On Pseudorandom Encodings

Thomas Agrikola[*]
Karlsruhe Institute of Technology
thomas.agrikola@kit.edu

Geoffroy Couteau[†]
IRIF, Paris-Diderot University, CNRS
couteau@irif.fr

Yuval Ishai[‡]
Technion
yuvali@cs.technion.ac.il

Stanisław Jarecki[§]
UC Irvine
stasio@ics.uci.edu

Amit Sahai[¶]
UCLA
sahai@cs.ucla.edu

April 17, 2020

Abstract.

We initiate a systematic 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 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 relation between pseudorandom encoding schemes and efficient invertible sampling algorithms, we reveal a connection between adaptively secure multi-party computation and questions in the domain of steganography.

[*] Supported by ERC Project PREP-CRYPTO 724307.

[†] Supported by ERC Projects PREP-CRYPTO 724307 and CryptoCloud 339563. Work done in part while visiting UCLA and the Technion.

[‡] Supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, BSF grant 2018393, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India.

[§] Supported by the NSF SaTC award 1817143.

[¶] 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.

Table of Contents

1 Introduction

The problem of *compression* has been extensively studied in the field of information theory and, more recently, in computational complexity and cryptography [GS85; Wee04; TVZ05; HLR07]. 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* [RR99; TV00; TUZ01; DRV12] and *resource-bounded Kolmogorov complexity* [LV90; LV19] – 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 compression algorithms, which combines the core properties of both 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 using a low-entropy secret key. Applying a standard symmetric-key encryption scheme gives rise to the following brute-force attack: try to decrypt with different keys until obtaining x' in the support of X. Variants of this attack arise in different scenarios, including password-authenticated key exchange [BM92], honey encryption [JR14], subliminal communication and steganography [HPRV19], and more. A natural solution is to use perfect compression: if x can be compressed to a random-looking string before being encrypted, it cannot be distinguished from another random-looking string 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_λ admits a *pseudorandom encoding* if there exist efficient encoding and decoding algorithms $(\mathsf{E}_X, \mathsf{D}_X)$, where D_X is deterministic, such that

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

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

Here, "≈" denotes some notion of indistinguishability (we will consider both computational and statistical indistinguishability), and the probability is over the randomness of both E_X and X_λ. The polynomial $n(\lambda)$ denotes the output length of the encoding algorithm E_X. We refer to Equation (1) as *correctness* and to Equation (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

E_X and 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, it 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 can admit a pseudorandom encoding. Henceforth, we denote this hypothesis the *pseudorandom encoding hypothesis* (PREH). A distribution family ensemble $(X_m)_{m \in L}$ 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 along with pointers to the relevant technical sections. 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 (Section 6 and Theorem 20).
- We establish a two-way relation between PRE and the previously studied notion of invertible sampling (Theorem 1). This reveals unexpected connections between seemingly unrelated problems in cryptography (e.g., between adaptive MPC and "honey encryption").
- We bootstrap "adaptive PRE" from "static PRE" (Theorem 2). As a consequence, one can base succinct adaptive MPC on standard IO as opposed to subexponential IO [CsW19] (Corollary 1).
- Under a plausible assumption, we rule out even *weak* PRE in the common *random* string model (Corollaries 8 and 10). Together with Theorem 7, this rules out adaptive MPC for general randomized functionalities in the common random string model.[2]
- We use PRE to obtain a general compiler from standard MPC protocols to covert MPC protocols (Section 7.3).

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 theoretical notion of indistinguishability. We denote the corresponding hypotheses that every efficiently samplable distribution can be

[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).

[2] This should be contrasted with positive results for "adaptively well-formed" functionalities [CLOS02], as well as positive results for general functionalities based on a (structured) common *reference* string [DKR15; CGP15; GP15; CPV17; CsW19].

pseudorandomly encoded according to the above variants as $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$, $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ and $\mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$.[3]

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 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 receive the CRS as input and the distributions in Equations (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 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 "c" and "ac", respectively. At first sight, the notion cPREH (that is, the non-adaptive variant of the pseudorandom encoding hypothesis in the CRS model) might seem rather restrictive. However, it will turn out to be highly useful. Below, we elaborate on the main flavors of pseudorandom encoding which we study in this work, and outline our feasibility and infeasibility results.

1.1.1 Deterministic, statistical pseudorandom encodings. The notion of pseudorandom encodings with a deterministic encoding algorithm and information theoretical 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 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, E_X is an optimal and efficient compression algorithm for X, with decompression algorithm 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 [TVZ05]. 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 [GS85]) 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 [HILL99]).

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, [Imp95]. One can circumvent this impossibility by studying whether compression can be achieved for more restricted classes of distributions, as was done e.g. in [TVZ05]. 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

[3] 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 results in 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 but ignore this restriction in this overview.

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.

Example. To illustrate the above, we elaborate on the example we outline at the beginning of the introduction, focusing more specifically on password-authenticated key exchange. Password authenticated key-exchange (PAKE) 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 simple and widely used protocol for PAKE is the protocol of [BM92] which has a very simple structure: the parties use their low-entropy password to encrypt the flows of a Diffie-Hellman key exchange using a block cipher. That is, party A sends $\mathsf{Enc}_{\mathsf{pw}_A}(g^a)$ where a is a random exponent in the discrete-log-hard group generated by g and pw_A is its password. Party B sends $\mathsf{Enc}_{\mathsf{pw}_B}(g^b)$ similarly. Both parties use their password to decrypt the received ciphertext and reconstruct the shared key g^{ab}. To withstand offline dictionary attacks, we must guarantee that knowing $\mathsf{Enc}_{\mathsf{pw}_A}(g^a)$ does not allow to check efficiently whether a candidate password pw' is correct. A standard way to achieve this is – when the underlying group is an elliptic curve – to rely on *point compression* algorithms for representing group elements. Point compression algorithms allow to optimally compress the representation of group elements over elliptic curves and can guarantee that random group elements are compressed into uniformly random bit strings. When the underlying block cipher satisfies the relatively standard property that decrypting a random ciphertext (of an arbitrary plaintext) under an incorrect secret key yields a random plaintext, this suffices to achieve a secure PAKE.

1.1.2 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 [HILL99] from Yao entropy [Yao82]. 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 [Wee04] and a separation for conditional distributions [HLR07] 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 [HLR07] 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, e.g. 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.

1.1.3 Randomized, statistical pseudorandom encodings. We now consider the relaxation of perfect compression by allowing the encoding algorithm to be randomized while still requiring information theoretical indistinguishability from randomness. This flavor of pseudorandom encoding is strongly related with the notion of *honey encryption* [JR14]. 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 [JR14], where it was shown that honey encryption can naturally be constructed by composing a block 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 [JR14] 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., n-bit products of two primes of the same length. Since no efficient algorithm is known to test membership to the support of this distribution, natural approaches seem to break down. In fact, we show in this work that this difficulty is inherent: building upon techniques from [BCPR14; IKOS10], we demonstrate the impossibility of (randomized, statistical) pseudorandom encodings for all efficient 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.

1.1.4 Randomized, computational pseudorandom encodings. Finally, we turn towards studying the weakest flavor of pseudorandom encodings, where the encoding algorithm is randomized and the encoded distribution is only computationally indistinguishable from randomn.

Relation to invertible sampling. We show a simple and unexpected connection with the notion of *invertible sampling* [DN00]. Informally, invertible sampling refers to the task of finding, given samples from a distribution, random coins that *explain* this sample. This notion was formally defined and studied in [IKOS10]. Following their definition, 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

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

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 authors of [IKOS10] study whether every efficient sampler admits invertible sampling; they denote this hypothesis the *invertible sampling hypothesis* (ISH). 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 important cryptographic primitives, such as oblivious sampling, trusted common reference string generations, and adaptively secure computation (which we will elaborate upon below). Building upon this connection, the impossibility result of [IKOS10] translates to our setting.

Theorem (informal). *Assuming the existence of extractable one-way functions (EOWF),* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *does not hold.*

This equivalence suggests that towards a realizable notion of pseudorandom encodings, a further relaxation is due. Thus, we ask whether the above impossibility result translates to the CRS model. We obtain both a positive and a negative result. On the negative side, we show that the result of [IKOS10] can be extended to the CRS model at the cost of assuming *unbounded auxiliary-input* extractable one-way functions, a very strong flavor of EOWFs.

Theorem (informal). *Assuming the existence of extractable one-way functions with* unbounded *common auxiliary input,* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *does not hold.*

On the positive side, however, [DKR15] build a so-called explainability compiler which implies pseudorandom encodings for all efficiently samplable distributions in the CRS model, assuming the existence of (polynomially secure) indistinguishability obfuscation.[4]

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

The above results seem contradictory at first sight. Our negative result should be interpreted with caution: The assumption of extractable one-way functions with unbounded common auxiliary input requires the existence of an extractor. However, assuming indistinguishability obfuscation, there are explicit families of adversaries and auxiliary inputs for which not even a heuristic extractor is known [BCPR14].[5]

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* [Zha16], which can be based on the exponential hardness of the decisional Diffie-Hellman problem – a strong assumption, but much more standard than indistinguishability obfuscation. However, the applicability of the resulting notion turns out to be rather restricted.

1.1.5 Overview. In Figure 1, we provide a general summary of the many flavors of the pseudorandom encoding hypothesis, and its connections to a wide variety of other primitives.

[4] Informally, an indistinguishability obfuscator iO is a compiler which 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 programs C_1 and C_2 compute the same function, then $\mathsf{iO}(C_1)$ and $\mathsf{iO}(C_2)$ are computationally indistinguishable. IO has been introduced in [BGIRSVY01] and has first been instantiated in [GGHRSW13]. Since its emergence, the existence of an indistinguishability obfuscator has become a popular assumption in computational complexity and cryptography.

[5] Note that [IKOS10] claim that their impossibility result for ISH extends to the CRS model. While technically true, the conflicting assumption, however, is false assuming indistinguishability obfuscation. Hence, assuming indistinguishability obfuscation, the impossibility result of [IKOS10] does not extend to the CRS model.

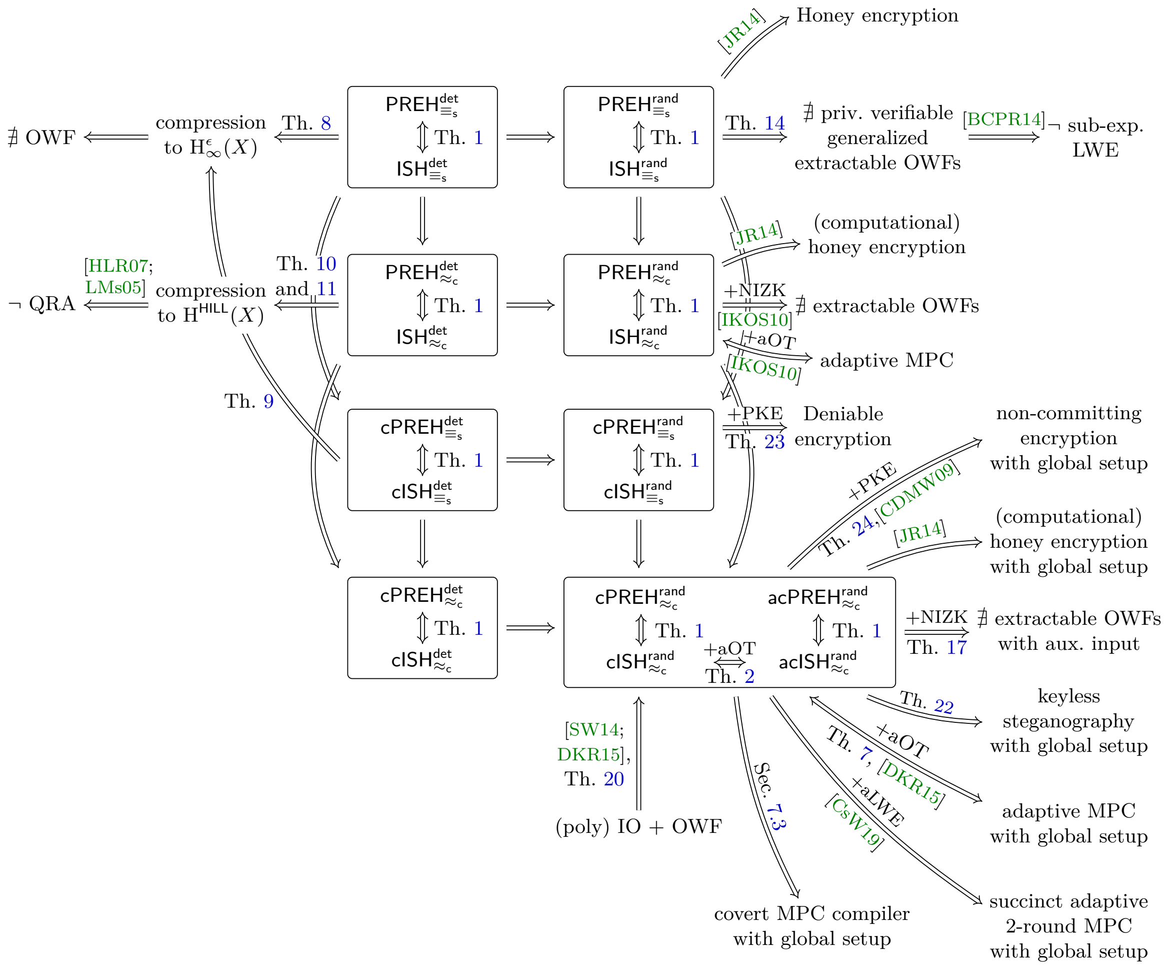
[JR14]
Honey encryption
∄ OWF
compression to $\mathrm{H}^{\epsilon}_{\infty}(X)$
Th. 8
$\mathsf{PREH}^{\mathsf{det}}_{\equiv_s}$
Th. 1
$\mathsf{ISH}^{\mathsf{det}}_{\equiv_s}$
$\mathsf{PREH}^{\mathsf{rand}}_{\equiv_s}$
Th. 1
$\mathsf{ISH}^{\mathsf{rand}}_{\equiv_s}$
Th. 14
∄ priv. verifiable generalized extractable OWFs
[BCPR14]
¬ sub-exp. LWE
[JR14]
(computational) honey encryption
[HLR07; LMs05]
¬ QRA
compression to $\mathrm{H}^{\mathsf{HILL}}(X)$
Th. 10 and 11
$\mathsf{PREH}^{\mathsf{det}}_{\approx_c}$
Th. 1
$\mathsf{ISH}^{\mathsf{det}}_{\approx_c}$
$\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$
Th. 1
$\mathsf{ISH}^{\mathsf{rand}}_{\approx_c}$
+NIZK
[IKOS10]
∄ extractable OWFs
+aOT
[IKOS10]
adaptive MPC
Th. 9
$\mathsf{cPREH}^{\mathsf{det}}_{\equiv_s}$
Th. 1
$\mathsf{cISH}^{\mathsf{det}}_{\equiv_s}$
$\mathsf{cPREH}^{\mathsf{rand}}_{\equiv_s}$
Th. 1
$\mathsf{cISH}^{\mathsf{rand}}_{\equiv_s}$
+PKE
Th. 23
Deniable encryption
+PKE
Th. 24, [CDMW09]
non-committing encryption with global setup
[JR14]
(computational) honey encryption with global setup
$\mathsf{cPREH}^{\mathsf{det}}_{\approx_c}$
Th. 1
$\mathsf{cISH}^{\mathsf{det}}_{\approx_c}$
$\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$
Th. 1
$\mathsf{cISH}^{\mathsf{rand}}_{\approx_c}$
+aOT
Th. 2
$\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$
Th. 1
$\mathsf{acISH}^{\mathsf{rand}}_{\approx_c}$
+NIZK
Th. 17
∄ extractable OWFs with aux. input
Th. 22
keyless steganography with global setup
+aOT
Th. 7, [DKR15]
adaptive MPC with global setup
+aLWE
[CsW19]
succinct adaptive 2-round MPC with global setup
[SW14; DKR15], Th. 20
(poly) IO + OWF
Sec. 7.3
covert MPC compiler with global setup

Fig. 1. An overview of the relations between the pseudorandom encoding hypothesis and other fields of cryptography and computer science. Note that since the deterministic variants of the pseudorandom encoding hypothesis are unconditionally 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*.

1.2 Implications and applications of our results

The previous section provides a broad informal overview of the various flavors of pseudorandom encodings, giving some intuition about how they relate to a wide variety of other cryptographic primitives, with a focus on our feasibility and infeasibility results for these flavors of pseudorandom encoding. 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.

1.2.1 New results for adaptively secure computation. As mentioned above, a sampler admits invertible sampling if and only if it can be pseudorandomly encoded. Due to [IKOS10], invertible sampling is strongly related to fully adaptive multi-party computation. Multi-party computation (MPC) allows parties to jointly evaluate possibly randomized functions $\mathcal{F}$ on their inputs without revealing them. [IKOS10] show 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.

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. However, as mentioned above, we only know how to instantiate the non-adaptive pseudorandom encoding hypothesis $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ in the CRS model.

Relying on [DKR15], by a non black-box use of an oblivious transfer protocol, we show that this problem can be circumvented such that even the *non-adaptive* variant $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ suffices for adaptive MPC in the CRS model. Our result further provides a powerful result: given that adaptive MPC directly implies the adaptive variant $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, we therefore obtain an instantiation of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ from $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$.

Theorem (informal). *If $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ holds and a two-round adaptively secure oblivious transfer protocol exists, then $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ holds.*

Static-to-adaptive reductions in cryptography are generally non-trivial, and often come at a strong cost in security when they rely on complexity leveraging. Our connection avoids relying on complexity leveraging by relying on adaptively-secure oblivious transfer (which can be based on standard cryptographic assumptions, e.g. LWE or DDH) in a non-black-box way. Based on this new approach to avoid complexity leveraging in static-to-adaptive reductions, we obtain several new results.

First, due to the equivalence to invertible sampling, we obtain the first instantiation of an adaptive explainability compiler [DKR15] without complexity leveraging and, hence, based only on polynomial hardness assumptions. The recent paper [CsW19] uses such an adaptive explainability compiler to obtain succinct adaptive MPC, where succinct means that the communication complexity is sub-linear in the complexity of the evaluated function. Due to our instantiation of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ from polynomial IO, we improve the results of [CsW19] by relaxing the requirement for sub-exponentially secure IO to polynomially secure IO.

Corollary (informal). *Assuming the existence of polynomially secure indistinguishability obfuscation and the adaptive hardness of the learning with errors problem, succinct adaptive MPC is possible in the global CRS model.*

Furthermore, due to the equivalence of adaptive MPC in the global CRS model and pseudorandom encodings with CRS, adaptive MPC with global common *random* string conflicts with EOWFs with common *but benign* auxiliary input (where benign means that the distribution is a natural distribution; distributions used in the obfuscation-based refutation of EOWFs with auxiliary inputs are highly unnatural, and must in particular encode some for of obfuscated circuit).

Corollary (informal). *If there exist extractable one-way functions with respect to common but benign auxiliary input, then adaptive MPC in the common random string model is impossible.*

In contrast to EOWFs with unbounded common auxiliary input, no infeasibility results are known for EOWFs with common but benign auxiliary input. This resolves an open question asked in [CsW19], whether adaptive MPC is possible with a common *random* string.

1.2.2 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 [CDNO97], which is known to exist assuming indistinguishability obfuscation [SW14]. Deniable encryption enables to "explain" ciphertexts produced for 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 [CFGN96].

Covert multi-party computation. Covert multi-party computation [vHL05; CGOS07] is an intriguing cryptographic primitive which aims at letting a group of parties jointly compute a function on their private inputs, with a very strong security guarantee: if the output of the protocol is not "favorable" to the parties, the transcript of their interaction should not leak any information, and in particular, should not reveal whether any given party was *actually taking part to the protocol*, even to the participants themselves. This strong form of secure computation aims at providing security guarantees when the very act of starting a computation with other parties should remain hidden. A classical example is that of a CIA agent infiltrated in a terrorist group, willing to make a handshake with another individual to find out whether he is also a CIA agent. Here, we show that the existence of pseudorandom encodings allows to build a general compiler which transforms a standard secure computation protocol into a covert secure computation protocol, in a round-preserving way. That is, it allows to encode each round of a protocol satisfying a weak security notion, in a reversible way, such that the encoded protocol satisfies the strong notion of covert security. Together with the equivalence between adaptively secure MPC and pseudorandom encodings, this unveils a connection between two seemingly unrelated notions of secure computation:

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

1.2.3 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 due to [JR14]. 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, computational honey encryption for all efficiently samplable distributions exists in the CRS model.*

1.3 Open questions

The certainly most intriguing question which we have to leave open is whether the weakest variant of the pseudorandom encoding hypothesis $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ can be based on weaker assumptions than indistinguishability obfuscation. A possible starting point towards this is to study whether $\mathcal{P} \neq \mathcal{NP}$ in conjunction with $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ implies the existence of one-way functions, or whether one-way functions in conjunction with $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ imply public-key encryption. We only obtain very partial results in this direction, showing that relaxed variants of this hypothesis can be based on the exponential hardness of DDH. An instantiation of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ would certainly have a significant impact on several areas of cryptography.

1.4 Overview of techniques

In this paragraph, we elaborate on some of our technical results in more detail. In the following, we use the PPT sampler S synonymous to the distribution (family) ensemble it induces.

The relation to invertible sampling. A PPT sampler S is *inverse samplable* [DN00; IKOS10], 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 Equation (3) as *closeness* and to Equation (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 [IKOS10], 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 E_S and an efficient deterministic decoding algorithm 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 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 D_S on uniform randomness is indistinguishable from applying 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 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.

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

Note that by correctness, y and y' are identical with overwhelming probability. Therefore, A is indistinguishable to $\{y' \leftarrow S(1^\lambda), r \leftarrow \mathsf{E}_S(y') \colon (r, y')\}$. Since sampling y' via 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, closeness 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 "c" and "ac", respectively. The above equivalence extends to all introduced variations of the pseudorandom encoding and invertible sampling hypotheses.

On the relation to fully adaptively secure multi-party computation. Due to the equivalence between pseudorandom encodings and invertible sampling, there is a strong link to fully adaptive (semi-honest) multi-party computation (AMPC) for all randomized functionalities. Multi-party computation provides protocols enabling parties to jointly evaluate a function on their inputs without revealing them. This is formalized via the real/ideal model paradigm, [Can00]. A functionality is said to be securely realizable if there exists a protocol such that the output of the real execution of that protocol is indistinguishable from the output of an ideal computation, where a trusted third party exists. This is required to be true even in the presence of an adversary. Hence, any adversarial behavior in the real world must be emulated in the ideal world. In the setting of multi-party computation, adaptive security means that an adversary may adaptively decide which parties to corrupt and when. In this work, we study MPC protocols tolerating *any number* of adaptive corruptions. Note that in the real world, upon corruption of a party, an adversary learns the internal randomness of that party.[6] In the ideal world, however, the internal randomness of the trusted third party can not be learned. Hence, in case of randomized functionalities, this entails an asymmetry between the adversary and its emulation making this notion particularly challenging to achieve. Consider the adversary which first observes the entire execution of the protocol and post-execution corrupts all parties. Upon each corruption the adversary learns the internal randomness explaining every protocol message of that party. Corruption of the last party is the most challenging since it basically corresponds to inverse sampling the computation of that party. The connection between fully adaptively secure multi-

[6] We do not assume secure erasures, [Lin09]. Assuming secure erasures, an adversary does not learn the internal randomness of corrupted parties.

party computation and invertible sampling was discovered in [IKOS10]. Due to [IKOS10] and our equivalence between invertible sampling and pseudorandom encodings, $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is *equivalent* to AMPC for all randomized functionalities in the plain model.

On a technical level, [IKOS10] first observe that given a randomized functionality $\mathcal{F}$, the corresponding de-randomized functionality $\mathcal{G}$ (where all parties additionally to their input provide a share of the random tape for $\mathcal{F}$) can be adaptively realized assuming an ideal oblivious transfer (OT) functionality, [Kil88; IPS08]. To adaptively realize $\mathcal{F}$ assuming an ideal $\mathcal{G}$ functionality, the challenging part consists of the simulation of an adversary which post-execution corrupts both parties. As noted above, this requires the simulator to provide a well-distributed random tape only given the parties' input and the output of $\mathcal{F}$. If $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is true and the de-randomized functionality $\mathcal{G}$ evaluates the alternative sampler $\overline{\mathcal{F}}$ instead of $\mathcal{F}$, this task is in fact feasible. As described above, we explore relaxations of the pseudorandom encoding hypothesis by admitting access to a non-programmable common reference string. If the adaptive pseudorandom encoding hypothesis in the CRS model $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is true, a similar proof as above yields AMPC for all randomized functionalities in the global common reference string (CRS) model. The global CRS model admits all parties, the adversary and the simulator access to some public reference string.[7] However, the instantiation of $\mathsf{cISH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, and hence of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, due to [DKR15] only allows static choice of inputs. That is $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ only provides any guarantees as long as the inputs to the sampler are chosen independently of the CRS. This is not the case in the above proof. An adversary may very well choose the inputs to the parties adaptively after seeing the CRS. Hence, the above proof strategy can not be applied. [DKR15] solve this issue by avoiding to use the explainable sampler on the parties' inputs directly. Instead, they use protocol messages of an adaptively secure two-round oblivious transfer protocol as inputs to the sampler. Those protocol messages do depend on the parties' inputs, however, by the security of the OT protocol, they can be simulated without knowing the inputs. Note that computational closeness actually suffices. Hence, static $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ suffices for AMPC for all PPT functionalities.

Conversely, due to [IKOS10] and our equivalence of invertible sampling and pseudorandom encodings, if every randomized functionality can be securely realized in the presence of adaptive adversaries in the plain model (i.e. without CRS), then $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is true. By a similar proof, we obtain that if every randomized functionality can be securely realized in the presence of adaptive adversaries with global CRS, then $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ is true. Note that the CRS of the MPC protocol acts as the CRS in the context of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$. Let $\mathcal{F}$ be a randomized (two-party) functionality and let $\mathit{\Pi}$ be a protocol which adaptively realizes $\mathcal{F}$. We define the alternative sampler to execute the protocol without any corruptions and output whatever the protocol returns as result. By indistinguishability between the real and the ideal world, this output is computationally indistinguishable from an output produced by the functionality $\mathcal{F}$ on the same inputs. This indistinguishability holds even if the adversary adaptively chooses the parties' inputs depending on the CRS. Hence, the alternative sampler satisfies *adaptive* closeness. For the purpose of defining a corresponding inverse sampler $\overline{S}^{-1}$, we consider the adversary $\mathcal{A}$ which observes the entire protocol without corruption and post-execution corrupts first party P_1 and outputs P_1's internal randomness and then P_2 and outputs P_2's internal randomness. By assumption, $\mathit{\Pi}$ realizes $\mathcal{F}$ in the presence of $\mathcal{A}$. Therefore, there exists a simulator Sim in the ideal world which is able to emulate $\mathcal{A}$'s behavior in the real world. We define the inverse sampler $\overline{S}^{-1}$ as follows. On input of crs, inputs (x_1, x_2) and some sample z, $\overline{S}^{-1}$ executes the simulator Sim on input of crs, z and the parties inputs x_1 and x_2 (in the same order as in the ideal world). The simulator Sim produces internal randomness r'_1 for party P_1 and r'_2 for party P_2 such that the real and the ideal world are indistinguishable and $\overline{S}^{-1}$ outputs (r'_1, r'_2). Since the adversary $\mathcal{A}$ additionally is

[7] Note that in contrast to the local CRS model, the simulator is given the CRS as input and neither knows corresponding trapdoors nor is able to alter its distribution.

in the position to choose the parties' inputs (x_1, x_2) adaptively depending on the CRS, *adaptive* invertibility follows.

This is a somewhat unexpected result since, in conclusion, the static variant $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ in conjunction with an adaptively secure two-round OT protocol implies AMPC with global CRS for all randomized functionalities which in turn implies the *adaptive* variant $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$. To the best of our knowledge, this constitutes the first instantiation of this essential primitive based only on polynomial-time hardness assumptions. The recent work [CsW19] realizes *succinct* AMPC for all randomized functionalities assuming an adaptive version of the explainability compiler from [DKR15] and that the adaptive learning with errors problem is (polynomially) hard. We recall that an explainability compiler corresponds to $\mathsf{cISH}^{\mathsf{rand}}_{\approx_c}$ with statistical closeness. Note that computational adaptive closeness is sufficient for [CsW19]. Owing to the lack of an adaptive instantiation, this incurred a sub-exponential reduction loss relative to indistinguishability obfuscation and one-way functions due to complexity leveraging [BB04]. Using our instantiation of $\mathsf{acISH}^{\mathsf{rand}}_{\approx_c}$, we are able to relax the assumptions made in [CsW19] from sub-exponential IO, sub-exponentially secure one-way functions and adaptive LWE to polynomial IO, adaptive LWE and a two-round adaptively secure OT protocol.

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_λ 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 [Wee04; TVZ05]. Pseudorandom encoding partially recovers the notion of compression if we require the encoding algorithm to be deterministic. If a source X_λ can be pseudorandomly encoded with a deterministic encoding algorithm having output length $n(\lambda)$, then X_λ 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, [TVZ05].

Clearly, the requirement for correctness, poses a lower bound on the encoding length $n(\lambda)$, [Sha48]. Conversely, requiring the encoding algorithm 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_λ 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 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 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_λ 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_λ. 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_λ 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 [HILL99] is defined via the computational indistinguishability from a truly random distribution. More formally, a distribution X_λ has HILL entropy at least k, if there exists a distribution with min-entropy k which is computationally indistinguishable from

X_λ. Hence, the encoding length $n(\lambda)$ is upper bounded by the HILL entropy of the source X_λ. Another important notion of computational entropy is the notion of Yao entropy [Yao82]. Yao entropy is defined via the incompressibility of a distribution. More precisely, a distribution X_λ has Yao entropy at least k if X_λ 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 [HLR07; LMs05], 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.[8] The above observations, particularly Equation (5), can be extended to conditional HILL and conditional Yao entropy, by considering $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ for PPT algorithms with input. Therefore, if the Quadratic Residuosity Assumption is true, $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ 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.

On the tension between invertible sampling and extraction. In cryptography and complexity theory, there is a special class of assumptions stating that if an adversary is able derive an information from his inputs, then this adversary *must know* how this information was derived based on its inputs. Knowledge is formalized computationally via the existence of a corresponding extractor which, given the code and the randomness used by the adversary, is able to *extract* how the output was computed. Extractable one-way functions (EOWFs) [BCPR14] constitute a popular instance of this class of assumptions. In this context, for every adversary which (given the function description) outputs an image of the function, there exists an extractor which, given the used randomness, is able to extract the used pre-image. Note that this does not contradict one-wayness of such functions since the randomness used to compute the image is not known in general. However, combined with the ability that every PPT algorithm can be inverse sampled, this becomes a problem. On a high level, given a PPT algorithm which outputs an image of an EOWF, there exists an inverse sampler which given the image, produces well-distributed random coins to result in that image. Due to the notion of knowledge, the extractor, given such coins, is able to extract a pre-image, hence, compromising one-wayness. Thus, even allowing the encoding algorithm to be randomized, the pseudorandom encoding hypothesis remains a very strong notion.

On a technical level, since invertibility does not guarantee indistinguishability between the randomness used by the original sampler and inverse sampled randomness, for the above reasoning to work, an adversary breaking one-wayness must work with the alternative sampler. This entails the necessity of a mechanism which guarantees that the output produced by the alternative sampler is in the range of the EOWF. Using a non-interactive zero-knowledge (NIZK) proof to certify membership in the range resolves this problem. By the above reasoning, the existence of extractable one-way functions in conjunction with NIZK proof systems conflicts with $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ as already observed in [IKOS10].

Many applications of EOWFs require that the adversary and the corresponding extractor are given access to some common auxiliary input. This auxiliary input can be required to be bounded by some fixed polynomial $b(\lambda)$ or by an arbitrary polynomial. Applying the above strategy to

[8] 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.

EOWFs with unbounded common auxiliary input, we obtain a contradiction to $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ and NIZK proof systems. Note that it is crucial that the auxiliary input is unbounded since the sampler for which we apply $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ depends on the adversary.[9] We recall that due to [DKR15], there is an instantiation of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ based on polynomial indistinguishability obfuscation and one-way functions. This, however, is not a contradiction since IO itself contradicts EOWFs with unbounded common auxiliary input [BCPR14]. This follows from the observation that if the auxiliary information is an obfuscated circuit which given a function description k evaluates that function on the pre-image $\mathsf{PRF}(k)$ for some hard-wired pseudorandom function PRF, the adversary which evaluates this circuit on the function key does have an extractor. Owing to this contradiction, there are relaxations of EOWFs with common auxiliary input, where the auxiliary input comes from some benign distribution which is considered unlikely to encode a malicious obfuscation. For instance, the uniform distribution over $\{0,1\}^{b(\lambda)}$ is conjectured to be benign. This notion suffices to obtain a contradiction between $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in the common *random* string model in conjunction with NIZK proof systems, and EOWFs with unbounded common but benign auxiliary input. For EOWFs with unbounded common but benign auxiliary input, in contrast to EOWFs with unbounded common auxiliary input, no infeasibility results are known. Hence, the existence of EOWFs with unbounded common but benign auxiliary input is considered plausible. Since $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in the common random string model is implied by AMPC for all randomized functionalities in the common random string model, this provides evidence that AMPC for all randomized functionality is impossible in the common random string model.

The notion of generalized EOWFs (GEOWFs) due to [BCPR14] generalizes EOWFs by letting the image to pre-image relation be an arbitrary relation over triplets of function keys, images and pre-images. If that relation is not efficiently testable without the originally used pre-image, the GEOWF is called privately verifiable. [BCPR14] provides an instantiation of privately verifiable GEOWFs from one-way functions and privately verifiable non-interactive universal arguments, which can be derived from the privately verifiable P-delegation scheme from [KRR14] which in turn can be based on the sub-exponential learning with errors problem. Considering information theoretical pseudorandom encodings, an efficient testing algorithm for the relation becomes unnecessary. Thus, $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}$ contradicts the sub-exponential learning with errors problem.

1.5 Acknowledgments

We would like to thank the anonymous reviewers for many helpful comments.

[9] More precisely, the adversary can be viewed as the universal adversary executing the instructions given in the auxiliary input.

2 Preliminaries

We denote by $[n]$ the set $\{1, \ldots, n\}$. Throughout this paper, λ denotes a security parameter which is given as input to all algorithms. A probabilistic polynomial time (PPT) algorithm (also referred to as an *efficient* algorithm) runs in time polynomial in the (implicit) security parameter λ. In this paper, we consider non-uniform polynomial time adversaries, i.e. polynomial time adversaries receiving a polynomially bounded auxiliary input (or advice) depending only on the security parameter.[10] A function $f(\lambda)$ is *negligible* if for any polynomial p there exists a bound $B \in \mathbb{N}$ such that, for any integer $k \geq B$, $|f(k)| \leq \frac{1}{|p(k)|}$. A function g is *overwhelming* if $1 - g(\lambda)$ is a negligible function.

Given a finite set A, the notation $x \leftarrow A$ means a uniformly random assignment of an element of A to the variable x. Given a probability distribution D, the notation $x \leftarrow D$ means sampling an element according to the distribution D and assigning that element to x. We denote the uniform distribution over bitstrings of length n by U_n. Let X, Y be two distributions over a set A. Then, the statistical distance between X and Y is defined as $\Delta(X, Y) := \sum_{a \in A} |\Pr[X = a] - \Pr[Y = a]|$. We say that two distributions are statistically close if their statistical distance is negligible.

