# On Pseudorandom Encodings

Thomas Agrikola[1(✉)], Geoffroy Couteau[2], Yuval Ishai[3], Stanisław Jarecki[4], and Amit Sahai[5]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
thomas.agrikola@kit.edu
[2] IRIF, Paris-Diderot University, CNRS, Paris, France
couteau@irif.fr
[3] Technion, Haifa, Israel
yuvali@cs.technion.ac.il
[4] UC Irvine, Irvine, USA
stasio@ics.uci.edu
[5] UCLA, Los Angeles, USA
sahai@cs.ucla.edu

**Abstract.** We initiate a study of *pseudorandom encodings*: efficiently computable and decodable encoding functions that map messages from a given distribution to a random-looking distribution. For instance, every distribution that can be perfectly and efficiently compressed admits such a pseudorandom encoding. Pseudorandom encodings are motivated by a variety of cryptographic applications, including password-authenticated key exchange, "honey encryption" and steganography.

The main question we ask is whether *every* efficiently samplable distribution admits a pseudorandom encoding. Under different cryptographic assumptions, we obtain positive and negative answers for different flavors of pseudorandom encodings, and relate this question to problems in other areas of cryptography. In particular, by establishing a two-way

T. Agrikola—Supported by ERC Project PREP-CRYPTO 724307 and by the German Federal Ministry of Education and Research within the framework of the project KASTEL_SKI in the Competence Center for Applied Security Technology (KASTEL).
G. Couteau—Supported by ERC Projects PREP-CRYPTO 724307 and CryptoCloud 339563. Work done in part while visiting UCLA and the Technion.
Y. Ishai—Supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, BSF grant 2018393, ISF grant 2774/20, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India. Work done in part while visiting UCLA.
S. Jarecki—Supported by the NSF SaTC award 1817143.
A. Sahai—Supported in part by DARPA SAFEWARE and SIEVE awards, NTT Research, NSF Frontier Award 1413955, and NSF grant 1619348, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, NTT Research, or the U.S. Government.

relation between pseudorandom encoding schemes and efficient invertible sampling algorithms, we reveal a connection between adaptively secure multiparty computation for randomized functionalities and questions in the domain of steganography.

## 1 Introduction

The problem of *compression* has been extensively studied in the field of information theory and, more recently, in computational complexity and cryptography [23,27,40,42]. Informally, given a distribution $X$, compression aims to efficiently encode samples from $X$ as short strings while at the same time being able to efficiently recover these samples. While the typical information-theoretic study of compression considers the case of compressing multiple independent samples from the same source $X$, its study in computer science, and in particular in this work, considers the "single-shot" case. Compression in this setting is closely related to *randomness condensers* [18,34,38,39] and *resource-bounded Kolmogorov complexity* [32,33] – two well-studied problems in computational complexity. Randomness condensers, which relax randomness extractors, are functions that efficiently map an input distribution into an output distribution with a higher entropy rate. A randomness condenser can be viewed as an efficient compression algorithm, without a corresponding efficient decompression algorithm. The resource-bounded Kolmogorov complexity of a string is the smallest description length of an efficient program that outputs this string. This program description can be viewed as a compressed string, such that decoding is efficiently possible, while finding the compressed string may be inefficient.

An important property of efficient compression algorithms, which combines the efficiency features of randomness condensers and resource-bounded Kolmogorov complexity, is their ability to efficiently produce "random-looking" outputs while allowing the original input to be efficiently recovered. Despite the large body of work on compression and its computational variants, this fundamental property has, to our knowledge, never been the subject of a dedicated study. In this work, we fill this gap by initiating such a study. Before formalizing the problem, we give a simple motivating example.

Consider the goal of encrypting a sample $x$ from a distribution $X$ (say, a random 5-letter English word from the Merriam-Webster Dictionary) using a low-entropy secret key $k$. Applying a standard symmetric-key encryption scheme with a key derived from $k$ gives rise to the following brute-force attack: Try to decrypt with different keys until obtaining $x'$ in the support of $X$. In the typical case that wrong keys always lead to $x'$ outside the support of $X$, this attack successfully recovers $x$. Variants of this attack arise in different scenarios, including password-authenticated key exchange [4], honey encryption [30], subliminal communication and steganography [26], and more. A natural solution is to use perfect compression: if $x$ can be compressed to a *uniformly random* string $\hat{x} \in \{0,1\}^n$ before being encrypted, it cannot be distinguished from another random string $\hat{x}' \in \{0,1\}^n$ obtained by trying the wrong key. Note, however,

that compression may be an overkill for this application. Instead, it suffices to efficiently encode $x$ into a (possibly longer) *pseudorandom* string from which $x$ can be efficiently decoded. This more general solution motivates the question we consider in this work.

*Encoding into the Uniform Distribution.* We initiate the study of encoding distributions into a random-looking distribution. Informally, we say that a distribution ensemble $X_\lambda$ admits a *pseudorandom encoding* if there exist efficient encoding and decoding algorithms $(\mathsf{E}_X, \mathsf{D}_X)$, where $\mathsf{D}_X$ is deterministic, such that

$$\Pr\left[y \leftarrow X_\lambda \colon \mathsf{D}_X(\mathsf{E}_X(y)) = y\right] \text{ is overwhelming and} \tag{1}$$

$$\left\{y \leftarrow X_\lambda \colon \mathsf{E}_X(y)\right\} \approx U_{n(\lambda)}. \tag{2}$$

Here, "$\approx$" denotes some notion of indistinguishability (we will consider both computational and statistical indistinguishability), and the probability is over the randomness of both $\mathsf{E}_X$ and $X_\lambda$. The polynomial $n(\lambda)$ denotes the output length of the encoding algorithm $\mathsf{E}_X$. We refer to Eq. (1) as *correctness* and to Eq. (2) as *pseudorandomness*. It will also be useful to consider distribution ensembles parameterized by an input $m$ from a language $L$. We say that such a distribution ensemble $(X_m)_{m\in L}$ admits a pseudorandom encoding if there exist efficient algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ as above satisfying correctness and pseudorandomness for all $m \in L$, where $\mathsf{E}_X$ and $\mathsf{D}_X$ both additionally receive $m$ as input. Note that we insist on the decoding algorithm being efficient. This is required for our motivating applications.[1] Note also that encoding and decoding above are *keyless*; that is, we want encoded samples to be close to uniform *even though anyone can decode them*. This is a crucial distinction from, for instance, encryption schemes with pseudorandom ciphertexts, which look uniformly distributed to everyone except the owner of the decryption key, and cannot be efficiently decrypted except by the owner of the decryption key. Here, we seek to simultaneously achieve pseudorandomness and correctness for all parties.

Our motivation for studying pseudorandom encodings stems from the fact that this very natural problem appears in a wide variety of – sometimes seemingly unrelated – problems in cryptography. We already mentioned steganography, honey encryption, and password-authenticated key exchange; we will cover more such connections in this work. Yet, this notion of encoding has to our knowledge never been studied systematically. In this work we study several natural flavors of this notion, obtain positive and negative results about realizing them, and map their connections with other problems in cryptography.

The main focus of this work is on the hypothesis that *all* efficiently samplable distributions admit a pseudorandom encoding. Henceforth, we refer to this hypothesis the *pseudorandom encoding hypothesis* (PREH).

For describing our results, it will be convenient to use the following general notion of efficiently samplable distributions. A distribution family ensemble

[1] Without this requirement, the problem can be solved using non-interactive commitment schemes with the additional property that commitments are pseudorandom (which exist under standard cryptographic assumptions).

$(X_m)_{m\in L}$ (for some language $L \subseteq \{0,1\}^*$) is efficiently samplable if there exists a probabilistic polynomial time (PPT) algorithm $S$ such that $S(m)$ is distributed according to $X_m$ for every $m \in L$. In case the distribution does not depend on additional inputs, $L$ can be considered equal to $\mathbb{N}$.

*Overview of Contributions.* Following is a brief summary of our main contributions. We will give an expanded overview of the contributions and the underlying techniques in the rest of this section.

- We provide a unified study of different flavors of pseudorandom encodings (PRE) and identify computational, randomized PRE in the CRS model as a useful and achievable notion.
- We establish a two-way relation between PRE and the previously studied notion of invertible sampling This reveals unexpected connections between seemingly unrelated problems in cryptography (e.g., between adaptively secure computation for general functionalities and "honey encryption").
- We bootstrap "adaptive PRE" from "static PRE" As a consequence, one can base succinct adaptively secure computation on standard $\mathsf{iO}$ as opposed to subexponential $\mathsf{iO}$ [15].
- We use PRE to obtain a compiler from standard secure multiparty computation (MPC) protocols to *covert* MPC protocols.

### 1.1 Flavors of Pseudorandom Encoding

The notion of pseudorandom encoding has several natural flavors, depending on whether the encoding algorithm is allowed to use randomness or not, and whether the pseudorandomness property satisfies a computational or information-theoretic notion of indistinguishability. We denote the corresponding hypotheses that every efficiently samplable distribution can be pseudorandomly encoded according to the above variants as $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$, $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_s}$, $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$ and $\mathsf{PREH}^{\mathsf{det}}_{\equiv_s}$.[2]

Further, we explore relaxations which rely on a trusted setup assumption: we consider the pseudorandom encoding hypothesis in the *common reference string model*, in which a common string sampled in a trusted way from some distribution is made available to the parties. This is the most common setup assumption in cryptography and it is standard to consider the feasibility of cryptographic primitives in this model to overcome limitations in the plain model. That is, we ask whether for every efficiently samplable distribution $X$, there exists an

[2] We note that not all efficiently samplable distributions can be pseudorandomly encoded with a deterministic encoding algorithm. For instance, a distribution which has one very likely event and many less likely ones requires one specific encoding to appear with high probability. Thus, we formally restrict the deterministic variants of the pseudorandom encoding hypothesis to only hold for "compatible" samplers, which still results in interesting connections. In this overview, however, we ignore this restriction.

efficiently samplable CRS distribution and efficient encoding and decoding algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ as above, such that correctness and pseudorandomness hold, where the encoding and decoding algorithm as well as the distinguisher receive the CRS as input, and the distributions in Eqs. (1) and (2) are additionally over the choice of the CRS.

Considering distributions which may depend on an input $m \in L$ further entails two different flavors. On the one hand, we consider the notion where inputs $m$ are chosen adversarially but *statically* (that is, independent of the CRS) and, on the other hand, we consider the stronger notion where inputs $m$ are chosen adversarially and *adaptively* depending on the CRS. We henceforth denote these variants by the prefix "$\mathsf{c}$" and "$\mathsf{ac}$", respectively.

*Static-to-Adaptive Transformation.* The adaptive notion, where inputs may be chosen depending on the CRS, is clearly stronger than the static notion. However, surprisingly, the very nature of pseudorandom encodings allows one to apply an indirection argument similar to the one used in [11, 12, 25], which yields a static-to-adaptive transformation.

**Theorem (informal).** *If all efficiently samplable distributions can be pseudorandomly encoded in the CRS model with a* static *choice of inputs, then all efficiently samplable distributions can be pseudorandomly encoded in the CRS model with an* adaptive *choice of inputs.*

Static-to-adaptive transformations in cryptography are generally non-trivial, and often come at a big cost in security when they rely on a "complexity leveraging" technique. This connection and its application we will discuss below are a good demonstration of the usefulness of the notion of pseudorandom encodings.

*Relaxing Compression.* The notion of statistical deterministic pseudorandom encodings recovers the notion of optimal compression. Hence, this conflicts with the existence of one-way functions.[3] In our systematic study of pseudorandom encodings, we gradually relax perfect compression in several dimensions, while maintaining one crucial property – the indistinguishability of the encoded distribution from true randomness.

