Intuitively, the slackness parameter γ represents the precision with which our reduction estimates probabilities. Higher precision (smaller γ) requires a longer running time (proportional to $1/\gamma^2$), creating a trade-off. A formal version of the theorem appears as Theorem 2.

Example 1 (Secret-Sharing-Based Amplification of Weak Encryption)**.** With the aid of Theorem 2, we analyze the previously described secret-sharing based amplifier for encryption. Let n take the role of the security parameter. Given a message m, the amplifier C secret-shares it into n shares $m_1, \ldots, m_n$, queries its n oracles $\vec{R} = (R_1, \ldots, R_n)$ (each an encryption scheme with an independent key) on the corresponding shares, and outputs the ciphertexts $R_1(m_1), \ldots, R_n(m_n)$. Suppose we use a $(t = 0.5n)$-out-of-n secret-sharing scheme that is robust against a 0.2-fraction of errors. Correctness amplification is immediate using Chernoff. For security: Our goal is to show that for every pair of messages x_0, x_1, the corresponding ciphertext distributions $C_0^{\vec{R}} := C(x_0)$ and $C_1^{\vec{R}} := C(x_1)$ are computationally indistinguishable even if R_i is instantiated with weakly-secure encryption scheme.

We apply Theorem 2 as follows. Define the i-th ideal oracle by $I_i(m_i) := R_i(\mathbf{0})$, namely, an oracle that ignores its input share and outputs an encryption of the all-zero message under the i-th scheme. For these choices, $(C_0^{(\cdot)}, C_1^{(\cdot)})$ is a single-query $(k = n - t + 1)$-out-of-n *perfect* indistinguishability combiner with parameters $s = \infty$ and $\varepsilon = 0$. Indeed, for every k-ideal vector of oracles $\vec{S}$, the output distribution of C depends only on $< t$ shares (corresponding to bad candidates) and hence becomes independent of the input message x_b. We now consider the case in which each R_i is instantiated with a $(\delta = 0.1)$-private encryption scheme. Then the real and ideal oracles are δ-indistinguishable by polynomial-size adversaries, and the theorem yields $\delta' \leq 2^n \cdot 0.2^{n/2} \leq 2^{-\Omega(n)}$ and $\varepsilon' = 0$. Consequently, $C_0^{\vec{R}}$ and $C_1^{\vec{R}}$ cannot be efficiently distinguished with non-negligible advantage (better than $2^{-\Omega(n)} + \alpha(n)$ for an arbitrary inverse polynomial $\alpha(\cdot)$), establishing the security of the amplifier.[1] $\diamond$

Additional examples appear in Section 5.

Finally, as our main application, we obtain the first unconditional security amplification result for functional encryption. To this end, we construct a new FE combiner, building on and extending the techniques of [JKMS20], and then apply the combiner-to-amplifier theorem (Theorem 2).

Informal Theorem 3 (See Theorems 5 and 6)**.** *For any constant* $\delta < 1/(6e)$*, any* δ*-weak functional encryption scheme can be unconditionally amplified to full security, achieving negligible (resp., subexponential) error provided the base scheme is weakly secure against polynomial-size (resp., subexponential-size) adversaries.*

The above implies FE amplification starting with any constant error, as we can reduce it to any smaller constant using nesting as observed by [JKMS20].

Informal Corollary 1. *For any constant* $\delta < 1$*, any* δ*-weak functional encryption scheme can be unconditionally amplified to full security, achieving negligible (resp., subexponential) error provided the base scheme is weakly secure against polynomial-size (resp., subexponential-size) adversaries.*

[1] If the base scheme is (subexp, δ)-secure, we can reduce the distinguishing advantage to 1/subexp as well.

Computational Theorem Limitations. Let us briefly point out the two limitations of our computational combiner-is-amplifier theorem: First, we view each candidate as a randomized function that can be accessed at most once. In particular, this only allows us to accommodate weakly selective security (and not adaptive security). For some primitives, such as CPA-secure public-key encryption, this may not be sufficient for direct amplification; although in the case of PKE (as well as some other cases) selective security can be later upgraded to (adaptive) CPA security. Second, the combiner must guarantee indistinguishability even when the non-ideal candidates are inefficient. For a detailed discussion of these limitations and potential extensions, see Remark 4.

2 Technical Overview

The idea that robust combiners should act as natural security amplifiers is quite intuitive. For the sake of this discussion, let us focus on $(1, n)$ combiners. Say we are given such a combiner C and a candidate R that is δ-close to ideal I, then one might hope $C^{\vec{R}}$ achieves δ^n security. This intuition relies on the assumption that with probability $1-\delta$ the real candidate behaves ideally and otherwise fails, or more formally that $R \equiv (1-\delta)I + \delta F$ for some residual "failure" candidate F. However, standard notions of closeness such as statistical distance (maximal distinguishing advantage) do not offer us that guarantee. Specifically, conditioning on an "agreement event" could bias the behavior of the ideal system. To demonstrate this, let us get back to the XOR-amplification example from the introduction. Consider $R \equiv (0, U_{m-1})$ and $I \equiv U_m$, where $U_m \leftarrow \{0,1\}^m$ and $\equiv$ denotes equality in distribution. Then $R \overset{s}{\approx}_{0.5} I$ (i.e., their statistical distance is 0.5), yet we cannot write $R \equiv 0.5I + 0.5F$ for any distribution F, since strings starting with 1 have positive mass under I but zero mass under R. Indeed, as already mentioned, in the case that C is the XOR combiner, we do not achieve any amplification at all.

Although the above argument does not go through as stated, the underlying intuition can be formalized with a more careful coupling-based analysis. It is well known that if random variables R and I satisfy $R \overset{s}{\approx}_{\delta} I$, then there exists a joint distribution (X, Y) with marginals R, I such that $\Pr[X = Y] = 1 - \delta$. Lanzenberger and Maurer [LM20] proved that this fundamental property generalizes to the setting of random systems. Another way to interpret it is that we can write $R \equiv (1-\delta)S + \delta R'$ and $I \equiv (1-\delta)S + \delta I'$ for some shared system S and residual failure systems R', I'. We refer to this below as *the system decomposition lemma*.

2.1 The Statistical Setting

Using the above lemma in the setting of (k, n) combiners, we decompose the systems one by one. Specifically, replacing the last system R_n with ideal I_n results in a $(k-1, n-1)$ combiner, and replacing it with one of the failure systems R'_n, I'_n results in a $(k, n-1)$ combiner. (We do not replace with S_n as it cancels out.) Explicitly, for $b \in \{0,1\}$, we have

$$C_b^{\vec{R}_n} \equiv (1-\delta) \cdot C_b^{\vec{R}_{n-1}, S_n} + \delta \cdot C_b^{\vec{R}_{n-1}, R'_n},$$
$$C_b^{\vec{R}_{n-1}, I_n} \equiv (1-\delta) \cdot C_b^{\vec{R}_{n-1}, S_n} + \delta \cdot C_b^{\vec{R}_{n-1}, I'_n},$$

yielding

$$\Delta(C_0^{\vec{R}_n}, C_1^{\vec{R}_n}) \leq \Delta(C_0^{\vec{R}_{n-1}, I_n}, C_1^{\vec{R}_{n-1}, I_n}) + \delta \cdot \Delta(C_0^{\vec{R}_{n-1}, S_n}, C_1^{\vec{R}_{n-1}, S_n}) + \delta \cdot \Delta(C_0^{\vec{R}_{n-1}, I'_n}, C_1^{\vec{R}_{n-1}, I'_n}),$$

and the recurrence relation $\Delta(k, n) \leq \Delta(k-1, n-1) + 2\delta \cdot \Delta(k, n-1)$, for the absolute value of distinguishing between $(C_0^{(\cdot)}, C_1^{(\cdot)})$. For the base cases, $k = 0$ means we have satisfied the sufficient

ideal systems requirement, so we can use the security guarantee of the combiner $\Delta(0,n)=\varepsilon$. On the other hand, $k=n$ means we are out of options and need to switch all remaining systems to ideal. A simple hybrid argument gives us $\vec{R} \overset{s}{\approx}_{n\cdot\delta} \vec{I}$, yielding $\Delta(n,n)=2n\cdot\delta+\varepsilon$ due to

$$C_0^{\vec{R}_n} \overset{s}{\approx}_{n\cdot\delta} C_0^{\vec{I}_n} \overset{s}{\approx}_{\varepsilon} C_1^{\vec{I}_n} \overset{s}{\approx}_{n\cdot\delta} C_1^{\vec{R}_n} .$$

Finally, using Pascal's rule $\binom{n}{k}=\binom{n-1}{k-1}+\binom{n-1}{k}$ and $2\delta<1$, we derive $\delta'(k,n)\le\binom{n}{k-1}(2\delta)^{n-k+1}$ and $\varepsilon'(k,n)\le\binom{n}{k}\varepsilon$. In the full proof, we improve these bounds with a tight analysis of the recurrence relation by counting grid paths, and a tighter bound than the hybrid argument's $n\cdot\delta$. We also derive another upper bound $\varepsilon'(k,n)\le\varepsilon(1-2\delta)^{-k}$, which is useful if $\binom{n}{k}$ is too large and δ is small enough.

Comparison with Prior Techniques. We note that our proof technique extends to neutralizing constructions, where in fact we could shave a factor 2 off δ' due to the base case (n,n) switching between $\vec{R}$ and $\vec{I}$ only once. In comparison, the bound of [LM20, Theorem 3] for $\delta'(k,n)$ is slightly tighter than ours, with the simplified bound above being the leading term in both cases. For example, for $(k,n)=(2,3)$, the simplified bound is $6\delta^2$, our tighter bound is $6\delta^2-2\delta^3$ and their tighter bound is $6\delta^2-3\delta^3$. Furthermore, they are able to capture weaker neutralizing constructions where non-ideal systems must be real. On the other hand, their proof is more complex and their theorem is limited to the case where the combiner is *perfect*, i.e., the combiner's error ε is assumed to be zero. Indeed, the technique from [LM20] seems unsuitable for dealing with a non-zero error $\varepsilon>0$ (which is crucial for computational constructions).

Let us illustrate this point with an overview of the technique used for the simpler case $k=1$, introduced by [MPR07] (see also [Tes10, Section 4.2.1]) and extended to $k>1$ by [LM20] using a related approach with similar limitations. The idea is to use the XOR lemma to argue that for R,I with $R \overset{s}{\approx}_{\delta} I$, the following oracles are δ^n-close: in $\vec{S}_{\text{Odd}}$ each candidate is randomly chosen to be real/ideal conditioned on the total number of ideal oracles being odd, and $\vec{S}_{\text{Even}}$ is similarly defined. Since $\vec{S}_{\text{Odd}}$ always contains at least one ideal candidate, according to the definition of a $(1,n)$ neutralizing construction, we have $C^{\vec{S}_{\text{Odd}}}\equiv C^{\vec{I}}$. As for $\vec{S}_{\text{Even}}$, the only case where there is no ideal candidate is the all-real $\vec{R}$, occurring with probability 2^{-n+1}. We conclude that $C^{\vec{I}} \overset{s}{\approx}_{\delta^n} (1-2^{-n+1})\cdot C^{\vec{I}}+2^{-n+1}\cdot C^{\vec{R}}$, and hence $C^{\vec{I}} \overset{s}{\approx}_{(2\delta)^n/2} C^{\vec{R}}$. However, if we try to add an error ε to the security guarantee, it is multiplied by a factor 2^n, which could very well make it greater than 1. In comparison, our ε' bound grows at most by a factor $\min(n,\,1/(1-2\delta))$ for $k=1$, which is just some constant for any constant $\delta<0.5$.

2.2 The Computational Setting

To extend the combiner-is-amplifier theorem to the computational setting, we must address two weaknesses in the underlying hypothesis: (1) the combiner itself guarantees only computational security (provided that k candidates are secure); and (2) the weak candidates are only computationally close to ideal. We observe that the first item can be handled with no change to the proof, by considering the distinguishing advantage of a specific adversary D over $(C_0^{\vec{R}}, C_1^{\vec{R}})$. The second item requires additional care and is addressed using the hardcore lemma, as explained below.

We briefly mention that, to the best of our knowledge, item (1) has not been studied in the neutralizing constructions literature. Moreover, as discussed above, the techniques of [LM20] are generally not suitable for accommodating a nonzero error, even in the statistical case. For (2), Maurer and Tessaro [MT09] proved a tight neutralizing bound of $(2\delta)^n/2$ for the special case $k=1$,

and extended their result to computationally weak cc-stateless candidates.[2] However, as we will see, this result imposes a strong requirement on the indistinguishability properties of the underlying primitives, specifically indistinguishability under multiple queries.

The Hardcore Lemma. A central tool for extending statistical bounds to the computational setting is the Hardcore Lemma [Imp95, Hol05]. Impagliazzo [Imp95] originally introduced the lemma to structure mild hardness of decision problems into a dense "hard core" set of inputs, and Holenstein later provided a uniform and tighter proof [Hol05]. We utilize the hardcore lemma here in its indistinguishability formulation [MT10, BG24]. This formulation can be viewed as the computational analogue of coupling. Recall that for statistically close random variables $R \overset{\mathsf{s}}{\approx}_\delta I$, coupling guarantees that we can write $R \equiv (1-\delta)S + \delta R'$ and $I \equiv (1-\delta)S + \delta I'$. The hardcore lemma provides a similar guarantee for computationally close random variables $R \overset{\mathsf{c}}{\approx}_\delta I$, but with a relaxation: instead of a single identical core, the random variables decompose into two distinct but computationally indistinguishable cores S_R and S_I. Explicitly, this allows us to write $R \equiv (1-\delta)S_R + \delta R'$ and $I \equiv (1-\delta)S_I + \delta I'$, where $S_R \overset{\mathsf{c}}{\approx} S_I$. In particular, as observed by [BG24], for the hybrid $H := (1-\delta)S_I + \delta R'$ we have $R \overset{\mathsf{c}}{\approx} H \overset{\mathsf{s}}{\approx}_\delta I$. Getting back to the setting of indistinguishability combiners, this allows us to reduce the computational setting to the statistical setting; that is, we first obtain the statistical bound for $\vec{H}$ and $\vec{I}$, and then switch to $\vec{R}$ which is indistinguishable under efficient transformations.

A significant hurdle is that the hardcore lemma *fails for general interactive systems* because the reduction requires rewinding, which is incompatible with stateful interaction [HR08]. Tessaro [Tes11] generalized the lemma to cc-stateless systems (where rewinding is not an issue), but their technique inherently requires the systems to be δ-indistinguishable under many queries. This may be natural for primitives like PRFs or PRPs, but for general cryptographic primitives weakness is often quantified with respect to "one-shot" security games.[3] We observe that the hybrid methodology above can be extended to the single-query setting, because if for every query $q \in \mathcal{Q}$ we have $R(q) \overset{\mathsf{c}}{\approx}_\delta I(q)$, then we can invoke the random variable version to obtain $R(q) \overset{\mathsf{c}}{\approx} H(q) \overset{\mathsf{s}}{\approx}_\delta I(q)$, and define the single-query system H accordingly. Some care is needed here, though, because H may very well be an inefficient system. Still, we observe that for single-query systems $R \overset{\mathsf{c}}{\approx} H$, the hybrid argument $\vec{R} \overset{\mathsf{c}}{\approx} \vec{H}$ still works even if only $\vec{R}$ is efficient, intuitively because we can fix the queries and their responses one by one as non-uniform advice. This $\vec{R} \leftrightarrow \vec{H}$ switching technique allows us to generalize the statistical bound to the computational single-query setting. Intuitively, the fact that H may be inefficient relates to the fact that our combiner must withstand inefficient "bad" candidates (that are explainable by the base real/ideal systems).

2.3 FE Amplification

Finally, we demonstrate the usefulness of our single-query combiner-is-amplifier theorem by obtaining the first unconditional security amplification for functional encryption. To be more precise, the underlying combiner is computational, assuming OWFs, but since weak FE implies OWFs we do not require any additional assumption.

Functional Encryption. Roughly speaking, a functional encryption scheme allows us to encrypt messages $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(x)$ and generate functional keys $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(f)$ for functional

[2] The system selects a fixed query-response behavior upfront, and maintains no history.

[3] This reflects the goal of amplifying the weakest qualitative guarantee that can be lifted into a standard security notion.

decryption $f(x) \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$. (Our technique is oblivious to whether the underlying weak FE is public key or secret key, hence we omit that.) We require *single-key security*,[4] meaning that given two messages x and x' such that $f(x) = f(x')$, an adversary cannot tell encryptions of x and x' apart, even if they are given FSK_f—we refer to this as IND-security. An alternative security notion is that $(\mathsf{CT}, \mathsf{FSK}_f)$ can be simulated given only $(f, f(x))$—we refer to this as SIM-security. Our amplifier constructs an IND-secure FE given a weak SIM-secure FE, and we rely on the fact that these two definitions are essentially equivalent (see Section A).

A crucial property in FE applications is *succinctness*—the complexity of the encryption algorithm must not scale with the circuit size $|f|$. In fact, without succinctness, FE can be constructed from a plain encryption scheme [SS10] (and in particular can be easily amplified). Therefore, it is essential that the constructed amplifier preserves succinctness.

Limitations of Existing FE Combiners. In order to use our combiner-is-amplifier theorem, we need an FE combiner. FE combiners are studied explicitly in [JMS20] and implicitly in [JKMS20].[5] Both works employ MPC-based techniques in the form of *homomorphic secret sharing* (HSS) [BGI16, BGI+18]. However, we cannot use these combiners directly in our theorem for several reasons. First, the combiner of [JMS20] sends multiple queries to each candidate. In addition, the amplifier construction presented in [JKMS20] does not induce a proper combiner with negligible error for a constant δ. More importantly, we require our combiner to remain secure even against *inefficient bad candidates*, which necessitates a stronger "semi-malicious" privacy notion from the underlying MPC (or HSS) protocol. To address these limitations, we present a new FE combiner that builds on the HSS-based approach of [JMS20, JKMS20], while introducing several important modifications.

Homomorphic Secret Sharing (HSS). Roughly speaking, we require the following (somewhat non-standard) variant of an n-party HSS scheme. Given an input x, a sharing algorithm $\mathsf{HSS.Share}(x)$ splits x into n shares $(x_1, \ldots, x_n)$, each of size $|x| \cdot \mathrm{poly}(\lambda)$. Intuitively, each of these shares will be given to an *HSS party*. Given a public target function f, the procedure $\mathsf{HSS.Eval}(f, i, \cdot)$ generates, for every party $i \in [n]$, a local processing function f_i of size $|f| \cdot \mathrm{poly}(\lambda)$. Each f_i is applied to the corresponding local share, producing an output share $y_i = f_i(x_i)$. Finally, given all the output shares $(y_1, \ldots, y_n)$, the value $f(x)$ can be recovered by applying a decoder $\mathsf{HSS.Dec}(y_1, \ldots, y_n)$. Unlike the standard setting of HSS, we impose no "simplicity" requirement on the decoder. However, note that the shares are independent of the target function, and their size depends only on the security parameter.

Just like in standard secret-sharing (or MPC protocols), security is parameterized with a threshold $t \in [n]$, and we require that the shares $x_T = (x_i)_{i \in T}$ of any t-bounded corrupted subset $T \subseteq [n]$ should be independent of x. Moreover, just like in standard MPC protocols, we should be able to jointly simulate x_T together with the outputs of the "honest" parties $y_{\overline{T}} = (y_i)_{i \notin T}$ based on f and the output $f(x)$. We refer to this notion of security as *t-out-of-n privacy*. Jumping ahead, we will soon see that a somewhat stronger notion of security is needed.

HSS-Based Amplification. The above notion of HSS naturally gives rise to an FE amplifier, analogous to the SS-based combiner used for standard encryption schemes. Specifically,

[4] The literature contains additional notions of FE, such as *collusion-resistant* FE. We focus on the single key notion for its simplicity and the fact that it is already expressive enough to imply all other common notions of FE.