A source X is a probability distribution on strings. A family of sources is a probability ensemble $(X_\lambda)_{\lambda \in \mathbb{N}}$, where X_λ is distributed on $\{0,1\}^{p(\lambda)}$ for some polynomial p. A family of sources $(X_m)_{m \in L}$ can also be indexed by strings from some language $L \subseteq \{0,1\}^+$, where X_m is distributed on $\{0,1\}^{p(|m|)}$ for some polynomial p. We say that a source X_λ is efficiently samplable if there is a PPT algorithm S such that $S(1^\lambda)$ is distributed according to X_λ for all $\lambda \in \mathbb{N}$. We say that a source X_m indexed by strings is efficiently samplable if there exists a PPT algorithm S such that $S(m)$ is distributed according to X_m for all $m \in L$.

In game based proofs, it will be useful to let out_i denote the output of game $\mathbf{G}_i$. If we want to make the adversary $\mathcal{A}$ playing $\mathbf{G}_i$ explicit, we write $out_{i,\mathcal{A}}$. Unless stated otherwise, we consider stateful adversaries.

The hypotheses stated in the following are formulated for the class of all PPT algorithms S, possibly excluding pathological cases. In some cases it is sufficient to consider a weaker variant of these hypotheses which is only required to be true for a specific class of PPT algorithms $\mathcal{S}$. In this case we say that the respective hypothesis holds for the class $\mathcal{S}$.

3 The pseudorandom encoding hypothesis

We the study the ability to encode efficiently samplable distributions into the uniform distribution. In the following, an efficiently samplable distribution will be defined by the corresponding sampler S with input space L. A distribution defined via S can be pseudorandomly encoded if there exists an efficient potentially randomized encoding algorithm E_S and an efficient deterministic decoding algorithm D_S such that for all $m \in L$, the probability for the event $\mathsf{D}_S(\mathsf{E}_S(S(m))) = S(m)$ is overwhelming and the distribution $\mathsf{E}_S(S(m))$ is indistinguishable from the uniform distribution $U_{n(\lambda)}$. We work with the hypothesis that every efficiently samplable distribution can be pseudorandomly encoded. In this section, we formally define the pseudorandom encoding hypothesis and its variations.

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

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

[10] Note that by a coin-fixing argument, it is sufficient to consider non-uniform deterministic adversaries. Most of our results apply for uniform PPT adversaries as well. In case we make explicit use of the non-uniformity of the adversary, we remark this explicitly.

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 in Figure 2.

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

```
$Exp^{\mathsf{pre}}_{\mathcal{A},m,1}(\lambda)$
$u \leftarrow \{0,1\}^{n(\lambda)}$
return $\mathcal{A}(m,u)$
```

Fig. 2. The pseudorandomness experiments.

Remark 1. Definition 1 formulated for PPT algorithms S which do not admit an input m is called the *weak* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$.

Remark 2 ($\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}, \mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}, \mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$). Definition 1 can be tuned in two dimensions: the encode algorithm can be required to be deterministic or allowed to be randomized, and the pseudorandomness property can be required to hold statistically or computationally. We denote these variants as $\mathsf{PREH}^{\beta}_{\alpha}$, where $\alpha \in \{\approx_{\mathsf{c}}, \equiv_{\mathsf{s}}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$.

Remark 3. Definition 1 demands indistinguishability between encoded samples and the uniform distribution over all bitstrings of some length $n(\lambda)$. This requirement can be relaxed to indistinguishability from the uniform distribution over some efficiently samplable and efficiently recognizable set $\mathcal{R}$ of size N, where elements in $\mathcal{R}$ can be represented with $O(\log N)$ bits.

Deterministic encoding and compatible samplers.

Requiring the encoding algorithm E_S to be deterministic entails the existence of what we call "incompatible samplers" for which $\mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ and even $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ are unconditionally false. For instance, consider the sampler S^* which on input of 1^λ uniformly chooses an element from the set $\{00, 01, 10\} \subset \{0,1\}^2$. Assume $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is true for S^*. Then, correctness requires that with overwhelming probability over the sampling process $y \leftarrow S^*(1^\lambda)$, $\mathsf{D}_{S^*}(\mathsf{E}_{S^*}(y)) = y$. Hence, E_{S^*} must map into the set $\{0,1\}^k$ for $k \geq 2$ (otherwise there is an correctness error of at least $\frac{1}{3}$). Pseudorandomness, on the other hand, requires that $\mathsf{E}_{S^*}(S^*(1^\lambda))$ is computationally indistinguishable from uniform distribution over $\{0,1\}^k$. However, since E_{S^*} is a deterministic algorithm, $|\mathsf{supp}(\mathsf{E}_{S^*}(S^*(1^\lambda)))| = 3$. Therefore, there exists at least an element in $\{0,1\}^k \setminus \mathsf{supp}(\mathsf{E}_{S^*}(S^*(1^\lambda)))$. An adversary can easily determine this element by evaluating E_{S^*} on each element in the support of S^* (if the support of the sampler was super-polynomial, this would not be possible for a PPT adversary, but very well possible for an unbounded one).

Another example of such an incompatible sampler is a sampler with large support but low min-entropy (i.e. a sampler that has (at least one) very likely output and many much less likely outputs). For instance, consider the sampler S' with probability distribution

$$\Pr[S'(1^\lambda) = 0^\lambda] = \frac{1}{2}$$
$$\Pr[S'(1^\lambda) = 1 \parallel x] = \frac{1}{2^\lambda} \text{ for each } x \in \{0,1\}^{\lambda-1}$$

Assume $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is true for S'. Then, correctness requires that an overwhelming fraction of the elements of the form $1 \parallel x$ for $x \in \{0,1\}^{\lambda-1}$ are correctly decodable. Furthermore, since the element 0^λ has a non-negligible probability to occur, it needs to be correctly decodable. Let the support of $\mathsf{E}_{S'}(S'(1^\lambda))$ be (a subset of) $\{0,1\}^{\lambda-1}$. Pseudorandomness requires that $\mathsf{E}_{S'}(S'(1^\lambda))$ is indistinguishable from uniform distribution over $\{0,1\}^{\lambda-1}$. However, since $\mathsf{E}_{S'}$ is deterministic, the value $\mathsf{E}_{S'}(0^\lambda)$ occurs with probability at least $\frac{1}{2}$ and, due to correctness, all other values $\mathsf{E}_{S'}(1 \parallel x)$ occur with far lower probability.

In order to obtain a meaningful definition of $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ and $\mathsf{PREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$, we restrict these hypotheses to only hold for specific classes of what we refer to as "compatible samplers" $\mathcal{S}^{\mathsf{comp}}$ and $\mathcal{S}^{\mathsf{stat}}$, respectively.

Definition 2 (Compatibility with deterministic encodings). *A sampler S is* statistically compatible with deterministic encodings *if there exists a set A whose cardinality is a power of 2, such that $\Pr[S(1^\lambda) \in A]$ is overwhelming, and S is ϵ-flat on A, i.e. for all $a \in A$ we have $|\Pr[S(1^\lambda) = a] - \frac{1}{|A|}| \leq \epsilon(\lambda)$, for some negligible function ϵ. The class $\mathcal{S}^{\mathsf{stat}}$ contains all samplers which statistically compatible with deterministic encodings.*

A sampler S is computationally compatible with deterministic encodings *if $S \in \mathcal{S}^{\mathsf{stat}}$ or if $|\mathsf{supp}(S(1^\lambda))|$ is super-polynomial and the min-entropy $\mathrm{H}_\infty(S(1^\lambda)) \in \omega(\log(\lambda))$ (i.e. the most likely event occurs with negligible probability). The class $\mathcal{S}^{\mathsf{comp}}$ contains all samplers which are computationally compatible with deterministic encodings.*

If S admits an input $z \in L$, S is statistically or computationally compatible with deterministic encodings if the corresponding criterion is met for all $z \in L$.

The above criterion for $\mathcal{S}^{\mathsf{stat}}$ may seem unnatural. We note that by relaxing Definition 1 as noted in Remark 3, requiring high min-entropy suffices for a sampler to be statistically compatible with deterministic encodings.

We note that ϵ-flatness is a weaker criterion than statistical closeness to the uniform distribution over A. ϵ-flatness on A corresponds to closeness to the uniform distribution over A with respect to the infinity norm. Statistical closeness, however, is formalized with respect to the Manhatten norm.

Let iPRG be an injective PRG with stretch ℓ. Let S be the sampler which on input of 1^λ draws a uniform seed $s \in \{0,1\}^\lambda$ and outputs $\mathsf{iPRG}(s) \in \{0,1\}^{\ell(\lambda)}$. Clearly, S is statistically compatible with deterministic encodings.

Lemma 1. *Let PRG be a PRG with stretch ℓ and let S be the sampler which on input of 1^λ produces the distribution $\mathsf{PRG}(U_\lambda)$. Then, $S \in \mathcal{S}^{\mathsf{stat}}$.*

Proof. Let $A' := \mathsf{PRG}(\{0,1\}^\lambda)$ and let $A'' \subseteq \{0,1\}^{\ell(\lambda)} \setminus A'$ such that for $A := A' \cup A''$ we have $|A| = 2^\lambda$. We have $\Pr[S(1^\lambda) \in A] = 1$.

For flatness on A, we upper bound the probability of the most likely event in A. Due to pseudorandomness of PRG, all (non-uniform) PPT adversary distinguishing the distributions $S(1^\lambda)$ and $U_{\ell(\lambda)}$ have a negligible advantage. Assume there exists an $a \in A'$, such that $\Pr[S(1^\lambda) = a] \geq \delta(\lambda)$, for a non-negligible function δ. Then, the most likely value in A' could be used as polynomial advice string allowing a non-uniform adversary to recognize the distribution $S(1^\lambda)$ with non-negligible probability $\delta(\lambda)$. (A uniform adversary can sample a random seed and evaluate the PRG. The thereby obtained output equals the most likely event a with probability $\delta(\lambda)$. Hence, a uniform adversary has advantage at least $\delta(\lambda)^2$ in distinguishing.) Therefore, for all $a \in A'$, $\Pr[S(1^\lambda) = a] \leq \epsilon(\lambda)$ for some negligible function ϵ. Hence, for all $a \in A$, we have $|\Pr[S(1^\lambda) = a] - \frac{1}{|A|}| \leq \epsilon(\lambda) + 2^{-\lambda}$ which is negligible. □

Lemma 2. *Let PRG be a PRG with stretch ℓ. Let S be the sampler which on input of $x \in L$ produces the distribution $\{y_1 \leftarrow \mathsf{PRG}(U_{|x|}), y_2 \leftarrow D_{x,y_1} : (y_1, y_2)\}$ for any distribution D. Then, $S \in \mathcal{S}^{\mathsf{comp}}$.*

Proof. Let $\lambda := |x|$. Using the argument of Lemma 1, all images in $\mathsf{PRG}(\{0,1\}^\lambda)$ have only a negligible probability to occur. Therefore, for all $x \in L$ and all $(y_1, y_2) \in \mathsf{supp}(S(x))$, we have $\Pr[S(1^\lambda, x) = (y_1, y_2)]$ is negligible. Therefore, $S \in \mathcal{S}^{\mathsf{comp}}$. □

Impossibility of universal encoding.

It is essential that the decoding algorithm D_S depends on the sampler S, since due to pseudorandomness, D_S on input of a random string needs to produce a sample that is in some sense close to the distribution produced by S. This argument does not hold necessarily for the encode algorithm. In [TVZ05], universal compression was studied. This translates to the following definition of PREH with universal encoding.

Definition 3 **($\mathsf{PREH}_\alpha^\beta$ with universal encoding).** *Let $\mathcal{S}$ be a class of sampling algorithms. We say $\mathsf{PREH}_\alpha^\beta$* with universal sampling *is true for the class $\mathcal{S}$, if there exists a universal encoding algorithm E, such that for every PPT algorithm $S \in \mathcal{S}$, there exists an efficient deterministic decoding algorithm D_S, such that correctness and pseudorandomness are satisfied.*

In contrast to universal compression, pseudorandom encoding with universal encoding is impossible.

Lemma 3. PREH *with universal encoding is not possible for arbitrary classes of samplers.*

Proof. Consider the class of sampling algorithms $\mathcal{S} = \{S_1, S_2, S_3\}$ over $\{0,1\}^{p(\lambda)}$ with support $Y_{i,\lambda} := \mathsf{supp}(S_i(1^\lambda))$. We require that for $i \in \{1,2\}$, $|Y_{i,\lambda}| = 2^{k+1}$, $|Y_{1,\lambda} \cap Y_{2,\lambda}| = 2^k$, $Y_{1,\lambda} \cap Y_{2,\lambda} = Y_{3,\lambda}$ for $k \in O(\log \lambda)$, and that S_1, S_2 and S_3 produce uniform samples over their support (i.e. correspond to flat distributions). We note that $S_1, S_2, S_3 \in \mathcal{S}^{\mathsf{stat}}$. For notational convenience we omit the dependency on λ in the following.

Let $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$. Assume toward a contradiction, that $\mathsf{PREH}_\alpha^\beta$ with universal encoding is true for the class $\mathcal{S}$ of sampling algorithms above. Let $\{0,1\}^{n(\lambda)}$ be the range of E. Due to correctness, for $i \in \{1,2\}$, there is a negligible function ϵ, such that

$$\begin{aligned}\Pr_{y \leftarrow Y_i}[\mathsf{D}_{S_i}(\mathsf{E}(y)) \neq y] &= \sum_{y' \in Y_i} \Pr_{y \leftarrow Y_i}[\mathsf{D}_{S_i}(\mathsf{E}(y)) \neq y \wedge y = y'] \\ &= \frac{1}{|Y_i|} \sum_{y' \in Y_i} \Pr_{y \leftarrow Y_i}[\mathsf{D}_{S_i}(\mathsf{E}(y)) \neq y \mid y = y'] \leq \epsilon(\lambda)\end{aligned}$$

Since $|Y_i|$ is polynomial, for each $y \in Y_i$, $\Pr[\mathsf{D}_{S_i}(\mathsf{E}(y)) = y] \geq 1 - \epsilon'(\lambda)$ for some negligible function ϵ', where the probability is solely over the randomness of E. Hence, for all $y \in Y_3 = Y_1 \cap Y_2$,

$$\Pr[u \leftarrow \mathsf{E}(y) \colon y = \mathsf{D}_{S_1}(u) = \mathsf{D}_{S_2}(u) = \mathsf{D}_{S_3}(u)] \geq 1 - \epsilon''(\lambda)$$

for some negligible function ϵ''.

Due to Theorem 1 (see Section 4.3), for $i \in \{1,2,3\}$, the distribution $\{u \leftarrow U_{n(\lambda)} \colon \mathsf{D}_{S_i}(u)\}$ is computationally (statistically) close to the distribution produced by $S_i(1^\lambda)$. Hence, (up to some negligible fraction ϵ''') the algorithms D_{S_1} and D_{S_2} map at most half of the strings in $\{0,1\}^{n(\lambda)}$ into the same set $Y_1 \cap Y_2 = Y_3$.

We build an adversary $\mathcal{A}$ on pseudorandomness with respect to sampler S_3 as follows. On input of u, $\mathcal{A}$ outputs 1 if and only if $\mathsf{D}_{S_1}(u) = \mathsf{D}_{S_2}(u)$.

$$\begin{aligned}\Pr[Exp^{\mathsf{pre}}_{\mathcal{A},0}(\lambda) = 1] &\geq 1 - \epsilon''(\lambda) \\ \Pr[Exp^{\mathsf{pre}}_{\mathcal{A},1}(\lambda) = 1] &\leq \frac{1}{2} - \epsilon'''(\lambda)\end{aligned}$$

Therefore, $\mathcal{A}$ has a non-negligible advantage $Adv^{\mathsf{pre}}_{\mathcal{A}}(\lambda)$. □

3.1 The pseudorandom encoding hypothesis with setup

We obtain a natural relaxation of the pseudorandom encoding hypothesis by introducing public parameters. We refer to these public parameters as *global* or *non-programmable* common reference string. That is, a distribution defined via S can be pseudorandomly encoded in this relaxed sense, if there exists a probabilistic setup algorithm Setup_S and encode and decode algorithms as before such that for all $m \in L$, the event $\mathsf{D}_S(crs, \mathsf{E}_S(crs, S(m))) = S(m)$ is overwhelming, 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 $(\mathsf{Setup}_S(1^\lambda), U_{n(\lambda)})$.

Definition 4 (Pseudorandom encoding hypothesis with setup, $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$). $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *holds if for every PPT algorithm S, there exists a PPT algorithm* Setup_S*, efficient algorithms* $(\mathsf{E}_S, \mathsf{D}_S)$*, where* D_S *is deterministic and* E_S *is randomized (with output length $n(\lambda)$) satisfying the following two requirements.*

Correctness. *For all PPT adversaries $\mathcal{A}$,*

$$Adv^{\mathsf{dec\text{-}error}}_{\mathcal{A}}(\lambda) := \Pr\left[m \leftarrow \mathcal{A}(1^\lambda), crs \leftarrow \mathsf{Setup}_S(1^\lambda), y \leftarrow S(m) \colon \mathsf{D}_S(crs, m, \mathsf{E}_S(crs, m, y)) = y\right]$$

is overwhelming.

Pseudorandomness. *For all PPT adversaries $\mathcal{A}$,*

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

is negligible, where $Exp^{\mathsf{crs\text{-}pre}}_{\mathcal{A},0}$ and $Exp^{\mathsf{crs\text{-}pre}}_{\mathcal{A},1}$ are defined in Figure 3.

```
$Exp^{\mathsf{crs\text{-}pre}}_{\mathcal{A},0}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := S(m; r)$
return $\mathcal{A}(crs, m, \mathsf{E}_S(crs, m, y))$
```

```
$Exp^{\mathsf{crs\text{-}pre}}_{\mathcal{A},1}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$u \leftarrow \{0,1\}^{n(\lambda)}$
return $\mathcal{A}(crs, m, u)$
```

Fig. 3. The static pseudorandomness experiments with setup.

We note that assuming non-uniform adversaries, correctness and pseudorandomness defined in Definition 4 can be equivalently defined by quantifying over all messages $m \in L$. Definition 4 is static in the sense that the inputs $m \in L$ are required to be chosen statically, i.e. independently of crs. In the following, we define the corresponding adaptive variant.

Definition 5 (Adaptive pseudorandom encoding hypothesis with setup, $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$). *For every PPT algorithm S, there exists a PPT algorithm* Setup_S*, efficient algorithms* $(\mathsf{E}_S, \mathsf{D}_S)$*, where* D_S *is deterministic and* E_S *is randomized (with output length $n(\lambda)$) satisfying the following two requirements.*

Correctness. *For all PPT adversaries $\mathcal{A}$,*

$$Adv^{\mathsf{a\text{-}dec\text{-}error}}_{\mathcal{A}}(\lambda) := \Pr\left[crs \leftarrow \mathsf{Setup}_S(1^\lambda), m \leftarrow \mathcal{A}(crs), y \leftarrow S(m) \colon \mathsf{D}_S(crs, m, \mathsf{E}_S(crs, m, y)) = y\right]$$

is overwhelming.

Pseudorandomness. *For all PPT adversaries $\mathcal{A}$,*

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

is negligible, where $Exp^{\mathsf{a\text{-}crs\text{-}pre}}_{\mathcal{A},0}$ and $Exp^{\mathsf{a\text{-}crs\text{-}pre}}_{\mathcal{A},1}$ are defined in Figure 4.

Definition 6 ($\mathsf{cPREH}^{\mathsf{det}}_{\approx_\mathsf{c}}, \mathsf{cPREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}, \mathsf{cPREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}, \mathsf{acPREH}^{\mathsf{det}}_{\approx_\mathsf{c}}, \mathsf{acPREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}, \mathsf{acPREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$). *Definitions 4 and 5 can be tuned in two dimensions: the encoding algorithm can be required to be deterministic or allowed to be randomized, and the closeness and invertibility properties can be required to hold statistically or computationally. We denote these variants as $\mathsf{cPREH}^{\beta}_{\alpha}$ and $\mathsf{acPREH}^{\beta}_{\alpha}$, respectively, where $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$.*

```
$Exp^{\textsf{a-crs-pre}}_{\mathcal{A},0}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := S(m;r)$
return $\mathcal{A}(crs, m, \mathsf{E}_S(crs, m, y))$
```

```
$Exp^{\textsf{a-crs-pre}}_{\mathcal{A},1}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$u \leftarrow \{0,1\}^{n(\lambda)}$
return $\mathcal{A}(crs, m, u)$
```

Fig. 4. The adaptive pseudorandomness experiments with setup.

Remark 4 (Universal setup). In Definitions 4 and 5, we allow the algorithm Setup_S to depend on the sampler S. A natural strengthening of this definition is to require the existence of a *universal setup algorithm* $\mathsf{Setup}(1^\lambda, B)$ for $B \in \mathbb{N}$ which provides the above guarantees for all samplers which can be represented with B bits. We refer to this variant as universal $\mathsf{cPREH}^\beta_\alpha$ and universal $\mathsf{acPREH}^\beta_\alpha$, respectively.

Remark 5 (Common random string). We will denote the strengthening of the pseudorandom encoding hypothesis as in Definitions 4 and 5, where the setup algorithm Setup_S is required to sample uniform random strings, as the pseudorandom encoding hypothesis with a *common random string.*

Note that in Definitions 4 and 5, we implicitly only consider *legitimate* adversaries which guarantee that $m \in L$. For the sake of avoiding notational overhead, we do not make this explicit.

4 Pseudorandom encodings and invertible sampling

In this section we explain relation between pseudorandom encodings and invertible sampling, [DN00]. In Sections 4.1 and 4.2, we restate the invertible sampling hypothesis of [IKOS10] and define several variants thereof. In Section 4.3, we prove that a distribution can be pseudorandomly encoded if and only if it is inverse samplable. This extends to all of the introduced variations of the pseudorandom encoding hypothesis and the invertible sampling hypothesis.

4.1 The invertible sampling hypothesis

A PPT algorithm S is inverse samplable according to [DN00; IKOS10] if there exists an alternative PPT algorithm $\overline{S}$ and a corresponding inverse sampler $\overline{S}^{-1}$ such that $\overline{S}$ (on every input) induces a distribution which is computationally indistinguishable from the distribution induced by S (on identical inputs) and $\overline{S}^{-1}$ inverses the computation of $\overline{S}$. That is, $\overline{S}^{-1}$ on input of an output produced by $\overline{S}$ produces computationally well-distributed random coins for $\overline{S}$ to produce the given output.

Definition 7 (Invertible sampling hypothesis, $\mathsf{ISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$, [IKOS10]). *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 $m \in L$,*

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

where $Exp^{\mathsf{close}}_{\mathcal{A},m,0}$ and $Exp^{\mathsf{close}}_{\mathcal{A},m,1}$ are defined in Figure 5.

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

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

where $Exp^{\mathsf{inv}}_{\mathcal{A},m,0}$ and $Exp^{\mathsf{inv}}_{\mathcal{A},m,1}$ are defined in Figure 5.

```
$Exp^{\mathsf{close}}_{\mathcal{A},m,0}(\lambda)$
$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := S(m; r)$
return $\mathcal{A}(m, y)$
```

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

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

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

Fig. 5. The closeness and invertibility experiments.

Definition 8 ($\mathsf{ISH}^{\mathsf{det}}_{\approx_\mathsf{c}}, \mathsf{ISH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}, \mathsf{ISH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$). *Definition 7 can be tuned in two dimensions: the inverse sampler can be required to be deterministic or allowed to be randomized, and the closeness and invertibility properties can be required to hold statistically or computationally. We denote these variants as $\mathsf{ISH}^{\beta}_{\alpha}$, where $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$.*

4.2 The invertible sampling hypothesis with setup

The invertible sampling hypothesis can be naturally relaxed by introducing public parameters (or global CRS), henceforth denoted *crs*. This allows the alternative sampler and the inverse sampler to use *crs*. Closeness and invertibility are defined against adversaries knowing the *crs* but choosing the inputs statically.

Definition 9 (Invertible sampling hypothesis with setup, $\mathsf{cISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$). *For every PPT algorithm S there exists a PPT algorithm Setup_S, a PPT algorithm $\overline{S}$ (with randomness space $\{0,1\}^{n(\lambda)}$) and an efficient randomized $\overline{S}^{-1}$ satisfying the following two properties.*

Closeness. *For all PPT adversaries $\mathcal{A}$,*

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

where $Exp^{\mathsf{crs\text{-}close}}_{\mathcal{A},0}$ and $Exp^{\mathsf{crs\text{-}close}}_{\mathcal{A},1}$ are defined in Figure 6.

Invertibility. *For all PPT adversaries $\mathcal{A}$,*

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

where $Exp^{\mathsf{crs\text{-}inv}}_{\mathcal{A},0}$ and $Exp^{\mathsf{crs\text{-}inv}}_{\mathcal{A},1}$ are defined in Figure 6.

```
$Exp^{\mathsf{crs\text{-}close}}_{\mathcal{A},0}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := S(m; r)$
return $\mathcal{A}(crs, y)$
```

```
$Exp^{\mathsf{crs\text{-}close}}_{\mathcal{A},1}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
return $\mathcal{A}(crs, y)$
```

```
$Exp^{\mathsf{crs\text{-}inv}}_{\mathcal{A},0}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
return $\mathcal{A}(crs, r, y)$
```

```
$Exp^{\mathsf{crs\text{-}inv}}_{\mathcal{A},1}(\lambda)$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
$\overline{r} \leftarrow \overline{S}^{-1}(crs, m, y)$
return $\mathcal{A}(crs, \overline{r}, y)$
```

Fig. 6. The static closeness and invertibility experiments with setup.

Definition 9 is static in the sense that closeness and invertibility adversaries are required to statically choose the challenge input $m \in L$. In the following, we define the corresponding adaptive version, where adversaries are allowed to choose the challenge input $m \in L$ depending on *crs*.

Definition 10 (Adaptive invertible sampling hypothesis with setup, $\mathsf{acISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$). *For every PPT algorithm S, there exists a PPT algorithm Setup_S, a PPT algorithm $\overline{S}$ (with randomness space $\{0,1\}^{n(\lambda)}$) and an efficient randomized $\overline{S}^{-1}$ satisfying the following two properties.*

(Adaptive) closeness. *For all PPT adversaries $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{\textsf{a-crs-close}}(\lambda) := \left|\Pr[Exp_{\mathcal{A},0}^{\textsf{a-crs-close}}(\lambda) = 1] - \Pr[Exp_{\mathcal{A},1}^{\textsf{a-crs-close}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $Exp_{\mathcal{A},0}^{\textsf{a-crs-close}}$ and $Exp_{\mathcal{A},1}^{\textsf{a-crs-close}}$ are defined in Figure 7.

(Adaptive) invertibility. *For all PPT adversaries $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{\textsf{a-crs-inv}}(\lambda) := \left|\Pr[Exp_{\mathcal{A},0}^{\textsf{a-crs-inv}}(\lambda) = 1] - \Pr[Exp_{\mathcal{A},1}^{\textsf{a-crs-inv}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $Exp_{\mathcal{A},0}^{\textsf{a-crs-inv}}$ and $Exp_{\mathcal{A},1}^{\textsf{a-crs-inv}}$ are defined in Figure 7.

```
$Exp_{\mathcal{A},0}^{\textsf{a-crs-close}}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := S(m; r)$
return $\mathcal{A}(y)$
```

```
$Exp_{\mathcal{A},1}^{\textsf{a-crs-close}}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
return $\mathcal{A}(y)$
```

```
$Exp_{\mathcal{A},0}^{\textsf{a-crs-inv}}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
return $\mathcal{A}(r, y)$
```

```
$Exp_{\mathcal{A},1}^{\textsf{a-crs-inv}}(\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$m \leftarrow \mathcal{A}(crs)$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \overline{S}(crs, m; r)$
$\overline{r} \leftarrow \overline{S}^{-1}(crs, m, y)$
return $\mathcal{A}(\overline{r}, y)$
```

Fig. 7. The adaptive closeness and invertibility experiments with setup.

Definition 11 ($\mathsf{cISH}^{\mathsf{det}}_{\approx_\mathsf{c}}, \mathsf{cISH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}, \mathsf{cISH}^{\mathsf{det}}_{\equiv_\mathsf{s}}, \mathsf{acISH}^{\mathsf{det}}_{\approx_\mathsf{c}}, \mathsf{acISH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}, \mathsf{acISH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$). *Definitions 9 and 10 can be tuned in two dimensions: the inverse sampler can be required to be deterministic or allowed to be randomized, and the closeness and invertibility properties can be required to hold statistically or computationally. We denote these variants as $\mathsf{cISH}^\beta_\alpha$ and $\mathsf{acISH}^\beta_\alpha$, respectively, where $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$.*

Remark 6 (Universal setup). In Definitions 9 and 10, we allow the algorithm Setup_S to depend on the sampler S. A natural strengthening of this definition is to require the existence of a *universal setup algorithm* $\mathsf{Setup}(1^\lambda, B)$ for $B \in \mathbb{N}$ which provides the above guarantees for all samplers which can be represented with B bits. We refer to this variant as universal $\mathsf{cISH}^\beta_\alpha$ and universal $\mathsf{acISH}^\beta_\alpha$, respectively.

Remark 7 (Common random string). We will denote the strengthening of the invertible sampling hypothesis as in Definitions 9 and 10, where the setup algorithm Setup_S is required to sample uniform random strings, as the invertible sampling hypothesis with a *common random string.*

Again, we note that in Definitions 9 and 10, we implicitly only consider *legitimate* adversaries which guarantee that $m \in L$.

4.3 Equivalence of pseudorandom encodings and invertible sampling

We formally prove the following theorem.

Theorem 1. *Let $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$. $\mathsf{PREH}^\beta_\alpha$ is true if and only if $\mathsf{ISH}^\beta_\alpha$ is true.*

It is straight forward to extend Theorem 1 to the non-adaptive and adaptive variants with setup.

4.3.1 Every inverse samplable distribution can be pseudorandomly encoded

Lemma 4. *Let $\alpha \in \{\approx_\mathsf{c}, \equiv_\mathsf{s}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$. If $\mathsf{ISH}^\beta_\alpha$ holds, then $\mathsf{PREH}^\beta_\alpha$ holds.*

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

Assume $\mathsf{ISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ holds. Let S be a PPT algorithm. $\mathsf{ISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ 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).

Correctness.

We consider a series of hybrids, see Figure 8.

$\mathbf{G}_0$	$\mathbf{G}_1$	$\mathbf{G}_2$
$r \leftarrow \{0,1\}^{n(\lambda)}$	$r \leftarrow \{0,1\}^{n(\lambda)}$	$r \leftarrow \{0,1\}^{p(\lambda)}$
$y := \overline{S}(m;r)$	$y := \overline{S}(m;r)$	$y := S(m;r)$
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. 8. Hybrids used in the proof of correctness of Lemma 4.

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, $\epsilon_{\mathsf{dec\text{-}error}}(\lambda) = \Pr[y \leftarrow S(m) \colon \overline{S}(m, \overline{S}^{-1}(m,y)) \neq y] = \Pr[out_{2,\mathcal{A}} = 1] = |\Pr[out_{2,\mathcal{A}} = 1] - \Pr[out_{0,\mathcal{A}} = 1]|$.

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 Figure 9.

$\mathbf{G}_0$	$\mathbf{G}_1$	$\mathbf{G}_2$
$r \leftarrow \{0,1\}^{p(\lambda)}$	$r \leftarrow \{0,1\}^{n(\lambda)}$	$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := S(m;r)$	$y := \overline{S}(m;r)$	**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. 9. Hybrids used in the proof of pseudorandomness of Lemma 4.

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 $\overline{\mathcal{A}}'$. □

4.3.2 Every pseudorandomly encodable distribution can be inversely sampled

Lemma 5. *Let $\alpha \in \{\approx_{\mathsf{c}}, \equiv_{\mathsf{s}}\}$ and $\beta \in \{\mathsf{rand}, \mathsf{det}\}$. If $\mathsf{PREH}^{\beta}_{\alpha}$ holds, then $\mathsf{ISH}^{\beta}_{\alpha}$ holds.*

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

Assume $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ holds. Let S be a PPT algorithm. $\mathsf{PREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ implies that for S there exist efficient algorithms E_S (potentially randomized) with output length $n(\lambda)$ and 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 Figure 10.

$\mathbf{G}_0$	$\mathbf{G}_1$	$\mathbf{G}_2$	$\mathbf{G}_3$	$\mathbf{G}_4$
$r_S \leftarrow \{0,1\}^{p(\lambda)}$	$r_S \leftarrow \{0,1\}^{p(\lambda)}$	$r_S \leftarrow \{0,1\}^{p(\lambda)}$	$r_S \leftarrow \{0,1\}^{p(\lambda)}$	$r_D \leftarrow \{0,1\}^{n(\lambda)}$
$y_S := S(m; r_S)$	$y_S := S(m; r_S)$	$y_S := S(m; r_S)$	$y_S := S(m; r_S)$	$y_D := \mathsf{D}_S(m, r_D)$
return $\mathcal{A}(m, y_S)$	$r_D \leftarrow \mathsf{E}_S(m, y_S)$	$r_D \leftarrow \mathsf{E}_S(m, y_S)$	$r_D \leftarrow \{0,1\}^{n(\lambda)}$	**return** $\mathcal{A}(m, y_D)$
	$y_D := \mathsf{D}_S(m, r_D)$	$y_D := \mathsf{D}_S(m, r_D)$	$y_D := \mathsf{D}_S(m, r_D)$	
	return $\mathcal{A}(m, y_S)$	**return** $\mathcal{A}(m, y_D)$	**return** $\mathcal{A}(m, y_D)$	

Fig. 10. Hybrids used in the proof of closeness of Lemma 5.

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, [Sho04]) bounds $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq \Pr[F]$. Correctness guarantees that for all $m \in L$, $\Pr[F] = \Pr[y_S \leftarrow S(m) \colon \mathsf{D}_S(m, \mathsf{E}_S(m, y_S)) \neq y_S] = \epsilon_{\mathsf{dec\text{-}error}}(\lambda)$ 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 $\Pr[out_2 = 1] = \Pr[Exp^{\mathsf{pre}}_{\overline{\mathcal{A}},m,0}(\lambda) = 1]$. □

Finally, the difference between $\mathbf{G}_3$ and $\mathbf{G}_4$ is only conceptional and $\Pr[out_3 = 1] = \Pr[out_4 = 1]$. Hence, $Adv^{\mathsf{close}}_{\mathcal{A},m}(\lambda) = |\Pr[out_4 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda) + \epsilon_{\mathsf{dec\text{-}error}}(\lambda)$ for some PPT adversary $\overline{\mathcal{A}}$.

Invertibility.

We consider a sequence of hybrids, see Figure 11.

```
$\mathbf{G}_0$
$r \leftarrow \{0,1\}^{n(\lambda)}$
$y := \mathsf{D}_S(m, r)$
$\overline{r} \leftarrow \mathsf{E}_S(m, y)$
return $\mathcal{A}(m, \overline{r}, y)$
```

```
$\mathbf{G}_1$
$r_S \leftarrow \{0,1\}^{p(\lambda)}$
$y_S := S(m; r_S)$
$r_D \leftarrow \mathsf{E}_S(m, y_S)$
return $\mathcal{A}(m, r_D, y_S)$
```

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

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

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

```
$\mathbf{G}_5$
$r_D \leftarrow \{0,1\}^{n(\lambda)}$
$y_D := \mathsf{D}_S(m, r)$
return $\mathcal{A}(m, r_D, y_D)$
```

Fig. 11. Hybrids used in the proof of invertibility of Lemma 5.

