Statistical Randomized Encodings: A Complexity Theoretic View

Shweta Agrawal *, Yuval Ishai **, Dakshita Khurana ***, and Anat Paskin-Cherniavsky †

Abstract. A randomized encoding of a function $f(x)$ is a randomized function $\hat{f}(x,r)$, such that the "encoding" $\hat{f}(x,r)$ reveals $f(x)$ and essentially no additional information about x. Randomized encodings of functions have found many applications in different areas of cryptography, including secure multiparty computation, efficient parallel cryptography, and verifiable computation.

We initiate a complexity-theoretic study of the class SRE of languages (or boolean functions) that admit an efficient statistical randomized encoding. That is, $\hat{f}(x,r)$ can be computed in time poly($|x|$), and its output distribution on input x can be sampled in time poly($|x|$) given $f(x)$, up to a small statistical distance.

We obtain the following main results.

- **Separating SRE from efficient computation:** We give the first examples of promise problems and languages in SRE that are widely conjectured to lie outside P/poly. Our candidate promise problems and languages are based on the standard Learning with Errors (LWE) assumption, a non-standard variant of the Decisional Diffie Hellman (DDH) assumption and the "Abelian Subgroup Membership problem" (which generalizes Quadratic-Residuosity and a variant of DDH).
- **Separating SZK from SRE:** We explore the relationship of SRE with the class SZK of problems possessing statistical zero knowledge proofs. It is known that SRE $\subseteq$ SZK. We present an oracle separation which demonstrates that a containment of SZK in SRE cannot be proved via relativizing techniques.

1 Introduction

A randomized encoding (RE) of a function [13,5] allows one to represent a complex function $f(x)$ by a "simpler" randomized function, $\hat{f}(x,r)$, such that the "encoding" $\hat{f}(x,r)$ reveals $f(x)$ but no other information about x[1]. More specifically, there should exist an (unbounded) *decoder* that computes $f(x)$ given $\hat{f}(x,r)$, and an efficient randomized *simulator* that simulates the output of

* IIT Delhi. Email: shweta@cse.iitd.ac.in

** Technion. Email: yuvali@cs.technion.ac.il

*** UCLA and Center for Encrypted Functionalities. Email: dakshita@cs.ucla.edu.

† UCLA and Ariel University. Email: anatpc@ariel.ac.il

[1] It also reveals $|x|$. This is unavoidable, as otherwise the output of $\hat{f}$ is one of two disjoint distributions supported over a finite domain, which puts $f(x)$ in BPP.

the encoder $\hat{f}(x,r)$, only given $|x|$ and $f(x)$. We refer to the former decoding requirement as *correctness* and to the latter simulation requirement as *privacy*. Privacy can either be perfect, statistical, or computational, depending on the required notion of "closeness" between the simulated distribution and the output distribution of $\hat{f}$. The complexity class SRE (resp. PRE, CRE) is defined to be the class of boolean functions $f : \{0,1\}^* \to \{0,1\}$, or equivalently languages, admitting a randomized encoding $\hat{f}$ that can be computed in polynomial time and having statistical (resp. perfect, computational) privacy. In this paper, we initiate the study of the class SRE of functions admitting a statistical randomized encoding (SRE).

As a cryptographic primitive, randomized encodings were first studied explicitly by Ishai and Kushilevitz [13], although they were used implicitly in prior work in the context of secure multiparty computation [19,16,11]. They have found application in different areas of cryptography, such as parallel implementations of cryptographic primitives [5], verifiable computation and secure delegation of computations [6], secure multiparty computation [8,9,13,14,4], and even in algorithm design [15]. We refer the reader to [3] for a survey of such applications.

The *parallel complexity* of randomized encodings was studied by Applebaum et al. [5], who demonstrated that all functions in the complexity class NC^1 (and even certain functions that are conjectured not to be in NC [2]) admit an SRE in NC^0. This establishes a provable speedup in the context of parallel time complexity. It is natural to ask a similar question in the context of *sequential* time complexity. For which functions (if any) can an SRE enable a super-polynomial speedup? This question is the focus of our work.

Characterizing the class SRE. Let us consider the power of the class SRE of all functions admitting a polynomial-time computable statistical randomized encoding. It is evident that $\mathsf{P} \subseteq \mathsf{SRE}$, where $\hat{f}(x,r)$ simply outputs $f(x)$. This satisfies both the correctness and privacy requirements. But is $\mathsf{SRE} \subseteq \mathsf{P}$?