[5] While the construction of [JKMS20] lacks a proof that it serves as an amplifier, it does seem to work as a combiner.

given a weak FE candidate FE^w, the encryption algorithm $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(x)$ first computes shares $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(x)$ and then encrypts each share independently as $\mathsf{CT}_i \leftarrow \mathsf{FE.Enc}_i^w(x_i)$ under FE^w, using fresh keys. The functional-key generation $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(f)$ proceeds by computing $\mathsf{FSK}_{f_i} \leftarrow \mathsf{FE.Keygen}_i^w(f_i)$ for functions defined as $f_i := \mathsf{HSS.Eval}(f, i, \cdot)$. Finally, functional decryption $y \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$ evaluates $y_i \leftarrow \mathsf{FE.Dec}_i^w(\mathsf{FSK}_{f_i}, \mathsf{CT}_i)$ and reconstructs the output as $y := \mathsf{HSS.Dec}(y_1, \ldots, y_n)$. Correctness follows directly from the correctness of the weak FE and HSS. Succinctness relies on the fact that the HSS function and share sizes grow only by a factor polynomial in the security parameter.

The main idea of the security analysis is as follows. We are given a weak FE candidate, a function f, and inputs x, x' such that $f(x) = f(x') = y$. We define a single-query indistinguishability combiner that, given either x or x', first shares it as $(x_1, \ldots, x_n)$, and then sends to each oracle i the query (x_i, y_i), where $y_i = f_i(x_i)$. The real oracle $R_i(x_i, y_i)$ ignores y_i and returns $(\mathsf{CT}_i, \mathsf{FSK}_{f_i})$, namely an encryption of x_i together with a functional key for f_i. The ideal oracle $I_i(x_i, y_i)$ ignores x_i and returns a simulated ciphertext and functional key pair, as produced by the weak FE simulator $\mathsf{FE.Sim}^w(f_i, y_i)$. By the security of the weak FE scheme, the real and ideal oracles are δ-computationally close. Thus, to invoke our main theorem and obtain amplification, it remains to show that the above construction forms a secure combiner with respect to the ideal oracles.

At a high level, we show that if at least $k = n - t + 1$ of the oracles are "good"—namely, behave ideally (like the simulator) and therefore depend only on y_i—then computational indistinguishability is preserved between using x or x' as input. Indeed, for any t-subset of bad oracles $T \subseteq [n]$, we can use the HSS simulator to simulate x_T and $y_{[n]} = (y_i)_{i \in [n]}$ (based on the output $y = f(x)$ and the function f), and then feed (x_i, y_i) to "bad" oracles $i \in T$ and $(\bot, y_i)$ to "good" oracles $i \notin T$.

However, some care is required, since the combining property must hold even when the "bad" candidates are *inefficient* and the simulation security is only *computational*. To address this, we strengthen the privacy guarantees of the underlying HSS so that security holds even against *computationally unbounded* corrupted parties. Concretely, it suffices to require that the corresponding shares $x_T \leftarrow \mathsf{HSS.Share}(x)$ are distributed independently of x, and that the simulator operates correctly for any *semi-malicious* choice of shares x_T. Specifically, the simulation $(y_1, \ldots, y_n) \leftarrow \mathsf{HSS.Sim}(f, y, T, x_T)$ should succeed for any fixed valid (i.e., in the support) choice of shares x_T provided to the bad candidates. In our amplification proof, this property allows us to arbitrarily fix the inputs and outputs of the "bad" oracles and treat them as non-uniform advice.[6] For a high-level comparison detailing how our use of the hardcore lemma avoids the simulation flaw of prior work [JKMS20], see Remark 6.

Constructing HSS. We construct the required HSS primitive using a combination of 3-party-local multiparty randomized encoding (MPRE) [ABT18] and an elegant virtualization technique from [JKMS20]. We now describe the construction.

For a target function f and a parameter m, we first define an m-party functionality $f^{\oplus}(z_1, \ldots, z_m)$, in which each party $i \in [m]$ provides an additive share z_i of an input z. The functionality reconstructs z and outputs $f(z)$ to all parties. We then construct a 3-party-local MPRE for $f^{\oplus}$.

At a high level, $f^{\oplus}$ is "encoded" by a functionality $\hat{f}((z_1, r_1), \ldots, (z_m, r_m))$ that takes as input each party's share z_i together with private randomness r_i. Given a sample from $\hat{f}(\vec{z}, \vec{r})$ (for uniformly random r_i's), one can efficiently recover $f^{\oplus}(\vec{z})$. Conversely, given $f^{\oplus}(\vec{z})$, one can simulate $\hat{f}(\vec{z}, \vec{r})$ along with the local views $(z_i, r_i)_{i \in T}$ for any strict subset $T \subsetneq [m]$ of the parties. (In fact,

[6] Note that although the corrupted HSS parties may be computationally unbounded, the simulation itself is only required to be computational. Put differently, x_T (and y_T) are statistically independent of x, whereas the joint distribution $(x_T, (y_i)_{i \in [n]})$ is only computationally indistinguishable with respect to x. This non-standard combination of information-theoretic and computational security is a central ingredient in our amplification result.

we will require a semi-malicious notion of security here as well.) The MPRE $\hat{f}$ is 3-party local, meaning that it can be decomposed into functions $\{\hat{f}_{i,j,k}\}_{i,j,k\in[m]}$, where each $\hat{f}_{i,j,k}$ depends only on inputs from parties i, j, k. We construct such an MPRE from one-way functions using standard techniques.

Next, for appropriate choices of t and n (discussed below), we construct a t-out-of-n HSS using the virtualization technique of [JKMS20]. We begin by splitting the input x into m additive shares $z_1, \ldots, z_m$, and use the MPRE to define m virtual MPRE parties, where the ith party holds (z_i, r_i). Then, we group the m MPRE parties into n subsets $S_1, \ldots, S_n \subseteq [m]$, each representing an HSS party S_j that "contains" all MPRE parties indexed by S_j. Accordingly, the share of the jth HSS party is $(z_i, r_i)_{i\in S_j}$, and its local computation f_j corresponds to the MPRE computations $(\hat{f}_{i_1,i_2,i_3})_{i_1,i_2,i_3\in S_j}$ over all triples (i_1, i_2, i_3) contained in S_j.

The subsets are required to satisfy two combinatorial properties: (1) for correctness, every triple $\{i_1, i_2, i_3\}$ must be covered by some subset S_j; and (2) for privacy (parameterized by $t \in [n]$), the union of any t subsets should not cover all elements of $[m]$. In [JKMS20], a randomized construction of such subsets is used, but it incurs an error that is too large for our setting.[7]

To address this, we provide two explicit deterministic constructions of subsets that incur no error. The first achieves $t = n/3$ at the cost of taking m to be exponential in n, forcing n to be logarithmic in the security parameter λ. In the second construction, we settle for $t = \Theta(n^{1/3})$, which allows m to be polynomial in n and thus n can be taken to be $\mathrm{poly}(\lambda)$. Instantiating our combiner with the first construction yields our main FE amplifier, which achieves full security with negligible error. Finally, by combining both amplifiers, we extend our results to the subexponential security regime. For further details, see Section 8.2.

3 Preliminaries

For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. For a sequence $(x_1, \ldots, x_n)$ and a subset $V \subseteq [n]$, we denote by x_V the subsequence corresponding to indices in V. For a distribution X over a finite set $\mathcal{X}$, we use $x \leftarrow X$ to denote the result of sampling according to X, and $x \leftarrow \mathcal{X}$ to denote a uniformly random sample from the set.

We rely on standard computational concepts and notation:

- We say that a function $p : \mathbb{N} \to \mathbb{R}$ is *polynomially bounded*, often denoted $p \leq \mathrm{poly}$, if there exist constants $d, N \in \mathbb{N}$ such that $\forall n \geq N : p(n) \leq n^d$. We say that a function $\mu : \mathbb{N} \to [0, 1]$ is *negligible*, often denoted $\mu = \mathrm{negl}$, if for every constant $c > 0$ there exists an $N \in \mathbb{N}$ such that $\forall n \geq N : \mu(n) \leq n^{-c}$.

- A PPT algorithm is a probabilistic polynomial-time algorithm. A family of circuits $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ is $s(\lambda)$-*sized* if $\forall \lambda \in \mathbb{N} : \mathrm{size}(C_\lambda) \leq s(\lambda)$. It is *polynomial-sized* if $s \leq \mathrm{poly}$. We follow the common practice of modeling any efficient adversary as a family of polynomial-size circuits $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$. We often omit the subscript when it is clear from the context and shorten to $\{C\}_\lambda$, where $\lambda \in \mathbb{N}$ is the usual notation for the security parameter.

- For two random variables X, Y and $\varepsilon \in [0, 1]$, we write $X \overset{\mathsf{s}}{\approx}_\varepsilon Y$ to denote the fact that the *statistical distance* between X and Y is at most ε, and say that X and Y are ε *statistically indistinguishable*. Similarly, for $s \in \mathbb{N}$, we write $X \overset{\mathsf{c}}{\approx}_{s,\varepsilon} Y$ to denote the fact that the

[7] Specifically, for amplifying a constant δ, we must take $n = O(\log \lambda)$ to ensure the term ε' in our combiner-is-amplifier theorem remains small. For this choice of n, the probabilistic construction of [JKMS20] yields a non-negligible indistinguishability error of $\varepsilon = 1/\mathrm{poly}(\lambda)$.

computational distance between X and Y against s-sized circuits is at most ε, namely for every such Boolean circuit C we have $|\mathbb{E}[C(X) - C(Y)]| \leq \varepsilon$. In that case, we say that X and Y are ε *computationally indistinguishable against s-sized circuits.*

The above naturally extends to ensembles: for two ensembles $\{X\}_\lambda, \{Y\}_\lambda$ and a function $\varepsilon : \mathbb{N} \to [0,1]$, we write $X \overset{\mathsf{s}}{\approx}_\varepsilon Y$ if for all large enough $\lambda \in \mathbb{N}$, $X_\lambda \overset{\mathsf{s}}{\approx}_{\varepsilon(\lambda)} Y_\lambda$. Similarly, for $s : \mathbb{N} \to \mathbb{N}$, we write $X \overset{\mathsf{c}}{\approx}_{s,\varepsilon} Y$ if for all large enough $\lambda \in \mathbb{N}$, $X_\lambda \overset{\mathsf{c}}{\approx}_{s(\lambda),\varepsilon(\lambda)} Y_\lambda$. We omit the bound s on the size when referring to any polynomial-sized family of distinguishers. In addition, we may drop ε from the subscript when it is a negligible function.

Lemma 1 (Chu-Vandermonde Identity (Alternative Form))**.** *For any $j, k, n \in \mathbb{N}$ with $j \leq k \leq n$, we have*

$$\sum_{m=j}^{n-k+j} \binom{m}{j}\binom{n-m}{k-j} = \binom{n+1}{k+1}.$$

Proof. In order to pick a subset of size $k+1$ out of $n+1$, we can first pick the $(j+1)$-th smallest element to be $(m+1)$ for $m \in \{j, \ldots, n-k+j\}$, then another j smaller elements in $[m]$, and $k-j$ larger elements out of $\{m+2, \ldots, n+1\}$. □

3.1 Finite Discrete Systems

In this paper, we focus on finite discrete systems. Our tree-based formalism is essentially equivalent to the standard characterization of random systems using sequences of conditional distributions [Mau02], and our classification is based on [MR11]. We emphasize that our framework aims to capture the external *behavior* of a system, not its implementation. A reader already comfortable with the notion of a system may skip to Section 3.2.

Formalizing Interactive Systems. We characterize external interaction using three types of systems: *resources*, *distinguishers*, and *converters*. In a nutshell:

- *Resources* abstract randomized stateful (possibly inefficient) algorithms: Given an input, the algorithm updates its internal state and samples an output based on a probability distribution that depends on the internal state.
- *Distinguishers* abstract randomized oracle-aided algorithms with a binary output. When the oracles are instantiated with a resource, the algorithm defines a probability distribution over 0/1 values.
- *Converters* abstract oracle-aided randomized stateful (possibly inefficient) algorithms. Given an input, the algorithm makes a series of calls to its oracles while updating its internal state, and generates an output. By instantiating the oracles with resources, the converter simplifies into a resource.

We denote by $\mathcal{X}, \mathcal{Y}$ the spaces for incoming and outgoing messages. We always assume these are finite sets that include the special symbol $\perp$: if a resource accepts at most $h \in \mathbb{N}$ queries any subsequent queries are assumed to return $\perp$, and similarly, any invalid queries (not in input space) are assumed to return $\perp$. See more notational conventions below.

We model systems using alternating trees, which naturally capture the back-and-forth of stateful interaction: each alternating level represents a shift between queries and responses, and every node corresponds to a partial transcript.

Definition 1 (Finite Alternating Tree)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h \in \mathbb{N}$, an $(h, \mathcal{X}, \mathcal{Y})$ alternating tree is a rooted tree of height $2h$ where every node at even depth has $|\mathcal{X}|$ children labeled by the elements of $\mathcal{X}$, and every node at odd depth has $|\mathcal{Y}|$ children labeled by the elements of $\mathcal{Y}$.

Definition 2 (Resource)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h \in \mathbb{N}$, an h-round $(\mathcal{X}, \mathcal{Y})$-resource is an $(h, \mathcal{X}, \mathcal{Y})$ alternating tree where every *odd-depth* node v is assigned a distribution $D_v : \mathcal{Y} \to [0, 1]$.

Given a sequence of inputs $(x_i)_{i\in[h]}$, the outputs $(y_i)_{i\in[h]}$ are obtained by advancing from the root v_ϵ to the x_1th child v_{x_1}, sampling $y_1 \leftarrow D_{v_{x_1}}$ and advancing according to y_1 then x_2 arriving at v_{x_1,y_1,x_2}, sampling $y_2 \leftarrow D_{v_{x_1,y_1,x_2}}$ and so on.

For example, a random variable Y over $\mathcal{Y}$ could be modeled as a 1-round resource that ignores its first query x and returns $y \leftarrow Y$ (i.e., $D_v \equiv Y$ for all depth-1 nodes).

Definition 3 (Distinguisher)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h \in \mathbb{N}$, an h-round $(\mathcal{X}, \mathcal{Y})$-distinguisher is an $(h, \mathcal{X}, \mathcal{Y})$ alternating tree where every *internal even-depth* node v is assigned a distribution $D_v : \mathcal{X} \to [0, 1]$, and every *leaf* v is assigned a boolean value $b_v \in \{0, 1\}$.

Given a distinguisher D and a resource S, their interaction D^S results in a boolean distribution: fill the odd-depth nodes of D using the distributions of S to obtain a tree where all internal nodes are assigned distributions and all leaves are assigned boolean values—which naturally induces a boolean distribution. We remark that due to our $\perp$ conventions, the interaction is well defined even if S is an h'-round resource for $h' < h$.

Definition 4 (Converter)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h, n \in \mathbb{N}$, an h-round n-oracle-aided $(\mathcal{X}, \mathcal{Y})$-converter is an h-round $(\mathcal{X}, \mathcal{Y}')$-resource for $\mathcal{Y}' := (\{0\} \cup [n]) \times \mathcal{Y}$.

A 0-oracle-aided converter corresponds to a plain $(\mathcal{X}, \mathcal{Y})$-resource. Intuitively, the augmented alphabet $\mathcal{Y}'$ routes the converter's outgoing messages either to the external distinguisher (index 0) or to one of its n underlying oracles. Given an index $i \in [n]$ and a $(\mathcal{Y}, \mathcal{X})$-resource S_i, we naturally obtain an $(n-1)$-oracle-aided converter by embedding it.[8]

Notational Conventions.

1. Interaction: Given a distinguisher D, an n-oracle-aided converter C, and n resources $\vec{S} = (S_1, \ldots, S_n)$, we denote the execution of D and $C^{\vec{S}}$ using $D(C^{\vec{S}})$ rather than $D^{C^{\vec{S}}}$ for notational convenience.

2. Terminology: We may refer to a resource simply as a system, and to an n-oracle-aided converter simply as an n-oracle-aided system.

3. Alphabets: In the above, it is possible to consider distinct input and output spaces for every resource, or even for every individual node in the tree. However, for the sake of simplicity, we choose to consider a single universal input space $\mathcal{X}$ and output space $\mathcal{Y}$, which can be viewed as the union of all sub-spaces. Furthermore, throughout most of the paper, we do not explicitly mention the sets $(\mathcal{X}, \mathcal{Y})$ and also omit the upper bound h on the number of rounds (when irrelevant).

4. Convex Combinations: Given systems S_0, S_1 and $p \in [0, 1]$, we denote by $(1-p)S_0 + pS_1$ the system equivalent to sampling $b \leftarrow \mathrm{Ber}(p)$ and using S_b.[9] Furthermore, for a set $\mathcal{X}$ and given

[8] See Section C for more details.

[9] Equivalently, we can view a system as a consistent weight assignment over all partial transcripts (nodes) by taking the multiplication of probabilities along the path to the root. Then we simply take the point-wise convex combination over these weights.

an ensemble of systems $\{S_x\}_{x\in\mathcal{X}}$ with a random variable X over $\mathcal{X}$, we denote by S_X the system equivalent to sampling $x \leftarrow X$ and using S_x.

5. Implementation: For a resource S, we define size(S) as the size of the smallest circuit realizing it. We allow circuits to receive advice sampled from arbitrary distributions, as we deal with indistinguishability where the best such advice can later be fixed. For a converter $C^{(\cdot)}$, we define size$(C^{(\cdot)})$ as the size of its oracle-aided circuit. We define the composed size size$(C^{\vec{S}})$ as the size of the circuit obtained by replacing the oracle gates in $C^{(\cdot)}$ with the circuits realizing $\vec{S}$, capturing the actual cost of the combined implementation rather than the minimal circuit size of the resulting resource.

   We may sometimes abuse notation and associate an implementation with the system it realizes, feeding it to a distinguisher or a converter.

Definition 5 (Statistical Distance of Systems)**.** For $\delta \in [0,1]$, we say that two systems R and I are δ statistically close, denoted $R \overset{\mathsf{s}}{\approx}_\delta I$, if for every (unbounded adaptive) distinguisher D we have

$$\left|\mathbb{E}\left[D^R - D^I\right]\right| \leq \delta\,.$$

If $\delta = 0$, we write $R \equiv I$.

Lemma 2 (System Decomposition Lemma, Reinterpretation of [LM20, Theorem 2])**.** *For any finite discrete systems R and I with statistical distance $\delta \in [0,1]$, there exist systems S, R', I' such that*

$$R \equiv (1-\delta)S + \delta R'\,,$$
$$I \equiv (1-\delta)S + \delta I'\,.$$

In [LM20], randomized systems are modeled as probability distributions over deterministic systems. To be more accurate, since this representation is not unique, randomized systems are in fact seen as equivalence classes over these distributions. It is shown that the maximal distinguishing advantage between two random systems is equal to the minimal statistical distance between distributions in their equivalence classes. Then, standard coupling of probability distributions implies the above.

