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 [20,5] 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 [20], although they were used implicitly in prior work in the context of secure multiparty computation [32,23,15]. 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 [9,11,20,21,4], and even in algorithm design [22]. 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 the this language which builds on the random self reduction for DDH demonstrated by Naor and Reingold [24].
- 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 [16].

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 [24] for DDH. However, not only do we randomize the DDH exponents following [24], but also randomize the generator of the DDH group. Finally, in order to 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 [16].
  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 [29] 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 [8] 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) [14].

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 recall their definitions in Appendix A for completeness.

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. This candidate is outlined in Appendix B. The details of another candidate based on the Abelian Subgroup Membership problem, are in Appendix E.

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 $\mathbf{R}$ 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. The details are in Appendix C.2.

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. The complete proof is in Appendix C. Recall that,

$$\mathsf{enc}_{\mathsf{SRE}}(1^n, \mathbf{A}, \mathbf{As}+\mathbf{e}) = \Big(\mathbf{RA},\ \mathbf{RA}(\mathbf{s}+\mathbf{t}) + (\mathbf{Re}+\mathbf{r}_0)\Big) \quad \text{where}$$
$$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^n,\ \ \mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^{m\times m},\ \ \mathbf{r}_0 \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]^m.$$

Note that 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 $\mathbf{e} \in \mathbb{Z}_p^m \setminus [-p^{2/3}, p^{2/3}]^m$ and $\Delta\big((\mathcal{R}_{\mathbf{A}},\ \mathcal{R}_{\mathbf{e}}), (\mathbf{U}_{m\times n},\ \mathbf{u}_m)\big) \le p^{-0.16m}$. Then the following hold:
  - *Correctness.* By standard averaging arguments, we prove that all entries of $\mathbf{Re}+\mathbf{r}_0$ are larger than $p^{2/3+4\delta}$ with probability $\ge 1 - p^{-0.13m}$. Next, we bound away the probability that randomizing an instance in No_n puts it into the set Yes_n. Note that this may happen if the randomized instance, with secret and error vectors $(\mathbf{s}, \mathbf{e})$, may also be expressed with some $(\tilde{\mathbf{s}}, \tilde{\mathbf{e}})$ where $\tilde{\mathbf{e}}$ is small. We show that the fraction of such values is at most $p^{-\delta m}$, which together with the above, yields $p^{-0.1m}$-correctness.

- *Privacy.* We establish that a random sample $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathbb{Z}_p^{m \times n} \times \mathbb{Z}_p^m$ is $(1 - p^{-0.1m})$ close to the distribution induced by $\mathsf{SRE.enc}$ on a No_n instance. We do this in two broad steps. First, we show that randomly chosen $(\mathbf{A}, \mathbf{b})$ are such that, w.h.p. $\mathbf{b}$ can be expressed as $\mathbf{As} + \mathbf{e}$ for some $\mathbf{s}$ and large error $\mathbf{e}$. Here, we must be careful to exclude instances $\mathbf{b}$ that can be seen as having small error for a different secret $\mathbf{s}'$. Second, we establish that $\mathbf{A}, \mathbf{e}$ corresponding to the instance are "good" for LHL w.h.p. – i.e., the probability that $\mathbf{RA}$ or $\mathbf{Re}$ is not uniform is small.

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 $x \notin L, x' \in L$).*

Now, we study the relation between the classes SRE and SZK.

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.*

Next, we explore whether the containment 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[5].

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.

[5] 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.

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.

However, proving either of the two cases is true is significantly more involved than in the BPP setting of [1] (see Appendix D.3). Finally, we can show that this language has an SZK proof. The full proof of Theorem 2 is in Appendix D.

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 [16].

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 RSR $\subset$ 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. Babai, L.: Local expansion of vertex-transitive graphs and random generation in finite groups. In: STOC. pp. 164–174 (1991)
8. Baker, T.P., Gill, J., Solovay, R.: Relativizatons of the P =? NP question. SIAM J. Comput. 4(4), 431–442 (1975)
9. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. STOC '88, ACM
10. Buhrman, J., Kaas, R.: Mean, median and mode in binomial distributions. Statistica Neerlandica 34, 13–18 (1980)
11. Chaum, D., Crépeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. pp. 11–19. STOC '88, ACM, New York, NY, USA (1988)
12. Cramer, H.: On the order of magnitude of the difference between consecutive prime numbers. Acta Arith 2 pp. 23–46 (1936)
13. Döttling, N., Müller-Quade, J.: Lossy codes and a new variant of the learning-with-errors problem. In: Eurocrypt (2013)
14. Dvir, Z., Gutfreund, D., Rothblum, G.N., Vadhan, S.: On approximating the entropy of polynomial mappings. In: In Proceedings of the 2nd Innovations in Computer Science Conference. pp. 460–475 (2011)
15. Feige, U., Killian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing. pp. 554–563. STOC '94, New York, NY, USA (1994)
16. Galbraith, S.D., Rotger, V.: Easy decision-diffie-hellman groups
17. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
18. Hazewinkel, Michiel, e..: Riemann hypothesis, generalized (2001)
19. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing. STOC '89 (1989)
20. 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)
21. 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)
22. Ishai, Y., Kushilevitz, E., Paskin-Cherniavsky, A.: From randomizing polynomials to parallel algorithms. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ITCS '12, ACM, New York, NY, USA (2012)
23. Kilian, J.: Founding crytpography on oblivious transfer. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. pp. 20–31. STOC '88, ACM, New York, NY, USA (1988)
24. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM 51(2) (Mar 2004)

25. Nicely, T.R.: New maximal prime gaps and first occurrences. Mathematics of Computation 68 (227) p. 1311–1315 (1999)
26. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J.ACM 56(6) (2009), extended abstract in STOC'05
27. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
28. Rivest, R.L., Silverman, R.: Are 'strong' primes needed for RSA. IACR Cryptology ePrint Archive 2001, 7 (2001)
29. 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
30. Stadler, M.: Publicly verifiable secret sharing. pp. 190–199. Springer-Verlag (1996)
31. Wang, Y.: On the least primitive root of a prime. Sientia Sinica, 10(1) pp. 1–14 (1961)
32. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: FOCS. pp. 162–167 (1986)

