Separating Selective Opening Security From Standard Security, Assuming IO

Justin Holmgren* Brent Waters†

Abstract

Assuming the hardness of LWE and the existence of IO, we construct a public-key encryption scheme that is IND-CCA secure but fails to satisfy even a weak notion of indistinguishability security with respect to selective opening attacks. Prior to our work, such a separation was known only from stronger assumptions such as differing inputs obfuscation (Hofheinz, Rao, and Wichs, PKC 2016).

Central to our separation is a new hash family, which may be of independent interest. Specifically, for any $S(\lambda) = \lambda^{O(1)}$, any $n(\lambda) = \lambda^{O(1)}$, and any $m(\lambda) = \lambda^{\Theta(1)}$, we construct a hash family mapping $n(\lambda)$ bits to $m(\lambda)$ bits that is *somewhere statistically* correlation intractable (SS-CI) for all relations $R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}$ that are *enumerable* by circuits of size $S(\lambda)$.

We give two constructions of such a hash family. Our first construction uses IO, and generically "boosts" any hash family that is SS-CI for the smaller class of *functions* that are *computable* by circuits of size $S(\lambda)$. This weaker hash variant can be constructed based solely on LWE (Peikert and Shiehian, CRYPTO 2019). Our second construction is based on the existence of a circular secure FHE scheme, and follows the construction of Canetti et al. (STOC 2019).

1 Introduction

Defining the necessary security properties for public-key encryption is a subtle affair. While the standard notion of IND-CCA [RS91] (or sometimes the simpler and weaker notion of IND-CPA security [GM84]) is broadly accepted as the gold standard, there are situations in which it is clearly insufficient. One such scenario is implementing secure channels in the presence of adaptive corruptions, for example in a multi-party computation protocol [CFGN96]. In this scenario, the encryption scheme can be subjected to what is known as a *selective opening attack* (SOA).

In a selective opening attack, an adversary is given a public key pk along with n ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$, and can then choose a subset $I \subseteq [n]$. Then for each $i \in I$, the adversary is given the message m_i and randomness r_i such that ct_i is an encryption of m_i using randomness r_i. The surprisingly difficult question is: what does such an adversary learn about $(m_i)_{i \notin I}$?

Perhaps surprisingly, for many encryption schemes satisfying the standard IND-CCA notion of security, the answer is not "nothing"! Before we can explain the details and nuances of this claim, we must first define security against selective opening attacks. This is itself non-trivial, and there are two main definitional variants.

Simulation-Based SOA Security The more stringent notion is a simulation-based notion in the style of semantic security style [GM84], and was first described by Dwork, Naor, Reingold, and Stockmeyer [DNRS99] in the setting of commitment schemes. Later, this definition was adapted to public-key encryption by Bellare, Hofheinz, and Yilik [BHY09]. Loosely speaking, this definition requires that for every adversary, there is an efficient *simulator* that produces indistinguishable output, while only interacting with an idealized "black box" model of the encryption scheme. Specifically, the simulator chooses the set I without seeing any ciphertexts, and in response is given only the *messages* $(m_i)_{i \in I}$.

*NTT Research. E-mail: `justin.holmgren@ntt-research.com`.

†UT Austin and NTT Research. E-mail: `bwaters@cs.utexas.edu`.

Indistinguishability-Based SOA Security The second notion, which is strictly weaker than the former, is a game-based definition, called "indistinguishability under selective opening attacks with respect to conditionally resampleable distributions" (abbreviated IND-SOA-CRS). This notion requires that any computationally bounded adversary can only win the following game (played against a "challenger") with probability negligibly close to 1/2.

The adversary first specifies a distribution $\mathcal{M}$ on vectors of messages, and the challenger samples $\mathbf{m} = (m_1, \ldots, m_n)$ from this distribution. $\mathcal{M}$ is required to be efficiently sampleable, and also efficiently "conditionally re-sampleable" — we elaborate more on the latter requirement later. The challenger then generates encryptions $\mathsf{ct}_i := \mathsf{Enc}(m_i; r_i)$, where r_i denotes the randomness used in the encryption process, and sends $(\mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ to the adversary. After the adversary specifies a set $I \subseteq [n]$ and receives $\big((m_i, r_i)\big)_{i \in I}$, the challenger samples $\mathbf{m}'$ from the distribution $\mathcal{M}$ conditioned on $m'_i = m_i$ for all $i \in I$. The challenger finally sends either $\mathbf{m}'$ or $\mathbf{m}$ to the adversary, and the adversary wins if he can guess which of the two he received.

Relation To Other Assumptions In Cryptography A long line of work has sought to clarify how both variants of SOA security relate to IND-CPA and IND-CCA security, as well as to other standard assumptions in cryptography.

We briefly summarize the history of what is now known. It was first shown by [BHY09] that any "lossy" encryption scheme must be IND-SOA-CRS secure, and if the lossy encryption scheme additionally satisfies an "efficient openability" requirement, then it must also satisfy the stronger notion of SS-SOA security. On the other hand, [BDWY12] proved that if an encryption scheme satisfies a *binding* property antithetical to lossiness, then the encryption scheme cannot satisfy SS-SOA security.

A major gap in our understanding, and the focus of this work, is whether an encryption scheme satisfying IND-CPA (or even IND-CCA) security must also satisfy IND-SOA-CRS security. The most salient prior works on this are those of Hofheinz and Rupp [HR14] and Hofheinz, Rao, and Wichs [HRW16].

The work of [HR14] showed that IND-CCA security does not imply a strengthening of IND-SOA-CRS security, in which the adversary is allowed to make decryption queries analogous to those in the IND-CCA security game. At a high level, they add extra functionality to the decryption algorithm that, when made available to the adversary as an oracle, preserves IND-CCA security while destroying selective opening security.

Subsequently, the work of [HRW16] adapted these ideas to the plain notion of IND-SOA-CRS security — at a high level, they replace the oracle by an obfuscated program that comprises part of the public key. They thus obtain a conditional separation that relies on strong and non-standard assumptions. Specifically, they assumed the existence of:

- a hash family that is correlation intractable for a specific class $\mathcal{R}$ of binary relations (more in the technical overview). That is, for all relations $R \in \mathcal{R}$, it must be difficult given a random hash function H, to find an input x such that $(x, H(x)) \in R$.
- an encryption scheme PKE that is puncturable (a strong form of IND-CCA security). Such an encryption scheme was constructed from IO and one-way functions in [CHV15].
- public-coin differing inputs obfuscation (pc-diO) [IPS15]. This is a strong notion of security for a circuit obfuscator $\mathcal{O}$, stipulating that if it is hard to find an input on which a randomly sampled pair of circuits (C_0, C_1) differ, even given the coins used to sample those circuits, then $\big(C_0, C_1, \mathcal{O}(C_0)\big)$ should be indistinguishable from $\big(C_0, C_1, \mathcal{O}(C_1)\big)$.

After long lines of research on both correlation intractability ([KRR17, CCRR18, HL18, CCH+19, PS19, HLR21]) and indistinguishability obfuscation (too many papers to list, culminating in the seminal breakthrough of [JLS21, JLS22]), we now know how to construct both from well-studied cryptographic assumptions.

On the other hand, there is significant evidence *against* the existence of differing inputs obfuscation [BP15, GGHW17, BSW16]. While this evidence does not apply to the *public-coin* flavor of differing inputs obfuscation used in [HRW16], pc-diO still stands out as a qualitatively stronger assumption than the others, and one which is unlikely to be instantiable from a falsifiable assumption [Nao03, GK16].

In this work, we show how to reduce the assumptions of the separation to only indistinguishability obfuscation and LWE.

Correlation Intractable Hashing As an important tool in our separation, we construct hash families satisfying a strong form of correlation intractability, which we believe is the right "iO-friendly" notion — *somewhere statistical* [CCH+19] correlation intractability for any "efficiently enumerable" relation R.

By "somewhere statistical" correlation intractability, we mean that the hash H is indistinguishable from a hash H' for which there *does not exist* any x satisfying $\big(x, H'(x)\big) \in R$. This is clearly important in the context of an iO, where two programs need to be perfectly functionally equivalent for their obfuscations to indistinguishable.

By an "efficiently enumerable" relation R, we mean that there exists an efficient algorithm that on input x, outputs a small set containing all y for which $(x, y) \in R$. While this may seem to be a limited class of relations, the work of [HLR21] demonstrated that by using powerful tools from coding theory, any form of correlation intractability for efficiently enumerable relations can be boosted to a much larger class of relations. In particular, this larger class includes a relation that was already identified in [HRW16] as relevant to separating IND-SOA-CRS from IND-CPA security.

The use of iO for building cryptographic applications has been driven by the interplay between iO itself with "iO friendly" primitives such as puncturable PRFs [SW21]. We expect that our results on correlation intractability will similarly find other future applications.

2 Technical Overview

We first explain the counterexample of [HRW16] and their analysis.

2.1 The Counterexample of Hofheinz, Rao, and Wichs

The encryption scheme extends a puncturable encryption scheme by appending to the public key pk the following auxiliary values, which are not used by an honest encryptor or decryptor.

- a hash function h, and
- an obfuscated program $\widetilde{\mathsf{SOAHelper}} \leftarrow \mathcal{O}(\mathsf{SOAHelper}[h, \mathsf{sk}])$ (here $\mathsf{SOAHelper}$ is a circuit that has h embedded, as well as the secret key sk corresponding to pk, and is described in more detail below).

For technical reasons that we elaborate on below, their encryption scheme needs to be limited to a message space of size $\lambda^{O(1)}$, where λ denotes the security parameter.

Another central ingredient in the circuit $\mathsf{SOAHelper}[h, \mathsf{sk}]$ is an error-correcting code $\mathcal{C}$ with constant rate R, constant relative distance, block length $n = \Theta(\lambda)$, and a polynomial-time algorithm for correcting $\mathcal{C}$ from a constant fraction δ of errors ([HRW16] describe their scheme in terms of polynomials, but we find the more abstract coding terminology to be more compelling for this overview).

$\mathsf{SOAHelper}[h, \mathsf{sk}]$ takes as input a ciphertext vector $\mathbf{ct} = (\mathsf{ct}_1, \dots, \mathsf{ct}_n)$ and a vector of openings $\mathbf{o} = \big((\mu_1, r_1), \dots, (\mu_{Rn/2}, r_{Rn/2})\big)$, and computes in two main steps.

1. Use sk to decrypt $\mathbf{ct}$, resulting in a message vector $\mathbf{m}$, and use the error-correction algorithm for $\mathcal{C}$ to find $\mathbf{c} \in \mathcal{C}$ that is δ-close to $\mathbf{m}$. If there is no such $\mathbf{c}$ then output $\perp$.
2. Compute $(i_1, \dots, i_{Rn/2}) := h(\mathbf{ct})$, and check that

   $$\mathsf{ct}_{i_j} = \mathsf{Enc}\big(\mathsf{pk}, \mu_j; r_j\big) \text{ for all } j \in [Rn/2], \tag{1}$$

   If so, output $\mathbf{c}$; otherwise, output $\perp$.

The Scheme Is Not IND-SOA-CRS Secure The attack on IND-SOA-CRS security for this scheme is fairly straight-forward. The adversary requests encryptions $\mathbf{ct}$ of a random codeword $\mathbf{c} \in \mathcal{C}$, computes $\mathbf{i} := h(\mathbf{ct})$ and asks the challenger for a vector $\mathbf{o}$ of openings, where each o_j is an opening of ct_{i_j}.