3.2 Indistinguishability Combiners

Definition 6 (Indistinguishability Combiner)**.** For finite sets $(\mathcal{X},\mathcal{Y})$, $\varepsilon \in [0,1]$, $k,n \in \mathbb{N}$ with $k \leq n$, an (ε,k,n) indistinguishability combiner with respect to ideal $(\mathcal{Y},\mathcal{X})$-systems $\vec{I} = (I_1,\ldots,I_n)$ against an $(\mathcal{X},\mathcal{Y})$-distinguisher class $\mathcal{D}$, is a pair of n-oracle-aided $(\mathcal{X},\mathcal{Y})$-systems $(C_0^{(\cdot)}, C_1^{(\cdot)})$ such that for all $(\mathcal{Y},\mathcal{X})$-systems $\vec{S} = (S_1,\ldots,S_n)$:

$$|\{i \in [n] \mid S_i \equiv I_i\}| \geq k \implies \forall D \in \mathcal{D}: \left|\mathbb{E}\left[D(C_0^{\vec{S}}) - D(C_1^{\vec{S}})\right]\right| \leq \varepsilon\,.$$

For $s \in \mathbb{N}$, an (s,ε,k,n) indistinguishability combiner is an (ε,k,n) indistinguishability combiner against the distinguisher class $\mathcal{D}$ induced by s-sized oracle-aided circuits. If $\mathcal{D}$ consists of all (unbounded adaptive) distinguishers, we say the combiner is *statistically* secure. Furthermore, if $\varepsilon = 0$, we say the combiner is *perfect*.

Remark 1. While the above definition may seem too restrictive at first since it targets a specific choice for the ideal systems $\vec{I}$, typically the same combiner $(C_0^{(\cdot)}, C_1^{(\cdot)})$ satisfies it with respect to any ideal $\vec{I}$ in some "good class" of systems.

Ensembles of Indistinguishability Combiners. The definition naturally extends to the setting of ensembles where for each $\lambda \in \mathbb{N}$ we have $(\mathcal{X}_\lambda, \mathcal{Y}_\lambda)$, $\varepsilon(\lambda) \in [0,1]$, $k(\lambda), n(\lambda) \in \mathbb{N}$, $\vec{I}_\lambda = (I_{\lambda,1}, \ldots, I_{\lambda,n(\lambda)})$, $\mathcal{D}_\lambda$ and $n(\lambda)$-oracle-aided systems $(C^{(\cdot)}_{\lambda,0}, C^{(\cdot)}_{\lambda,1})$. Accordingly, an $(s(\lambda), \varepsilon(\lambda), k(\lambda), n(\lambda))$ indistinguishability combiner corresponds to the setting where the distinguisher class $\mathcal{D}_\lambda$ consists of all distinguishers implementable by $s(\lambda)$-sized oracle-aided circuits. A combiner that satisfies this for all choices of $s(\lambda)$ (resp., all polynomials $s(\lambda)$) is denoted by $(\infty, \varepsilon(\lambda), k(\lambda), n(\lambda))$-combiner (resp., $(\mathrm{poly}(\lambda), \varepsilon(\lambda), k(\lambda), n(\lambda))$-combiner) and is referred to as *statistical* (resp., *computational*). We may omit the error $\varepsilon(\lambda)$ when it is negligible. The notion of a *perfect* combiner remains unchanged, i.e., corresponds to unbounded $s(\lambda)$ and $\varepsilon(\lambda) \equiv 0$.

4 The Statistically Close Oracle Model

In this section, we show that any indistinguishability combiner is inherently an amplifier for statistically weak candidates. We emphasize that, although the candidate systems are required to be statistically close to ideal, the theorem applies even when the combiner itself only guarantees indistinguishability against a computationally bounded class of distinguishers.

Theorem 1 (Indistinguishability Combiners Are Amplifiers)**.** *For $\varepsilon \in [0,1]$, $\delta \in (0, 0.5)$, $k, n \in \mathbb{N}$, given an (ε, k, n) indistinguishability combiner $(C^{(\cdot)}_0, C^{(\cdot)}_1)$ with respect to ideal systems $\vec{I} = (I_1, \ldots, I_n)$ against a distinguisher class $\mathcal{D}$, let $\vec{R} = (R_1, \ldots, R_n)$ be a tuple of n real systems such that $\forall i \in [n] : R_i \overset{\mathsf{s}}{\approx}_\delta I_i$. Then, we have*

$$\forall D \in \mathcal{D} : \left| \mathbb{E}\left[D(C^{\vec{R}}_0) - D(C^{\vec{R}}_1) \right] \right| \leq \delta' + \varepsilon' \,, \tag{2}$$

where

$$\delta' := (2\delta)^{n-k+1} \cdot \sum_{j=1}^{k} \binom{n}{k-j} (-\delta)^{j-1} \leq \binom{n}{k-1} (2\delta)^{n-k+1} \,,$$

$$\varepsilon' := \varepsilon \cdot \sum_{i=0}^{n-k} \binom{i+k-1}{i} (2\delta)^i \leq \varepsilon \cdot \min\left(\binom{n}{k}, (1-2\delta)^{-k} \right) .$$

Proof. Let us denote by $\Delta(k,n) = \Delta_{\varepsilon,\delta}(k,n)$ the worst-case upper bound on Eq. (2). That is, for every (ε, k, n) combiner with respect to some ideal systems and distinguisher class, and for any real systems such that each is δ-close statistically to ideal, $\Delta(k,n)$ bounds the distinguishing advantage of said class when the real systems are used. Our proof proceeds by deriving a recurrence relation for $\Delta(k,n)$ and solving it combinatorially using the boundary conditions $k = 0$ and $k = n$.

Fix any distinguisher $D \in \mathcal{D}$, and let $\Delta^D C^{\vec{R}} := \mathbb{E}\left[D(C^{\vec{R}}_0) - D(C^{\vec{R}}_1) \right]$. By the system decomposition lemma (Lemma 2), for all $i \in [n]$, there exist systems S_i, R'_i, I'_i such that

$$R_i \equiv (1-\delta) S_i + \delta R'_i \,,$$
$$I_i \equiv (1-\delta) S_i + \delta I'_i \,.$$

Then, we have

$$\Delta^D C^{\vec{R}} = (1-\delta) \Delta^D C^{\vec{R}_{n-1}, S_n} + \delta \Delta^D C^{\vec{R}_{n-1}, R'_n} \,,$$
$$\Delta^D C^{\vec{R}_{n-1}, I_n} = (1-\delta) \Delta^D C^{\vec{R}_{n-1}, S_n} + \delta \Delta^D C^{\vec{R}_{n-1}, I'_n} \,,$$

which together imply that

$$\Delta^D C^{\vec{R}} = \Delta^D C^{\vec{R}_{n-1},I_n} + \delta\Delta^D C^{\vec{R}_{n-1},R'_n} - \delta\Delta^D C^{\vec{R}_{n-1},I'_n}\,,$$
$$\left|\Delta^D C^{\vec{R}}\right| \le \left|\Delta^D C^{\vec{R}_{n-1},I_n}\right| + \delta\left(\left|\Delta^D C^{\vec{R}_{n-1},R'_n}\right| + \left|\Delta^D C^{\vec{R}_{n-1},I'_n}\right|\right)\,.$$

Observe that by fixing the last system to ideal I_n we obtain an $(\varepsilon,\ k{-}1,\ n{-}1)$ combiner with respect to $\vec{I}_{n-1}$ against the same distinguisher class, and for $\vec{R}_{n-1}$ we still have that $\forall i \in [n-1] : R_i \overset{\mathsf{s}}{\approx}_\delta I_i$. Similarly, fixing the last system to R'_n or I'_n results in an $(\varepsilon,\ k,\ n-1)$ combiner. Hence, we derive the recurrence relation

$$\Delta(k,n) \le \Delta(k-1,n-1) + 2\delta\cdot\Delta(k,n-1)\,.$$

For the base cases: (1) $\Delta(0,n) \le \varepsilon$ since enough ideal systems are used and we can invoke the combiner's guarantee. (2) $\Delta(n,n) \le 2(1-(1-\delta)^n) + \varepsilon$ by using a standard product bound $\vec{R} \overset{\mathsf{s}}{\approx}_{1-(1-\delta)^n} \vec{I}$ since with probability $(1-\delta)^n$ we get the shared system $\vec{S}$, and thus

$$C_0^{\vec{R}} \overset{\mathsf{s}}{\approx}_{1-(1-\delta)^n} C_0^{\vec{I}} \overset{\mathsf{s}}{\approx}_{\varepsilon} C_1^{\vec{I}} \overset{\mathsf{s}}{\approx}_{1-(1-\delta)^n} C_1^{\vec{R}}\,.$$

Structurally, it is more convenient to separate the base cases into a main term and an error term. Specifically, we define:

$$\begin{aligned}
&\delta'(0,n) := 0\,, && \varepsilon'(0,n) := \varepsilon\,,\\
&\delta'(n,n) := 2(1-(1-\delta)^n)\,, && \varepsilon'(n,n) := \varepsilon\,,\\
&\delta'(k,n) := \delta'(k-1,n-1) + 2\delta\cdot\delta'(k,n-1)\,, && \varepsilon'(k,n) := \varepsilon'(k-1,n-1) + 2\delta\cdot\varepsilon'(k,n-1)\,.
\end{aligned}$$

Since the sum $\delta'(k,n)+\varepsilon'(k,n)$ bounds the base cases and exactly matches the recurrence relation, we obtain the upper bound $\Delta(k,n) \le \delta'(k,n) + \varepsilon'(k,n)$.

The simpler (and somewhat looser) upper bounds of

$$\delta'(k,n) \le \binom{n}{k-1}(2\delta)^{n-k+1} \qquad \text{and} \qquad \varepsilon'(k,n) \le \binom{n}{k}\varepsilon\,,$$

are immediately derived using Pascal's rule $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ with $2\delta \le 1$, noting for $\delta'(n,n)$ that $1-(1-\delta)^n \le n\delta$ (union bound). For the second simplified bound of $\varepsilon'(k,n) \le \varepsilon\cdot(1-2\delta)^{-k}$, we observe that

$$(1-2\delta)^{-(k-1)} + 2\delta(1-2\delta)^{-k} = (1-2\delta)^{-k}(1-2\delta+2\delta) = (1-2\delta)^{-k}\,.$$

We now proceed to derive the exact closed-form solutions for both δ' and ε'. We begin by solving the recurrence for $\delta'(k,n)$ where $0 < k < n$. This is naturally modeled as a combinatorial problem over weighted lattice paths on $\mathbb{N}^2$, where the recursive sequence corresponds to directed downward walks along the grid. Since the base case $\delta'(0,n)$ is zero, it suffices to count paths from (k,n) to (i,i) for $i \in [k]$, using only steps $(0,-1)$ ("down") and $(-1,-1)$ ("down-left"), and avoiding all points (j,j) with $j > i$, since the recursion stops upon first reaching the diagonal $k = n$ (i.e., a point (j,j)). (Note that we do not need to actively avoid the other base case $(0,j)$, since it is impossible to pass through on the way to (i,i).) This is equivalent to counting paths from (k,n) to $(i,i+1)$, since the final step must be a down step (to avoid $(i+1,i+1)$) and the quantity $n-k$ is non-increasing along any valid path so we also avoid any other (j,j). Finally, each down step

contributes a factor 2δ and the total number of down steps from (k,n) to (i,i) is $n-k$, so any such path accumulates weight $(2\delta)^{n-k}$. The resulting expression is

$$\begin{aligned}
\delta'(k,n) &= \sum_{i=1}^{k} \binom{n-(i+1)}{k-i} (2\delta)^{n-k} \cdot \delta'(i,i) && (3)\\
&= (2\delta)^{n-k} \sum_{i=1}^{k} \binom{n-i-1}{k-i} \cdot 2(1-(1-\delta)^i) && (4)\\
&= (2\delta)^{n-k} \sum_{i=1}^{k} \binom{n-i-1}{k-i} \cdot (-2) \sum_{j=1}^{i} \binom{i}{j} (-\delta)^j && (5)\\
&= (2\delta)^{n-k+1} \sum_{i=1}^{k} \binom{n-i-1}{k-i} \sum_{j=1}^{i} \binom{i}{j} (-\delta)^{j-1} && (6)\\
&= (2\delta)^{n-k+1} \sum_{j=1}^{k} (-\delta)^{j-1} \sum_{i=j}^{k} \binom{n-i-1}{k-i} \binom{i}{j} && (7)\\
&= (2\delta)^{n-k+1} \sum_{j=1}^{k} \binom{n}{k-j} (-\delta)^{j-1}, && (8)
\end{aligned}$$

where in Eq. (5) we used the binomial theorem $(1-\delta)^i - 1 = \sum_{j=1}^{i} \binom{i}{j}(-\delta)^j$; in Eq. (7) we changed the order of summation from $i \in [k], j \le i$ to $j \in [k], i \ge j$; and finally in the last equation we used the Chu–Vandermonde identity (Lemma 1) with $j \le n-k-1+j \le n-1$:

$$\sum_{i=j}^{k} \binom{n-i-1}{k-i} \binom{i}{j} = \sum_{i=j}^{k} \binom{n-i-1}{n-k-1} \binom{i}{j} = \binom{n}{n-k+j} = \binom{n}{k-j}.$$

Next, to solve $\varepsilon'(k,n)$ for $0 < k < n$, our approach is similar but now we also need to take into account the non-zero $\varepsilon'(0,i)$ base case (relevant range $i \in [n-k]$). We cannot pass through (j,j) on the way to $(0,i)$ as the difference $n-k$ is non-increasing, so we only need to avoid passing through $(0,j)$ for $j > i$. Hence, we count the number of paths to $(1,i+1)$, with the last step necessarily being down-left. Each down step contributes a factor 2δ and the total number of down steps from (k,n) to $(0,i)$ is $n-k-i$, so any such path accumulates weight $(2\delta)^{n-k-i}$. The resulting expression is

$$\begin{aligned}
\varepsilon'(k,n) &= \sum_{i=1}^{k} \binom{n-i-1}{k-i} (2\delta)^{n-k} \cdot \varepsilon'(i,i) + \sum_{i=1}^{n-k} \binom{n-i-1}{k-1} (2\delta)^{n-k-i} \cdot \varepsilon'(0,i)\\
&= \varepsilon \left((2\delta)^{n-k} \sum_{i=1}^{k} \binom{n-i-1}{n-k-1} + \sum_{i=1}^{n-k} \binom{n-i-1}{n-k-i} (2\delta)^{n-k-i} \right)\\
&= \varepsilon \left((2\delta)^{n-k} \sum_{i=n-k-1}^{n-2} \binom{i}{n-k-1} + \sum_{i=0}^{n-k-1} \binom{i+k-1}{i} (2\delta)^i \right)\\
&= \varepsilon \left((2\delta)^{n-k} \binom{n-1}{n-k} + \sum_{i=0}^{n-k-1} \binom{i+k-1}{i} (2\delta)^i \right)\\
&= \varepsilon \sum_{i=0}^{n-k} \binom{i+k-1}{i} (2\delta)^i,
\end{aligned}$$

where in the second to last equation we used the hockey-stick identity

$$\sum_{i=r}^{n} \binom{i}{r} = \binom{n+1}{r+1}.$$

This completes the proof of the theorem. □

The above theorem extends naturally to ensembles by applying it to every $\lambda \in \mathbb{N}$.

Corollary 1 (Indistinguishability Combiners Are Amplifiers; Ensemble Version)**.** *For functions $\varepsilon : \mathbb{N} \to [0,1]$, $\delta : \mathbb{N} \to (0, 0.5)$ and $k, n : \mathbb{N} \to \mathbb{N}$, given an $(\varepsilon(\lambda), k(\lambda), n(\lambda))$ indistinguishability combiner $\left\{C^{(\cdot)}_{\lambda,0}, C^{(\cdot)}_{\lambda,1}\right\}_\lambda$ with respect to ideal systems $\left\{\vec{I}_\lambda = (I_{\lambda,1}, \ldots, I_{\lambda,n(\lambda)})\right\}_\lambda$ against a distinguisher class $\{\mathcal{D}_\lambda\}_\lambda$, let $\left\{\vec{R}_\lambda = (R_{\lambda,1}, \ldots, R_{\lambda,n(\lambda)})\right\}_\lambda$ be a tuple of $n(\lambda)$ real systems such that for all $\lambda \in \mathbb{N}$ and all $i \in [n(\lambda)]$, it holds that $R_{\lambda,i} \overset{\mathsf{s}}{\approx}_{\delta(\lambda)} I_{\lambda,i}$. Then, we have*

$$\forall \lambda \in \mathbb{N},\ \forall D \in \mathcal{D}_\lambda :\ \left|\mathbb{E}\left[D(C^{\vec{R}_\lambda}_{\lambda,0}) - D(C^{\vec{R}_\lambda}_{\lambda,1})\right]\right| \leq \delta'(\lambda) + \varepsilon'(\lambda)\,,$$

where

$$\delta'(\lambda) := (2\delta(\lambda))^{n(\lambda)-k(\lambda)+1} \cdot \sum_{j=1}^{k(\lambda)} \binom{n(\lambda)}{k(\lambda)-j} (-\delta(\lambda))^{j-1}\,,$$

$$\varepsilon'(\lambda) := \varepsilon(\lambda) \cdot \sum_{i=0}^{n(\lambda)-k(\lambda)} \binom{i+k(\lambda)-1}{i} (2\delta(\lambda))^i\,.$$

Inherent Inefficiency. Taking a closer look at the proof of Theorem 1, we observe that even if both R and I are efficiently implementable systems, there is no guarantee on the efficiency of R' and I' given by the decomposition lemma. What we can guarantee is that R' and I' are "explainable" by R and I, i.e., every transcript is in the support of the original systems (in fact, they are even δ-dense). As a result, the theorem strongly relies on the fact that the combiner properly works even when the given candidates are *inefficient*, and we cannot relax the notion of combiners to work only with respect to efficient candidates.

5 The Computationally Close Oracle Model

In this section, we generalize the combiner-is-amplifier theorem to the computational setting. To do so, we restrict the underlying real and ideal candidates to be single-round systems. That is, the combiner makes at most one query to each candidate oracle over the entire execution. Formally, this is captured by switching from general interactive systems to randomized functions (over finite domain and codomain), where any subsequent queries to an oracle are answered with $\perp$. Importantly, since the ideal system is modeled as a randomized function, the notion is limited to non-adaptive cryptographic games. For a detailed discussion of these limitations, see Remark 4.

As noted in the introduction, one can emulate the case of multiple non-adaptive queries by a single-query combiner with batch queries (potentially increasing the indistinguishability error). Extensions to the adaptive setting are discussed in Section 5.3.

Definition 7 (Single-Query Indistinguishability Combiner)**.** For finite sets $(\mathcal{X}, \mathcal{Y})$ and randomized functions $F_1^*, \ldots, F_n^* : \mathcal{X} \to \mathcal{Y}$, we say that $(C_0^{(\cdot)}, C_1^{(\cdot)})$ is a single-query indistinguishability combiner with respect to $\vec{F^*}$ if it is an indistinguishability combiner with respect to $\vec{I} = (I_1, \ldots, I_n)$, where I_i is the single-round system induced by F_i^*.

5.1 Useful Lemmas

First, we introduce some lemmas that are useful in generalizing statistical bounds to the computational setting (regardless of combiners).

Lemma 3 (Hybrid Indistinguishability Lemma [MT10, BG24])**.** *Let X_0 and X_1 be random variables, and let $\delta, \varepsilon \in (0,1)$ and $s \in \mathbb{N}$. Define $s' := \frac{s \cdot \varepsilon^2}{128(\log |\operatorname{Im}(X_0)| + \log |\operatorname{Im}(X_1)| + 1)}$. Assume that*

$$X_0 \overset{\mathsf{c}}{\approx}_{s,\delta} X_1.$$

Then there exists a hybrid distribution X' such that

$$X_0 \overset{\mathsf{c}}{\approx}_{s',\varepsilon(1-\delta)} X' \overset{\mathsf{s}}{\approx}_{\delta} X_1 .$$

Note that the first indistinguishability is computational and the second is statistical. The above lemma generalizes to randomized functions F_0 and F_1 where indistinguishability is guaranteed for every input choice x.