A Preliminaries

Definition 4 (Promise Problem). *A promise problem Π is a pair of non-intersecting sets, denoted* $(\mathsf{Yes}, \mathsf{No})$ *where* $\mathsf{Yes}, \mathsf{No} \in \{0,1\}^*$ *and* $\mathsf{Yes} \cap \mathsf{No} = \emptyset$. *The set* $\mathsf{Yes} \cup \mathsf{No}$ *is called the promise.*

We let $\mathsf{P/poly}$ denote the class of languages that can be recognized by a polynomial-size circuit family. Depending on the context, we may extend the definition of $\mathsf{P/poly}$ to include promise problems as well.

Statistical Zero Knowledge (SZK). Next, we define an interactive proof system and statistical zero knowledge, these are well-known definitions taken verbatim from existing literature.

Definition 5 (Interactive Proof System [17]). *An interactive proof system for a language L, is a protocol between a probabilistic unbounded Turing Machine, the "Prover" P, and a PPT Turing Machine, the "Verifier" V, who receive a common input x. L is said to be in $IP[r]$ if after at most r rounds of interaction, the verifier outputs either accept or reject, such that:*

- *(Completeness) For all $x \in L$, the probability (over the randomness of P and V) that V accepts, is at least $1 - 2^{-|x|}$.*
- *(Soundness) For all $x \notin L$, the probability (over the randomness of V and arbitrary prover strategy P) that V rejects, is at least $1 - 2^{-|x|}$.*

The prover and verifier may exchange at most $poly(|x|)$ messages. Note that the verifier's coin tosses are allowed to be private.

Definition 6 (Statistical Zero Knowledge Proof [17]). *A statistical zero knowledge (SZK) proof for a language L is an interactive proof (according to Definition 5) for L between a probabilistic unbounded Turing Machine, the "Prover" P, and a PPT Turing Machine, the "Verifier" V, such that for all k, there exists N such that for all verifiers V, there exists a PPT simulator S_V where for all $x \in L, |x| > N$: $\Delta(S_V(x), V(x)) \leq \frac{1}{|x|^k}$.*

Definition 7 (The class SZK). *The class SZK is naturally defined to be the set of all languages which admit an SZK proof according to Definition 6.*

B SRE for DDH′ based on Decisional Diffie Hellman

Here, we devise a candidate *language* DDH' for separation whose hardness can be conjectured based on its similarity to the Decisional Diffie Hellman assumption.

Candidate Language. Our language DDH' is defined as follows.

Definition 8. *For every input length n, fix p_n to be a prime of size $\lfloor n/4 \rfloor$, such that $(p_n - 1)$ has prime factor $q_n > \sqrt{p_n}$. Then, define candidate language*

$$\mathsf{DDH}' = \bigcup_{n \in \mathbb{N}} \{\langle g, g^a, g^b, g^{a \cdot b} \rangle : \{a, b\} \in \mathbb{Z}_{q_n} \text{ and } g \text{ is an element of order } q_n \text{ in } \mathbb{Z}^*_{p_n}\}$$

Now, we describe a deterministic strategy to compute prime p_n for every input size n. A prime p_n such that $p_n - 1$ has a prime factor greater than $\sqrt{p_n}$ is called a cryptographic strong prime. These primes find wide applications in cryptography, and were introduced first in context of the RSA cryptosystem. The procedure for finding such primes is easy and has been outlined in [27,28]. We briefly discuss this procedure below. Start with $2^{\lfloor n/8 \rfloor - 1} + 1$ and sequentially test all integers for primality. Set the first such prime to p_n^{--}. Compute p_n^- as the least prime of the form $p_n^- = a_n^{--} p_n^{--} + 1$, for some integer a_n^{--}. This can be accomplished by trying $a_n^{--} = 2, 4, 6 \ldots$ until a suitable prime p_n^- is found. In a similar manner, compute prime $p_n = a_n^- p_n^- + 1$ by trying integer $a_n^- = 2, 4, 6 \ldots$. Set q_n to be p_n^-. From [27,28], we know that assuming the extended Reimann Hypothesis, such a prime will be found in $\mathsf{poly}(n)$ attempts.

We conjecture that DDH' is hard because it is similar to the standard DDH assumption. We briefly recall this assumption here.

DDH Assumption. Let p be a random n-bit prime, q a prime divisor of $p - 1$ and g be an element of order q in $\mathbb{Z}^*_p$. Then, no PPT Turing Machine with polynomial advice, on input a tuple $\langle p, q, g, x, y, z \rangle$ can decide whether there exist $\{a, b\} \in \mathbb{Z}_q$ such that $x = g^a, y = g^b, z = g^{a \cdot b}$.

The DDH assumption is equivalent [30,24] to the more popular DDH(II) assumption which is as follows. Let p be a random n-bit prime, q a prime divisor of $p - 1$, g an element of order q in $\mathbb{Z}^*_p$ and $a, b, c \in \mathbb{Z}_q$. Then the distinguishing advantage of any PPT Turing Machine with polynomial advice, that gets as input a tuple $\langle p, q, g, g^a, g^b, g^c \rangle$, is $\mathsf{negl}(n)$ between the following two distributions on the input: $\{\langle p, q, g, g^a, g^b, g^c \rangle \mid c = a \cdot b\}$ and $\{\langle p, q, g, g^a, g^b, g^c \rangle \mid c \leftarrow \mathbb{Z}_Q\}$.

However, the hardness of DDH' does not follow from DDH, because we deterministically *fix* p, q per n. On the other hand, in DDH, these parameters come from a distribution. While fixing parameters allows us to get a uniform SRE, we only conjecture that DDH' remains hard.

SRE Construction for DDH′.