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

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 $out_2 = 1 \land \neg \Leftrightarrow out_3 \land \neg F$. The Difference Lemma (due to Shoup, [Sho04]) bounds $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq \Pr[F]$. Due to correctness, for all $m \in L$, $\Pr[F] = \Pr[y_S \leftarrow S(m) \colon \mathsf{D}_S(m, \mathsf{E}_S(m, y_S)) \neq y_S] = \epsilon_{\mathsf{dec\text{-}error}}(\lambda)$ 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, $|\Pr[out_5 = 1] - \Pr[out_0 = 1]| \leq 2 \cdot Adv^{\mathsf{pre}}_{\overline{\mathcal{A}},m}(\lambda) + 2 \cdot \epsilon_{\mathsf{dec\text{-}error}}(\lambda)$ for some PPT adversary $\overline{\mathcal{A}}$. □

The above proof directly generalizes to the non-adaptive and adaptive variants of pseudorandom encodings and invertible sampling with public parameters. As a demonstration, consider the proof of closeness of Lemma 5. More specifically, consider the game hop from $\mathbf{G}_1$ to $\mathbf{G}_2$. In these games, the (stateful) adversary $\mathcal{A}$ picks m (either statically or adaptively after seeing crs). The error event F occurs if $y_S \neq y_D$, where $y_S \leftarrow S(m)$ and $y_D \leftarrow \mathsf{D}_S(crs, m, \mathsf{E}_S(crs, m, y_S))$. We can hence construct an adversary $\overline{\mathcal{A}}$ against correctness which chooses m (non-)adaptively like $\mathcal{A}$. Hence, the probability that F occurs can be upper bounded by $Adv^{\mathsf{dec\text{-}error}}_{\overline{\mathcal{A}}}(\lambda)$ or $Adv^{\mathsf{a\text{-}dec\text{-}error}}_{\overline{\mathcal{A}}}(\lambda)$, respectively, which are both negligible by correctness.

5 Pseudorandom encodings and fully adaptively secure multi-party computation

Due to the equivalence of pseudorandom encodings and invertible sampling, we obtain an equivalence between pseudorandom encodings and fully adaptively secure multi-party computation. Due to [IKOS10], $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in conjunction with some adaptively secure oblivious transfer protocol implies fully adaptively secure multi-party computation for all randomized functionalities in the plain model. Interestingly, due to [DKR15], the static variant of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in conjunction with an adaptively secure two-round oblivious transfer protocol suffices for fully adaptively secure multi-party computation for all randomized functionalities in the global common reference string model. This observation yields the following connection between the static and the adaptive variant of the pseudorandom encoding hypothesis with computational indistinguishability, randomized encoding algorithm and setup.

Theorem 2. *If $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ is true and an adaptively secure two-round OT protocol exists, then $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ is true.*

In this section we first introduce some preliminaries on adaptive multi-party computation, henceforth denoted AMPC. In Section 5.2, we restate the result of [DKR15] that (static) $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in conjunction with an adaptively secure two-round OT protocol implies AMPC for all PPT functionalities in the presence of semi-honest adaptive adversaries. Conversely, AMPC for all PPT functionalities in the presence of semi-honest adaptive adversaries implies (adaptive) $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$, see Theorem 7 in Section 5.3.

Adaptive MPC.

The definition of security of multi-party protocols follows the real/ideal model paradigm, [Can00]. A protocol Π is said to be secure if the output of the real execution of the protocol is indistinguishable from the output of an ideal computation, where a trusted third party exists.

Let $\mathcal{F}$ be a PPT functionality. Let x_i be the input to party P_i. We only consider semi-honest adversaries. Semi-honest adversaries are bound to follow the protocol specification while trying to obtain as much information as possible. There are general techniques to transform protocols which are secure with respect to semi-honest adversaries to protocols which are secure with respect to malicious adversaries at the cost of a local CRS which is inherent for malicious security, [CLOS02]. In the following, we describe the stand-alone model for adaptive MPC on a very high level. We refer the reader to [Can00; Lin09] for more details.

The ideal execution, [Can00]. The ideal execution involves an ideal PPT adversary Sim (also called the simulator), a PPT environment $\mathcal{Z}$ and a trusted third party T. The ideal execution proceeds through several stages. In the *first corruption stage*, Sim adaptively decides to corrupt a party. When Sim corrupts a party, Sim learns that party's input and $\mathcal{Z}$ learns the identity of the corrupted party. In the *computation stage*, the uncorrupted and the corrupted parties send their inputs to T. The trusted party T evaluates the (possibly randomized) function $(z_1, \dots, z_n) \leftarrow \mathcal{F}(x_1, \dots, x_n)$ and sends the output to the respective parties. In the *second corruption stage*, Sim learns the outputs of the corrupted parties and again adaptively decides to corrupt a party. Upon corruption of a party, Sim learns that party's input *and output* and $\mathcal{Z}$ learns the corrupted party's identity. In the *output stage*, the uncorrupted parties output what they received from T to $\mathcal{Z}$, the corrupted parties output $\perp$ to $\mathcal{Z}$ and Sim outputs an arbitrary string to $\mathcal{Z}$. Finally, in the *post-execution corruption stage*, as long as the environment $\mathcal{Z}$ did not halt, $\mathcal{Z}$ sends corruption requests to Sim. Sim generates an arbitrary answer based on its view so far. Furthermore, Sim may additionally corrupt parties as in the second corruption stage.

Let $\mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(1^\lambda)$ denote the output distribution of the environment $\mathcal{Z}$ when interacting as described above with the simulator Sim and parties $P_1, \dots, P_n$ on inputs $x_1, \dots, x_n$ chosen by $\mathcal{Z}$.

The real execution, [Can00]. Initially, the environment $\mathcal{Z}$ (adaptively) chooses inputs $x_1, \ldots, x_n$ and sends each party P_i its input. Before the communication rounds start, the adversary $\mathcal{A}$ receives an initial message from $\mathcal{Z}$. While there exist uncorrupted parties which did not halt, $\mathcal{A}$ may adaptively decide to corrupt new parties. Upon corruption of P_i, $\mathcal{Z}$ learns the identity of P_i, $\mathcal{A}$ learns the input x_i, P_i's internal state (i.e. the random tape) and all messages received so far. From that time on, $\mathcal{A}$ is in control of the messages this party sends. Furthermore, if a corrupted party receives a message, $\mathcal{A}$ also learns that message. Additionally, $\mathcal{A}$ determines the order in which the uncorrupted parties are activated. If all parties are corrupted or all uncorrupted parties halted, each uncorrupted party and $\mathcal{A}$ produce outputs. The environment $\mathcal{Z}$ learns all of these outputs. Similarly to the post-execution corruption stage in the ideal execution, while $\mathcal{Z}$ did not halt, $\mathcal{Z}$ may instruct $\mathcal{A}$ to corrupt more parties or $\mathcal{A}$ may decide himself to corrupt more parties. Upon corruption, $\mathcal{A}$ learns P_i's input, randomness and all messages received so far and $\mathcal{Z}$ learns the corrupted party's identity. Additionally, $\mathcal{A}$ sends P_i's internal state to $\mathcal{Z}$.

Let $\mathcal{Z}^{\mathcal{A},\Pi}(1^\lambda)$ denote the output of the environment $\mathcal{Z}$ when interacting with the adversary $\mathcal{A}$ and parties $P_1, \ldots, P_n$ running the protocol Π on inputs $x_1, \ldots, x_n$ chosen by $\mathcal{Z}$.

Definition 12 (Adaptive security in the stand-alone model, [Can00]). *We say a protocol Π computes functionality $\mathcal{F}$ in the adaptive semi-honest model if for every PPT adversary $\mathcal{A}$, there exists a PPT simulator* Sim*, such that for all PPT environments $\mathcal{Z}$,*

$$\left|\Pr[\mathcal{Z}^{\mathcal{A},\Pi}(1^\lambda) = 1] - \Pr[\mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(1^\lambda) = 1]\right|$$

is negligible.

Common reference string model. In the common reference string (CRS) model, all parties have access to a string which is honestly generated by a trusted third non-participating party. There are two widely used variants of the CRS model. In the *programmable* or *local* CRS model, the simulator may generate the CRS. This enables the simulator to sample the CRS along with corresponding trapdoors which results in an asymmetry between the simulator and the adversary facilitating simulation. However, as soon as two different protocols use the same programmable CRS, all security guarantees related to that CRS break down, see [CDPW07]. In the *non-programmable* or *global* CRS model, the simulator receives the CRS as input and, hence, has no additional power compared to the adversary. A global CRS can be made public and used by any number of protocols without compromising security.

Adaptive security in the global CRS model (with respect to some efficiently samplable CRS distribution Setup) is defined as Definition 12 with the difference that all parties including the environment, the adversary and the simulator receive the CRS as input. We stress that the environment may decide on the inputs for the parties adaptively after seeing the CRS.

The universal composability (UC) framework, [Can01]. The following paragraph describes the UC framework on a very high level. We refer the reader to [Can01] for more details on the model.

In contrast to the stand-alone model for secure computation, the UC model allows the environment to be interactive. In particular, every message that is transmitted between parties is sent to the environment which can arbitrarily decide how to deal with it. Hence, the strongest possible adversary in this model is the so called dummy adversary who simply forwards all instructions he receives from the environment. The UC framework offers a set of security-preserving composition theorems which allow for a modular analysis. If a protocol Π UC-realizes a functionality $\mathcal{F}$, a protocol Π' which uses protocol Π as a subroutine can be proven to securely realize a functionality $\mathcal{F}'$ in the $\mathcal{F}$-hybrid model, that is having access to the ideal functionality $\mathcal{F}$.

Definition 13 (Adaptive UC-security, [Can01]). *Let $\mathcal{A}$ be the dummy adversary and let θ denote the dummy protocol. We say a protocol Π UC-realizes a functionality $\mathcal{F}$ in the $\mathcal{G}$-hybrid* *model in the presence of semi-honest adaptive adversaries if for all PPT environments $\mathcal{Z}$, there exists a simulator* Sim*, such that*

$$\left|\Pr[\mathsf{real}[\mathcal{Z}, \mathcal{A}, \pi^{\mathcal{G}}] = 1] - \Pr[\mathsf{ideal}[\mathcal{Z}, \mathsf{Sim}, \theta^{\mathcal{F}}] = 1]\right|$$

is negligible, where $\mathsf{real}[\mathcal{Z}, \mathcal{A}, \pi^{\mathcal{G}}]$ *denotes the output distribution of $\mathcal{Z}$ when interacting in the real world with $\mathcal{A}$ and $\pi^{\mathcal{G}}$ and* $\mathsf{ideal}[\mathcal{Z}, \mathsf{Sim}, \theta^{\mathcal{F}}]$ *denotes the output distribution of $\mathcal{Z}$ when interacting in the ideal world with* Sim *and $\theta^{\mathcal{F}}$.*

UC-security is a stronger notion than stand-alone security. More precisely, if there exists a protocol which UC-realizes a functionality according to Definition 13, then there exists a protocol which securely computes that functionality according to Definition 12.

In the UC framework, the global CRS model corresponds to the $\mathcal{F}^{\mathsf{crs}}_{\mathsf{Setup}}$-hybrid model. The ideal functionality $\mathcal{F}^{\mathsf{crs}}_{\mathsf{Setup}}$ can be queried by any party and the adversary. Upon receiving such a query, $\mathcal{F}^{\mathsf{crs}}_{\mathsf{Setup}}$ gives the CRS crs to the querying party. If crs is not initialized, $\mathcal{F}^{\mathsf{crs}}_{\mathsf{Setup}}$ samples crs according to $\mathsf{Setup}(1^{\lambda})$.

5.1 UC-secure AMPC in the plain model

[IKOS10] already observed that adaptively secure oblivious transfer in conjunction with $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ yields AMPC for all randomized functionalities in the plain model.

Theorem 3 (informal, [Kil88; IPS08]). *Any deterministic functionality can be UC-realized in the OT-hybrid model in the presence of semi-honest adversaries adaptively corrupting any number of parties.*

Theorem 4 ([IKOS10]). *Assume* $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *holds, then any functionality $\mathcal{F}$ can be UC-realized, in the $\mathcal{F}_{\mathsf{OT}}$-hybrid-model, in the presence of semi-honest adaptive adversaries corrupting any number of parties.*

Proof. For self-containment, we restate the proof of [IKOS10]. We consider functionalities $\mathcal{F}$ giving the same output to all parties. (This is without loss of generality since evaluating the function $\mathcal{F}'((x_1, k_1), (x_2, k_2)) := \mathcal{F}_1(x_1, x_2) \oplus k_1 \parallel \mathcal{F}_2(x_1, x_2) \oplus k_2$ yields the general case, see [GL91].) I.e. the functionality $\mathcal{F}$ takes as input x_1, x_2 (the inputs of the parties P_1, P_2, respectively) and internal randomness ρ, and outputs z to both parties. We consider the case of two parties (the proof easily extends to the case of many parties).

We view $\mathcal{F}$ as a PPT algorithm S. Due to $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$, there exist $\overline{S}, \overline{S}^{-1}$ satisfying closeness and invertibility. Define a deterministic functionality $\mathcal{G}$ which takes inputs (x_1, ρ_1) from party P_1 and (x_2, ρ_2) from party P_2, and evaluates $\overline{S}(x_1, x_2; \rho_1 \oplus \rho_2)$. Due to Theorem 3, $\mathcal{G}$ can be UC-realized in the OT-hybrid model in the presence of semi-honest adversaries adaptively corrupting any number of parties.

Now we turn to realize $\mathcal{F}$ in the $\mathcal{G}$-hybrid model. Party P_i chooses randomness ρ_i, feeds (x_i, ρ_i) into $\mathcal{G}$ and waits for the output. We assume that $\mathcal{Z}$ first observes the entire protocol before corrupting parties. The simulator Sim works as follows. Sim only receives the output z of the ideal functionality $\mathcal{F}$ as input. If both parties are corrupted (first P_1, then P_2), Sim outputs a uniformly random string ρ_1 to explain the internal randomness of party P_1, computes $\rho \leftarrow \overline{S}^{-1}((x_1, x_2), z)$ and outputs $\rho_2 := \rho \oplus \rho_1$ to explain the internal randomness of party P_2 (note that Sim learns the inputs x_1, x_2 at the time of corrupting P_1, P_2, respectively). Due to invertibility and closeness, for all environments $\mathcal{Z}$, $|\Pr[\mathsf{real}[\mathcal{Z}, \mathcal{A}, \pi^{\mathcal{G}}] = 1] - \Pr[\mathsf{ideal}[\mathcal{Z}, \mathsf{Sim}, \theta^{\mathcal{F}}] = 1]|$ is negligible. More precisely, consider the hybrids in Figure 12. Note that $\Pr[out_0 = 1] = \Pr[\mathsf{real}[\mathcal{Z}, \mathcal{A}, \pi^{\mathcal{G}}] = 1]$ and $\Pr[out_2 = 1] = \Pr[\mathsf{ideal}[\mathcal{Z}, \mathsf{Sim}, \theta^{\mathcal{F}}] = 1]$. Furthermore, $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv^{\mathsf{inv}}_{\mathcal{A},(x_1,x_2)}(\lambda)$ and $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq Adv^{\mathsf{close}}_{\mathcal{A}',(x_1,x_2)}(\lambda)$.

□

```
$\mathbf{G}_0$
$\rho_1, \rho_2 \leftarrow \{0,1\}^{p(\lambda)}$
$z := \mathcal{G}(x_1, \rho_1, x_2, \rho_2)$
upon corruption of $P_1$ do
    give $(x_1, \rho_1, z)$ to $\mathcal{Z}$
upon corruption of $P_2$ do
    give $(x_2, \rho_2)$ to $\mathcal{Z}$
```

```
$\mathbf{G}_1$
$\rho_1 \leftarrow \{0,1\}^{p(\lambda)}$
$z \leftarrow \overline{S}(x_1, x_2)$
upon corruption of $P_1$ do
    give $(x_1, \rho_1, z)$ to $\mathcal{Z}$
upon corruption of $P_2$ do
    $\rho_2 \leftarrow \rho_1 \oplus \overline{S}^{-1}((x_1, x_2), z)$
    give $(x_2, \rho_2)$ to $\mathcal{Z}$
```

```
$\mathbf{G}_2$
$\rho_1 \leftarrow \{0,1\}^{p(\lambda)}$
$z \leftarrow S(x_1, x_2)$
upon corruption of $P_1$ do
    give $(x_1, \rho_1, z)$ to $\mathcal{Z}$
upon corruption of $P_2$ do
    $\rho_2 \leftarrow \rho_1 \oplus \overline{S}^{-1}((x_1, x_2), z)$
    give $(x_2, \rho_2)$ to $\mathcal{Z}$
```

Fig. 12. Hybrids used in proof of Theorem 4.

Assuming $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$, the above strategy can be applied to show that every randomized functionality can be UC-realized in the (global CRS, OT)-hybrid model in the presence of semi-honest adversaries adaptively corrupting any number of parties.

Theorem 5. *Assume* $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *holds, then any functionality* $\mathcal{F}$ *can be UC-realized, in the (global CRS,* $\mathcal{F}_{\mathsf{OT}}$*)-hybrid-model, in the presence of semi-honest adaptive adversaries corrupting any number of parties.*

However, due to [DKR15], it is possible to realize UC-AMPC for every functionality only assuming the static version $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ (at the cost of additionally assuming the existence of adaptively secure oblivious transfer).

5.2 UC-secure AMPC with global setup

We first introduce some necessary preliminaries.

Definition 14 (Garbled circuits, [Yao86; DKR15]). *A garbling scheme consists of two PPT algorithms* $(\mathsf{Gen}_{\mathsf{GC}}, \mathsf{Eval}_{\mathsf{GC}})$*, where* $\mathsf{Eval}_{\mathsf{GC}}$ *is deterministic. Let* C *be a randomized circuit taking* n*-bit inputs and using* k *bits of randomness. On input of* 1^λ *and a circuit* C*,* $\mathsf{Gen}_{\mathsf{GC}}$ *outputs a garbled circuit* GC*,* $2n$ *input-wire labels* $y_{1,0}, y_{1,1}, \ldots, y_{n,0}, y_{n,1} \in \{0,1\}^\lambda$ *and* $2k$ *random-wire labels* $w_{1,0}, w_{1,1}, \ldots, w_{k,0}, w_{k,1} \in \{0,1\}^\lambda$*. On input of a garbled circuit* GC*,* n *input-wire labels* $y_1, \ldots, y_n$ *and* k *random-wire labels* $w_1, \ldots, w_k$*,* $\mathsf{Eval}_{\mathsf{GC}}$ *outputs a value* z*. We require that the garbling scheme satisfies the following two requirements.*

Correctness. *For every* $(\mathsf{GC}, \{y_{i,0}, y_{i,1}\}_{i\in[n]}, \{w_{i,0}, w_{i,1}\}_{i\in[k]}) \in \mathsf{supp}(\mathsf{Gen}_{\mathsf{GC}}(1^\lambda, C))$*, every* $x \in \{0,1\}^n$ *and every* $r \in \{0,1\}^k$*, we have*

$$\mathsf{Eval}_{\mathsf{GC}}(\mathsf{GC}, (y_{i,x_i})_{i\in[n]}, (w_{i,r_i})_{i\in[k]}) = C(x; r).$$

Security. *There exists a PPT algorithm* $\mathsf{Sim}_{\mathsf{GC}}$*, such that for all inputs* $x \in \{0,1\}^n$ *and all randomnesses* $r \in \{0,1\}^k$*, the distributions* $\{\mathsf{Sim}_{\mathsf{GC}}(1^\lambda, C, C(x; r))\}$ *and*

$$\{(\mathsf{GC}, \{y_{i,0}, y_{i,1}\}_{i\in[n]}, \{w_{i,0}, w_{i,1}\}_{i\in[k]}) \leftarrow \mathsf{Gen}_{\mathsf{GC}}(1^\lambda, C) \colon (\mathsf{GC}, (y_{i,x_i})_{i\in[n]}, (w_{i,r_i})_{i\in[k]})\}$$

are computationally indistinguishable.

Definition 15 (Adaptively secure two-round OT protocol, [DKR15]). *A* two-round OT protocol Π_{OT} *consists of the three PPT algorithms* $(\mathsf{R}_{\mathsf{OT}}, \mathsf{S}_{\mathsf{OT}}, \mathsf{E}_{\mathsf{OT}})$*.*

$\mathsf{R}_{\mathsf{OT}}(1^\lambda, b; r_R)$ *takes as input a bit* b *and randomness* r_R *and outputs the first protocol message* OT_1*.*

$\mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1); r_S)$ *takes as input the first message* OT_1*, a tuple* $(y_0, y_1) \in (\{0,1\}^\lambda)^2$ *and randomness* r_S *and outputs the second protocol message* OT_2*.*

$\mathsf{E}_{\mathsf{OT}}(\mathsf{OT}_2, b, r_R)$ *takes as input* OT_2 *the bit* b *and the randomness* r_R *and outputs* $y \in \{0,1\}^\lambda$*.*

We require the following properties.

Correctness. *For all $b \in \{0,1\}$, $y_0, y_1 \in \{0,1\}^\lambda$,*

$$\Pr\left[r_R, r_S \leftarrow \{0,1\}^*, \mathsf{OT}_1 := \mathsf{R}_{\mathsf{OT}}(b; r_R), \mathsf{OT}_2 := \mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_0); r_S) \colon \mathsf{E}_{\mathsf{OT}}(\mathsf{OT}_2, b, r_b) = y_b\right]$$

is overwhelming.

Security. *There exists an efficient simulator* $\mathsf{SimOT} = (\mathsf{SimOT}_1, \mathsf{SimOT}_2)$*, where* SimOT_2 *is deterministic, such that the following distributions are computationally indistinguishable.*

$$\{r_R, r_S \leftarrow \{0,1\}^*, \mathsf{OT}_1 := \mathsf{R}_{\mathsf{OT}}(b; r_R) \colon (r_R, \mathsf{OT}_1, \mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1); r_S))\}$$

$$\{(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{state}) \leftarrow \mathsf{SimOT}_1(1^\lambda), r_R := \mathsf{SimOT}_2(b, y_b, \mathsf{state}) \colon (r_R, \mathsf{OT}_1, \mathsf{OT}_2)\}$$

A suitable instantiation can be found in [CLOS02].

As in [DKR15], we will use the following property of Π_{OT}. Let SimOT'_1 be the algorithm that runs SimOT_1 and only outputs $(\mathsf{OT}_1, \mathsf{state})$. Let further SimOT'_2 be the algorithm that takes b and state as input, runs $\mathsf{SimOT}_2(b, 0^\lambda, \mathsf{state})$ and outputs the resulting r_R. Security of Π_{OT} implies that the following distributions are computationally indistinguishable for all $b \in \{0,1\}$.

$$\{r_R \leftarrow \{0,1\}^* \colon (r_R, \mathsf{R}_{\mathsf{OT}}(b; r_R))\}$$

$$\{(\mathsf{OT}_1, \mathsf{state}) \leftarrow \mathsf{SimOT}'_1(1^\lambda), r_R := \mathsf{SimOT}'_2(1^\lambda, b, \mathsf{state}) \colon (r_R, \mathsf{OT}_1)\}$$

$\mathsf{cISH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ corresponds to the existence of an explainability compiler [DKR15] with computational closeness and, due to Theorem 1, is equivalent to $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$. Computational closeness suffices to obtain UC-secure AMPC for all PPT functionalities with global setup by the same arguments as in [DKR15]. Since Theorem 6 is crucial for Theorem 2, we sketch the proof techniques from [DKR15] and emphasize why static $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with computational closeness suffices.

Let C be a randomized circuit describing the functionality we want to UC-realize in the global CRS model in the presence of semi-honest adaptive adversaries. Let $P_1, \dots, P_n$ denote the involved parties.

```
$S(\{\mathsf{OT}_{1,i}\}_i; (r_{\mathsf{GC}}, \{r_{S,i}\}_i, (r_i)_i))$
$(\mathsf{GC}, \{y_{i,b}\}_{i,b}, \{w_{i,b}\}_{i,b}) \leftarrow \mathsf{Gen}_{\mathsf{GC}}(C; r_{\mathsf{GC}})$
$\mathsf{OT}_{2,i} \leftarrow \mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_{1,i}, (y_{i,0}, y_{i,1}); r_{S,i})$
return $(\mathsf{GC}, \{w_{i,r_i}\}_i, \{\mathsf{OT}_{2,i}\}_i)$
```

```
Protocol $\Pi$
Common input: $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
Input of party $P_i$: $x_i$
1 :  Each $P_i$: $\mathsf{OT}_{1,i} := \mathsf{R}_{\mathsf{OT}}(x_i; r_{R,i})$
     each $P_i$ sends $\mathsf{OT}_{1,i}$ to $P_n$
2 :  $P_n$: $(\mathsf{GC}, \{\mathsf{OT}_{2_i}\}_i, \{w_i\}_i) := \overline{S}(crs, \{\mathsf{OT}_{1,i}\}_i; r_n)$
     $P_n$ sends $\mathsf{OT}_{2,i}$ to each $P_i$
3 :  Each $P_i$: $y_i \leftarrow \mathsf{E}_{\mathsf{OT}}(x_i, r_{R,i}, \mathsf{OT}_{2_i})$
     each $P_i$ sends $y_i$ to $P_n$ (secure channel)
4 :  $P_n$: $z := \mathsf{Eval}_{\mathsf{GC}}(\mathsf{GC}, \{y_i\}_i, \{w_i\}_i)$
     $P_n$ sends $z$ to each $P_i$ (secure channel)
```

Fig. 13. Sampler algorithm S (left) and description of protocol Π (right), [DKR15].

Theorem 6 ([DKR15]). *Assume* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ *holds,* $\mathsf{Gen}_{\mathsf{GC}}$ *is a secure garbling scheme and* Π_{OT} *is a semi-honest adaptively secure two-round OT protocol. Then, for every probabilistic functionality C, there exists a protocol Π (cf. Figure 13) which UC-realizes C in the global CRS model in the presence of semi-honest adaptive adversaries corrupting any number of parties.*

Proof. We sketch the proof strategy here. The proof slightly differs from [DKR15] since $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ only guarantees computational closeness as opposed to statistical closeness provided by the explainability compiler.

Let $\mathcal{Z}$ be the environment and let $\mathcal{A}$ be the dummy adversary. We assume the global (alias non-programmable) CRS model, i.e. each party including the environment $\mathcal{Z}$, the adversary $\mathcal{A}$ and the simulator Sim have access to a common reference string sampled according to the distribution Setup_S. In contrast to a local CRS, the CRS is always an external input to the parties. Hence, not even the simulator can manipulate its distribution nor knows a corresponding trapdoor (e.g. the used coins). We assume (without loss of generality) that $\mathcal{Z}$ first observes the entire protocol before corrupting parties.

```
$\mathbf{G}_0$
$x_1, \dots, x_n \leftarrow \mathcal{Z}(crs)$
$\mathsf{OT}_{1,i} \leftarrow \mathsf{R}_{\mathsf{OT}}(x_i; r_{R,i})$
    1st round messages: $\{\mathsf{OT}_{1,i}\}_{i \in [n-1]}$
$(\mathsf{GC}, \{w_{i,r_i}\}_i, \{\mathsf{OT}_{2,i}\}_i) \leftarrow \overline{S}(crs, \{\mathsf{OT}_{1,i}\}_i; r_n)$
    2nd round messages: $\{\mathsf{OT}_{2,i}\}_{i \in [n-1]}$
upon corruption of $P_i$ do
    if $i \neq n$ then
        $y_i \leftarrow \mathsf{E}_{\mathsf{OT}}(x_i, r_{R,i}, \mathsf{OT}_{2,i})$
        give $(x_i, z, y_i, r_{R,i})$ to $\mathcal{Z}$
    if $i = n$ then
        $y_j \leftarrow \mathsf{E}_{\mathsf{OT}}(x_j, r_{R,j}, \mathsf{OT}_{2,j})$
        give $(\{(x_j, z, y_j, r_{R,j})\}_{j \in [n]}, r_n)$ to $\mathcal{Z}$
```

```
$\mathsf{Sim}(crs, z)$
$(\mathsf{OT}_{1,i}, \mathsf{OT}_{2,i}, \mathsf{state}_i) \leftarrow \mathsf{SimOT}_1(1^\lambda)$
    1st round messages: $\{\mathsf{OT}_{1,i}\}_{i \in [n-1]}$
    2nd round messages: $\{\mathsf{OT}_{2,i}\}_{i \in [n-1]}$
$count := 0$
upon corruption of $P_i$ do
    receive $x_i, z$
    if $count = 0$ then
        $(\mathsf{GC}, \{y_i\}_i, \{w_i\}_i) \leftarrow \mathsf{Sim}_{\mathsf{GC}}(C, z)$
        $r_n^* \leftarrow \overline{S}^{-1}(crs, (\{\mathsf{OT}_{1,i}\}_i, (\mathsf{GC}, \{w_{i,r_i}\}_i, \{\mathsf{OT}_{2,i}\}_i)))$
    $r_{R,i} \leftarrow \mathsf{SimOT}_2(x_i, y_i, \mathsf{state}_i)$
    give $(x_i, z, y_i, r_{R,i})$ to $\mathcal{Z}$
    $count$++
```

Fig. 14. Definition of game $\mathbf{G}_0$ describing the view of $\mathcal{Z}$ in the real world $\mathsf{real}[\mathcal{Z}, \mathcal{A}, \Pi]$ and of the corresponding simulator Sim, [DKR15].

To prove security, we proceed over a series of hybrids as follows.

Game $\mathbf{G}_0$. See Figure 14 for a complete description. $\mathbf{G}_0$ describes the view of $\mathcal{Z}$ in the real world.

Game $\mathbf{G}_1$. Initially, we modify $\mathbf{G}_0$ such that the oblivious transfer messages of round 1 are simulated via $\mathsf{SimOT}_1'(1^\lambda)$. The randomness $r_{R,i}$ is produced via $\mathsf{SimOT}_2'(1^\lambda, x_i, \mathsf{state}_i)$. Due to the security of the OT protocol, this change is indistinguishable. Furthermore, due to this change, the actual inputs x_i of the parties are not used before corruption of those parties.

Game $\mathbf{G}_2$. Next, we change game $\mathbf{G}_1$ to inverse sample random coins r_n^* for $\overline{S}$ and give those to the environment on corruption of party P_n (instead of giving the random coins r_n actually used by $\overline{S}$). Since the input to $\overline{S}$ (and $\overline{S}^{-1}$) is independent of the inputs x_i chosen by $\mathcal{Z}$ after seeing crs, the static version of invertibility suffices.

Game $\mathbf{G}_3$. Since the actual coins used by $\overline{S}$ are not visible to $\mathcal{Z}$ anymore, we are able to apply closeness. I.e. we change $\mathbf{G}_2$ to use S instead of using $\overline{S}$. Again, since the input to $\overline{S}$ (and S) is independent of the inputs x_i chosen by $\mathcal{Z}$ after seeing crs, the static version of closeness suffices.

Game $\mathbf{G}_4$. After unwrapping the definition of S, we define $y_i := y_{i,x_i}$ (the labels generated by $\mathsf{Gen}_{\mathsf{GC}}$ in S) instead of computing y_i via $\mathsf{E}_{\mathsf{OT}}(x_i, r_{R,i}, \mathsf{OT}_{2,i})$. Indistinguishability follows from the correctness (and security) of the OT protocol.

Game $\mathbf{G}_5$. Next, we simulate both rounds of the OT messages via $\mathsf{SimOT}_1(1^\lambda)$. Indistinguishability again follows from security of the OT protocol.

Game $\mathbf{G}_6$. Finally, we generate $(\mathsf{GC}, \{y_{i,x_i}\}_i, \{w_{i,r_i}\}_i)$ via $\mathsf{Sim}_{\mathsf{GC}}(C, C(x; r))$. Indistinguishability follows from the security of the garbling scheme. Hence, the view of $\mathcal{Z}$ is identical to the view produced by $\mathsf{Sim}(crs, z)$, see Figure 14.

□

Remark 8. By viewing each party as polynomially many sub-parties, Theorem 6 can be bootstrapped to allow for functionalities which take polynomially long inputs from each party.

Remark 9. In Theorem 6, the requirement for an adaptively secure two-round OT protocol can be relaxed to an adaptively secure two-round OT protocol in the global CRS model. Due to [CPR16], adaptively secure two-round OT in the global CRS model exists based on IO and one-way functions.

5.3 AMPC with global setup implies adaptive pseudorandom encodings with setup

In the following we consider the standalone model [Can00]. Note that in the standalone model, the environment is strictly weaker than in the UC-framework.

Theorem 7. *If for all (two party) PPT functionalities $\mathcal{F}$, there exists a protocol Π that securely implements $\mathcal{F}$ in the global CRS model in the presence of adaptive semi-honest adversaries with post-execution corruption corrupting any number of parties, then* $\mathsf{acISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ *(without universal setup) is true.*

Proof. The proof follows the ideas of [IKOS10]. Let S be an arbitrary PPT algorithm. Let x_1 denote the input of party P_1 and x_2 denote the input of party P_2.

Consider the randomized functionality $\mathcal{F}$ which computes $z := S(x_1, x_2; \rho)$ and outputs z to P_1 and $\bot$ to P_2, where ρ is the internal random coins of the functionality. We denote the message space of S by L. Let Π be a protocol which securely realizes $\mathcal{F}$ in the global CRS model (where the common reference string is drawn according some distribution $\mathsf{Setup}_\mathcal{F}$ possibly depending on the functionality $\mathcal{F}$) in the presence of adaptive semi-honest adversaries (with post-execution corruption). Further, let p_1 and p_2 be stateful PPT algorithms modeling the protocol messages exchanged between the parties P_1 and P_2. More formally, $(m_{2i+1}, z_i) \leftarrow p_1(crs, x_1, m_{2i}; r_1)$ and $m_{2j} \leftarrow p_2(crs, x_2, m_{2j-1}; r_2)$ for $i \geq 0, j \geq 1$ and $r_1, r_2 \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}$ (for some sufficiently large polynomial poly). (Without loss of generality, we let p_1 additionally output z_i which equals $\bot$ during the protocol execution and contains the output of party P_1 after the execution.)

```
$\overline{S}(crs, (x_1, x_2); (r_1, r_2))$
$z := \bot, m := \bot$
while $z = \bot$ do
    $(m, z) \leftarrow p_1(crs, x_1, m, r_1)$
    $m \leftarrow p_2(crs, x_2, m, r_2)$
return $z$
```

```
$\overline{S}^{-1}(crs, (x_1, x_2), z; r_S)$
$\{m'_i\}_{i \in [poly(\lambda)]} \leftarrow \mathsf{Sim}_1(crs; r_S)$
$(z, r'_1) \leftarrow \mathsf{Sim}_2(x_1, z)$
$r'_2 \leftarrow \mathsf{Sim}_3(x_2)$
return $(r'_1, r'_2)$
```

Fig. 15. Definition of the alternative sampler $\overline{S}$ (left) and the corresponding inverse sampler $\overline{S}^{-1}$ (right).