- SRE **for trivial hard languages.** First, we consider unary languages, i.e., languages $L \subseteq \{0\}^*$. These languages admit the trivial SRE defined by $\hat{f}(x) = x$. Indeed, the decoder can be defined by $D(z) = f(z)$ and the simulator, on input $(1^n, b)$, can output 0^n. Privacy holds since there is only one input of every length. However, such unary languages may not even be decidable, as illustrated for example by the language $\mathsf{U_{HP}}$ - the unary encoding of the halting problem, which admits an SRE but is not decidable. This example also extends to "trivial" binary languages such that for a given input length, all inputs are either in the language or not. However, note that such trivial languages are always contained in the class $\mathsf{P/poly}$, namely the class of functions admitting polynomial-size (but possibly non-uniform) circuits. This demonstrates that getting a candidate separation between SRE and P or even PSPACE is not enough; to demonstrate the power of randomized encodings over efficient computation in a meaningful way, we must separate the class SRE from $\mathsf{P/poly}$.

- **Is SRE more powerful that P/poly?** Let us now examine the relationship of SRE and P/poly. To begin, observe that for functions with long outputs, it is easy to find candidate functions that are not known to be efficiently computable by non-uniform circuits, but admit an efficient SRE. For example, assume there exists a family of one way permutations $\{f_n\}_{n\in\mathbb{N}}$ secure against non-uniform adversaries. Then the seemingly hard function $f^{-1}(x)$ can be encoded by the identity $\hat{f}^{-1}(x) = x$. As f^{-1} is also a permutation, this encoding is both private and correct. However, for boolean functions, the question looks much more interesting. To the best of our knowledge, no previous candidates for languages or promise problems that are conjectured to lie outside P/poly but admit efficient SRE have been proposed. This is one of the questions we study in this work.
- **Is SZK more powerful that SRE?** Another natural question about randomized encodings is their relationship with the class SZK of languages admitting statistical zero knowledge proofs. It is not hard to show that SRE $\subseteq$ SZK [2].[2] This implies that SRE is unlikely to contain NP. Based on current examples for SZK languages it seems likely that the containment SRE $\subseteq$ SZK is strict, but no formal evidence was given in this direction. This motivates the question of finding an oracle relative to which SZK is not contained in SRE.

Why is the class SRE *interesting?* As has been pointed out already, for functions that are efficiently computable, the SRE can just compute the function itself. Therefore, the class SRE is interesting only when the functions themselves are *not* efficiently computable, in which case the complexity of the decoder must inherently be super-polynomial. While most known applications of randomized encodings of functions require the decoder to be efficient, there are some applications that do not (see [3]). Moreover, even in cases where the decoder is required to be efficient, SRE functions can be "scaled down" so that decoding takes a feasible time T whereas encoding time is sub-polynomial in T. For instance, the computation of an SRE function can be delegated from a weak client to a powerful but untrusted server by directly applying an SRE on instances of a small size n, such that the server may be allowed to run in time $\mathsf{exp}(n)$ while the client is only required to run in time $\mathsf{poly}(n)$. Indeed, many real-life problems require exponential time to solve using the best known algorithms.

1.1 Our Results

Our results can be summarized as follows.

1. **Separating SRE from P/poly:**
   We provide three candidates to separate SRE from efficient computation.

[2] Here and in the following, when writing SRE $\subseteq$ SZK we restrict SRE to only contain languages L that are *non-trivial* in the sense that for every sufficiently large input length n there are inputs x_0, x_1 of length n such that $x_0 \in L$ and $x_1 \notin L$. This excludes languages such as the unary undecidable language mentioned earlier. The containment proof in [2] implicitly assumes non-triviality.

- We give a candidate *language*, for which we conjecture hardness based on a *non-standard* variant of the DDH assumption. We give an efficient SRE for this language which builds on the random self reduction for DDH demonstrated by Naor and Reingold [17].
- Next, we give a candidate (dense) *promise problem*, the hardness of which follows from the hardness of the *standard* Learning with Errors assumption. We devise an efficient SRE for this promise problem.
- Last, we design a non-uniform SRE for the Abelian subgroup membership ASM family of promise problems. This problem generalizes quadratic residuosity and (an instance of an augmented) co-DDH problem. We also give a specific instance of this promise problem, which is a language, and conjecture that this language is outside of $\mathsf{P/poly}$ based on a variant of co-DDH, an assumption introduced in [12].

2. **Separating SZK from SRE:** We show the existence of an oracle, relative to which $\mathsf{SZK} \not\subset \mathsf{SRE}$. This oracle separation implies that the containment $\mathsf{SZK} \subseteq \mathsf{SRE}$ (if true) cannot be proved via relativizing proof techniques.

1.2 Overview of Main Techniques

We now give an overview of the main techniques used for our separations.

Separating SRE *from* $\mathsf{P/poly}$. We provide several SRE constructions for problems that are conjectured to lie outside $\mathsf{P/poly}$. It may be helpful to point out here, that problems in SRE also admit an SZK proof, and the existence of hard problems in SZK implies the existence of one-way functions. Therefore, we cannot hope to get an unconditional result, or even one based on $\mathsf{P} \neq \mathsf{NP}$. We have the following candidates based on various assumptions, which we later summarize in Table 1.