Evaluating $\widetilde{\mathsf{SOAHelper}}$ on input $(\mathbf{ct}, \mathbf{o})$ yields $\mathbf{c}$ with probability 1. For the adversary to be able to distinguish $\mathbf{c}$ from a "re-sampled" $\tilde{\mathbf{c}}$ that is sampled from $\mathcal{C}$ conditioned on $\tilde{c}_{i_j} = c_{i_j}$, we just need to show that with high probability $\tilde{\mathbf{c}} \neq \mathbf{c}$.

This follows from the fact that $\mathcal{C}$ has high rate compared to the fraction of indices that are opened. There are with high probability many other codewords $\mathbf{c}'$ with $c_{i_j} = c'_{i_j}$ for all $j \in [\lambda]$. This means that a "re-sampled" message $\tilde{\mathbf{c}}$ will with high probability be different from $\mathbf{c}$, enabling the adversary to easily distinguish.

The Scheme Is IND-CPA Secure The challenge is to prove that this scheme satisfies IND-CCA (or even IND-CPA) security, despite the inclusion of sk in the public key via $\mathcal{O}(\mathsf{SOAHelper}[h, \mathsf{sk}])$. In this overview, we will focus for simplicity on IND-CPA security. Additionally, we will suppose that the challenge messages are sampled uniformly at random rather than being chosen adversarially as a function of the public key. This modification is without loss of generality because the message space is polynomially sized.

The main idea in proving security is to show that $\widetilde{\mathsf{SOAHelper}}$ is indistinguishable from a program $\widetilde{\mathsf{SOAHelper}}'$ that only uses a *punctured* secret key $\mathsf{sk}\{\mathsf{ct}^\star\}$ that is useless for decrypting the challenge ciphertext $\mathsf{ct}^\star$. That such a key exists is part of the definition of puncturable encryption.

Intuitively, the main reason for this indistinguishability is the error correction in Step 1. Suppose we replace sk by a punctured key $\mathsf{sk}\{\mathsf{ct}^\star\}$ that cannot help with decrypting $\mathsf{ct}^\star$. Then $\widetilde{\mathsf{SOAHelper}}'$ instead computes a message vector $\mathbf{m}'$ differing from $\mathbf{m}$ in at most one coordinate i, with $m'_i = \perp$. On one hand, if $\mathbf{m}$ is far from $\mathcal{C}$ then so is $\mathbf{m}'$, so both programs output $\perp$. On the other hand, if $\mathbf{m}$ is very close to $\mathcal{C}$, then $\mathbf{m}$ and $\mathbf{m}'$ correct to the same codeword $\mathbf{c}$, which again implies that both programs output the same value.

However, these two cases do not cover all possibilities. There exists a third "boundary" case in which $\mathbf{m}$ is just close enough to $\mathcal{C}$ for the error correction algorithm to succeed, but $\mathbf{m}'$ (having one entry changed to $\perp$) is not. In this case, $\widetilde{\mathsf{SOAHelper}}'$ outputs $\perp$ in Step 1. Step 2 is intended to ensure that $\widetilde{\mathsf{SOAHelper}}$ also outputs $\perp$.

Specifically, suppose $\mathbf{ct}$ decrypts to such a "boundary" message vector $\mathbf{m}$, whose closest codeword is $\mathbf{c}$. Since the error correction algorithm corrects up to a δ fraction of errors, the set $I = \{i : c_i = m_i\}$ has size $(1-\delta) \cdot n$. The only way that Step 2 might *not* output $\perp$ is if when computing $\mathbf{i} := h(\mathbf{ct})$, every i_j is in I. Define R_{sk} to be the relation consisting of all such "bad" pairs $(\mathbf{ct}, \mathbf{i})$.

The authors of [HRW16] observed that R_{sk} is evasive, i.e. for any $\mathbf{ct}$, a random choice of $\mathbf{i}$ is unlikely to satisfy $(\mathbf{ct}, \mathbf{i}) \in R$. Motivated by this, they assume the existence of a hash function H that is *correlation intractable* for R_{sk}, i.e. it is computationally infeasible to find any $\mathbf{ct}$ with $(\mathbf{ct}, H(\mathbf{ct})) \in R_{\mathsf{sk}}$, even if given the random coins used to sample H. Indeed any reasonable cryptographic hash function (with a random salt) can be conjectured to satisfy this correlation intractability.

It follows immediately that if pcdiO is an obfuscator satisfying the strong notion of *public-coin differing inputs obfuscation*, then $\mathsf{pcdiO}(\mathsf{SOAHelper}[h, \mathsf{sk}])$ and $\mathsf{pcdiO}(\mathsf{SOAHelper}[h, \mathsf{sk}\{c^\star\}])$ are computationally indistinguishable.

2.2 This Work

We observe that if $\mathsf{SOAHelper}[H, \mathsf{sk}]$ and $\mathsf{SOAHelper}[H, \mathsf{sk}\{c^\star\}]$ were functionally *equivalent*, then for any *indistinguishability* obfuscator $\mathsf{i}\mathcal{O}$, it would hold that $\mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[H, \mathsf{sk}]) \approx_c \mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[H, \mathsf{sk}\{c^\star\}])$. In particular, this would hold if H were to perfectly avoid R_{sk}, i.e. for *all* ciphertext vectors $\mathbf{ct}$, satisfy $(\mathbf{ct}, H(\mathbf{ct})) \notin R_{\mathsf{sk}}$. However, this seems difficult to achieve becauses H should not depend on sk.

Instead, we construct a hash family that is *somewhere statistically correlation intractable* [CCH+19] for all R_{sk}, and sample H from this family. That is, given any sk, it is possible to sample a hash function $\tilde{H}_{\mathsf{sk}}$

that perfectly avoids R_{sk}, and yet is is computationally indistinguishable from H even to a distinguisher who knows sk.

We thus have

$$\begin{aligned}\mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[H,\mathsf{sk}]) &\approx_c \mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[\tilde{H}_{\mathsf{sk}},\mathsf{sk}])\\ &\approx_c \mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[\tilde{H}_{\mathsf{sk}},\mathsf{sk}\{c^\star\}])\\ &\approx_c \mathsf{i}\mathcal{O}(\mathsf{SOAHelper}[H,\mathsf{sk}\{c^\star\}]).\end{aligned}$$

2.2.1 SS-CI Hashing for Enumerable Relations

By now, we have reduced our goal to constructing an SS-CI hash for all R_{sk}. Our first step is to invoke a lemma of Holmgren, Lombardi, and Rothblum [HLR21] to further reduce to constructing an SS-CI hash for all efficiently *enumerable* relations — relations R with a polynomial-size circuit that on input x, outputs *all* y for which $(x, y) \in R$.

For this, we build on Peikert and Shiehian's LWE-based construction of a hash family that is SS-CI for the strictly smaller class of polynomial-size computable functions [PS19]. We prove that that obfuscating this family with IO (and sufficient padding) yields a hash family that is SS-CI for all polynomial-size enumerable relations. More specifically, if R is an enumerable relation, we prove that an IO obfuscation of the PS hash function H is indistinguishable from an IO-obfuscated circuit that, on input x, performs the following steps:

1. Enumerate the possible "bad" outputs y (those for which for which $(x, y) \in R$), and choose a $y^\star$ that is not bad.
2. If $H(x)$ is bad, output $y^\star$; otherwise, output $H(x)$.

We remark that without the "somewhere statistical" requirement, it was observed already in [CCH+19] that any hash family that is CI for efficiently computable functions is also CI for efficiently enumerable relations.

Finally, we present a more direct construction that may also be of interest. This construction is based on the existence of a circular secure FHE scheme, and is similar to a construction in [CCH+19] of an SS-CI hash family for efficiently enumerable relations.

This hash family is most easily described in terms of its indistinguishable mode for avoiding a relation R that is enumerated by a circuit E. In this mode, the hash key is an FHE encryption of (sk, E), where sk is the FHE secret key. To evaluate the hash function on input x, one uses homomorphic evaluation on the hash key to compute a ciphertext $\hat{y}^\star$ whose corresponding plaintext is *not* equal to any $\mathsf{Dec}(\mathsf{sk}, y_i)$ for $(y_1, \ldots, y_\ell) := E(x)$. This implies that $\hat{y}^\star \notin \{y_1, \ldots, y_\ell\}$ as desired. In the "honest" mode for this hash family, the hash key instead is an encryption of $(\mathsf{sk}, \mathbf{0})$, where $\mathbf{0}$ is an all-0 string of the same length as E.

We leave it as an interesting open question whether one can construct an SS-CI hash family for efficiently enumerable relations based only on the LWE assumption.

3 Preliminaries

We write $f : X \xrightarrow{\$} Y$ to denote a probabilistic function that on input $x \in X$, uses randomness to sample a value in Y.

3.1 Ensembles and Asymptotics

Definition 3.1. *If $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ are ensembles of random variables, $\mathcal{X}$ and $\mathcal{Y}$ are said to be* computationally indistinguishable *(denoted $\mathcal{X} \approx_c \mathcal{Y}$) if for all polynomial-size circuit ensembles $\{D_\lambda\}_{\lambda\in\Lambda}$, we have*

$$\left|\Pr[D_\lambda(\mathcal{X}_\lambda) = 1] - \Pr[D_\lambda(\mathcal{Y}_\lambda) = 1\right| \le \lambda^{-\omega(1)}. \tag{2}$$

The left-hand side of Eq. (2) is called the advantage of D_λ in distinguishing $\mathcal{X}_\lambda$ from $\mathcal{Y}_\lambda$.

Lemma 3.2. *Computational indistinguishability ($\approx_c$) is an equivalence relation. That is, for any random variable ensembles $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, we have:*

1. *(Reflexivity) $\mathcal{X} \approx_c \mathcal{X}$.*
2. *(Symmetry) If $\mathcal{X} \approx_c \mathcal{Y}$ then $\mathcal{Y} \approx_c \mathcal{X}$.*
3. *(Transitivity) If $\mathcal{X} \approx_c \mathcal{Y}$ and $\mathcal{Y} \approx_c \mathcal{Z}$ then $\mathcal{X} \approx_c \mathcal{Z}$.*

Lemma 3.3 (Hybrid arguments)**.** *Let $p : \mathbb{N} \to \mathbb{N}$ be polynomially bounded, and let $\{\mathcal{X}_{\lambda,i}\}_{\lambda\in\mathbb{N},i\in[p(i)]}$ be a random variable ensemble such that for every $\{i_\lambda \in [p(\lambda)-1]\}_{\lambda\in\mathbb{N}}$, it holds that $\{\mathcal{X}_{\lambda,i_\lambda}\}_{\lambda\in\mathbb{N}} \approx_c \{\mathcal{X}_{\lambda,i_\lambda+1}\}_{\lambda\in\mathbb{N}}$. Then*

$$\{\mathcal{X}_{\lambda,1}\}_\lambda \approx_c \{\mathcal{X}_{\lambda,p(\lambda)}\}_\lambda.$$

Proof. Suppose otherwise for contradiction. That is, suppose that there is a polynomial-size circuit ensemble $\{D_\lambda\}_{\lambda\in\mathbb{N}}$, a function $\epsilon(\lambda) \geq \lambda^{-O(1)}$, and an infinite set $\Lambda \subseteq \mathbb{N}$ such that for all $\lambda \in \Lambda$, D_λ distinguishes $\mathcal{X}_{\lambda,1}$ from $\mathcal{X}_{\lambda,p(\lambda)}$ with advantage at least $\epsilon(\lambda)$.