Lemma 6. *The setup algorithm* $\mathsf{Setup}_S := \mathsf{Setup}_\mathcal{F}$ *together with the alternative sampler* $\overline{S}$ *defined in Figure 15 satisfy adaptive closeness.*

Proof. Since we only consider protocols Π with at most polynomially many rounds, $\overline{S}$ is a PPT algorithm.

Intuitively, the output distributions of S and $\overline{S}$ are computationally close because of "correctness" of the protocol Π.

Let $\mathcal{B}$ be the adversary on the protocol Π which only observes the protocol and does not interfere at all. By Definition 12, there exists a simulator Sim such that for all environments $\mathcal{Z}$, $|\Pr[crs \leftarrow \mathsf{Setup}_\mathcal{F}(1^\lambda) \colon \mathcal{Z}^{\mathcal{A},\Pi}(crs) = 1] - \Pr[crs \leftarrow \mathsf{Setup}_\mathcal{F}(1^\lambda) \colon \mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(crs) = 1]|$ is negligible. Note that since $\mathcal{B}$ corrupts no parties, Sim also corrupts no parties.

Let $\mathcal{A}$ be an adversary on closeness. We construct an environment $\mathcal{Z}$ which distinguishes between the real and the ideal world as follows. Initially, $\mathcal{Z}$ receives crs as input, calls $\mathcal{A}$ on input of crs and obtains $(x_1, x_2) \in L$. $\mathcal{Z}$ uses x_1 as input for party P_1 and x_2 as input for P_2. Then, $\mathcal{Z}$ executes the protocol (without any interference) and finally receives the output $(z, \perp)$. $\mathcal{Z}$ calls $\mathcal{A}$ on input z and outputs the $\mathcal{A}$'s output.

In the ideal world, $\mathcal{Z}$ receives the outputs z and $\perp$ for P_1 and P_2 from the trusted third party T (since no party is corrupted) which are computed by $\mathcal{F}$. Hence, if $\mathcal{Z}$ is in the ideal world, it simulates $Exp_{\mathcal{A},0}^{\mathsf{a\text{-}crs\text{-}close}}$. In the real world, $\mathcal{Z}$ interacts with the actual parties running protocol Π and, hence, simulates $Exp_{\mathcal{A},1}^{\mathsf{a\text{-}crs\text{-}close}}$. Therefore, $Adv_{\mathcal{A}}^{\mathsf{a\text{-}crs\text{-}close}}(\lambda) \leq |\Pr[crs \leftarrow \mathsf{Setup}_{\mathcal{F}}(1^\lambda) \colon \mathcal{Z}^{\mathcal{A},\Pi}(crs) = 1] - \Pr[crs \leftarrow \mathsf{Setup}_{\mathcal{F}}(1^\lambda) \colon \mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(crs) = 1]|$ for the environment $\mathcal{Z}$ and the adversary $\mathcal{B}$ which do not interfere with the protocol execution. □

Let $\mathcal{B}$ be an adversary that operates as follows. During the protocol, $\mathcal{B}$ records the transcripts $\{m_i\}_{i \in [\mathsf{poly}'(\lambda)]}$ and outputs them. At the end of the protocol, $\mathcal{B}$ corrupts P_1 obtaining the input x_1, output z and random tape r_1. $\mathcal{B}$ directly outputs (z, r_1). Afterwards, $\mathcal{B}$ (post-execution) corrupts P_2 and obtains the input x_2 and the random tape r_2 and outputs r_2.

Since we assume Π securely realizes $\mathcal{F}$ (in particular in the presence of $\mathcal{B}$), there exists a (stateful) simulator $\mathsf{Sim} := (\mathsf{Sim}_1, \mathsf{Sim}_2, \mathsf{Sim}_3)$ producing outputs that are computationally indistinguishable from the outputs $\mathcal{B}$ produces. Sim_1 simulates the execution before a corruption occurs. On input of crs and a random tape r_S, Sim_1 produces $\{m'_i\}_{i \in [\mathsf{poly}'(\lambda)]}$. On corruption of P_1 (in the post-execution corruption stage), Sim_2 obtains x_1, z and outputs a string r'_1 corresponding to the randomness that P_1 used to produce its messages in the protocol (using input x_1) and outputs (z, r'_1). On corruption of P_2 (in the post-execution corruption stage), Sim_3 obtains x_2 and outputs a string r'_2 corresponding to the randomness that P_2 used to produce its messages (using input x_2).

Lemma 7. *The setup algorithm* $\mathsf{Setup}_S := \mathsf{Setup}_{\mathcal{F}}$ *together with the alternative sampler* $\overline{S}$ *and the inverse sampler* $\overline{S}^{-1}$ *defined in Figure 15 satisfy adaptive invertibility.*

Proof. Since $\mathsf{Sim}_1, \mathsf{Sim}_2, \mathsf{Sim}_3$ are PPT algorithms, $\overline{S}^{-1}$ is a PPT algorithm.

Recall that $\mathcal{F}(x_1, x_2; \rho) := S(x_1, x_2; \rho)$. Let $\mathcal{F}'(x_1, x_2; \rho') := \overline{S}(x_1, x_2; \rho')$. Intuitively, we consider an intermediate game, where $\mathcal{Z}$ interacts with the simulator Sim and the modified ideal functionality $\mathcal{F}'$.

Claim. For all PPT environments $\mathcal{Z}$,

$$\left|\Pr[crs \leftarrow \mathsf{Setup}_{\mathcal{F}}(1^\lambda) \colon \mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(crs) = 1] - \Pr[crs \leftarrow \mathsf{Setup}_{\mathcal{F}}(1^\lambda) \colon \mathcal{Z}^{\mathsf{Sim},\mathcal{F}'}(crs) = 1]\right| \leq \mathsf{negl}(\lambda). \tag{6}$$

Proof. Let $\mathcal{Z}$ be a PPT environment. We construct an adversary $\mathcal{B}$ on adaptive closeness. $\mathcal{B}$ simulates the entire interaction between the $\mathcal{Z}$, the simulator Sim and the trusted party T, which evaluates the functionality $\mathcal{F}$ (or $\mathcal{F}'$). In particular, $\mathcal{B}$ takes control of the party T. On input of crs, $\mathcal{B}$ simulates the interaction between $\mathcal{Z}$ and Sim (given crs) until the computation stage. In the computation stage, the parties P_1 and P_2 send their inputs (which may have been adaptively chosen based on crs) to $\mathcal{B}$. $\mathcal{B}$ outputs $(x_1, x_2) \in L$ to $Exp_{\mathcal{A},b}^{\mathsf{a\text{-}crs\text{-}close}}$ and receives an output $z := y_b$, where $y_0 \leftarrow S(x_1, x_2)$ and $y_1 \leftarrow \overline{S}(x_1, x_2)$. $\mathcal{B}$ sends z to P_1 and P_2 and continues to simulate the interaction between $\mathcal{Z}$ and Sim. This is possible, because the randomness which is actually used by T is not never needs to be known. Finally, $\mathcal{Z}$ halts and outputs a bit b'. $\mathcal{B}$ outputs b'. □

Let $\mathcal{A}$ be an adversary on adaptive invertibility. We construct an environment $\mathcal{Z}$ that distinguishes between the real execution of Π with adversary $\mathcal{B}$ and the ideal execution with Sim and the ideal functionality $\mathcal{F}'$. Initially, $\mathcal{Z}$ receives crs and calls $\mathcal{A}$ on input of crs to obtain $(x_1, x_2) \in L$. $\mathcal{Z}$ uses x_1 as input for P_1 and x_2 as input for P_2. After the execution of the protocol and the post-execution corruptions, $\mathcal{Z}$ receives $(r_1, r_2), z$ (either by the adversary $\mathcal{B}$ or the simulator Sim). Finally, $\mathcal{Z}$ calls $\mathcal{A}$ on input of $((r_1, r_2), z)$ and outputs $\mathcal{A}'s$ output.

In the real world, z is the output of the real protocol and r_1, r_2 is the actual randomness used by the parties. Hence, by definition of $\overline{S}$, (r_1, r_2) is the randomness actually used by $\overline{S}$ to produce the output z. Therefore, $\Pr[\mathit{Exp}^{\textsf{a-crs-inv}}_{\mathcal{A},0}(\lambda) = 1] = \Pr[\mathit{crs} \leftarrow \mathsf{Setup}_{\mathcal{F}}(1^\lambda) \colon \mathcal{Z}^{\mathcal{B},\Pi}(\mathit{crs}) = 1]$.

In the ideal world, z is produced by the functionality $\mathcal{F}'$ (hence, by $\overline{S}$) and $(r_1, r_2) := (r_1', r_2')$ is produced by $\overline{S}^{-1}((x_1, x_2), z)$. Hence, $\Pr[\mathit{Exp}^{\textsf{a-crs-inv}}_{\mathcal{A},1}(\lambda) = 1] = \Pr[\mathit{crs} \leftarrow \mathsf{Setup}_{\mathcal{F}} \colon \mathcal{Z}^{\mathsf{Sim},\mathcal{F}'}(\mathit{crs}) = 1]$. Hence,

$$\begin{aligned} \mathit{Adv}^{\textsf{a-crs-inv}}_{\mathcal{A}}(\lambda) &= \left|\Pr_{\mathit{crs}}[\mathcal{Z}^{\mathcal{B},\Pi}(\mathit{crs}) = 1] - \Pr_{\mathit{crs}}[\mathcal{Z}^{\mathsf{Sim},\mathcal{F}'}(\mathit{crs}) = 1]\right| \\ &\le \left|\Pr_{\mathit{crs}}[\mathcal{Z}^{\mathcal{B},\Pi}(\mathit{crs}) = 1] - \Pr_{\mathit{crs}}[\mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(\mathit{crs}) = 1]\right| \\ &\quad + \left|\Pr_{\mathit{crs}}[\mathcal{Z}^{\mathsf{Sim},\mathcal{F}}(\mathit{crs}) = 1] - \Pr_{\mathit{crs}}[\mathcal{Z}^{\mathsf{Sim},\mathcal{F}'}(\mathit{crs}) = 1]\right| \end{aligned}$$

which is negligible by assumption and Equation (6). □

Therefore, for every PPT algorithm S, there exists an algorithm Setup_S, an alternative sampler $\overline{S}$ and an inverse sampler $\overline{S}^{-1}$ such that adaptive closeness and adaptive invertibility hold. This concludes the proof. □

Combining Theorems 6 and 7 proves Theorem 2. This yields the first instantiation of an *adaptive* explainability compiler [DKR15] without complexity leveraging and, hence, based only on polynomial hardness assumptions. The recent paper [CsW19] uses such an adaptive explainability compiler to obtain adaptive MPC with communication complexity sub-linear circuit size. Their construction relies on complexity leveraging which entails a sub-exponential loss relative to IO and one-way functions. Hence, we obtain the following corollary improving on [CsW19] in a black-box way.

Corollary 1. *Assuming polynomially secure IO and the adaptive hardness of LWE, then succinct adaptive MPC is possible in the global CRS model.*

6 Classification of the different flavors of pseudorandom encodings

In this section we classify the different variants of the pseudorandom encoding hypothesis. In Section 6.1, we study the pseudorandom encoding hypothesis with deterministic encoding algorithm and identify a relation to compression. In Section 6.2, we study the pseudorandom encoding hypothesis with randomized encoding and its conflicts with extractable one-way functions. In Section 6.3, we describe an instantiation of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ (with universal setup) based on indistinguishability obfuscation and one-way functions due to [SW14; DKR15]. Finally, in Section 6.4, we bootstrap $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ with a common *random* string from the construction in Section 6.3 additionally assuming *weak* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ with a common *random* string.

6.1 Deterministic encoding

For the purpose of classifying pseudorandom encodings with a deterministic encoding algorithm, we first introduce some notions of entropy and computational analogues thereof.

Definition 16 (Min-entropy, [Rey11]). *The min-entropy of a distribution X is defined as $\mathrm{H}_\infty(X) = -\log \max_{x \in \mathsf{supp}(X)} \Pr[X = x]$.*

(By log we always mean the logarithm to base 2.) Intuitively, this captures the ability to guess the value of a distribution in a single attempt. However, this is a very pessimistic view. For many purposes, it suffices to work with a distribution with is statistically close (i.e. has negligible statistical distance) to a distribution with high min-entropy.[11]

[11] We note that if some distribution Y is ϵ-close to a distribution Z, $\mathrm{H}_\infty(Y) \ge -\log(2^{-\mathrm{H}_\infty(Z)} + \epsilon(\lambda)) = \mathrm{H}_\infty(Z) - \log(1 + \epsilon(\lambda)2^{\mathrm{H}_\infty(Z)})$.

Definition 17 (ϵ-smooth min-entropy, [Rey11]). *A source X has ϵ-smooth min-entropy at least k, denoted as $\mathrm{H}_\infty^\epsilon(X) \geq k$, if there exists a distribution Y with $\Delta(X, Y) \leq \epsilon$ such that $\mathrm{H}_\infty(Y) \geq k$.*

In many cases, some information Z that is correlated to the actual source X is known. Since for our purposes, the conditional part Z is not under adversarial control, we use the notion of *average* conditional min-entropy as used in [HLR07; DORS08].

Definition 18 (Average min-entropy, [HLR07; DORS08; Rey11]). *Let (Y, Z) be a joint distribution. The average min-entropy of X conditioned on Z is*

$$\widetilde{\mathrm{H}}_\infty(X \mid Z) := -\log(\mathop{\mathbb{E}}_{z \leftarrow Z}[\max_x \Pr[X_z = x]]).$$

Definition 19 (Average ϵ-smooth min-entropy, [DORS08; Rey11]). *Let (Y, Z) be a joint distribution. The distribution Y has average ϵ-smooth min-entropy at least k conditioned on Z, denoted as $\widetilde{\mathrm{H}}_\infty^\epsilon(Y \mid Z) \geq k$, if there exists a joint distribution (Y', Z') with $\Delta((Y, Z), (Y', Z')) \leq \epsilon$ such that $\widetilde{\mathrm{H}}_\infty(Y' \mid Z') \geq k$.*

Let X be an efficiently samplable distribution. In the literature, there are several computational notions of entropy. HILL entropy [HILL99] constitutes a natural computational variant of min-entropy. A source has high HILL entropy, if it is computationally indistinguishable from a source that has high min-entropy.

Definition 20 (HILL entropy, [HILL99; BSW03]). *A distribution X has* HILL entropy *at least k, denoted by $\mathrm{H}_{\epsilon,s}^{\mathsf{HILL}}(X) \geq k$, if there exists a distribution Y such that $\mathrm{H}_\infty(Y) \geq k$ and $|\Pr[x \leftarrow X \colon \mathcal{A}(x) = 1] - \Pr[y \leftarrow Y \colon \mathcal{A}(y) = 1]| \leq \epsilon$ for all $\mathcal{A}$ of size at most s.*

Shannon's theorem [Sha48] states that the minimum compression length (over all compression and decompression algorithms) of a distribution equals its average entropy (up to small additive terms). Yao entropy [Yao82] constitutes the corresponding computational counterpart. Intuitively, a source has high Yao entropy, if it can not be *efficiently* compressed.

Definition 21 (Yao entropy, [Yao82; BSW03]). *A distribution X has* Yao entropy *at least k, denoted by $\mathrm{H}_{\epsilon,s}^{\mathsf{Yao}}(X) \geq k$, if for every pair of circuits (E, D) of total size s with outputs of E having length ℓ, $\Pr_{x \leftarrow X}[\mathsf{D}(\mathsf{E}(x)) = x] \leq 2^{\ell - k} + \epsilon$.*

When we omit the subscripts for $\mathrm{H}^{\mathsf{HILL}}$ and $\mathrm{H}^{\mathsf{Yao}}$, we mean $\mathrm{H}_{\epsilon,s}^{\mathsf{HILL}}$ and $\mathrm{H}_{\epsilon,s}^{\mathsf{Yao}}$ for any negligible ϵ and polynomial s, respectively. Since compressibility implies distinguishability, HILL entropy implies Yao entropy. The converse, however, is believed to be false, [Wee04; HLR07].

Definition 22 (Conditional HILL entropy, [HLR07]). *For a distribution (X, Z), we say that X has HILL entropy at least k conditioned on Z, denoted by $\mathrm{H}_{\epsilon,s}^{\mathsf{HILL}}(X \mid Z) \geq k$, if there exists a collection of distributions Y_z giving rise to a joint distribution (Y, Z), such that $\widetilde{\mathrm{H}}_\infty(Y \mid Z) \geq k$ and $|\Pr[(x, z) \leftarrow (X, Z) \colon \mathcal{A}(x, z) = 1] - \Pr[(y, z) \leftarrow (Y, Z) \colon \mathcal{A}(y, z) = 1]| \leq \epsilon$ for all circuits $\mathcal{A}$ of size at most s.*

Conditional Yao entropy is defined by simply giving the compressor and decompressor algorithm the value z as input.

Definition 23 (Conditional Yao entropy, [HLR07]). *For a distribution (X, Z), we say that X has Yao entropy at least k conditioned on Z, denoted by $\mathrm{H}_{\epsilon,s}^{\mathsf{Yao}}(X \mid Z) \geq k$, if for every pair of circuits (E, D) of total size at most s with outputs of E having length ℓ, $\Pr_{(x,z) \leftarrow (X,Z)}[\mathsf{D}(z, \mathsf{E}(z, x)) = x] \leq 2^{\ell - k} + \epsilon$.*

6.1.1 Information theoretic guarantees and compression

Traditionally, the theory of compression mostly considers family of sources that are not indexed by strings. Let Σ be some alphabet. In this work, we always consider $\Sigma := \{0,1\}$.

Definition 24 ([Wee04; TVZ05]). *For functions* $\mathsf{E}_: \Sigma^* \to \Sigma^*$ *and* $\mathsf{D}_: \Sigma^* \to \Sigma^*$*, we say* $(\mathsf{E}_X, \mathsf{D}_X)$ compresses source X to length m with decoding error ϵ *if*

1. $\Pr[x \leftarrow X \colon \mathsf{D}_X(\mathsf{E}_X(x)) \neq x] \leq \epsilon$*, and*
2. $\mathbb{E}[|\mathsf{E}_X(X)|] \leq m$*.*

Definition 25 ([Wee04; TVZ05]). *We say source* X *is* compressible to length (exactly) m *if there exist functions* E_X *and* D_X *such that* $(\mathsf{E}_X, \mathsf{D}_X)$ *compresses* X *to length (exactly)* m*.*

Lemma 8 ([TVZ05]). *Let* X_λ *be a source on* $\{0,1\}^\lambda$ *which is compressible to length* m *with decoding error* ϵ *by algorithms* $(\mathsf{E}_X, \mathsf{D}_X)$*. Further, let* $m_0 \in \mathbb{N}$ *be a lower bound on the output length of* E_X*, i.e. such that for all* $x \in \mathsf{supp}(X_\lambda)$*,* $|\mathsf{E}_X(x)| \geq m_0$*. Then,* X_λ *is compressible to length* $m + \epsilon(\lambda - m_0) + 1$ *with decoding error* 0*.*

Proof (sketch). To show this, [TVZ05] construct an encoding algorithm E'_X which on input of x tests if the $\mathsf{D}_X(\mathsf{E}_X(x)) = x$. If this is the case, E'_X outputs $0 \parallel \mathsf{E}_X(x)$. Else, E'_X outputs $1 \parallel x$. □

The statistical deterministic variant of the pseudorandom encoding hypothesis is strongly related to compression.

Theorem 8. *If (weak)* $\mathsf{PREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$ *is true for* X*, i.e. there exist deterministic algorithms* $(\mathsf{E}_X, \mathsf{D}_X)$ *with* E_X *having output length* n *satisfying correctness and pseudorandomness. Then, the* ϵ*-smooth min-entropy* $\mathrm{H}^\epsilon_\infty(X) \geq n$ *for some negligible function* ϵ*.*

Proof. Consider the distribution $Y' := \mathsf{D}_X(U_n)$. As already seen in the proof of Theorem 1, Y' and X are statistically indistinguishable due to correctness and pseudorandomness. Hence, the statistical distance $\Delta(X, Y') \leq \delta(\lambda)$ for some negligible δ.

Due to correctness and since E_X is deterministic, $\Pr[x \leftarrow X \colon \mathsf{D}_X(\mathsf{E}_X(x)) = x] = \Pr[x \leftarrow X \colon \mathsf{E}_X(\mathsf{D}_X(\mathsf{E}_X(x))) = \mathsf{E}_X(x)] \geq 1 - \nu(\lambda)$ for some negligible function ν. Applying pseudorandomness, we get that the probability

$$\Pr\left[u \leftarrow U_n \colon \mathsf{E}_X(\mathsf{D}_X(u)) = u\right] \geq 1 - \nu'(\lambda)$$

for some negligible function ν', where the probability is only over the choice of u. Therefore, D_X operates almost injectively on the set $\{0,1\}^n$. More formally, let $V_0 \subset \{0,1\}^n$ denote the set of all u such that $\mathsf{E}_X(\mathsf{D}_X(u)) = u$ and let $V_1 := \{0,1\}^n \setminus V_0$. We have that, $|V_1|/|\{0,1\}^n| \leq \nu'(\lambda)$. Let $\overline{V_1}$ be some arbitrary subset of $\{0,1\}^{p(\lambda)} \setminus \mathsf{D}_X(V_0)$ such that $|\overline{V_1}| = |V_1|$ (for some polynomial p). Let Y'' be the uniform distribution over $\mathsf{D}_X(V_0) \cup \overline{V_1}$. Note that $|\mathsf{D}_X(V_0) \cup \overline{V_1}| = 2^n$ and $\Pr[Y' \in \overline{V_1}] \leq \Pr[Y' \notin \mathsf{D}_X(V_0)] \leq \nu'(\lambda)$. The statistical distance between Y' and Y'' is negligible.

$$\begin{aligned}\Delta(Y', Y'') &= \sum_{a \in \mathsf{D}_X(V_0) \cup \overline{V_1}} \left|\Pr[Y' = a] - \Pr[Y'' = a]\right| \\ &= \underbrace{\sum_{a \in \mathsf{D}_X(V_0)} \left|\Pr[Y' = a] - \Pr[Y'' = a]\right|}_{\leq 1 - |V_0| \cdot 2^{-n} \leq 1 - (1 - \nu'(\lambda))} + \underbrace{\sum_{a \in \overline{V_1}} \left|\Pr[Y' = a] - \Pr[Y'' = a]\right|}_{\leq \sum_{a \in \overline{V_1}} (\Pr[Y' = a] + \Pr[Y'' = a]) \leq 2 \cdot \nu'(\lambda)} \leq 3 \cdot \nu'(\lambda)\end{aligned}$$

Furthermore, since Y'' is the uniform distribution over a set of size 2^n, its min-entropy equals n. Since $\Delta(X, Y') \leq \delta(\lambda)$ and $\Delta(Y', Y'') \leq 3\nu'(\lambda)$ for negligible functions δ and ν', we have $\mathrm{H}^{\delta + 3\nu'}_\infty(X) \geq \mathrm{H}_\infty(Y'') = n$. □

Corollary 2. *If* $\mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ *is true for some source* X*, there exists an* $n \leq \mathrm{H}^{\epsilon}_{\infty}(X)$ *(for some negligible function* ϵ*), such that* X *is compressible to length exactly* n*. The decoding error can then be eliminated applying Lemma 8.*

Hence, for $\mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ to be true for a source X, it is necessary that X is efficiently compressible. However, this is not a sufficient criterion since a compression algorithm can have some structure which makes it easily distinguishable from uniform randomness (e.g. the compression algorithm used in the proof of Lemma 8).

The distribution induced by pseudorandom generators can provably not be compressed. However, those distributions have low ϵ-smooth min-entropy.

Lemma 9. *Let* iPRG *be an injective pseudorandom generator with polynomial stretch* $\mathsf{poly}(\cdot)$ *and let* ϵ *be a negligible function. Then,* $\mathrm{H}^{\epsilon}_{\infty}(\mathsf{iPRG}(U_\lambda)) \leq \lambda + \delta(\lambda)$ *for some negligible function* δ*.*

Proof. Since $\mathsf{iPRG}(U_\lambda)$ is a uniform distribution over $\mathsf{iPRG}(\{0,1\}^\lambda)$, it suffices to consider all ϵ-close flat distributions over $\{0,1\}^{\mathsf{poly}(\lambda)}$ to obtain an upper bound on $\mathrm{H}^{\epsilon}_{\infty}(\mathsf{iPRG}(U_\lambda))$. Let D the uniform distribution on $\mathsf{supp}(\mathsf{iPRG}(U_\lambda)) \cup A$, where $\mathsf{supp}(\mathsf{iPRG}(U_\lambda)) \cap A = \varnothing$. Let $|A| := k$. Then,

$$\begin{aligned}\Delta(\mathsf{iPRG}(U_\lambda), D) &= \sum_{a \in \mathsf{supp}(\mathsf{iPRG}(U_\lambda))} \left| \frac{1}{2^\lambda} - \frac{1}{2^\lambda + k} \right| + \sum_{a \in A} \left| \frac{1}{2^\lambda + k} \right| \\ &= 1 - \frac{2^\lambda}{2^\lambda + k} + \frac{k}{2^\lambda + k} = \frac{2k}{2^\lambda + k}.\end{aligned}$$

Hence, for $\Delta(\mathsf{iPRG}(U_\lambda), D) \leq \epsilon(\lambda)$, k is upper bounded by

$$k \leq k\underbrace{(2 - \epsilon(\lambda))}_{\geq 1} \leq 2^\lambda \cdot \epsilon(\lambda).$$

Therefore, the min-entropy of D is at most $\mathrm{H}_\infty(D) = \log(2^\lambda + k) \leq \log(2^\lambda + 2^\lambda \epsilon) = \lambda + \log(1 + \epsilon)$, which is negligibly close to λ. Hence, $\mathrm{H}^{\epsilon}_{\infty}(\mathsf{iPRG}(U_\lambda)) \leq \lambda + \delta(\lambda)$ for some negligible function δ. □

Lemma 10. *Let* PRG *be a pseudorandom generator with polynomial stretch* $\mathsf{poly}(\cdot)$ *and let* ϵ *be a negligible function. Then,* $\mathrm{H}^{\epsilon}_{\infty}(\mathsf{PRG}(U_\lambda)) \leq \lambda + \delta(\lambda)$ *for some negligible function* δ*.*

Proof. Due to Lemma 1, there exists a uniform distribution D over a set of size 2^λ such that $\mathsf{PRG}(U_\lambda)$ and D are statistically close. By a similar argument as in Lemma 9, in order to upper bound the ϵ-smooth min-entropy of $\mathsf{PRG}(U_\lambda)$, it suffices to consider all ϵ-close flat distributions. Then, doe to the computations already made in the proof of Lemma 9, Lemma 10 follows. □

Therefore, we obtain the following corollary.

Corollary 3. *If one-way functions exist,* $\mathsf{PREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ *is false.*

Intuitively, a common reference string can not add additional entropy to $\mathsf{E}_X(crs, X)$. I.e. if $\mathsf{cPREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ is true for some source X, the algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ implied by $\mathsf{cPREH}^{\mathsf{det}}_{\equiv_{\mathsf{s}}}$ compress that source to its ϵ-smooth min-entropy. To prove this, we introduce the following technical lemma.

Lemma 11 (Splitting lemma). *Let* X *and* Y_x *be distributions giving rise to a joint distribution* (Y, X)*. Let* $A \subset \mathsf{supp}(Y, X)$ *such that* $\Pr_{(y,x) \leftarrow (Y,X)}[(y, x) \in A] \geq 1 - \mu(\lambda)$ *for some negligible function* μ*. Further, for all* x*, let* $p_x := \Pr_{y \leftarrow Y_x}[(y, x) \in A]$*. Then, there exists negligible functions* ν, ν' *and a set* $G \subseteq \mathsf{supp}(X)$*, such that for all* $x \in G$*,* $p_x \geq 1 - \nu(\lambda)$ *and* $\Pr_{x \leftarrow X}[x \in G] \geq 1 - \nu'(\lambda)$*.*

Proof. Define $G_n := \{x \in \mathsf{supp}(X) \colon p_x \geq 1 - n \cdot \mu(\lambda)\}$.

$$\begin{aligned}
1 - \mu(\lambda) &\leq \Pr_{(y,x)\leftarrow(Y,X)}[(y,x) \in A] \\
&= \sum_{x \in G_n} \Pr[X = x] \cdot \underbrace{p_x}_{\leq 1} + \sum_{x \notin G_n} \Pr[X = x] \cdot \underbrace{p_x}_{<1-n\cdot\mu(\lambda)} \\
&< \Pr_{x\leftarrow X}[x \in G_n] + (1 - n \cdot \mu(\lambda)) \cdot \underbrace{\sum_{x \notin G_n} \Pr[X = x]}_{=1-\Pr_{x\leftarrow X}[x \in G_n]} \\
&= \Pr_{x\leftarrow X}[x \in G_n] + 1 - n \cdot \mu(\lambda) - \Pr_{x\leftarrow X}[x \in G_n] + n \cdot \mu(\lambda) \cdot \Pr_{x\leftarrow X}[x \in G_n]
\end{aligned}$$

Hence,

$$\Pr_{x\leftarrow X}[x \in G_n] \geq 1 - \frac{1}{n}.$$

Without loss of generality, we assume $\mu(\lambda) > 0$ for all $\lambda \in \mathbb{N}$. (If $\mu(\lambda) = 0$ for some λ, then for $G := \mathsf{supp}(X)$, $p_x = 1$ for all $x \in G$, and $\Pr_{x\leftarrow X}[x \in G] = 1$.) Then, $\sqrt{\mu(\lambda)}$ is well defined and negligible. Let $n := \sqrt{\mu(\lambda)}^{-1}$. Then, by definition of G_n, for all $x \in G_n$, $p_x \geq 1 - n \cdot \mu(\lambda) = 1 - \sqrt{\mu(\lambda)}$. Furthermore, $\Pr_{x\leftarrow X}[x \in G_n] \geq 1 - \frac{1}{n} = 1 - \sqrt{\mu(\lambda)}$. □

Theorem 9. *If (weak)* $\mathsf{cPREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$ *is true for* X*, i.e. there exist a setup algorithm* Setup_X *and deterministic algorithms* $(\mathsf{E}_X, \mathsf{D}_X)$ *with* E_X *having output length* n *satisfying correctness and pseudorandomness. Then, the* ϵ*-smooth min-entropy* $\mathrm{H}^{\epsilon}_{\infty}(X) \geq n$ *for some negligible function* ϵ*.*

Proof. Note that $\mathrm{H}^{\epsilon}_{\infty}(X) = \widetilde{\mathrm{H}}^{\epsilon}_{\infty}(X \mid \mathsf{Setup}_X(1^\lambda))$ as the distribution X is independent of the CRS. Hence, it suffices to upper bound the average ϵ-smooth min-entropy $\widetilde{\mathrm{H}}^{\epsilon}_{\infty}(X \mid \mathsf{Setup}_X(1^\lambda))$.

The proof is similar to the proof of Theorem 8. We consider the distribution $Y'_{crs} := \mathsf{D}_X(crs, U_n)$. Due to Theorem 1, the distributions $\{crs \leftarrow \mathsf{Setup}_X(1^\lambda), u \leftarrow U_n \colon (crs, \mathsf{D}_X(crs, u))\}$ and $\{crs \leftarrow \mathsf{Setup}_X(1^\lambda), y \leftarrow X \colon (crs, y)\}$ are statistically indistinguishable. Hence,

$$\varDelta((X \mid \mathsf{Setup}_X(1^\lambda), (Y' \mid \mathsf{Setup}_X(1^\lambda)) \leq \delta(\lambda)$$

for some negligible function δ.

By a similar argument as in Theorem 8, due to correctness and pseudorandomness we have

$$\Pr\left[crs \leftarrow \mathsf{Setup}_X(1^\lambda), u \leftarrow U_n \colon \mathsf{E}_X(crs, \mathsf{D}_X(crs, u)) = u\right] \geq 1 - \nu'(\lambda)$$

for some negligible function ν', where the probability is over the choice of crs and u.

The Splitting lemma (Lemma 11) implies that there is exists $G \subseteq \mathsf{supp}(\mathsf{Setup}_X(1^\lambda))$ such that $\Pr[crs \leftarrow \mathsf{Setup}_X(1^\lambda) \colon crs \in G] \geq 1 - \nu''(\lambda)$ and for all $crs \in G$, $\Pr[u \leftarrow U_n \colon \mathsf{E}_X(crs, \mathsf{D}_X(crs, u)) = u] \geq 1 - \nu'''(\lambda)$ for some negligible functions ν'', ν'''. Hence, conditioned on $crs \in G$, $\mathsf{D}_X(crs, \cdot)$ operates almost injectively on the set $\{0,1\}^n$. More formally, let $V_0^{crs} \subset \{0,1\}^n$ denote the set of all u such that $\mathsf{E}_X(crs, \mathsf{D}_X(crs, u)) = u$ and let $V_1^{crs} := \{0,1\}^n \setminus V_0^{crs}$. For $crs \in G$, we have ${}^{|V_1^{crs}|}/_{|\{0,1\}^n|} \leq \nu'''(\lambda)$. Let $\overline{V_1^{crs}}$ be some arbitrary subset of $\{0,1\}^{p(\lambda)} \setminus \mathsf{D}_X(crs, V_0^{crs})$ such that $|\overline{V_1^{crs}}| = |V_1^{crs}|$ (for some polynomial p). Let Y'' be the uniform distribution over $\mathsf{D}_X(V_0^{crs}) \cup \overline{V_1^{crs}}$. Note that $|\mathsf{D}_X(crs, V_0^{crs}) \cup \overline{V_1^{crs}}| = 2^n$ and for $crs \in G$, we have $\Pr[Y'_{crs} \in \overline{V_1^{crs}}] \leq \Pr[Y'_{crs} \notin$

$\mathsf{D}_X(crs, V_0^{crs})] \leq \nu'(\lambda)$. Hence,

$$\begin{aligned}
&\sum_{crs \in G} \Pr[\mathsf{Setup}_X = crs] \cdot \begin{pmatrix} \displaystyle\sum_{a \in \mathsf{D}_X(crs, V_0^{crs})} |\Pr[Y'_{crs} = a] - \Pr[Y''_{crs} = a]| \\ + \displaystyle\sum_{a \in \overline{V_1^{crs}}} |\Pr[Y'_{crs} = a] - \Pr[Y''_{crs} = a]| \end{pmatrix} \\
&\leq \sum_{crs \in G} \Pr[\mathsf{Setup}_X = crs] \cdot \Big(1 - \frac{|V_0^{crs}|}{2^n} + \Pr[Y'_{crs} \in \overline{V_1^{crs}}] + \Pr[Y''_{crs} \in \overline{V_1^{crs}}]\Big) \\
&\leq \sum_{crs \in G} \Pr[\mathsf{Setup}_X = crs] \cdot \Big(\frac{|V_1^{crs}|}{2^n} + \Pr[Y'_{crs} \notin \mathsf{D}_X(crs, V_0^{crs})] + \frac{|\overline{V_1^{crs}}|}{2^n}\Big) \\
&\leq 3 \cdot \nu'''(\lambda) \qquad (7) \\
&\sum_{crs \notin G} \Pr[\mathsf{Setup}_X = crs] \cdot \Big(\sum_{a \in \mathsf{supp}(Y''_{crs})} |\Pr[Y'_{crs} = a] - \Pr[Y''_{crs} = a]| \Big) \\
&\leq \sum_{crs \notin G} \Pr[\mathsf{Setup}_X = crs] \cdot 2 \quad \leq \quad 2 \cdot \nu''(\lambda) \qquad (8)
\end{aligned}$$

Due to Equations (7) and (8) we have that $\Delta((Y', \mathsf{Setup}_X), (Y'', \mathsf{Setup}_X)) \leq 2 \cdot \nu''(\lambda) + 3 \cdot \nu'''(\lambda)$.