Theorem 3. $\mathsf{DDH}' \in \mathsf{SRE}$*, if the extended Reimann Hypothesis (ERH) holds.*

We import the following Theorem from [24].

Imported Lemma 1. *[24] There exists a probabilistic polynomial time algorithm R such that on any input $\langle p, q, g, g^a, g^b, g^c \rangle$ where p is prime, q is a prime divisor of $p-1$, g an element of order q in $\mathbb{Z}_p^*$ and a, b, c are three elements in $\mathbb{Z}_q$ the output of R is $\langle p, q, g, g^{a'}, g^{b'}, g^{c'} \rangle$ where if $c = ab$ then a' and b' are uniform in $\mathbb{Z}_q$ and $c' = a'b'$, and if $c \neq ab$ then a', b', c' are all uniform in $\mathbb{Z}_q$.*

Proof of Theorem 3. We construct an SRE for DDH' in the following manner.

- $\mathsf{enc}_{\mathsf{SRE}}(v = \langle g, x, y, z \rangle)$
  - Compute the primes p_n, q_n as a deterministic function of the input length $n = |v|$, as in the description of language DDH'.
  - Compute $(p_n, q_n, g, x', y', z') = R(p_n, q_n, g, x, y, z)$ where R is from Imported Lemma 1[6].
  - Pick $r \leftarrow \mathbb{Z}_q$ and output $\langle g^r, (x')^r, (y')^r, (z')^r \rangle$ modulo $\mathbb{Z}_p^*$.
- $\mathsf{dec}_{\mathsf{SRE}}(\hat{v} = \langle \hat{g}, \hat{x}, \hat{y}, \hat{z} \rangle)$
  - Compute the primes p_n, q_n as a deterministic function of the input length $n = |\hat{v}|$, as in the description of language DDH'.
  - Check that $\hat{g}$ is of order q in $\mathbb{Z}_p$, and that there exist $a, b \in \mathbb{Z}_q$ such that $\hat{x} = \hat{g}^a, y = \hat{g}^b, z = \hat{g}^{a.b}$. If yes, accept. Otherwise, reject.

Correctness. It is easy to see that if $v \in L$, then $\mathsf{dec}_{\mathsf{SRE}}(\mathsf{enc}_{\mathsf{SRE}}(v))$ always accepts. If $v \notin L$, then $\Pr[\mathsf{dec}_{\mathsf{SRE}}(\mathsf{enc}_{\mathsf{SRE}}(v)) \text{ rejects}] = 1 - \mathsf{negl}(n)$.

Privacy. The simulator $S(1^n)$ generates primes p, q as a deterministic function of the input length n. Next, it samples $g \leftarrow \mathbb{Z}_p$ such that $g^q = 1$ (this can be done in probabilistic polynomial time). If $v \in L$, S outputs $(g^a, g^b, g^{a.b})$ for a, b uniform in $\mathbb{Z}_q$ and if $v \notin L$, S outputs (g^a, g^b, g^c) for a, b, c uniform in $\mathbb{Z}_q$. Perfect privacy follows from Imported Lemma 1, and re-randomization of generator g. □

[6] Note that proof of Imported Lemma 1 is constructive, in that [24] given an efficient construction the algorithm R.

C SRE for DLWE′

C.1 Preliminaries

We first state (without proof) the definition of Universal Hash Functions and the Leftover Hash Lemma [19].

Definition 9. *A family $\mathcal{H}$ of hash functions $h : \{0,1\}^n \to \{0,1\}^\ell$ is called* universal *if, for every $x, y \in \{0,1\}^n$ with $x \neq y$,*

$$\Pr_{h \in \mathcal{H}}[h(x) = h(y)] \leq 2^{-\ell}$$

Lemma 2 (Leftover Hash Lemma(LHL)[19]). *Let X be a random variable with universe U and $H_\infty(X) \geq k$. Fix $\epsilon > 0$. Let $\mathcal{H}$ be a universal hash family of size 2^d with output length $\ell = k - 2\log(1/\epsilon)$. Define*

$$Ext(x, h) = h(x)$$

Then Ext is a strong $(k, \epsilon/2)$ extractor with seed length d and output length ℓ. Equivalently, the statistical distance $\Delta((h, h(x)), (h, U)) \leq \epsilon/2$.

C.2 Hardness of DLWE′ from standard DLWE

Here, we show that hardness of DLWE' against $\mathsf{P/poly}$ follows from the hardness of Learning with Errors (LWE). LWE is the problem of determining a secret vector $\mathbf{s}$ over $\mathbb{Z}_q$ given a polynomial number of "noisy" inner products on $\mathbf{s}$. The decisional variant DLWE is to distinguish such samples from random. Formally, the (average-case) problem is defined as:

Definition 10 ([26]). *Let $n \geq 1$ and $p \geq 2$ be integers, and let χ be a probability distribution on $\mathbb{Z}_p$. For $\mathbf{s} \in \mathbb{Z}_p^n$, let $A_{\mathbf{s},\chi}$ be the probability distribution on $\mathbb{Z}_p^n \times \mathbb{Z}_p$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_p^n$ uniformly at random, choosing $e \in \mathbb{Z}_p$ according to χ, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$.*

The $\mathsf{DLWE}_{p,n,\chi}$ problem is: for uniformly random $\mathbf{s} \in \mathbb{Z}_p^n$, given a $\mathsf{poly}(n)$ number of samples that are either (all) from $A_{\mathbf{s},\chi}$ or (all) uniformly random in $\mathbb{Z}_p^n \times \mathbb{Z}_p$, output 0 if the former holds and 1 if the latter holds. We say the $\mathsf{DLWE}_{p,n,\chi}$ problem is infeasible *if for all polynomial-time algorithms $\mathcal{A}$, the probability that $\mathcal{A}$ solves the DLWE problem (over $\mathbf{s}$ and $\mathcal{A}$'s random coins) is negligibly close to $1/2$ as a function of n.*