Definition 8. Let F_0 and F_1 be two randomized functions from $\mathcal{X}$ to $\mathcal{Y}$. We say that F_0 and F_1 are ε-indistinguishable against s-sized circuits, and write

$$F_0 \overset{\mathsf{c}}{\approx}_{s,\varepsilon} F_1 ,$$

if for every $x \in \mathcal{X}$ we have

$$F_0(x) \overset{\mathsf{c}}{\approx}_{s,\varepsilon} F_1(x) .$$

We say that F_0 and F_1 are ε-indistinguishable statistically, and write

$$F_0 \overset{\mathsf{s}}{\approx}_{\varepsilon} F_1 ,$$

if for every $x \in \mathcal{X}$ we have

$$F_0(x) \overset{\mathsf{s}}{\approx}_{\varepsilon} F_1(x) .$$

In order to obtain the generalized HIL version, we simply apply the random-variable version on every possible input.

Lemma 4 (Generalized Hybrid Indistinguishability Lemma)**.** *Let F_0 and F_1 be two randomized functions from $\mathcal{X}$ to $\mathcal{Y}$, and let $\delta, \varepsilon \in (0,1)$ and $s \in \mathbb{N}$. Define $s' := \frac{s \cdot \varepsilon^2}{128(\log|\operatorname{Im}(F_0)| + \log|\operatorname{Im}(F_1)| + 1)}$. Assume that*

$$F_0 \overset{\mathsf{c}}{\approx}_{s,\delta} F_1.$$

Then there exists a hybrid randomized function F' such that

$$F_0 \overset{\mathsf{c}}{\approx}_{s',\varepsilon(1-\delta)} F' \overset{\mathsf{s}}{\approx}_{\delta} F_1.$$

Again, the first indistinguishability is computational and the second is statistical.

Remark 2. Note that F' may not be efficiently samplable, even if F_0 and F_1 are. In fact, we may not even be able to efficiently store a sample $f' \leftarrow F'$ using n.u. advice (unless $|Q|$ is small enough).

Proof. For every $x \in \mathcal{X}$, we have that $F_0(x) \stackrel{\mathsf{c}}{\approx}_{s,\delta} F_1(x)$, so we can apply Lemma 3 to get a hybrid r.v. F'_x such that

$$F_0(x) \stackrel{\mathsf{c}}{\approx}_{s',\varepsilon(1-\delta)} F'_x \stackrel{\mathsf{s}}{\approx}_{\delta} F_1(x)\,.$$

Then, we can define a (randomized inefficient) function $F' : \mathcal{X} \to \mathcal{Y}$ that, given x, samples from F'_x. We immediately get that $F_0 \stackrel{\mathsf{c}}{\approx}_{s',\varepsilon(1-\delta)} F'$ because $\forall x \in \mathcal{X} :\ F_0(x) \stackrel{\mathsf{c}}{\approx}_{s',\varepsilon(1-\delta)} F'(x)$, and $F_1 \stackrel{\mathsf{s}}{\approx}_{\delta} F'$ because $\forall x \in \mathcal{X} :\ F_1(x) \stackrel{\mathsf{s}}{\approx}_{\delta} F'(x)$. □

Finally, we show that the hybrid argument generalizes to the direct product of n randomized functions $\vec{F}$ and $\vec{F}'$, which is non-trivial because we allow for inefficient functionalities. Specifically, it is sufficient that only one of $\vec{F}, \vec{F}'$ is efficient, as we may use hardcoded samples for the other. We remark that even if both are inefficient we could still prove a weaker version of the lemma, under the additional assumption that the distinguisher is non-adaptive—that is, the queries $(q_1, \ldots, q_n)$ are chosen in advance and do not depend on responses from the functions.

Lemma 5 (Single-Query Hybrid Argument)**.** *Let $s, n \in \mathbb{N}$ and $\varepsilon \in (0,1)$ be given, and let $\vec{F} = (F_1, \ldots, F_n)$ and $\vec{F}' = (F'_1, \ldots, F'_n)$ be tuples (direct product) of n randomized functions from $\mathcal{X}$ to $\mathcal{Y}$, such that*

$$\forall i \in [n] :\ F_i \stackrel{\mathsf{c}}{\approx}_{s,\varepsilon} F'_i\,.$$

Then, for every s'-sized distinguisher D that makes at most one query to each oracle, we have

$$\left|\mathbb{E}\left[D^{\vec{F}} - D^{\vec{F}'}\right]\right| \leq \varepsilon n\,,$$

for

$$s' := s - \min \mathrm{size}(\vec{F}, \vec{F}')\,.$$

Proof. Assume without loss of generality that $\mathrm{size}(\vec{F}) \leq \mathrm{size}(\vec{F}')$ and that D is deterministic. We proceed by induction on n. The base case $n = 1$ is immediate. For the inductive step, let us consider the execution of $D^{\vec{F}}$. Let $i_1 \in [n]$ be the index of the initial function queried, yielding query q_{i_1} and (random) response a_{i_1}. Let $V := [n] \setminus \{i_1\}$ denote the remaining indices. If we switch the i_1-th function to F'_{i_1}, the expectation changes by at most ε:

$$\left|\mathbb{E}\left[D^{\vec{F}} - D^{\vec{F}_V, F'_{i_1}}\right]\right| \leq \varepsilon\,,$$

since otherwise we can distinguish between $F_{i_1}(q_{i_1})$ and $F'_{i_1}(q_{i_1})$ by running the distinguisher with $\vec{F}_V$. Next, for every fixed response to the first query, we can use the induction hypothesis on $\vec{F}_V$ and $\vec{F}'_V$ to get

$$\left|\mathbb{E}\left[D^{\vec{F}_V, a_{i_1}} - D^{\vec{F}'_V, a_{i_1}}\right]\right| \leq \varepsilon(n-1)\,,$$

since the size of D with the hardcoding of a_{i_1} is

$$s' + \mathrm{len}(\mathcal{Y}) \leq s' + \mathrm{size}(F_{i_1}) \leq s - \mathrm{size}(\vec{F}_V)\,.$$

Seeing that the above holds for every fixed a_{i_1}, it must also hold for the convex combination $F'_{i_1}(q_{i_1})$; then we have

$$\left|\mathbb{E}\left[D^{\vec{F}_V,F'_{i_1}} - D^{\vec{F}'}\right]\right| \leq \varepsilon(n-1)\,.$$

We conclude with the triangle inequality

$$\left|\mathbb{E}\left[D^{\vec{F}} - D^{\vec{F}'}\right]\right| \leq \left|\mathbb{E}\left[D^{\vec{F}} - D^{\vec{F}_V,F'_{i_1}}\right]\right| + \left|\mathbb{E}\left[D^{\vec{F}_V,F'_{i_1}} - D^{\vec{F}'}\right]\right| \leq \varepsilon n\,,$$

completing the proof of the lemma. □

5.2 The Amplification Theorem

Below, by $\max_{b\in\{0,1\}} \mathrm{size}(C_b^{\vec{F}})$, we denote the circuit size needed to implement the oracle-aided $C_0^{(\cdot)}, C_1^{(\cdot)}$ plus the implementation cost of the oracles $\vec{F}$ themselves. Also, we denote by γ the computational slackness parameter—the closer we want to get to the statistical bound, the more we pay in circuit size.

Theorem 2 (Single-Query Indistinguishability Combiners Are Computational Amplifiers)**.** *For $\varepsilon,\gamma\in[0,1]$, $\delta\in(0,0.5)$, $k,n,s_c,s_f\in\mathbb{N}$, given a single-query (s_c,ε,k,n) indistinguishability combiner $(C_0^{(\cdot)},C_1^{(\cdot)})$ with respect to ideal functions $\vec{F}^* = (F_1^*,\ldots,F_n^*)$, let $\vec{F} = (F_1,\ldots,F_n)$ be a tuple of n real functions such that $\forall i\in[n]: F_i \overset{\mathsf{c}}{\approx}_{s_f,\delta} F_i^*$. Then for every s-sized distinguisher D, we have*

$$\left|\mathbb{E}\left[D(C_0^{\vec{F}}) - D(C_1^{\vec{F}})\right]\right| \leq \delta' + \varepsilon' + 2n\gamma\,,$$

where

$$\begin{aligned}
\delta' &:= (2\delta)^{n-k+1}\cdot\sum_{j=1}^{k}\binom{n}{k-j}(-\delta)^{j-1} \leq \binom{n}{k-1}(2\delta)^{n-k+1}\,,\\
\varepsilon' &:= \varepsilon\cdot\sum_{i=0}^{n-k}\binom{i+k-1}{i}(2\delta)^i \leq \varepsilon\cdot\min\left(\binom{n}{k},\,(1-2\delta)^{-k}\right)\,,\\
s' &:= \frac{s_f\cdot\gamma^2}{128(\max_i(\log|\mathrm{Im}(F_i)| + \log|\mathrm{Im}(F_i^*)|)+1)}\,,\\
s &:= \min\left(s_c,\; s' - \max_{b\in\{0,1\}}\mathrm{size}(C_b^{\vec{F}})\right)\,.
\end{aligned}$$

Proof. Let $\vec{F}' = (F'_1,\ldots,F'_n)$ be the hybrid functions guaranteed by the generalized hybrid indistinguishability lemma (Lemma 4) such that $\forall i\in[n]:\ F_i \overset{\mathsf{c}}{\approx}_{s',\gamma} F'_i \overset{\mathsf{s}}{\approx}_{\delta} F_i^*$. Next, using the triangle

inequality and the single-query hybrid argument (Lemma 5) we get that

$$\begin{aligned}
&\left|\mathbb{E}\left[D(C_0^{\vec{F}}) - D(C_1^{\vec{F}})\right]\right| \leq \\
&\left|\mathbb{E}\left[D(C_0^{\vec{F}'}) - D(C_1^{\vec{F}'})\right]\right| + \\
&\left|\mathbb{E}\left[D(C_0^{\vec{F}}) - D(C_0^{\vec{F}'})\right]\right| + \\
&\left|\mathbb{E}\left[D(C_1^{\vec{F}}) - D(C_1^{\vec{F}'})\right]\right| \leq \\
&\left|\mathbb{E}\left[D(C_0^{\vec{F}'}) - D(C_1^{\vec{F}'})\right]\right| + 2 \cdot n\gamma\,,
\end{aligned}$$

since the induced distinguisher between $\vec{F}$ and $\vec{F}'$ using either $C_0^{(\cdot)}$ or $C_1^{(\cdot)}$ is small enough:

$$s + \max_{b\in\{0,1\}} \text{size}(C_b^{\vec{F}}) \leq s'\,.$$

Finally, using that $\forall i \in [n] : F'_i \overset{\mathsf{s}}{\approx}_\delta F^*_i$, we apply the result for statistically close oracles (Theorem 1):

$$\left|\mathbb{E}\left[D(C_0^{\vec{F}'}) - D(C_1^{\vec{F}'})\right]\right| \leq \delta' + \varepsilon'\,.$$

That is, we view $\vec{F}'$ and $\vec{F}^*$ as the real and ideal single-round systems (all further queries answered with $\perp$), and $\mathcal{D}$ is the class of $s \leq s_c$-sized distinguishers. □

We may use the above theorem to derive, in a straightforward fashion, a corollary focusing on the setting of efficient (polynomial-size) combiner and functions, with indistinguishability against polynomial-size adversaries and negligible slackness.

Corollary 2. *For functions $\varepsilon : \mathbb{N} \to [0,1]$, $\delta : \mathbb{N} \to (0, 0.5)$ and polynomially bounded $k, n : \mathbb{N} \to \mathbb{N}$, given an efficient single-query $(\text{poly}, \varepsilon, k, n)$ indistinguishability combiner $\left\{(C_0^{(\cdot)}, C_1^{(\cdot)})\right\}_\lambda$ with respect to ideal functions $\left\{\vec{F}^* = (F_1^*, \ldots, F_{n(\lambda)}^*)\right\}_\lambda$ with polynomially bounded output length, let $\left\{\vec{F} = (F_1, \ldots, F_{n(\lambda)})\right\}_\lambda$ be some efficient real functions such that for every sequence of indices $\{i_\lambda \in [n(\lambda)]\}_\lambda$, it holds that $F_{i_\lambda} \overset{\mathsf{c}}{\approx}_{\delta(\lambda)} F^*_{i_\lambda}$. Then, for every polynomial-size distinguisher $\{D\}_\lambda$ and all large enough $\lambda \in \mathbb{N}$, we have*

$$\left|\mathbb{E}\left[D(C_0^{\vec{F}}) - D(C_1^{\vec{F}})\right]\right| \leq \delta' + \varepsilon' + \text{negl}\,,$$

where

$$\begin{aligned}
\delta' &:= (2\delta)^{n-k+1} \cdot \sum_{j=1}^{k} \binom{n}{k-j} (-\delta)^{j-1} \leq \binom{n}{k-1} (2\delta)^{n-k+1}\,, \\
\varepsilon' &:= \varepsilon \cdot \sum_{i=0}^{n-k} \binom{i+k-1}{i} (2\delta)^i \leq \varepsilon \cdot \min\left(\binom{n}{k}, (1-2\delta)^{-k}\right)\,.
\end{aligned}$$

Proof. Let $\{D_\lambda\}_\lambda$ be a distinguisher ensemble of size $s(\lambda) \le \mathrm{poly}(\lambda)$, and let $\mu(\lambda)$ be an arbitrarily small inverse polynomial. For each $\lambda \in \mathbb{N}$, define the slackness parameter $\gamma(\lambda) := \mu(\lambda)/2n(\lambda)$, $s_c(\lambda) := s(\lambda)$, and

$$s_f(\lambda) := \frac{\left(s(\lambda) + \max_{b\in\{0,1\}} \mathrm{size}(C^{\vec{F}_\lambda}_{\lambda,b})\right) \cdot 128 \left(\max_{i\in[n(\lambda)]} \left(\log|\mathrm{Im}(F_{\lambda,i})| + \log\left|\mathrm{Im}(F^*_{\lambda,i})\right|\right) + 1\right)}{\gamma(\lambda)^2}.$$

By our assumptions, both $s_c(\lambda)$ and $s_f(\lambda)$ are polynomially bounded. Hence, for all large enough $\lambda \in \mathbb{N}$: our combiner is $(s_c(\lambda), \varepsilon(\lambda), k(\lambda), n(\lambda))$ secure, and for every $i \in [n(\lambda)]$, we have $F_{\lambda,i} \overset{\mathsf{c}}{\approx}_{s_f(\lambda),\delta(\lambda)} F^*_{\lambda,i}$.

Applying Theorem 2, we deduce that for all large enough λ,

$$\left|\mathbb{E}\left[D_\lambda(C^{\vec{F}_\lambda}_{\lambda,0}) - D_\lambda(C^{\vec{F}_\lambda}_{\lambda,1})\right]\right| \le \delta'(\lambda) + \varepsilon'(\lambda) + \mu(\lambda).$$

Since this bound holds for any arbitrarily small inverse polynomial $\mu(\lambda)$, the additional error is bounded by a negligible function $\mathrm{negl}(\lambda)$. □

Remark 3 (How to Use Theorem 2)**.** Generally speaking, we envision applying the above theorem as follows. First, we construct an oracle-aided amplifier Amp^O. In the analysis, we then fix an arbitrary weak candidate O and a distinguisher D for a given security property i (there may be multiple such security games). Next, we define a combiner $(C^{(\cdot)}_0, C^{(\cdot)}_1)$, along with real and ideal systems $\vec{R}, \vec{I}$ and a distinguisher D', all of which may depend on (O, D, i). These are chosen so that the computational indistinguishability between the real and ideal systems is closely tied to the weakness of O, while the advantage of D' in distinguishing $C^{\vec{R}}_0$ from $C^{\vec{R}}_1$ closely reflects the ability of D to win the i-th security game against Amp^O. In particular, the combiner need not be specified a priori; rather, it is constructed as part of the analysis and may be tailored to the specific candidate, security game, and adversary under consideration.

We present some examples where our Theorem 2 could be used, see Examples 1 to 4.

Example 2 (XOR Lemma)**.** First, Yao's XOR lemma: let (B, Z) be jointly distributed over $\{0,1\} \times \{0,1\}^\ell$ such that B is δ-unpredictable given side information Z. Equivalently, $(B, Z) \overset{\mathsf{c}}{\approx}_{\delta/2} (U, Z)$ or $(0 \oplus B, Z) \overset{\mathsf{c}}{\approx}_\delta (1 \oplus B, Z)$. We would like to prove that $(\star)$ for every polynomial $p(\cdot)$, no $p(n)$-size adversary can predict the random variable $\bigoplus_{i=1}^n B_i$ with advantage better than $\delta^n + \mathrm{negl}$ given side information $Z = (Z_1, \ldots, Z_n)$.

For this, let us view the distribution (B, Z) as a (degenerate) real oracle F that ignores its input and outputs a sample (b, z), and the distribution (U, Z) as the ideal oracle F^* where U samples a uniform bit that is independent of Z. For $b \in \{0,1\}$ define the (non-interactive) combiner $C^{(\cdot)}_b$ that given n samples $(b_1, z_1), \ldots, (b_n, z_n)$, one from each oracle, outputs the bit $b \bigoplus_{i=1}^n b_i$ together with $z = (z_1, \ldots, z_n)$. Observe that the resulting combiner $(C^{(\cdot)}_0, C^{(\cdot)}_1)$ is a perfect 1-out-of-n indistinguishability combiner (i.e., for arbitrary $s_c = \infty$ and $\varepsilon = 0$), because one ideal sample where $b_i \leftarrow U$ is sufficient to perfectly hide b. Invoking Corollary 2, we obtain

$$C_0(\vec{R}) \overset{\mathsf{c}}{\approx}_{\delta^n + \mathrm{negl}} C_1(\vec{R}),$$

assuming $n(\lambda), \ell(\lambda) = \mathrm{poly}(\lambda)$ and using that $R \overset{\mathsf{c}}{\approx}_{\delta/2} I$, hence $\delta' = \binom{n}{0}(2 \cdot \delta/2)^n = \delta^n$ (with $\varepsilon' = 0$) and $(\star)$ follows.

Similarly, we could amplify a δ-weak bit encryption scheme where $E(0) \overset{\mathsf{c}}{\approx}_{\delta} E(1)$, using the XOR transformation, to δ^n security error. This is done by considering the ideal system that ignores its input bit and encrypts a random bit instead; then the real and ideal systems are $\delta/2$ indistinguishable, since with probability $1/2$ they encrypt the same bit. (Note that defining the ideal system to encrypt zero results in suboptimal parameters.)

Example 3 (Bitwise-XOR Lemma)**.** We can also generalize the XOR lemma to m-bit strings, under two different definitions. The first definition, which is stronger, asserts that for every $m_0, m_1 \in \{0,1\}^m$ we have $(m_0 \oplus M, Z) \overset{\mathsf{c}}{\approx}_{\delta} (m_1 \oplus M, Z)$. The second definition only requires that $(M, Z) \overset{\mathsf{c}}{\approx}_{\delta(1-2^{-m})} (U_m, Z)$, which corresponds to the restricted case of $m_0 = 0$ and $m_1 \leftarrow U_m \setminus \{0\}$ under the first definition. In the boolean setting $m = 1$ the second also implies the first because $U_m \setminus \{0\} \equiv 1$ which is the only other message, but this implication is no longer true for general $m > 1$.

For the first definition, consider any challenge messages $m'_0, m'_1 \in \{0,1\}^m$, and the combiner

$$C_b((m_1, z_1), \ldots, (m_n, z_n)) := \left(m'_b \bigoplus_{i=1}^{n} m_i,\ \vec{z} \right) .$$

We view the distribution (M, Z) as real, and $0.5(M, Z) + 0.5(m'_0 \oplus m'_1 \oplus M, Z)$ as ideal. Then the above is a perfect 1-out-of-n combiner because

$$m'_0 \oplus U_1 \cdot (m'_0 \oplus m'_1) \equiv m'_1 \oplus U_1 \cdot (m'_0 \oplus m'_1) .$$

According to our assumption, real and ideal are $\delta/2$-indistinguishable, and hence we obtain $\delta' = \delta^n$ as in the boolean setting.