*Example.* To illustrate the importance of this property, we elaborate on the example we outline at the beginning of the introduction, focusing more specifically on password-authenticated key exchange (PAKE). A PAKE protocol allows two parties holding a (low entropy) common password to jointly and confidentially generate a (high entropy) secret key, such that the protocol is resilient against offline dictionary attacks, and no adversary can establish a shared key with a party if he does not know the matching password. A widely used PAKE protocol due to Bellovin and Merritt [4] has a very simple structure: the parties use their low-entropy password to encrypt the flows of a key-exchange protocol

[3] If perfect compression exists, pseudorandom generators cannot exist (observation attributed to Levin in [23]).

using a block cipher. When the block cipher is modeled as a random cipher, it has the property that decrypting a ciphertext (of an arbitrary plaintext) under an incorrect secret key yields a fresh random plaintext. Thus, Bellovin and Merritt point out that the security of their PAKE protocol requires that "the message to be encrypted by the password must be indistinguishable from a random number." This is easy to achieve for Diffie-Hellman key exchange over the multiplicative group of integers modulo a prime $p$. However, for elliptic curve groups this is no longer the case, and one needs to resort to alternative techniques including nontrivial *point compression* algorithms that compress the representation of a random group element into a nearly uniform bitstring [6].

Clearly, our relaxation of compression does not preserve the useful property of obtaining outputs that are *shorter* than the inputs. However, the remaining pseudorandomness property is good enough for many applications.

In the following, we elaborate on our weakest notion of pseudorandom encodings, that is, pseudorandom encodings allowing the encoding algorithm to be randomized and providing a computational pseudorandomness guarantee. We defer the discussion on the stronger statistical or deterministic variants to Sect. 1.3, where we derive negative results for most of these stronger notions, which leaves computational randomized pseudorandom encodings as the "best possible" notion that can be realized for general distributions.

**Randomized, Computational Pseudorandom Encodings.** Computational randomized pseudorandom encodings allow the encoding algorithm to be randomized and require only computational pseudorandomness.

*Relation to Invertible Sampling.* We show a simple but unexpected connection with the notion of *invertible sampling* [9,17,22]. Informally, invertible sampling refers to the task of finding, given samples from a distribution, random coins that *explain* the sample. Invertible sampling allows to *obliviously* sample from distributions, that is, sampling from distributions *without knowing the corresponding secrets*. This can be useful for, e.g., sampling common reference strings without knowing the random coins or public keys without knowing the corresponding secret keys. A natural relaxation of this notion was systematically studied by Ishai, Kumarasubramanian, Orlandi and Sahai [29]. Concretely, a PPT sampler $S$ is *inverse samplable* if there exists an alternative PPT sampler $\overline{S}$ and a PPT inverse sampler $\overline{S}^{-1}$ such that

$$\begin{aligned}\big\{y \leftarrow S(1^\lambda)\colon y\big\} &\approx_{\mathsf{c}} \big\{y \leftarrow \overline{S}(1^\lambda)\colon y\big\},\\ \big\{y \leftarrow \overline{S}(1^\lambda; r)\colon (r, y)\big\} &\approx_{\mathsf{c}} \big\{y \leftarrow \overline{S}(1^\lambda)\colon (\overline{S}^{-1}(1^\lambda, y), y)\big\}.\end{aligned}$$

Note that the inverse sampling algorithm is only required to efficiently inverse-sample from another distribution $\overline{S}$, but this distribution must be computationally close to the distribution induced by $S$. The main question studied in [29] is whether *every* efficient sampler admits such an invertible sampler. They refer to this hypothesis as the *invertible sampling hypothesis* (ISH), and show that

ISH is equivalent to adaptive MPC for general randomized functionalities that may hide their internal randomness. In this work, we show the following two-way relation with pseudorandom encoding.

**Theorem (informal).** *A distribution admits a pseudorandom encoding* if and only if *it admits invertible sampling.*

Intuitively, the efficient encoding algorithm corresponds to the inverse sampling algorithm, and decoding an encoded string corresponds to sampling with the de-randomized alternative sampler $\overline{S}$. This equivalence immediately extends to all variants of pseudorandom encodings and corresponding variants of invertible sampling we introduce in this work. Invertible sampling is itself connected to other useful cryptographic notions, such as oblivious sampling, trusted common reference string generations, and adaptively secure computation (which we will elaborate upon below).

Building on this connection, the impossibility result of [29] translates to our setting. On a high level, extractable one-way functions (EOWFs) conflict with invertible sampling because they allow to extract a "secret" (in this case a preimage) from an image, independently of how it was computed. This conflicts with invertible sampling because invertible sampling is about sampling without knowing the secrets.

**Theorem (informal, [29]).** *Assuming the existence of extractable one-way functions (EOWF) and a non-interactive zero-knowledge proof system,* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ *does not hold.*

This suggests that towards a realizable notion of pseudorandom encodings, a further relaxation is due. Thus, we ask whether the above impossibility result extends to the CRS model. In the CRS model, the above intuition why ISH conflicts with EOWFs fails, because the CRS can contain an obfuscated program that samples an image using some secret, but does not output this secret.

Dachman-Soled, Katz, and Rao [16] (building on the universal deniable encryption construction of Sahai and Waters [35]) construct a so-called "explainability compiler" that implies $\mathsf{cISH}^{\mathsf{rand}}_{\approx_c}$ based on indistinguishability obfuscation[4] ($\mathsf{iO}$). By our equivalence theorem above, this implies pseudorandom encodings for all efficiently samplable distributions in the CRS model, with static choice of inputs, from $\mathsf{iO}$. Invoking the static-to-adaptive transformation detailed above, this also applies to the adaptive variant.

**Theorem (informal).** *Assuming the existence of (polynomially secure) indistinguishability obfuscation and one-way functions,* $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ *holds.*

[4] Informally, an $\mathsf{iO}$ scheme is a PPT algorithm that takes as input a circuit $C$ and produces another circuit $\mathsf{iO}(C)$ such that $C$ and $\mathsf{iO}(C)$ compute the same function, but $\mathsf{iO}(C)$ is unintelligible in the following sense. If two circuits $C_1$ and $C_2$ compute the same function, then $\mathsf{iO}(C_1)$ and $\mathsf{iO}(C_2)$ are computationally indistinguishable. The notion of $\mathsf{iO}$ was introduced in [2] and first instantiated in [21].

Note that [29] claim that their impossibility result extends to the CRS model, whereas the above theorem seems to suggest the opposite. We show that technically the result of [29] does extend to the CRS model at the cost of assuming *unbounded auxiliary-input* extractable one-way functions, a strong flavor of EOWFs that seems very unlikely to exist but cannot be unconditionally ruled out.

**Theorem (informal).** *Assuming the existence of extractable one-way functions with* unbounded *common auxiliary input and a non-interactive zero-knowledge proof system,* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ *does not hold.*

In fact, this apparent contradiction has been the source of some confusion in previous works: the work of [29] makes an informal claim that their impossibility result for ISH extends to the CRS model. However, due to the connection between ISH and adaptively secure MPC (which we will discuss in more details later on), this claim was challenged in [16]: the authors achieve a construction of adaptively secure MPC for all functionalities assuming $\mathsf{iO}$, which seemingly contradicts the claim of [29]. The authors of [16] therefore stated that the "impossibility result of Ishai et al. [...] does not hold in the CRS model." Our extension clarifies that the distinction is in fact more subtle: the result of [29] *does* extend to the CRS model, but at the cost of assuming EOWF with *unbounded auxiliary inputs*. This does not contradict the constructions based on $\mathsf{iO}$, because $\mathsf{iO}$ and EOWF with unbounded auxiliary inputs are known to be contradictory [5].

*Overview.* In Fig. 1, we provide a general summary of the many flavors of the pseudorandom encoding hypothesis, and how they relate to a wide variety of other primitives.

*Further Relaxation.* We further study an additional relaxation of pseudorandom encodings, where we allow the encoding algorithm to run in super-polynomial time. We show that this relaxed variant can be achieved from cryptographic primitives similar to *extremely lossy functions* [45], which can be based on the exponential hardness of the decisional Diffie-Hellman problem – a strong assumption, but (still) more standard than indistinguishability obfuscation. However, the applicability of the resulting notion turns out to be rather restricted.

### 1.2 Implications and Applications of Our Results

In this section, we elaborate on the implications of the techniques we develop and the results we obtain for a variety of other cryptographic primitives.

**New Results for Adaptively Secure Computation.** As mentioned above, a sampler admits invertible sampling if and only if it can be pseudorandomly encoded. A two-way connection between invertible sampling and *adaptively* secure MPC for general randomized functionalities was established in [29]. An

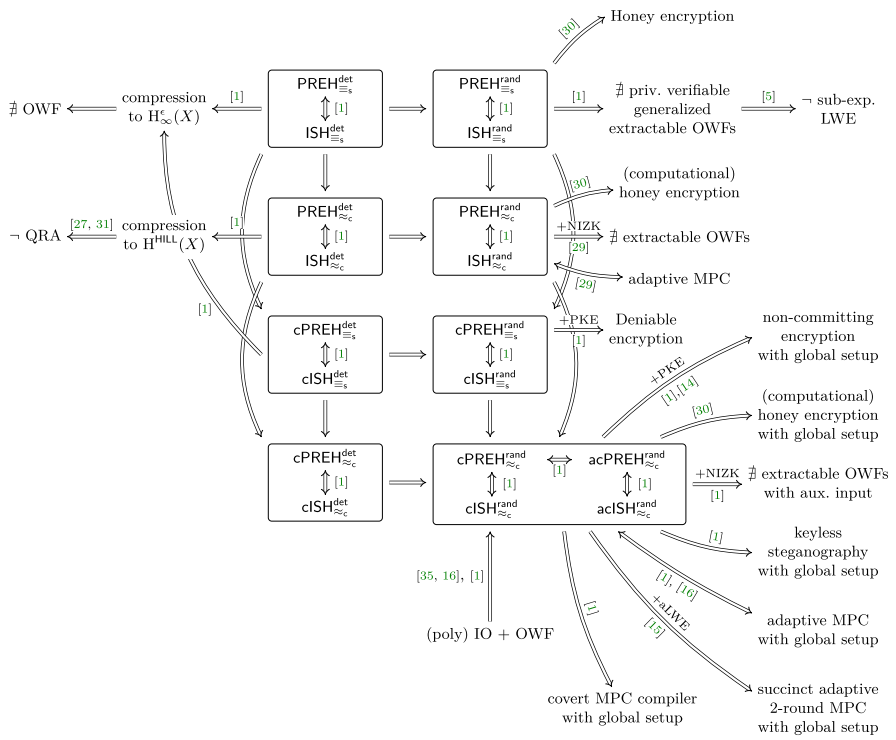


**Fig. 1.** An overview of the relations between the pseudorandom encoding hypothesis and other fields of cryptography and computational complexity theory. For simplicity, our static-to-adaptive transformation only appears in the computational, randomized setting in this overview, but also applies to the other settings. (Since the deterministic variants of the pseudorandom encoding hypothesis are impossible for some pathologic samplers, the arrows between deterministic and randomized variants of the pseudorandom encoding hypothesis are to be read as if the deterministic variant is true *for some sampler*, then the corresponding randomized variant is true *for that sampler*.)

MPC protocol allows two or more parties to jointly evaluate a (possibly randomized) functionality $\mathcal{F}$ on their inputs without revealing anything to each other except what follows from their inputs and outputs. This should hold even in the presence of an adversary who can corrupt *any* number of parties in an adaptive (sequential) fashion. When we write "adaptive MPC", we mean adaptive MPC for *all* randomized functionalities. This should be contrasted with weaker notions of adaptive MPC for strict subsets of corrupted parties [3, 9, 20] or for adaptively well-formed functionalities[5] [10] which can both be done from mild assumptions. The connection from [29] shows that adaptive MPC for all randomized functions is possible if and only if every PPT sampler admits invertible sampling, i.e., the invertible sampling hypothesis is true.