Imported Theorem 2. *(LWE is hard for polynomial samples and sublinear uniform error[13]) Let n be a security parameter, $p = p(n)$ be an integer modulus, $m = m(n) = \mathsf{poly}(n)$ be an integer with $m \geq 3n$ and $r \geq n^{1/2+\epsilon} \cdot m$ be an integer such that $r/p \in (0, 1/10)$, and $\epsilon > 0$ is an arbitrary small constant. Then the LWE problem with parameters n, p and uniformly distributed errors in $[-r, r]^m$ is at least as hard (quantumly) as solving worst case problems on $(n/2)$-dimensional lattices to within a factor $O(n^{1+\epsilon} \cdot mp/r)$.*

For our purposes, the error distribution χ is the uniform distribution upto a maximum size of $r = \epsilon \cdot p$ where ϵ is chosen to satisfy $\epsilon \cdot p \geq n^{1/2+\epsilon} \cdot m$.

Now, we show that hardness of DLWE' follows from the hardness of DLWE with the same parameters.

Theorem 4. *For security parameter n and $p = n^{40}, \epsilon = p^{-0.95}, m = n^2$, the DLWE' problem according to Definition 3 is not in $\mathsf{P/poly}$ if the DLWE problem according to Definition 10 is hard for the same parameters.*

Proof. Assume that there exists an efficient adversary $\mathcal{B}$ who given $(\mathbf{A}, \mathbf{b}) \in \mathsf{Yes} \cup \mathsf{No}$ (always) correctly classifies whether $(\mathbf{A}, \mathbf{b}) \in \mathsf{Yes}$, or $(\mathbf{A}, \mathbf{b}) \in \mathsf{No}$, we construct an adversary $\mathcal{D}$ who distinguishes DLWE with non-negligible advantage as follows.

$\mathcal{D}$ receives $(\mathbf{A}, \mathbf{b})$ where either $(\mathbf{A}, \mathbf{b} \leftarrow \mathcal{A}^m_{\mathbf{s},\chi})$ or $(\mathbf{A}, \mathbf{b} \leftarrow \mathcal{U}_{m\times n} \times \mathcal{U}_m)$ and invokes $\mathcal{B}\big(\mathbf{A}, \mathbf{b}\big)$. If $\mathcal{B}$ accepts $((\mathbf{A}, \mathbf{b}) \in \mathsf{Yes})$, then $\mathcal{D}$ outputs 1, else $\mathcal{D}$ outputs 0.

We can prove (refer Lemma 3 and Lemma 6), that $\Pr\left[(\mathbf{A}, \mathbf{b}) \in \mathsf{Yes} | (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{A}^m_{\mathbf{s},\chi}\right] = 1 - \mathsf{negl}(n)$ and $\Pr\left[(\mathbf{A}, \mathbf{b}) \in \mathsf{No} | (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{U}_{m\times n} \times \mathcal{U}_m\right] = 1 - 2/n$. It follows that

$$\begin{aligned}
&\Pr\left[\mathcal{D} \text{ outputs } 0 \mid (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{A}^m_{\mathbf{s},\chi}\right] \\
&\geq \Pr\left[(\mathbf{A}, \mathbf{b}) \in \mathsf{Yes} \mid (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{A}^m_{\mathbf{s},\chi}\right] \cdot \Pr\left[\mathcal{B} \text{ accepts} \mid (\mathbf{A}, \mathbf{b}) \in \mathsf{Yes}\right] \\
&= (1 - \mathsf{negl}(n)) \qquad \textit{and,}
\end{aligned}$$

$$\begin{aligned}
&\Pr\left[\mathcal{D} \text{ outputs } 1 \mid (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{U}_{m\times n} \times \mathcal{U}_m\right] \\
&\geq \Pr\left[(\mathbf{A}, \mathbf{b}) \in \mathsf{No} \mid (\mathbf{A}, \mathbf{b}) \leftarrow \mathcal{U}_{m\times n} \times \mathcal{U}_m\right] \cdot \Pr\left[\mathcal{B} \text{ rejects} \mid (\mathbf{A}, \mathbf{b}) \in \mathsf{No}\right] \\
&= 1 - \frac{2}{n} - \mathsf{negl}(n) \geq 1 - \frac{3}{n}
\end{aligned}$$

Thus, $\mathcal{D}$ has non-negligible distinguishing advantage.

C.3 Correctness of SRE for DLWE'

In this section, we provide the details in the correctness analysis of our SRE for DLWE'.

Correctness of Yes encodings

This follows directly by observing that if $\mathbf{e} \in [-p^\delta, p^\delta]^m, \mathbf{R} \in [-p^{2/3}, p^{2/3}]^{m\times m}$, $\mathbf{r}_0 \in [-p^{2/3+3\delta}, p^{2/3+3\delta}]^{m\times m}$, then $\mathbf{Re} + \mathbf{r}_0 \in [-p^{2/3+4\delta}, p^{2/3+4\delta}]^{m\times m}$.

Correctness of No encodings

For $\mathbf{e}$ such that $\mathbf{e} \xleftarrow{\$} (\mathbb{Z}_p \setminus [-p^{2/3}, p^{2/3}])^m$, we can use the Markov inequality to prove that with probability at least $\left(1 - (\frac{p^{2/3+4\delta}}{p})^m\right) = (1 - p^{-0.13m})$, the

value $\mathbf{Re}$ for small $\mathbf{R}$, lies in $(\mathbb{Z}_p \setminus [-p^{2/3+4\delta}, p^{2/3+4\delta}])^m$. Then, with nearly the same probability, for small $\mathbf{r}_o \in [-p^{2/3+3\delta}, p^{2/3+3\delta}]$, $\mathbf{e}'' = \mathbf{Re} + \mathbf{r}_0$ lies in $(\mathbb{Z}_p \setminus [-p^{2/3+4\delta}, p^{2/3+4\delta}])^m$.