- **Candidate language related to DDH.**
  Our first candidate is a language, which we call DDH', whose hardness is related to the Decisional Diffie Hellman (DDH) assumption. We consider inputs of the form $\langle g, g^a, g^b, g^c \rangle$ where g is any generator of a fixed DDH group per input length. Roughly, the input is in the language iff it corresponds to a DDH tuple, that is, if $g^c = g^{ab}$ in a fixed group generated by g.
  Our SRE for this problem builds on the random self-reduction given by Naor and Reingold [17] for DDH. However, not only do we randomize the DDH exponents following [17], but also randomize the generator of the DDH group. Finally, in order to devise a candidate language, we must fix the description of the group and its generator, given just the length of the input. We achieve this by suggesting an efficient, deterministic procedure to generate a DDH group and other parameters required by the encoding algorithm, given the input length. However, note that the hardness of DDH' cannot be reduced to the standard DDH. This is because DDH is an average case assumption, where the public parameters are chosen randomly. In our case, we must fix the public parameters per input length, and DDH does not guarantee that this restriction preserves hardness. We conjecture however, that DDH' remains infeasible for fixed parameters.

- **Dense promise problem based on LWE.**
  Our second example is a (dense) promise problem DLWE', whose hardness reduces to the hardness of the standard LWE problem. DLWE' approximately classifies noisy codewords $(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e})$ into Yes and No instances, depending upon on the size of the error vector $\mathbf{e}$. Roughly speaking, Yes instances correspond to small errors and No instances to large errors.
  Note that, an SRE encoding of input $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$ must be oblivious of all information about $\mathbf{A}, \mathbf{s}, \mathbf{e}$ except the relative size of the error vector $\mathbf{e}$. We begin by using the additive homomorphism of the LWE secret to mask $\mathbf{s}$. Specifically, we choose a random vector $\mathbf{t}$ and compute $\mathbf{b}' = \mathbf{b} + \mathbf{At} = \mathbf{A}(\mathbf{s} + \mathbf{t}) + \mathbf{e}$. Now, $\mathbf{b}'$ no longer retains information about $\mathbf{s}$. To hide $\mathbf{A}$, we multiply $(\mathbf{A}, \mathbf{b})$ by a random low norm matrix $\mathbf{R}$ and invoke the leftover hash lemma to argue that $\mathbf{RA}$ looks random even when $\mathbf{R}$'s entries are chosen from a relatively small range. For No instances, $\mathbf{e}$ is large enough that $\mathbf{Re}$ also hides $\mathbf{e}$ via LHL, but to hide the smaller $\mathbf{e}$ of Yes instances, we must add additional noise $\mathbf{r}_0$. This extra noise is large enough to hide $\mathbf{e}$ but not large enough to affect correctness. For more details, please see Section 3.1.

- **Generalizing QR, and candidate language related to co-DDH.**
  Our final candidate is the Abelian Subgroup Membership (promise) problem ASM, which generalizes the quadratic residuosity problem QR_N for composite modulus N. ASM is specified by an abelian group G, and a subgroup H of G, such that $I(G/H) = \mathbb{Z}_q^t$ for prime q, integer t and some isomorphism I. We define Yes instances to be well-formed $x \in H$, and No instances to be well-formed $x \in G \setminus H$. We note that $\mathsf{QR}_N \in \mathsf{P/poly}$, and therefore is not a candidate for separation. However, we present a different candidate language, which is an instance of ASM, and which we conjecture to lie outside $\mathsf{P/poly}$ based on a variant of the $\mathsf{co\text{-}DDH}$ assumption in [12].
  At a high level, our SRE for the generalized ASM promise problem is constructed as follows. Given input x,
  - Compute $y = x \cdot h$ for random $h \xleftarrow{\$} H$.
  - Pick random elements $(x_1, x_2, \ldots x_{t-1}) \xleftarrow{\$} G$.
    Define $\mathbf{X} = [I(x_1), \ldots, I(x_{t-1}), I(y)]$.
  - Pick $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_q^{t \times t}$. Output $\mathbf{R} \cdot \mathbf{X}$.

  The first step randomizes x *within* its coset[3], erasing all information except the coset of x. Next, observe that membership of x in the subgroup H is encoded by the rank of $\mathbf{X}$ – if $x \in H$ then $\mathbf{X}$ is singular, whereas if $x \notin H$, then $\mathbf{X}$ is non-singular with high probability. Thus, randomizing $\mathbf{X}$ via $\mathbf{RX}$ hides everything except the rank of $\mathbf{X}$, effectively erasing coset information about x. The decoder learns whether $x \in H$ by computing the rank of $\mathbf{RX}$. Finally, we amplify the privacy and correctness parameters by applying a generic masking technique, that may be of independent interest.