[5] Adaptively well-formed functionalities do not hide internal randomness.

We show that this result generalizes to the global CRS model. More precisely, we prove the adaptive variant of the pseudorandom encoding hypothesis in the CRS model $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ is equivalent to adaptive MPC in the global CRS model.[6]

As detailed above, the static pseudorandom encoding hypothesis $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ in the CRS model follows from $\mathsf{iO}$ (and one-way functions). Applying our static-to-adaptive transformation, the same holds for the adaptive variant. Thus, we obtain the first instantiation of an adaptive explainability compiler [16] without complexity leveraging and, hence, based only on polynomial hardness assumptions. The recent work of Cohen, shelat, and Wichs [15] uses such an adaptive explainability compiler to obtain succinct adaptive MPC, where "succinct" means that the communication complexity is sublinear in the complexity of the evaluated function. Due to our instantiation of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ from polynomial $\mathsf{iO}$, we improve the results of [15] by relaxing the requirement for subexponentially secure $\mathsf{iO}$ to polynomially secure $\mathsf{iO}$ in a black-box way.

**Corollary (informal).** *Assuming the existence of polynomially secure indistinguishability obfuscation and the adaptive hardness of the learning with errors problem, then malicious, two-round, UC-secure adaptive MPC and sublinear communication complexity is possible (in the local CRS model, for all deterministic functionalities).*

**Steganography and Covert Multi-party Computation.** We explore the connection of the pseudorandom encoding hypothesis to various flavors of steganography. The goal of steganography, informally, is to embed secret messages in distributions of natural-looking messages, in order to hide them from external observers. While the standard setting for steganography relies on shared secret keys to encode the messages, we show that pseudorandom encodings naturally give rise to a strong form of *keyless steganography*. Namely, one can rely on pseudorandom encodings to encode any message into an innocent-looking distribution, without truly hiding the message (since anyone can decode the stream), but providing *plausible deniability*, in the sense that, even with the decoded message, it is impossible to tell apart whether this message was indeed encoded by the sender, or whether it is simply the result of decoding the innocent distribution.

**Corollary (informal).** *Assuming pseudorandom encodings, then there exists a keyless steganographic protocol which provides plausible deniability.*

Plausible deniability is an important security notion; in particular, an important cryptographic primitive in this area is the notion of (sender-)deniable encryption [8], which is known to exist assuming indistinguishability obfuscation [35]. Deniable encryption enables to "explain" ciphertexts produced for

[6] Together with the conflict between $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ and EOWFs with unbounded auxiliary input, this corrects a claim made in [16] that the impossibility result of adaptive MPC from [29] would not extend to the CRS model.

some message to any arbitrary other message by providing corresponding random coins for a faked encryption process. We view it as an interesting open problem to build deniable encryption under the pseudorandom encoding hypothesis together with more standard cryptographic primitives; we make a first step in this direction and show the following: the *statistical* variant of pseudorandom encodings, together with the existence of public-key encryption, implies deniable encryption. Interestingly, we also show that the computational randomized pseudorandom encoding hypothesis suffices to imply non-committing encryption, a weaker form of deniable encryption allowing to explain only *simulated* ciphertexts to arbitrary messages [9].

*Covert Secure Computation.* Covert MPC [13, 41] is an intriguing flavor of MPC that aims at achieving the following strong security guarantee: if the output of the protocol is not "favorable," the transcript of the interaction should not leak any information to the parties parties, including whether any given party was *actually taking part in the protocol*. This strong form of MPC aims at providing security guarantees when the very act of starting a computation with other parties should remain hidden. As an example [41], suppose that a CIA agent who infiltrated a terrorist group wants to make a handshake with another individual to find out whether she is also a CIA agent. Here, we show that pseudorandom encodings give rise to a general compiler transforming a standard MPC protocol into a covert one, in a round-preserving way. The idea is to encode each round of the protocol such that encoded messages look random. Together with the equivalence between adaptively secure MPC and pseudorandom encodings, this gives a connection between two seemingly unrelated notions of secure computation.

**Corollary (informal).** *Assuming adaptively secure MPC for all functionalities, there exists a round-preserving compiler that transforms a large class of "natural" MPC protocols into covert MPC protocols (in the static, semi-honest setting).*

**Other Results.** Due to our infeasibility results of $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$, distribution transforming encoders (DTEs) for all efficiently samplable distributions are infeasible. Even the computational relaxation of DTEs is infeasible assuming extractable one-way functions. Since all currently known constructions of honey encryption rely on DTEs, we conditionally refute the existence of honey encryption based on the DTE-then-encrypt framework from [30]. On the positive side, due to our feasibility result of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, computational honey encryption is feasible in the CRS model.

**Theorem (informal).** *Assuming* $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ *and a suitable symmetric-key encryption scheme (modeled as a random cipher), computational honey encryption for all efficiently samplable distributions exists in the CRS model.*

### 1.3 Negative Results for Stronger Notions of Pseudorandom Encodings

Below we describe how we gradually relax optimal compression via different notions of pseudorandom encodings and derive infeasibility results for all variants of pseudorandom encodings which restrict the encoding algorithm to be deterministic or require an information-theoretic pseudorandomness guarantee. This leaves computational randomized pseudorandom encodings as the best possible achievable notion.

**Deterministic, Statistical Pseudorandom Encodings.** The notion of pseudorandom encodings with a deterministic encoding algorithm and information-theoretic indistinguishability is perhaps the simplest notion one can consider. As we will prove in this paper, this notion recovers the notion of optimal compression: since the encoding algorithm for some source $X$ is deterministic, it can be seen with an entropy argument that the output size of $\mathsf{E}_X$ must be at most $\mathrm{H}_\infty(X)$, the min-entropy of $X$; otherwise, the distribution $\{\mathsf{E}_X(X)\}$ can necessarily be distinguished from random with some statistically non-negligible advantage. Therefore, $\mathsf{E}_X$ is an optimal and efficient compression algorithm for $X$, with decompression algorithm $\mathsf{D}_X$; this is true even for the relaxation in the CRS model. The existence of efficient compression algorithms for various categories of samplers was thoroughly studied [40]. In particular, the existence of compression algorithms for all efficiently samplable sources implies the inexistence of one-way functions (this is an observation attributed to Levin in [23]) since compressing the output of a pseudorandom generator to its entropy would distinguish it from a random string, and the existence of one-way functions implies the existence of pseudorandom generators [24]).

**Theorem (informal).** *Assuming the existence of one-way functions, neither* $\mathsf{PREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$ *nor* $\mathsf{cPREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$ *hold.*

This is a strong impossibility result, as one-way functions dwell among the weakest assumptions in cryptography, [28]. One can circumvent this impossibility by studying whether compression can be achieved for more restricted classes of distributions, as was done e.g. in [40]. Our work can be seen as pursuing an orthogonal direction. We seek to determine whether a relaxed notion of compression can be achieved for *all* efficiently samplable distributions. The relaxations we consider comprise the possibility to use randomness in the encoding algorithm, and weakening the requirement on the encoded distribution to being only computationally indistinguishable from random. Clearly, these relaxations remove one of the most important features of compression algorithms, which is that their outputs are smaller than their inputs (i.e., they compress). Nevertheless, the indistinguishability of the encoded distribution from the uniform distribution is another crucial feature of optimal compression algorithms, which has independent applications.

**Deterministic, Computational Pseudorandom Encodings.** We now turn towards a relaxation where the encoded distribution is only required to be computationally indistinguishable from random, but the encoding algorithm is still required to be deterministic. This flavor is strongly connected to an important problem in cryptography: the problem of separating HILL entropy [24] from Yao entropy [44]. HILL and Yao entropy are different approaches of formalizing computational entropy, i.e., the amount of entropy a distribution appears to have from the viewpoint of a computationally bounded entity. Informally, a distribution has high HILL entropy if it is computationally close to a distribution with high min-entropy; a distribution has high Yao entropy if it cannot be compressed efficiently. Finding a distribution which, under standard cryptographic assumptions, has high Yao entropy, but low HILL entropy constitutes a long standing open problem in cryptography. Currently, only an oracle separation [42] and a separation for conditional distributions [27] are known. To establish the connection between $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$ and this problem, we proceed as follows: informally, a deterministic pseudorandom encoding must necessarily *compress its input to the HILL entropy of the distribution.* That is, the output size of the encoding cannot be much larger than the HILL entropy of the distribution. This, in turn, implies that a distribution which admits such a pseudorandom encoding cannot have high Yao entropy.

In this work, we formalize the above argument, and show that the *conditional* separation of HILL and Yao entropy from [27] suffices to refute $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$. This separation holds under the assumption that non-interactive zero-knowledge proofs with some appropriate structural properties exist (which in turn can be based on standard assumptions such as the quadratic residuosity assumption). Thus, we obtain the following infeasibility result:

**Theorem (informal).** *If the quadratic residuosity assumption holds, then* $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$ *does not hold.*

Hence, we may conclude that towards a feasible variant of pseudorandom encodings for all efficiently samplable distributions, requiring the encoding algorithm to be deterministic poses a strong restriction.

**Randomized, Statistical Pseudorandom Encodings.** We now consider the relaxation of perfect compression by allowing the encoding algorithm to be randomized while still requiring information-theoretic indistinguishability from randomness. This flavor of pseudorandom encoding was used in the context of *honey encryption* [30]. Honey encryption is a cryptographic primitive which has been introduced to mitigate attacks on encryption schemes resulting from the use of low-entropy passwords. Honey encryption has the property that decrypting a ciphertext with an incorrect key always yields a valid-looking plaintext which seems to come from the expected distribution, thereby mitigating brute-force attacks. This is the same property that was useful in the previous PAKE example.

The study of honey encryption was initiated in [30], where it was shown that honey encryption can naturally be constructed by composing a block cipher (modeled as a random cipher) with a *distribution transforming encoder* (DTE), a notion which is equivalent to our notion of pseudorandom encoding with randomized encoding and statistical pseudorandomness. The focus of [30] was on obtaining such DTEs for simple and useful distributions. In contrast, we seek to understand the feasibility of this notion for *arbitrary* distributions. Intuitively, it is not straightforward to encode any efficient distribution into the uniform distribution; consider for example the distribution over RSA moduli, i.e., products of two random $n$-bit primes. Since no efficient algorithm is known to test membership in the support of this distribution, natural approaches seem to break down. In fact, we show in this work that this difficulty is inherent: building on techniques from [5,29], we demonstrate the impossibility of (randomized, statistical) pseudorandom encodings for all efficiently samplable distributions, under a relatively standard cryptographic assumption.

**Theorem (informal).** *Assuming the sub-exponential hardness of the learning with errors (LWE) problem,* $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$ *does not hold.*

This result directly implies that under the same assumption, there exist efficiently samplable distributions (with input) for which no distribution transforming encoder exists. We view it as an interesting open problem whether this result can be extended to rule out the existence of honey encryption for arbitrary distributions under the same assumption.

### 1.4 Open Questions and Subsequent Work

The most intriguing question left open by our work is whether the weakest variant of the pseudorandom encoding hypothesis $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, which is implied by $\mathsf{iO}$, also *implies* $\mathsf{iO}$. Very recently, this question was settled in the affirmative by Wee and Wichs [43] under the LWE assumption. More concretely, by modifying a heuristic $\mathsf{iO}$ construction of Brakerski et al. [7], they show that $\mathsf{iO}$ is implied by LWE if one is additionally given an *oblivious LWE-sampler* in the CRS model. Such a sampler, given a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, generates outputs that are indistinguishable from LWE samples $\boldsymbol{A} \cdot \boldsymbol{s} + \boldsymbol{e}$ without knowing the secrets $\boldsymbol{s}$ or the noise $\boldsymbol{e}$. The existence of an oblivious LWE sampler is nontrivial even under the LWE assumption, because $\boldsymbol{A}$ can be such that $\boldsymbol{A} \cdot \boldsymbol{s} + \boldsymbol{e}$ is not pseudorandom; however, such a sampler still follows from the invertible sampling hypothesis [29], which we show to be equivalent to the pseudorandom encoding hypothesis. By proposing an explicit heuristic construction of (a relaxed flavor of) an oblivious LWE sampler, the end result of [43] is a construction of $\mathsf{iO}$ from a new "falsifiable" assumption.