For the second definition, we note that the above approach fails because $M'_1 \oplus M'_1 \not\equiv 0$ for two independent samples from $U_m \setminus \{0\}$ (i.e., we cannot invert). Instead, we observe that it implies the first definition with a multiplicative penalty of $2(1 - 2^{-m})$ as

$$(m_0 \oplus M, Z) \overset{\mathsf{c}}{\approx}_{\delta(1-2^{-m})} (m_0 \oplus U_m, Z) \equiv (m_1 \oplus U_m, Z) \overset{\mathsf{c}}{\approx}_{\delta(1-2^{-m})} (m_1 \oplus M, Z) .$$

Therefore, in this weaker setting, the amplification parameter becomes $\delta' = (2\delta(1 - 2^{-m}))^n$.

We remark that in the statistical setting, by employing a slightly tighter analysis using the neutralizing notion, we may shave a factor 2 and obtain $\delta' = (2\delta(1 - 2^{-m}))^n/2$. However, if we are only guaranteed that $(M, Z) \overset{\mathsf{s}}{\approx}_{\delta(1-2^{-m})} (U_m, Z')$ for some Z', then our approach still holds while the neutralizing framework fails.

Remark 4 (Limitations: Single-Query Oracles and Handling Inefficient Candidates)**.** As established, Theorem 2 restricts the combiner to making a single query to each oracle (i.e., each candidate is "single-use"). However, there is no restriction on the interaction between the external distinguisher and the combiner. The distinguisher may be fully adaptive, and the combiner itself can query its oracles adaptively (e.g., deciding which to query next based on previous answers), as long as each oracle is called at most once.

Still, the restriction to a single query entails certain limitations. To illustrate its impact, consider the secret-sharing example (Example 1) when the underlying weak scheme is public-key encryption. In this setting, we model the oracle as returning both the public key and the ciphertext corresponding to the queried message. Since the oracle can be queried only once, the message must be fixed before the public key is revealed. Consequently, the resulting security guarantee applies only to

messages that are independent of the public key. Nevertheless, for PKE there are standard transformations (e.g., via hybrid encryption) that upgrade such a guarantee to full security for messages that depend on the public-key. Similarly, for FE one may begin with a weak selective-security notion and subsequently amplify it to a full-fledged primitive, or even to iO (see Section 6). More broadly, however, it is desirable to extend the theorem to support multiple adaptive oracle queries directly, without relying on primitive-specific transformations; such extensions are discussed below in Section 5.3.

Finally, our theorem requires the combiner to guarantee indistinguishability even if the non-ideal (bad) candidates are inefficient. That is, the definition of a combiner remains unchanged from the statistical setting. While this restricts the types of combiners we can use—a challenge we must explicitly overcome when constructing our FE amplifier—we note that our theorem also does not require the *ideal* oracles to be efficient (a definitional relaxation).

5.3 Extensions

At a high level, the computational generalization relies on the fact that for single-round systems, if $R_i \overset{\mathsf{c}}{\approx}_\delta I_i$, then there exists a hybrid system R'_i such that $R_i \overset{\mathsf{c}}{\approx} R'_i \overset{\mathsf{s}}{\approx}_\delta I_i$. The statistical combiner-is-amplifier bound is then applied to $\vec{I}$ and $\vec{R}'$, and adapted to $\vec{R}$ against efficient transformations relying on the computational indistinguishability of $\vec{R}$ and $\vec{R}'$.

CC-Stateless Systems and Unlimited Queries. A similar approach can be applied when R_i and I_i are cc-stateless systems indistinguishable against *multiple adaptive queries* [Tes11]. Roughly speaking, a cc-stateless system is equivalent to a convex combination over function-induced systems. That is, conceptually, a cc-stateless system first draws a deterministic function $f \leftarrow F$, and then answers all subsequent queries consistently according to f.

While single-round systems trivially fall into this category, in the general cc-stateless setting, the existence of a hybrid system R'_i can only be guaranteed if R_i and I_i are indistinguishable against an a priori unbounded number of queries.[10] This setting naturally captures primitives like pseudorandom functions (PRFs) weakly indistinguishable from a random oracle against an a priori unbounded number of queries, and thus allows our combiner-is-amplifier theorem to capture, for example, the natural XOR amplifier for weak PRFs [MT09]. Let us elaborate on another natural application in the context for one-way functions.

Example 4 (Direct Product Threshold Amplification for One-Way Functions)**.** OWFs naturally correspond to cc-stateless systems as follows: The real system R samples an image $y \leftarrow f(U_\lambda)$ as an initial state and answers each query x with (y, b), where b indicates if $f(x) = y$. The ideal system I sets an initial state identically, but it always responds with $(y, 0)$. It is straightforward to see that the hardness of inverting the function naturally corresponds to the hardness of distinguishing between the systems.

This allows us, for example, to derive a bound on the hardness of inverting at least $(n-k+1)$-out-of-n weak OWFs, using the following perfect (k, n) combiner $C_b^{\vec{S}}$: given a query $(x_1, \ldots, x_n)$, the combiner sends each x_i to system S_i and receives a response (y_i, b_i). It then sets b' to 1 only if $\sum_{i=1}^{n} b_i \geq n - k + 1$, and responds with $(y_1, \ldots, y_n, b \cdot b')$. Then $(C_0^{(\cdot)}, C_1^{(\cdot)})$ is a perfect (k, n) indistinguishability combiner (against unlimited queries) w.r.t. ideal $\vec{I}$ since if at least k bits $(b_{i_1}, \ldots, b_{i_k})$ are always set to 0, b' is always set to 0 as well and the output bit $b \cdot b'$ becomes independent of b.$\diamond$

[10] In fact, to ensure $R_i \overset{\mathsf{c}}{\approx}_\gamma R'_i \overset{\mathsf{s}}{\approx}_\delta I_i$ against h adaptive queries, we need $R_i \overset{\mathsf{c}}{\approx}_\delta I_i$ against $\approx h/\gamma^2$ queries.

We remark that the above example naturally generalizes to publicly verifiable puzzles, or more generally privately verifiable puzzles with unlimited attempts. Closely related settings were studied in [CHS05, IJK09, HR08, Jut10, HS11].

Resettable Indistinguishability. As a final remark, we informally note that our combiner-is-amplifier framework can be generalized even further. While Tessaro [Tes11] proved that the indistinguishability hardcore lemma extends to cc-stateless systems, it can be observed that the lemma also generalizes to systems indistinguishable under reset attacks. Roughly speaking, the hardcore lemma fails for general interactive systems because the reduction attempts to combine many weak distinguishers (each working against a small subset of the randomness) by running all of them and taking the majority. However, in the general interactive setting, executing the first weak distinguisher alters the state of the system, meaning the views of all subsequent distinguishers are altered (and in particular, could just be $\perp$). To solve this, the reduction needs the ability to "reset" the system—a mechanism that is inherently available in the cc-stateless setting.

In case the real and ideal systems have efficient interactive implementations over initial states, it is sufficient to consider combiners that only guarantee indistinguishability for efficient "bad" candidates (as opposed to our general combiner-is-amplifier theorem where "bad" candidates can be arbitrary). The idea is that we can view the combiner as a combiner over random variables (initial states and bits indicating real/ideal), simulating the oracle systems on its own. Then, the "inefficient bad candidates" are just samples of initial states that could be hardcoded as non-uniform advice.

6 Amplification of Functional Encryption

In this section, we use our single-query combiner-is-amplifier theorem to achieve security amplification for weak FE schemes with sufficiently small constant indistinguishability error. We first construct a single-query FE combiner using homomorphic secret sharing, then apply our main theorem. We note that we cannot use the existing FE combiner due to [JMS20], as it sends many queries to each candidate. Also, the amplifier construction presented in [JKMS20] does not induce a proper combiner with negligible error for a constant δ due to a randomized choice of sets that we fix. Of even greater significance, we need a stronger semi-malicious privacy notion, since our combiner must be secure against inefficient "bad" candidates.

6.1 Weak Functional Encryption

Below we describe a variant that is known in the literature as single-key, weakly selective secure, weakly compact PKFE [GS16].

Definition 9 (Functional Encryption)**.** A functional encryption scheme FE, for a function class $\mathcal{F}$ (represented by Boolean circuits) and message space $\{0,1\}^*$, consists of four PPT algorithms ($\mathsf{FE.Setup}$, $\mathsf{FE.Keygen}$, $\mathsf{FE.Enc}$, $\mathsf{FE.Dec}$) with the following syntax:

- $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)$: takes as input the unary representation of the security parameter and a size bound $\lambda, p \in \mathbb{N}$, and outputs the master public key (for encryption) and master secret key (for functional-key generation).
- $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)$: takes as input the master secret key and a function $f \in \mathcal{F}$, and outputs a (secret) functional key.

- $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x)$: takes as input the master public key and a message $x \in \{0,1\}^*$, and outputs a ciphertext.
- $y \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: takes as input a functional key and a ciphertext, and outputs a message.

We next define the relevant properties:

- Correctness: for all $\lambda, p, \ell \in \mathbb{N}$, message $x \in \{0,1\}^\ell$ and function $f \in \mathcal{F}$ with domain $\{0,1\}^\ell$ and size $|f| \leq p$, we have

$$\Pr\left[\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT}) = f(x)\right] = 1\,,$$

  over $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)$, $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)$, $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x)$.

- ε-SIM-Security: there exists a PPT simulator $\mathsf{FE.Sim}$ such that for all $\{x, f, p\}_\lambda$ with $|f| \leq p = \mathrm{poly}(\lambda)$ we have

$$(\mathsf{MPK},\, \mathsf{FSK}_f,\, \mathsf{CT}) \overset{\mathsf{c}}{\approx}_{\varepsilon(\lambda)} \mathsf{FE.Sim}(1^\lambda,\, 1^p,\, f,\, f(x))\,,$$

  over $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)$, $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)$, $\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x)$.

- ε-IND-Security: for every $\{x_0, x_1, f, p\}_\lambda$ with $|x_0| = |x_1|, f(x_0) = f(x_1)$ and $|f| \leq p = \mathrm{poly}(\lambda)$, we have

$$(\mathsf{MPK},\, \mathsf{FSK}_f,\, \mathsf{CT}_0) \overset{\mathsf{c}}{\approx}_{\varepsilon(\lambda)} (\mathsf{MPK},\, \mathsf{FSK}_f,\, \mathsf{CT}_1)\,,$$

  over $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)$, $\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)$, $\mathsf{CT}_b \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x_b)$.

- Succinctness: for every $\lambda, p, \ell \in \mathbb{N}$, the circuit size of $\mathsf{FE.Enc}(\mathsf{MPK}, \cdot)$, over $\{0,1\}^\ell$ where $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)$, is upper bounded by $p^{1-c} \cdot \mathrm{poly}(\lambda, \ell)$ for some universal constant $0 < c \leq 1$. If $c = 1$, we say that the scheme is fully succinct.

We note that weak FE implies weak OWFs, which can be amplified to standard OWFs [Yao82].

Remark 5. We note that the literature contains several common variants of FE, to which our results also extend (either directly or by amplifying our weaker version, then lifting). We choose to focus on weak FE because: (1) it is relatively simple, (2) its connection to other notions of FE is well understood, and (3) it is sufficient (in its subexponentially hard version) for the construction of indistinguishability obfuscation. We briefly address how our results generalize for these notions:

- **From Weak to Strong FE:** The weak notion of FE we address, in its fully secure version (i.e., post amplification), can be upgraded to a strong notion of FE that is both *adaptively secure* and *resistant to unbounded collusion*, which is considered the gold standard in the context of FE. Such a transformation (with only a polynomial loss in security) is shown in [KNTY19] (building on a sequence of works on upgrading FE such as [ABSV15, GS16, LM16]).

  Also, as expected, the stronger notions imply our weak notion either directly or via known transformations [AJS15, BV15], and this also holds in the setting of large security error.

  Thus, when it comes to amplification, it is sufficient to focus on this weak notion.

- **Indistinguishability Obfuscation:** A main application of FE is the construction of indistinguishability obfuscation (IO) [AJ15, BV15]. For this application, weak FE is sufficient,[11] but it is required that the FE has a subexponentially small indistinguishability error. Our amplifier can reach this level of indistinguishability, assuming that the weak FE we start from is (weakly) indistinguishable against adversaries of subexponential size (see Theorem 6). We note that while previous constructions of IO went through weak FE as a stepping stone (e.g., [AJS18, AJL+19]), state-of-the-art constructions [JLS22] manage to construct strong FE directly.

  We note that in the setting of subexponential security, our amplifier also indirectly implies IO amplification. Starting from δ-weak IO against subexponential-size adversaries, we can apply the IO-based construction of FE from [GGH+13]. This construction would imply a 2δ-weak single-key FE against subexponential attackers, which can then be amplified and used to construct IO.

- **Secret Key FE:** We focus on public-key FE, which is somewhat more common. Our proof for weak public-key FE can be easily adapted to weak secret-key FE. We note, however, that in the setting of secret-key FE, upgrading from a weak scheme to a strong one (collusion-resistant and adaptively secure) requires subexponential security. In this case, it can actually be upgraded all the way to IO [KNT18].

6.2 Homomorphic Secret Sharing

Below we describe a special variant of HSS, a notion first introduced in [BGI16, BGI+18]. In the following, we assume that a functionality f is given as a Boolean circuit. An efficiently computable functionality $f = \{f\}_\lambda$ is a (non-uniform) family of circuits of size $s(\lambda) \leq \text{poly}(\lambda)$ over $\ell(\lambda) \leq \text{poly}(\lambda)$ input bits. The construction is parametrized by some functions $t(\lambda), n(\lambda)$.[12] For ease of reading, we sometimes omit the dependency on λ and simply write s, ℓ and t, n. In Section 8 we show how to instantiate semi-malicious HSS assuming OWFs, for any $t \leq \text{poly}(\lambda)$ and $n \geq \binom{3t}{3}$, or for $n \geq 3t$ if $t = O(\log(\lambda))$.

Definition 10 (Homomorphic Secret Sharing)**.** A homomorphic secret sharing scheme HSS consists of three PPT algorithms $(\mathsf{HSS.Share}, \mathsf{HSS.Eval}, \mathsf{HSS.Dec})$ with the following syntax:

- $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$: takes as input the unary representation of the security parameter $\lambda \in \mathbb{N}$ and an input $x \in \{0,1\}^\ell$. The sharing algorithm outputs n shares, each of size $\ell' = \ell \cdot \text{poly}(\lambda)$.

- $y_i := \mathsf{HSS.Eval}(1^\lambda, f, i, x_i)$: deterministic; takes as input the unary representation of the security parameter $\lambda \in \mathbb{N}$, a functionality f where s and ℓ denote the circuit size of f and the input length, an index $i \in [n]$ and a share $x_i \in \{0,1\}^{\ell'}$. The evaluation's circuit complexity is $s' = s \cdot \text{poly}(\lambda)$, and it outputs an output share y_i.

- $y := \mathsf{HSS.Dec}(1^\lambda, f, y_1, \ldots, y_n)$: deterministic; the decoder takes as input the unary representation of the security parameter $\lambda \in \mathbb{N}$, a functionality f and an n-tuple $(y_1, \ldots, y_n)$ (supposedly output shares in the image of the evaluator).

[11] In fact, an even weaker notion of FE, in which a single function is known at setup time, is sufficient. Our amplification theorem readily extends to this weaker version.

[12] We parametrize combiners with a lower bound on the number of good candidates k, and secure computation with an upper bound on the number of malicious parties t.

The HSS should satisfy the following properties:

- Perfect Correctness: for every security parameter $\lambda \in \mathbb{N}$, every functionality f with input length ℓ, and every input $x \in \{0,1\}^\ell$:

$$\Pr\left[\mathsf{HSS.Dec}(1^\lambda, f, y_1, \ldots, y_n) = f(x)\right] = 1\,,$$

  over $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$, $\forall i \in [n] : y_i := \mathsf{HSS.Eval}(1^\lambda, f, i, x_i)$.

- Semi-Malicious t-Privacy:

  1. $\mathsf{HSS.Share}$ is t-wise input-independent, namely for every security parameter $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^\ell$ and subset $T \subseteq [n]$ of size $|T| < t$:

$$(x_i)_{i\in T} \equiv (x_i^*)_{i\in T}\,,$$ [13]

     over $(x_i)_{i\in[n]} \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$ and $(x_i^*)_{i\in[n]} \leftarrow \mathsf{HSS.Share}(1^\lambda, 0^\ell)$.

  2. There exists an efficient randomized simulator $\mathsf{HSS.Sim}$ such that for every ensemble $\{f, x, T, x_T\}_\lambda$, where f is a function with input length ℓ, $x \in \{0,1\}^\ell$, $T \subseteq [n]$ with $|T| < t$ and x_T are valid shares $(x_i)_{i\in T} \in \mathrm{Supp}(\mathsf{HSS.Share}(1^\lambda, x)_T)$:

$$(y_1, \ldots, y_n) \overset{\mathsf{c}}{\approx} \mathsf{HSS.Sim}(1^\lambda, f, T, x_T, f(x))\,,$$

     over conditional input sharing $(x_1, \ldots, x_n) \leftarrow (\mathsf{HSS.Share}(1^\lambda, x) \mid x_T)$ and the randomness of the simulator, where $\forall i \in [n] : y_i := \mathsf{HSS.Eval}(1^\lambda, f, i, x_i)$.

6.3 The Amplifier

The construction of our FE amplifier is described in Fig. 1. To show IND-security, given x, x', f, our goal is to define a related single-query indistinguishability combiner $(C^{(\cdot)}_{x,f}, C^{(\cdot)}_{x',f})$ w.r.t. some ideal $\vec{F}^*$, such that for some real $\vec{F}'$ with $\forall i : F'_i \overset{\mathsf{c}}{\approx}_\delta F^*_i$, plugging $\vec{F}'$ into the combiner results in distributions identical to $(\mathsf{MPK}, \mathsf{FSK}_f, \mathsf{CT})$ and $(\mathsf{MPK}, \mathsf{FSK}_f, \mathsf{CT}')$ given by the FE amplifier (up to output formatting). The combiner is defined in Fig. 2 and the real/ideal functionalities will be defined later, but essentially the ideal functionality corresponds to the weak simulator and the real one to running the weak FE normally.

Theorem 3. *Given a (t,n)-HSS and $\{x, x', f\}_\lambda$ such that $f : \{0,1\}^\ell \to \{0,1\}^m$, $x, x' \in \{0,1\}^\ell$, $f(x) = f(x')$ and $|f| \le \mathrm{poly}(\lambda)$, the n-oracle-aided pair $\left\{C^{(\cdot)}_{x,f}, C^{(\cdot)}_{x',f}\right\}_\lambda$ described in Fig. 2, is a $(\mathrm{poly}, \mathrm{negl}, n-t+1, n)$ indistinguishability combiner w.r.t. any ideal $\left\{\vec{F}^*\right\}_\lambda$ such that $\mathrm{size}(\vec{F}^*) \le \mathrm{poly}(\lambda)$ and each $F^*_i(\cdot,\cdot)$ ignores its first input.*

Proof. In the following we omit the security parameter for ease of notation. Let $\vec{F}^* = (F^*_1, \ldots, F^*_n)$ be any ideal functions that are efficient and ignore their first input, namely $\mathrm{size}(\vec{F}^*) \le \mathrm{poly}(\lambda)$ and $\forall i \in [n] : F^*_i(\cdot,\cdot) \equiv F^*_i(\bot,\cdot)$. We are given some $\vec{F} = (F_1, \ldots, F_n)$ such that $F_i \equiv F^*_i$ for at least $n-t+1$ indices $G \subseteq [n]$. Note that for $i \notin G$ the function F_i is arbitrary (over the given codomain) and in particular may be inefficient. We need to show that $C^{\vec{F}}_{x,f} \overset{\mathsf{c}}{\approx} C^{\vec{F}}_{x',f}$. Consider the hybrid $H^{\vec{F}}_{y,f}$ defined in Fig. 3, where $y = f(x) = f(x')$. Then we claim that $C^{\vec{F}}_{x,f} \overset{\mathsf{c}}{\approx} H^{\vec{F}}_{y,f} \overset{\mathsf{c}}{\approx} C^{\vec{F}}_{x',f}$.