Then by the triangle inequality, there must exist $\{i_\lambda \in [p(\lambda)-1]\}_{\lambda\in\Lambda}$ such that for all $\lambda \in \Lambda$, D_λ distinguishes $\mathcal{X}_{\lambda,i_\lambda}$ from $\mathcal{X}_{\lambda,i_\lambda+1}$ with advantage at least $\epsilon(\lambda)/p(\lambda) \geq \lambda^{-O(1)}$, which is a contradiction. □

3.2 Relations

Definition 3.4. *A* relation *R is a subset $R \subseteq X \times Y$, where the sets X and Y are respectively called the* domain *and* codomain *of R.*

Relations are a generalization of functions. A function $f : X \to Y$ is just a relation with domain X and codomain Y with the property that for each x, there is exactly one y (denoted $f(x)$) for which $(x, y) \in f$. Generalizing function evaluation notation, we write $R(x)$ to denote the set $\{y \in Y : (x, y) \in R\}$.

Definition 3.5 (Relational Inverses and Compositions)**.** *Let Q and R be relations with*

$$Q \subseteq X \times Y \qquad\qquad R \subseteq Y \times Z.$$

- *The* inverse *of R, denoted R^{-1}, is defined as*

$$R^{-1} \subseteq Z \times Y$$
$$R^{-1} \stackrel{\mathsf{def}}{=} \{(z, y) : (y, z) \in R\}.$$

- *The* composition *of R with Q, denoted $R \circ Q$, is defined as*

$$R \circ Q \subseteq X \times Z$$
$$R \circ Q \stackrel{\mathsf{def}}{=} \{(x, z) : \exists y \text{ s.t. } (x, y) \in Q \text{ and } (y, z) \in R\}.$$

3.3 Public-Key Encryption

Definition 3.6 (PKE Syntax)**.** *A* public-key encryption scheme *with* message space *$\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ syntactically consists of a tuple of algorithms $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that:*

- ***Perfect Correctness.*** *For all $\lambda \in \mathbb{N}$ and all $m \in \mathcal{M}_\lambda$, when sampling*

$$\begin{aligned}&\mathsf{sk} \leftarrow \{0,1\}^\lambda, \mathsf{pk} := \mathsf{Gen}(\mathsf{sk})\\ &\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)\\ &m' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}),\end{aligned}$$

  it holds with probability 1 that $m = m'$.

A PKE scheme is also generally required to satisfy one of several possible security properties. We define these properties separately.

3.3.1 Puncturable Encryption

As in [HRW16], we rely on the notion and existence of "puncturable encryption", which is defined and constructed from iO and one-way functions in [CHV15]. Loosely speaking, while CCA security preserves the security of a ciphertext c even when the adversary is given oracle access to an "all-but-c" decryption oracle, in a puncturable encryption scheme the adversary is instead given an actual key that allows the adversary to simulate this decryption oracle on his own.

Definition 3.7 (Puncturable Encryption)**.** *A* puncturable encryption scheme with message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ *is a public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *along with supplemental algorithms* $\widetilde{\mathsf{Enc}}$ *and* $\mathsf{Puncture}$ *satisfying the following properties:*

- ***Puncturability.*** *Let $\ell(\lambda)$ denote the length of ciphertexts $\mathsf{Enc}(\mathsf{Gen}(\mathsf{sk}), m)$, for $\mathsf{sk} \in \{0,1\}^\lambda$ and $m \in \mathcal{M}_\lambda$.*

  Then, for all $\mathsf{sk} \in \{0,1\}^\lambda$ and $c_0, c_1 \in \{0,1\}^{\ell(\lambda)}$, the output of $\mathsf{Puncture}(\mathsf{sk}, \{c_0, c_1\})$ is a circuit $\mathsf{sk}_{\setminus\{c_0,c_1\}}$ such that for all $c \in \{0,1\}^{\ell(\lambda)}$,

$$\mathsf{sk}_{\setminus\{c_0,c_1\}}(c) = \begin{cases} \mathsf{Dec}(\mathsf{sk}, c) & \textit{if } c \notin \{c_0, c_1\} \\ \perp & \textit{otherwise.} \end{cases}$$

- ***Security.*** *For every polynomial-size circuit $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\left| \Pr\left[\mathsf{Exp}_{\mathsf{punc\text{-}ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A}_\lambda) = 1\right] - \frac{1}{2} \right| \leq \lambda^{-\omega(1)},$$

  where $\mathsf{Exp}_{\mathsf{punc\text{-}ind\text{-}cca}}$ is defined in Fig. 1.

- ***Ciphertext Sparseness.*** *When sampling $\mathsf{sk} \leftarrow \{0,1\}^\lambda$, $\mathsf{pk} := \mathsf{Gen}(1^\lambda)$, $\mathsf{ct} \leftarrow \widetilde{\mathsf{Enc}}(\mathsf{pk})$, and $m \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, it holds with overwhelming probability that $m = \perp$.*

$\underline{\mathsf{Exp}_{\mathsf{punc\text{-}ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A})}$

1. $\mathsf{sk} \leftarrow \{0,1\}^\lambda$, $\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$
2. $m \leftarrow \mathcal{A}(1^\lambda)$
3. $c_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$, $c_1 \leftarrow \widetilde{\mathsf{Enc}}(\mathsf{pk})$,
4. $\mathsf{sk}\{c_0, c_1\} \leftarrow \mathsf{Puncture}(\mathsf{sk}, \{c_0, c_1\})$
5. $b \leftarrow \{0,1\}$
6. return 1 if $\mathcal{A}(\mathsf{pk}, c_b, c_{1-b}, \mathsf{sk}\{c_0, c_1\}) = b$, and 0 otherwise.

Figure 1: The security experiment to determine whether an adversary $\mathcal{A}$ violates the security of a puncturable encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \widetilde{\mathsf{Enc}}, \mathsf{Dec}, \mathsf{Puncture})$.

Imported Theorem 3.8 ([CHV15])**.** *If indistinguishability obfuscation and one-way functions exist, then puncturable encryption exists.*

3.3.2 Fully Homomorphic Encryption

Definition 3.9 (Fully Homomorphic Encryption)**.** *A (secret-key)* fully homomorphic encryption (FHE) scheme *for a class* $\{\mathbb{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *of circuits is a triple of polynomial-time algorithms* $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*, where* Enc *is probabilistic, such that:*

- *(Perfect Evaluation Correctness) For all* $\lambda, n \in \mathbb{N}$*, all circuits* $C \in \mathbb{C}_\lambda$ *with* n *inputs, and all* $x \in \{0,1\}^n$*, when computing*

$$\begin{aligned} &\mathsf{sk} \leftarrow \{0,1\}^\lambda \\ &\mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{sk}, x) \\ &\mathsf{ct}_y := \mathsf{Eval}(C, \mathsf{ct}_x) \\ &y := \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_y) \end{aligned} \tag{3}$$

  it holds with probability 1 *that* $y = C(x)$*.*

- *(Compactness) There exists a polynomially bounded function* $B : \mathbb{N} \to \mathbb{N}$ *such that for all* $\lambda, n, m \in \mathbb{N}$*, all circuits* $C \in \mathbb{C}_\lambda$ *with* n *inputs and* m *outputs, and all strings* $x \in \{0,1\}^n$*, the ciphertext* ct_y *sampled in Eq. (3) has length* $m \cdot B(\lambda)$*.*

The notion of FHE is due to Rivest, Adleman, and Dertouzos [RAD+78], and the first candidate construction (for all circuits of any fixed polynomial size) is due to Gentry [Gen09] based on ideal lattices. Later, Brakerski and Vaikuntanathan [BV11] constructed FHE based only on the hardness of learning with errors (LWE), which is a more standard cryptographic assumption with a host of desirable properties.

One of our hash family constructions will rely on FHE with an additional circular security property.

Definition 3.10 (Circular Security)**.** *We say that an FHE scheme as in Definition 3.9 is* circular secure *if for any polynomial-length message ensembles* $\{m_\lambda^{(0)} \in \{0,1\}^{\ell_\lambda}\}_{\lambda\in\mathbb{N}}$ *and* $\{m_\lambda^{(1)} \in \{0,1\}^{\ell_\lambda}\}_{\lambda\in\mathbb{N}}$*, we have*

$$\left\{\mathsf{Enc}(\mathsf{sk}, \mathsf{sk}\|m_\lambda^{(0)}) \middle| \mathsf{sk} \leftarrow \{0,1\}^\lambda\right\} \approx_c \left\{\mathsf{Enc}(\mathsf{sk}, \mathsf{sk}\|m_\lambda^{(1)}) \middle| \mathsf{sk} \leftarrow \{0,1\}^\lambda\right\}.$$

It is not known how to construct circular-secure FHE based on indistinguishability obfuscation and LWE, but natural constructions such as that of [BV11] are conjectured to be circular secure.

3.4 Circuit Obfuscation

Definition 3.11 (Circuit Equivalence)**.** *Let* C_0 *and* C_1 *be circuits with* n *input bits. We say that* C_0 *and* C_1 *are* functionally equivalent *(denoted* $C_0 \equiv C_1$*) if for all* $x \in \{0,1\}^n$*,* $C_0(x) = C_1(x)$*.*

Definition 3.12 (Indistinguishability Obfuscation)**.** *An indistinguishability obfuscator is a p.p.t. algorithm* $\mathsf{i}\mathcal{O} : \{0,1\}^* \xrightarrow{\$} \{0,1\}^*$ *such that:*

Correctness *If* C *is any boolean circuit, then every* $\tilde{C}$ *in the support of* $\mathsf{i}\mathcal{O}(C)$ *is a circuit that is functionally equivalent to* C*.*

Security *If* $\{C_\lambda^0\}_{\lambda\in\mathbb{N}}$ *and* $\{C_\lambda^1\}_{\lambda\in\mathbb{N}}$ *are ensembles of circuits with* $|C_\lambda^0| = |C_\lambda^1| = \lambda^{\Theta(1)}$ *and* $C_\lambda^0 \equiv C_\lambda^1$*, then*

$$\{\mathsf{i}\mathcal{O}(C_\lambda^0)\}_\lambda \approx_c \{\mathsf{i}\mathcal{O}(C_\lambda^1)\}_\lambda.$$

Applications of indistinguishability obfuscation generally rely on the simple fact that given any circuit, one can efficiently find a functionally equivalent circuit of any polynomially *larger* size.

Fact 3.13 (Padding)**.** *There is a polynomial-time algorithm that takes as input a circuit* C *and an integer* $p \geq |C|$*, and outputs a circuit* C' *satisfying* $C \equiv C'$ *and* $|C'| = p$*. We denote this circuit by* $\mathsf{Pad}_p(C)$*.*

4 Notions of Security for Public-Key Encryption

4.1 Chosen-Ciphertext Attacks

In chosen-ciphertext attacks, the adversary is given the ability to make decryption queries to any string other than the challenge ciphertext c. To define this formally, we write $\mathsf{Dec}(\mathsf{sk}, \cdot)_{-c}$ to denote the function that agrees with $\mathsf{Dec}(\mathsf{sk}, \cdot)$ on all inputs except for c; on input c the $\mathsf{Dec}(\mathsf{sk}, \cdot)_{-c}$ returns $\perp$.

4.1.1 IND-CCA and $-IND-CCA Security

Definition 4.1. *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be* IND-CCA secure *if for all polynomial-size adversaries* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, the probability that* $\mathsf{Exp}_{\mathsf{ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A}_\lambda)$ *outputs is* $\frac{1}{2} + \lambda^{-\omega(1)}$.

Definition 4.2. *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message spaces* $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ *is said to be* $-IND-CCA secure *if for all polynomial-size adversaries* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, the probability that* $\mathsf{Exp}_{\$\text{-}\mathsf{ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A}_\lambda)$ *outputs is* $\frac{1}{2} + \lambda^{-\omega(1)}$.