Furthermore, since for all $crs \in \mathsf{supp}(\mathsf{Setup}_X(1^\lambda))$, Y''_{crs} is the uniform distribution over a set of 2^n elements,

$$\widetilde{\mathrm{H}}_\infty(Y'' \mid \mathsf{Setup}_X(1^\lambda)) = -\log \Big(\mathop{\mathbb{E}}_{crs \leftarrow \mathsf{Setup}_X(1^\lambda)} \max_a \Pr[Y''_{crs} = a] \Big) = n.$$

Therefore, we have $\mathrm{H}_\infty^{\delta + 2\nu'' + 3\nu'''}(X) = \widetilde{\mathrm{H}}_\infty^{\delta + 2\nu'' + 3\nu'''}(X \mid \mathsf{Setup}_X(1^\lambda)) \geq n$, where δ, ν'', ν''' are negligible functions. □

Corollary 4. *If one-way functions exist,* $\mathsf{cPREH}^{\mathsf{det}}_{\equiv_\mathsf{s}}$ *is false.*

6.1.2 Computational guarantees and pseudoentropy

We study the relation of the pseudorandom encoding hypothesis with deterministic encoding algorithm to HILL and Yao entropy. Interestingly, $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ poses a lower bound on the HILL entropy of a source and an upper bound on its Yao entropy.

HILL entropy. Intuitively, the algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ implied by $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ compress a source X to its HILL entropy.

Theorem 10. *If (weak)* $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ *is true for X, i.e. there exist deterministic algorithms* $(\mathsf{E}_X, \mathsf{D}_X)$ *with* E_X *having output length n satisfying correctness and pseudorandomness. Then,* $\mathrm{H}^{\mathsf{HILL}}(X) \geq n$.

Proof. We employ a similar strategy as in the proof of Theorem 8. Let $Y' := \mathsf{D}_X(U_n)$. As already seen in the proof of Theorem 1, Y' and X are computationally indistinguishable due to correctness and pseudorandomness.

Due to correctness, $\Pr[x \leftarrow X \colon \mathsf{D}_X(\mathsf{E}_X(x)) = x] \geq 1 - \nu(\lambda)$ for some negligible function ν. Since E_X is required to be deterministic, we get $\Pr[x \leftarrow X \colon \mathsf{E}_X(\mathsf{D}_X(\mathsf{E}_X(x))) = \mathsf{E}_X(x)] \geq 1 - \nu(\lambda)$. Applying pseudorandomness, we get that the probability

$$\Pr\left[u \leftarrow U_n \colon \mathsf{E}_X(\mathsf{D}_X(u))\right] \geq 1 - \nu'(\lambda)$$

for some negligible function ν', where the probability is only over the choice of u. Let $V_0 \subset \{0,1\}^n$ denote the set of all u such that $\mathsf{E}_X(\mathsf{D}_X(u)) = u$ and let $V_1 := \{0,1\}^n \setminus V_0$. We have that, ${}^{|V_1|}/_{|\{0,1\}^n|} \leq \nu'(\lambda)$.

Let $\overline{V_1}$ be some arbitrary subset of $\{0,1\}^{p(\lambda)} \setminus \mathsf{D}_X(V_0)$ such that $|\overline{V_1}| = |V_1|$ (for some polynomial p). Let Y'' be the uniform distribution over $\mathsf{D}_X(V_0) \cup \overline{V_1}$. The min-entropy of Y'' equals n. By an argument already made in Theorem 8, the statistical distance between Y' and Y'' is negligible.

Therefore, the distributions X and Y'' are computationally indistinguishable and, hence, $\mathrm{H}^{\mathsf{HILL}}(X) \geq n$. □

This result can be generalized to conditional HILL entropy using strong $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$.

Theorem 11. *Let (X, Z) be a joint distribution. More precisely, let Z be a distribution over words of length λ. For $z \in \mathsf{supp}(Z)$, let X_z denote the conditional distribution when $Z = z$. If (strong) $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ is true for X, i.e. there exist two deterministic polynomial time algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ with E_X having output length n satisfying correctness and pseudorandomness. Then, $\mathrm{H}^{\mathsf{HILL}}(X \mid Z) \geq n$.*

Proof. For each $z \in \mathsf{supp}(Z)$, consider the distribution $Y'_z := \mathsf{D}_X(z, U_n)$. Due to Theorem 1, for all adversarially chosen z, the distributions $\{(Y'_z, z)\}$ and $\{(X_z, z)\}$ are computationally indistinguishable due to correctness and pseudorandomness.

Due to correctness, for all adversarially chosen z, $\Pr[x \leftarrow X_z \colon \mathsf{D}_X(z, \mathsf{E}_X(z, x)) = x] \geq 1 - \nu(\lambda)$ for some negligible function ν. Since E_X is required to be deterministic, we have that for all adversarially chosen z, $\Pr[x \leftarrow X_z \colon \mathsf{E}_X(z, \mathsf{D}_X(z, \mathsf{E}_X(z, x))) = \mathsf{E}_X(z, x)] \geq 1 - \nu(\lambda)$. Applying pseudorandomness, we get that there exists a negligible function ν' such that for all adversarially chosen z,

$$\Pr\left[u \leftarrow U_n \colon \mathsf{E}_X(z, \mathsf{D}_X(z, u)) = u\right] \geq 1 - \nu'(\lambda),$$

where the probability is only over the choice of u. Let $V_{0,z} \subseteq \{0,1\}^n$ denote the set of all $u \in \{0,1\}^n$ such that $\mathsf{E}_X(z, \mathsf{D}_X(z, u)) = u$ holds and let $V_{1,z} := \{0,1\}^n \setminus V_{0,z}$. We have that, ${}^{|V_{1,z}|}\!/\!{}_{|\{0,1\}^n|} \leq \nu'(\lambda)$.

Let $\overline{V_{1,z}}$ be some arbitrary subset of $\{0,1\}^{p(\lambda)} \setminus \mathsf{D}_X(z, V_{0,z})$ such that $|\overline{V_{1,z}}| = |V_{1,z}|$ (for some polynomial p). Let Y''_z be the uniform distribution over $\mathsf{D}_X(z, V_{0,z}) \cup \overline{V_{1,z}}$.

$$\widetilde{\mathrm{H}}_\infty(Y'' \mid Z) = -\log\Big(\mathop{\mathbb{E}}_{z \leftarrow Z}\big[\max_{y \in \mathsf{supp}(Y''_z)} \Pr[Y''_z = y]\big]\Big) = n.$$

By a similar argument as in Theorem 8, for all adversarially chosen z, the statistical distance between Y'_z and Y''_z is negligible.

$$\begin{aligned} \mathit{\Delta}(Y'_z, Y''_z) = & \underbrace{\sum_{a \in \mathsf{D}_X(z, V_{z,0})} \left|\Pr[Y'_z = a] - \Pr[Y''_z = a]\right|}_{\leq 1 - |V_{0,z}| \cdot 2^{-n} \leq 1 - (1 - \nu'(\lambda))} + \underbrace{\sum_{a \in \overline{V_{1,z}}} \left|\Pr[Y' = a] - \Pr[Y'' = a]\right|}_{\leq \sum_{a \in \overline{V_{1,z}}} (\Pr[Y' = a] + \Pr[Y'' = a]) \leq 2 \cdot \nu'(\lambda)} \\ \leq & \; 3 \cdot \nu'(\lambda) \end{aligned}$$

Hence, the (joint) distributions (X, Z) and (Y'', Z) are computationally indistinguishable. Therefore, $\mathrm{H}^{\mathsf{HILL}}(X \mid Z) \geq n$. □

We recall that in the proof of Theorem 9, we used the observation that independent random variables A and B satisfy $\mathrm{H}_\infty(A) = \mathrm{H}_\infty(A \mid B)$. However, this does not necessarily extend to the computational case, meaning that in our case $\mathrm{H}^{\mathsf{HILL}}(X \mid \mathsf{Setup}_X(1^\lambda), Z)$ and $\mathrm{H}^{\mathsf{HILL}}(X \mid Z)$ are not equal even though the random variables X and Setup_X are independent. Hence, in order to extend Theorem 11 to $\mathsf{cISH}^{\mathsf{det}}_{\approx_\mathsf{c}}$, we need that there exists a "good" CRS crs such that $Y'_{\mathit{crs},z}$ is almost injective on $\{0,1\}^n$ and such that $\{(\mathit{crs}, y)\}$ and $\{(\mathit{crs}, \mathsf{D}_X(\mathit{crs}, z, u))\}$ are computationally indistinguishable for this fixed crs. However, since the adversary is quantified after the crs, a non-uniform adversary could know the randomness which was used to generate crs compromising all security guarantees. Therefore, we can only hope to obtain an upper bound on n depending on the conditional HILL entropy $\mathrm{H}^{\mathsf{HILL}}(X \mid \mathsf{Setup}_X(1^\lambda), Z)$.

Theorem 12. *Let (X, Z) be a joint distribution. More precisely, let Z be a distribution over words of length λ. For $z \in \mathsf{supp}(Z)$, let X_z denote the conditional distribution when $Z = z$. If (strong) $\mathsf{cPREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ is true for X, i.e. there exists a PPT algorithm Setup_X and two deterministic polynomial time algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ with E_X having output length n satisfying correctness and pseudorandomness. Then, $\mathrm{H}^{\mathsf{HILL}}(X \mid \mathsf{Setup}_X(1^\lambda), Z) \geq n$.*

Proof. The proof is similar to the proof of Theorem 11. We consider the distribution $Y'_{crs,z} := \mathsf{D}_X(crs, z, U_n)$.

Due to Theorem 1, we have that for all adversarially chosen z, the distributions $\{crs \leftarrow \mathsf{Setup}_X(1^\lambda), y \leftarrow X_z \colon (crs, z, y)\}$ and $\{crs \leftarrow \mathsf{Setup}_X(1^\lambda), u \leftarrow U_n \colon (crs, z, \mathsf{D}_X(crs, z, u))\}$ are computationally indistinguishable.

Due to correctness and pseudorandomness we have that there exists a negligible function ν', such that for all adversarially chosen z,

$$\Pr\left[crs \leftarrow \mathsf{Setup}_X(1^\lambda), u \leftarrow U_n \colon \mathsf{E}_X(crs, z, \mathsf{D}_X(crs, z, u)) = u\right] \geq 1 - \nu'(\lambda),$$

where the probability is over the choice of crs and u. The Splitting lemma (Lemma 11) implies that for all z[12], there is exists $G_z \subseteq \mathsf{supp}(\mathsf{Setup}_X(1^\lambda))$ such that $\Pr[crs \leftarrow \mathsf{Setup}_X(1^\lambda) \colon crs \in G_z] \geq 1 - \nu''_z(\lambda)$ and for all $crs \in G_z$, $\Pr[u \leftarrow U_n \colon \mathsf{E}_X(crs, z, \mathsf{D}_X(crs, z, u)) = u] \geq 1 - \nu'''_z(\lambda)$ for some negligible functions ν''_z, ν'''_z. Hence, conditioned on $crs \in G_z$, $\mathsf{D}_X(crs, z, \cdot)$ operates almost injectively on the set $\{0,1\}^n$.

Let $V^{crs}_{0,z} \subseteq \{0,1\}^n$ denote the set of all $u \in \{0,1\}^n$ such that $\mathsf{E}_X(crs, z, \mathsf{D}_X(crs, z, u)) = u$ holds and let $V^{crs}_{1,z} := \{0,1\}^n \setminus V^{crs}_{0,z}$. For $crs \in G_z$, we have ${|V_{1,z}{}^{crs}|}/{|\{0,1\}^n|} \leq \nu'''_z(\lambda)$.

Let $\overline{V^{crs}_{1,z}}$ be some arbitrary subset of $\{0,1\}^{p(\lambda)} \setminus \mathsf{D}_X(crs, z, V^{crs}_{0,z})$ such that $|\overline{V^{crs}_{1,z}}| = |V^{crs}_{1,z}|$ (for some polynomial p). Let $Y''_{crs,z}$ be the uniform distribution over $\mathsf{D}_X(crs, z, V^{crs}_{0,z}) \cup \overline{V^{crs}_{1,z}}$.

$$\widetilde{\mathrm{H}}_\infty(Y'' \mid \mathsf{Setup}_X(1^\lambda), Z) = -\log\Big(\mathop{\mathbb{E}}_{crs \leftarrow \mathsf{Setup}_X(1^\lambda), z \leftarrow Z}\big[\max_{y \in \mathsf{supp}(Y''_{crs,z})} \Pr[Y''_{crs,z} = y]\big]\Big) = n.$$

Furthermore, a similar computation as in Theorem 9, for all adversarially chosen z, the statistical distance between $(Y'_z, \mathsf{Setup}_X(1^\lambda))$ and $(Y''_z, \mathsf{Setup}_X(1^\lambda))$ is negligible. Hence, the (joint) distributions $(X, \mathsf{Setup}_X(1^\lambda), Z)$ and $(Y'', \mathsf{Setup}_X(1^\lambda), Z)$ are computationally indistinguishable. Therefore, $\mathrm{H}^{\mathsf{HILL}}(X \mid \mathsf{Setup}_X(1^\lambda), Z) \geq n$.

□

Yao entropy. Furthermore, the existence of algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ as implied by $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ give an upper bound for the Yao entropy of a source.

Lemma 12. *If (weak) $\mathsf{PREH}^{\mathsf{det}}_{\approx_\mathsf{c}}$ is true for X, i.e. there exist deterministic algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ with E_X having output length n satisfying correctness and pseudorandomness. Then, $\mathrm{H}^{\mathsf{Yao}}(X) < n + \delta(\lambda)$ for some negligible δ.*

Proof. Due to correctness, there exists a pair of efficient algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ (with E_X having output length n) such that $\Pr_{x \leftarrow X}[\mathsf{D}_X(\mathsf{E}_X(x)) = x] \geq 1 - \nu(\lambda)$ for some negligible function ν. Hence, by Definition 21, $\mathrm{H}^{\mathsf{Yao}}(X) < k$, for all k satisfying

$$\begin{aligned} & 1 - \nu(\lambda) \geq 2^{n-k} + \epsilon(\lambda) \\ \iff \quad & 2^k \cdot (1 - \nu(\lambda) - \epsilon(\lambda)) \geq 2^n \\ \iff \quad & k \geq n - \log(1 - \nu(\lambda) - \epsilon(\lambda)). \end{aligned}$$

□

[12] Note that here we use that the adversary is non-uniform.

This result can be generalized to conditional Yao entropy as follows.

Lemma 13. *Let (X, Z) be a joint distribution. More precisely, let Z be a distribution over words of length λ. For $z \in \mathsf{supp}(Z)$, let X_z denote the conditional distribution when $Z = z$. If (strong) $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is true for X, i.e. there exist two deterministic polynomial time algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ with E_X having output length n satisfying correctness and pseudorandomness. Then, $\mathrm{H}^{\mathsf{Yao}}(X \mid Z) < n + \delta(\lambda)$ for some negligible δ.*

Proof. Due to correctness, for all adversarially chosen z, $\Pr_{x \leftarrow X_z}[\mathsf{D}_X(z, \mathsf{E}_X(z, x)) = x] \geq 1 - \nu(\lambda)$ for some negligible function ν. Hence, by Definition 23, $\mathrm{H}^{\mathsf{Yao}}(X \mid Z) < k$, for all k satisfying $k \geq n - \log(1 - \nu(\lambda) - \epsilon(\lambda))$. □

Lemma 14. *Let (X, Z) be a joint distribution. More precisely, let Z be a distribution over words of length λ. For $z \in \mathsf{supp}(Z)$, let X_z denote the conditional distribution when $Z = z$. If (strong) $\mathsf{cPREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is true for X, i.e. there exists a setup algorithm Setup and two deterministic polynomial time algorithms $(\mathsf{E}_X, \mathsf{D}_X)$ with E_X having output length n satisfying correctness and pseudorandomness. Then, $\mathrm{H}^{\mathsf{Yao}}(X \mid Z) < n + \delta(\lambda)$ for some negligible δ.*

Proof. The Splitting lemma (Lemma 11) implies that for all z,[13] there is exists $G_z \subseteq \mathsf{supp}(\mathsf{Setup}_X(1^\lambda))$ such that $\Pr[crs \leftarrow \mathsf{Setup}_X(1^\lambda) \colon crs \in G_z] \geq 1 - \nu''_z(\lambda)$ and for all $crs \in G_z$, $\Pr[u \leftarrow U_n \colon \mathsf{E}_X(crs, z, \mathsf{D}_X(crs, z, u)) = u] \geq 1 - \nu'''_z(\lambda)$ for some negligible functions ν''_z, ν'''_z.

We exploit the non-uniformity of the definition. In particular, we define the compression and decompression algorithms $\mathsf{E}'_X(y, z; \mathsf{aux}_\lambda) := \mathsf{E}_X(\mathsf{aux}_\lambda, z, y)$ and $\mathsf{D}'_X(u, z; \mathsf{aux}_\lambda) := \mathsf{D}_X(\mathsf{aux}_\lambda, z, u)$, where aux_λ denotes the non-uniform auxiliary input. □

Due to [HLR07], assuming suitable non-interactive zero-knowledge proof systems and pseudorandom generators, there exists a joint distribution (X, Z) with high conditional Yao entropy but low conditional HILL entropy. The sampler S from [HLR07] takes as input a NIZK common random string σ, samples a seed from U_λ and outputs $y_1 := \mathsf{PRG}(s)$ together with a proof y_2 that y_1 is in the image of PRG. Due to Lemma 2, $S \in \mathcal{S}^{\mathsf{comp}}$. This yields the following corollary.

Corollary 5. *If there exist a pseudorandom generator and a (single-theorem) NIZK proof system such that (i) for an overwhelming fraction of common random strings, the number of accepting proofs for each statement are limited, and (ii) the simulated random string is independent of the statement as in [HLR07]. Then $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is false.*

Together with the results of [HLR07; LMs05], we obtain the following corollary.

Corollary 6. *A Blum integer is a natural number $N = p \cdot q$ such that p, q are primes with $p \equiv q \equiv 3 \bmod 4$. Assume that for a randomly chosen Blum integer $N = p \cdot q$, the distributions $\{y \leftarrow \mathbb{Z}_N^\times \text{ s.t. } \left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = 1 \colon (N, y)\}$ and $\{y \leftarrow \mathbb{Z}_N^\times \text{ s.t. } \left(\frac{y}{N}\right) = 1 \colon (N, y)\}$ are computationally indistinguishable. Then, $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$ is false.*

On refuting $\mathsf{cPREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$. Theorem 12 only yields an upper bound on the encoding length n depending on $\mathrm{H}^{\mathsf{HILL}}(X \mid \mathsf{Setup}_X(1^\lambda), Z)$. Since the Yao entropy of any source $\mathrm{H}^{\mathsf{Yao}}(X \mid \mathsf{Setup}_X(1^\lambda), Z)$ is upper bounded by $n + \mathsf{negl}(\lambda)$, we can not apply the result from [HLR07] to refute $\mathsf{cPREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$.

On refuting weak $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$. It is currently not known if plain HILL and plain Yao entropy can be separated as in [HLR07]. Such a separation would refute weak $\mathsf{PREH}^{\mathsf{det}}_{\approx_{\mathsf{c}}}$.

[13] Note that here we use that the adversary is non-uniform.

6.2 Randomized encoding

In the following we prove positive and negative results on the validity of the pseudorandom encoding hypothesis with a randomized encoding algorithm. In particular, in Section 6.2.2 we refute $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$ based on sub-exponential LWE, in Section 6.2.3 we refute $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ based on extractable one-way functions with *unbounded* auxiliary input. On the positive side, in Section 6.3, give a construction of perfectly correct $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with universal setup based on indistinguishability obfuscation and one-way functions following [SW14; DKR15] together with Theorem 1. In Section 6.4, we bootstrap $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with a common random string from the construction in Section 6.3 additionally assuming *weak* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with a common random string. Since $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with a common random string in conjunction with NIZK proof systems contradicts EOWFs with common but benign auxiliary information, this refutes even *weak* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ with common random string.

6.2.1 (Generalized) extractable one-way functions

In this section, we define (generalized) extractable one-way functions (with respect to common auxiliary input) as in [BCPR14].

Definition 26 (Function family ensemble). *A* function family*, indexed by a key space K_λ, is a set of functions $f^\lambda = \{f_k\}_{k \in K_\lambda}$ where each function has the same domain and range. A* function family ensemble *$\mathcal{F} = \{f^\lambda\}_{\lambda\in\mathbb{N}}$ is an ensemble of function families with key spaces $\{K_\lambda\}_{\lambda\in\mathbb{N}}$.*

Definition 27 (Extractable one-way function family ensembles (EOWFs) without auxiliary information, [BCPR14]). *A function family ensemble is called a* extractable one-way function family ensemble without auxiliary information *if the following two properties are satisfied.*

One-wayness. *For every PPT adversary $\mathcal{A}$, $\Pr[Exp^{\mathsf{ow}}_{\mathcal{A}}(\lambda) = 1]$ is negligible, where $Exp^{\mathsf{ow}}_{\mathcal{A}}$ is defined in Figure 16.*

Extractability. *For every PPT adversary X (using a random tape of length $m(\lambda)$), there exists a PPT algorithm K_X such that $\Pr[Exp^{\mathsf{ext}}_{X,K_X}(\lambda) = 1]$ is overwhelming, where $Exp^{\mathsf{ext}}_{X,K_X}$ is defined in Figure 16.*

```
$Exp^{\mathsf{ow}}_{\mathcal{A}}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$x \leftarrow \mathsf{domain}(f_k)$
$y := f_k(x)$
$x' \leftarrow \mathcal{A}(k, y)$
return $f_k(x') = y$
```

```
$Exp^{\mathsf{ext}}_{X,K_X}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$r_X \leftarrow \{0,1\}^{m(\lambda)}$
$y = X(k; r_X)$
$x \leftarrow K_X(k, r_X)$
return $\big(f_k(x) = y\big) \vee \big(\forall x' \colon f_k(x') \neq y\big)$
```

Fig. 16. One-way and extraction game.

Definition 28 (Extractable one-way function family ensembles (EOWFs) with common auxiliary information, [BCPR14]). *A function family ensemble is called a* one-way extractable function family ensemble with common auxiliary information *if the following properties are satisfied.*

One-wayness. *As in Definition 27.*

One-wayness (stronger). *For every PPT adversary $\mathcal{A}$, for every polynomial b and for every $z \in \{0,1\}^{b(\lambda)}$, $\Pr[Exp^{\mathsf{ow\text{-}aux}}_{\mathcal{A},z}(\lambda) = 1]$ is negligible, where $Exp^{\mathsf{ow\text{-}aux}}_{\mathcal{A},z}$ is defined in Figure 17.*

Extractability. *For every PPT adversary X, there exists a PPT algorithm K_X such that for every polynomial b and every $z \in \{0,1\}^{b(\lambda)}$, $\Pr[Exp^{\mathsf{ext\text{-}aux}}_{X,K_X,z}(\lambda) = 1]$ is overwhelming, where $Exp^{\mathsf{ext\text{-}aux}}_{X,K_X,z}$ is defined in Figure 17.*

```
$Exp^{\mathsf{ow\text{-}aux}}_{\mathcal{A},z}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$x \leftarrow \mathsf{domain}(f_k)$
$y := f_k(x)$
$x' \leftarrow \mathcal{A}(k, y, z)$
return $f_k(x') = y$
```

```
$Exp^{\mathsf{ext\text{-}aux}}_{X,K_X,z}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$r_X \leftarrow \{0,1\}^{m(\lambda)}$
$y = X(k, z; r_X)$
$x \leftarrow K_X(k, z, r_X)$
return $\big(f_k(x) = y\big) \vee \big(\forall x' \colon f_k(x') \neq y\big)$
```

Fig. 17. One-way and extraction game with common auxiliary input.

We note that the weaker notion of one-wayness without auxiliary input is sufficient for us. Assuming non-uniform adversaries, one-wayness without auxiliary input and one-wayness with auxiliary input are equivalent.

Definition 29 (Extractable one-way function family ensembles (EOWFs) with b-bounded common auxiliary information, [BCPR14]). *Like Definition 28 but with a fixed polynomial b determining the length of the common auxiliary information.*

Definition 30 (Generalized extractable one-way function family ensembles (GEOWFs) with common auxiliary information, [BCPR14]). *A function family ensemble $\mathcal{F}$ is called a* generalized one-way extractable function family ensemble with common auxiliary information*, with respect to a relation $R^{\mathcal{F}}_k$ over triplets $(k, y, x) \in \mathcal{K}_\lambda \times \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}$, if the following properties are satisfied.*

$R^{\mathcal{F}}$-Hardness. *For every PPT adversary $\mathcal{A}$, for every polynomial b and every $z \in \{0,1\}^{b(\lambda)}$,* $\Pr[Exp^{\mathsf{g\text{-}hard\text{-}aux}}_{\mathcal{A},z}(\lambda) = 1]$ *is negligible, where $Exp^{\mathsf{g\text{-}hard\text{-}aux}}_{\mathcal{A},z}$ is defined in Figure 18.*

$R^{\mathcal{F}}$-Extractability. *For every PPT adversary X, there exists a PPT algorithm K_X such that for every polynomial b and every $z \in \{0,1\}^{b(\lambda)}$,* $\Pr[Exp^{\mathsf{g\text{-}ext\text{-}aux}}_{X,K_X,z}(\lambda) = 1]$ *is overwhelming, where $Exp^{\mathsf{g\text{-}ext\text{-}aux}}_{X,K_X,z}$ is defined in Figure 18.*

We call $\mathcal{F}$

- publicly verifiable *if there exists a PPT algorithm T such that*
$$T(k, f_k(x), x') = 1 \Leftrightarrow R^{\mathcal{F}}_k(f_k(x), x') = 1.$$
- privately verifiable *if there exists a PPT algorithm T such that*
$$T(k, x, x') = 1 \Leftrightarrow R^{\mathcal{F}}_k(f_k(x), x') = 1.$$

```
$Exp^{\mathsf{g\text{-}hard\text{-}aux}}_{\mathcal{A},z}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$x \leftarrow \mathsf{domain}(f_k)$
$y := f_k(x)$
$x' \leftarrow \mathcal{A}(k, y, z)$
return $R^{\mathcal{F}}_k(y, x')$
```

```
$Exp^{\mathsf{g\text{-}ext\text{-}aux}}_{X,K_X,z}(\lambda)$
$k \leftarrow \mathcal{K}_\lambda$
$r_X \leftarrow \{0,1\}^{m(\lambda)}$
$y = X(k, z; r_X)$
$x \leftarrow K_X(k, z, r_X)$
return $\big(R^{\mathcal{F}}_k(y, x)\big) \vee \big(\forall x' \colon f_k(x') \neq y\big)$
```

Fig. 18. $R^{\mathcal{F}}$-hardness and $R^{\mathcal{F}}$-extraction game with common auxiliary input.

Definition 31 (Generalized extractable one-way function family ensembles (GEOWFs) with b-bounded common auxiliary information, [BCPR14]). *Like Definition 30 but with a fixed polynomial b determining the length of the common auxiliary information.*

Privately verifiable generalized extractable one-way functions with bounded auxiliary input can be instantiated based on falsifiable assumptions due to [BCPR14].

Theorem 13 ([BCPR14]). *Assuming the learning with errors problem is sub-exponentially hard, there exists a $(b(\lambda) - \omega(1))$-bounded privately verifiable GEOWF family ensemble.*

6.2.2 Information theoretic guarantees and privately verifiable GEOWFs

Assuming information theoretic indistinguishability, adversaries are unbounded and, hence, private verifiability is not a restriction.

Theorem 14. *If privately verifiable generalized extractable one-way function family ensembles without auxiliary information exist, then* $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$ *is false.*

Proof. The proof strategy follows the ideas of [IKOS10]. Let $\mathcal{F}$ be a privately verifiable GEOWF family ensemble (without auxiliary input) with respect to relation $R^{\mathcal{F}}$.

$\mathsf{PREH}^{\mathsf{rand}}_{\equiv_{\mathsf{s}}}$ implies that for the algorithm S (given in Figure 19) there exists an alternative sampler $\overline{S}$ and a corresponding inverse sampler $\overline{S}^{-1}$ satisfying closeness and invertibility of Definition 7 against unbounded adversaries. Since $\mathcal{F}$ is a GEOWF family ensemble, for the algorithm $\overline{S}$, there exists an extractor $K_{\overline{S}}$ satisfying extractability from Definition 30 (without auxiliary information).

```
$S(k)$
  $x \leftarrow \mathsf{domain}(f_k)$
  $y := f_k(x)$
  return $y$

$\mathcal{A}(1^\lambda, k, y)$
  $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(k,y)$
  $x' \leftarrow K_{\overline{S}}(k, r'_X)$
  return $x'$
```

Fig. 19. Description of the sampler S and of the adversary $\mathcal{A}$ on one-wayness of the privately verifiable GEOWF.

We prove that for $\mathcal{A}$ given in Figure 19, $\Pr[Exp^{\mathsf{g\text{-}hard}}_{\mathcal{A}}(\lambda) = 1]$ is overwhelming. Let $\widetilde{T}$ be an *unbounded* algorithm that given (k, y, x), computes the relation $R^{\mathcal{F}}_k(y, x)$.[14] We proceed over a series of hybrids, see Figure 20.

```
$\mathbf{G}_0$
  $k \leftarrow \mathcal{K}_\lambda$
  $x \leftarrow \mathsf{domain}(f_k)$
  $y := f_k(x)$
  $x' \leftarrow \mathcal{A}(k,y)$
  return $\widetilde{T}(k,y,x')$

$\mathbf{G}_1$
  $k \leftarrow \mathcal{K}_\lambda$
  $x \leftarrow \mathsf{domain}(f_k)$
  $y := f_k(x)$
  $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(k,y)$
  $x' \leftarrow K_{\overline{S}}(k, r'_{\overline{S}})$
  return $\widetilde{T}(k,y,x')$

$\mathbf{G}_2$
  $k \leftarrow \mathcal{K}_\lambda$
  $y \leftarrow S(k)$
  $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(k,y)$
  $x' \leftarrow K_{\overline{S}}(k, r'_{\overline{S}})$
  return $\widetilde{T}(k,y,x')$

$\mathbf{G}_3$
  $k \leftarrow \mathcal{K}_\lambda$
  $y \leftarrow \overline{S}(k)$
  $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(k,y)$
  $x' \leftarrow K_{\overline{S}}(k, r'_{\overline{S}})$
  return $\widetilde{T}(k,y,x')$

$\mathbf{G}_4$
  $k \leftarrow \mathcal{K}_\lambda$
  $r_{\overline{S}} \leftarrow \{0,1\}^*$
  $y \leftarrow \overline{S}(k; r_{\overline{S}})$
  $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(k,y)$
  $x' \leftarrow K_{\overline{S}}(k, r'_{\overline{S}})$
  return $\widetilde{T}(k,y,x')$

$\mathbf{G}_5$
  $k \leftarrow \mathcal{K}_\lambda$
  $r_{\overline{S}} \leftarrow \{0,1\}^*$
  $y \leftarrow \overline{S}(k; r_{\overline{S}})$
  $x' \leftarrow K_{\overline{S}}(k, r_{\overline{S}})$
  return $\widetilde{T}(k,y,x')$
```

Fig. 20. Hybrids used in the proof of Theorem 14.

The individual game hops are justified as follows. The difference between games $\mathbf{G}_0$ and $\mathbf{G}_1$, and games $\mathbf{G}_1$ and $\mathbf{G}_2$ are both only conceptual, hence, $\Pr[out_0 = 1] = \Pr[out_1 = 1] = \Pr[out_2 = 1]$.

Claim. There exists an unbounded adversary $\overline{\mathcal{A}}$ such that $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| \leq Adv^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k}}(\lambda)$ (for some $\overline{k} \in \mathcal{K}_\lambda$).

Proof. Construct an adversary $\overline{\mathcal{A}}$ on statistical closeness. $\overline{\mathcal{A}}$ receives as input a key $\overline{k}$ and some y that has either been sampled via $S(\overline{k})$ or via $\overline{S}(\overline{k})$. $\overline{\mathcal{A}}$ computes $r'_{\overline{S}}$ and x' as in game $\mathbf{G}_2$, calls

[14] Since $\mathcal{F}$ is privately verifiable, an efficient algorithm computing the relation $R^{\mathcal{F}}_k$ additionally requires the preimage of y as an input. For our purpose, it suffices to consider an inefficient testing algorithm $\widetilde{T}$.

the (inefficient) testing algorithm $\widetilde{T}$ on input of $(\overline{k}, y, x')$ and outputs the resulting output. Hence, $\Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k},0}(\lambda) = 1] = \Pr[out_2 = 1 \mid k = \overline{k}]$ and $\Pr[Exp^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k},1}(\lambda) = 1] = \Pr[out_3 = 1 \mid k = \overline{k}]$. Since $|\Pr[out_3 = 1] - \Pr[out_2 = 1]| = |\sum_{\overline{k} \in \mathcal{K}_\lambda} \Pr_{k \leftarrow \mathcal{K}_\lambda}[k = \overline{k}] \cdot (\Pr[out_3 = 1 \mid k = \overline{k}] - \Pr[out_2 = 1 \mid k = \overline{k}])| = |\sum_{\overline{k} \in \mathcal{K}_\lambda} \Pr_{k \leftarrow \mathcal{K}_\lambda}[k = \overline{k}] \cdot Adv^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k}}(\lambda)|$, $|\Pr[out_3 = 1] - \Pr[out_2 = 1]|$ is negligible. □

The difference between the games $\mathbf{G}_3$ and $\mathbf{G}_4$ is again only conceptual and $\Pr[out_3 = 1] = \Pr[out_4 = 1]$.

Claim. There exists an unbounded adversary $\overline{\mathcal{A}}$ such that $|\Pr[out_5 = 1] - \Pr[out_4 = 1]| \leq Adv^{\mathsf{inv}}_{\overline{\mathcal{A}},\overline{k}}(\lambda)$ (for some $\overline{k} \in \mathcal{K}_\lambda$).

Proof. Construct an unbounded adversary $\overline{\mathcal{A}}$ on statistical invertibility. On input of $(\overline{k}, r, y)$, $\overline{\mathcal{A}}$ calls $K_{\overline{S}}$ on input of $(\overline{k}, r)$ and obtains x'. Finally, $\overline{\mathcal{A}}$ calls the inefficient testing algorithm $\widetilde{T}$ on input of $(\overline{k}, y, x')$ and outputs the resulting output. If $\overline{\mathcal{A}}$ plays the experiment $Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},\overline{k},0}$, r is the randomness used to produce y as $\overline{S}(\overline{k}; r)$. Hence, in this case, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_5$ and $\Pr[Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},\overline{k},0}(\lambda) = 1] = \Pr[out_5 = 1 \mid k = \overline{k}]$. If, on the other hand, $\overline{\mathcal{A}}$ plays the experiment $Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},\overline{k},1}$, r is some inverse sampled randomness (i.e. the output of $\overline{S}^{-1}(\overline{k}, y)$). Hence, in this case, $\overline{\mathcal{A}}$ perfectly simulates game $\mathbf{G}_4$ and $\Pr[Exp^{\mathsf{inv}}_{\overline{\mathcal{A}},\overline{k},1}(\lambda) = 1] = \Pr[out_4 = 1 \mid k = \overline{k}]$. Thus, by the same computations made in the reduction to closeness, the claim follows. □

Summing up, we have that $|\Pr[out_5 = 1] - \Pr[out_0 = 1]|$ is negligible.