Finally, for random $\mathbf{RA}$ and random $\mathbf{s}'$, we must exclude the probability that $\mathbf{RAs}' + \mathbf{e}''$ is such, that there exists some other pair $\tilde{\mathbf{s}'}, \tilde{\mathbf{e}}$ where $\mathbf{RAs}' + \mathbf{e}'' = \mathbf{RA}\tilde{\mathbf{s}'} + \tilde{\mathbf{e}}$. For a fixed $\mathbf{RA}$, we can eliminate sets $|e| = p^{\delta m}, |s| = p^n and |b| = p^m$. This is at most a fraction $p^{(\delta-1)m+n} \leq 1/p^{(1-\delta)m}$. Then, with probability at least $1 - 1/p$, the encoding of a No instance is decoded correctly.

C.4 Privacy of SRE for DLWE'

In this section, we provide the details in the privacy analysis of our SRE for DLWE'.

Recall that $\mathbf{R}$ is a matrix chosen uniformly at random in $[-p^{2/3}, p^{2/3}]^{m \times m}$. We denote by $\mathbf{R}_i$ the i^{th} row of $\mathbf{R}$. Also, we denote by $\mathcal{R}_{\mathbf{A}}$ the distribution induced by picking $\mathbf{R}$ as above and outputting $\mathbf{RA}$ for a given $\mathbf{A}$. For a fixed row $\mathbf{R}_i \in \mathbb{Z}_p^m$, we denote the distribution of picking $\mathbf{R}_i$ as above and outputting $\mathbf{R}_i\mathbf{A} \in \mathbb{Z}_p^n$ as $\mathcal{R}^i_{\mathbf{A}}$ Similarly, we denote by $\mathcal{R}_{\mathbf{e}}$ the distribution induced by picking $\mathbf{R}$ as above and giving output $\mathbf{Re}$ for a fixed $\mathbf{e}$.

Then, we establish the following lemmas.

Privacy of Yes encodings

We prove in Lemma 3, that a majority of $\mathbf{A} \in \mathbb{Z}_p^{m \times n}$ are such that the distribution $\mathbf{RA}$ induced by multiplying, fixed $\mathbf{A}$, with random $\mathbf{R} \in [-p^{2/3}, p^{2/3}]^{m \times m}$, is close to uniform. Next, in Lemma 4 we show that for Yes instances, the distributions induced by $\mathbf{r}_0 + \mathbf{Re}$ and $\mathbf{r}_0$, where $\mathbf{r}_0 \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^m$, are close. Then, we can combine the two lemmas (via the triangle inequality for statistical distance) to obtain that the output of the simulator is close to a random encoding of an arbitrary Yes instance.

Lemma 3. *Define the sets*

$$\mathsf{Good}_Y \triangleq \left\{ \mathbf{A},\ \mathbf{s},\ \mathbf{e} : \mathbf{A} \in \mathbb{Z}_p^{m \times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in [-p^\delta,\ p^\delta]^m \text{ and } \Delta(\mathcal{R}_{\mathbf{A}}, \mathcal{U}_{m \times n}) \leq p^{-0.16m} \right\}$$

$$\mathsf{Yes}' \triangleq \left\{ \mathbf{A},\ \mathbf{s},\ \mathbf{e} : \mathbf{A} \in \mathbb{Z}_p^{m \times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in [-p^\delta,\ p^\delta]^m \right\}$$

Then, the density of elements of Good_Y *in* Yes' *is overwhelming. Concretely,*

$$\frac{|\mathsf{Good}_Y|}{|\mathsf{Yes}'|} \geq 1 - p^{-0.16m}$$

where $m(n), p(n)$ *are defined appropriately for large enough* n.

Proof. For $\mathbf{A} \in \mathbb{Z}^p_{m \times n}$, define hash function $h_{\mathbf{A}}(R_i)$ from $\mathbb{Z}_p^m$ to $\mathbb{Z}_p^n$ as $h_{\mathbf{A}}(R_i) = R_i\mathbf{A}$. The corresponding family of hash functions $\{h_{\mathbf{A}}(\cdot)\}_{\mathbf{A}}$ is a universal hash function family.

We observe that this family $\{h_{\mathbf{A}}(\cdot)\}_{\mathbf{A}}$ is a universal family of hash functions from $\mathbb{Z}_p^m$ to $\mathbb{Z}_p^n$, on inputs $\mathbf{R_i}$ with min-entropy $k = 2m \log p/3$. The output length of this family is $\ell = n \log p$. Then, the Leftover Hash Lemma gives us that

$$\Delta\big(\mathcal{R}^i_{\mathbf{A}}, \mathcal{U}_n\big) \leq p^{-m/3}.$$

Then, we can use a standard averaging argument to show at least a fraction $(1 - p^{-m/6})$ of $\mathbf{A}$'s satisfy $\Delta(\mathcal{R}^i_{\mathbf{A}}, \mathcal{U}_n) \leq p^{-m/6}$. By a union bound on the m rows of $\mathbf{R}$, for $\mathbf{A}$ chosen above, $\Delta(\mathcal{R}_{\mathbf{A}}, \mathcal{U}_{m\times n}) \leq mp^{-(1/6)m} \leq p^{-0.16m}$ for large enough m, and the lemma follows[7].

Lemma 4. *Let $\mathbf{r}_0 \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]$ be a random variable. Let $\mathbf{r}_1 = \mathbf{r}_0 + \mathbf{a}$, for some fixed $\mathbf{a}$. Let the (uniform) distribution induced by $\mathbf{r}_0$ be denoted by $\mathcal{R}_0$, and the distribution induced by $\mathbf{r}_1$ be denoted by $\mathcal{R}_1$. Then, $\Delta(\mathcal{R}_0, \mathcal{R}_1) = \frac{|\mathbf{a}|}{p^{2/3+3\delta}}$, where $|\cdot|$ denotes the maximum-norm of vector $\mathbf{a}$.*

Proof. By definition of statistical distance,

$$\begin{aligned}\Delta(\mathcal{R}_0, \mathcal{R}_1) &= \sum_{i\in[-p^{2/3+3\delta}, p^{2/3+3\delta}]} \Pr[\mathbf{r}_0 = i] - \Pr[\mathbf{r}_1 = i] \\ &= \frac{|\mathbf{a}|}{p^{2/3+3\delta}}\end{aligned}$$

Thus, the statistical distance between the distribution induced by random choice of $\mathbf{r}_0 \xleftarrow{\$} [-p^{2/3+3\delta}, p^{2/3+3\delta}]$, and $\mathbf{r}_0 + \mathbf{Re}$ for a fixed value of $\mathbf{Re}$ is $\frac{|\mathbf{Re}|}{|\mathbf{r}_0|}$ is at most $p^{-\delta m}$.