Experiment $\mathsf{Exp}_{\mathsf{ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A})$	**Experiment** $\mathsf{Exp}_{\$\text{-}\mathsf{ind\text{-}cca}}(1^\lambda, \mathsf{PKE}, \mathcal{A})$
1. $\mathsf{sk} \leftarrow \{0,1\}^\lambda; \mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$	1. $\mathsf{sk} \leftarrow \{0,1\}^\lambda; \mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk})$	2. $m_0, m_1 \leftarrow \mathcal{M}_\lambda$
3. $b \leftarrow \{0,1\}$	3. $b \leftarrow \{0,1\}$
4. $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$	4. $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
5. $b' := \mathcal{A}^{\mathsf{Dec}(\mathsf{sk},\cdot)_{-c}}(\mathsf{pk}, (m_0, m_1), c)$	5. $b' := \mathcal{A}^{\mathsf{Dec}(\mathsf{sk},\cdot)_{-c}}(\mathsf{pk}, (m_0, m_1), c)$
6. Return 1 if $b = b'$; otherwise return 0.	6. Return 1 if $b = b'$; otherwise return 0.

Figure 2: The experiments for determining whether an adversary $\mathcal{A}$ violates the IND-CCA or $-IND-CCA security of a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Theorem 4.3 ([HRW16, Theorem A.2])**.** *Let* PKE *be a public-key encryption scheme with a polynomial-size message space* $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$*. If* PKE *is $-IND-CCA secure, then* PKE *is IND-CCA secure.*

4.2 Selective Opening Attacks and IND-SOA-CRS Security

IND-SOA-CRS security requires that the adversary should be unable to distinguish $(m_i)_{i\notin I}$ from $(m'_i)_{i\notin I}$, where $(m'_1, \ldots, m'_n)$ are sampled from the same distribution as $(m_1, \ldots, m_n)$, conditioned on $m_i = m'_i$ for all $i \in I$, as long as these distributions are always efficiently sampleable. To help with formalizing the definition, we will introduce the following notation. If A is an algorithm, then we write $\mathsf{Coins}_A(x)$ to denote the distribution on $\{0,1\}^*$ of random coins used by A on input x.

Definition 4.4. *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ *is said to satisfy* IND-SOA-CRS security *if it holds that*

$$\Pr\left[\mathsf{Exp}_{\mathsf{ind\text{-}soa\text{-}crs}}(1^\lambda, \mathsf{PKE}, \mathcal{A}_\lambda) = 1\right] \le \frac{1}{2} + \lambda^{-\omega(1)},$$

where $\mathsf{Exp}_{\mathsf{ind\text{-}soa\text{-}crs}}$ *is described in Fig. 3.*

Experiment $\mathsf{Exp}_{\mathsf{ind\text{-}soa\text{-}crs}}(1^\lambda, \mathsf{PKE}, \mathcal{A})$

Key Generation The challenger first samples $\mathsf{sk} \leftarrow \{0,1\}^\lambda$ and computes $\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$

Choosing a Message Distribution The adversary $\mathcal{A}$, given the public key pk, outputs $n \in \mathbb{N}$ and a distribution $\mathcal{D}$ on n-tuples of messages, represented by a (polynomial-size) sampling circuit *and* a "conditional resampling" circuit $\mathcal{R}$ for $\mathcal{D}$. Given $I \subseteq [n]$ and $(m_i)_{i \in I}$, $\mathcal{R}$ samples $(m'_1, \ldots, m'_n)$ from $\mathcal{D}$ conditioned on $m'_i = m_i$ for all $i \in I$.
If the adversary outputs $\mathcal{R}$ that is *not* a conditional resampling circuit for $\mathcal{D}$, then the experiment outputs 0, representing a loss for the adversary.

Sampling "True" Messages and Corresponding Ciphertexts Let $\rho(\lambda)$ denote the number of random bits used by Enc input (pk, m) when pk is in the support of $\mathsf{Gen}(\mathsf{sk})$ for $\mathsf{sk} \in \{0,1\}^\lambda$.
The challenger samples a tuple of messages $(m_1^{(0)}, \ldots, m_n^{(0)}) \leftarrow \mathcal{D}$, samples independent encryption randomnesses $r_1, \ldots, r_n \leftarrow \{0,1\}^{\rho(\lambda)}$, and computes encryptions $c_i = \mathsf{Enc}(\mathsf{pk}, m_i; r_i)$ for all $i \in [n]$.

Selective Opening $\mathcal{A}$ is given all of these ciphertexts $(c_1, \ldots, c_n)$, outputs a subset $I \subseteq [n]$, and for each $i \in I$ is given the "opening" (m_i, r_i).

Sampling Consistent Alternative Messages The challenger samples $(m_1^{(1)}, \ldots, m_n^{(1)}) \leftarrow \mathcal{D}$ conditioned on the constraint that $m_i^{(1)} = m_i^{(0)}$ for all $i \in I$.

The Distinguishing Test The challenger samples $b \leftarrow \{0,1\}$ uniformly at random, $\mathcal{A}$ is given $m_1^{(b)}, \ldots, m_n^{(b)}$, and $\mathcal{A}$ is finally asked to guess b. If $\mathcal{A}$ guesses correctly, the experiment returns 1, representing a win for the adversary; otherwise, the experiment returns 0.

Figure 3: The experiment determining whether an adversary $\mathcal{A}$ violates IND-SOA-CRS security for a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Although the notion of IND-SOA-CRS security is natural, it will be simpler for us to work with a simpler, more specialized security notion that is implied by IND-SOA-CRS security.

Shamir Secret Sharing (SSS-) SOA security, which was introduced[1] in [HRW16], is a special case of IND-SOA-CRS security focusing on a particular rather than adversarial choice of message distribution. Specifically, SSS-SOA security focuses on public-key encryption schemes whose message space is a sufficiently large field $\mathbb{F}_\lambda$, and uses message distributions given by Shamir secret sharing. That is, the i^{th} message is $F(i)$, where F is a random degree-λ univariate polynomial over $\mathbb{F}_\lambda$. The security requirement is that the adversary must be unable to guess $F(0)$ with probability noticeably larger than $1/|\mathbb{F}_\lambda|$.

Definition 4.5. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme whose message space (for security parameter λ) is a finite field $\mathbb{F}_\lambda$.* PKE *is said to satisfy* SSS-SOA security *if for all polynomial-size adversaries* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$*, the experiment* $\mathsf{Exp}_{\mathsf{sss\text{-}soa}}(1^\lambda, \mathsf{PKE}, \mathcal{A}_\lambda)$*, depicted in Fig. 4, outputs* 1 *with probability* $\frac{1}{|\mathbb{F}_\lambda|} + \mathrm{negl}(\lambda)$.

[1] Actually, our notion is even more specialized than the notion in [HRW16], which they called SecShare-SOA security; their notion allowed for more general choices of degree, field size, and number of polynomial evaluation points, but is otherwise the same.

Experiment $\mathsf{Exp}_{\mathsf{sss\text{-}soa}}(1^\lambda, \mathsf{PKE}, \mathcal{A})$

1. $\mathsf{sk} \leftarrow \{0,1\}^\lambda$, $\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$.
2. Let F be a uniformly random degree-λ univariate polynomial over $\mathbb{F}_\lambda$.
3. For $i \in \{1, \ldots, 3\lambda\}$, sample $r_i \leftarrow \mathsf{Coins}_{\mathsf{Enc}}(\mathsf{pk}, F(i))$, and define $c_i := \mathsf{Enc}\big(\mathsf{pk}, F(i); r_i\big)$.
4. $(i_1, \ldots, i_\lambda) \leftarrow \mathcal{A}(\mathsf{pk}, c_1, \ldots, c_{3\lambda})$. The i_j need not be distinct, but must be in the range $\{1, \ldots, 3\lambda\}$; if any are outside this range, return 0.
5. $m^\star \leftarrow \mathcal{A}\Big(\big(F(i_1), r_{i_1}\big), \ldots, \big(F(i_\lambda), r_{i_\lambda}\big)\Big)$.
6. If $m^\star = F(0)$, then return 1; otherwise return 0.

Figure 4: The experiment determining whether an adversary $\mathcal{A}$ violates the SSS-SOA security of a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ whose message space is a finite field $\mathbb{F}_\lambda$ satisfying $|\mathbb{F}_\lambda| \geq 3\lambda$.

Theorem 4.6 ([HRW16, Theorem 3.2])**.** *If a public-key encryption scheme satisfies IND-SOA-CRS security, then it satisfies SSS-SOA security.*

5 Somewhere Statistical Correlation Intractability

In this section we construct hash families that are *somewhere statistically* correlation intractable (SS-CI) for efficiently enumerable relations. Previously SS-CI hash families were known only for efficiently computable functions.

Definition 5.1 (Enumerable Relations)**.** *We say that a relation $R \subseteq X \times Y$ is (S, ℓ)-*enumerable *if there is a size-S circuit E that on input $x \in X$, outputs $(y_1, \ldots, y_{\ell'}) \in Y^{\leq \ell}$ such that for all $y \in Y$, if $(x, y) \in R$ then $y \in \{y_1, \ldots, y_{\ell'}\}$. In increasing levels of specificity, we can say that E is an* enumerator*, an ℓ-*enumerator*, or an (S, ℓ)-*enumerator *for R.*

We say that R is size-S enumerable *if it is (S, ∞)-enumerable (or equivalently, (S, S)-enumerable), and we say that R is ℓ-*enumerable *if it is (∞, ℓ)-enumerable.*

Definition 5.2. *We say that a function $h : X \to Y$* perfectly avoids *a binary relation R if for* all *$x \in X$, $h(x) \notin R(x)$. We say that a hash family ensemble $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$* statistically avoids *a relation ensemble $R = \{R_\lambda\}_{\lambda \in \mathbb{N}}$ if when sampling $H \leftarrow \mathcal{H}_\lambda$, it holds with all but $\lambda^{-\omega(1)}$ probability that H perfectly avoids R_λ.*

Definition 5.3 (Somewhere Statistical Correlation Intractability [CCH+19])**.** *Let $R = \{R_\lambda\}_{\lambda \in \mathbb{N}}$ be a binary relation ensemble. An hash family ensemble $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ is said to be* somewhere statistically correlation intractable (SS-CI) *for R if there exists a computationally indistinguishable hash family ensemble $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ that statistically avoids R. We say that such a family $\mathcal{F}$ is an R-*avoiding mode *of $\mathcal{H}$.*

5.1 Boosting SS-CI: From Functions to Enumerable Relations

Our first construction relies on indistinguishability obfuscation (IO), and as such is inherently a private-coin construction. The computational assumptions besides IO are minimal: we start with any SS-CI hash family for polynomial-size *computable functions*, and we boost this to an SS-CI hash family for polynomial-size *enumerable relations*, which is a strictly larger class.