[3] This step is similar to the classic SRE for QR_p which encodes x by $x \cdot r^2$ for randomly chosen r. However, this is insufficient even for QR_N where N is composite (hence for ASM), as it leaks coset information of x.

Candidate	Language	Hardness
DDH'	Language	Non-Std DDH
DLWE'	(Dense) Promise Problem	Std LWE
$\mathsf{ASM}(\text{co-}\mathsf{DDH})$	Language*	Non-Std co-DDH

Table 1. Our Candidates. The SREs are uniform and private against non-uniform adversaries. If not a language, we exhibit a promise problem. The * denotes that a specific instance of ASM is a language, though ASM is in general a promise problem.

Separating SZK *from* SRE Applebaum [2] showed that any language that admits an SRE encoding also admits an SZK proof. This was done by reducing SRE to the statistical distance problem [18] which admits a two-round SZK protocol. The question of whether this containment is strict is still open.

We give an oracle separation between the classes SZK and SRE. We diagonalize over oracle SRE encoders to obtain a language that is not in oracle-SRE, but admits an oracle-SZK proof. Our technique involves generalizing the method of [1] that separates oracle-SZK machines from oracle-BPP machines, with the oracle being determined during diagonalization. This technique is reminiscent of the one in [7] showing that any proof for P=NP does not relativize. However, our setting diverges from that of [1] in two ways.

First, we diagonalize over SRE encoders such that decoders are unbounded. However, in the presence of unbounded machines, an oracle similar to [1] would be only as powerful as the plain model. To deal with this, we derive an alternate definition for SRE, where the output of PPT encoders falls into two distinct distributions over a polynomially large support (unlike binary output BPP machines). In order to derive an outlying language via diagonalization in this new setting, we must account for the size of the support. We stress here that our separation does not reduce to the $\mathsf{SZK} - \mathsf{BPP}$ separation in [1], and can in fact, be viewed as a generalization of their result.

1.3 Related Work

The classes PREN, SREN and CREN have been defined by Applebaum, Ishai and Kushilevitz [6] as the class of functions that admit perfect (resp. statistical, computational) randomized encodings in NC^0 with a polynomial-time decoder. In contrast, in this work we do not restrict the complexity of decoding the output. Applebaum [2] observed that $\mathsf{QR}_p \in \mathsf{SREN}$ while not known to be in NC, suggesting a separation between these classes.

Aiello and Håstad[1] gave a technique for the oracle separation of SZK from BPP, by diagonalizing over oracle-BPP machines. Our technique for the oracle separation of SZK from uniform SRE follows in their broad outline, but must be adapted to oracle-SRE machines whose outputs are over a large support. Also, note that SRE has been used in the past for reducing the complexity of complete problems for a subclass of SZK (more specifically, the class SZKP_L of problems having statistical zero-knowledge proofs where the honest verifier and its simulator are computable in logarithmic space) [10].

2 Preliminaries

In this section, we define basic notation and recall some definitions which will be used in our paper. Given a vector x, $|x|$ denotes its size. We let $\mathsf{size}(C)$ denote the size of a circuit C and $\mathsf{size}(f)$ denote the size of the smallest circuit computing f. The statistical distance between two distributions $\mathcal{X}$ and $\mathcal{Y}$ over space Ω, is defined as $\Delta(\mathcal{X},\mathcal{Y}) \equiv \frac{1}{2}\Sigma_{u\in\Omega}|\Pr_{X\sim\mathcal{X}}[X=u] - \Pr_{Y\sim\mathcal{Y}}[Y=u]|$.

The definition of a promise problem, the class $\mathsf{P/poly}$ (extended to also include promise problems) and the class SZK, are mostly standard in the literature.

We now formally define the notion of a statistical randomized encoding of a function, language or promise problem. Similarly to the previous definition from [5], our definition requires the encoding to be uniform by default.

Definition 1 (Statistical randomized encodings ((ϵ,δ)-SRE)). *[5] Let $f:\{0,1\}^* \to \{0,1\}^*$ be a function and $l(n)$ an output length function such that $|f(x)| = l(|x|)$ for every $x \in \{0,1\}^*$. We say that $\hat{f}:\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a $\epsilon(n)$-private $\delta(n)$-correct (uniform) statistical randomized encoding of f (abbreviated (ϵ,δ)-SRE), if the following holds:*