Finally, we can directly apply the triangle inequality to argue that the output of the simulator is at most $(p^{-0.05m} + p^{-m/6})$-far from the actual distribution of the encodings of Yes instances.

Privacy of No encodings

In Lemma 5, we prove that a majority of $\mathbf{A}$ and $\mathbf{e}$ are such that the distributions $\mathbf{RA}$ and $\mathbf{Re}$ induced by multiplying, fixed $\mathbf{A}$ (resp. $\mathbf{e}$), with random $\mathbf{R} \in [-p^{2/3}, p^{2/3}]^{m\times m}$, is close to uniform. Next, we show that a majority of vectors $(\mathbf{A}, \mathbf{b})$ chosen uniformly at random, are No instances (and infact, have high error according to the No decoding). Thus we establish that the support of No instances and the support of (uniform) random $(\mathbf{A}, \mathbf{b})$ are close. Finally, we use the fact that the distribution of No instances is uniform over their support, and the fact that their support is close to the uniform distribution, to prove that the simulator's output is close to the encoding of an arbitrary No instance.

[7] Recall that $m = n^2, p = n^{40}, \epsilon = p^{-0.95}$.

Lemma 5. *Let* $\mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^{m\times m}$. *Then, define the sets*

$$\begin{aligned}\mathsf{Good}_N \triangleq \Big\{ &\mathbf{A}, \mathbf{s}, \mathbf{e} : \mathbf{A} \in \mathbb{Z}_p^{m\times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3},\ p^{2/3}])^m) \\ &\wedge\ \Delta\big((\mathcal{R}_{\mathbf{A}}, \mathcal{R}\mathbf{e}),\ (\mathcal{U}_{m\times n}, \mathcal{U}_m)\big) = O(p^{-0.16m}) \Big\} \\ \mathsf{No}' \triangleq \Big\{ &\mathbf{A},\ \mathbf{s},\ \mathbf{e} : \mathbf{A} \in \mathbb{Z}_p^{m\times n},\ \mathbf{s} \in \mathbb{Z}_p^n,\ \mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3},\ p^{2/3}])^m) \Big\} \setminus \mathsf{Yes}'\end{aligned}$$

Then,

$$\frac{|\mathsf{Good}_N|}{|\mathsf{No}'|} = 1 - p^{-0.16m}$$

Proof. Fix a matrix $\mathbf{A}$, such that for $\mathbf{R} \xleftarrow{\$} [-p^{2/3}, p^{2/3}]^{m\times m}$, $\mathbf{RA}$ is $p^{-0.16m}$-close to uniform. From Claim 1, we know that the fraction of such $\mathbf{A} \in \mathbb{Z}_p^{m\times n}$ is at least $(1 - p^{-0.16m})$.

For a fixed $\mathbf{A}$, we first prove that even conditioned on some matrix $\mathbf{A}' = \mathbf{RA}$, the matrix $\mathbf{R}$ still has sufficient entropy so that the hash function $h_e(\mathbf{R}) = \mathbf{Re} \mod p$ is close to uniform with high probability (via LHL).

We claim that for any $\mathbf{v} \in \mathbb{Z}_p^n$ (the range of the hash function), the set of its pre-images under $h_{\mathbf{A}}$ for a fixed $\mathbf{A}$ chosen as above (such that $\mathbf{RA}$ is $p^{-0.16}$-close to uniform), has size at least as large as $p^{2m/3-n-2}$. Moreover, the distribution of the pre-images $\mathbf{R}_i$ is uniform conditioned on $\mathbf{v}$.

We argue this by contradiction. Fix a $\mathbf{v} \in \mathbb{Z}_p^n$ and let R_v denote the set of preimages of $\mathbf{v}$ under $h_{\mathbf{A}}$, i.e. $R_v \triangleq \{\mathbf{r} \in \mathbb{Z}_p^m \mid h_{\mathbf{A}}(\mathbf{r}) = \mathbf{v} \mod p\}$. Assume for contradiction that $|R_v| < p^{2m/3-n-2}$. Now, note that the probability assigned to $\mathbf{v}$ by the uniform distribution is $\frac{2p^{2m/3-n}}{2p^{2m/3}}$ and by $\mathcal{R}^i_{\mathbf{A}}$ is $\frac{2p^{2m/3-n-2}}{2p^{2m/3}}$. Thus

$$\begin{aligned}\Delta(\mathcal{R}^i_{\mathbf{A}};\, \mathcal{U}_n) &\geq |\Pr_{\mathcal{R}^i_{\mathbf{A}}}(\mathbf{v}) - \Pr_{\mathcal{U}_n}(\mathbf{v})| \\ &= |((2p)^{2m/3-n} - (2p)^{2m/3-n-2})/((2p)^{2m/3})| \\ &= \Theta((2p)^{-n})\end{aligned}$$

On the other hand, it follows from Lemma 3 that

$$\Delta(\mathcal{R}^i_{\mathbf{A}};\, \mathcal{U}_n) \leq p^{-0.16m} \ll \Theta((2p)^{-n})$$

which is a contradiction for all $i \in [m]$.