Construction 5.4. *For any integers n, m, any hash family $\mathcal{H} : \{0,1\}^n \to \{0,1\}^m$, any obfuscator $\mathcal{O}$, and any integer p upper bounding the circuit size of hash functions in $\mathcal{H}$, we define a "boosted" hash family*

$\mathsf{Boost}^{\mathcal{O}}(\mathcal{H}, p)$, *that is sampled as follows:*

$$\begin{aligned}&H \leftarrow \mathcal{H}\\&\tilde{H} \leftarrow \mathcal{O}\big(\mathsf{Pad}_p(H)\big)\\&\textbf{\textit{return}}\ \tilde{H}\end{aligned}$$

Theorem 5.5. *Let $n, m, S : \mathbb{N} \to \mathbb{N}$ be polynomially bounded functions, let $\mathsf{i}\mathcal{O}$ be an indistinguishability obfuscator, and let $\mathcal{H} = \{\mathcal{H}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be a p.p.t.-sampleable hash family ensemble that is SS-CI for size-S computable functions.*

There exists a polynomially-bounded function $p : \mathbb{N} \to \mathbb{N}$ such that the (p.p.t.-sampleable) hash family ensemble $\mathcal{G} = \left\{\mathsf{Boost}^{\mathsf{i}\mathcal{O}}\big(\mathcal{H}_\lambda, p(\lambda)\big)\right\}_{\lambda\in\mathbb{N}}$ is SS-CI for every (S, S)-enumerable relation ensemble $R = \{R_\lambda\}_{\lambda\in\mathbb{N}}$. Moreover, there exists a p.p.t. algorithm $\widetilde{\mathsf{Samp}}$ such that if E_λ is a size-$S(\lambda)$ circuit that enumerates R_λ, then $\left\{\widetilde{\mathsf{Samp}}(E_\lambda)\right\}_{\lambda\in\mathbb{N}}$ is an R-avoiding mode of $\mathcal{H}$.

Proof. Define $p(\lambda)$ as the maximum possible size of the circuit $\mathsf{Avoid}_i^{h,E}$, described in Fig. 5, for h in the support of $\mathcal{H}_\lambda$, $i \in [S(\lambda)]$, and E a size-$S(\lambda)$ circuit. The size of $\mathsf{Avoid}_i^{h,E}$ is $|h| + |E| + \lambda^{O(1)} = |h| + \lambda^{O(1)}$. Each h in the support of $\mathcal{H}_\lambda$ has size $\lambda^{O(1)}$ by the p.p.t.-sampleability of $\mathcal{H}$, so $p(\lambda)$ is also $\lambda^{O(1)}$. Let $\mathsf{i}\mathcal{O}_\lambda$ denote the probabilistic function $\mathsf{i}\mathcal{O}_\lambda(\cdot) = \mathsf{i}\mathcal{O}\big(\mathsf{Pad}_{p(\lambda)}(\cdot)\big)$.

Let $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be an arbitrary (S, S)-enumerable relation ensemble, with E_λ denoting a corresponding enumerator circuit of size $S(\lambda)$ for R_λ. We define a related hash family ensemble $\mathcal{G}^R = \{\mathcal{G}^R_\lambda\}_{\lambda\in\mathbb{N}}$, via the following sampling procedure for $\mathcal{G}^R_\lambda$, and claim that $\mathcal{G}^R$ is an R-avoiding mode for $\mathcal{G}$:

$$\begin{aligned}&H \leftarrow \mathcal{H}_\lambda\\&\tilde{H} \leftarrow \mathsf{i}\mathcal{O}_\lambda\Big(\mathsf{Avoid}^{H,E_\lambda}_{S(\lambda)}\Big) \quad \text{(see Fig. 5)}\\&\textbf{return}\ \tilde{H}.\end{aligned}$$

It is clear from the definition that $\mathcal{G}^R$ does in fact statistically avoid R, and moreover that $\mathcal{G}^R$ is p.p.t.-sampleable given E_λ as input.

It remains to establish that $\mathcal{G}^R$ is computationally indistinguishable from $\mathcal{G}$. We prove this via a hybrid argument. For $\lambda \in \mathbb{N}$ and $i \in \{0, 1, \ldots, S(\lambda)\}$, define $\mathcal{G}^R_{\lambda,i}$ as the hash family with the following sampling procedure:

$$\begin{aligned}&H \leftarrow \mathcal{H}_\lambda\\&\tilde{H} \leftarrow \mathsf{i}\mathcal{O}_\lambda\Big(\mathsf{Avoid}^{H,E_\lambda}_{i}\Big) \quad \text{(see Fig. 5)}\\&\textbf{return}\ \tilde{H}.\end{aligned}$$

Theorem 5.5 follows from applying Lemma 3.3 to Lemmas 5.6 and 5.7 below.

Lemma 5.6. $\mathcal{G} \approx_c \{\mathcal{G}^R_{\lambda,0}\}_{\lambda\in\mathbb{N}}$.

Lemma 5.7. *For any $\big\{i_\lambda \in [S(\lambda)]\big\}_{\lambda\in\mathbb{N}}$, it holds that $\big\{\mathcal{G}^R_{\lambda,i_\lambda-1}\big\}_{\lambda\in\mathbb{N}} \approx_c \big\{\mathcal{G}^R_{\lambda,i_\lambda}\big\}_{\lambda\in\mathbb{N}}$.*

Proof of Lemma 5.6. Observe that for any h and E, $\mathsf{Avoid}_0^{h,E}$ is functionally equivalent to h. For h in the support of $\mathcal{H}_\lambda$, and E being the size-$S(\lambda)$ circuit E_λ, it also holds by the definition of $p(\lambda)$ that $|h| \le \left|\mathsf{Avoid}_0^{h,E_\lambda}\right| \le p(\lambda)$. Thus the claim follows directly from the security of $\mathsf{i}\mathcal{O}$. ☐

Proof of Lemma 5.7. Fix $\{i_\lambda \in [S(\lambda)]\}_{\lambda\in\mathbb{N}}$, and for $\lambda \in \mathbb{N}$, let $U_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ be the function ("unique relation") that on input x, outputs the i_λ^{th} output of $E_\lambda(x)$. The ensemble $U = \{U_\lambda\}_{\lambda\in\mathbb{N}}$ is size-S computable, inheriting that property from $\{E_\lambda\}$. This implies the existence of a U-avoiding mode

$\mathcal{H}^U = \{\mathcal{H}^U_\lambda\}_{\lambda\in\mathbb{N}}$ of $\mathcal{H}$. Then

$$\begin{aligned}
\{\mathcal{G}_{\lambda,i_\lambda-1}\} &= \left\{ i\mathcal{O}_\lambda\left(\mathsf{Avoid}^{H,E}_{i_\lambda-1}\right) \middle| H \leftarrow \mathcal{H}_\lambda \right\} \\
&\approx_c \left\{ i\mathcal{O}_\lambda\left(\mathsf{Avoid}^{H,E}_{i_\lambda-1}\right) \middle| H \leftarrow \mathcal{H}^U_\lambda \right\} &(4)\\
&\approx_c \left\{ i\mathcal{O}_\lambda\left(\mathsf{Avoid}^{H,E}_{i_\lambda}\right) \middle| H \leftarrow \mathcal{H}^U_\lambda \right\} &(5)\\
&\approx_c \left\{ i\mathcal{O}_\lambda\left(\mathsf{Avoid}^{H,E}_{i_\lambda}\right) \middle| H \leftarrow \mathcal{H}_\lambda \right\} &(6)\\
&= \{\mathcal{G}_{\lambda,i_\lambda}\},
\end{aligned}$$

where Eqs. (4) and (6) are by the computational indistinguishability of $\mathcal{H}^U_\lambda$ from $\mathcal{H}$, and Eq. (5) is by IO security ($\mathsf{Avoid}^{H,E}_{i_\lambda-1}$ and $\mathsf{Avoid}^{H,E}_{i_\lambda}$ are functionally equivalent with high probability when sampling $H \leftarrow \mathcal{H}^U_\lambda$ because $\mathcal{H}^U_\lambda$ statistically avoids U, i.e. there is usually no $x \in \{0,1\}^{n(\lambda)}$ for which $H(x) = y_{i_\lambda}$, where $(y_1, \ldots, y_\ell) := E_\lambda(x)$). □

Hard-wired subroutines:

- Circuits E and h, both taking n-bit inputs.

On input $x \in \{0,1\}^n$:

1. Compute $(y_1, \ldots, y_\ell) \leftarrow E(x)$.
2. Let z be a canonical (e.g. lexicographically smallest) element of $\{0,1\}^{m(\lambda)} \setminus \{y_1, \ldots, y_\ell\}$.
3. If $h(x) \in \{y_1, \ldots, y_{\min(i,\ell)}\}$, then output z. Else, output $h(x)$.

Figure 5: The circuit $\mathsf{Avoid}^{h,E}_i$

□

Corollary 5.8. *Assuming the existence of an indistinguishability obfuscator and the hardness of LWE.*

Then for every $c > 0$, there exists a p.p.t.-sampleable hash family ensemble $\mathcal{H}$ that is SS-CI for every λ^c-size enumerable relation ensemble $R = \{R_\lambda\}_{\lambda\in\mathbb{N}}$.

Moreover, there exists a p.p.t. algorithm $\widetilde{\mathsf{Samp}}$ such that if E_λ is a size-λ^c-enumerator for R_λ, then $\left\{\widetilde{\mathsf{Samp}}(E_\lambda)\right\}_{\lambda\in\mathbb{N}}$ is an R-avoiding mode of $\mathcal{H}$.

Proof. Assuming the hardness of LWE, Peikert and Shiehian [PS19] proved that for all $n, m : \mathbb{N} \to \mathbb{N}$ with $m(\lambda) = \lambda^{\Theta(1)}$, and all constants $c > 0$, there exists a hash family $\mathcal{H} = \{\mathcal{H}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ that is somewhere statistically correlation intractable for all functions $f : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ that are computable by circuits of size-λ^c.

The corollary follows by applying Theorem 5.5 to this hash family. □

5.2 SS-CI for Enumerable Relations, Directly

Next, we prove that a mild generalization of the construction of Canetti et al. [CCH+19] yields, for every constant $c > 0$, a hash family that achieves *somewhere statistical* correlation intractability for $\mathsf{SIZE}(\lambda^c)$-*enumerable* relations. Besides relying on a computational assumptions (circular-secure FHE) that is formally incomparable to IO and LWE, this construction has the advantage that it can yield a *public-coin* hash family (e.g. if the FHE scheme has pseudorandom ciphertexts). As this advantage is not relevant to the present work, we do not elaborate on it further.

There are several parameters in the construction; we overview them now. First, the parameters describing what we want out of hash family (which all depend on a security parameter λ):

- Output length $\hat{m} = \hat{m}(\lambda)$. This needs to satisfy $\hat{m}(\lambda) \geq \lambda^{\Omega(1)}$ for correlation intractability to be plausible[2], and also needs to satisfy $\hat{m}(\lambda) \leq \lambda^{O(1)}$ for the hash family to be efficient. Altogether $\hat{m}(\lambda) = \lambda^{\Theta(1)}$.
- Input length $n = n(\lambda)$. This just needs to satisfy $n(\lambda) \leq \lambda^{O(1)}$ so that the hash family can be efficient.
- Circuit size S such that the hash family achieves SS-CI for all size-S enumerable relations $R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{\hat{m}(\lambda)}$.

A central idea in the construction is to interpret hash function's outputs (evaluated) FHE ciphertexts. This introduces a couple more parameters:

- We will need to interpret the hash function outputs as encryptions of m-bit messages, where 2^m is greater than the number of outputs given by the enumerator for R. In particular it suffices to set $m > \log_2(S)$.
- The FHE scheme will be used not with security parameter λ, but with a security parameter λ' that is more closely related to $\hat{m}(\lambda)$. We will want λ' to be as large as possible, while still permitting the interpretation of the hash output as an encryption of an m-bit message.

Theorem 5.9. *Let $\mathcal{FHE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a circular secure secret-key FHE scheme as in Definitions 3.9 and 3.10. Let $n, \hat{m}, S : \mathbb{N} \to \mathbb{N}$ be polynomially bounded functionswith $\hat{m}(\lambda) = \lambda^{\Theta(1)}$.*

Then the hash family $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ depicted in Fig. 6 is polynomial-time sampleable and SS-CI for the class of size-S enumerable relations.