Lemma 15. $\Pr[out_5 = 1]$ *is overwhelming.*

Proof (of Lemma 15). Since $\mathcal{F}$ is a GEOWF family ensemble, we have that $\Pr_{\mathbf{G}_5}[R^{\mathcal{F}}_k(y, x') = 1 \vee y \notin \mathsf{image}(f_k)]$ is overwhelming. Using a union bound, we get $\Pr_{\mathbf{G}_5}[R^{\mathcal{F}}_k(y, x') = 1 \vee y \notin \mathsf{image}(f_k)] \leq \Pr_{\mathbf{G}_5}[R^{\mathcal{F}}_k(y, x') = 1] + \Pr_{\mathbf{G}_5}[y \notin \mathsf{image}(f_k)]$.

In the following, we prove that $\Pr_{\mathbf{G}_5}[y \notin \mathsf{image}(f_k)]$ is negligible. By construction, S on input k always produces values $y \in \mathsf{image}(f_k)$. Intuitively, if $\Pr_{\mathbf{G}_5}[y \notin \mathsf{image}(f_k)] = \Pr[k \leftarrow \mathcal{K}_\lambda, y \leftarrow \overline{S}(k) \colon y \notin \mathsf{image}(f_k)]$ is non-negligible, an unbounded adversary can distinguish between outputs of $S(k)$ and outputs of $\overline{S}(k)$ simply by testing all possible preimages. Let $\overline{\mathcal{A}}$ be the adversary on closeness which outputs 1 if and only if $y \notin \mathsf{image}(f_k)$. Hence, $\Pr_{\mathbf{G}_5}[y \notin \mathsf{image}(f_k)] = \sum_{\overline{k} \in \mathcal{K}_\lambda} \Pr[k = \overline{k}] \cdot Adv^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k}}(\lambda)$. Since for all $k \in \mathcal{K}_\lambda$, $Adv^{\mathsf{close}}_{\overline{\mathcal{A}},\overline{k}}(\lambda)$ is negligible, $\Pr_{\mathbf{G}_5}[y \notin \mathsf{image}(f_k)]$ is negligible. □

□

From Theorem 14 together with Theorem 13, we obtain the following corollary.

Corollary 7. *Assuming the learning with errors problem is sub-exponentially hard,* $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}$ *is false.*

Theorem 15. *If privately verifiable generalized extractable one-way function family ensembles with unbounded auxiliary information exist, then* $\mathsf{cPREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}$ *is false.*

The proof is an easy modification of the proof of Theorem 14.

6.2.3 Computational guarantees and EOWFs with common auxiliary information

Assuming computational indistinguishability, the above proof strategy must be adapted since the adversary actually uses the alternative sampler of a sampler which outputs an image in the range of the EOWF. Computational indistinguishability does not suffice to force the alternative sampler to output an image in the range of the EOWF. Hence, for the purpose to force the alternative sampler to output an image of a given extractable one-way function, we follow the lines of [IKOS10] and require the original sampler to additionally provide a non-interactive zero-knowledge proof certifying that the provided image is in the range of the given EOWF.

Definition 32 (Non-interactive zero-knowledge proof system). *A non-interactive zero-knowledge (NIZK) proof system for the $\mathcal{NP}$-language L with witness relation R_L is a tuple of PPT algorithms (P, V) satisfying the following conditions for all sufficiently large λ (for some polynomial ℓ).*

Correctness. *For all $(x, w) \in R_L$, for all $\sigma \in \{0,1\}^{\ell(\lambda)}$, $\Pr[V(\sigma, x, P(\sigma, x, w)) = 1] = 1$.*
Statistical soundness. *For all unbounded adversaries $\mathcal{A}$,*

$$Adv^{\mathsf{soundness}}_{\mathcal{A}}(\lambda) := \Pr[\sigma \leftarrow \{0,1\}^{\ell(\lambda)}, (x, \pi) \leftarrow \mathcal{A}(\sigma) \colon V(\sigma, x, \pi) = 1 \wedge x \notin L]$$

is negligible.
Computational zero-knowledge. *There exists a tuple of PPT algorithms* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that for all non-uniform PPT algorithms $\mathcal{A}$,*

$$Adv^{\mathsf{zk}}_{\mathcal{A}}(\lambda) := |\Pr[Exp^{\mathsf{zk}}_{\mathcal{A},0}(\lambda) = 1] - \Pr[Exp^{\mathsf{zk}}_{\mathcal{A},1}(\lambda) = 1]|$$

is negligible, where $Exp^{\mathsf{zk}}_{\mathcal{A},0}$ and $Exp^{\mathsf{zk}}_{\mathcal{A},1}$ are defined in Figure 21.

```
$Exp^{\mathsf{zk}}_{\mathcal{A},0}(\lambda)$
$\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
$(x, w) \leftarrow \mathcal{A}(\sigma)$ s.t. $(x, w) \in R_L$
$\pi \leftarrow P(\sigma, x, w)$
return $\mathcal{A}(\pi)$
```

```
$Exp^{\mathsf{zk}}_{\mathcal{A},1}(\lambda)$
$(\sigma, \tau) \leftarrow \mathsf{Sim}_1(1^\lambda)$
$(x, w) \leftarrow \mathcal{A}(\sigma)$ s.t. $(x, w) \in R_L$
$\pi \leftarrow \mathsf{Sim}_2(\tau, x)$
return $\mathcal{A}(\pi)$
```

Fig. 21. Zero-knowledge games.

As already observed in [IKOS10], $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ in conjunction with NIZK proof systems conflicts with EOWFs.

Theorem 16 ([IKOS10]). *If extractable one-way function family ensembles without auxiliary information and NIZK proof systems for $\mathcal{NP}$ exist, then $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ is false.*

This result extends to $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ at the cost of assuming unbounded common auxiliary inputs.

Theorem 17. *If there exist extractable one-way function family ensembles with unbounded common auxiliary information and NIZK proof systems for $\mathcal{NP}$, then $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ is false.*

Proof. We adapt the proof from [IKOS10]. Let $\mathcal{F}$ be an extractable one-way function family ensemble with common auxiliary information. Let $L_\lambda := \{(k, y) \in \mathcal{K}_\lambda \times \{0,1\}^{m(\lambda)} \mid \exists x \in \mathsf{domain}(f_k) \colon f_k(x) = y\}$. Let (P, V) be a NIZK proof system for L_λ.

$\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ implies that for the PPT algorithm S (see Figure 22), there exists a PPT algorithm Setup_S, an alternative sampler $\overline{S}$ and a corresponding inverse sampler $\overline{S}^{-1}$ satisfying closeness and invertibility as in Definition 9.

The common auxiliary input is necessary to give the adversary X and the extractor K_X access to the same common reference string. Further, it is crucial to assume *unbounded* auxiliary input since the adversary X can be considered to be a universal adversary which simply executes the code which is contained in the auxiliary input. Since the size of the adversary can not be bounded in advance, neither can the size of the auxiliary input.

Since $\mathcal{F}$ is an extractable one-way function family ensemble with common auxiliary information, we have that for the algorithm X, there exists an extractor K_X such that for every polynomial b and every $crs \in \{0,1\}^{b(\lambda)}$ (hence, in particular, for every $z := crs$ produced by $\mathsf{Setup}_S(1^\lambda)$), $\Pr[Exp^{\mathsf{ext\text{-}aux}}_{X,K_X,z}(\lambda) = 1]$ is overwhelming.

We prove that adversary $\mathcal{A}$ given in Figure 22 has an overwhelming probability to break the one-wayness of $\mathcal{F}$. We proceed over a sequence of hybrids, see Figures 23 and 24.

The individual game hops are justified as follows. The difference between the games $\mathbf{G}_0$ and $\mathbf{G}_1$ is only conceptual, hence, $\Pr[out_0 = 1] = \Pr[out_1 = 1]$.

```
$S(k, \sigma)$
    $x \leftarrow \mathsf{domain}(f_k)$
    $y := f_k(x)$
    $\pi \leftarrow P(\sigma, (k, y), x)$
    $\mathbf{return}\ (y, \pi)$
```

```
$X(k, z =: crs; \sigma \parallel r_{\overline{S}})$
    $(y, \pi) \leftarrow \overline{S}(crs, (k, \sigma); r_{\overline{S}})$
    $\mathbf{return}\ y$
```

```
$\mathcal{A}(1^\lambda, k, y)$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $(\sigma, \tau) \leftarrow \mathsf{Sim}_1(1^\lambda)$
    $\pi \leftarrow \mathsf{Sim}_2(\tau, (k, y))$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ x'$
```

Fig. 22. Description of the sampler S, of the adversary X on extractability of the EOWF and of the adversary $\mathcal{A}$ on one-wayness of the EOWF. Note that depending on definition of one-wayness, the CRS crs could also be auxiliary input to $\mathcal{A}$.

```
$\mathbf{G}_0$
    $k \leftarrow \mathcal{K}_\lambda$
    $x \leftarrow \mathsf{domain}(f_k)$
    $y := f_k(x)$
    $x' \leftarrow \mathcal{A}(k, y)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_1$
    $k \leftarrow \mathcal{K}_\lambda$
    $x \leftarrow \mathsf{domain}(f_k)$
    $y := f_k(x)$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $(\sigma, \tau) \leftarrow \mathsf{Sim}_1(1^\lambda)$
    $\pi \leftarrow \mathsf{Sim}_2(\tau, (k, y))$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_2$
    $k \leftarrow \mathcal{K}_\lambda$
    $x \leftarrow \mathsf{domain}(f_k)$
    $y := f_k(x)$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $\pi \leftarrow P(crs, (k, y), x)$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_3$
    $k \leftarrow \mathcal{K}_\lambda$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $(y, \pi) \leftarrow S(k, \sigma)$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

Fig. 23. Hybrids used in the proof of Theorem 17.

```
$\mathbf{G}_4$
    $k \leftarrow \mathcal{K}_\lambda$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $(y, \pi) \leftarrow \overline{S}(crs, (k, \sigma))$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_5$
    $k \leftarrow \mathcal{K}_\lambda$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $r_{\overline{S}} \leftarrow \{0,1\}^*$
    $(y, \pi) \leftarrow \overline{S}(crs, (k, \sigma); r_{\overline{S}})$
    $r'_{\overline{S}} \leftarrow \overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$
    $r'_X := \sigma \parallel r'_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_6$
    $k \leftarrow \mathcal{K}_\lambda$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $r_{\overline{S}} \leftarrow \{0,1\}^*$
    $(y, \pi) \leftarrow \overline{S}(crs, (k, \sigma); r_{\overline{S}})$

    $r'_X := \sigma \parallel r_{\overline{S}}$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

```
$\mathbf{G}_7$
    $k \leftarrow \mathcal{K}_\lambda$
    $\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
    $crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
    $r_{\overline{S}} \leftarrow \{0,1\}^*$
    $r'_X := \sigma \parallel r_{\overline{S}}$
    $y \leftarrow X(k, crs; r'_X)$
    $x' \leftarrow K_X(k, crs, r'_X)$
    $\mathbf{return}\ f_k(x') = y$
```

Fig. 24. Hybrids used in the proof of Theorem 17.

Claim. There exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_2 = 1] - \Pr[out_1 = 1]| \leq Adv^{\mathsf{zk}}_{\overline{\mathcal{A}}}(\lambda)$.

Proof. Construct adversary $\overline{\mathcal{A}}$. Initially, $\overline{\mathcal{A}}$ receives σ as input and produces k, x, y, crs as in game $\mathbf{G}_1$ and outputs the statement (k, y) together with a witness x to the experiment. In return, $\overline{\mathcal{A}}$ receives a proof π and proceeds as in $\mathbf{G}_1$ to produce $r'_{\overline{S}}, r'_X, x'$. Finally, $\overline{\mathcal{A}}$ outputs 1 if $f_k(x') = y$ and 0 otherwise. If $\overline{\mathcal{A}}$ plays experiment $Exp^{\mathsf{zk}}_{\overline{\mathcal{A}},1}$, σ was sampled along with some trapdoor τ and the proof π is simulated, hence, $\Pr[out_1 = 1] = \Pr[Exp^{\mathsf{zk}}_{\overline{\mathcal{A}},1}(\lambda) = 1]$. If, on the other hand, $\overline{\mathcal{A}}$ plays experiment $Exp^{\mathsf{zk}}_{\overline{\mathcal{A}},0}$, σ is uniformly random from $\{0,1\}^{\ell(\lambda)}$ and the proof π is produced by P using the given witness, hence, $\Pr[out_2 = 1] = \Pr[Exp^{\mathsf{zk}}_{\overline{\mathcal{A}},0}(\lambda) = 1]$. □

The difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ is again only conceptual and $\Pr[out_2 = 1] = \Pr[out_3 = 1]$.

Claim. There exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_4 = 1] - \Pr[out_3 = 1]| \leq Adv^{\mathsf{crs\text{-}close}}_{\overline{\mathcal{A}}}(\lambda)$.

Proof. Construct a PPT adversary $\overline{\mathcal{A}}$ on (static) closeness. Initially, $\overline{\mathcal{A}}$ produces (k, σ) as in game $\mathbf{G}_3$ and outputs them to the experiment. In the second phase, $\overline{\mathcal{A}}$ receives crs and (y, π), where (y, π) has either been sampled using $S(k, \sigma)$ or using $\overline{S}(crs, (k, \sigma))$. Further, $\overline{\mathcal{A}}$ proceeds as in $\mathbf{G}_3$ producing $r'_{\overline{S}}, r'_X, x'$. Finally, $\overline{\mathcal{A}}$ outputs $f_k(x') = y$. Hence, $\Pr[out_3 = 1] = \Pr[Exp^{\mathsf{crs\text{-}close}}_{\overline{\mathcal{A}},0}(\lambda) = 1]$ and $\Pr[out_4 = 1] = \Pr[Exp^{\mathsf{crs\text{-}close}}_{\overline{\mathcal{A}},1}(\lambda) = 1]$. □

The difference between games $\mathbf{G}_4$ and $\mathbf{G}_5$ is again only conceptual, hence, $\Pr[out_4 = 1] = \Pr[out_5 = 1]$.

Claim. There exists a PPT adversary $\overline{\mathcal{A}}$, such that $|\Pr[out_6 = 1] - \Pr[out_5 = 1]| \leq Adv^{\mathsf{crs\text{-}inv}}_{\overline{\mathcal{A}}}(\lambda)$.

Proof. Construct a PPT adversary $\overline{\mathcal{A}}$ on (static) invertibility. Initially, $\overline{\mathcal{A}}$ (statically) produces (k, σ) as in $\mathbf{G}_5$ and outputs them to the experiment. In the second phase, $\overline{\mathcal{A}}$ receives $(crs, r^*, (y, \pi))$, where (y, π) has been sampled using $\overline{S}(crs, (k, \sigma))$ and r^* either is the actual randomness used for that or the inverse sampled randomness produced via $\overline{S}^{-1}(crs, (k, \sigma), (y, \pi))$. Afterwards, $\overline{\mathcal{A}}$ proceeds as in $\mathbf{G}_5$ producing $r'_X := \sigma \parallel r^*$ and x'. Finally, $\overline{\mathcal{A}}$ outputs $f_k(x') = y$. Hence, $\Pr[out_5 = 1] = \Pr[Exp^{\mathsf{crs\text{-}inv}}_{\overline{\mathcal{A}},1}(\lambda) = 1]$ and $\Pr[out_6 = 1] = \Pr[Exp^{\mathsf{crs\text{-}inv}}_{\overline{\mathcal{A}},0}(\lambda) = 1]$. □

The difference between games $\mathbf{G}_6$ and $\mathbf{G}_7$ is again only conceptual, hence, $\Pr[out_6 = 1] = \Pr[out_7 = 1]$.

Thus, we have that $|\Pr[out_0 = 1] - \Pr[out_7 = 1]|$ is negligible.

Lemma 16. $\Pr[out_7 = 1]$ *is overwhelming.*

Proof (of Lemma 16). Since $\mathcal{F}$ is an extractable one-way function, we have that $\Pr_{\mathbf{G}_7}[f_k(x') = y \vee (k, y) \notin L_\lambda] \leq \Pr_{\mathbf{G}_7}[f_k(x') = y] + \Pr_{\mathbf{G}_7}[(k, y) \notin L_\lambda]$ is overwhelming. In the following, we prove that $\Pr_{\mathbf{G}_7}[(k, y) \notin L_\lambda]$ is negligible. We proceed over a series of hybrids, see Figures 25 and 26.

$\mathbf{H}_0$	$\mathbf{H}_1$	$\mathbf{H}_2$
$k \leftarrow \mathcal{K}_\lambda$	$k \leftarrow \mathcal{K}_\lambda$	$k \leftarrow \mathcal{K}_\lambda$
$\sigma \parallel r_{\overline{S}} = r_X \leftarrow \{0,1\}^{\ell(\lambda)}$	$\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$	$\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$y = X(k, crs; r_X)$	$(y, \pi) = \overline{S}(crs, (k, \sigma))$	$(y, \pi) = \overline{S}(crs, (k, \sigma))$
return $(k, y) \notin L_\lambda$	**return** $(k, y) \notin L_\lambda$	**return** $(k, y) \notin L_\lambda \wedge V(\sigma, (k, y), \pi) = 0$

Fig. 25. Hybrids used in the proof of Lemma 16.

The individual game hops are justified as follows. The difference between $\mathbf{H}_0$ and $\mathbf{H}_1$ is only conceptual and $\Pr[out_{\mathbf{H}_0} = 1] = \Pr[out_{\mathbf{H}_1} = 1]$.

```
$\mathbf{H}_3$
$k \leftarrow \mathcal{K}_\lambda$
$\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$(y, \pi) = \overline{S}(crs, (k, \sigma))$
return $V(\sigma, (k, y), \pi) = 0$
```

```
$\mathbf{H}_4$
$k \leftarrow \mathcal{K}_\lambda$
$\sigma \leftarrow \{0,1\}^{\ell(\lambda)}$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$(y, \pi) = S((k, \sigma))$
return $V(\sigma, (k, y), \pi) = 0$
```

Fig. 26. Hybrids used in the proof of Lemma 16.

Claim. There exists a PPT adversary $\overline{\mathcal{A}}$ such that $|\Pr[out_{\mathbf{H}_2} = 1] - \Pr[out_{\mathbf{H}_1} = 1]| \leq Adv_{\overline{\mathcal{A}}}^{\mathsf{soundness}}(\lambda)$.

Proof. The only possibility that the output of $\mathbf{H}_1$ and $\mathbf{H}_2$ differ is the case that $(k, y) \notin L_\lambda$ but $V(\sigma, (k, y), \pi) = 1$ (assuming, that the output of V is binary). Hence, $|\Pr[out_{\mathbf{H}_2} = 1] - \Pr[out_{\mathbf{H}_1} = 1]| \leq \Pr_{\mathbf{H}_1}[(k, y) \notin L_\lambda \wedge V(\sigma, (k, y), \pi) = 1]$.

Construct adversary $\overline{\mathcal{A}}$ on soundness of the NIZK proof system. On input of σ, $\overline{\mathcal{A}}$ produces $k, crs, (y, \pi)$ as in $\mathbf{H}_1$ and outputs the statement (k, y) and the proof π. Thus, $\Pr_{\mathbf{H}_1}[(k, y) \notin L_\lambda \wedge V(\sigma, (k, y), \pi) = 1] \leq Adv_{\overline{\mathcal{A}}}^{\mathsf{soundness}}(\lambda)$. □

The distribution underlying $\mathbf{H}_2$ and $\mathbf{H}_3$ is identical. The only difference between these games is that the condition that $\mathbf{H}_3$ outputs 1 is less restrictive than the one of $\mathbf{H}_2$, hence, $\Pr[out_{\mathbf{H}_2} = 1] \leq \Pr[out_{\mathbf{H}_3} = 1]$.

In contrast to the preceding games, the simulation of game $\mathbf{H}_3$ is efficient.

Claim. There exists a PPT adversary $\overline{\mathcal{A}}$ such that $|\Pr[out_{\mathbf{H}_4} = 1] - \Pr[out_{\mathbf{H}_3} = 1]| \leq Adv_{\overline{\mathcal{A}}}^{\mathsf{crs\text{-}close}}(\lambda)$.

Proof. Construct a PPT adversary $\overline{\mathcal{A}}$ on static closeness. Initially, $\overline{\mathcal{A}}$ produces (k, σ) as in $\mathbf{H}_3$ and outputs it to the experiment. On input of $(crs, \overline{y} := (y, \pi))$, where $\overline{y}$ is either sampled via $S(k, \sigma)$ or via $\overline{S}(crs, (k, \sigma))$. Finally, $\overline{\mathcal{A}}$ outputs $V(\sigma, (k, y), \pi) = 0$. We have that $\Pr[out_{\mathbf{H}_3} = 1] = \Pr[Exp_{\overline{\mathcal{A}},1}^{\mathsf{crs\text{-}close}}(\lambda) = 1]$ and $\Pr[out_{\mathbf{H}_4} = 1] = \Pr[Exp_{\overline{\mathcal{A}},0}^{\mathsf{crs\text{-}close}}(\lambda) = 1]$. □

Finally, due to the definition of S and the correctness of the NIZK proof system (P, V), $\Pr[out_{\mathbf{H}_4} = 1] = 0$. This proves that $\Pr[out_{\mathbf{H}_0} = 1]$ is negligible. □

This concludes the proof. □

By an easy modification of the above proof, we can refute universal $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$.

Theorem 18. *If there exist extractable one-way function family ensembles with unbounded common auxiliary information and NIZK proof systems for $\mathcal{NP}$, then universal $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ is false.*

We stress that assuming *unbounded* common auxiliary input is necessary since the adversary X can be considered the universal adversary. On a technical level, the common auxiliary input which leads to a contradiction is an output of the universal setup algorithm Setup on input of a sufficiently large bound B on the supported circuit size. This bound B depends on the adversary size.

Discussion. We stress that we are able to prove that EOWFs with unbounded common auxiliary input (in conjunction with NIZK proof systems) implies that $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ is false. Furthermore, due to Theorem 20 and Remark 11 in Section 6.3, $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ can be instantiated from indistinguishability obfuscation and one-way functions. We restate a theorem from [BCPR14].

Theorem 19 ([BCPR14]). *Assuming indistinguishability obfuscation for all circuits, neither EOWFs nor GEOWFs exist with respect to unbounded common auxiliary information.*

Thus, the above is not a contradiction because due to Theorem 19, indistinguishability obfuscation for all polynomial sized circuits does not exist assuming EOWFs with unbounded common auxiliary input.

Common but benign auxiliary information. The definition due to [BCPR14] of (G)EOWFs with common auxiliary input requires that extractability holds for all common auxiliary inputs and hence also for a *worst-case* choice of common auxiliary input. This requirement can be weakened such that the common auxiliary input is drawn from some specific distribution. This distribution is called *benign* if it is unlikely that a common auxiliary input sampled according to this distribution encodes a malicious obfuscation[15]. In particular, the uniform distribution over $\{0,1\}^{b(\lambda)}$ is conjectured to be benign. This notion of (G)EOWFs with common but benign auxiliary information does not contradict IO.

However, by an easy modification of the proof of Theorem 17, the existence (G)EOWFs with common but benign auxiliary input (drawn uniformly at random from $\{0,1\}^{b(\lambda)}$) contradicts $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ with common *random* string.

Corollary 8. *If there exist extractable one-way function family ensembles with common* but benign *auxiliary information, particularly for auxiliary inputs drawn uniformly at random from* $\{0,1\}^{b(\lambda)}$*, and NIZK proof systems for* $\mathcal{NP}$*, then* $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$*, where the setup algorithm produces uniform random strings, is false.*

Remark 10. [CsW19] pose the question whether adaptive MPC is possible in the global common random string model. Corollary 8 in combination with Theorem 7 answer this question negatively assuming extractable one-way functions with common *but benign* auxiliary input.

Key-less extractable one-way function ensembles. Definitions 27, 29 and 31 can be defined for *key-less* function ensembles, where the key space $\mathcal{K}_\lambda$ contains exactly one element (for every λ). This poses much stronger requirements on one-wayness and extractability since these properties are bound to be met for one fixed key as opposed to the keyed variants.

As observed in [IKOS10], Theorems 14 and 16 can be adapted to the *weak* variants of $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$ and $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_s}$, respectively.[16] However, key-less (G)EOWFs with unbounded auxiliary input are impossible since the adversary might get a random image of the (fixed) (G)EOWF as auxiliary input. An extractor given the output of the adversary together with its random tape and the same auxiliary input must break one-wayness in order to produce a pre-image. Hence, Theorem 17 can not be adapted to refute weak $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$.

6.3 Static pseudorandom encodings with universal setup from IO

$\mathsf{cISH}^{\mathsf{rand}}_{\approx_c}$ (and hence $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$) is implied by the existence of an explainability compiler [DKR15] which can be built from indistinguishability obfuscation and one-way functions. Using the ideas from [SW14; DKR15], we obtain perfectly correct $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ with universal setup. Let S be a PPT algorithm and let $U_B(C,x;r)$ be the universal circuit that accepts any circuit C which can be represented with B bits and evaluates C on input x and randomness r.

Let B be an upper bound on the bitlength which is necessary to describe a sampler. Let $\ell_{in} = |m|$ be an upper bound on the bitlength of the inputs, ℓ_{out} be an upper bound on bitlength of the outputs and ℓ_r be an upper bound on the bitlength of random tape of such samplers. Let PRG be a PRG that maps $\{0,1\}^\lambda$ to $\{0,1\}^{2\lambda}$. Let $|u_1| = \ell_1 = 2B + 2\ell_{in} + 2\ell_{out} + 5\lambda$, $|u_2| = \ell_2 = B + \ell_{in} + \ell_{out} + 2\lambda$.

Theorem 20. *Let* iO *be a perfectly correct indistinguishability obfuscator,* F_1, F_2, F_3 *be puncturable PRFs satisfying the following additional properties*

[15] By malicious obfuscation we mean an obfuscated circuit which renders extraction from an adversary which simply executes the obfuscated circuit on the given key infeasible, see [BCPR14].

[16] For weak $\mathsf{PREH}^{\mathsf{rand}}_{\approx_c}$, the NIZK proof system must be replaced by a NIWI proof system (without CRS) and therefore the sampler S samples two images of the EOWF and a NIWI proof using one of the preimages as witness.

```
$\mathsf{Setup}(1^\lambda, B)$
    $\Lambda_E \leftarrow \mathsf{iO}(C_E[k_2, k_2])$
    $\Lambda_D \leftarrow \mathsf{iO}(C_D[k_1, k_2, k_2])$
    return $crs := (\Lambda_E, \Lambda_D)$
```

```
$\mathsf{E}_S(crs, m, y; r)$
    return $\Lambda_E(S, m, y, r)$
```

```
$\mathsf{D}_S(crs, m, u)$
    return $\Lambda_D(S, m, u)$
```

```
$C_E[k_2, k_3](S, m, y, r)$
    $e_1 := F_2(S, m, y, \mathsf{PRG}(r))$
    $e_2 := F_3(e_1) \oplus (S, m, y, \mathsf{PRG}(r))$
    return $(e_1, e_2)$
```

```
$C_D[k_1, k_2, k_3](S, m, u)$
    $(S', m', y', r') := F_2(u_1) \oplus u_2$
    if $\big((S', m') = (S, m) \wedge$
        $u_1 = F_2(S', m', y', r')\big)$ then
      return $y'$
    $x := F_1(S, m, u)$
    return $U_B(S, m; x)$
```

Fig. 27. Instantiation of perfectly correct $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ due to [SW14; DKR15].

- F_1 *is extracting when the input min-entropy is greater than* $\ell_r + 2(\lambda+1) + 2$ *with error less than* $2^{-(\lambda+1)}$ *and has input length* $\ell_1 + \ell_2 + \ell_{in} + B$ *and output length* ℓ_r *(such a PRF exists from one-way functions since* $\ell_1 + \ell_2 + \ell_{in} + B \geq \ell_r + 2(\lambda+1) + 2$*),*
- F_2 *is statistically injective and has input length* $B + \ell_{in} + \ell_{out} + 2\lambda$ *and output length* ℓ_1 *(such a PRF exists from one-way functions since* $\ell_1 \geq 2(B + \ell_{in} + \ell_{out} + 2\lambda) + \lambda$*),*
- F_3 *has input length* ℓ_1 *and output length* ℓ_2*.*

Then, perfectly correct $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ *with universal setup is true.*

Proof. **Perfect correctness.** Let $crs =: (\Lambda_E, \Lambda_D)$ be parameters in the support of $\mathsf{Setup}(1^\lambda, B)$. Let S be a PPT algorithm represented as a polynomial sized circuit, $m \in L$ be an input for S and let $y \in \mathsf{supp}(S(m))$. Due to perfect correctness of iO we have

$$\begin{aligned}
&\mathsf{D}_S(crs, m, \mathsf{E}_S(crs, m, y; r))\\
&= C_D(S, m, C_E(S, m, y, r))\\
&= C_D(S, m, (\underbrace{F_2(S, m, y, \mathsf{PRG}(r))}_{=e_1}, \underbrace{F_3(e_1) \oplus (S, m, y, \mathsf{PRG}(r))}_{=e_2})) = y.
\end{aligned}$$

Pseudorandomness. The proof can be divided into two main steps, see Figure 28.

```
$\mathbf{G}_0$
    $m^* \leftarrow \mathcal{A}(1^\lambda)$
    sample $k_1, k_2, k_3$
    $r^* \leftarrow \{0,1\}^*$
    $x^* \leftarrow \{0,1\}^{p(\lambda)}$
    $y^* := S^*(m^*; x^*)$
    $e_1^* := F_2(S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $e_2^* := F_3(e_1^*) \oplus (S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $\Lambda_E \leftarrow \mathsf{iO}(C_E[k_2, k_3])$
    $\Lambda_D \leftarrow \mathsf{iO}(C_D[k_1, k_2, k_3])$
    $crs := (\Lambda_E, \Lambda_D)$
    return $\mathcal{A}(crs, e^*)$
```

```
$\mathbf{G}_1$
    $m^* \leftarrow \mathcal{A}(1^\lambda)$
    sample $k_1, k_2, k_3$
    $r^*, u^* \leftarrow \{0,1\}^*$
    $x^* := F_1(S^*, m^*, u^*)$
    $y^* := S^*(m^*; x^*)$
    $e_1^* := F_2(S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $e_2^* := F_3(e_1^*) \oplus (S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $\Lambda_E \leftarrow \mathsf{iO}(C_E[k_2, k_3])$
    $\Lambda_D \leftarrow \mathsf{iO}(C_D[k_1, k_2, k_3])$
    $crs := (\Lambda_E, \Lambda_D)$
    return $\mathcal{A}(crs, e^*)$
```

```
$\mathbf{G}_2$
    $m^* \leftarrow \mathcal{A}(1^\lambda)$
    sample $k_1, k_2, k_3$
    $r^*, u^* \leftarrow \{0,1\}^*$
    $x^* := F_1(S^*, m^*, u^*)$
    $y^* := S^*(m^*; x^*)$
    $e_1^* := F_2(S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $e_2^* := F_3(e_1^*) \oplus (S^*, m^*, y^*, \mathsf{PRG}(r^*))$
    $\Lambda_E \leftarrow \mathsf{iO}(C_E[k_2, k_3])$
    $\Lambda_D \leftarrow \mathsf{iO}(C_D[k_1, k_2, k_3])$
    $crs := (\Lambda_E, \Lambda_D)$
    return $\mathcal{A}(crs, u^*)$
```

Fig. 28. Hybrids used in the proof of pseudorandomness for Theorem 20.

Lemma 17. *For all (potentially unbounded) adversaries* $\mathcal{A}$*,* $|\Pr[out_1 = 1] - \Pr[out_0 = 1]|$ *is negligible.*

The proof works as the proof of IND-CPA security of the deniable encryption scheme from [SW14] or the proof of statistical functional equivalence from [DKR15].

Lemma 18. *For all PPT adversaries $\mathcal{A}$, $|\Pr[out_2 = 1] - \Pr[out_1 = 1]|$ is negligible.*

The proof is similarly as the proof of explainability of the deniable encryption scheme from [SW14] or the proof of explainability from [DKR15]. □

Remark 11. By allowing the setup algorithm Setup_S to depend on S and replacing the universal circuit in C_D with S, we obtain $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ without universal setup.

Theorem 2 together with [CPR16], we obtain the following corollary.

Corollary 9. *Assuming polynomially secure IO for all circuits and one-way functions, then $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ (with universal setup) is true.*

6.4 Bootstrapping pseudorandom encodings with a common random string

Theorem 21. *Assume (i) $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with universal setup Setup is true for all PPT samplers (up to some fixed bound on their size when represented as circuits) and (ii) weak $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ is true for the PPT algorithm Setup such that the corresponding setup algorithm $\mathsf{Setup}'_{\mathsf{Setup}}$ produces uniform random strings. Then, $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with universal setup Setup'' producing uniform random strings is true for all PPT samplers.*

Proof. (i) implies the existence of a universal setup algorithm Setup such that for all PPT algorithms S, there exist efficient algorithms $(\mathsf{E}_S, \mathsf{D}_S)$ satisfying correctness and pseudorandomness. Further, (ii) guarantees the existence of a setup algorithm $\mathsf{Setup}'_{\mathsf{Setup}}$ producing uniform random strings such that for the sampler Setup there are efficient algorithms $(\mathsf{E}'_{\mathsf{Setup}}, \mathsf{D}'_{\mathsf{Setup}})$ satisfying correctness and pseudorandomness. Let $\{0,1\}^{n'(\lambda)}$ be the range of $\mathsf{E}'_{\mathsf{Setup}}$ (note that $\{0,1\}^{n'(\lambda)}$ is allowed to depend on the sampler Setup).

We define the (universal) setup algorithm Setup'' as in Figure 29. Let S be some PPT algorithm. We define $(\mathsf{E}''_S, \mathsf{D}''_S)$ corresponding to S as in Figure 29.

```
$\mathsf{Setup}''(1^\lambda)$
$crs' \leftarrow \mathsf{Setup}'(1^\lambda)$
$u \leftarrow \{0,1\}^{n'(\lambda)}$
$crs'' := crs' \,\|\, u$
return $crs''$
```

```
$\mathsf{E}''_S(crs'', m, y)$
parse $crs'' =: crs' \,\|\, u$
$crs := \mathsf{D}'_{\mathsf{Setup}}(crs', u)$
$u'' \leftarrow \mathsf{E}_S(crs, m, y)$
return $u''$
```

```
$\mathsf{D}''_S(crs'', m, u'')$
parse $crs'' =: crs' \,\|\, u$
$crs := \mathsf{D}'_{\mathsf{Setup}}(crs', u)$
$y \leftarrow \mathsf{D}_S(crs, m, u'')$
return $y$
```

Fig. 29. Algorithms for $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with common *random* string.