Whether $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ implies $\mathsf{iO}$ under weaker or different assumptions than LWE remains open. A potentially easier goal is using $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ to construct public-key encryption from one-way functions. This is related to the possibility of constructing oblivious transfer from any public-key encryption in which

public keys and ciphertexts are obliviously samplable [19,22], which is implied by public-key encryption and $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$. Here $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is used to bypass the black-box separation between public-key encryption and oblivious transfer [22].

Finally, there is a lot of room for relaxing the intractability assumptions we use to rule out the statistical ($\mathsf{cPREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$) and deterministic ($\mathsf{cPREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$) flavors of pseudorandom encodings.

**Organization.** In Sect. 2, we provide a technical overview of a selection of our results. In Sect. 3, we provide condensed definitions of pseudorandom encodings and invertible sampling and a formal proof of their equivalence and in Sect. 4 we describe the static-to-adaptive transformation. We refer the reader to the full version [1] for more details and for the other results we described.

## 2 Overview of Techniques

In this section, we elaborate on some of our technical results in more detail. In the following, we identify a PPT sampler $S$ with the distribution (family) ensemble it induces.

*The Relation to Invertible Sampling.* A PPT sampler $S$ is *inverse samplable* [17,29], if there exists an alternative sampler $\overline{S}$ inducing a distribution which is computationally indistinguishable to the distribution induced by $S$ such that the computations of $\overline{S}$ can be efficiently inverted. Efficiently inverting the computation of $\overline{S}$ means that there exists an efficient inverse sampler $\overline{S}^{-1}$ which, given an output of $\overline{S}$, recovers a well-distributed random tape for $\overline{S}$ to compute the given output in the following sense. The inverse sampled random tape is required to be computationally indistinguishable from the actually used random tape. More formally, a PPT sampler $S$ is inverse samplable if there exists an efficient alternative sampler $\overline{S}$ and an efficient inverse sampler $\overline{S}^{-1}$ such that

$$\{y \leftarrow S(1^\lambda) \colon y\} \approx_{\mathsf{c}} \{y \leftarrow \overline{S}(1^\lambda) \colon y\}, \tag{3}$$

$$\{y \leftarrow \overline{S}(1^\lambda; r) \colon (r, y)\} \approx_{\mathsf{c}} \{y \leftarrow \overline{S}(1^\lambda) \colon (\overline{S}^{-1}(1^\lambda, y), y)\}. \tag{4}$$

We refer to Eq. (3) as *closeness* and to Eq. (4) as *invertibility*. If the sampler $S$ admits an input $m$, the above is required to hold for all inputs $m$ in the input space $L$, where $\overline{S}$ and $\overline{S}^{-1}$ both additionally receive $m$ as input. In accordance with [29], we refer to the hypothesis that all PPT algorithms with input are inverse samplable as the *invertible sampling hypothesis*. Restricting the invertible sampling hypothesis to algorithms which do not admit inputs is denoted the *weak* invertible sampling hypothesis.

The concepts of inverse samplability and pseudorandom encodings are tightly connected. Suppose a PPT algorithm $S$ is inverse samplable. Then, there exists an alternative and an inverse sampler $(\overline{S}, \overline{S}^{-1})$ satisfying closeness and invertibility. Invertibility guarantees that the inverse sampler $\overline{S}^{-1}$ on input of a sample $y$ from $\overline{S}(1^\lambda)$, outputs a computationally well-distributed random tape $r$. Hence,

with overwhelming probability over the choice of $y \leftarrow \overline{S}(1^\lambda)$ and $r \leftarrow \overline{S}^{-1}(y)$, the alternative sampler on input of $r$, recovers $y$. In other words, the inverse sampler $\overline{S}^{-1}$ can be seen as encoding a given sample $y$, whereas the *de-randomized* alternative sampler $\overline{S}$ given this encoding *as random tape*, is able to recover $y$. Looking through the lens of pseudorandom encoding, this almost proves correctness except that $y$ is sampled according to $\overline{S}(1^\lambda)$ instead of $S(1^\lambda)$. This difference can be bridged due to closeness. We now turn towards showing pseudorandomness of the encoded distribution. Due to closeness, the distributions $\{y \leftarrow \overline{S}(1^\lambda) \colon (\overline{S}^{-1}(1^\lambda, y), y)\}$ and $\{y \leftarrow S(1^\lambda) \colon (\overline{S}^{-1}(1^\lambda, y), y)\}$ are computationally indistinguishable. Invertibility guarantees that, given a sample $y$ from $\overline{S}(1^\lambda)$, an encoding of $y$ is indistinguishable to uniformly chosen randomness conditioned on the fact that decoding yields $y$. Removing $y$ from this distribution, almost corresponds to pseudorandomness, except that $y$ is sampled according to $\overline{S}(1^\lambda)$ instead of $S(1^\lambda)$. Again, we are able to bridge this gap due to closeness. Note that we crucially use the fact that the initial randomness used by $\overline{S}$ resides outside of the view of an adversary. Summing up, if a PPT sampler $S$ is inverse samplable, then it can be pseudorandomly encoded.

Interestingly, this connection turns out to be bidirectional. Suppose a PPT algorithm $S$ can be pseudorandomly encoded. Then, there exists an efficient encoding algorithm $\mathsf{E}_S$ and an efficient deterministic decoding algorithm $\mathsf{D}_S$ satisfying correctness and pseudorandomness. Looking through the lens of invertible sampling, we identify the decoding algorithm to correspond to the alternative sampler (viewing the random tape of the alternative sampler as explicit input to $\mathsf{D}_S$) and the encoding algorithm to correspond to the inverse sampler. Pseudorandomness guarantees that $\mathsf{E}_S(S(1^\lambda))$ is indistinguishable from uniform randomness. Hence, applying the decode algorithm $\mathsf{D}_S$ on uniform randomness is indistinguishable from applying $\mathsf{D}_S$ to outputs of $\mathsf{E}_S(S(1^\lambda))$. Correctness guarantees that $\mathsf{D}_S(\mathsf{E}_S(y))$ for $y$ sampled according to $S(1^\lambda)$ recovers $y$ with overwhelming probability. Thus, the distribution induced by applying $\mathsf{D}_S$ on uniform randomness is computationally close to the distribution induced by $S(1^\lambda)$. This shows closeness. For the purpose of arguing about invertibility, consider the distribution $A := \{y \leftarrow \mathsf{D}_S(r) \colon (r, y)\}$. Due to pseudorandomness $r$ can be considered an encoded sample from $S(1^\lambda)$. Hence, $A$ is indistinguishable to the distribution, where $r$ is produced by $\mathsf{E}_S(y')$ for some *independent* $y' \leftarrow S(1^\lambda)$, i.e.

$$\big\{y \leftarrow \mathsf{D}_S(r) \colon (r, y)\big\} \approx_{\mathsf{c}} \big\{y' \leftarrow S(1^\lambda), r \leftarrow \mathsf{E}_S(y'), y \leftarrow \mathsf{D}_S(r) \colon (r, y)\big\}.$$

Note that by correctness, $y$ and $y'$ are identical with overwhelming probability. Therefore, $A$ is indistinguishable to $\big\{y' \leftarrow S(1^\lambda), r \leftarrow \mathsf{E}_S(y') \colon (r, y')\big\}$. Since sampling $y'$ via $\mathsf{D}_S$ applied on uniform randomness is computationally close to the above distribution due to closeness, invertibility follows. Summing up, a sampler $S$ can be pseudorandomly encoded *if and only if* it is inverse samplable.

Likewise to the variations and relaxations described for pseudorandom encodings, we vary and relax the notion of invertible sampling. The inverse sampler can be required to be deterministic or allowed to be randomized. Further, close-

ness and invertibility can be required to hold information theoretically or computationally. We denote these variants as $\mathsf{ISH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, $\mathsf{ISH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$, $\mathsf{ISH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ and $\mathsf{ISH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$. To circumvent impossibilities in the plain model, we also define the relaxations in the common reference string model in static and adaptive flavors, denoted the prefix "$\mathsf{c}$" and "$\mathsf{ac}$", respectively. The above equivalence extends to all introduced variations of the pseudorandom encoding and invertible sampling hypotheses.

*The Static-to-Adaptive Transformation.* The static variant of pseudorandom encodings in the CRS model only guarantees correctness and pseudorandomness as long as the input $m$ for the sampler $S$ is chosen independently of the CRS. The adaptive variant, on the other hand, provides correctness and pseudorandomness even for adaptive choices of inputs. Adaptive notions always imply their static analogues. Interestingly, for pseudorandom encodings, the opposite direction is true as well. The core idea is to use an *indirection* argument (similar to [11,12,25]) to delay CRS generation until during the actual encoding process. Thus, the advantage stemming from adaptively choosing the input is eliminated.

Suppose that the static variant of the pseudorandom encoding hypothesis in the CRS model is true and let $S$ be some PPT sampler. Since $S$ can be pseudorandomly encoded in the CRS model with static choice of inputs, there exist algorithms $(\mathsf{Setup}', \mathsf{E}', \mathsf{D}')$ such that static correctness and pseudorandomness hold. Further, the algorithm $\mathsf{Setup}'$ can also be pseudorandomly encoded as above. Let $(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}'')$ be the corresponding algorithms such that static correctness and pseudorandomness hold. Note that since the sampler $\mathsf{Setup}'$ does not expect an input, static and adaptive guarantees are equivalent.

Then, the sampler $S$ can be pseudorandomly encoded in the CRS model with *adaptive* choice of inputs as follows. Initially, we sample a common reference string $crs''$ via $\mathsf{Setup}''(1^\lambda)$ and make it available to the parties. Given $crs''$ and a sample $y$ from $S(m)$, adaptive encoding works in two phases. First, a fresh CRS $crs'$ is sampled via $\mathsf{Setup}'(1^\lambda)$ and pseudorandomly encoded via $r_1 \leftarrow \mathsf{E}''(crs'', crs')$. Second, the given sample $y$ is pseudorandomly encoded via $r_2 \leftarrow \mathsf{E}'(crs', m, y)$. The encoding of $y$ then consists of $(r_1, r_2)$. To decode, the CRS $crs'$ is restored via $\mathsf{D}''(crs'', r_1)$. Then, using $crs'$, the original sample $y$ is recovered via $\mathsf{D}'(crs', m, r_2)$.

Since $crs'$ is chosen freshly during the encoding process, the input $m$ which may depend on $crs''$, cannot depend on $crs'$. Further, the distribution $\mathsf{Setup}''$ does not expect an input. Hence, static guarantees suffice.

To realize that adaptive pseudorandomness holds, consider the encoding of $S(m)$ for some adaptively chosen message $m$. Since the view of $\mathcal{A}$ when choosing the message $m$ is independent of $crs'$, static pseudorandomness can be applied to replace the distribution $\mathsf{E}'(crs', m, S(m))$ with uniform randomness. Further, since the sampler $\mathsf{Setup}'$ does not expect any input, static pseudorandomness suffices to replace the distribution $\mathsf{E}''(crs'', \mathsf{Setup}'(1^\lambda))$ with uniform randomness. This proves adaptive pseudorandomness.