The directly constructed hash family $\mathcal{H}_\lambda = \{H_{\mathsf{ct}}\}$

Key Sampling: A hash key consists of an $\mathcal{FHE}$ ciphertext ct sampled as

1. $\mathsf{sk} \leftarrow \{0,1\}^{\lambda'}$, where λ' is an FHE security parameter defined as follows. Let $B(\cdot)$ denote the ratio of ciphertext length to message length as a function of security parameter as in Definition 3.9, define $m := \lceil \log(S(\lambda)) \rceil + 1$. and define λ' as the largest integer for which $m \cdot B(\lambda') \leq \hat{m}(\lambda)$.
2. $\mathsf{ct} \leftarrow \mathsf{Enc}\big(\mathsf{sk}, (\mathsf{sk}, 0^{\tilde{S}})\big)$, where $\tilde{S} = \tilde{O}(S)$ is the maximum number of bits required to represent a circuit of size $S = S\big(n(\lambda)\big)$.

Evaluation of H_{ct} on input $x \in \{0,1\}^{n(\lambda)}$

1. Let C_x denote a circuit that on input $(\mathsf{sk}, E) \in \{0,1\}^{\lambda'} \times \{0,1\}^{\tilde{S}}$, computes as follows:
   (a) $(\hat{y}_1, \ldots, \hat{y}_\ell) := E(x)$, where E is interpreted as a size-S circuit, and each $\hat{y}_i$ is truncated or padded to a length of $m \cdot B(\lambda')$ bits.
   (b) $y_i := \mathsf{Dec}(\mathsf{sk}, \hat{y}_i)$ for $i \in \{1, \ldots, \ell\}$.
   (c) Output some canonical (e.g. the lexicographically first) $z \in \{0,1\}^m \setminus \{y_1, \ldots, y_\ell\}$. (Such a z exists because $\ell \leq S < 2^m$).
2. Compute and output $\hat{y} := \mathsf{Eval}(C_x, \mathsf{ct})$ padded to length $\hat{m}(\lambda)$.

Figure 6: The hash family construction for Theorem 5.9.

[2]Indeed, it may be helpful to think of $\hat{m}(\lambda)$ as the "true" security parameter for the hash family. We do not take this approach because we find our theorem statement to be more easily applied.

Proof of Theorem 5.9. We first prove that the FHE security parameter λ', computed in the key sampling procedure, satisfies $\lambda' \geq \lambda^{\Omega(1)}$, where λ is the hash family security parameter. First note that by assumption $\hat{m}(\lambda) = \lambda^{\Omega(1)}$ and $m \leq O\Big(\log\big(S(\lambda)\big)\Big) \leq O(\log \lambda) \leq \lambda^{o(1)}$, which implies that $\hat{m}(\lambda)/m \geq \lambda^{\Omega(1)}$. Since λ' is the *largest* integer for which $B(\lambda') \leq \hat{m}(\lambda)/m$, and $B(\cdot)$ is polynomially bounded, we have $B(\lambda') \geq \Omega(\hat{m}(\lambda)/m) \geq \lambda^{\Omega(1)}$, which implies (again because $B(\cdot)$ is polynomially bounded) that $\lambda' \geq \lambda^{\Omega(1)}$.

If R is a size-S enumerable relation, then the statistical mode $\mathcal{H}^R = \{\mathcal{H}^R_\lambda\}_{\lambda\in\mathbb{N}}$ for R is nearly the same as $\mathcal{H}$, with the sole difference between $\mathcal{H}^R_\lambda$ and $\mathcal{H}_\lambda$ being in the key sampling process. In $\mathcal{H}^R_\lambda$, the ciphertext ct is sampled as $\mathsf{ct} \leftarrow \mathsf{Enc}\big(\mathsf{sk}, (\mathsf{sk}, E)\big)$, where E is a size-$S(\lambda)$ circuit enumerating $R|_{n(\lambda)}$. The indistinguishability of $\mathcal{H}^R$ from $\mathcal{H}$ follows immediately from the circular security of $\mathcal{FHE}$.

We next show that all functions in the support of $\mathcal{H}^R_\lambda$ avoid R. Let $\mathsf{sk} \in \{0,1\}^{\lambda'}$ be arbitrary and let ct be arbitrary in the support of $\mathsf{Enc}\big(\mathsf{sk}, (\mathsf{sk}, E)\big)$. Suppose for the sake of contradiction that for some $x \in \{0,1\}^{n(\lambda)}$, it holds that $H_{\mathsf{ct}}(x) \in R(x) \subseteq E(x)$. Then applying $\mathsf{Dec}(\mathsf{sk}, \cdot)$ to both sides, we would have for some $y \in E(x)$,

$$\begin{aligned}
\mathsf{Dec}(\mathsf{sk}, y) &= \mathsf{Dec}\big(\mathsf{sk}, H_{\mathsf{ct}}(x)\big) \\
&= \mathsf{Dec}\Big(\mathsf{sk}, \mathsf{Eval}\Big(C_x, \mathsf{Enc}\big(\mathsf{sk}, (\mathsf{sk}, E)\big)\Big)\Big) \\
&= C_x(\mathsf{sk}, E),
\end{aligned}$$

which contradicts the construction of C_x. □

6 From Enumerable to Projection-Enumerable Relations

In this section we re-cast the results of Holmgren, Lombardi, and Rothblum [HLR21] in a form more amenable to our use. Loosely speaking, [HLR21] proved that any hash family that is correlation intractable for all *efficiently enumerable* relations can be used to build a hash family that is correlation intractable for the broader class of relations $R \subseteq X \times Y^t$ whose *composition* with any *projection* is efficiently enumerable. That is, given any x and i, one can efficiently enumerate the set $\{\pi_i(y) : (x, y) \in R\}$, where $\pi_i(y_1, \ldots, y_t) = y_i$. We call this latter class of relations efficiently projection-enumerable.

Definition 6.1 (Projection-Enumerable Relations)**.** *We say that a relation $R \subseteq X \times [q]^t$ is (S,ℓ)-*projection-enumerable *if there is a size-S circuit e such that for all $i \in [t]$, $e(i, \cdot)$ is an ℓ-enumerator (as in Definition 5.1) for the composition $\pi_i \circ R$, where $\pi_i : [q]^t \to [q]$ denotes the projection $\pi_i(y_1, \ldots, y_t) := y_i$.*

More concretely, if $(x, \mathbf{y}) \in R$, then $y_i \in e(i, x)$ for all $i \in [t]$. We say that such a circuit e is an ℓ-projection-enumerator *for R.*

The main idea of [HLR21] is the following. To construct a CI hash for an efficiently ℓ-projection-enumerable relation

$$R \subseteq X \times [q]^t,$$

first construct a hash H' that outputs a short seed, then compose H' with an error-correcting code Enc that "expands" this seed to a value in $[q]^t$.

It suffices for H' to be correlation intractable for $\mathsf{Enc}^{-1} \circ R$ — if $(x, \mathsf{Enc}(H'(x))) \in R$, then by definition $(x, H'(x)) \in \mathsf{Enc}^{-1} \circ R$. As long as t is sufficiently large, [HLR21] show how to choose Enc such that $\mathsf{Enc}^{-1} \circ R$ is (efficiently) $\lambda^{O(1)}$-enumerable.

Lemma 6.2 (Implicit in the proof of [HLR21, Theorem 5.1])**.** *Let $n, q, t, S, \ell \in \mathbb{N}$ denote integers that are polynomially bounded functions of a security parameter λ, satisfying $\ell < q$ and $t \geq \lambda^{\Omega(1)}/\log(q/\ell)$.*

Then there exists a function $\mathsf{Enc} : \{0,1\}^k \to [q]^t$ (with $k \geq \lambda^{\Omega(1)}$) such that if $R \subseteq \{0,1\}^n \times [q]^t$ is any (S,ℓ)-projection-enumerable relation, then the composition $\mathsf{Enc}^{-1} \circ R$ is (L, L)-enumerable for $L = \lambda^{O(1)}$.

Moreover, such a (circuit computing) Enc can be sampled in $\lambda^{O(1)}$ time along with an oracle circuit $\mathsf{Enum}^{(\cdot)}$ such that with all but $\lambda^{-\omega(1)}$ probability, if P is any ℓ-projection-enumerator for $R \subseteq \{0,1\}^n \times [q]^t$, then Enum^P is an L-enumerator for $\mathsf{Enc}^{-1} \circ R$.

Formalizing the discussion above, Lemma 6.2 in conjunction with the results of Section 5 implies the following corollary.

Proposition 6.3. *Let $n, q, t, S, \ell \in \mathbb{N}$ be as in the hypotheses of Lemma 6.2, and assume either:*

- *the existence of indistinguishability obfuscation and the hardness of learning with errors; or*
- *the existence of circular-secure FHE.*

Then there exists a p.p.t. sampleable hash family $\mathcal{H}$ that is somewhere statistically correlation intractable for the set of (S, ℓ)-projection-enumerable relations $R \subseteq \{0,1\}^n \times [q]^t$.

Proof. Fix $k \geq \lambda^{\Omega(1)}$ and $L \leq \lambda^{O(1)}$ as in Lemma 6.2.

Let $\mathcal{H}' = \{\mathcal{H}'_\lambda\}$ be a p.p.t.-sampleable hash family, consisting of functions mapping $\{0,1\}^n \to \{0,1\}^k$, such that:

- $\mathcal{H}'$ is SS-CI for any (L, L)-enumerable relation R', and
- An R'-avoiding mode for $\mathcal{H}'$ is efficiently sampleable given an (L, L)-enumerator for R'.

Such a hash family is guaranteed to exist by either Theorem 5.5 or Theorem 5.9.

We define $\mathcal{H} = \mathsf{Enc} \circ \mathcal{H}'$, where Enc is a circuit sampled from the distribution given by Lemma 6.2. Let R be any relation that is (S, ℓ)-projection-enumerable by P. We claim that $\mathcal{H}$ is SS-CI for R, and moreover that an R-avoiding mode for $\mathcal{H}$ is efficiently sampleable given P.

Sampling $(\mathsf{Enc}, \mathsf{Enum}^{(\cdot)})$ as in Lemma 6.2, it holds with high probability that $R' := \mathsf{Enc}^{-1} \circ R$ is (L, L)-enumerable by Enum^P and hence there is an R'-avoiding mode $\mathcal{F}'_{\mathsf{Enc}}$ of $\mathcal{H}'$ that is efficiently sampleable jointly with Enc given P.

Finally, we claim that $\mathsf{Enc} \circ \mathcal{F}'_{\mathsf{Enc}}$ is a p.p.t.-sampleable R-avoiding mode of $\mathcal{H}$. Indeed, $\mathsf{Enc} \circ \mathcal{F}'_{\mathsf{Enc}}$:

- is indistinguishable from $\mathcal{H}$ because every $\mathcal{F}'_{\mathsf{Enc}}$ is indistinguishable from $\mathcal{H}'$,
- avoids R because if $\Big(x, \mathsf{Enc}\big(f'(x)\big)\Big) \in R$ for some x, then $\big(x, f'(x)\big) \in \mathsf{Enc}^{-1} \circ R$.
- is p.p.t.-sampleable given P because $(\mathsf{Enc}, \mathsf{Enum}^{(\cdot)})$ is p.p.t.-sampleable and $\mathcal{F}'_{\mathsf{Enc}}$ is p.p.t.-sampleable given Enum^P.

□

7 Separating IND-SOA from IND-CCA

This section closely follows that of Hofheinz, Rao, and Wichs [HRW16]; the difference is that we do not require as strong security properties from the underlying obfuscator $\mathcal{O}$ and the hash family $\mathcal{H}$. Specifically, our separation relies only on the existence of IO and the hardness of LWE.