- **Length regularity.** *There exist polynomially-bounded and efficiently computable length functions $m(n), s(n)$ such that for every $x \in \{0,1\}^n$ and $r \in \{0,1\}^{m(n)}$, we have $|\hat{f}(x,r)| = s(n)$.*
- **Efficient encoding.** *There exists a polynomial-time* encoding algorithm *denoted by $\mathsf{enc}(\cdot,\cdot)$ that, given $x \in \{0,1\}^*$ and $r \in \{0,1\}^{m(|x|)}$, outputs $\hat{f}(x,r)$.*
- **δ-correctness.** *There exists an unbounded* decoder dec*, such that for every $x \in \{0,1\}^n$ we have $\Pr[\mathsf{dec}(1^n,\ \hat{f}(x,U_{m(n)})) \neq f(x)] \leq \delta(n)$.*
- **ϵ-privacy.** *There exists a probabilistic polynomial-time* simulator S*, such that for every $x \in \{0,1\}^n$ we have $\Delta(S(1^n,f(x)),\hat{f}(x,U_{m(n)})) \leq \epsilon(n)$.*

An (ϵ,δ)-SRE of a language $L \subset \{0,1\}^*$ *is an (ϵ,δ)-SRE of the corresponding boolean function $f:\{0,1\}^* \to \{0,1\}$. When ϵ and δ are omitted, they are understood to be negligible functions.*

Extensions. A *non-uniform* (ϵ,δ)-SRE of f is defined similarly, except that the encoding algorithm is implemented by a family of polynomial-size circuits. For a *partial function* f, defined over a subset $X \subseteq \{0,1\}^*$, the correctness and privacy requirements should only hold for every $x \in X$. An (ϵ,δ)-SRE of a promise problem $(\mathsf{Yes},\mathsf{No})$ is an (ϵ,δ)-SRE of the corresponding partial boolean function.

Definition 2 (The class SRE[4]). *The class SRE is defined to be the set of all languages that admit an SRE (namely, an (ϵ,δ)-SRE for some negligible ϵ,δ). For concrete functions $\epsilon(n),\delta(n)$, we use (ϵ,δ)-SRE to denote the class of languages admitting an (ϵ,δ)-SRE.*

[4] The difference between the class SRE and the class SREN defined in [5] is that SRE allows the encoding algorithm to run in polynomial time whereas SREN restricts the encoding algorithm to be in NC^0.

3 Separating SRE from Efficient Computation

We devise three candidates for separating SRE from efficient computation. In this section, we outline one candidate promise problem, that belongs to SRE and is unlikely to be in P/poly based on the standard LWE assumption.

We also devise a candidate language based on a non-standard, but plausible, hardness assumption related to DDH. The final candidate is based on the Abelian Subgroup Membership problem. Please refer to the full version for details on these candidates.

3.1 Learning With Errors (LWE)-based promise problem.

In this section, we devise a candidate *promise problem* DLWE' based on the hardness of the Learning with Errors (LWE) assumption.

Definition 3. $\mathsf{DLWE}' = \{\mathsf{Yes}, \mathsf{No}\}$ *where* Yes *and* No *are defined as follows.*

$$\mathsf{Yes} = \bigcup_n \mathsf{Yes}_n,\ \mathsf{No} = \bigcup_n \mathsf{No}_n$$

The parameters m, p, ϵ are set per input length n as $m = n^2, p = n^{40}, \delta = 0.05$.

$$\mathsf{Yes}_n \triangleq \Big\{(\mathbf{A}, \mathbf{As}+\mathbf{e}) \mid \mathbf{A} \in \mathbb{Z}_p^{m\times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in [-p^\delta,\ p^\delta]^m, \Delta(\mathcal{R}_\mathbf{A}, \mathcal{U}_{m\times n}) \leq p^{-0.16m}\Big\}$$

$$\mathsf{No}_n \triangleq \Big\{(\mathbf{A}, \mathbf{As}+\mathbf{e}) \mid \mathbf{A} \in \mathbb{Z}_p^{m\times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in \mathbb{Z}_p^m \setminus [-p^{2/3},\ p^{2/3}]^m,$$
$$\Delta\big((\mathcal{R}_\mathbf{A}, \mathcal{R}_\mathbf{e}), (\mathcal{U}_{m\times n}, \mathcal{U}_m)\big) \leq p^{-0.16m}\Big\} \setminus \mathsf{Yes}_n$$

Here, $\mathcal{R}_\mathbf{A}$ denotes the distribution $\mathbf{RA}(mod\ p)$ induced by choosing $\mathbf{R}$ uniformly in $[-p^{2/3}, p^{2/3}]^{m\times m}$. Similarly, $\mathcal{R}_\mathbf{e}$ denotes the distribution $\mathbf{Re}(mod\ p)$ induced by choosing $\mathbf{R}$ (same as before) uniformly in $[-p^{2/3}, p^{2/3}]^{m\times m}$. $\mathcal{U}_{m\times n}$ and $\mathcal{U}_m$ denote the uniform distribution in $\mathbb{Z}_p^{m\times n}$ and $\mathbb{Z}_p^m$ respectively.