The adaptive variant of correctness follows similarly from the static variant of correctness. Consider the distribution of decoding an encoded sample of $S(m)$, where $m$ is adaptively chosen. Since the sampler $\mathsf{Setup}'$ does not expect an input,

static correctness can be applied to replace decoding $\mathsf{D}''(crs'', r_1)$ with the $crs'$ sampled during encoding. Again, since $crs'$ does not lie in the view of the adversary when choosing the message $m$, static correctness guarantees that decoding succeeds with overwhelming probability. This proves adaptive correctness.

*On Deterministic Pseudorandom Encoding and Compression.* The notion of pseudorandom encoding is inspired by the notion of compression. A tuple of deterministic functions $(\mathsf{E}_X, \mathsf{D}_X)$ is said to compress a source $X_\lambda$ to length $m(\lambda)$ with decoding error $\epsilon(\lambda)$, if (i) $\Pr[\mathsf{D}_X(\mathsf{E}_X(X_\lambda)) \neq X_\lambda] \leq \epsilon(\lambda)$ and (ii) $\mathbb{E}[|\mathsf{E}_X(X_\lambda)|] \leq m(\lambda)$, see [40,42]. Pseudorandom encoding partially recovers the notion of compression if we require the encoding algorithm to be deterministic. If a source $X_\lambda$ can be pseudorandomly encoded with a deterministic encoding algorithm having output length $n(\lambda)$, then $X_\lambda$ is compressible to length $n(\lambda)$. Note, however, that the converse direction is not true. Compression and decompression algorithms for a compressible source do not necessarily encode that source pseudorandomly. The output of a compression algorithm is not required to look pseudorandom and, in some cases, admits a specific structure which makes it easily distinguishable from uniform randomness, e.g. instances using Levin search, [40].

Clearly, the requirement for correctness, poses a lower bound on the encoding length $n(\lambda)$, [36]. Conversely, requiring the encoding algorithm $\mathsf{E}_X$ to be deterministic means that the only source of entropy in the distribution $\mathsf{E}_X(X_\lambda)$ originates from the source $X_\lambda$ itself. Hence, for the distributions $\mathsf{E}_X(X_\lambda)$ and the uniform distribution over $\{0,1\}^{n(\lambda)}$ to be indistinguishable, the encoding length $n(\lambda)$ must be "sufficiently small". We observe that correctness together with the fact that $\mathsf{E}_X$ is deterministic implies that the event $\mathsf{E}_X(\mathsf{D}_X(\mathsf{E}_X(X_\lambda))) = \mathsf{E}_X(X_\lambda)$ occurs with overwhelming probability. Applying pseudorandomness yields that $\mathsf{E}_X(\mathsf{D}_X(U_{n(\lambda)})) = U_{n(\lambda)}$ holds with overwhelming probability, wherefore we can conclude that $\mathsf{D}_X$ operates almost injectively on the set $\{0,1\}^{n(\lambda)}$. Hence, the (smooth) min-entropy of $\mathsf{D}_X(U_{n(\lambda)})$ is at least $n(\lambda)$.

Considering information theoretical pseudorandomness, the distributions $\mathsf{D}_X(U_{n(\lambda)})$ and $X_\lambda$ are statistically close. Hence, by the reasoning above, the encoding length $n(\lambda)$ is upper bounded by the (smooth) min-entropy of the source $X_\lambda$. In conclusion, if a distribution can be pseudorandomly encoded such that the encoding algorithm is deterministic satisfying statistical pseudorandomness, then this distribution is compressible to its (smooth) min-entropy. Using a technical "Splitting Lemma", this extends to the relaxed variant of the pseudorandom encoding hypothesis in the CRS model.

Considering computational pseudorandomness, by a similar argument as above, we obtain that $X_\lambda$ is computationally close to a distribution with min-entropy $n(\lambda)$. This does not yield a relation between the encoding length and the min-entropy of the source. However, we do obtain relations to computational analogues of entropy. Computational entropy is the amount of entropy a distribution appears to have from the perspective of a computationally bounded entity. The notion of HILL entropy [24] is defined via the computational indistinguishability from a truly random distribution. More formally, a distribution

$X_\lambda$ has HILL entropy at least $k$, if there exists a distribution with min-entropy $k$ which is computationally indistinguishable from $X_\lambda$. Hence, the encoding length $n(\lambda)$ is upper bounded by the HILL entropy of the source $X_\lambda$. Another important notion of computational entropy is the notion of Yao entropy [44]. Yao entropy is defined via the incompressibility of a distribution. More precisely, a distribution $X_\lambda$ has Yao entropy at least $k$ if $X_\lambda$ cannot be efficiently compressed to length less than $k$ (and successfully decompressed). If a distribution can be pseudorandomly encoded with deterministic encoding, then it can be compressed to the encoding length $n(\lambda)$. This poses an upper bound on the Yao entropy of the source. In summary, this yields

$$n(\lambda) \leq \mathrm{H}^{\mathsf{HILL}}(X_\lambda) \quad \text{and} \quad \mathrm{H}^{\mathsf{Yao}}(X_\lambda) \leq n(\lambda). \tag{5}$$

However, due to [27,31], if the Quadratic Residuosity Assumption (QRA) is true, then there exist distributions which have low *conditional* HILL entropy while being *conditionally* incompressible, i.e. have high conditional Yao entropy.[7] The above observations, particularly Eq. (5), can be extended to conditional HILL and conditional Yao entropy, by considering $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$ for PPT algorithms with input. Therefore, if the Quadratic Residuosity Assumption is true, $\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$ cannot be true for those distributions.

Unfortunately, we do not know whether this extends to the relaxed variants of the pseudorandom encoding hypothesis admitting access to a CRS. On a high level, the problem is that the HILL entropy, in contrast to the min-entropy, does not remain untouched when additionally conditioning on some common reference string distribution, even though the initial distribution is independent of the CRS. Hence, the splitting technique can not be applied here.

## 3 Pseudorandom Encodings and Invertible Sampling

In this section, we formally define pseudorandom encodings and invertible sampling. We will work with the hypothesis that every efficiently samplable distribution can be pseudorandomly encoded and invertible sampled and we refer to these hypotheses as the pseudorandom encoding hypothesis and the invertible sampling hypothesis, respectively. This section is a condensed and much less detailed version of [1].

**Definition 1 (Pseudorandom encoding hypothesis, $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$).** *For every PPT algorithm $S$, there exist efficient algorithms $\mathsf{E}_S$ (the encoding algorithm) with output length $n(\lambda)$ and $\mathsf{D}_S$ (the decoding algorithm), where $\mathsf{D}_S$ is deterministic and $\mathsf{E}_S$ is randomized satisfying the following two properties.*

Correctness. *For all inputs $m \in L$, $\epsilon_{\mathsf{dec-error}}(\lambda) := \Pr\left[y \leftarrow S(m) \colon \mathsf{D}_S(m, \mathsf{E}_S(m, y)) \neq y\right]$ is negligible.*

[7] Let $(X, Z)$ be a joint distribution. The conditional computational entropy is the entropy $X$ appears to have to a bounded adversary when additionally given $Z$.

Pseudorandomness. *For all PPT adversaries $\mathcal{A}$ and all inputs $m \in L$,*

$$Adv^{\mathsf{pre}}_{\mathcal{A},m}(\lambda) := \left|\Pr[Exp^{\mathsf{pre}}_{\mathcal{A},m,0}(\lambda) = 1] - \Pr[Exp^{\mathsf{pre}}_{\mathcal{A},m,1}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

*where $Exp^{\mathsf{pre}}_{\mathcal{A},m,0}$ and $Exp^{\mathsf{pre}}_{\mathcal{A},m,1}$ are defined below.*

| $\underline{Exp^{\mathsf{pre}}_{\mathcal{A},m,0}(\lambda)}$ | $\underline{Exp^{\mathsf{pre}}_{\mathcal{A},m,1}(\lambda)}$ |
|---|---|
| $r \leftarrow \{0,1\}^{p(\lambda)}$ | $u \leftarrow \{0,1\}^{n(\lambda)}$ |
| $y := S(m;r)$ | **return** $\mathcal{A}(m,u)$ |
| **return** $\mathcal{A}(m, \mathsf{E}_S(m,y))$ | |

**Definition 2 (Invertible sampling hypothesis, $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$, [29]).** *For every PPT algorithm $S$, there exists a PPT algorithm $\overline{S}$ (the alternate sampler) with randomness space $\{0,1\}^{n(\lambda)}$ and an efficient randomized algorithm $\overline{S}^{-1}$ (the inverse sampler), satisfying the following two properties.*

Closeness. *For all PPT adversaries $\mathcal{A}$ and all inputs $m \in L$,*

$$Adv^{\mathsf{close}}_{\mathcal{A},m}(\lambda) := \left|\Pr[Exp^{\mathsf{close}}_{\mathcal{A},m,0}(\lambda) = 1] - \Pr[Exp^{\mathsf{close}}_{\mathcal{A},m,1}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

*where $Exp^{\mathsf{close}}_{\mathcal{A},m,0}$ and $Exp^{\mathsf{close}}_{\mathcal{A},m,1}$ are defined below.*

Invertibility. *For all PPT adversaries $\mathcal{A}$ and all inputs $m \in L$,*

$$Adv^{\mathsf{inv}}_{\mathcal{A},m}(\lambda) := \left|\Pr[Exp^{\mathsf{inv}}_{\mathcal{A},m,0}(\lambda) = 1] - \Pr[Exp^{\mathsf{inv}}_{\mathcal{A},m,1}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

*where $Exp^{\mathsf{inv}}_{\mathcal{A},m,0}$ and $Exp^{\mathsf{inv}}_{\mathcal{A},m,1}$ are defined below.*

| $\underline{Exp^{\mathsf{close}}_{\mathcal{A},m,0}(\lambda)}$ | $\underline{Exp^{\mathsf{close}}_{\mathcal{A},m,1}(\lambda)}$ | $\underline{Exp^{\mathsf{inv}}_{\mathcal{A},m,0}(\lambda)}$ | $\underline{Exp^{\mathsf{inv}}_{\mathcal{A},m,1}(\lambda)}$ |
|---|---|---|---|
| $r \leftarrow \{0,1\}^{p(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ |
| $y := S(m;r)$ | $y := \overline{S}(m;r)$ | $y := \overline{S}(m;r)$ | $y := \overline{S}(m;r)$ |
| **return** $\mathcal{A}(m,y)$ | **return** $\mathcal{A}(m,y)$ | **return** $\mathcal{A}(m,r,y)$ | $\overline{r} \leftarrow \overline{S}^{-1}(m,y)$ |
| | | | **return** $\mathcal{A}(m,\overline{r},y)$ |

**Theorem 1.** $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ *is true if and only if* $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$ *is true.*

**Lemma 1.** *If* $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$ *holds, then* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ *holds.*

*Proof.* Assume $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$ holds. Let $S$ be a PPT algorithm. $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$ implies that there exists an alternative sampler $\overline{S}$ (with randomness space $\{0,1\}^{n(\lambda)}$) and a corresponding inverse sampler $\overline{S}^{-1}$ satisfying closeness and invertibility.