Theorem 7.1. *Assume the hardness of learning with errors (LWE), and the existence of an indistinguishability obfuscator $\mathsf{i}\mathcal{O}$. Then there exists a public-key encryption scheme that is IND-CCA secure but not IND-SOA secure.*

7.1 An SOA Helper Circuit

The idea of the separation of [HRW16], which we follow, is to augment the public key of a public-key encryption scheme with an obfuscated "helper" circuit that makes the scheme insecure against selective opening attacks, while preserving IND-CPA (and even IND-CCA) security. This circuit is depicted in Fig. 7 below.

Embedded Values: encryption public key pk, decryption circuit D, and hash circuit H
Inputs: a ciphertext tuple $\mathbf{c} = (c_1, \ldots, c_{3\lambda})$ and openings $(\mu_1, r_1), \ldots, (\mu_\lambda, r_\lambda)$.

1. Compute $(i_1, \ldots, i_\lambda) := H(\mathbf{c})$.
2. Check that $c_1, \ldots, c_{3\lambda}$ are all distinct, and that for each $j \in [\lambda]$, $c_{i_j} = \mathsf{Enc}(\mathsf{pk}, \mu_j; r_j)$. If not, then output $\bot$.
3. For $j \in [\lambda]$, set $m_{i_j} := \mu_j$; for all $i \in [3\lambda] \setminus \{i_1, \ldots, i_\lambda\}$, set $m_i := D(c_i) \in \mathbb{F} \cup \{\bot\}$.
4. Use Reed-Solomon decoding to find a degree-λ polynomial $F : \mathbb{F} \to \mathbb{F}$ with a maximally sized "agreement set" $I_F = \{i : F(i) = m_i\}$. If $|I_f| \leq 2\lambda$ for all degree-λ polynomials f, then output $\bot$.
5. If $F(i_j) = \mu_j$ for all $j \in [\lambda]$, then output $F(0)$.
6. Otherwise output $\bot$.

Figure 7: The circuit $\mathsf{SOAHelper}(\mathsf{pk}, D, H)$.

While it is straight-forward to see that this augmentation makes the encryption scheme not IND-SOA secure, the bulk of the work is to ensure that the scheme still satisfies \$-IND-CCA security. We will prove this when PKE is a *puncturable* encryption scheme, and H is sampled from a hash family satisfying a notion of correlation intractability that we can instantiate either from IO and LWE, or from circular-secure fully homomorphic encryption.

We will need somewhere statistical correlation intractability for a specific class of relations, which was also defined in [HRW16].

Definition 7.2. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with message space* $\mathbb{F} = \{\mathbb{F}_\lambda\}$ *and ciphertext space* $\mathcal{C}_\lambda$*. Then for* $\mathsf{sk} \in \{0,1\}^\lambda$*, we define the relation* $R^{\mathsf{PKE}}_{\mathsf{sk}} \subseteq \mathcal{C}_\lambda^{3\lambda} \times [3\lambda]^\lambda$ *such that* $\big((c_i)_{i \in [3\lambda]}, (i_j)_{j \in [\lambda]}\big) \in R^{\mathsf{PKE}}_{\mathsf{sk}}$ *iff* $i_j \in I$ *for all* $j \in [\lambda]$*, where the set* I *is defined as follows.*

> *Compute* $m_i := \mathsf{Dec}(\mathsf{sk}, c_i)$ *for all* $i \in [3\lambda]$*, and let* $F : \mathbb{F} \to \mathbb{F}$ *be a degree-*λ *polynomial that maximizes the size of the set* $I_F = \big\{i : F(i) = m_i\big\}$*.*
>
> *If* $|I_F| = 2\lambda + 1$*, then define* $I = I_F$*. Otherwise, define* $I = \emptyset$*.*

The point of defining $R^{\mathsf{PKE}}_{\mathsf{sk}}$ is that if H is a function that perfectly avoids $R^{\mathsf{PKE}}_{\mathsf{sk}}$, then it does not affect the functionality of $\mathsf{SOAHelper}$ if we change the decryption subroutine D in $\mathsf{SOAHelper}$ to one that returns $\bot$ when given the challenge ciphertext $c^\star$. In particular, this modified decryption subroutine can be implemented with a punctured secret key, and we can then argue that $\mathsf{SOAHelper}$ does not allow the adversary to learn anything about its challenge ciphertext.

We will use the following notation. For a function f, we write $f_{\setminus S}$ to denote the function that agrees with f on all inputs not in S, but outputs $\bot$ for all inputs in S.

Lemma 7.3. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme, and let* $\mathsf{pk} = \mathsf{Gen}(\mathsf{sk})$*. If* h *is a function that perfectly avoids* $R^{\mathsf{PKE}}_{\mathsf{sk}}$ *and* c *is any string, then* $\mathcal{S} := \mathsf{SOAHelper}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, \cdot), h)$ *and* $\mathcal{S}' := \mathsf{SOAHelper}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{c\}}, h)$ *are functionally equivalent.*

Proof. Let $\big((c_i)_{i \in [3\lambda]}, (\mu_j, r_j)_{j \in [\lambda]}\big)$ be an arbitrary input. When $\mathcal{S}$ is executed on this input, let $(m_i)_{i \in [3\lambda]}$, F, and I_f (for all degree-λ polynomials f) denote the local variables of the same name that are computed as part of this execution. Define $(m'_i)_{i \in [3\lambda]}$, F', and I'_f (for all degree-λ polynomials f) analogously for the execution of $\mathcal{S}'$.

We prove that $\mathcal{S}$ and $\mathcal{S}'$ produce the same output by casework:

- If $m_i = m'_i$ for all $i \in [3\lambda]$, then clearly $\mathcal{S}$ and $\mathcal{S}'$ produce the same output.

- Otherwise, there exists some $i^\star \in [3\lambda]$ such that

$$m_i' = \begin{cases} \perp & \text{if } i = i^\star \\ m_i & \text{otherwise.} \end{cases} \tag{7}$$

  This implies for all f that

$$|I_f| - 1 \leq |I_f'| \leq |I_f|. \tag{8}$$

  Consider which line of $\mathcal{S}$ returns the output (the possibilities are 2, 4, 5, or 6).

  – If line 2, then the output is determined before $\mathcal{S}$ and $\mathcal{S}'$ differ, so the outputs must be the same.
  – If line 4, then $\mathcal{S}$ outputs $\perp$ because $|I_f| \leq 2\lambda$ for all degree-λ polynomials f. But $|I_f'| \leq |I_f|$ by Eq. (8), so $\mathcal{S}'$ outputs $\perp$ too.
  – If line 5 or 6, we consider $|I_F|$ (which must be at least $2\lambda + 1$).
    * If there is some f' satisfying $|I_{f'}'| \geq 2\lambda+1$ (which in particular follows if $|I_F| \geq 2\lambda+2$ because $|I_F'| \geq |I_F| - 1$), then $\mathcal{S}'$ produces its output in line 5 or 6. We have $F = F'$ because distinct degree-λ univariate polynomials can agree on at most λ points.
      Hence $\mathcal{S}$ and $\mathcal{S}'$ produce the same output.
    * If $|I_{f'}'| \leq 2\lambda$ for all f' (which as above implies that $|I_F| = 2\lambda + 1$), then $\mathcal{S}'$ outputs $\perp$ in line 4. $\mathcal{S}$ also outputs $\perp$ on line 5, because H avoids $R_{\mathsf{sk}}^{\mathsf{PKE}}$, meaning that there is some j such that $F(i_j) \neq \mu_j$.

□

We observe that $R_{\mathsf{sk}}^{\mathsf{PKE}}$ is projection-enumerable, which loosely speaking implies the existence of a hash family that is SS-CI for $R_{\mathsf{sk}}^{\mathsf{PKE}}$.

Claim 7.4. *For any public-key encryption scheme* PKE*, there exists a polynomially bounded function* $S : \mathbb{N} \to \mathbb{N}$ *such that for any* $\lambda \in \mathbb{N}$ *and* $\mathsf{sk} \in \{0,1\}^\lambda$*, the relation* $R_{\mathsf{sk}}^{\mathsf{PKE}}$ *is* $(S, 2\lambda+1)$*-projection enumerable.*

Proof. Given $\lambda \in \mathbb{N}$, $\mathsf{sk} \in \{0,1\}^\lambda$, $\mathbf{c} \in \mathcal{C}_\lambda^{3\lambda}$,and $j \in [\lambda]$, the definition of $R_{\mathsf{sk}}^{\mathsf{PKE}}$ describes how to either:

- compute a set I of size $2\lambda + 1$ such that $R_{\mathsf{sk}}^{\mathsf{PKE}}(\mathbf{c}) \subseteq I^\lambda$; or
- determine that $R_{\mathsf{sk}}^{\mathsf{PKE}}(\mathbf{c}) = \emptyset$, which is a strictly stronger conclusion — in this case, we can define I to be an arbitrary set of size $2\lambda + 1$ and it will still vacuously hold that $R_{\mathsf{sk}}^{\mathsf{PKE}}(\mathbf{c}) \subseteq I^\lambda$.

The only computationally non-trivial step in the definition of $R_{\mathsf{sk}}^{\mathsf{PKE}}$ is the computation of the degree-λ polynomial $F : \mathbb{F} \to \mathbb{F}$. This task is equivalent to decoding Reed-Solomon codes with errors in the unique-decoding regime, and is well-known to have polynomial-time algorithms, e.g.[3] that of Welch and Berlekamp [WB86].

Given an algorithm for decoding Reed-Solomon codes, the definition of $R_{\mathsf{sk}}^{\mathsf{PKE}}$ clearly implies that it is $\big(\mathrm{poly}(\lambda), 2\lambda + 1\big)$-projection-enumerable, and moreover a $(2\lambda + 1)$-projection-enumerator is efficiently computable given sk. □

Corollary 7.5. *Assume the existence of an indistinguishability obfuscator and the hardness of LWE.*

Then there exists a p.p.t. sampleable hash family $\mathcal{H}$ *that is somewhere statistically correlation intractable for all* $\{R_{\mathsf{sk}}^{\mathsf{PKE}}\}_{\lambda\in\mathbb{N},\mathsf{sk}\in\{0,1\}^\lambda}$*, and moreover an* $R_{\mathsf{sk}}^{\mathsf{PKE}}$*-avoiding mode of* $\mathcal{H}$ *is efficiently sampleable given* sk*.*

[3] We refer the reader to the notes of Mary Wootters [Woo] for a more detailed history of Reed-Solomon decoding, including the more efficient but complicated algorithms of Peterson [Pet60] and Berlekamp-Massey [Mas69] that preceded Berlekamp-Welch.