Correctness. We use a sequence of two games, see Figure 30. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ is only conceptual and thus $\Pr[out_0 = 1] = \Pr[out_1 = 1]$. Due to correctness (i), for all PPT adversaries $\mathcal{A}$, $\Pr[m \leftarrow \mathcal{A}(1^\lambda), crs \leftarrow \mathsf{Setup}(1^\lambda), y \leftarrow S(m) \colon \mathsf{D}_S(crs, m, \mathsf{E}_S(crs, m, y)) = y]$ and, hence, $\Pr[out_2 = 1]$ are overwhelming. Due to Lemma 5, by correctness and pseudorandomness

```
$\mathbf{G}_0$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs'' \leftarrow \mathsf{Setup}''(1^\lambda)$
$y \leftarrow S(m)$
$u'' \leftarrow \mathsf{E}''_S(crs, m, y)$
$y'' := \mathsf{D}''_S(crs, m, u'')$
return $y = y''$
```

```
$\mathbf{G}_1$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda)$
$u \leftarrow \{0,1\}^{n'(\lambda)}$
$y \leftarrow S(m)$
$crs := \mathsf{D}'_{\mathsf{Setup}}(crs', u)$
$u'' \leftarrow \mathsf{E}_S(crs, m, y)$
$y'' \leftarrow \mathsf{D}_S(crs, m, u'')$
return $y = y''$
```

```
$\mathbf{G}_2$
$m \leftarrow \mathcal{A}(1^\lambda)$
$crs' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda)$
$y \leftarrow S(m)$
$crs \leftarrow \mathsf{Setup}(1^\lambda)$
$u'' \leftarrow \mathsf{E}_S(crs, m, y)$
$y'' \leftarrow \mathsf{D}_S(crs, m, u'')$
return $y = y''$
```

Fig. 30. Hybrids used in the proof of correctness of Theorem 21.

(ii), the distributions

$$\{\mathit{crs}' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda), \mathit{crs} \leftarrow \mathsf{Setup}(1^\lambda) \colon (\mathit{crs}', \mathit{crs})\}$$
$$\text{and } \{\mathit{crs}' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda), \mathit{crs} \leftarrow \mathsf{D}'_{\mathsf{Setup}}(\mathit{crs}', U_{n'(\lambda)}) \colon (\mathit{crs}', \mathit{crs})\}$$

are computationally indistinguishable. Hence, $|\Pr[out_2 = 1] - \Pr[out_1 = 1]|$ is negligible.
Pseudorandomness. Let $\mathcal{A}$ be an adversary on pseudorandomness. We proceed over a sequence of three hybrids, see Figure 31.

$\mathbf{G}_0^b$	$\mathbf{G}_1^b$	$\mathbf{G}_2^b$
$m \leftarrow \mathcal{A}(1^\lambda)$	$m \leftarrow \mathcal{A}(1^\lambda)$	$m \leftarrow \mathcal{A}(1^\lambda)$
$\mathit{crs}' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda)$	$\mathit{crs}' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda)$	$\mathit{crs}' \leftarrow \mathsf{Setup}'_{\mathsf{Setup}}(1^\lambda)$
$u \leftarrow \{0,1\}^{n'(\lambda)}$	$\mathit{crs} \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathit{crs} \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathit{crs}'' := \mathit{crs}' \parallel u$	$u \leftarrow \mathsf{E}'_{\mathsf{Setup}}(\mathit{crs}', \mathit{crs})$	$u \leftarrow \mathsf{E}'_{\mathsf{Setup}}(\mathit{crs}', \mathit{crs})$
$\mathit{crs} := \mathsf{D}'_{\mathsf{Setup}}(\mathit{crs}', u)$	$\mathit{crs}'' := \mathit{crs}' \parallel u$	$\mathit{crs}'' := \mathit{crs}' \parallel u$
$y \leftarrow S(m)$	$y \leftarrow S(m)$	$y \leftarrow S(m)$
$u_0'' \leftarrow \mathsf{E}_S(\mathit{crs}, m, y)$	$u_0'' \leftarrow \mathsf{E}_S(\mathit{crs}, m, y)$	$u_0'' \leftarrow \{0,1\}^{n(\lambda)}$
$u_1'' \leftarrow \{0,1\}^{n(\lambda)}$	$u_1'' \leftarrow \{0,1\}^{n(\lambda)}$	$u_1'' \leftarrow \{0,1\}^{n(\lambda)}$
return $\mathcal{A}(\mathit{crs}'', u_b'')$	**return** $\mathcal{A}(\mathit{crs}'', u_b'')$	**return** $\mathcal{A}(\mathit{crs}'', u_b'')$

Fig. 31. Hybrids used in the proof of pseudorandomness of Theorem 21.

For all PPT adversaries $\mathcal{A}$, $|\Pr[out_1^b = 1] - \Pr[out_0^b = 1]|$ is negligible due correctness and pseudorandomness (ii). The proof follows from an argument already made to prove Theorem 1, more precisely the argument used to prove invertibility (the indistinguishability between Game1 and Game5).

For all PPT adversaries $\mathcal{A}$, $|\Pr[out_2^b = 1] - \Pr[out_1^b = 1]|$ is negligible due to psdueorandomness (i). Further, $\Pr[out_2^0 = 1] = \Pr[out_2^1 = 1]$ which concludes the proof of indistinguishability between $\mathbf{G}_0^0$ and $\mathbf{G}_0^1$. □

Combining Theorem 21 and Corollary 8, we obtain the following corollary refuting weak $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ with common random string.

Corollary 10. *If there exist extractable one-way function family ensembles with common but benign auxiliary information and indistinguishability obfuscation for all circuits, then there exists a PPT algorithm A such that* weak $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ *for A, where* Setup_A *produces uniform random strings, is false.*

Remark 12. If weak $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$ was true such that the setup algorithms always produce uniform random strings, then the global common reference string model could be replaced with the global common random string model (or the non-programmable random oracle model) in every setting in cryptography.

7 Relations

In this section we describe further applications of pseudorandom encodings. This unifies several areas in cryptography. In Section 7.1, we show that pseudorandom encodings yields honey encryption due to [JR14] for arbitrary message distributions, even such which admit inputs. In Section 7.2, we define a keyless version of steganography. That is, in contrast to symmetric-key or public-key steganography, parties do not need any secret information in order to covertly communicate with each other. In Section 7.3, we define the notion of covert secure computation and give a general compiler which transforms secure computation protocol into a covert MPC protocol. In Section 7.4, we analyze the relation of pseudorandom encodings and certain PKE variants which are known to imply (fully) adaptive MPC.

7.1 Honey encryption

Honey encryption schemes [JR14] are symmetric key encryption schemes which offer security even against unbounded adversaries, i.e. even against adversaries which are able find the used symmetric key. This is a particularly useful notion when the key comes from a distribution with low min-entropy like passwords. On decryption with a wrong key, a honey encryption scheme behaves indistinguishably from decryption with the actually used key.

Let D_1 be an efficiently samplable key distribution over the key space $\mathcal{K}_\lambda$ and let D_2 be an efficiently samplable message distribution over the message space $\mathcal{M}_\lambda$.

Definition 33 (Honey encryption, [JR14]). *A symmetric encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *is called* honey encryption scheme for key distribution D_1 and plaintext distribution D_2 *if it satisfies the following property.*

Security against message recovery (with respect to D_1, D_2). *For all unbounded adversaries $\mathcal{A}$, the advantage*

$$Adv^{\mathsf{mr}}_{\mathcal{A},D_1,D_2}(\lambda) = \Pr[Exp^{\mathsf{mr}}_{\mathcal{A},D_1,D_2}(\lambda) = 1]$$

is negligibly close to $2^{-\mu_1}$, where $\mu_1 = \mathrm{H}_\infty(D_1)$ and $Exp^{\mathsf{mr}}_{\mathcal{A},D_1,D_2}$ is defined as in Figure 32.

```
$Exp^{\mathsf{mr}}_{\mathcal{A},D_1,D_2}(\lambda)$
$k \leftarrow D_1(1^\lambda)$
$m^* \leftarrow D_2(1^\lambda)$
$c^* \leftarrow \mathsf{Enc}(k, m^*)$
$m \leftarrow \mathcal{A}(c^*)$
return $m = m^*$
```

```
$Exp^{\mathsf{dte}}_{\mathcal{A},0}(\lambda)$
$y^* \leftarrow S(1^\lambda)$
$u^* \leftarrow \mathsf{E}_S(m^*)$
return $\mathcal{A}(u^*)$
```

```
$Exp^{\mathsf{dte}}_{\mathcal{A},1}(\lambda)$
$u^* \leftarrow \{0,1\}^{n(\lambda)}$
$y^* := \mathsf{D}_S(u^*)$
return $\mathcal{A}(u^*)$
```

Fig. 32. Message recovery experiment and DTE security experiment.

Definition 34 (Distribution-transforming encoder, [JR14; JRT16]). *A* distribution-transforming *encoder (DTE) for a distribution sampled by a sampler S over $\mathcal{Y}_{S,\lambda}$ is a tuple of efficient algorithms* $(\mathsf{E}_S, \mathsf{D}_S)$*, where* D_S *is deterministic, such that the following properties are satisfied.*

Perfect correctness. *For all $y \in \mathcal{Y}_{S,\lambda}$,* $\Pr[\mathsf{D}_S(\mathsf{E}_S(m)) = m] = 1$*, where the probability is over the randomness of* E_S*.*

DTE security. *For all unbounded adversaries, $Adv^{\mathsf{dte}}_{\mathcal{A}}(\lambda) := |\Pr[Exp^{\mathsf{dte}}_{\mathcal{A},0}(\lambda) - Exp^{\mathsf{dte}}_{\mathcal{A},1}(\lambda)]|$ is negligible, where $Exp^{\mathsf{dte}}_{\mathcal{A},0}$ and $Exp^{\mathsf{dte}}_{\mathcal{A},1}$ are defined in Figure 32.*

The above notions are defined with respect to unbounded adversaries. This is a simplification meant to capture the fact that an adversary is able to perform work proportional to $2^{-\mu_1}$. For low-entropic key distributions, a computational variant of the above definition suffices. We refer to this as *computational* honey encryption.

Relaxing the perfect correctness requirement from Definition 34 recovers the definition of (weak) $\mathsf{PREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}$ (or weak $\mathsf{PREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ in the computational case) and still suffices to imply honey encryption via the DTE-then-Encrypt framework due to [JR14] given a secret-key encryption scheme (with message space matching the range of E_S). Applying the DTE-then-Encrypt framework on $\mathsf{acPREH}^{\mathsf{rand}}_{\equiv_\mathsf{s}}$ and $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ yields honey encryption in a CRS model for high-entropic and low-entropic key distributions, respectively. Note that the adaptive versions are only necessary when considering message distributions D_2 with input.

7.2 Keyless steganography

Pseudorandom encodings yield a notion of keyless steganography. In the case of $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ all parties need access to some public parameters, but none of the parties needs access to a secret key. We adopt the notation from [BL18]. Since decoding does not involve any secret information, any attack corresponds to an equivalent of SS-CCA-security for public-key stegosystems (PKStS), where the decoding may even be applied on the challenge. Hence, the definition of SS-CCA-security must be adapted such that the message to be hidden is not chosen by the adversary but sampled according to a predefined message distribution. Let $dl(\lambda)$ be the document length and $ol(\lambda)$ be the output length.

Definition 35 (Keyless stegosystem (KlStS) for distribution S). *A keyless stegosystem for message distribution S is a triple of PPT algorithms* ($\mathsf{KlStS.Gen}, \mathsf{KlStS.E}_S, \mathsf{KlStS.D}_S$)*, where*

- $\mathsf{KlStS.Gen}(1^\lambda)$ *produces public parameters pp (without corresponding secret information),*
- $\mathsf{KlStS.E}_S$ *on input of pp, a message y sampled from S, a history $\mathit{hist} \in (\Sigma^{dl(\lambda)})^*$ and some state information $s \in \{0,1\}^*$, produces a document $d \in \Sigma^{dl}$ (by being able to sample from $C_{\lambda,dl(\lambda)}$).* $\mathsf{KlStS.E}_S^C(pp, m, \mathit{hist})$ *denotes sampling $ol(\lambda)$ documents using* $\mathsf{KlStS.E}_S$ *one-by-one.*
- $\mathsf{KlStS.D}_S$ *on input of pp, and a sequence of documents $d_1, \ldots, d_{ol(\lambda)}$, and outputs a message m'.*

We require KlStS *to meet the following properties.*

Universality. KlStS *works on any channel without prior knowledge of the distribution of the channel.*

Reliability. *The probability*

$$\Pr\left[\mathsf{KlStS.D}_S(pp, \mathsf{KlStS.E}_S(pp, m, \mathit{hist}), \mathit{hist}) \neq m\right]$$

that decoding fails is negligible, where the probability is over the choice of pp, m and the random coins of $\mathsf{KlStS.E}_S$*.*

Security. KlStS *is secure (on channel C), if for all PPT adversaries $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{\mathsf{klsts\text{-}sec}}(\lambda) := |\Pr[Exp_{\mathcal{A},C,0}^{\mathsf{klsts\text{-}sec}}(\lambda) = 1] - \Pr[Exp_{\mathcal{A},C,1}^{\mathsf{klsts\text{-}sec}}(\lambda) = 1]|$$

is negligible.

```
$Exp_{\mathcal{A},C,b}^{\mathsf{klsts\text{-}sec}}(\lambda)$
$pp \leftarrow \mathsf{KlStS.Gen}(1^\lambda)$
$m \leftarrow \mathcal{A}(pp)$
$m^* \leftarrow S(m)$
$\mathsf{hist}^* \leftarrow \mathcal{A}(1^\lambda, pp, m^*)$
$\boldsymbol{d}_0^* \leftarrow \mathsf{KlStS.E}_S^C(pp, m^*, \mathsf{hist}^*)$
$\boldsymbol{d}_1^* \leftarrow C_{\lambda,dl(\lambda),\mathsf{hist}^*}^{ol(\lambda)}$
return $\mathcal{A}(\boldsymbol{d}_b^*)$
```

```
$\mathsf{KlStS.Setup}_S(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_S(1^\lambda)$
$H \leftarrow \mathcal{H}$
return $pp := (crs, H)$
```

```
$\mathsf{KlStS.E}_S((crs, H), m)$
$u \leftarrow \mathsf{E}_S(crs, m)$
$u_1 \| \cdots \| u_\ell =: u$
for $i \in [\ell]$ do
    do
        $d_i \leftarrow C(\lambda, hist)$
    until $prefix_m(H(d_i)) == u_i$
return $(d_1, \ldots, d_\ell)$
```

```
$\mathsf{KlStS.D}_S((crs, H), (d_1, \ldots, d_\ell))$
for $i \in [\ell]$ do
    $u_i := prefix_m(H(d_i))$
$u := u_1 \| \cdots \| u_\ell$
return $\mathsf{D}_S(crs, u)$
```

Fig. 33. Definition of $Exp_{\mathcal{A},C,0}^{\mathsf{klsts\text{-}sec}}(\lambda)$ and description of a keyless stegosystem KlStS.

Applying a similar strategy as in [vH04; Hop05], we obtain the following theorem.

Theorem 22. *Let S be an efficiently samplable message distribution and let $\mathcal{H}$ be a family of pairwise independent hash functions. If* $\mathsf{acPREH}_{\approx_c}^{\mathsf{rand}}$ *is true for S, then* ($\mathsf{KlStS.Gen}_S, \mathsf{KlStS.E}_S, \mathsf{KlStS.D}_S$) *defined above and in Figure 33 is a keyless stegosystem for the message distribution S.*

Functionality $\mathcal{F}_{\mathsf{CMPC}}(f, g)$

$\mathcal{F}_{\mathsf{CMPC}}$ runs with an adversary $\mathcal{A}$ and a set of n parties $(P_i)_{i \leq n}$, some of whome might not be actually taking part to the protocol, and knows the description of two n-entry functions f and g (the latter is the admissibility function). Before the protocol, $\mathcal{A}$ can *corrupt* a subset of the parties; the remaining parties are denoted *honest parties*. Honest non-participating parties are assumed to automatically send $\perp$ to the trusted party. The trusted party ignores all messages that are not correctly formatted.

Input. The trusted party waits until it received an input x_i from each party P_i.

Send output to adversary. Let $\boldsymbol{x} = (x_i)_{i \leq n}$. If any entry of $\boldsymbol{x}$ is $\perp$, $\mathcal{F}_{\mathsf{CMPC}}$ sets the output y to $\perp$. Else, it checks that $g(\boldsymbol{x}) = 1$, and sets y to $\perp$ if this check fails. If the check passes, it sets $y \leftarrow f(\boldsymbol{x})$. The trusted party sends y to $\mathcal{A}$.

Send output to honest parties. The trusted party waits until it receives from $\mathcal{A}$ a list of honest parties that should get the output. It sends y to those parties, and $\perp$ to the remaining honest parties.

The honest participating parties output whatever output they received from the functionality. Non-participating and corrupted parties output $\perp$ by convention. The adversary outputs its entire view.

Fig. 34. Ideal Functionality for Covert Multi-Party Computation.

The result basically follows from the Leftover Hash Lemma [HILL99] and the ability to embed the message distribution into the uniform distribution due to $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$. Note that since in $Exp^{\mathsf{klsts\text{-}sec}}_{\mathcal{A},C,b}(\lambda)$, the adversary does not know the challenge message m^*, it is not necessary that each encoding of a message corresponds to exactly one stegotext, see [Hop05].

7.3 Covert multi-party computation

The notion of covert secure computation was first introduced in [vHL05] as a collection of desirable properties, such as strong internal covertness, strong fairness and final covertness. Later in [CGOS07], Chandran et al. proposed a more unified, simulation-based definition. We start by recalling the model of [CGOS07].

Ideal Model. We consider n parties, each party P_i holding an input x_i. Let $\boldsymbol{x}$ be the vector $(x_1, \cdots, x_n)$. All the participating parties send their input to the functionality, while the non-participating parties are assumed to send $\perp$. Then, if any of the parties had input a $\perp$, the functionality sets $\perp$ to be the output of the protocol. Else, let $g : (\{0,1\}^*)^n \mapsto \{0,1\}$ be the function which determines whether on input $\boldsymbol{x}$, the output is *favorable* ($g(\boldsymbol{x}) = 1$) or *non-favorable* ($g(\boldsymbol{x}) = 0$). In the latter case, the functionality sets $\perp$ to be the output of the protocol. In the former case, the functionality computes the output $f(\boldsymbol{x})$. The output is sent to any subset of players, chosen by the adversary. The functionality is represented on Figure 34. For an adversary $\mathcal{A}$, an execution of $\mathcal{F}_{\mathsf{CMPC}}(f, g)$ with participation data $\boldsymbol{p}$ (indicating which players are taking part to the protocol) and input $\boldsymbol{x}$ (where the input of non-participating parties is $\perp$) is defined as the output of the parties together with the output of the adversary. It is denoted $\mathsf{IDEAL}_{f,g,\mathcal{A}}(\boldsymbol{p}, \boldsymbol{x})$.

Real Model. Honest participating parties follow the specifications of the protocol. Honest non-participating parties are assumed to send uniformly random messages. We consider *static semi-honest corruption* of players, in which the corrupted players are chosen once-for-all by the adversary before the start of the protocol, and follow the specifications of the protocol. Honest participating parties compute their output as specified, non-participating and corrupted parties output $\perp$ by convention, and the adversary outputs its entire view of the execution of the protocol. For an adversary $\mathcal{A}$, an execution of a protocol $\Pi(f, g)$ in the real model with participation data $\boldsymbol{p}$ and input $\boldsymbol{x}$ is defined as the output of the parties together with the output of the adversary. It is denoted $\mathsf{REAL}_{\Pi(f,g),\mathcal{A}}(\boldsymbol{p}, \boldsymbol{x})$.

Definition 36 (Covert security [CGOS07]). *A protocol $\Pi(f, g)$ securely implements $\mathcal{F}_{\mathsf{CMPC}}(f, g)$ if for every probabilistic polynomial-time adversary $\mathcal{A}$ statically corrupting up to $n - 1$ players* *in the real model, there is an expected polynomial-time adversary S corrupting at most $n-1$ players in the ideal model, such that for any $(\boldsymbol{p}, \boldsymbol{x}) \in \{0,1\}^n \times (\{0,1\}^*)^n$,*

$$\{\mathsf{IDEAL}_{f,g,S}(\boldsymbol{p}, \boldsymbol{x})\} \approx_{\mathsf{c}} \{\mathsf{REAL}_{\Pi(f,g),\mathcal{A}}(\boldsymbol{p}, \boldsymbol{x})\}$$

We now show that pseudorandom encodings enable to convert a large class of secure computation protocols (satisfying the standard static semi-honest security notion) into protocols secure in the covert model of [CGOS07]. Covertness is a very strong security notion, protecting the participants from even leaking the information that they took part to the protocol. Interestingly, since the pseudorandom encoding hypothesis is equivalent to the existence of adaptive MPC for all functionalities, our result create a link between two extreme but seemingly unrelated notions of secure computation: adaptive multi-party computation for all functionalities implies the existence of a generic compiler for covert multi-party computation. We find this link to be intriguing.

7.3.1 A compiler for covert oblivious transfer

As a warm-up, we start from a static, semi-honest oblivious transfer protocol, and show how to compile it into a covert OT using pseudorandom encodings. Consider a two-round OT, as follows:

$\mathsf{R}_{\mathsf{OT}}(1^\lambda, b; r_R)$ takes as input a bit b and randomness r_R and outputs the first protocol message OT_1.

$\mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1); r_S)$ takes as input the first message OT_1, a tuple $(y_0, y_1) \in (\{0,1\}^\lambda)^2$ and randomness r_S and outputs the second protocol message OT_2.

$\mathsf{E}_{\mathsf{OT}}(\mathsf{OT}_2, b, r_R)$ takes as input OT_2 the bit b and the randomness r_R and outputs $y \in \{0,1\}^\lambda$.

Covert oblivious transfer was defined in [vHL05]; a covert OT protocol realizes the standard OT functionality, and satisfies two additional properties:

- (receiver indistinguishability) For any b, the distribution induced by $\mathsf{R}_{\mathsf{OT}}(1^\lambda, b)$ is indistinguishable from the uniform distribution (over bistrings of an appropriate size);
- (sender indistinguishability) For any OT_1, the distribution

  $$\{\mathsf{OT}_2 \leftarrow \mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1)) \ : \ (y_0, y_1) \leftarrow (\{0,1\}^n)^2\}$$

  is indistinguishable from the uniform distribution (over bistrings of an appropriate size).

Now, let $(\mathsf{E}_1, \mathsf{D}_1)$ be a pseudorandom encoding for the sampler $\mathsf{R}_{\mathsf{OT}}(1^\lambda, b)$ which takes as input $(1^\lambda, b)$, and let $(\mathsf{E}_2, \mathsf{D}_2)$ be a pseudorandom encoding for the sampler $\mathsf{S}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1))$ which takes as input a string OT_1 and samples (y_0, y_1) at random. Consider the following covert OT protocol, built from an arbitrary two-round OT:

$\mathsf{covR}_{\mathsf{OT}}(1^\lambda, b; r_R)$ compute $a_1 \leftarrow \mathsf{R}_{\mathsf{OT}}(1^\lambda, b; r_R^0)$ and output $\mathsf{OT}_1 \leftarrow \mathsf{E}_1(a_1, 0; r_R^1)$.

$\mathsf{covS}_{\mathsf{OT}}(\mathsf{OT}_1, (y_0, y_1))$ compute $a_1' \leftarrow \mathsf{D}_1(\mathsf{OT}_1, 0)$, $a_2 \leftarrow \mathsf{S}_{\mathsf{OT}}(a_1', (y_0, y_1))$, and output $\mathsf{OT}_2 \leftarrow \mathsf{E}_2(a_2, a_1')$.

$\mathsf{covE}_{\mathsf{OT}}(\mathsf{OT}_2, b, r_R)$ reconstruct (a_1, OT_1) from $r_R = (r_R^0, r_R^1)$. compute $a_2' \leftarrow \mathsf{D}_2(\mathsf{OT}_2, a_1)$ and output $\mathsf{E}_{\mathsf{OT}}(a_2', b, r_R^0)$.

We now prove that the above protocol realizes the OT functionality and satisfies receiver and sender indistinguishability.

- (receiver indistinguishability) $\mathsf{E}_1(\mathsf{R}_{\mathsf{OT}}(1^\lambda, b), 0) \approx_{\mathsf{c}} \mathsf{E}_1(\mathsf{R}_{\mathsf{OT}}(1^\lambda, 0), 0) \approx_{\mathsf{c}} U$, where U denotes the uniform distribution (over bitstrings of some appropriate length). The first indistinguishability follows from the receiver security of the underlying semi-honest OT protocol, and the second follows from the pseudorandomness of E_1.

- (sender indistinguishability) follows directly from the pseudorandomness of E_2.
- (OT functionality) sender security and receiver security are trivially inherited from the underlying semi-honest OT: neither E_1 nor E_2 use any of the private inputs of the parties. It remains to verify that correctness is preserved. This follows from the receiver security of the underlying OT and the correctness of the encoding: it must hold for any b that $\mathsf{D}_1(\mathsf{E}_1(m, 0), 0) = m$ when $m \leftarrow \mathsf{R}_{\mathsf{OT}}(1^\lambda, b)$, since correctness of encoding guarantees that this holds when $m \leftarrow \mathsf{R}_{\mathsf{OT}}(1^\lambda, 0)$; but then, it must hold as well when $b = 1$, otherwise $(\mathsf{E}_1, \mathsf{D}_1)$ would allow to break the receiver security of the underlying OT. The correctness of $\mathsf{covE}_{\mathsf{OT}}$ follows directly from the correctness of the encoding, which concludes the proof.

7.3.2 A general compiler for covert protocols

Compiling a secure computation protocol into a covert MPC protocol requires some care, since both models have syntactic differences: a covert MPC protocol must, by design, have an admissibility function g defining whether the inputs of the parties are admissible, in addition to the target function f to be computed. We therefore focus on compiling secure protocols which already satisfy this syntactic requirement.

Consider an arbitrary two-party protocol $\Pi_{f,g}$, where A has input x and B has input y, which implements the following functionality $\mathcal{F}_{f,g}$: it first computes an admissibility function $g(x, y)$ on the inputs. If $g(x, y) = 1$, it outputs random shares of the target function $f(x, y)$ to the parties; else, it outputs random values to all parties. We assimilate A (resp. B) to a sampler that take as input x (resp. y) together with the transcript T of the protocol so far, and outputs the next message of A (resp. B). Eventually, we only assume from Π a weak indistinguishability-style security guarantee, namely that for any inputs x, y, no adversary passively corrupting B (resp. A) can distinguish the transcript of an interaction with A on input x (resp. B on input y) from the transcript of an interaction with A on input 0 (resp. B on input 0). Note that the protocol outputs random shares, so the output distribution of B is independent of A's input.

Let $(\mathsf{E}_A, \mathsf{D}_A)$ be a pseudorandom encoding associated to the sampler A, and $(\mathsf{E}_B, \mathsf{D}_B)$ be a pseudorandom encoding associated to the sampler B. The compiled protocol $\mathsf{cov}\Pi_{f,g}$ is constructed as follows: at round i, given a transcript T up to round $i - 2$, the message m_B received from Bob at round $i - 1$, and an input x, Alice computes $m \leftarrow \mathsf{D}_B(||0)$, appends m to T, samples $m' \leftarrow A(T||x)$, and sends $m_A \leftarrow \mathsf{E}_A(m, T||0)$. Bob proceeds symmetrically.

Proof Sketch. A straightforward generalization of the argument for the covert OT protocol shows that $\mathsf{cov}\Pi_{f,g}$ satisfies an indistinguishability-style notion of covertness: at each round, it holds that

$$\mathsf{E}_A(A(T||x), T||0) \approx_{\mathsf{c}} \mathsf{E}_A(A(T||0), T||0) \approx_{\mathsf{c}} U,$$

where U denotes the uniform distribution over strings of appropriate length; the first indistinguishability follows from the security of $\Pi_{f,g}$, and the second from the pseudorandomness of E_A. Similarly, security of $\Pi_{f,g}$ and correctness of the pseudorandom encodings guarantee the correctness of $\mathsf{cov}\Pi_{f,g}$. If the protocol $\Pi_{f,g}$ further satisfies a simulation-style security notion and securely implements the functionality $\mathcal{F}_{f,g}$, the protocol $\mathsf{cov}\Pi_{f,g}$ can be shown to covertly implement the functionality $\mathcal{F}_{\mathsf{CMPC}}(f, g)$.

7.4 Deniable encryption

As noted in [CDNO97], sender deniable encryption yields adaptively secure MPC for an arbitrary number of corruptions (and hence in conjunction with adaptively secure oblivious transfer implies $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$). However, it is not clear if $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$ together with the existence of a PKE scheme suffices to obtain deniable encryption. Recall, that the explainability compiler of [DKR15] which is based on the deniable encryption scheme of [SW14] corresponds to $\mathsf{cISH}^{\mathsf{rand}}_{\approx_{\mathsf{c}}}$, where closeness actually holds information theoretically. Let cISH' denote this variant of $\mathsf{cISH}^{\mathsf{rand}}_{\approx_c}$, where closeness holds information theoretically.

Definition 37 (Publicly deniable encryption, [SW14]). *A publicly deniable encryption scheme for message space $\mathcal{M}$ is a tuple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Explain})$ *such that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an IND-CPA secure encryption scheme and the following property is satisfied.*

Indistinguishability of explanation. *For all PPT adversaries $\mathcal{A}$,*

$$Adv^{\mathsf{ind\text{-}expl}}_{\mathcal{A}}(\lambda) := \left|\Pr[Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},0}(\lambda) = 1] - \Pr[Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},1}(\lambda) = 1]\right|$$

is negligible, where $Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},0}$ and $Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},1}$ are defined in Figure 35.

```
$Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},0}(\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$m^* \leftarrow \mathcal{A}(pk)$
$u^* \leftarrow \{0,1\}^*$
$c^* \leftarrow \mathsf{Enc}(pk, m^*; u^*)$
return $\mathcal{A}(c^*, u^*)$
```

```
$Exp^{\mathsf{ind\text{-}expl}}_{\mathcal{A},1}(\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$m^* \leftarrow \mathcal{A}(pk)$
$u^* \leftarrow \{0,1\}^*$
$c^* \leftarrow \mathsf{Enc}(pk, m^*; u^*)$
$r^* \leftarrow \{0,1\}^*$
$e^* \leftarrow \mathsf{Explain}(pk, c^*; r^*)$
return $\mathcal{A}(c^*, e^*)$
```

```
$Exp^{\mathsf{ind\text{-}expl}'}_{\mathcal{A},0}(\lambda)$
$m^* \leftarrow \mathcal{A}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$u^* \leftarrow \{0,1\}^*$
$c^* \leftarrow \mathsf{Enc}(pk, m^*; u^*)$
return $\mathcal{A}(pk, c^*, u^*)$
```

```
$Exp^{\mathsf{ind\text{-}expl}'}_{\mathcal{A},1}(\lambda)$
$m^* \leftarrow \mathcal{A}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$u^* \leftarrow \{0,1\}^*$
$c^* \leftarrow \mathsf{Enc}(pk, m^*; u^*)$
$r^* \leftarrow \{0,1\}^*$
$e^* \leftarrow \mathsf{Explain}(pk, c^*; r^*)$
return $\mathcal{A}(pk, c^*, e^*)$
```

Fig. 35. Indistinguishability of explanation experiments in its adaptive variant (left) and static variant (right).

We only consider publicly deniable encryption schemes with message space $\{0, 1\}$, since a deniable encryption scheme with message space $\{0, 1\}$ implies a publicly deniable encryption scheme for message space $\{0, 1\}^n$ (for a polynomial n in λ).

For message space $\{0, 1\}$, indistinguishability of explanation is equivalent to the indistinguishability of $Exp^{\mathsf{ind\text{-}expl}'}_{\mathcal{A},0}$ and $Exp^{\mathsf{ind\text{-}expl}'}_{\mathcal{A},1}$ as defined in Figure 35.

cISH' in conjunction with a PKE scheme yields (publicly) sender deniable encryption.

Theorem 23. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme for message space* $\{0, 1\}$. *If* cISH' *holds, then there exists a publicly deniable encryption scheme for message space* $\{0, 1\}$.

Proof. cISH' implies that for the PPT algorithm Enc there is a setup algorithm $\mathsf{Setup}_{\mathsf{Enc}}$, an alternative sampler $\overline{\mathsf{Enc}}$ and an inverse sampler $\overline{\mathsf{Enc}}^{-1}$ satisfying statistical closeness and computational invertibility. We define a publicly deniable encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}', \mathsf{Explain}')$ in Figure 36.

```
$\mathsf{Gen}'(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$pk' := (crs, pk)$, $sk' := sk$
return $(pk', sk')$
```

```
$\mathsf{Enc}'(pk', m)$
parse $(crs, pk) =: pk'$
$c \leftarrow \overline{\mathsf{Enc}}(crs, (pk, m))$
return $c$
```

```
$\mathsf{Dec}'(sk, c)$
$m \leftarrow \mathsf{Dec}(sk, c)$
return $m$
```

```
$\mathsf{Explain}'(pk', m)$
parse $(crs, pk) =: pk'$
$u \leftarrow \overline{\mathsf{Enc}}^{-1}(crs, (pk, m), c)$
return $u$
```

Fig. 36. Publicly deniable encryption scheme from cISH'.

Correctness. Let $m \in \{0, 1\}^*$ be a plaintext.

$$\epsilon_1 := \Pr[((crs, pk), sk) = (pk', sk') \leftarrow \mathsf{Gen}'(1^\lambda), c \leftarrow \mathsf{Enc}(pk, m) \colon \mathsf{Dec}(sk, c) \neq m]$$

$$\epsilon_2 := \Pr[((crs, pk), sk) = (pk', sk') \leftarrow \mathsf{Gen}'(1^\lambda), c \leftarrow \overline{\mathsf{Enc}}(pk, m) \colon \mathsf{Dec}(sk, c) \neq m]$$

Consider an *unbounded* adversary $\mathcal{A}$ on closeness which on input of (crs, pk, c), computes sk (using exhaustive search) and outputs 1 if $\mathsf{Dec}(sk, c) \neq m$ and 0 otherwise. The advantage of this adversary is $Adv^{\mathsf{crs\text{-}close}}_{\mathcal{A},m}(\lambda) = |\epsilon_1 - \epsilon_2|$. Hence, $\epsilon_2 \leq Adv^{\mathsf{crs\text{-}close}}_{\mathcal{A},m}(\lambda) + \epsilon_1$ and therefore negligible due to statistical closeness and correctness of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathbf{G}_0$	$\mathbf{G}_1$	$\mathbf{G}_2$	$\mathbf{G}_3$
$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$c^* \leftarrow \overline{\mathsf{Enc}}(pk, 0)$	$c^* \leftarrow \mathsf{Enc}(pk, 0)$	$c^* \leftarrow \mathsf{Enc}(pk, 1)$	$c^* \leftarrow \mathsf{Enc}(pk, 1)$
return $\mathcal{A}((crs, pk), c^*)$	**return** $\mathcal{A}((crs, pk), c^*)$	**return** $\mathcal{A}((crs, pk), c^*)$	**return** $\mathcal{A}((crs, pk), c^*)$

Fig. 37. Hybrids used in the proof of IND-CPA security for Theorem 23.