For $m \in L, y \in \{0,1\}^*, r \in \{0,1\}^{n(\lambda)}$, we define the algorithms $\mathsf{E}_S(m,y) := \overline{S}^{-1}(m,y)$ (potentially randomized) and $\mathsf{D}_S(m,r) := \overline{S}(m;r)$ (deterministic).

| $\mathbf{G}_0$ | $\mathbf{G}_1$ | $\mathbf{G}_2$ |
|---|---|---|
| $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r \leftarrow \{0,1\}^{p(\lambda)}$ |
| $y := \overline{S}(m;r)$ | $y := \overline{S}(m;r)$ | $y := S(m;r)$ |
| **return** $\mathcal{A}(m,r,y)$ | $\overline{r} \leftarrow \overline{S}^{-1}(m,y)$ | $\overline{r} \leftarrow \overline{S}^{-1}(m,y)$ |
| | **return** $\mathcal{A}(m,\overline{r},y)$ | **return** $\mathcal{A}(m,\overline{r},y)$ |

**Fig. 2.** Hybrids used in the proof of correctness.

**Correctness.** We consider a series of hybrids, see Fig. 2.

Game $\mathbf{G}_0$ is identical to $Exp^{\mathsf{inv}}_{\mathcal{A},m,0}$ and game $\mathbf{G}_1$ is identical to $Exp^{\mathsf{inv}}_{\mathcal{A},m,1}$. Hence, $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{inv}}_{\mathcal{A},m}(\lambda)$.

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq Adv^{\mathsf{close}}_{\overline{\mathcal{A}},m}(\lambda)$.

*Proof.* Construct an adversary $\overline{\mathcal{A}}$ on closeness. On input of $(m,y)$, $\overline{\mathcal{A}}$ computes $\overline{r} \leftarrow \overline{S}^{-1}(m,y)$, calls $\mathcal{A}$ on input of $(m,\overline{r},y)$ and outputs the resulting output. If $y$ is sampled using $\overline{S}(m;r)$ (for $r \leftarrow \{0,1\}^{n(\lambda)}$), $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_1$ for $\mathcal{A}$. If $y$ is sampled using $S(m;r)$ (for $f \leftarrow \{0,1\}^{p(\lambda)}$), $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_2$ for $\mathcal{A}$. Therefore, $\Pr[out_1 = 1] = \Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},m,1}(\lambda) = 1]$ and $\Pr[out_2 = 1] = \Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$. □

Thus, we have that $|\Pr[out_2 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{close}}_{\overline{\mathcal{A}},m}(\lambda) + Adv^{\mathsf{inv}}_{\overline{\mathcal{A}}',m}(\lambda)$ for some PPT adversaries $\overline{\mathcal{A}}, \overline{\mathcal{A}}'$.

Consider the adversary $\mathcal{A}$ distinguishing between game $\mathbf{G}_0$ and game $\mathbf{G}_2$ that on input of $(m,r,y)$, outputs 0 if $\overline{S}(m;r) = y$ and outputs 1 otherwise. By definition, $\mathcal{A}$ always outputs 0 in $\mathbf{G}_0$. Hence, $\epsilon_{\mathsf{dec-error}}(\lambda) = \Pr[y \leftarrow S(m) \colon \overline{S}(m, \overline{S}^{-1}(m,y)) \neq y] = \Pr[out_{2,\mathcal{A}} = 1] = |\Pr[out_{2,\mathcal{A}} = 1] - \Pr[out_{0,\mathcal{A}} = 1]|$.

**Pseudorandomness.** We consider a sequence of hybrids starting from $Exp^{\mathsf{pre}}_{\mathcal{A},m,0}$ and concluding in $Exp^{\mathsf{pre}}_{\mathcal{A},m,1}$, see Fig. 3.

| $\mathbf{G}_0$ | $\mathbf{G}_1$ | $\mathbf{G}_2$ |
|---|---|---|
| $r \leftarrow \{0,1\}^{p(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r \leftarrow \{0,1\}^{n(\lambda)}$ |
| $y := S(m;r)$ | $y := \overline{S}(m;r)$ | **return** $\mathcal{A}(m,r)$ |
| $u \leftarrow \overline{S}^{-1}(m,y)$ | $u \leftarrow \overline{S}^{-1}(m,y)$ | |
| **return** $\mathcal{A}(m,u)$ | **return** $\mathcal{A}(m,u)$ | |

**Fig. 3.** Hybrids used in the proof of pseudorandomness.

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{close}}_{\overline{\mathcal{A}},m}(\lambda)$.

*Proof.* Construct a PPT adversary $\overline{\mathcal{A}}$ on the closeness property as follows. On input of $(m, y)$, $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input of $(m, \overline{S}^{-1}(m, y))$ and outputs the resulting output.

If $y \leftarrow S(m)$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_0$ for $\mathcal{A}$, and if $y \leftarrow \overline{S}(m)$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_1$ for $\mathcal{A}$. Hence, $\Pr[out_0 = 1] = \Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$ and $\Pr[out_1 = 1] = \Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},m,1}(\lambda) = 1]$. □

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq Adv^{\mathsf{inv}}_{\overline{\mathcal{A}},m}(\lambda)$.

*Proof.* We construct a PPT adversary $\overline{\mathcal{A}}$ on the invertibility property. On input of $(m, r, y)$, $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input of $(m, r)$ and outputs its output.

If $r \leftarrow \overline{S}^{-1}(m, y)$ for $y \leftarrow \overline{S}(m)$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_1$ for $\mathcal{A}$. If $r \leftarrow \{0,1\}^{n(\lambda)}$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_2$ for $\mathcal{A}$. Therefore, $\Pr[out_1 = 1] = \Pr[Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$ and $\Pr[out_2 = 1] = \Pr[Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},m,1}(\lambda) = 1]$. □

Hence, $Adv^{\mathsf{pre}}_{\mathcal{A},m}(\lambda) = |\Pr[out_2 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{close}}_{\overline{\mathcal{A}},m}(\lambda) + Adv^{\mathsf{inv}}_{\overline{\mathcal{A}}',m}(\lambda)$ for some PPT adversaries $\overline{\mathcal{A}}$ and $\overline{\mathcal{A}}'$. □

**Lemma 2.** *If* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ *holds, then* $\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$ *holds.*

*Proof.* We prove the statement for the computational randomized case. The remaining cases are similar.

Assume $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ holds. Let $S$ be a PPT algorithm. $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$implies that for $S$ there exist efficient algorithms $\mathsf{E}_S$ (potentially randomized) with output length $n(\lambda)$ and $\mathsf{D}_S$ (deterministic) satisfying correctness and pseudorandomness.

For $m \in L, r \in \{0,1\}^{n(\lambda)}, y \in \{0,1\}^*$, we define the alternative sampler as $\overline{S}(m; r) := \mathsf{D}_S(m, r)$ (randomized) and the corresponding inverse sampler $\overline{S}^{-1}(m, y) := \mathsf{E}_S(m, y)$ (potentially randomized).

**Closeness.** Let $\mathcal{A}$ be an adversary on closeness. We consider a sequence of games starting from $Exp^{\mathsf{close}}_{\mathcal{A},m,0}$ and concluding in $Exp^{\mathsf{close}}_{\mathcal{A},m,1}$, see Fig. 4.

| $\mathbf{G}_0$ | $\mathbf{G}_1$ | $\mathbf{G}_2$ | $\mathbf{G}_3$ |
|---|---|---|---|
| $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ |
| $y_S := S(m; r_S)$ | $y_S := S(m; r_S)$ | $y_S := S(m; r_S)$ | $y_S := S(m; r_S)$ |
| **return** $\mathcal{A}(m, y_S)$ | $r_D \leftarrow \mathsf{E}_S(m, y_S)$ | $r_D \leftarrow \mathsf{E}_S(m, y_S)$ | $r_D \leftarrow \{0,1\}^{n(\lambda)}$ |
| | $y_D := \mathsf{D}_S(m, r_D)$ | $y_D := \mathsf{D}_S(m, r_D)$ | $y_D := \mathsf{D}_S(m, r_D)$ |
| | **return** $\mathcal{A}(m, y_S)$ | **return** $\mathcal{A}(m, y_D)$ | **return** $\mathcal{A}(m, y_D)$ |

**Fig. 4.** Hybrids used in the proof of closeness.

The difference between game $\mathbf{G}_0$ and game $\mathbf{G}_1$ is only conceptional, hence, $\Pr[out_0 = 1] = \Pr[out_1 = 1]$.

$\mathbf{G}_1$ and $\mathbf{G}_2$ proceed exactly identical if $y_S = y_D$. More formally, let $F$ be the event that $y_S \neq y_D$. We have that $out_1 = 1 \wedge \neg F \Leftrightarrow out_2 \wedge \neg F$. Hence, the Difference Lemma (due to Shoup, [37]) bounds $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq \Pr[F]$. Correctness guarantees that for all $m \in L$, $\Pr[F] = \Pr[y_S \leftarrow S(m) \colon \mathsf{D}_S(m, \mathsf{E}_S(m, y_S)) \neq y_S] = \epsilon_{\mathsf{dec-error}}(\lambda)$ is negligible.

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda)$.

*Proof.* Construct an adversary $\overline{\mathcal{A}}$ on pseudorandomness as follows. On input of $(m, u =: r_D)$, $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input $(m, \mathsf{D}_S(m, r_D))$ and outputs the resulting output. If $u \leftarrow \mathsf{E}_S(m, y)$ for $y \leftarrow S(m)$, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_2$ for $\mathcal{A}$. Otherwise, if $u$ is uniformly random over $\{0,1\}^{n(\lambda)}$, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_3$ for $\mathcal{A}$. Hence, $\Pr[out_3 = 1] = \Pr[Exp^{\mathsf{pre}}_{\overline{\mathcal{A}},m,1}(\lambda) = 1]$ and $\Pr[out_2 = 1] = \Pr[Exp^{\mathsf{pre}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$. □

Hence, $Adv^{\mathsf{close}}_{\mathcal{A},m}(\lambda) = |\Pr[out_3 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda) + \epsilon_{\mathsf{dec-error}}(\lambda)$ for some PPT adversary $\overline{\mathcal{A}}$.

**Invertibility.** We consider a sequence of hybrids, see Fig. 5.

| $\mathbf{G}_0$ | $\mathbf{G}_1$ | $\mathbf{G}_2$ |
|---|---|---|
| $r \leftarrow \{0,1\}^{n(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ |
| $y := \mathsf{D}_S(m, r)$ | $y_S := S(m; r_S)$ | $y_S := S(m; r_S)$ |
| $\overline{r} \leftarrow \mathsf{E}_S(m, y)$ | $r_D \leftarrow \mathsf{E}_S(m, y_S)$ | $r_D \leftarrow \mathsf{E}_S(m, y_S)$ |
| **return** $\mathcal{A}(m, \overline{r}, y)$ | **return** $\mathcal{A}(m, r_D, y_S)$ | $y_D := \mathsf{D}_S(m, r_D)$ |
| | | **return** $\mathcal{A}(m, r_D, y_S)$ |

| $\mathbf{G}_3$ | $\mathbf{G}_4$ | $\mathbf{G}_5$ |
|---|---|---|
| $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_S \leftarrow \{0,1\}^{p(\lambda)}$ | $r_D \leftarrow \{0,1\}^{n(\lambda)}$ |
| $y_S := S(m; r_S)$ | $y_S := S(m; r_S)$ | $y_D := \mathsf{D}_S(m, r)$ |
| $r_D \leftarrow \mathsf{E}_S(m, y_S)$ | $r_D \leftarrow \{0,1\}^{n(\lambda)}$ | **return** $\mathcal{A}(m, r_D, y_D)$ |
| $y_D := \mathsf{D}_S(m, r_D)$ | $y_D := \mathsf{D}_S(m, r_D)$ | |
| **return** $\mathcal{A}(m, r_D, y_D)$ | **return** $\mathcal{A}(m, r_D, y_D)$ | |

**Fig. 5.** Hybrids used in the proof of invertibility.

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda) + \epsilon_{\mathsf{dec-error}}(\lambda)$.