7.2 The Separating Encryption Scheme

Construction 7.6. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a puncturable encryption scheme with auxiliary algorithms* $(\widetilde{\mathsf{Enc}}, \mathsf{Puncture})$*. Then our construction* $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *is defined such that:*

- *On input* $\mathsf{sk} \in \{0,1\}^\lambda$*,* Gen' *outputs* $\mathsf{pk}' = (\mathsf{pk}, H, \widetilde{\mathsf{SOAHelper}})$*, where:*
  1. $\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$*;*
  2. H *is sampled from a hash family* $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ *that is SS-CI for all relations* $\{\mathcal{R}^{\mathsf{PKE}}_{\mathsf{sk}}\}_{\mathsf{sk}\in\{0,1\}^\lambda}$ *as defined in Definition 7.2; and*
  3. $\widetilde{\mathsf{SOAHelper}}$ *is sampled as*

$$\widetilde{\mathsf{SOAHelper}} \leftarrow \mathsf{i}\mathcal{O}\Big(\mathsf{Pad}_p\big(\mathsf{SOAHelper}(\mathsf{pk}, D, H)\big)\Big),$$

     where:
     - D *is a circuit mapping* $c \mapsto \mathsf{Dec}(\mathsf{sk}, c)$*;*
     - $\mathsf{SOAHelper}(\mathsf{pk}, D, H)$ *is a circuit defined in Fig. 7; and*
     - $p = p(\lambda)$ *is a polynomially-bounded integer such that* $p(\lambda) \geq |\mathsf{SOAHelper}(\mathsf{pk}, D, h)|$ *for any hash family* $\mathbb{H}$ *and circuit* D *listed in Fig. 9, and any* h *in the support of* $\mathbb{H}$*.*
- $\mathsf{Enc}'\big((\mathsf{pk}, H, \widetilde{\mathsf{SOAHelper}}), m\big)$ *outputs* $\mathsf{Enc}(\mathsf{pk}, m)$
- $\mathsf{Dec}'(\mathsf{sk}, c)$ *outputs* $\mathsf{Dec}(\mathsf{sk}, c)$*.*

Proposition 7.7. *Construction 7.6 is not SSS-SOA secure.*

Proof. The following is a polynomial-time strategy for winning the SSS-SOA game with probability 1.

1. Given $\mathsf{pk}' = (\mathsf{pk}, h, \widetilde{\mathsf{SOAHelper}})$ and ciphertexts $\mathbf{c} = (c_1, \ldots, c_{3\lambda})$ from the SSS-SOA challenger, compute $(i_1, \ldots, i_\lambda) \leftarrow h(\mathbf{c})$, and respond with $(i_1, \ldots, i_\lambda)$.
2. When the challenger responds with openings $\mathbf{o} = \big((\mu_1, r_1), \ldots, (\mu_\lambda, r_\lambda)\big)$ such that $c_{i_j} = \mathsf{Enc}(\mathsf{pk}, \mu_j; r_j)$ for all $j \in [\lambda]$, compute $m^\star := \widetilde{\mathsf{SOAHelper}}(\mathbf{c}, \mathbf{o})$ and output $m^\star$.

□

Proposition 7.8. *Construction 7.6 is \$-IND-CCA secure.*

Proof. Let $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ be any polynomial-size circuit ensemble, and suppose for contradiction that the Bernoulli random variable $\mathsf{Exp}_{\text{\$-ind-cca}}(1^\lambda, \mathsf{PKE}', \mathcal{A})$, is equal to 1 with probability noticeably larger than 1/2. We will show a contradiction by constructing "hybrid" random variables $\mathsf{Hyb}_0, \ldots, \mathsf{Hyb}_8$ such that $\mathsf{Exp}_{\text{\$-ind-cca}}(1^\lambda, \mathsf{PKE}', \mathcal{A}) \equiv \mathsf{Hyb}_0 \approx \cdots \approx \mathsf{Hyb}_8 \equiv \mathrm{Ber}(1/2)$. Each of these hybrids shares a similar structure, depicted in Fig. 8. The hybrids differ only in that the following parameters are instantiated differently, as specified in Fig. 9.

- what "decryption oracle" $\mathcal{O}$ is given to the adversary;
- what "embedded decrypter" circuit D is hard-wired in $\mathsf{SOAHelper}$);
- what hash family $\mathbb{H}$ the hash function H is sampled from; and
- what "challenge ciphertext" $c^\star$ is given to the adversary.

1. $\mathsf{sk} \leftarrow \{0,1\}^\lambda$; $\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})$;
2. $H \leftarrow \boxed{\mathbb{H}}$;
3. $m_0, m_1 \leftarrow \mathcal{M}_\lambda$;
4. $b \leftarrow \{0,1\}$;
5. $c_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$;
6. $\tilde{c} \leftarrow \widetilde{\mathsf{Enc}}(\mathsf{pk})$;
7. $\mathsf{sk}_{\setminus\{\tilde{c}, c_b\}} \leftarrow \mathsf{Puncture}\big(\mathsf{sk}, \{\tilde{c}, c_b\}\big)$;
8. $\widetilde{\mathsf{SOAHelper}} \leftarrow \mathsf{i}\mathcal{O}\Big(1^\lambda, \mathsf{Pad}_p\big(\mathsf{SOAHelper}(\mathsf{pk}, \boxed{D}, H)\big)\Big)$
9. $b' \leftarrow \mathcal{A}^{\boxed{\mathcal{O}}}(\mathsf{pk}, m_0, m_1, \boxed{c^\star}, \widetilde{\mathsf{SOAHelper}}, H)$
10. Return 1 if $b = b'$, and 0 otherwise.

Figure 8: The hybrid experiments for the proof of Proposition 7.8, parameterized by $\mathcal{O}$, D, $\mathbb{H}$, and $c^\star$ (all of which are outlined in a box to visually emphasize the parts of the experiment that vary between hybrids)

Hybrid	Decryption Oracle $\mathcal{O}$	Embedded Decrypter D	Hash Family $\mathbb{H}$	Challenge $c^\star$
0	$\mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{c_b\}}$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\mathcal{H}_\lambda$	c_b
1	$\mathsf{sk}_{\setminus\{\tilde{c}, c_b\}}$			
2			$\tilde{\mathcal{H}}_\lambda^{R_{\mathsf{sk}}^{\mathsf{PKE}}}$	
3		$\mathsf{sk}_{\setminus\{\tilde{c}, c_b\}}$		
4			$\mathcal{H}_\lambda$	
5				$\tilde{c}$
6			$\tilde{\mathcal{H}}_\lambda^{R_{\mathsf{sk}}^{\mathsf{PKE}}}$	
7		$\mathsf{Dec}(\mathsf{sk}, \cdot)$		
8	$\mathsf{Dec}(\mathsf{sk}, \cdot)$			

Figure 9: The choice of parameters $\mathcal{O}$, D, $\mathbb{H}$, and $c^\star$ used in each hybrid experiment. Each blank space means "same as above." $\tilde{\mathcal{H}}^{R_{\mathsf{sk}}^{\mathsf{PKE}}}$ denotes an $R_{\mathsf{sk}}^{\mathsf{PKE}}$-avoiding mode of $\mathcal{H}$.

Lemma 7.9. Hyb_0 *is identical to* $\mathsf{Exp}_{\$\text{-ind-cca}}(1^\lambda, \mathsf{PKE}', \mathcal{A}_\lambda)$.

Proof. Hyb_0 describes the same process as $\mathsf{Exp}_{\$\text{-ind-cca}}(1^\lambda, \mathsf{PKE}', \mathcal{A}_\lambda)$, with only superficial changes in presentation. □

Lemma 7.10. $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$

Proof. The ciphertext sparseness property of PKE implies that with high probability, $\mathsf{Dec}(\mathsf{sk}, \tilde{c}) = \bot$. Thus the "modification" from Hyb_0 to Hyb_1 wherein the oracle D is changed from $\mathsf{Dec}(\mathsf{sk}, \cdot)$ to $\mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{\tilde{c}\}}$ with high probability changes nothing. □

Lemma 7.11. $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

Proof. Follows from the indistinguishability of $\tilde{\mathcal{H}}_\lambda^{R_{\mathsf{sk}}^{\mathsf{PKE}}}$ from $\mathcal{H}_\lambda$. □

Lemma 7.12. $\mathsf{Hyb}_2 \approx \mathsf{Hyb}_3$.

Proof. By Lemma 7.3, it holds with high probability when sampling

$$\begin{aligned}&\mathsf{sk} \leftarrow \{0,1\}^\lambda\\&\mathsf{pk} := \mathsf{Gen}(\mathsf{sk})\\&H \leftarrow \tilde{\mathcal{H}}^{R^{\mathsf{PKE}}_{\mathsf{sk}}}\end{aligned}$$

that for all c, $\mathsf{SOAHelper}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, \cdot), H)$ is functionally equivalent to $\mathsf{SOAHelper}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{c\}}, H)$, which is in turn (w.h.p.) functionally equivalent to $\mathsf{SOAHelper}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{c,\tilde{c}\}}, H)$ because $\mathsf{Dec}(\mathsf{sk}, \tilde{c}) = \bot$ w.h.p. when sampling $\tilde{c} \leftarrow \widetilde{\mathsf{Enc}}(\mathsf{pk})$. Finally, $\mathsf{Dec}(\mathsf{sk}, \cdot)_{\setminus\{c,\tilde{c}\}}$ is functionally equivalent to the punctured key $\mathsf{sk}\{c, \tilde{c}\}$. The indistinguishability of Hyb_2 and Hyb_3 then follows from the security of $\mathsf{i}\mathcal{O}$. ☐

Lemma 7.13. $\mathsf{Hyb}_3 \approx \mathsf{Hyb}_4$.

Proof. Analogous to Lemma 7.11 ☐

Lemma 7.14. $\mathsf{Hyb}_4 \approx \mathsf{Hyb}_5$.

Proof. Follows directly from the punctured security property of PKE. ☐

Lemma 7.15. $\mathsf{Hyb}_5 \approx \mathsf{Hyb}_6$.

Proof. Analogous to Lemma 7.11. ☐

Lemma 7.16. $\mathsf{Hyb}_6 \approx \mathsf{Hyb}_7$.

Proof. Analogous to Lemma 7.12. ☐

Lemma 7.17. $\mathsf{Hyb}_7 \approx \mathsf{Hyb}_8$.

Proof. Hyb_7 and Hyb_8 differ only when $\mathcal{A}$ manages to query $\mathcal{O}$ on c_b. However, conditioned on the adversary's view, the min-entropy of c_b is super-logarithmic in λ. This implies that $\mathcal{A}$ can only query $\mathcal{O}$ on c_b with probability that is negligible in λ. ☐

☐

References

[BDWY12] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 645–662. Springer, 2012.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2009.

[BP15] Elette Boyle and Rafael Pass. Limits of extractability assumptions with distributional auxiliary input. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 236–261. Springer, 2015.

[BSW16] Mihir Bellare, Igors Stepanovs, and Brent Waters. New negative results on differing-inputs obfuscation. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 792–821. Springer, 2016.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1082–1090. ACM, 2019.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 91–122. Springer, 2018.

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 639–648. ACM, 1996.

[CHV15] Aloni Cohen, Justin Holmgren, and Vinod Vaikuntanathan. Publicly verifiable software watermarking. *IACR Cryptol. ePrint Arch.*, page 373, 2015.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 523–534. IEEE Computer Society, 1999.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[GGHW17] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. *Algorithmica*, 79(4):1353–1373, 2017.

[GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 505–522. Springer, 2016.

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

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 850–858. IEEE Computer Society, 2018.

[HLR21] Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 750–760. ACM, 2021.

[HR14] Dennis Hofheinz and Andy Rupp. Standard versus selective opening security: Separation and equivalence results. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 591–615. Springer, 2014.

[HRW16] Dennis Hofheinz, Vanishree Rao, and Daniel Wichs. Standard security does not imply indistinguishability under selective opening. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 121–145, 2016.

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 668–697. Springer, 2015.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[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, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2022.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251. Springer, 2017.

[Mas69] J. Massey. Shift-register synthesis and bch decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2003.

[Pet60] W. Wesley Peterson. Encoding and error-correction procedures for the bose-chaudhuri codes. *IRE Trans. Inf. Theory*, 6(4):459–470, 1960.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 89–114. Springer, 2019.

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

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[SW21] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM J. Comput.*, 50(3):857–908, 2021.

[WB86] Lloyd R Welch and Elwyn R Berlekamp. Error correction for algebraic block codes, December 30 1986. US Patent 4,633,470.

[Woo] Mary Wootters. Algebraic error correcting codes. http://web.stanford.edu/class/cs250/. Accessed: 2024-02-12.