We must explicitly subtract Yes_n from No_n because there may exist $\mathbf{s}, \mathbf{e}$ and $\tilde{\mathbf{s}}, \tilde{\mathbf{e}}$ such that $\mathbf{As}+\mathbf{e} = \mathbf{A}\tilde{\mathbf{s}}+\tilde{\mathbf{e}}$ and $\tilde{\mathbf{e}} \in (\mathbb{Z}_p \setminus [-p^{2/3},\ p^{2/3}])^m$ but $\mathbf{e} \in [-p^\delta,\ p^\delta]^m$, resulting in an overlap between the sets Yes_n and No_n. The condition involving the statistical distance is a technicality required for using the leftover hash lemma in the construction. The value $p^{-0.16m}$ in the definition is a representative inverse polynomial function in the input size n. We also define a new promise problem DLWE'' which is exactly the same as DLWE', except setting $p = 2^n$ for each input length n. The analysis of DLWE'' is the same except $p^{-0.16m}$ is $\mathsf{negl}(n)$.

It is easy to show that the hardness of DLWE' and DLWE'' against P/poly follows from the hardness of the standard decisional Learning with Errors problem DLWE for the same parameters.

Theorem 1. $\mathsf{DLWE}' \in (1/\mathsf{poly}, 1/\mathsf{poly})$-SRE *and* $\mathsf{DLWE}'' \in (\mathsf{negl}, \mathsf{negl})$-SRE.

Proof. We construct an SRE for DLWE' here. On input an instance of size n, the encoder, decoder, simulator compute parameters m, ϵ, δ, p as functions of n.

Encoding. The algorithm $\mathsf{enc}_{\mathsf{SRE}}(1^n, \mathbf{A}, \mathbf{b})$ is defined as follows.

1. Pick $\mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^{m\times m}$, $\mathbf{r}_0 \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]^m$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^n$.
2. Set $\mathbf{A}' = \mathbf{RA}$ and $\mathbf{b}' = \mathbf{r}_0 + \mathbf{Rb}$.
3. Output $(\mathbf{A}'', \mathbf{b}'') = (\mathbf{A}', \mathbf{A}'\mathbf{t} + \mathbf{b}')$.

Decoding. The algorithm $\mathsf{dec}_{\mathsf{SRE}}(1^n, \mathbf{A}'', \mathbf{b}'')$ accepts if and only if there exist $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{e} \in \mathbb{Z}_p^m$, such that $\mathbf{b}'' = \mathbf{A}''\mathbf{x} + \mathbf{e}''$, and $\mathbf{e}'' \in [-p^{2/3+4\delta}, p^{2/3+4\delta}]$.

Simulation. On input 1^n and a bit b where $b = 0/1$ represents membership in Yes/No respectively, the simulator does the following.

- If $b = 0$, pick $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_p^{m\times n}$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]^m$. Output $(\mathbf{U}, \mathbf{Ut}+\mathbf{e})$.
- If $b = 1$, pick $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_p^{m\times n}$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^m$. Output $(\mathbf{U}, \mathbf{u})$.

Analysis. We give a brief overview of the correctness and privacy arguments. Recall that,

$$\mathsf{enc}_{\mathsf{SRE}}(1^n, \mathbf{A}, \mathbf{As}+\mathbf{e}) = \Big(\mathbf{RA},\ \mathbf{RA}(\mathbf{s}+\mathbf{t}) + (\mathbf{Re}+\mathbf{r}_0)\Big) \quad \text{where}$$

$$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^n,\ \ \mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^{m\times m},\ \ \mathbf{r}_0 \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]^m.$$

Thus, the secret in $\mathbf{b}''$, namely $\mathbf{s}+\mathbf{t}$, is distributed uniformly in $\mathbb{Z}_p^n$.