[13] This can be relaxed to a small statistical distance.

Given a weak SIM-secure functional encryption scheme FE^w and a (t,n)-HSS, the construction works as follows:

$\underline{(\mathsf{MPK},\mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)}$

1. For all $i \in [n]$, call $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{FE.Setup}^w(1^\lambda, 1^{p'})$, where $p' \leq p \cdot \mathrm{poly}(\lambda)$ is the corresponding upper bound of $\mathsf{HSS.Eval}$.
2. Output $\mathsf{MPK} := (\mathsf{MPK}_1, \ldots, \mathsf{MPK}_n)$, $\mathsf{MSK} := (\mathsf{MSK}_1, \ldots, \mathsf{MSK}_n)$.

$\underline{\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x)}$

1. Share $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$.
2. For all $i \in [n]$, call $\mathsf{CT}_i \leftarrow \mathsf{FE.Enc}^w(\mathsf{MPK}_i, x_i)$.
3. Output $\mathsf{CT} := (\mathsf{CT}_1, \ldots, \mathsf{CT}_n)$.

$\underline{\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)}$

1. For all $i \in [n]$, let $f_i := \mathsf{HSS.Eval}(1^\lambda, f, i, \cdot)$ and call $\mathsf{FSK}_{f_i} \leftarrow \mathsf{FE.Keygen}^w(\mathsf{MSK}_i, f_i)$.
2. Output $\mathsf{FSK}_f := (\mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_n})$.

$\underline{y \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})}$

1. For all $i \in [n]$, call $y_i \leftarrow \mathsf{FE.Dec}^w(\mathsf{FSK}_{f_i}, \mathsf{CT}_i)$.
2. Output $y := \mathsf{HSS.Dec}(1^\lambda, f, y_1, \ldots, y_n)$.

Figure 1: FE Amplifier

The single-query indistinguishability combiner $C^{(\cdot)}_{x,f}$, given a (t,n)-HSS and $\forall i \in [n] : f_i := \mathsf{HSS.Eval}(1^\lambda, f, i, \cdot)$, is defined below. (The corresponding $C^{(\cdot)}_{x',f}$ is identically defined.)

$\underline{z \leftarrow C^{\vec{F}}_{x,f}}$

1. Share $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$.
2. For all $i \in [n]$, call
$$z_i \leftarrow F_i\left(x_i,\ f_i(x_i)\right) .$$
3. Output $z := (z_1, \ldots, z_n)$.

Figure 2: Combiner $\left(C^{(\cdot)}_{x,f}, C^{(\cdot)}_{x',f}\right)$

The hybrid $H^{\vec{F}}_{y,f}$, given a (t,n)-HSS, $\forall i \in [n] : f_i := \mathsf{HSS.Eval}(1^\lambda, f, i, \cdot)$ and $y = f(x) = f(x')$, $\ell = |x| = |x'|$, $G \subseteq [n]$, is defined below:

$\underline{z \leftarrow H^{\vec{F}}_{y,f}}$

1. Share $(x_1^*, \ldots, x_n^*) \leftarrow \mathsf{HSS.Share}(1^\lambda, 0^\ell)$.
   First change: $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$.
2. For all $i \notin G$, call
$$z_i \leftarrow F_i\left(x_i^*,\ f_i(x_i^*)\right) .$$
3. Invoke $(y_1, \ldots, y_n) \leftarrow \mathsf{HSS.Sim}(1^\lambda, f, \overline{G}, x^*_{\overline{G}}, y)$.
   Second change: $\forall i \in [n] : y_i \leftarrow f_i(x_i)$.
4. For all $i \in G \implies F_i \equiv F_i^*$, call
$$z_i \leftarrow F_i^*\left(\perp,\ y_i\right) .$$
5. Output $z := (z_1, \ldots, z_n)$.

Figure 3: Hybrid $H^{\vec{F}}_{y,f}$

Let us focus on x as the argument for x' is symmetric. We make the switch from $H_0 := H^{\vec{F}}_{y,f}$ to $H_2 := C^{\vec{F}}_{x,f}$ in two steps. Hybrid H_1 is defined by making the first change from $(x_1^*, \ldots, x_n^*) \leftarrow \mathsf{HSS.Share}(1^\lambda, 0^\ell)$ to $(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)$ (and substituting x_i for x_i^* in subsequent steps). By noting that $H^{\vec{F}}_{y,f}$ depends only on input shares in $\overline{G}$, and relying on the fact that $\mathsf{HSS.Share}$ is t-wise input independent with $\left|\overline{G}\right| < t$, we conclude that $H_0 \equiv H_1$.

For the second step, in H_1, we make the second change from $(y_1, \ldots, y_n) \leftarrow \mathsf{HSS.Sim}(1^\lambda, f, \overline{G}, x_{\overline{G}}, y)$ to $\forall i \in [n] : y_i \leftarrow f_i(x_i)$. First, we note that indeed $H_2 \equiv C^{\vec{F}}_{x,f}$, due to

$$\forall i \in G :\ F_i^*(\perp, y_i) = F_i^*(x_i, y_i) = F_i^*(x_i, f_i(x_i)) = F_i(x_i, f_i(x_i)) .$$

Next, relying on the security of $\mathsf{HSS.Sim}$, it is crucial that the simulation is computationally secure against any fixed choice of $x_{\overline{G}}$, since $(F_i)_{i \notin G}$ may be inefficient. In more detail, given a distinguisher between H_1 and H_2, we may fix the shares $x_{\overline{G}}$ in a way that preserves its advantage (averaging argument). Furthermore, we can fix the randomness of all $(F_i)_{i \notin G}$, and since they depend only on shares in $\overline{G}$, it fixes all responses $(z_i)_{i \notin G}$ which we can use as hardcoded advice. Finally, by noting that the second change exactly matches the security game of the HSS simulator, and the remaining computation is efficient, we deduce that $H_1 \overset{\mathsf{c}}{\approx} H_2$. □

Theorem 4. *Let* FE^w *be a* δ*-SIM-secure functional encryption scheme for* $\delta < 0.5$*. Then the scheme* FE *described in Fig. 1, instantiated with an* $(n/3, n)$*-*HSS *for* $n := 3\lceil \log \lambda \rceil$*, is a* $(\delta' + \mathrm{negl})$*-IND-secure functional encryption scheme where* $\delta' := \binom{n}{n/3}(2\delta)^{n/3} \leq (3e \cdot 2\delta)^{n/3}$.

Proof. Succinctness is due to the succinctness of the weak FE and the fact that $\mathsf{HSS.Eval}$ incurs only a $\mathrm{poly}(\lambda)$ overhead. Correctness follows from the weak FE and HSS correctness.

In order to show functional indistinguishability, consider $\{x, x', f, p\}_\lambda$ with $\ell := |x| = |x'|, y := f(x) = f(x')$, and $|f| \leq p = \text{poly}(\lambda)$. We omit the security parameter subscript for ease of notation. For $i \in [n]$, consider the ideal functionalities $F_i^*(a_1, a_2) := \mathsf{FE.Sim}^w(1^\lambda, 1^{p'}, f_i, a_2)$, where p' and f_i are defined in Fig. 1. Since F_i^* is efficient and the first input is ignored, the conditions of Theorem 3 are satisfied, and we deduce that $(C_{x,f}^{(\cdot)}, C_{x',f}^{(\cdot)})$ is a $(\text{poly}, \text{negl}, 2n/3 + 1, n)$ indistinguishability combiner w.r.t. $\vec{F}^*$.

Now, let $F_i'(a_1, a_2)$ be the real functionality that is defined according to the SIM-security challenge, namely it returns $(\mathsf{MPK}_i, \mathsf{FSK}_{f_i}, \mathsf{CT}_i)$ given by $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{FE.Setup}^w(1^\lambda, 1^{p'}), \mathsf{FSK}_{f_i} \leftarrow \mathsf{FE.Keygen}^w(\mathsf{MSK}_i, f_i), \mathsf{CT}_i \leftarrow \mathsf{FE.Enc}^w(\mathsf{MPK}_i, a_1)$. In case its inputs are inconsistent with our combiner's queries, namely $a_2 \neq f_i(a_1)$, it simply returns $F_i^*(a_1, a_2)$ so there is no distinguishing advantage at all. We observe that $C_{x,f}^{\vec{F}'}$ and $C_{x',f}^{\vec{F}'}$ correspond to the IND-security game of FE, up to formatting between $(\mathsf{MPK}_i, \mathsf{FSK}_{f_i}, \mathsf{CT}_i)_{i=1}^n$ and $((\mathsf{MPK}_i)_{i=1}^n, (\mathsf{FSK}_{f_i})_{i=1}^n, (\mathsf{CT}_i)_{i=1}^n)$.

Finally, since $\forall i \in [n] : F_i' \overset{\mathsf{c}}{\approx}_\delta F_i^*$ due to the δ-SIM-security of FE^w, we can apply Theorem 2 and get that $C_{x,f}^{\vec{F}'} \overset{\mathsf{c}}{\approx}_{\delta'+\text{negl}} C_{x',f}^{\vec{F}'}$ for $\delta' \leq \binom{n}{n/3}(2\delta)^{n/3}$, since $\varepsilon' \leq \binom{n}{2n/3+1} \cdot \text{negl} = \text{poly} \cdot \text{negl} = \text{negl}$, and the slackness parameter γ can be made as polynomially small as we want. In more detail, for every $s', (1/\gamma) \leq \text{poly}(\lambda)$, the size bound

$$\frac{\left(s' + \text{size}(C_{x,f}^{\vec{F}'})\right) \cdot \log |\text{Im}(\mathsf{FE}^w)|}{\gamma^2}$$

remains polynomial and hence the δ-SIM-security of FE^w applies, so the conditions of Theorem 2 hold and we derive $C_{x,f}^{\vec{F}'} \overset{\mathsf{c}}{\approx}_{\delta'+\text{negl}} C_{x',f}^{\vec{F}'}$. We conclude that FE is $(\delta' + \text{negl})$-IND-secure. □

In the theorem below, we employ a generic transformation from IND-security to SIM-security, as formalized in [DIJ+13]. This transformation utilizes trapdoor circuits based on SKE/OWFs. For the sake of completeness, an appropriate Lemma 6 is included in Section A.

Theorem 5. *Assuming the existence of a δ-IND-secure functional encryption scheme for some constant $\delta < 1/(6e)$, there also exists a* negl*-IND-secure functional encryption scheme.*

Proof. First, we note that weak FE implies OWFs which in turn imply SKE and an $(n/3, n)$-HSS for $n := 3\lceil \log \lambda \rceil$. We use the IND-security to SIM-security transformation (Lemma 6) to go from δ-IND-secure FE to $(\delta + \text{negl})$-SIM-secure FE. Next, we use the FE amplifier (Theorem 4) to obtain a $(1/\text{poly})$-IND-secure FE, as $(3e \cdot 2\delta) < 1$ and $n = \Omega(\log \lambda)$. We repeat this process once again: first the IND-security to SIM-security transformation to obtain a $(1/\text{poly})$-SIM-secure FE, followed by the FE amplifier resulting in a negl-IND-secure FE. □

Using similar ideas, along with a "heavier" combiner via $(\lambda^{c/3}, 5\lambda^c)$-HSS, we can generalize the above to the subexp setting. For more information, see Section B.

Theorem 6. *Assuming the existence of a δ-IND-secure subexp functional encryption scheme for some constant $\delta < 1/(6e)$, there also exists a* $(1/\text{subexp})$*-IND-secure subexp functional encryption scheme.*

We note without proof that for any constant $\delta < 1$, we can amplify δ-IND-secure FE to any smaller constant (such as $1/(6e)$) by nesting, as pointed out in [JKMS20].

Corollary 3. *Assuming the existence of a δ-IND-secure functional encryption scheme for some constant $\delta < 1$, there also exists a* negl*-IND-secure functional encryption scheme.*

Corollary 4. *Assuming the existence of a δ-IND-secure subexp functional encryption scheme for some constant $\delta < 1$, there also exists a* (1/subexp)*-IND-secure subexp functional encryption scheme.*

Remark 6. At a high level, let us outline how our handling of the hardcore lemma differs from the approach taken by [JKMS20]. The main challenge with the hardcore lemma is that the hardcore event, namely the conditioning of randomness on the "shared distribution" (coupling analog), depends on the secret input. In particular, even though the FE simulator does not initially depend on any input x, once we condition on the hardcore event $(\mathsf{MPK}, \mathsf{FSK}_f, \mathsf{CT}_x) \mid H_E \overset{\mathsf{c}}{\approx} \mathsf{FE.Sim}(f, y) \mid H_S$, the randomness chosen for the simulator H_S becomes dependent on x, and so does its output. Similarly, conditioning on the hardcore event not occurring $\overline{H_S}$ also creates a dependency on x.

Jain et al. [JKMS20] tried to simulate the hardcore event by hiding x inside a strong commitment and then using a leakage simulation lemma to simulate an inefficient transformation that breaks the commitment and then samples from the hardcore event. However, this approach failed because eventually they had to change the underlying committed value x, which fundamentally requires changing the leakage simulator as well. They implicitly assumed a single leakage simulator could work for both values, but because the required output distribution depends on the hidden secret, evaluating such a simulator inherently violates the commitment's hiding property.

Instead, we (1) employ a (t, n) semi-malicious HSS, ensuring that the inputs of fewer than t "bad" instances (where the hardcore event failed) are challenge-independent. Hence, no information is leaked and simulation is unnecessary. (2) We carefully analyze the impact of (small) leakage from "good" instances using the general combiner-is-amplifier theorem. In particular, the term $\delta' := \binom{n}{k-1}(2\delta)^{n-k+1}$ can be viewed as quantifying a penalty for the possible leakage, compared to the "idealized" bound of $\sum_{i=0}^{k-1}\binom{n}{i}\delta^{n-i}(1-\delta)^i$.

7 Simultaneous Amplification

In this section, we informally sketch a general framework for simultaneous amplification, capturing primitives such as Oblivious Transfer (OT) and Non-Interactive Zero Knowledge (NIZK). We emphasize that there are already known amplification results for OT [CK90, DKS99, Wul07] and NIZK [GJS19, BG24, AK25b], and while our parameters are generally looser, our focus here is on the generality and simplicity of the framework. Below we focus on two weak properties, but our approach could be generalized to any constant number.

Let us assume that for each property we have a single-query $(1, n)$ combiner that also preserves the other property. That is, if the property holds in at least one candidate it also holds in the resulting scheme, and if the other property holds in all candidates it also holds in the resulting scheme.

Then, we can naturally construct a single-query $(n^2 - n + 1, n^2)$ combiner for both properties as follows: divide the n^2 candidates into n blocks of size n. Since the number of bad candidates is strictly smaller than n, we are guaranteed that:

- In every block there is at least one good candidate.
- There exists some block where all candidates are good.

Now, we apply the first-property $(1, n)$ combiner on every block to get n candidates. All of them have the first property (due to the first guarantee) and at least one also has the second property (due to the second guarantee). We then apply the second-property $(1, n)$ combiner on these n candidates to obtain a scheme where both properties hold.

Intuitively, using the combiner-is-amplifier theorem, the simultaneous combiner allows us to simultaneously amplify δ-weak properties to roughly $\binom{n^2}{n}(2\delta)^n \leq (en \cdot 2\delta)^n$. If we start from $\delta < 1/\mathrm{poly}(\lambda)$ then we can set $n = \log \lambda$ and amplify straight to negligible. If δ is weaker, say $\delta < 1/\log \lambda$, we can first amplify it to $1/\mathrm{poly}(\lambda)$ by setting $n = (\log \lambda)/6$, then invoke the amplifier again to obtain negligible error. In fact, even if δ is weaker than that, we could still amplify by choosing an increasing sequence of n's according to $n = 1/6\delta$, so every step reduces δ exponentially, allowing us to start from any $\delta < 1/\log^{(c)} \lambda$ where c is some constant and $\log^{(c)}$ denotes composition c times (i.e., $\log^* \lambda - \log^*(1/\delta)$ is constant).

8 Instantiating HSS

In this section, we show how to instantiate a (t, n) semi-malicious HSS (Definition 10) assuming OWFs, for any $t \leq \mathrm{poly}(\lambda)$ and $n \geq \binom{3t}{3}$, or for $n \geq 3t$ if $t = O(\log(\lambda))$.

8.1 Triple MPRE

We begin with some notational conventions.

Functionalities. In the following, we assume that an n-party functionality is given as a circuit whose inputs are labeled by the parties' names and where each party holds the same number of input bits. Accordingly, the number of parties and the input length are implicitly given as part of the circuit's description. An efficiently computable functionality $f = \{f\}_\lambda$ is a (non-uniform) family of $\mathrm{poly}(\lambda)$-size circuits, where each f_λ computes an $n(\lambda)$-party functionality over $\ell(\lambda)$ bits per party. For ease of reading, we sometimes omit the dependency on λ and simply write n and ℓ. By default, we restrict our attention to *public-output* functionalities whose outputs are delivered to all parties. (This is w.l.o.g. via standard reductions.)

Multiparty Randomized Encoding (MPRE). We employ a multiparty randomized encoding [ABT18] which extends the more standard notion of a randomized encoding [IK00, IK02, AIK04].

Definition 11 (MPRE and Triple-MPRE)**.** A *multiparty randomized encoding* scheme MPRE with semi-malicious security against arbitrary coalitions (MPRE in short) consists of two polynomial-time deterministic algorithms, $\mathsf{MPRE.Comp}$ and $\mathsf{MPRE.Dec}$, with the following syntax:

- $\hat{f} := \mathsf{MPRE.Comp}(1^\lambda, f)$: the compiler takes as input the unary representation of the security parameter $\lambda \in \mathbb{N}$, and a multiparty functionality f where s, n and ℓ denote the circuit size of f, its number of parties, and the input length of each party. The compiler outputs a circuit representation of an n-party functionality $\hat{f}$ that takes from each party an input $x_i \in \{0,1\}^\ell$ and randomness $r_i \in \{0,1\}^\lambda$, with circuit size $\hat{s} = s \cdot \mathrm{poly}(\lambda, n)$, that generates an output $\hat{y} \in \{0,1\}^{\hat{m}}$ for $\hat{m} = s \cdot \mathrm{poly}(\lambda, n)$. We refer to $\hat{f}$ as an encoding of f.
- $y := \mathsf{MPRE.Dec}(1^\lambda, f, \hat{y})$: the decoder takes as input the unary representation of the security parameter $\lambda \in \mathbb{N}$, a multiparty functionality f and a string $\hat{y} \in \{0,1\}^{\hat{m}}$ (supposedly in the image of an encoding $\hat{f}$ of f).

The MPRE should satisfy the following properties:

- Perfect Correctness: for every security parameter λ, every n-party functionality f with input length ℓ, every input vector $x \in (\{0,1\}^{\ell})^n$ and randomness vector $r \in (\{0,1\}^{\lambda})^n$,

$$\mathsf{MPRE.Dec}\left(1^{\lambda}, f, \hat{f}(x,r)\right) = f(x)\,,$$

  where $\hat{f} = \mathsf{MPRE.Comp}(1^{\lambda},\, f)$.