Thus, we have that for a fixed hash family $h_{\mathbf{A}}$,

$$H_\infty(\mathbf{R}_i | \mathbf{AR}_i = \mathbf{v}) \geq (2m/3 - n - \Theta(1)) \log(p)$$

for all $\mathbf{v}$ and i.

Now, note that $\{h_{\mathbf{e}}\}_{\mathbf{e}}$, defined as $h_{\mathbf{e}}(\mathbf{R}) = \mathbf{Re} \mod p$ is a universal family of hash functions. Applying the LHL in a manner similar to Lemma 3, we conclude that

$$\Delta\big(\mathcal{R}_{\mathbf{e}},\, \mathcal{U}_m\big) \leq p^{-0.16m}$$

where $\mathbf{Re}$ is sampled conditioned on fixed $\mathbf{RA}$[8].

Summarizing, for at least a $(1 - p^{-0.16m})$ fraction of $\mathbf{A}$'s, for at least a $(1 - p^{-0.16m})$ fraction of $\mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3}, p^{2/3}])^m$'s, $(\mathcal{R}_{\mathbf{A}}, \mathcal{R}_{\mathbf{e}})$ is $p^{-0.16m}$- close to $(\mathcal{U}_{m\times n}, \mathcal{U}_m)$.

For uniformly chosen $\mathbf{s} \in \mathbb{Z}_p^n$, $(\mathbf{RA}, \mathbf{RAs} + \mathbf{Re})$ is a randomized function of $(\mathbf{RA}, \mathbf{Re})$, and the statistical distance from uniform can only decrease. Therefore the distribution induced by $(\mathbf{RA}, \mathbf{RAs} + \mathbf{Re})$ is $p^{-0.16m}$- close to $(\mathcal{U}_{m\times n}, \mathcal{U}_m)$.

Lemma 6. *For* $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} (\mathcal{U}_{m\times n}, \mathcal{U}_m)$, $\Pr\left[(\mathbf{A}, \mathbf{b}) \in \mathsf{No}\right] \geq 1 - 1/\mathsf{poly}(p)$, *for some polynomial* $\mathsf{poly}(\cdot)$.

Proof. First we argue that if $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} (\mathcal{U}_{m\times n}, \mathcal{U}_m)$, then we may express $\mathbf{b} = \mathbf{As} + \mathbf{e}$ so that $(\mathbf{A}, \mathbf{s}, \mathbf{e}) \in \mathsf{No}'$ defined in Lemma 5, with probability $1 - \Theta(1/p)$. We will define the sets

$$\mathsf{SmErr} \triangleq \{\mathbf{e} \mid \mathbf{e} \in [-p^{\delta}, p^{\delta}]^m\}$$
$$\mathsf{LgErr} \triangleq \{\mathbf{e} \mid \mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3},\ p^{2/3}])^m)$$

Now, note that,

$$\frac{|\mathsf{LgErr}|}{\mathbb{Z}_p^m} = (1 - p^{-1/3})^m \geq 1 - \frac{1}{n}$$

Next, we must eliminate all $(\mathbf{e}, \mathbf{s})$ pairs per $\mathbf{A}$ that are such that $\mathbf{b} = \mathbf{As} + \mathbf{e} = \mathbf{As}' + \mathbf{e}'$ for some $(\mathbf{s}', \mathbf{e}')$ where $\mathbf{e}' \in \mathsf{SmErr}$. We show that the fraction of $\mathbf{b}$ that can be generated using small error is small.

Formally, since $|\ \mathsf{SmErr}\ | = p^{\delta m}$, we have that

$$\frac{|\ \{\mathbf{b} \mid (\mathbf{b} = \mathbf{As} + \mathbf{e}), \mathbf{s} \in \mathbb{Z}_p^n, \mathbf{e} \in \mathsf{SmErr}\}\ |}{|\ \{\mathbf{b} \mid \mathbf{b} \in \mathbb{Z}_p^m\}\ |} = \frac{p^{\delta m + n}}{p^m} \leq 1/n$$

Hence, if $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} (\mathcal{U}_{m\times n}, \mathcal{U}_m)$, then we may express $\mathbf{b} = \mathbf{As} + \mathbf{e}$ so that $(\mathbf{A}, \mathbf{s}, \mathbf{e}) \in \mathsf{No}'$ with probability $1 - \Theta(1/p)$.

Next, by Lemma 5,

$$\frac{|\mathsf{Good}_N|}{|\mathsf{No}'|} \geq 1 - p^{-0.16m}$$

Since by definition, No comprises of random $(\mathbf{A}, \mathbf{b})$ such that $\mathbf{b} = \mathbf{As} + \mathbf{e}$ where $(\mathbf{A}, \mathbf{s}, \mathbf{e}) \in \mathsf{Good}_N$, we get the claim.

Moreover, by definition of the promise problem, the distribution $\mathcal{R}_{\mathbf{A}}$ induced by random choice of $\mathcal{R}$, and the distribution $\mathcal{R}_{\mathbf{e}}$ induced by random choice of $\mathcal{R}$ are close to uniform, over the support of No instances. By Lemma 6, these supports are $1/\mathsf{poly}(p)$-close. These arguments together prove that the output of the simulator is $1/\mathsf{poly}(p)$- close to the encoding of an arbitrary No instance.

[8] Actually, the LHL is applied to $\mathbf{e} \in \mathbb{Z}_p^m$, but $(1 - p^{-0.33})$ fraction of this set contains $|\mathbf{e}| > p^{2/3}$, and at least a $(1 - p^{-0.16})$ fraction of this set is such that $\mathcal{R}_{\mathbf{e}}$ is close to uniform. Therefore, for at least a $(1 - p^{-0.33} - p^{-0.16})$ fraction of $\mathbf{e}$ where $\mathbf{e} \in (\mathbb{Z}_p \setminus [-p^{2/3}, p^{2/3}]^m)$, $\mathcal{R}_{\mathbf{e}}$ is close to uniform

Extending to DLWE''.