- **Case 1:** $(\mathbf{A}, \mathbf{As}+\mathbf{e}) \in \mathsf{Yes}_n$. In this case, $\mathbf{e} \in [-p^\delta, p^\delta]^m$.
  Then, for $\mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^m$, $\mathbf{Re} \in [-p^{2/3+2\delta}, p^{2/3+2\delta}]^m$.
  Moreover, by choice of $\mathbf{r}_0$, we have $\mathbf{Re} << \mathbf{r}_0$, thus $\Delta\big(\mathbf{Re}+\mathbf{r}_0,\ \mathbf{r}_0\big) \le p^{-\delta m}$.
  By definition of the promise problem, we have that $\Delta(\mathcal{R}_{\mathbf{A}}, \mathbf{U}_{m\times n}) \le p^{-0.16m}$.
  Then the following hold:
  - *Correctness.* $\mathbf{Re}+\mathbf{r}_0 \in [-p^{2/3+4\delta}, p^{2/3+4\delta}]$. Thus, correctness is perfect.
  - *Privacy.* By the above arguments on the distribution of $(\mathbf{RA})$, $(\mathbf{s}+\mathbf{t})$ and $(\mathbf{Re}+\mathbf{r}_0)$, and by the simulator's choice of $(\mathbf{U}, \mathbf{t}, \mathbf{e})$, we can argue that the output distribution is at most $p^{-0.16m}$-far from the distribution induced by $\mathsf{SRE.enc}$ on an instance of Yes_n.
- **Case 2:** $(\mathbf{A}, \mathbf{As}+\mathbf{e}) \in \mathsf{No}_n$. We have that $\mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3}, p^{2/3}])^m$ and $\Delta((\mathcal{R}_{\mathbf{A}},\ \mathcal{R}_{\mathbf{e}}), (\mathbf{U}_{m\times n},\ \mathbf{u}_m)) \le p^{-0.16m}$. Then the following hold:
  - *Correctness.* Standard averaging arguments prove that all entries of $\mathbf{Re}+\mathbf{r}_0$ are larger than $p^{2/3+4\delta}$ with probability $\ge 1-p^{-0.13m}$. Moreover, the probability that randomizing an instance in No_n results in an encoding that corresponds to some 'small' error vector[5], $\le p^{-\delta m}$. Overall, we obtain $p^{-0.1m}$-correctness.
  - *Privacy.* We show that a random sample $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathbb{Z}_p^{m\times n} \times \mathbb{Z}_p^m$ (simulator output) is $(1-p^{-0.1m})$ close to the distribution induced by $\mathsf{SRE.enc}$ on a No_n instance. First, we show that randomly chosen $(\mathbf{A}, \mathbf{b})$ are such that, w.h.p. there exist no ($\mathbf{s}$, small[5] $\mathbf{e}$) such that $\mathbf{b} = \mathbf{As}+\mathbf{e}$. We also prove that w.h.p. $\mathbf{A}, \mathbf{e}$ corresponding to random $(\mathbf{A}, \mathbf{b})$ are such that the distributions $\mathcal{R}_{\mathbf{A}}$ and $\mathcal{R}_{\mathbf{e}}$ are close to uniform.

[5] Here, 'small' denotes error of magnitude less than $p^{2/3+4\delta}$, such that the instance wrongly decodes to Yes. However, in the rest of the paper, 'small' denotes error $\le p^\delta$.

4 Oracle Separation Between SRE and SZK

In this section, we crucially use the following Lemma about the class (ϵ, δ)-SRE. This Lemma follows directly from the definition of (ϵ, δ)-SRE.

Lemma 1. *Let $\mathcal{E}_x$ denote the distribution $\mathsf{enc}(x, r)$ for the algorithm $\mathsf{enc}(\cdot, \cdot)$ of a language L admitting an (ϵ, δ)-SRE, induced for any input x by picking r uniformly at random in $\{0,1\}^*$. Then, $\Delta(\mathcal{E}_x, \mathcal{E}_{x'}) \leq 2\epsilon$ iff $f(x) = f(x')$ (equivalently, both $x, x' \in L$ or both $x, x' \notin L$). Moreover, $\Delta(\mathcal{E}_x, \mathcal{E}_{x'}) \geq 1 - 2\delta$ iff $f(x) \neq f(x')$ (equivalently, either $x \in L, x' \notin L$ or $x \notin L, x' \in L$).*

In this section, we study the relation between the classes SRE and SZK. We recall the following theorem from [2].

Imported Theorem 1. *[2] Any non-trivial language that admits an (ϵ, δ)-SRE such that $(1 - 2\delta)^2 > 2\epsilon$, also admits an SZK proof.*

Here, we explore whether the containment $\mathsf{SRE} \subseteq \mathsf{SZK/poly}$ is strict. We give an oracle separation between the classes SZK (more precisely, the class SZK[2] of languages that admit a 2-round SZK proof - note that this is the strongest separation) and SRE, but restricted to the uniform setting. For any oracle A, we denote by SRE^A the class SRE where encoders have oracle access to A. Similarly, we denote by SZK^A the class SZK where verifiers have oracle access to A.

Theorem 2. *There exists an oracle A, such that $\mathsf{SZK}[2]^A \not\subset \mathsf{SRE}^A$.*

Proof Overview. Broadly, we diagonalize over all oracle SRE-encoder machines to obtain a language which does not have any SRE encoding. We construct this language in rounds, one for each input length. Specifically, we will ensure that for every input length n, the output of the encoder on inputs 0^n and 1^n is either less than $(1 - 2\delta)$ or more than ϵ, violating the definition of SRE from Lemma 1[6].

This is done via classifying the characteristic vector of the language into unique and redundant sets, such that it is impossible for any encoder with polynomially many oracle queries to distinguish between unique versus redundant characteristic. Moreover, a contrived language is set such that 0^n is never in the language, and 1^n is in the language iff the characteristic vector is unique.