IND-CPA security. For message space $\{0, 1\}$, IND-CPA security of $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is equivalent to the indistinguishability between the games $\mathbf{G}_0$ and $\mathbf{G}_3$ of Figure 37.

Claim. For all (unbounded) adversaries $\mathcal{A}$ there exists an (unbounded) adversary $\overline{\mathcal{A}}$ such that $|\Pr[out_1 = 1] - \Pr[out_0 = 1]| \leq Adv_{\overline{\mathcal{A}}}^{\mathsf{crs\text{-}close}}(\lambda)$.

Proof. Let $\mathcal{A}$ be an unbounded adversary distinguishing $\mathbf{G}_0$ and $\mathbf{G}_1$. Construct an adversary $\overline{\mathcal{A}}$ on closeness. Initially, $\overline{\mathcal{A}}$ samples $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and outputs $m := (pk, 0)$ to the experiment. In return, $\overline{\mathcal{A}}$ receives (crs, y), where y is either sampled using $\mathsf{Enc}(m)$ or $\overline{\mathsf{Enc}}(m)$. $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input of $((crs, pk), y)$. Hence, $\Pr[out_b = 1] = \Pr[Exp_{\overline{\mathcal{A}},1-b}^{\mathsf{crs\text{-}close}}(\lambda) = 1]$ for $b \in \{0, 1\}$. □

The game hop from $\mathbf{G}_1$ to $\mathbf{G}_2$ is justified by the IND-CPA security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The game hop from $\mathbf{G}_2$ to $\mathbf{G}_3$ is justified by statistical closeness.

Indistinguishability of explanation. We need to prove indistinguishability between the games described in Figure 38.

$Exp_{\mathcal{A},0}^{\mathsf{ind\text{-}expl}'}(\lambda)$	$Exp_{\mathcal{A},1}^{\mathsf{ind\text{-}expl}'}(\lambda)$
$m^* \leftarrow \mathcal{A}(1^\lambda)$	$m^* \leftarrow \mathcal{A}(1^\lambda)$
$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$	$crs \leftarrow \mathsf{Setup}_{\mathsf{Enc}}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$u^* \leftarrow \{0, 1\}^*$	$u^* \leftarrow \{0, 1\}^*$
$c^* \leftarrow \overline{\mathsf{Enc}}(pk, m^*; u^*)$	$c^* \leftarrow \overline{\mathsf{Enc}}(pk, m^*; u^*)$
return $\mathcal{A}((crs, pk), c^*, u^*)$	$r^* \leftarrow \{0, 1\}^*$
	$e^* \leftarrow \overline{\mathsf{Enc}}^{-1}(pk, c^*; r^*)$
	return $\mathcal{A}((crs, pk), c^*, e^*)$

Fig. 38. Unwrapped indistinguishability of explanation games used in the proof of Theorem 23.

Claim. For all PPT adversaries $\mathcal{A}$, $Adv_{\mathcal{A}}^{\mathsf{ind\text{-}expl}'}(\lambda)$ is negligible.

Proof. Let $\mathcal{A}$ be an adversary distinguishing $Exp_{\mathcal{A},0}^{\mathsf{ind\text{-}expl}'}$ and $Exp_{\mathcal{A},1}^{\mathsf{ind\text{-}expl}'}$ above. Construct an adversary $\overline{\mathcal{A}}$ on invertibility. Initially, $\overline{\mathcal{A}}$ calls $\mathcal{A}$, obtains m^*, samples $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and outputs $m := (pk, m^*)$ to the experiment. In return, $\overline{\mathcal{A}}$ receives (crs, r, y) from the experiment, where y is sampled using $\overline{\mathsf{Enc}}(pk, m^*)$ and r either is the randomness used or itself sampled from $\overline{\mathsf{Enc}}^{-1}(pk, m^*, y)$. $\overline{\mathcal{A}}$ calls $\mathcal{A}$ on input of $((crs, pk), y, r)$. Hence, $\Pr[Exp_{\mathcal{A},b}^{\mathsf{ind\text{-}expl}'}(\lambda) = 1] = \Pr[Exp_{\overline{\mathcal{A}},b}^{\mathsf{crs\text{-}inv}}(\lambda) = 1]$ for $b \in \{0, 1\}$. □

□

Whether $\mathsf{cISH}_{\approx_c}^{\mathsf{rand}}$ (without statistical closeness) implies deniable encryption remains open.

7.5 Non-committing encryption

Non-committing encryption is a powerful notion which is known to imply adaptive MPC for all but one corruptions, [CFGN96].

Definition 38 (Non-committing bit encryption encryption in the global CRS model). *A non-committing bit encryption scheme in the global CRS model is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$*, such that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a PKE scheme and the following distributions are computationally indistinguishable.*

$$\{crs \leftarrow \mathsf{Setup}(1^\lambda), (pk, sk) := \mathsf{Gen}(1^\lambda, crs; r_{\mathsf{Gen}}), c := \mathsf{Enc}(crs, pk, b; r_{\mathsf{Enc}})\colon (crs, pk, c, r_{\mathsf{Gen}}, r_{\mathsf{Enc}})\}$$
$$\{crs \leftarrow \mathsf{Setup}(1^\lambda), (pk, c, r^0_{\mathsf{Gen}}, r^1_{\mathsf{Gen}}, r^0_{\mathsf{Enc}}, r^1_{\mathsf{Enc}}) \leftarrow \mathsf{Sim}(1^\lambda; crs)\colon (crs, pk, c, r^b_{\mathsf{Gen}}, r^b_{\mathsf{Enc}})\}$$

Following the lines of [CDMW09], if $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ is true and there exists an IND-CPA secure PKE scheme, then there exists a non-committing encryption scheme.

Theorem 24. *Let* $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *be an IND-CPA secure PKE scheme for message space* $\{0,1\}^\lambda$*. If* $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ *holds, then there exists a non-committing encryption scheme.*

Proof (sketch).

```
$\mathsf{Gen}(1^\lambda, crs)$
$M_0, M_1 \leftarrow \{0,1\}^\lambda$
$T \leftarrow [4\lambda]$ s.t. $|T| = \lambda$
$(pk_i, sk_i) \leftarrow \begin{cases} \mathsf{Gen}'(1^\lambda) & \text{if } i \in T \\ \overline{\mathsf{OGen}}(crs) & \text{otherwise} \end{cases}$
$pk := (M_0, M_1, pk_1, \ldots, pk_{4\lambda})$
$sk := (T, (sk_i)_{i \in T})$
return $(pk, sk)$
```

```
$\mathsf{Enc}(crs, pk, b)$
$S \leftarrow [4\lambda]$ s.t. $|S| = \lambda$
$c_i \leftarrow \begin{cases} \mathsf{Enc}'(pk_i, M_b) & \text{if } i \in S \\ \overline{\mathsf{OEnc}}(crs, pk_i) & \text{otherwise} \end{cases}$
return $c := (c_i)_{i \in [4\lambda]}$
```

```
$\mathsf{Dec}(crs, sk, c)$
$J := \{\mathsf{Dec}'(sk_i, c_i) \mid i \in T\}$
if $M_0 \in J$ then
    return 0
else
    return 1
```

Fig. 39. Non-committing encryption scheme in the global CRS model based on [CDMW09].

Let OGen be the algorithm which on input of 1^λ calls $\mathsf{Gen}'(1^\lambda)$ and outputs only $(pk, \bot)$. Further let OEnc be the algorithm which on input of $(1^\lambda, pk)$ samples $m \leftarrow \{0,1\}^\lambda$ and outputs $c \leftarrow \mathsf{Enc}'(1^\lambda, pk, m)$. If $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ is true, then there exists alternative and inverse sampler for OGen and OEnc, denoted by $(\overline{\mathsf{OGen}}, \overline{\mathsf{OGen}}^{-1})$ and $(\overline{\mathsf{OEnc}}, \overline{\mathsf{OEnc}}^{-1})$, respectively. Note that the alternative sampler and the inverse sampler need access to the CRS. If the setup algorithm Setup is trivial, this corresponds to the notion of simulatable encryption due to [DN00] and yields non-committing encryption directly due to [CDMW09]. Figures 39 and 40 shows the construction of non-committing encryption in the global CRS model. Note that assuming *adaptive* $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_c}$ is necessary since the inputs to the sampler OEnc are sampled via $\mathsf{OGen}(crs)$ during the simulation and, hence, depend on the CRS. The indistinguishability between the real and the simulated distribution follows the same ideas as in [CDMW09].

□

7.6 Super-polynomial encoding

Extremely lossy functions due to [Zha16] are functions which can be set up in two computationally indistinguishable modes – an injective mode and a extremely lossy mode, where the range of the function is merely polynomial. A slight relaxation of this notion is what we call very lossy functions (VLFs). The difference to ELFs is that we require indistinguishability between functions with exponential range and *super-polynomial* range. The existence of these functions implies a relaxation of $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_c}$.

Definition 39 (Very lossy function). *A* Very lossy function *consists of an algorithm* $\mathsf{VLF.Gen}'$ *and a computable function* $N(M)$ *such that* $\log N$ *is polynomial in* $\log M$*.* $\mathsf{VLF.Gen}'$ *takes as input natural numbers* M*,* $r \in [M]$ *and a flag* $b \in \{\mathsf{inj}, \mathsf{lossy}\}$*, and outputs the description of a function* $f\colon [M] \to [N]$ *such that*

- *f is computable in time polynomial in* $\log M$*,*

$\mathsf{Sim}(crs)$

$$M_0, M_1 \leftarrow \{0,1\}^\lambda$$

$$S_0, T_0 \leftarrow [4\lambda] \text{ s.t. } |S_0| = |T_0| = \lambda$$

$$S_1, T_1 \leftarrow [4\lambda] \setminus (S_0 \cup T_0) \text{ s.t. } |S_0 \cap T_0| = |S_1 \cap T_1|$$

$$(pk_i, sk_i) \leftarrow \begin{cases} \mathsf{Gen}'(1^\lambda; r^{(i)}_{\mathsf{Gen}'}) & \text{if } i \in T_0 \cup S_0 \cup T_1 \cup S_1 \\ \overline{\mathsf{OGen}}(crs; r^{(i)}_{\mathsf{OGen}}) & \text{otherwise} \end{cases}$$

$$c_i \leftarrow \begin{cases} \mathsf{Enc}'(pk_i, M_0; r^{(i)}_{\mathsf{Enc}'}) & \text{if } i \in S_0 \\ \mathsf{Enc}'(pk_i, M_1; r^{(i)}_{\mathsf{Enc}'}) & \text{if } i \in S_1 \\ \overline{\mathsf{OEnc}}(crs, pk_i; r^{(i)}_{\mathsf{OEnc}}) & \text{otherwise} \end{cases}$$

$$\text{define } r^b_{\mathsf{Gen}} := (T_b, (u^{(b,i)}_{\mathsf{Gen}'})_{i \in [4\lambda]}) \text{ and } r^b_{\mathsf{Enc}} := (S_b, (u^{(b,i)}_{\mathsf{Enc}'})_{i \in [4\lambda]})$$

$$u^{(b,i)}_{\mathsf{Gen}'} \leftarrow \begin{cases} r^{(i)}_{\mathsf{Gen}'} & \text{if } i \in T_b \\ \overline{\mathsf{OGen}}^{-1}(crs, pk_i) & \text{if } i \in (T_0 \cup T_1 \cup S_0 \cup S_1) \setminus T_b \\ r^{(i)}_{\mathsf{OGen}} & \text{otherwise} \end{cases}$$

$$u^{(b,i)}_{\mathsf{Enc}'} \leftarrow \begin{cases} r^{(i)}_{\mathsf{Enc}'} & \text{if } i \in S_b \\ \overline{\mathsf{OEnc}}^{-1}(crs, (pk_i, M_{1-b}), c_i) & \text{if } i \in S_{1-b} \\ r^{(i)}_{\mathsf{OEnc}} & \text{otherwise} \end{cases}$$

$$\textbf{return } \left(pk := (M_0, M_1, (pk_i)_{i \in [4\lambda]}), c := (c_i)_{i \in [4\lambda]}, r^0_{\mathsf{Gen}}, r^1_{\mathsf{Gen}}, r^0_{\mathsf{Enc}}, r^1_{\mathsf{Enc}}\right)$$

Fig. 40. The simulator Sim for the non-committing encryption scheme in the global CRS model described in Figure 39. Sim is based on the simulator given in [CDMW09].

- *if $b = \mathsf{inj}$, f is injective with overwhelming probability (in $\log M$),*
- *for all $r \in [M]$, if $b = \mathsf{lossy}$, $|f([M])| \leq r$ with overwhelming probability (in $\log M$),*
- *there exists a super-polynomial function q such that for all PPT adversaries $\mathcal{A}$ and any $r \in [q(\log M), M]$,*

$$|\Pr[\mathcal{A}(\mathsf{VLF.Gen}'(M, r, \mathsf{inj})) = 1] - \Pr[\mathcal{A}(\mathsf{VLF.Gen}'(M, r, \mathsf{lossy})) = 1]|$$

is negligible.

Definition 40 (Strong regularity, [Zha16]). *A VLF* VLF *is* strongly regular *if for all $r \in [M]$, with overwhelming probability over the choice of $f \leftarrow \mathsf{VLF.Gen}(M, r)$ we have that the distribution $\{x \leftarrow [M] \colon f(x)\}$ is statistically close to uniform distribution over $f([M])$.*

A VLF is called strongly efficiently enumerable, if there exists a (potentially randomized) algorithm running in polynomial time in $\log M$ and r which given $f \leftarrow \mathsf{VLF.Gen}(M, r, \mathsf{lossy})$ and r outputs a set $S \subseteq [N]$ such that with overwhelming probability (over the choice of f and the randomness of the algorithm), $S = f([M])$. If VLF is strongly regular, then it also is strongly efficiently enumerable, [Zha16]. [Zha16] shows that strongly regular ELFs are implied by the exponential decisional Diffie-Hellman (DDH) assumption.

We note that in contrast to ELFs, an adversary learning r does not harm security.

Assuming the sub-exponential hardness of the decisional Diffie-Hellman (DDH) problem, the bounded adversary extremely lossy function instantiation from [Zha16] is a very lossy function according to Definition 39.

Theorem 25. *If strongly regular very lossy functions exist, then $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with super-polynomial encoding is true.*

Proof. Let S be a PPT sampler with input space L. For $m \in L$, let $\ell_r(|m|)$ denote the polynomial which upper bounds the number of random bits $S(m)$ takes, i.e. the random tape of S is uniform over $\{0,1\}^{\ell_r} = [2^{\ell_r}]$. Further, let ξ be a super-polynomial function. The setup algorithm, the encoding and decoding algorithms are defined in Figure 41. We prove the equivalent properties closeness and invertibility.

```
$\mathsf{Setup}(1^\lambda, 2^{\ell_r})$
$G \leftarrow \mathsf{ELF.Gen}(2^{\ell_r}, \xi, \mathsf{lossy})$
return $crs := G$
```

```
$\mathsf{E}_S(crs, m, y)$
$\mathcal{R} := \varnothing$
for $r' \in \mathsf{image}(G)$ do
    if $S(m; G(r')) = y$ then
        $\mathcal{R} \mathrel{+}= r'$
$r \leftarrow \mathcal{R}$
return $r$
```

```
$\mathsf{D}_S(crs, m, r)$
$y := S(m; G(r))$
return $y$
```

Fig. 41. Description of the universal setup algorithm, the super-polynomial time encoding algorithm and the polynomial time decoding algorithm.

Invertibility. The encoding algorithm produces perfectly distributed inverse sampled random tapes. Hence, adaptive invertibility follows.
Closeness. We start from the game $Exp_{\mathcal{A}}^{\mathsf{a\text{-}crs\text{-}close}}$ and switch the VLF to injective mode $G \leftarrow \mathsf{VLF.Gen}(2^{\ell_r}, 2^{\ell_r}, \mathsf{inj})$. This is computationally indistinguishable for the adversary. The inverse sampler will not work anymore, but since the adversary is polynomially bounded, he can not call the inverse sampler anyway. Strong regularity implies that for $G \leftarrow \mathsf{VLF.Gen}(2^{\ell_r}, 2^{\ell_r}, \mathsf{inj})$ the distribution $\{x \leftarrow [2^{\ell_r}] \colon G(x)\}$ is statistically close to uniform distribution over $f([2^{\ell_r}])$. Hence, $S(m; G(r))$ and $S(m; r)$ for uniform r from $[2^{\ell_r}]$ are statistically close. □

Unfortunately, $\mathsf{acPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with super-polynomial encoding algorithm (or, equivalently $\mathsf{acISH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with super-polynomial inverse sampler) does not suffice to imply adaptive MPC even if the simulator is allowed to run in super-polynomial time, [Pas03]. This is because plugging $\mathsf{cPREH}^{\mathsf{rand}}_{\approx_\mathsf{c}}$ with super-polynomial encoding algorithm into the proof of Theorem 4, the game hop from $\mathbf{G}_1$ to $\mathbf{G}_2$ can not be made, since the simulation of these games requires super-polynomial time and, hence, a reduction to closeness against PPT adversaries is not possible. We do, however, obtain a non-standard notion of adaptive MPC with super-polynomial simulation, namely for any functionality $\mathcal{F}$ we are able to adaptively realize a functionality $\overline{\mathcal{F}}$ which produces a computationally (against PPT adversaries) indistinguishable output distribution to the original functionality. We view it as an interesting problem to further study this notion of adaptive MPC.

References

[BB04] Dan Boneh and Xavier Boyen. "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles". In: *Advances in Cryptology – EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. Lecture Notes in Computer Science. Interlaken, Switzerland: Springer, Heidelberg, Germany, 2004, pp. 223–238. DOI: 10.1007/978-3-540-24676-3_14 (cit. on p. 15).

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. "On the existence of extractable one-way functions". In: *46th Annual ACM Symposium on Theory of Computing*. Ed. by David B. Shmoys. New York, NY, USA: ACM Press, 2014, pp. 505–514. DOI: 10.1145/2591796.2591859 (cit. on pp. 7–9, 16, 17, 46, 47, 53, 54).

[BGIRSVY01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (Im)possibility of Obfuscating Programs". In: *Advances in Cryptology – CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2001, pp. 1–18. DOI: 10.1007/3-540-44647-8_1 (cit. on p. 8).

[BL18] Sebastian Berndt and Maciej Liskiewicz. "On the Gold Standard for Security of Universal Steganography". In: *Advances in Cryptology – EUROCRYPT 2018, Part I*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. Lecture

Notes in Computer Science. Tel Aviv, Israel: Springer, Heidelberg, Germany, 2018, pp. 29–60. DOI: 10.1007/978-3-319-78381-9_2 (cit. on p. 59).

[BM92] Steven M. Bellovin and Michael Merritt. "Encrypted Key Exchange: Password-Based Protocols Secure against Dictionary Attacks". In: *1992 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 1992, pp. 72–84. DOI: 10.1109/RISP.1992.213269 (cit. on pp. 3, 6).

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. "Computational Analogues of Entropy". In: *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24-26, 2003, Proceedings*. Ed. by Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai. Vol. 2764. Lecture Notes in Computer Science. Springer, 2003, pp. 200–215. ISBN: 3-540-40770-7. DOI: 10.1007/978-3-540-45198-3_18. URL: https://doi.org/10.1007/978-3-540-45198-3_18 (cit. on p. 38).

[Can00] Ran Canetti. "Security and Composition of Multiparty Cryptographic Protocols". In: *Journal of Cryptology* 13.1 (Jan. 2000), pp. 143–202. DOI: 10.1007/s001459910006 (cit. on pp. 13, 29, 30, 35).

[Can01] Ran Canetti. "Universally Composable Security: A New Paradigm for Cryptographic Protocols". In: *42nd Annual Symposium on Foundations of Computer Science*. Las Vegas, NV, USA: IEEE Computer Society Press, 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888 (cit. on p. 30).

[CDMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. "Improved Non-committing Encryption with Applications to Adaptively Secure Protocols". In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Tokyo, Japan: Springer, Heidelberg, Germany, 2009, pp. 287–302. DOI: 10.1007/978-3-642-10366-7_17 (cit. on pp. 9, 65, 66).

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. "Deniable Encryption". In: *Advances in Cryptology – CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 1997, pp. 90–104. DOI: 10.1007/BFb0052229 (cit. on pp. 11, 62).

[CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. "Universally Composable Security with Global Setup". In: *TCC 2007: 4th Theory of Cryptography Conference*. Ed. by Salil P. Vadhan. Vol. 4392. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, 2007, pp. 61–85. DOI: 10.1007/978-3-540-70936-7_4 (cit. on p. 30).

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. "Adaptively Secure Multi-Party Computation". In: *28th Annual ACM Symposium on Theory of Computing*. Philadephia, PA, USA: ACM Press, 1996, pp. 639–648. DOI: 10.1145/237814.238015 (cit. on pp. 11, 64).

[CGOS07] Nishanth Chandran, Vipul Goyal, Rafail Ostrovsky, and Amit Sahai. "Covert Multi-Party Computation". In: *48th Annual Symposium on Foundations of Computer Science*. Providence, RI, USA: IEEE Computer Society Press, 2007, pp. 238–248. DOI: 10.1109/FOCS.2007.21 (cit. on pp. 11, 60, 61).

[CGP15] Ran Canetti, Shafi Goldwasser, and Oxana Poburinnaya. "Adaptively Secure Two-Party Computation from Indistinguishability Obfuscation". In: *TCC 2015: 12th Theory of Cryptography Conference, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Warsaw, Poland: Springer, Heidelberg, Germany, 2015, pp. 557–585. DOI: 10.1007/978-3-662-46497-7_22 (cit. on p. 4).

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. “Universally composable two-party and multi-party secure computation”. In: *34th Annual ACM Symposium on Theory of Computing*. Montréal, Québec, Canada: ACM Press, 2002, pp. 494–503. DOI: 10.1145/509907.509980 (cit. on pp. 4, 29, 33).

[CPR16] Ran Canetti, Oxana Poburinnaya, and Mariana Raykova. *Optimal-Rate Non-Committing Encryption in a CRS Model*. Cryptology ePrint Archive, Report 2016/511. http://eprint.iacr.org/2016/511. 2016 (cit. on pp. 35, 56).

[CPV17] Ran Canetti, Oxana Poburinnaya, and Muthuramakrishnan Venkitasubramaniam. “Better Two-Round Adaptive Multi-party Computation”. In: *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part II*. Ed. by Serge Fehr. Vol. 10175. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, 2017, pp. 396–427. DOI: 10.1007/978-3-662-54388-7_14 (cit. on p. 4).

[CsW19] Ran Cohen, abhi shelat, and Daniel Wichs. “Adaptively Secure MPC with Sublinear Communication Complexity”. In: *Advances in Cryptology – CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2019, pp. 30–60. DOI: 10.1007/978-3-030-26951-7_2 (cit. on pp. 4, 9, 10, 15, 37, 54).

[DKR15] Dana Dachman-Soled, Jonathan Katz, and Vanishree Rao. “Adaptively Secure, Universally Composable, Multiparty Computation in Constant Rounds”. In: *TCC 2015: 12th Theory of Cryptography Conference, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Warsaw, Poland: Springer, Heidelberg, Germany, 2015, pp. 586–613. DOI: 10.1007/978-3-662-46497-7_23 (cit. on pp. 4, 8–10, 14, 15, 17, 29, 32–34, 37, 46, 54–56, 62).

[DN00] Ivan Damgård and Jesper Buus Nielsen. “Improved Non-committing Encryption Schemes Based on a General Complexity Assumption”. In: *Advances in Cryptology – CRYPTO 2000*. Ed. by Mihir Bellare. Vol. 1880. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2000, pp. 432–450. DOI: 10.1007/3-540-44598-6_27 (cit. on pp. 7, 12, 23, 65).

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”. In: *SIAM J. Comput.* 38.1 (2008), pp. 97–139. DOI: 10.1137/060651380. URL: https://doi.org/10.1137/060651380 (cit. on p. 38).

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. “Randomness Condensers for Efficiently Samplable, Seed-Dependent Sources”. In: *TCC 2012: 9th Theory of Cryptography Conference*. Ed. by Ronald Cramer. Vol. 7194. Lecture Notes in Computer Science. Taormina, Sicily, Italy: Springer, Heidelberg, Germany, 2012, pp. 618–635. DOI: 10.1007/978-3-642-28914-9_35 (cit. on p. 3).

[GGHRSW13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. “Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits”. In: *54th Annual Symposium on Foundations of Computer Science*. Berkeley, CA, USA: IEEE Computer Society Press, 2013, pp. 40–49. DOI: 10.1109/FOCS.2013.13 (cit. on p. 8).

[GL91] Shafi Goldwasser and Leonid A. Levin. “Fair Computation of General Functions in Presence of Immoral Majority”. In: *Advances in Cryptology – CRYPTO’90*. Ed. by Alfred J. Menezes and Scott A. Vanstone. Vol. 537. Lecture Notes in

Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 1991, pp. 77–93. DOI: 10.1007/3-540-38424-3_6 (cit. on p. 31).

[GP15] Sanjam Garg and Antigoni Polychroniadou. "Two-Round Adaptively Secure MPC from Indistinguishability Obfuscation". In: *TCC 2015: 12th Theory of Cryptography Conference, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Warsaw, Poland: Springer, Heidelberg, Germany, 2015, pp. 614–637. DOI: 10.1007/978-3-662-46497-7_24 (cit. on p. 4).

[GS85] Andrew V. Goldberg and Michael Sipser. "Compression and Ranking". In: *17th Annual ACM Symposium on Theory of Computing*. Providence, RI, USA: ACM Press, 1985, pp. 440–448. DOI: 10.1145/22145.22194 (cit. on pp. 3, 5).

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A Pseudorandom Generator from any One-way Function". In: *SIAM J. Comput.* 28.4 (1999), pp. 1364–1396. DOI: 10.1137/S0097539793244708. URL: https://doi.org/10.1137/S0097539793244708 (cit. on pp. 5, 6, 15, 38, 60).

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. "Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility". In: *Advances in Cryptology – EUROCRYPT 2007*. Ed. by Moni Naor. Vol. 4515. Lecture Notes in Computer Science. Barcelona, Spain: Springer, Heidelberg, Germany, 2007, pp. 169–186. DOI: 10.1007/978-3-540-72540-4_10 (cit. on pp. 3, 6, 9, 16, 38, 45).

[Hop05] Nicholas Hopper. "On Steganographic Chosen Covertext Security". In: *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*. 2005, pp. 311–323. DOI: 10.1007/11523468_26. URL: https://doi.org/10.1007/11523468_26 (cit. on pp. 59, 60).

[HPRV19] Thibaut Horel, Sunoo Park, Silas Richelson, and Vinod Vaikuntanathan. "How to Subvert Backdoored Encryption: Security Against Adversaries that Decrypt All Ciphertexts". In: *ITCS 2019: 10th Innovations in Theoretical Computer Science Conference*. Ed. by Avrim Blum. Vol. 124. San Diego, CA, USA: LIPIcs, 2019, 42:1–42:20. DOI: 10.4230/LIPIcs.ITCS.2019.42 (cit. on p. 3).

[IKOS10] Yuval Ishai, Abishek Kumarasubramanian, Claudio Orlandi, and Amit Sahai. "On Invertible Sampling and Adaptive Security". In: *Advances in Cryptology – ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. Lecture Notes in Computer Science. Singapore: Springer, Heidelberg, Germany, 2010, pp. 466–482. DOI: 10.1007/978-3-642-17373-8_27 (cit. on pp. 7–9, 12, 14, 16, 23, 29, 31, 35, 48–50, 54).

[Imp95] Russell Impagliazzo. "A Personal View of Average-Case Complexity". In: *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*. 1995, pp. 134–147. DOI: 10.1109/SCT.1995.514853. URL: https://doi.org/10.1109/SCT.1995.514853 (cit. on p. 5).

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. "Founding Cryptography on Oblivious Transfer - Efficiently". In: *Advances in Cryptology – CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2008, pp. 572–591. DOI: 10.1007/978-3-540-85174-5_32 (cit. on pp. 14, 31).

[JR14] Ari Juels and Thomas Ristenpart. "Honey Encryption: Security Beyond the Brute-Force Bound". In: *Advances in Cryptology – EUROCRYPT 2014*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. Lecture Notes in Computer

Science. Copenhagen, Denmark: Springer, Heidelberg, Germany, 2014, pp. 293–310. DOI: 10.1007/978-3-642-55220-5_17 (cit. on pp. 3, 7, 9, 11, 57, 58).

[JRT16] Joseph Jaeger, Thomas Ristenpart, and Qiang Tang. "Honey Encryption Beyond Message Recovery Security". In: *Advances in Cryptology – EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. Lecture Notes in Computer Science. Vienna, Austria: Springer, Heidelberg, Germany, 2016, pp. 758–788. DOI: 10.1007/978-3-662-49890-3_29 (cit. on p. 58).

[Kil88] Joe Kilian. "Founding Cryptography on Oblivious Transfer". In: *20th Annual ACM Symposium on Theory of Computing*. Chicago, IL, USA: ACM Press, 1988, pp. 20–31. DOI: 10.1145/62212.62215 (cit. on pp. 14, 31).

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. "How to delegate computations: the power of no-signaling proofs". In: *46th Annual ACM Symposium on Theory of Computing*. Ed. by David B. Shmoys. New York, NY, USA: ACM Press, 2014, pp. 485–494. DOI: 10.1145/2591796.2591809 (cit. on p. 17).

[Lin09] Andrew Y. Lindell. "Adaptively Secure Two-Party Computation with Erasures". In: *Topics in Cryptology – CT-RSA 2009*. Ed. by Marc Fischlin. Vol. 5473. Lecture Notes in Computer Science. San Francisco, CA, USA: Springer, Heidelberg, Germany, 2009, pp. 117–132. DOI: 10.1007/978-3-642-00862-7_8 (cit. on pp. 13, 29).

[LMs05] Matt Lepinski, Silvio Micali, and abhi shelat. "Fair-Zero Knowledge". In: *TCC 2005: 2nd Theory of Cryptography Conference*. Ed. by Joe Kilian. Vol. 3378. Lecture Notes in Computer Science. Cambridge, MA, USA: Springer, Heidelberg, Germany, 2005, pp. 245–263. DOI: 10.1007/978-3-540-30576-7_14 (cit. on pp. 9, 16, 45).

[LV19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. ISBN: 978-3-030-11297-4. DOI: 10.1007/978-3-030-11298-1. URL: https://doi.org/10.1007/978-3-030-11298-1 (cit. on p. 3).

[LV90] Ming Li and Paul M. B. Vitányi. "Handbook of Theoretical Computer Science (Vol. A)". In: ed. by Jan van Leeuwen. Cambridge, MA, USA: MIT Press, 1990. Chap. Kolmogorov Complexity and Its Applications, pp. 187–254. ISBN: 0-444-88071-2. URL: http://dl.acm.org/citation.cfm?id=114872.114876 (cit. on p. 3).

[Pas03] Rafael Pass. "Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition". In: *Advances in Cryptology – EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. Lecture Notes in Computer Science. Warsaw, Poland: Springer, Heidelberg, Germany, 2003, pp. 160–176. DOI: 10.1007/3-540-39200-9_10 (cit. on p. 67).

[Rey11] Leonid Reyzin. "Some Notions of Entropy for Cryptography - (Invited Talk)". In: *ICITS 11: 5th International Conference on Information Theoretic Security*. Ed. by Serge Fehr. Vol. 6673. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, 2011, pp. 138–142. DOI: 10.1007/978-3-642-20728-0_13 (cit. on pp. 37, 38).

[RR99] Ran Raz and Omer Reingold. "On Recycling the Randomness of States in Space Bounded Computation". In: *31st Annual ACM Symposium on Theory of Computing*. Atlanta, GA, USA: ACM Press, 1999, pp. 159–168. DOI: 10.1145/301250.301294 (cit. on p. 3).

[Sha48] Claude Elwood Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27.3 (1948), pp. 379–423 (cit. on pp. 15, 38).

[Sho04] Victor Shoup. *Sequences of games: a tool for taming complexity in security proofs*. Cryptology ePrint Archive, Report 2004/332. http://eprint.iacr.org/2004/332. 2004 (cit. on pp. 27, 28).

[SW14] Amit Sahai and Brent Waters. "How to use indistinguishability obfuscation: deniable encryption, and more". In: *46th Annual ACM Symposium on Theory of Computing*. Ed. by David B. Shmoys. New York, NY, USA: ACM Press, 2014, pp. 475–484. DOI: 10.1145/2591796.2591825 (cit. on pp. 9, 11, 37, 46, 54–56, 62, 63).

[TUZ01] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. "Loss-less condensers, unbalanced expanders, and extractors". In: *33rd Annual ACM Symposium on Theory of Computing*. Crete, Greece: ACM Press, 2001, pp. 143–152. DOI: 10.1145/380752.380790 (cit. on p. 3).

[TV00] Luca Trevisan and Salil P. Vadhan. "Extracting Randomness from Samplable Distributions". In: *41st Annual Symposium on Foundations of Computer Science*. Redondo Beach, CA, USA: IEEE Computer Society Press, 2000, pp. 32–42. DOI: 10.1109/SFCS.2000.892063 (cit. on p. 3).

[TVZ05] Luca Trevisan, Salil P. Vadhan, and David Zuckerman. "Compression of Samplable Sources". In: *Computational Complexity* 14.3 (2005), pp. 186–227. DOI: 10.1007/s00037-005-0198-6. URL: https://doi.org/10.1007/s00037-005-0198-6 (cit. on pp. 3, 5, 15, 21, 39).

[vH04] Luis von Ahn and Nicholas J. Hopper. "Public-Key Steganography". In: *Advances in Cryptology – EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. Lecture Notes in Computer Science. Interlaken, Switzerland: Springer, Heidelberg, Germany, 2004, pp. 323–341. DOI: 10.1007/978-3-540-24676-3_20 (cit. on p. 59).

[vHL05] Luis von Ahn, Nicholas J. Hopper, and John Langford. "Covert two-party computation". In: *37th Annual ACM Symposium on Theory of Computing*. Ed. by Harold N. Gabow and Ronald Fagin. Baltimore, MA, USA: ACM Press, 2005, pp. 513–522. DOI: 10.1145/1060590.1060668 (cit. on pp. 11, 60, 61).

[Wee04] Hoeteck Wee. "On Pseudoentropy versus Compressibility". In: *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*. IEEE Computer Society, 2004, pp. 29–41. ISBN: 0-7695-2120-7. DOI: 10.1109/CCC.2004.1313782. URL: https://doi.org/10.1109/CCC.2004.1313782 (cit. on pp. 3, 6, 15, 38, 39).

[Yao82] Andrew Chi-Chih Yao. "Theory and Applications of Trapdoor Functions (Extended Abstract)". In: *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. IEEE Computer Society, 1982, pp. 80–91. DOI: 10.1109/SFCS.1982.45. URL: https://doi.org/10.1109/SFCS.1982.45 (cit. on pp. 6, 16, 38).

[Yao86] Andrew Chi-Chih Yao. "How to Generate and Exchange Secrets (Extended Abstract)". In: *27th Annual Symposium on Foundations of Computer Science*. Toronto, Ontario, Canada: IEEE Computer Society Press, 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25 (cit. on p. 32).

[Zha16] Mark Zhandry. "The Magic of ELFs". In: *Advances in Cryptology – CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2016, pp. 479–508. DOI: 10.1007/978-3-662-53018-4_18 (cit. on pp. 8, 65, 66).