- Semi-Malicious Privacy: there exists an efficient randomized simulator $\mathsf{MPRE.Sim}$ such that for every ensemble $\{f, x, V, r_V\}_{\lambda}$, where f is an n-party functionality, $x = (x_1, \ldots, x_n) \in (\{0,1\}^{\ell})^n$, $V \subseteq [n]$ is a subset of corrupted parties, and $r_V = (r_i \in \{0,1\}^{\lambda})_{i \in V}$ is their randomness, we have:

$$\hat{f}(x,r) \overset{\mathsf{c}}{\approx} \mathsf{MPRE.Sim}(1^{\lambda}, f, V, x_V, r_V, f(x))\,,$$

  where $\hat{f} := \mathsf{MPRE.Comp}(1^{\lambda},\, f)$, and the distributions are taken over the uniform choice of honest randomness $r_{\overline{V}} \leftarrow (\{0,1\}^{\lambda})^{n-|V|}$ and the internal randomness of the simulator.

An MPRE is a *Triple-MPRE* (or TMPRE, in short) if the encoding $\hat{f}$ has *party-locality 3*, i.e., each output of $\hat{f}$ depends on at most 3 parties. Equivalently, $\hat{f}(x,r)$ can be written as

$$\left(\hat{f}_{i,j,k}((x_i,r_i),(x_j,r_j),(x_k,r_k))\right)_{i,j,k \in [n]}\,.$$

8.1.1 Instantiating TMPRE

We sketch below how to construct TMPRE using standard tools from the literature. We employ the computationally private BMR garbling mechanism of gates [BMR90], assuming OWFs. We observe that:

- It has effective degree 3, namely each party i can efficiently preprocess $z_i \leftarrow \mathsf{Pre}(x_i, r_i)$ and the final output is given by $h(z_1, \ldots, z_n)$ for some degree-3 functionality h. Intuitively, we need one multiplication for the gate's logic, and another one to choose (multiplexer) between two key options. (See, e.g., [DI05, GIS18, ABT18].)

- It has semi-malicious privacy against arbitrary coalitions, because every gate is garbled using the XOR of pseudorandom strings provided by each player. Thus, even if we fix the randomness of all but one player, the XOR is still pseudorandom.

Finally, we apply a general compiler from effective degree 3 to player locality 3. Using the MPRE composition lemma [ABT18, Lemma 3.4], it is sufficient to show that every degree 3 multiparty functionality $\sum z_i z_j z_k$ has a perfect MPRE. For every monomial $T = \{i,j,k\}$ the participating parties sample random masks w_i^T, w_j^T, w_k^T and output $z_i z_j z_k + w_i^T + w_j^T + w_k^T$. Then, to compute the total sum of all masks, every party shares their local sum using additive secret sharing, and every pair of parties outputs the sum of their shares $s_{i,j} + s_{j,i}$. (See [ABG+20] for a similar transformation.)

We note that the circuit size and output size overhead is indeed a multiplicative factor $\mathrm{poly}(\lambda, n)$ as claimed, since every gate of f translates to a size $\mathrm{poly}(\lambda, n)$ garbled table, and the subsequent degree 3 to player locality 3 compiler incurs a multiplicative $\mathrm{poly}(n)$ overhead. We also note that we assumed each player is allocated λ bits of randomness, independent of $|f|$ and even n—this is not an issue since we can extend random strings using a PRG.

8.2 Triple-Covering (t, n)-Private Sets

Given $t, n : \mathbb{N} \to \mathbb{N}$ such that $t(\lambda) \leq n(\lambda) \leq \text{poly}(\lambda)$, we would like to construct n sets $S_1, \ldots, S_n \subseteq [m]$, for some $m(\lambda) \leq \text{poly}(\lambda)$, with the following two properties:

- Triple-Covering: for every triple of elements $\{i_1, i_2, i_3\} \subseteq [m]$, there exists some index $j \in [n]$ such that $\{i_1, i_2, i_3\} \subseteq S_j$.
- t-Privacy: for any subset of indices $T \subseteq [n]$ of size $|T| < t$, the union of their sets does not cover all elements $\bigcup_{j \in T} S_j \subsetneq [m]$.

We start by describing a straightforward construction for any $t \leq \text{poly}(\lambda)$ and $n \geq \binom{3t}{3}$. Let $m := 3t$, and consider all triples over $[m]$. By definition, the triple-covering property holds. Also, t-privacy holds since the union of fewer than t subsets is of size strictly smaller than m. (We remark that n can always be increased by duplicating existing sets.)

Next, in order to reduce the ratio of n/t, we focus on $t = O(\log(\lambda))$ and $n = 3t$.[14] We define the universe as

$$m := |\{V \subseteq [n] : |V| = n - t + 1\}| = \text{poly}(\lambda)\,,$$

and identify each element $i \in [m]$ with a unique subset $V_i \subseteq [n]$ of size $n - t + 1$. We define our sets via a bipartite graph where we connect $i \in [m]$ and $j \in [n]$ whenever $j \in V_i$. That is to say, for $j \in [n]$ define $S_j := \{i \in [m] \mid j \in V_i\}$. We now verify that this construction satisfies both required properties:

- Triple-Covering: for every $\{i_1, i_2, i_3\} \subseteq [m]$, there exists some j covering $\{i_1, i_2, i_3\} \subseteq S_j$, since the intersection of three subsets of density greater than $2/3$ cannot be empty. Specifically, let V_1, V_2, V_3 denote the subsets corresponding to i_1, i_2, i_3. We have that

$$\begin{aligned}\left|\overline{V_1} \cup \overline{V_2} \cup \overline{V_3}\right| &\leq \left|\overline{V_1}\right| + \left|\overline{V_2}\right| + \left|\overline{V_3}\right| = 3(t-1) < n\,,\\ |V_1 \cap V_2 \cap V_3| &= n - \left|\overline{V_1} \cup \overline{V_2} \cup \overline{V_3}\right| > 0\,.\end{aligned}$$

  Therefore, there exists some $j \in [n]$ such that $j \in V_1 \cap V_2 \cap V_3$, and since neighborhood is symmetric $\{i_1, i_2, i_3\} \subseteq S_j$.

- t-Privacy: given any $T \subseteq [n]$ of size $|T| < t$, we have $\left|\overline{T}\right| \geq n - t + 1$, so there exists some $i \in [m]$ such that $V_i \subseteq \overline{T}$. Thus, i is not connected to any $j \in T$, and we have $\bigcup_{j \in T} S_j \subseteq [m] \setminus \{i\}$.

8.3 The HSS Construction

We instantiate HSS by combining triple-covering sets with TMPRE; see Fig. 4 for the construction. Perfect correctness is straightforward as $f^{\oplus}(z_1, \ldots, z_m) = f(x)$, the TMPRE has perfect correctness, and the triple-covering property ensures we cover all $(\hat{y}_{i,j,k})_{i,j,k \in [m]}$.

For semi-malicious t-privacy, let us first recall the security definition (see Definition 10):

1. $\mathsf{HSS.Share}$ is t-wise input-independent, namely for every security parameter $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^{\ell}$ and subset $T \subseteq [n]$ of size $|T| < t$:

$$(x_j)_{j \in T} \equiv \left(x_j^*\right)_{j \in T}\,,$$

   over $(x_j)_{j \in [n]} \leftarrow \mathsf{HSS.Share}(1^{\lambda}, x)$ and $\left(x_j^*\right)_{j \in [n]} \leftarrow \mathsf{HSS.Share}(1^{\lambda}, 0^{\ell})$.

[14] Can be reduced to $n = 3t - 2$.

Given triple-covering (t, n)-private sets $S_1, \ldots, S_n \subseteq [m]$ and semi-malicious TMPRE, the HSS construction is defined below:

$\underline{(x_1, \ldots, x_n) \leftarrow \mathsf{HSS.Share}(1^\lambda, x)}$

1. For $i \in [m-1]$ sample $z_i \leftarrow \{0,1\}^{|x|}$, and set $z_m = x \bigoplus_{i \in [m-1]} z_i$.
2. For $i \in [m]$ sample $r_i \leftarrow \{0,1\}^\lambda$.
3. For $j \in [n]$ set $x_j := (z_i, r_i)_{i \in S_j}$.
4. Output $(x_1, \ldots, x_n)$.

$\underline{y_j := \mathsf{HSS.Eval}(1^\lambda, f, j, x_j)}$

1. Let $f^\oplus(z_1, \ldots, z_m) := f(\bigoplus_{i \in [m]} z_i)$, and compile
$$\left(\hat{f}_{i_1,i_2,i_3}\right)_{i_1,i_2,i_3 \in [m]} := \mathsf{TMPRE.Comp}(1^\lambda, f^\oplus)\,.$$
2. Output
$$y_j := \left(\hat{f}_{i_1,i_2,i_3}((z_{i_1}, r_{i_1}), (z_{i_2}, r_{i_2}), (z_{i_3}, r_{i_3}))\right)_{i_1,i_2,i_3 \in S_j}\,.$$

$\underline{y := \mathsf{HSS.Dec}(1^\lambda, f, y_1, \ldots, y_n)}$

1. Combine
$$\hat{y} := \bigcup_{j \in [n]} (\hat{y}_{i_1,i_2,i_3})_{i_1,i_2,i_3 \in S_j}\,.$$
2. Output $y := \mathsf{TMPRE.Dec}(1^\lambda, f^\oplus, \hat{y})$.

Figure 4: Homomorphic Secret Sharing

Given triple-covering (t, n)-private sets $S_1, \ldots, S_n \subseteq [m]$ and semi-malicious TMPRE, the HSS simulator is defined below:

$\underline{(y_1^*, \ldots, y_n^*) \leftarrow \mathsf{HSS.Sim}(1^\lambda, f, T, x_T, y)}$

1. Let $S := \bigcup_{j \in T} S_j$, and set $(z_i, r_i)_{i \in S}$ according to $(x_j)_{j \in T}$.
2. Call
$$\left(\hat{y}^*_{i,j,k}\right)_{i,j,k \in [m]} \leftarrow \mathsf{TMPRE.Sim}(1^\lambda, f^\oplus, S, z_S, r_S, y)\,.$$
3. For all $j \in [n]$, set
$$y_j^* := \left(\hat{y}^*_{i_1,i_2,i_3}\right)_{i_1,i_2,i_3 \in S_j}\,.$$
4. Output $(y_1^*, \ldots, y_n^*)$.

Figure 5: HSS Simulator

2. There exists an efficient randomized simulator $\mathsf{HSS.Sim}$ such that for every ensemble $\{f, x, T, x_T\}_\lambda$, where f is a function with input length ℓ, $x \in \{0,1\}^\ell$, $T \subseteq [n]$ with $|T| < t$ and x_T are valid shares $(x_j)_{j \in T} \in \mathrm{Supp}(\mathsf{HSS.Share}(1^\lambda, x)_T)$:
$$(y_1, \ldots, y_n) \overset{\mathsf{c}}{\approx} \mathsf{HSS.Sim}(1^\lambda, f, T, x_T, f(x))\,,$$
over conditional input sharing $(x_1, \ldots, x_n) \leftarrow (\mathsf{HSS.Share}(1^\lambda, x) \mid x_T)$ and the randomness of the simulator, where $\forall j \in [n] : y_j := \mathsf{HSS.Eval}(1^\lambda, f, j, x_j)$.

We are given any subset $T \subseteq [n]$ of size $|T| < t$, and let $S := \bigcup_{j \in T} S_j \subsetneq [m]$. We first observe that $\mathsf{HSS.Share}$ is t-wise input-independent: the shares $(x_j)_{j \in T}$ correspond to $(z_i, r_i)_{i \in S}$ which are completely independent of the input x, as additive secret sharing is $(m-1)$-private.

Next, the simulator is described in Fig. 5. Setting $(z_i, r_i)_{i \in S}$ according to $(x_j)_{j \in T}$ is well-defined and consistent with all shares, since x_T are guaranteed to be valid shares. In more detail, to derive (z_i, r_i) for any $i \in S$, we arbitrarily pick any $j \in T$ such that $i \in S_j$, parse $x_j = (z_k, r_k)_{k \in S_j}$, and derive the corresponding (z_i, r_i). While an index $i \in [m]$ may appear in multiple subsets, this extraction is well-defined. Indeed, we are guaranteed that the tuple (z_i, r_i) is consistent across all x_j that contain it, since x_T are valid shares (i.e., $(x_j)_{j \in T} \in \mathrm{Supp}(\mathsf{HSS.Share}(1^\lambda, x)_T)$). The conditional input sharing $(\mathsf{HSS.Share}(1^\lambda, x) \mid x_T)$ simply means we sample $r_{\overline{S}}$ at random, and $z_{\overline{S}}$ conditioned on $\bigoplus_{i \in [m]} z_i = x$. The TMPRE simulation works against $(r_i \leftarrow \{0,1\}^\lambda : i \notin S)$ for every fixed choice of $(z_i)_{i \in [m]}$ consistent with $y = f^\oplus(z_1, \ldots, z_m)$, and in particular over our sampling process.

Acknowledgments

We thank Eliran Kachlon for helpful discussions at early stages of this project. We thank Aayush Jain for bringing the problem of FE amplification to our attention and for initial discussions that motivated this work.

References

[ABG+20] Benny Applebaum, Zvika Brakerski, Sanjam Garg, Yuval Ishai, and Akshayaram Srinivasan. Separating two-round secure computation from oblivious transfer. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 71:1–71:18. LIPIcs, January 2020. 37

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 657–677. Springer, Berlin, Heidelberg, August 2015. 29

[ABT18] Benny Applebaum, Zvika Brakerski, and Rotem Tsabary. Perfect secure computation in two rounds. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 152–174. Springer, Cham, November 2018. 12, 36, 37

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004. 36

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Berlin, Heidelberg, August 2015. 30

[AJL+19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 284–332. Springer, Cham, August 2019. 5, 30

[AJN+16] Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 491–520. Springer, Berlin, Heidelberg, August 2016. 3

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation from functional encryption for simple functions. Cryptology ePrint Archive, Report 2015/730, 2015. 29

[AJS18] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. Cryptology ePrint Archive, Report 2018/615, 2018. 5, 30

[AK25a] Benny Applebaum and Eliran Kachlon. How to share an NP statement or combiners for zero-knowledge proofs. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 480–513. Springer, Cham, August 2025. 3

[AK25b] Benny Applebaum and Eliran Kachlon. NIZK amplification via leakage-resilient secure computation. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 462–479. Springer, Cham, August 2025. 3, 35

[BG24] Nir Bitansky and Nathan Geier. Amplification of non-interactive zero knowledge, revisited. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IX*, volume 14928 of *LNCS*, pages 361–390. Springer, Cham, August 2024. 3, 10, 21, 35

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539. Springer, Berlin, Heidelberg, August 2016. 11, 30

[BGI+18] Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of homomorphic secret sharing. In Anna R. Karlin, editor, *ITCS 2018*, volume 94, pages 21:1–21:21. LIPIcs, January 2018. 11, 30

[BKP+24] Nir Bitansky, Chethan Kamath, Omer Paneth, Ron D. Rothblum, and Prashant Nalini Vasudevan. Batch proofs are statistically hiding. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *56th ACM STOC*, pages 435–443. ACM Press, June 2024. 3

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd ACM STOC*, pages 503–513. ACM Press, May 1990. 37

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Berlin, Heidelberg, March 2011. 5

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015. 29, 30

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 17–33. Springer, Berlin, Heidelberg, February 2005. 3, 28

[CK90] Claude Crépeau and Joe Kilian. Weakening security assumptions and oblivious transfer (abstract). In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 2–7. Springer, New York, August 1990. 3, 35

[DI05] Ivan Damgård and Yuval Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 378–394. Springer, Berlin, Heidelberg, August 2005. 37

[DIJ+13] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 519–535. Springer, Berlin, Heidelberg, August 2013. 34

[DKS99] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 56–73. Springer, Berlin, Heidelberg, May 1999. 3, 35

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 342–360. Springer, Berlin, Heidelberg, May 2004. 3, 4

[FHNS16] Marc Fischlin, Amir Herzberg, Hod Bin Noon, and Haya Shulman. Obfuscation combiners. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 521–550. Springer, Berlin, Heidelberg, August 2016. 3

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013. 30

[GIS18] Sanjam Garg, Yuval Ishai, and Akshayaram Srinivasan. Two-round MPC: Information-theoretic and black-box. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 123–151. Springer, Cham, November 2018. 37

[GJS19] Vipul Goyal, Aayush Jain, and Amit Sahai. Simultaneous amplification: The case of non-interactive zero-knowledge. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 608–637. Springer, Cham, August 2019. 3, 35

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 273–301. Springer, 2011. 3

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. 3

[GS16] Sanjam Garg and Akshayaram Srinivasan. Single-key to multi-key functional encryption with polynomial loss. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 419–442. Springer, Berlin, Heidelberg, October / November 2016. 28, 29

[Her05] Amir Herzberg. On tolerant cryptographic constructions. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 172–190. Springer, Berlin, Heidelberg, February 2005. 3

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 393–411. Springer, Berlin, Heidelberg, March 2008. 3

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 96–113. Springer, Berlin, Heidelberg, May 2005. 3

[Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 664–673. ACM Press, May 2005. 3, 10

[HR05] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 478–493. Springer, Berlin, Heidelberg, August 2005. 3, 4

[HR08] Shai Halevi and Tal Rabin. Degradation and amplification of computational hardness. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 626–643. Springer, Berlin, Heidelberg, March 2008. 10, 28

[HS11] Thomas Holenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles (extended abstract). In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 19–36. Springer, Berlin, Heidelberg, March 2011. 28

[IJK09] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. *Journal of Cryptology*, 22(1):75–92, January 2009. 28

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000. 36

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256. Springer, Berlin, Heidelberg, July 2002. 36

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th FOCS*, pages 538–545. IEEE Computer Society Press, October 1995. 10

[Jai26] Aayush Jain. Author's note on "Amplifying the Security of Functional Encryption, Unconditionally". Personal website, 2026. https://sites.google.com/view/aayushjain/home. 5

[JKMS20] Aayush Jain, Alexis Korb, Nathan Manohar, and Amit Sahai. Amplifying the security of functional encryption, unconditionally. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 717–746. Springer, Cham, August 2020. 5, 7, 11, 12, 13, 28, 34, 35

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Cham, May / June 2022. 30

[JMS20] Aayush Jain, Nathan Manohar, and Amit Sahai. Combiners for functional encryption, unconditionally. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 141–168. Springer, Cham, May 2020. 3, 11, 28

[Jut10] Charanjit S. Jutla. Almost optimal bounds for direct product threshold theorem. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 37–51. Springer, Berlin, Heidelberg, February 2010. 28

[KNT18] Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Obfustopia built on secret-key functional encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 603–648. Springer, Cham, April / May 2018. 30

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Cham, August 2019. 29

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 443–468. Springer, Berlin, Heidelberg, October / November 2016. 29

[LM20] David Lanzenberger and Ueli Maurer. Coupling of random systems. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 207–240. Springer, Cham, November 2020. 4, 8, 9, 16

[LT13] Huijia Lin and Stefano Tessaro. Amplification of chosen-ciphertext security. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 503–519. Springer, Berlin, Heidelberg, May 2013. 3

[Mau02] Ueli M. Maurer. Indistinguishability of random systems. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, Berlin, Heidelberg, April / May 2002. 14

[MPR07] Ueli M. Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 130–149. Springer, Berlin, Heidelberg, August 2007. 4, 9

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *ICS 2011*, pages 1–21. Tsinghua University Press, January 2011. 14

[MT09] Ueli M. Maurer and Stefano Tessaro. Computational indistinguishability amplification: Tight product theorems for system composition. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 355–373. Springer, Berlin, Heidelberg, August 2009. 4, 9, 27

[MT10] Ueli M. Maurer and Stefano Tessaro. A hardcore lemma for computational indistinguishability: Security amplification for arbitrarily weak PRGs with optimal stretch. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 237–254. Springer, Berlin, Heidelberg, February 2010. 10, 21

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. 5

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 463–472. ACM Press, October 2010. 11

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005. 5

[Tes10] Stefano Tessaro. *Computational Indistinguishability Amplification*. PhD thesis, ETH Zurich, Zürich, Switzerland, 2010. 9

[Tes11] Stefano Tessaro. Security amplification for the cascade of arbitrarily weak PRPs: Tight bounds via the interactive hardcore lemma. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 37–54. Springer, Berlin, Heidelberg, March 2011. 10, 27, 28

[Wul07] Jürg Wullschleger. Oblivious-transfer amplification. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 555–572. Springer, Berlin, Heidelberg, May 2007. 3, 35

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. 3, 29

A Indistinguishability to Simulation Security for Weak FE

Definition 12 (Secret-Key Encryption)**.** A secret-key encryption scheme SKE for message space $\{0,1\}^*$ consists of two PPT algorithms ($\mathsf{SKE.Enc}$, $\mathsf{SKE.Dec}$) with the following syntax:

- $\mathsf{CT} \leftarrow \mathsf{SKE.Enc}(r, x)$: takes as input the secret key $r \in \{0,1\}^\lambda$ and a message $x \in \{0,1\}^*$, and outputs a ciphertext.
- $\hat{x} \leftarrow \mathsf{SKE.Dec}(r, \mathsf{CT})$: takes as input the secret key and a ciphertext, and outputs a message.