*Proof.* Let $\mathcal{A}$ be an adversary distinguishing $\mathbf{G}_0$ and $\mathbf{G}_1$. Construct an adversary $\overline{\mathcal{A}}$ on the closeness property. On input of $(m, y)$, $\overline{\mathcal{A}}$ computes $\overline{r} \leftarrow \mathsf{E}_S(m, y)$ and calls $\mathcal{A}$ on input $(m, \overline{r}, y)$. If $y \leftarrow \overline{S}(m)$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_0$ for $\mathcal{A}$. Else, if $y \leftarrow S(m)$, $\overline{\mathcal{A}}$ simulates game $\mathbf{G}_1$ for $\mathcal{A}$. Hence, $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| = Adv^{\mathsf{close}}_{\overline{\mathcal{A}},m}(\lambda)$. □

The difference between $\mathbf{G}_1$ and $\mathbf{G}_2$ is purely conceptional. Hence, $\Pr[out_1 = 1] = \Pr[out_2 = 1]$. $\mathbf{G}_2$ and $\mathbf{G}_3$ behave identical if $y_D = y_S$. Let $F$ denote the failure event $y_D \neq y_S$. We have that $out_2 = 1 \wedge \neg \Leftrightarrow out_3 \wedge \neg F$. The Difference Lemma (due to Shoup, [37]) bounds $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq \Pr[F]$. Due to correctness, for all $m \in L$, $\Pr[F] = \Pr[y_S \leftarrow S(m) \colon \mathsf{D}_S(m, \mathsf{E}_S(m, y_S)) \neq y_S] = \epsilon_{\mathsf{dec-error}}(\lambda)$ is negligible.

*Claim.* For all PPT adversaries $\mathcal{A}$, for all $m \in L$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_4 = 1] - \Pr[out_3 = 1]| \leq Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda)$.

*Proof.* Construct a PPT adversary $\overline{\mathcal{A}}$ on the pseudorandomness property. On input of $(m, u)$, $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input $(m, u =: r_D, \mathsf{D}_S(m, u) =: y_D)$ and outputs the resulting output. If $u \leftarrow \mathsf{E}_S(m, y)$ for $y \leftarrow S(m)$, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_3$ for $\mathcal{A}$. Otherwise, if $u$ is uniformly random over $\{0,1\}^{n(\lambda)}$, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_4$ for $\mathcal{A}$. Hence, $\Pr[out_3 = 1] = \Pr[Exp^{\mathsf{pre}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$ and $\Pr[out_4 = 1] = \Pr[Exp^{\mathsf{pre}}_{\overline{\mathcal{A}},m,1}(\lambda) = 1]$. □

The difference between $\mathbf{G}_4$ and $\mathbf{G}_5$ is again only conceptional and $\Pr[out_4 = 1] = \Pr[out_5 = 1]$. Hence, $|\Pr[out_5 = 1] - \Pr[out_0 = 1]| \leq 2 \cdot Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda) + 2 \cdot \epsilon_{\mathsf{dec-error}}(\lambda)$ for some PPT adversary $\overline{\mathcal{A}}$. □

## 4 Static-to-Adaptive Transformation

We obtain a natural relaxation of the pseudorandom encoding hypothesis by introducing public parameters. That is, a distribution defined via $S$ can be pseudorandomly encoded in this relaxed sense, if there exists a probabilistic setup algorithm $\mathsf{Setup}_S$ and encode and decode algorithms as before such that for all $m \in L$, the event $\mathsf{D}_S(crs, \mathsf{E}_S(crs, S(m))) = S(m)$ occurs with overwhelming probability, where the probability is also over the choice of $crs$, and the distribution $(\mathsf{Setup}_S(1^\lambda), \mathsf{E}_S(\mathsf{Setup}_S(1^\lambda), S(m)))$ is indistinguishable from the distribution $(\mathsf{Setup}_S(1^\lambda), U_{n(\lambda)})$. See the full version [1] for more details.

There are two variants of this definition. The input $m$ can be required to be chosen independently of $crs$ or allowed to be chosen depending on $crs$. Clearly, the adaptive variant implies the non-adaptive (or static) variant. Interestingly, the opposite direction is true as well by an "indirection" argument similar to the one from the work on universal samplers [25]. A similar technique was used in the context of non-committing encryption [11] and adaptively secure MPC [12].

**Theorem 2.** *Let* $\alpha \in \{\approx_{\mathsf{c}}, \equiv_{\mathsf{s}}\}$ *and* $\beta \in \{\mathsf{rand}, \mathsf{det}\}$. *If* $\mathsf{cPREH}^{\beta}_{\alpha}$ *is true, then* $\mathsf{acPREH}^{\beta}_{\alpha}$ *is true.*

*Proof.* We prove the statement for the computational randomized case. A very similar proof applies to the remaining cases.

Let $S$ be a PPT sampler with input space $L$. Since $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ is true, for the PPT sampler $S$, there exist $(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S)$ with output length $n'(\lambda)$ such that correctness and pseudorandomness hold (statically). Again, since $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ is true, for the PPT sampler $\mathsf{Setup}'_S$, there exist $(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}'')$ with output length $n''(\lambda)$ such that correctness and pseudorandomness hold (statically). [8] Note that $\mathsf{Setup}'_S$ does not expect an input.

In Fig. 6, we define algorithms $(\mathsf{Setup}_S, \mathsf{E}_S, \mathsf{D}_S)$ satisfying *adaptive* correctness and pseudorandomness.

| $\mathsf{Setup}_S(1^\lambda)$ | $\mathsf{E}_S(crs, m, y)$ | $\mathsf{D}_S(crs, m, r)$ |
|---|---|---|
| $crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$ | $crs' \leftarrow \mathsf{Setup}'_S(1^\lambda)$ | **parse** $r =: r_1 \,\|\|\, r_2$ |
| $crs := crs''$ | $r_1 \leftarrow \mathsf{E}''(crs'', crs')$ | $crs' := \mathsf{D}''(crs'', r_1)$ |
| **return** $crs$ | $r_2 \leftarrow \mathsf{E}'_S(crs', m, y)$ | $y := \mathsf{D}'_S(crs', m, r_2)$ |
| | **return** $r_1 \,\|\|\, r_2$ | **return** $y$ |

**Fig. 6.** Adaptive pseudorandom encodings.

On a high level, since $crs'$ is chosen freshly and independently after the adversary fixes the message $m$, selective security suffices. Furthermore, since the distribution of $crs'$ has no input, selective security is sufficient.

**Adaptive correctness.** We define a series of hybrid games to prove correctness, see Fig. 7. Game $\mathbf{G}_0$ corresponds to encoding and subsequently decoding a sample $y$ (for adaptively chosen input $m$) and game $\mathbf{G}_1$ is simply a reordering

| $\mathbf{G}_1$ | $\mathbf{G}_2$ | $\mathbf{G}_3$ |
|---|---|---|
| $crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$ | $crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$ | $crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$ |
| $crs' \leftarrow \mathsf{Setup}'_S(1^\lambda)$ | $crs' \leftarrow \mathsf{Setup}'_S(1^\lambda)$ | $m \leftarrow \mathcal{A}(crs'')$ |
| $r_1 \leftarrow \mathsf{E}''(crs'', crs')$ | $r_1 \leftarrow \mathsf{E}''(crs'', crs')$ | $crs' \leftarrow \mathsf{Setup}'_S(1^\lambda)$ |
| $crs'_D := \mathsf{D}''(crs'', r_1)$ | $crs'_D := \mathsf{D}''(crs'', r_1)$ | $y \leftarrow S(m)$ |
| $m \leftarrow \mathcal{A}(crs'')$ | $m \leftarrow \mathcal{A}(crs'')$ | $r_2 \leftarrow \mathsf{E}'_S(crs', m, y)$ |
| $y \leftarrow S(m)$ | $y \leftarrow S(m)$ | $y_D := \mathsf{D}'_S(crs', m, r_2)$ |
| $r_2 \leftarrow \mathsf{E}'_S(crs', m, y)$ | $r_2 \leftarrow \mathsf{E}'_S(crs', m, y)$ | **return** $y_D = y$ |
| $y_D := \mathsf{D}'_S(crs'_D, m, r_2)$ | $y_D := \mathsf{D}'_S(crs', m, r_2)$ | |
| **return** $y_D = y$ | **return** $y_D = y$ | |

**Fig. 7.** Hybrid games for the proof of adaptive correctness.

[8] For notational convenience, we do not write the sampler $\mathsf{Setup}'_S$ as index.

of the commands of $\mathbf{G}_0$. The game hop from $\mathbf{G}_0$ to $\mathbf{G}_1$ only conceptional and $\Pr[out_0 = 1] = \Pr[out_1 = 1]$.

*Claim.* For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq \epsilon^{\mathsf{c-dec-error}}_{(\mathsf{Setup}'',\mathsf{E}'',\mathsf{D}''),\overline{\mathcal{A}}}(\lambda)$.

*Proof.* The games $\mathbf{G}_1$ and $\mathbf{G}_2$ proceed exactly identically if $crs'_D = crs'$. Let $E$ be the event that $crs' \neq crs'_D$. We have that $out_1 = 1 \wedge \neg E \Leftrightarrow out_2 \wedge \neg E$. Due to correctness of $(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}'')$,

$$\Pr\left[\begin{array}{rcl} crs'' & \leftarrow & \mathsf{Setup}(1^\lambda) \\ crs' & \leftarrow & \mathsf{Setup}'_S(1^\lambda) \\ r_1 & \leftarrow & \mathsf{E}''(crs'', crs') \\ crs'_D & := & \mathsf{D}''(crs'', r_1) \end{array} : crs'_D \neq crs'\right]$$

is negligible. Hence, the Difference Lemma (due to Shoup, [37]) upper bounds

$$|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq \Pr[E].$$

□

The game hop from $\mathbf{G}_2$ to $\mathbf{G}_3$ only conceptional and $\Pr[out_2 = 1] = \Pr[out_3 = 1]$.

*Claim.* For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $\Pr[out_3 = 1] \geq 1 - \epsilon^{\mathsf{c-dec-error}}_{(\mathsf{Setup}'_S,\mathsf{E}'_S,\mathsf{D}'_S),\overline{\mathcal{A}}}(\lambda)$.

*Proof.* Due to correctness of $(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S)$, we have that for all PPT adversaries $\overline{\mathcal{A}}$,

$$\Pr\left[\begin{array}{rcl} m & \leftarrow & \overline{\mathcal{A}}(1^\lambda) \\ crs' & \leftarrow & \mathsf{Setup}'_S(1^\lambda) \\ y & \leftarrow & S(m) \\ r & \leftarrow & \mathsf{E}'_S(crs', m, y) \\ y_D & := & \mathsf{D}'_S(crs', m, r) \end{array} : y = y_D\right]$$

is overwhelming. Therefore, for all PPT adversaries $\mathcal{A}$, $\Pr[out_3 = 1]$ is overwhelming. □

**Adaptive Pseudorandomness.** We define a series of hybrid games to prove pseudorandomness, see Fig. 8.

Game $\mathbf{G}_0$ corresponds to the adaptive pseudorandomness game. That is, $\mathbf{G}_0$ first samples $crs''$, the adversary $\mathcal{A}$ chooses the message $m$ adaptively depending on $crs''$, and $\mathbf{G}_0$ then samples $y$ using $S(m)$, encodes that sample and gives the encoding to $\mathcal{A}$.

*Claim.* For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{crs-pre}}_{(\mathsf{Setup}'_S,\mathsf{E}'_S,\mathsf{D}'_S),\overline{\mathcal{A}}}(\lambda)$.