Intuitively, since encoders cannot distinguish between a unique versus redundant characteristic, one of the following cases will always occur. Either, there exists a redundant characteristic (implying that both 0^n and 1^n are not in the language) such that the encodings of 0^n and 1^n are more than ϵ-apart; or, there exists a unique characteristic (implying that 1^n is in the language while 0^n is not) such that the encodings of 0^n and 1^n are less than $(1 - 2\delta)$-apart. We set the language according to whichever of these cases is true. This ensures that the output of the encoders is not an SRE for this language.

[6] It is interesting to note that unlike the BPP-SZK [1] separation, a unary language is not helpful for separation since such a language will always have an SRE. Thus, our contrived language will be non-trivial and binary.

However, proving either of the two cases is true is significantly more involved than in the BPP setting of [1] (refer to the full version for details). Finally, we can show that this language has an SZK proof, this follows in a similar manner as [1].

5 Conclusion and Open Problems

In this paper, we study the class SRE of languages and promise problems that admit efficient statistical randomized encodings. We present the first candidates for SRE problems that are not in P/poly. These include a candidate promise problem based on the hardness of standard LWE, as well as candidate languages based on variants of the DDH assumption and the co-DDH assumption of [12].

Then, we explore the relationship of the class SRE with the class SZK of languages admitting statistical zero knowledge proofs. While it is known that all non-trivial languages in SRE are also in SZK [2], whether the converse holds is open. However, we exhibit an oracle and a (non-trivial) language that has an oracle-based SZK proof but does not have an oracle-based SRE. This shows that a containment of SZK in SRE cannot be proved via relativizing techniques.

Several natural questions remain open. The first is to identify a complete language in SRE, thereby obtaining a better characterization of this class. A second is to better understand the relation between statistical randomized encodings and random self-reductions (RSR). An RSR for a language or a promise problem can be viewed as a restricted form of SRE where the decoder just decides the problem itself. Our LWE-based language is a candidate for a problem in SRE which is not in RSR, thus supporting the conjecture that $\mathsf{RSR} \subset \mathsf{SRE}$. Is there an oracle separating these classes? Finally, it would be interesting to find additional (and preferably "useful") candidates for intractable problems in SRE, as well as natural polynomial-time solvable problems for which an SRE can provide polynomial speedup over the best known algorithms.

References

1. Aiello, W., Håstad, J.: Relativized perfect zero knowledge is not BPP. Inf. Comput.
2. Applebaum, B.: Cryptography in Constant Parallel Time. Ph.D. thesis, Technion
3. Applebaum, B.: Randomly encoding functions: A new cryptographic paradigm - (invited talk). In: Fehr, S. (ed.) ICITS. Lecture Notes in Computer Science, vol. 6673. Springer (2011)
4. Applebaum, B., Ishai, Y., Kushilevitz, E.: Computationally private randomizing polynomials and their applications. In: IEEE Conference on Computational Complexity. pp. 260–274. IEEE Computer Society (2005)
5. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC0. SIAM J. Comput. 36(4), 845–888 (2006)
6. Applebaum, B., Ishai, Y., Kushilevitz, E.: From secrecy to soundness: Efficient verification via secure computation. In: ICALP. pp. 152–163. Springer-Verlag, Berlin, Heidelberg (2010)

7. Baker, T.P., Gill, J., Solovay, R.: Relativizatons of the P =? NP question. SIAM J. Comput. 4(4), 431–442 (1975)
8. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC. ACM (1988)
9. Chaum, D., Crépeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: STOC. pp. 11–19. ACM, New York, NY, USA (1988)
10. Dvir, Z., Gutfreund, D., Rothblum, G.N., Vadhan, S.: On approximating the entropy of polynomial mappings. In: ICS. pp. 460–475 (2011)
11. Feige, U., Killian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: STOC. pp. 554–563 (1994)
12. Galbraith, S.D., Rotger, V.: Easy decision-diffie-hellman groups
13. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: FOCS. pp. 294–304. IEEE Computer Society (2000)
14. Ishai, Y., Kushilevitz, E.: Perfect constant-round secure computation via perfect randomizing polynomials. In: ICALP. Lecture Notes in Computer Science, vol. 2380, pp. 244–256. Springer (2002)
15. Ishai, Y., Kushilevitz, E., Paskin-Cherniavsky, A.: From randomizing polynomials to parallel algorithms. In: ITCS. ACM, New York, NY, USA (2012)
16. Kilian, J.: Founding crytpography on oblivious transfer. In: STOC. pp. 20–31. ACM, New York, NY, USA (1988)
17. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM 51(2) (Mar 2004)
18. Sahai, A., Vadhan, S.: A complete problem for statistical zero knowledge. J. ACM 50(2), 196–249 (Mar 2003), http://doi.acm.org/10.1145/636865.636868
19. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: FOCS. pp. 162–167 (1986)