We next define the relevant properties:

- Correctness: for all $\lambda \in \mathbb{N}$, secret key $r \in \{0,1\}^\lambda$ and message $x \in \{0,1\}^*$, we have
$$x = \mathsf{SKE.Dec}(r, \mathsf{SKE.Enc}(r, x))\,.$$
- Security: for every $\{x_0, x_1\}_\lambda$ with $|x_0| = |x_1| \leq \mathrm{poly}(\lambda)$, we have
$$\mathsf{CT}_0 \overset{\mathsf{c}}{\approx} \mathsf{CT}_1\,,$$
over $r \leftarrow \{0,1\}^\lambda, \mathsf{CT}_b \leftarrow \mathsf{SKE.Enc}(r, x_b)$.

Definition 13 (Trapdoor Circuit)**.** For a circuit $f : \{0,1\}^\ell \to \{0,1\}^{\ell'}$ and secret-key encryption scheme SKE, the trapdoor circuit $f^* : \{0,1\}^\ell \times \{0,1\} \times \{0,1\}^\lambda \to \{0,1\}^{\ell'}$ with hardcoded ciphertext CT (of a size ℓ' message) works as follows:

$$f^*[\mathsf{CT}](x, \beta, K) := \begin{cases} f(x) & \beta = 0 \\ \mathsf{SKE.Dec}(K, \mathsf{CT}) & \beta = 1 \end{cases}$$

Lemma 6. *Let FE' be a δ-IND-secure functional encryption scheme. Then, the scheme FE described in Fig. 6 is a $(\delta + \mathrm{negl})$-SIM-secure functional encryption scheme, with respect to the simulator described in Fig. 7.*

Given an IND-secure functional encryption scheme FE' and secret key encryption scheme SKE, the SIM-secure construction FE works as follows:

$\underline{(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^p)}$

1. Call $(\mathsf{MPK}', \mathsf{MSK}') \leftarrow \mathsf{FE.Setup}'(1^\lambda, 1^{p'})$, where $p' \leq p \cdot \mathrm{poly}(\lambda)$ is an upper bound on the trapdoor circuit size.
2. Sample a key for the SKE scheme $K \leftarrow \{0,1\}^\lambda$.
3. Output $\mathsf{MPK} := \mathsf{MPK}'$, $\mathsf{MSK} := (\mathsf{MSK}', K)$.

$\underline{\mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{MPK}, x)}$

1. Call $\mathsf{CT}' \leftarrow \mathsf{FE.Enc}'(\mathsf{MPK}', (x, 0, 0^\lambda))$.
2. Output $\mathsf{CT} := \mathsf{CT}'$.

$\underline{\mathsf{FSK}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{MSK}, f)}$

1. Call $\mathsf{CT}_0 \leftarrow \mathsf{SKE.Enc}(K, 0^{\ell'})$.
2. Call $\mathsf{FSK}'_{f^*} \leftarrow \mathsf{FE.Keygen}'(\mathsf{MSK}', f^*[\mathsf{CT}_0])$.
3. Output $\mathsf{FSK}_f := \mathsf{FSK}'_{f^*}$.

$\underline{y \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})}$

1. Call $y' \leftarrow \mathsf{FE.Dec}'(\mathsf{FSK}'_{f^*}, \mathsf{CT}')$.
2. Output $y := y'$.

Figure 6: FE IND-to-Sim

Given an IND-secure functional encryption scheme FE' and secret key encryption scheme SKE, the simulator $\mathsf{FE.Sim}$ works as follows:

$(\mathsf{MPK}, \mathsf{FSK}_f, \mathsf{CT}) \leftarrow \mathsf{FE.Sim}(1^\lambda, 1^p, f, y)$

1. Call $(\mathsf{MPK}', \mathsf{MSK}') \leftarrow \mathsf{FE.Setup}'(1^\lambda, 1^{p'})$, where $p' \leq p \cdot \mathrm{poly}(\lambda)$ is an upper bound on the trapdoor circuit size.
2. Sample a key for the SKE scheme $K \leftarrow \{0,1\}^\lambda$.
3. Call $\mathsf{CT}' \leftarrow \mathsf{FE.Enc}'(\mathsf{MPK}', (0^\ell, 1, K))$.
   First change: $\mathsf{CT}' \leftarrow \mathsf{FE.Enc}'(\mathsf{MPK}', (x, 0, 0^\lambda))$.
4. Call $\mathsf{CT}_y \leftarrow \mathsf{SKE.Enc}(K, y)$.
   Second change: $\mathsf{CT}_y \leftarrow \mathsf{SKE.Enc}(K, 0^{\ell'})$.
5. Call $\mathsf{FSK}'_{f^*} \leftarrow \mathsf{FE.Keygen}'(\mathsf{MSK}', f^*[\mathsf{CT}_y])$.
6. Output $(\mathsf{MPK}', \mathsf{FSK}'_{f^*}, \mathsf{CT}')$.

Figure 7: FE Simulator

Proof Sketch. Correctness is straightforward due to the definition of the trapdoor function f^* in case $\beta = 0$. Succinctness is due to the fact that the trapdoor function f^* does not incur too much of an overhead, namely $\mathrm{size}(f^*) \leq \mathrm{size}(f) \cdot \mathrm{poly}(\lambda)$.

For SIM-security, we are given $\{x, f, p\}_\lambda$ with $|f| \leq p = \mathrm{poly}(\lambda)$, and denote $y = f(x)$. The real execution differs from the simulation in two components: (1) the FE ciphertext encrypts $(x, 0, 0^\lambda)$ instead of $(0^\ell, 1, K)$, and (2) the hardcoded SKE ciphertext in f^* encrypts $0^{\ell'}$ instead of y. Denote by H_0 the simulated distribution, by H_1 the hybrid with (1) switched, and by H_2 the real execution where (1) and (2) are switched. Then $H_0 \overset{\mathsf{c}}{\approx}_\delta H_1$ due to the δ-IND-security of FE', since $f^*[\mathsf{CT}_y](x, 0, 0^\lambda) = f^*[\mathsf{CT}_y](0^\ell, 1, K)$. Also, $H_1 \overset{\mathsf{c}}{\approx} H_2$ due to the SKE security $\mathsf{SKE.Enc}(K, y) \overset{\mathsf{c}}{\approx} \mathsf{SKE.Enc}(K, 0^{\ell'})$, as we removed the dependency on K in change (1). We conclude that $H_0 \overset{\mathsf{c}}{\approx}_{\delta+\mathrm{negl}} H_2$. Thus, we have shown that FE is $(\delta + \mathrm{negl})$-SIM-secure. □

B Amplification of Subexp Functional Encryption

In this section, we outline the generalization of our techniques from Section 6 to the subexp setting. That is, instead of security against polynomial-size adversaries we consider security against subexponential-size adversaries, and we aim to achieve *subexponentially-small error* instead of negligible. Recall that for two ensembles $\{X\}_\lambda, \{Y\}_\lambda$ and functions $s : \mathbb{N} \to \mathbb{N}$, $\varepsilon : \mathbb{N} \to [0,1]$, we write $X \overset{\mathsf{c}}{\approx}_{s,\varepsilon} Y$ if for all large enough $\lambda \in \mathbb{N}$, $X_\lambda \overset{\mathsf{c}}{\approx}_{s(\lambda),\varepsilon(\lambda)} Y_\lambda$. In this section, we omit the bound s on the size when $s(\lambda) \geq 2^{\lambda^c}$ for some constant $c > 0$, and the bound ε on the error when $\varepsilon(\lambda) \leq 2^{-\lambda^c}$.

For the most part, the generalization is straightforward, by instantiating the computational parts (HSS and SIM-to-IND) using $(\mathrm{subexp}, 1/\mathrm{subexp})$ OWFs instead of $(\mathrm{poly}, \mathrm{negl})$. We note that $(\mathrm{subexp}, \mathrm{const})$ FE implies such OWFs. The main difference is that we now need to also use a

"heavy" combiner, to bring the error down from negl (or 1/poly) to 1/subexp.

Subexponential Definitions. We define subexp FE and subexp HSS identically to Definitions 9 and 10, maintaining the exact same syntax, correctness, and succinctness. The only change is that all security and privacy properties (SIM/IND-FE-security and HSS simulation) are upgraded to the subexponential regime as described above. That is, the default size bound is upgraded from $\mathrm{poly}(\lambda)$ to $\mathrm{subexp}(\lambda)$, and the default error bound is upgraded from $\mathrm{negl}(\lambda)$ to $1/\mathrm{subexp}(\lambda)$.

Generalizing Section 6. The following theorem is a straightforward generalization of Theorem 3 to the subexp regime, by utilizing a subexp HSS scheme. We do not repeat the proof as it is identical to the original version, by modifying the computational indistinguishability notion from $(\mathrm{poly}, \mathrm{negl})$ to $(\mathrm{subexp}, 1/\mathrm{subexp})$ as described above. We remark that we modified $\mathrm{size}(\vec{F}^*) \leq 2^{\lambda^{o(1)}}$ instead of $\leq \mathrm{poly}(\lambda)$ because the combiner can handle that (due to subexp HSS security), but it is not actually needed for FE amplification.

Theorem 7. *Given a subexp (t,n)-HSS and $\{x, x', f\}_\lambda$ such that $f : \{0,1\}^\ell \to \{0,1\}^m$, $x, x' \in \{0,1\}^\ell$, $f(x) = f(x')$ and $|f| \leq \mathrm{poly}(\lambda)$, the n-oracle-aided pair $\left\{C^{(\cdot)}_{x,f}, C^{(\cdot)}_{x',f}\right\}_\lambda$ described in Fig. 2, is a $(\mathrm{subexp},\ 1/\mathrm{subexp},\ n-t+1,\ n)$ indistinguishability combiner w.r.t. any ideal $\left\{\vec{F}^*\right\}_\lambda$ such that $\mathrm{size}(\vec{F}^*) \leq 2^{\lambda^{o(1)}}$ and each $F^*_i(\cdot,\cdot)$ ignores its first input.*

The next theorem is similar to Theorem 4, but it is more general as we want to construct both a "light" and a "heavy" combiner: one for $(\mathrm{const} \to 1/\mathrm{poly})$ and the other for $(1/\mathrm{poly} \to 1/\mathrm{subexp})$.

Theorem 8. *Let FE^w be a δ-SIM-secure subexp functional encryption scheme for $\delta < 0.5$. Then, given a subexp (t,n)-HSS, the scheme FE described in Fig. 1 is a $(\delta' + \varepsilon' + 1/\mathrm{subexp})$-IND-secure subexp functional encryption scheme, where*

$$\delta' := \binom{n}{t}(2\delta)^t \leq (2\delta \cdot en/t)^t, \quad \varepsilon' := (1-2\delta)^{-n+t-1}/\mathrm{subexp}\,.$$

Proof. Succinctness is due to the succinctness of the weak FE and the fact that $\mathsf{HSS.Eval}$ incurs only a $\mathrm{poly}(\lambda)$ overhead. Correctness follows from the weak FE and HSS correctness.

In order to show functional indistinguishability, consider $\{x, x', f, p\}_\lambda$ with $\ell := |x| = |x'|, y := f(x) = f(x')$, and $|f| \leq p = \mathrm{poly}(\lambda)$. We omit the security parameter subscript for ease of notation. For $i \in [n]$, consider the ideal functionalities $F^*_i(a_1, a_2) := \mathsf{FE.Sim}^w(1^\lambda, 1^{p'}, f_i, a_2)$, where p' and f_i are defined in Fig. 1. Since F^*_i is efficient and the first input is ignored, the conditions of Theorem 7 are satisfied, and we deduce that $(C^{(\cdot)}_{x,f}, C^{(\cdot)}_{x',f})$ is a $(\mathrm{subexp},\ 1/\mathrm{subexp},\ n-t+1,\ n)$ indistinguishability combiner w.r.t. $\vec{F}^*$.

Now, let $F'_i(a_1, a_2)$ be the real functionality that is defined according to the SIM-security challenge, namely it returns $(\mathsf{MPK}_i, \mathsf{FSK}_{f_i}, \mathsf{CT}_i)$ given by $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{FE.Setup}^w(1^\lambda, 1^{p'}), \mathsf{FSK}_{f_i} \leftarrow \mathsf{FE.Keygen}^w(\mathsf{MSK}_i, f_i), \mathsf{CT}_i \leftarrow \mathsf{FE.Enc}^w(\mathsf{MPK}_i, a_1)$. In case its inputs are inconsistent with our combiner's queries, namely $a_2 \neq f_i(a_1)$, it simply returns $F^*_i(a_1, a_2)$ so there is no distinguishing advantage at all. We observe that $C^{\vec{F}'}_{x,f}$ and $C^{\vec{F}'}_{x',f}$ correspond to the IND-security game of FE, up to formatting between $(\mathsf{MPK}_i, \mathsf{FSK}_{f_i}, \mathsf{CT}_i)_{i=1}^n$ and $((\mathsf{MPK}_i)_{i=1}^n, (\mathsf{FSK}_{f_i})_{i=1}^n, (\mathsf{CT}_i)_{i=1}^n)$.

Finally, since $\forall i \in [n] : F'_i \overset{\mathsf{c}}{\approx}_\delta F^*_i$ due to the δ-SIM-security of FE^w, we can apply Theorem 2 and get that $C^{\vec{F}'}_{x,f} \overset{\mathsf{c}}{\approx}_{\delta'+\varepsilon'+1/\mathrm{subexp}} C^{\vec{F}'}_{x',f}$ for $\delta' := \binom{n}{t}(2\delta)^t$, $\varepsilon' := (1-2\delta)^{-n+t-1}/\mathrm{subexp}$, and a slackness

parameter γ that is subexponentially small. In more detail, let c be such that 2^{λ^c} is a valid size bound for both the weak subexp FE and the combiner. We need to ensure that

$$s := \frac{2^{\lambda^c} \cdot \gamma^2}{\mathrm{poly}(\lambda)} - \mathrm{poly}(\lambda)$$

is subexponentially large, so we may choose $\gamma = 2^{-\lambda^c/3}$. □

We note that by instantiating the generic transformation from IND-security to SIM-security (Lemma 6) using subexp SKE (from subexp OWFs), it generalizes to the subexp setting. That is, it transforms a δ-IND-secure subexp functional encryption scheme into a $(\delta+1/\mathrm{subexp})$-SIM-secure subexp functional encryption scheme.

Theorem (Restating Theorem 6)**.** *Assuming the existence of a δ-IND-secure subexp functional encryption scheme for some constant $\delta < 1/(6e)$, there also exists a $(1/\mathrm{subexp})$-IND-secure subexp functional encryption scheme.*

Proof. First, we note that weak subexp FE implies subexp OWFs, which in turn imply subexp SKE and subexp (t, n)-HSS for any $t \le \mathrm{poly}(\lambda)$ and $n \ge \binom{3t}{3}$, or for $n \ge 3t$ if $t = O(\log(\lambda))$. We use the subexp IND-security to SIM-security transformation (Lemma 6) to go from δ-IND-secure subexp FE to $(\delta + 1/\mathrm{subexp})$-SIM-secure subexp FE. Next, we use the subexp FE amplifier (Theorem 8) with a subexp $(\log\lambda, 3\log\lambda)$-HSS to obtain a $(1/\mathrm{poly})$-IND-secure subexp FE, as $(2\delta \cdot en/t)^t = 1/\mathrm{poly}$ and $(1-2\delta)^{-n+t-1}/\mathrm{subexp} = \mathrm{poly}/\mathrm{subexp} = 1/\mathrm{subexp}$.

We again use the subexp IND-security to SIM-security transformation to obtain $(1/\mathrm{poly})$-SIM-secure subexp FE, say λ^{-c} for some constant $c > 0$. Finally, we use the subexp FE amplifier with a subexp $(\lambda^{c/3}, 5\lambda^c)$-HSS to obtain a $(1/\mathrm{subexp})$-IND-secure subexp FE, as $(2\lambda^{-c} \cdot en/t)^t = (\mathrm{const}/\mathrm{poly})^{\mathrm{poly}} = 1/\mathrm{subexp}$ and $(1-2\lambda^{-c})^{-n+t-1} \le (1-2\lambda^{-c})^{-5\lambda^c} \le \mathrm{const}$. □

C Converters and Resources

Definition (Resource; Definition 2)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h \in \mathbb{N}$, an h-round $(\mathcal{X}, \mathcal{Y})$-resource is an $(h, \mathcal{X}, \mathcal{Y})$ alternating tree where every *odd-depth* node v is assigned a distribution $D_v : \mathcal{Y} \to [0, 1]$.

Definition (Converter; Definition 4)**.** For finite sets $\mathcal{X}, \mathcal{Y}$ and $h, n \in \mathbb{N}$, an h-round n-oracle-aided $(\mathcal{X}, \mathcal{Y})$-converter is an h-round $(\mathcal{X}, \mathcal{Y}')$-resource for $\mathcal{Y}' := (\{0\} \cup [n]) \times \mathcal{Y}$.

A 0-oracle-aided converter corresponds to a plain $(\mathcal{X}, \mathcal{Y})$-resource. Given an index $k \in [n]$ and a $(\mathcal{Y}, \mathcal{X})$-resource S_k, we naturally obtain an $(n-1)$-oracle-aided converter:

For every even-depth node v with converter-label (k, y) for some $y \in \mathcal{Y}$, we assign a distribution over $\mathcal{X}$ according to S_k. Specifically, let $x_1, (i_1, y_1), x_2, (i_2, y_2), \ldots, x_\ell, (i_\ell, y_\ell)$ (where $i_\ell = k$) denote the path from the root to v. We filter this path to the subsequence y_j, x_{j+1} for all $j < \ell$ such that $i_j = k$, appended with y_ℓ. This defines the parallel path from the root in S_k to a node v^*, and we assign the distribution of v^* to v.

After assigning distributions to these even-depth nodes, we collapse the tree to bypass the internal queries to S_k. For a parent odd-depth node v, we absorb its probability of querying k by distributing that weight to its descendants. That is, for every child (i, y) where $i \neq k$, we update its probability to reflect both the direct choice and the probability of reaching it by passing through k:

$$D'_v(i, y) = D_v(i, y) + \sum_{y' \in \mathcal{Y}} \sum_{x \in \mathcal{X}} D_v(k, y') \cdot D_{v_{(k,y')}}(x) \cdot D_{v_{(k,y'),x}}(i, y)\,.$$

We do this bottom-up to ensure that the grandchild $v_{(k,y'),x}$ does not have any remaining $(k, \cdot)$ weights, which guarantees that v won't either. Finally, since collapsing bypasses two levels (the (k, y') query and x answer) for each internal call, any shortened paths are naturally padded with $\perp$ to maintain the $2h$ height.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il