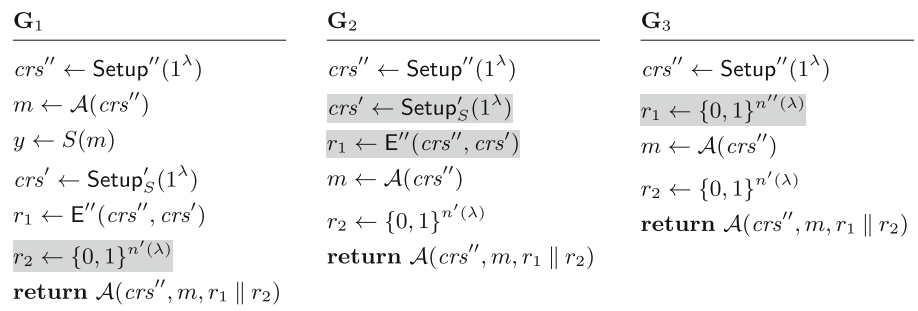


**Fig. 8.** Hybrid games for the proof of adaptive pseudorandomness.

*Proof.* Construct an adversary $\overline{\mathcal{A}}$ on *static* pseudorandomness relative to $(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S)$ as follows. On input of $1^\lambda$, $\overline{\mathcal{A}}$ samples $crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$ calls $\mathcal{A}$ on input of $crs''$, and outputs the message $m$ produced by $\mathcal{A}$. In return, $\overline{\mathcal{A}}$ receives $crs' \leftarrow \mathsf{Setup}'_S(1^\lambda)$ and either $u := \mathsf{E}'_S(crs', m, S(m))$ or a uniform random string $u \leftarrow \{0,1\}^{n'(\lambda)}$ from $Exp^{\mathsf{crs-pre}}_{(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S), \mathcal{A}, b}(\lambda)$. $\overline{A}$ computes $r_1 \leftarrow \mathsf{E}''(crs'', crs')$, calls $\mathcal{A}$ on input of $(crs'', m, r_1 \| u)$ and returns $\mathcal{A}$'s output.

If $\overline{\mathcal{A}}$ plays $Exp^{\mathsf{crs-pre}}_{(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S), \overline{\mathcal{A}}, 0}(\lambda)$, then it perfectly simulates $\mathbf{G}_0$. On the other hand, if $\overline{\mathcal{A}}$ plays $Exp^{\mathsf{crs-pre}}_{(\mathsf{Setup}'_S, \mathsf{E}'_S, \mathsf{D}'_S), \overline{\mathcal{A}}, 1}(\lambda)$, then it perfectly simulates $\mathbf{G}_1$.□

The game hop from $\mathbf{G}_1$ to $\mathbf{G}_2$ is only conceptional and $\Pr[out_2 = 1] = \Pr[out_1 = 1]$.

*Claim.* For all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq Adv^{\mathsf{crs-pre}}_{(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}''), \overline{\mathcal{A}}}(\lambda)$.

*Proof.* Construct an adversary $\overline{\mathcal{A}}$ on *static* pseudorandomness relative to $(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}'')$ as follows. On input of $1^\lambda$, $\overline{\mathcal{A}}$ returns $\perp$ since the input space $L$ of the sampler $\mathsf{Setup}'_S(1^\lambda)$ is empty. In return, $\overline{\mathcal{A}}$ receives $crs''$ sampled via $\mathsf{Setup}''(1^\lambda)$ and $u$ which is either produced via $\mathsf{E}''(crs'', \mathsf{Setup}'(1^\lambda))$ or via uniform sampling from $\{0,1\}^{n''(\lambda)}$. $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input of $crs''$ and receives a message $m$ from $\mathcal{A}$. Finally, $\overline{\mathcal{A}}$ samples $r_2 \leftarrow \{0,1\}^{n'(\lambda)}$, calls $\mathcal{A}$ on input of $(crs'', m, u \| r_2)$ and outputs his output.

If $\overline{\mathcal{A}}$ plays $Exp^{\mathsf{crs-pre}}_{(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}''), \overline{\mathcal{A}}, 0}(\lambda)$, then it perfectly simulates $\mathbf{G}_2$. On the other hand, if $\overline{\mathcal{A}}$ plays $Exp^{\mathsf{crs-pre}}_{(\mathsf{Setup}'', \mathsf{E}'', \mathsf{D}''), \overline{\mathcal{A}}, 1}(\lambda)$, then it perfectly simulates $\mathbf{G}_3$. □

**Acknowledgments.** We thank Daniel Wichs for suggesting the unconditional static-to-adaptive transformation, as well as anonymous reviewers for extremely useful feedback.

## References


1. Agrikola, T., Couteau, G., Ishai, Y., Jarecki, S., Sahai, A.: On pseudorandom encodings. Cryptology ePrint Archive, report 2020/445 (2020). https://eprint.iacr.org/2020/445
2. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_1
3. Beaver, D., Haber, S.: Cryptographic Protocols Provably Secure Against Dynamic Adversaries. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 307–323. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-47555-9_26
4. Bellovin, S.M., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Computer Society Press, (1992). doi: 10.1109/RISP.1992.213269
5. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Shmoys, D.B., ed., 46th ACM STOC, pp. 505–514. ACM Press, May/June 2014. doi: 10.1145/2591796.2591859
6. Boyd, C., Montague, P., Nguyen, K.: Elliptic Curve Based Password Authenticated Key Exchange Protocols. In: Varadharajan, V., Mu, Y. (eds.) ACISP 2001. LNCS, vol. 2119, pp. 487–501. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-47719-5_38
7. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Candidate iO from Homomorphic Encryption Schemes. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12105, pp. 79–109. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45721-1_4
8. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable Encryption. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052229
9. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th ACM STOC, pp. 639–648. ACM Press, May 1996. doi: https://doi.org/10.1145/237814.238015
10. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: 34th ACM STOC, pp. 494–503. ACM Press, May 2002. doi: 10.1145/509907.509980
11. Canetti, R., Poburinnaya, O., Raykova, M.: Optimal-Rate Non-Committing Encryption. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10626, pp. 212–241. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_8
12. Canetti, R., Poburinnaya, O., Venkitasubramaniam, M.: Better Two-Round Adaptive Multi-party Computation. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10175, pp. 396–427. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54388-7_14
13. Chandran, N., Goyal, V., Ostrovsky, R., Sahai, A.: Covert multi-party computation. In: 48th FOCS, pp. 238–248. IEEE Computer Society Press, October 2007. doi: 10.1109/FOCS.2007.21
14. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Improved Non-committing Encryption with Applications to Adaptively Secure Protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 287–302. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_17

15. Cohen, R., Shelat, A., Wichs, D.: Adaptively Secure MPC with Sublinear Communication Complexity. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11693, pp. 30–60. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26951-7_2
16. Dachman-Soled, D., Katz, J., Rao, V.: Adaptively Secure, Universally Composable, Multiparty Computation in Constant Rounds. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 586–613. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_23
17. Damgård, I., Nielsen, J.B.: Improved Non-committing Encryption Schemes Based on a General Complexity Assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_27
18. Dodis, Y., Ristenpart, T., Vadhan, S.: Randomness Condensers for Efficiently Samplable, Seed-Dependent Sources. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 618–635. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_35
19. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Commun. ACM **28**(6), 637–647 (1985). https://doi.org/10.1145/3812.3818
20. Garg, S., Sahai, A.: Adaptively Secure Multi-Party Computation with Dishonest Majority. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 105–123. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_8
21. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press, October 2013. doi: 10.1109/FOCS.2013.13
22. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. FOCS **2000**, 325–335 (2000). https://doi.org/10.1109/SFCS.2000.892121
23. Goldberg, A.V., Sipser, M.: Compression and ranking. In: 17th ACM STOC, pp. 440–448. ACM Press, May 1985. doi: 10.1145/22145.22194
24. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999). https://doi.org/10.1137/S0097539793244708
25. Hofheinz, D., Jager, T., Khurana, D., Sahai, A., Waters, B., Zhandry, M.: How to Generate and Use Universal Samplers. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 715–744. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_24
26. Wei, F., Li, Y., Roy, S., Ou, X., Zhou, W.: Deep Ground Truth Analysis of Current Android Malware. In: Polychronakis, M., Meier, M. (eds.) DIMVA 2017. LNCS, vol. 10327, pp. 252–276. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60876-1_12
27. Hsiao, C.-Y., Lu, C.-J., Reyzin, L.: Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 169–186. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_10
28. Impagliazzo, R.: A personal view of average-case complexity. In: Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, 19–22 June 1995, pp. 134–147 (1995). https://doi.org/10.1109/SCT.1995.514853

29. Ishai, Y., Kumarasubramanian, A., Orlandi, C., Sahai, A.: On Invertible Sampling and Adaptive Security. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 466–482. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_27
30. Juels, A., Ristenpart, T.: Honey Encryption: Security Beyond the Brute-Force Bound. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 293–310. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_17
31. Lepinski, M., Micali, S., Shelat, A.: Fair-Zero Knowledge. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 245–263. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_14
32. Li, M., Vitányi, P.M.B.: Handbook of theoretical computer science, vol. a, chapter Kolmogorov Complexity and Its Applications, pp. 187–254. MIT Press, Cambridge, MA, USA (1990). ISBN 0-444-88071-2. http://dl.acm.org/citation.cfm?id=114872.114876
33. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and Its Applications. Texts in Computer Science, 4th edn. Springer, New York (2019). https://doi.org/10.1007/978-3-030-11298-1. ISBN 978-3-030-11297-4
34. Raz, R., Reingold, O.: On recycling the randomness of states in space bounded computation. In: 31st ACM STOC, pp. 159–168. ACM Press, May 1999. doi: 10.1145/301250.301294
35. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 475–484. ACM Press, May/June 2014. https://doi.org/10.1145/2591796.2591825
36. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
37. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, report 2004/332 (2004). http://eprint.iacr.org/2004/332
38. Ta-Shma, A., Umans, C., Zuckerman, D.: Loss-less condensers, unbalanced expanders, and extractors. In: 33rd ACM STOC, pp. 143–152. ACM Press, July 2001. doi: 10.1145/380752.380790
39. Trevisan, L., Vadhan, S.P.: Extracting randomness from samplable distributions. In: 41st FOCS, pp. 32–42. IEEE Computer Society Press, November 2000. doi: 10.1109/SFCS.2000.892063
40. Trevisan, L., Vadhan, S.P., Zuckerman, D.: Compression of samplable sources. Comput. Compl. **14**(3), 186–227 (2005). https://doi.org/10.1007/s00037-005-0198-6
41. von Ahn, L., Hopper, N.J., Langford, J.: Covert two-party computation. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 513–522. ACM Press, May 2005. https://doi.org/10.1145/1060590.1060668
42. Wee, H.: On pseudoentropy versus compressibility. In: 19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21–24 June 2004, Amherst, MA, USA, pp. 29–41. IEEE Computer Society (2004). ISBN 0-7695-2120-7. doi: 10.1109/CCC.2004.1313782
43. Wee, H., Wichs, D.: Candidate obfuscation via oblivious LWE sampling. Cryptology ePrint Archive, report 2020/1042 (2020). https://eprint.iacr.org/2020/1042

44. Yao, A.C.-C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3–5 November 1982, pp. 80–91. IEEE Computer Society (1982). doi: 10.1109/SFCS.1982.45
45. Zhandry, M.: The Magic of ELFs. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 479–508. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_18

Karlsruher Institut für Technologie

OPEN ACCESS

# Repository KITopen

Dies ist ein Postprint/begutachtetes Manuskript.

Empfohlene Zitierung:

Agrikola, T.; Couteau, G.; Ishai, Y.; Jarecki, S.; Sahai, A.
On Pseudorandom Encodings.
2020. Theory of Cryptography – 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings.
doi: 10.5445/IR/1000127824

Zitierung der Originalveröffentlichung:

Agrikola, T.; Couteau, G.; Ishai, Y.; Jarecki, S.; Sahai, A.
On Pseudorandom Encodings.
2020. Theory of Cryptography – 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part III. Ed.: R. Pass, 639–669.
doi:10.1007/978-3-030-64381-2_23

**Lizenzinformationen: KITopen-Lizenz**