Recall that DLWE'' was identical to DLWE' except setting $p = 2^n$. The hardness of DLWE'' follows from the (less standard) sub-exponential hardness of DLWE. Then, the same analysis as above, extends to prove that DLWE'' is in $(\mathsf{negl}, \mathsf{negl})-\mathsf{SRE}$. Specifically, the correctness and privacy are again $O(p^{-m})$, which is now, $O(1/\exp(n))$ (instead of $O(1/n)$ for DLWE').

D Oracle Separation between SRE and SZK

Here, we prove Theorem 2.

For this proof, we denote by P a probabilistic unbounded prover interacting with a PPT verifier, denoted by V. An oracle-SRE is an SRE machine enc which may query some oracle A. This is denoted by enc^A. We give an oracle A such that $\mathsf{SZK}^A[2] \neq \mathsf{SRE}^A$ with $\varepsilon, \delta \leq 2^{-|x|}$. We diagonalize over all oracle SRE machines while ensuring that the language constructed via diagonalization admits a two-round SZK protocol.

D.1 Mapping Oracles to Languages

In this part, we define a map from an arbitrary set of strings, $A \in \Sigma^*$, to a set of binary strings, L_A. Looking ahead, A will correspond to the strings in the oracle set. This map is defined as follows:

1. Let a_n denote the characteristic vector of $A \cap \Sigma^n$. That is, a_n is a 2^n-bit string such that for all n-bit strings $i \in \Sigma^n$, $a_n[i] = 1$ if and only if $i \in A$, where $a_n[i]$ denotes the i^{th} bit of a_n.
2. Divide the first $\lfloor 2^n/3n \rfloor$ bits of a_n into segments of length $3n$. We will ignore the remaining bits of a_n.
3. Of these, let s_j denote the j^{th} segment, that is, the bits $1 + 3n(j-1)$ to $3nj$ in a_n. For each string v, of length $3n$, define R_v as the set of segments which have value v: $R = \{i | s_i = v\}$.
   - a_n is *unique* whenever $|R_v| \leq 1$ for all strings v of length $3n$. That is, no two segments share the same value. Note that there are 2^{3n} possible different values of length $3n$, to be allotted to $\lfloor 2^n/9n \rfloor$ segments.
   - a_n is *redundant* whenever there are exactly $\lfloor \sqrt{2^n/3n} \rfloor$ strings v with $\lfloor \sqrt{2^n/3n} \rfloor \leq |R_v| \leq \lfloor \sqrt{2^n/3n} \rfloor + 2$ and $|R_v| = 0$ for the remaining v.
   - a_n is *completely redundant* if it is 0^{2^n}.
4. Last, define the binary language L_A as follows:
   $1^n \in L_A$ if and only if a_n is unique; and $x \notin L_A$ if $x \neq 1^n$ for some n.

D.2 Diagonalization Over SRE

Recall from Lemma 1 that an (ϵ, δ)-SRE for a language is an encoding algorithm enc that satisfies the ε-privacy and δ-accuracy requirements.

Our first step is to enumerate all oracle-SRE encoders. That is, we lexicographically enumerate all PPT TMs enc such that they form an ε, δ-SRE for some language L. Call this enumeration $\mathsf{enc}_1^A, \mathsf{enc}_2^A, \ldots$. Next, we set A in rounds such that at round i, enc_i^A is not an (ε, δ)-SRE for L_A.

Without loss of generality, assume that for sufficiently large n, enc_i^A runs in time at most n^i on inputs of length n. We will determine A in rounds by putting strings in and out of the oracle set. A string that has not yet been put in or out of A will be called undetermined.

The general idea of the construction is to ensure that at round i, either $\Delta(\mathsf{enc}(1^{n_i}), \mathsf{enc}(0^{n_i})) > 2\epsilon$ or $\Delta(\mathsf{enc}(1^{n_i}), \mathsf{enc}(0^{n_i})) < (1 - 2\delta)$. In both cases, enc_i^A is not an (ε, δ)-SRE for L_A.

Let m_i for $i = 1, 2, \ldots$ be defined by $\mathsf{min}_{m \in \mathbb{N}}(9m^{2i+1/2}2^{2-m/2} \leq 1/2)$ (this setting will be useful later). Let $n_1, n_2, \ldots$ be a sequence of integers defined by $n_1 = \mathsf{max}\{20, m_1\}$ and $n_i = \mathsf{max}(n_{i-1}^{i-1} + 1, m_i)$ for $i = 2, 3, \ldots$. For all x not of length n_i for some i, set $x \notin A$. We set strings of length n_i in rounds.

Round i: Run enc_i^A on inputs 1^{n_i} and 0^{n_i}. Note that since enc_i can run for time at most n_i^i it cannot ask about strings of length n_{i+1} or greater. So when enc_i asks A about a string y we have 3 cases: $|y| = n_j$ for $j < i, |y| \neq n_j$ for $1 \leq j \leq i + 1$ and $|y_i| = n_i$. In the first case the answer has already been determined in a previous round and in the second case the answer was determined ahead of time. Therefore, $\Delta[(\mathsf{enc}_i^A(1^{n_i})), (\mathsf{enc}_i^A(0^{n_i}))]$ is determined only by answers to queries of the third type, that is, by a_{n_i}.

Denote by $s(a^{n_i})$, the statistical distance $\Delta[(\mathsf{enc}_i^A(1^{n_i})), (\mathsf{enc}_i^A(0^{n_i}))]$ over the randomness of enc_i^A, for fixed characteristic string a^{n_i}. Note that $0^{n_i} \notin L_A$ for all n_i. Consider the following two cases.

- If there exists a redundant a^{n_i} such that $s(a^{n_i}) \geq 2^{-n_i}$, then set A according to this a^{n_i}.
- If no redundant a^{n_i} exists with $s(a^{n_i}) \geq 2^{-n_i}$, then find a unique a^{n_i} with $s(a^{n_i}) \leq 1 - 2^{-n_i}$, and set A according to this a^{n_i}. Lemma 7 shows that such a unique a^{n_i} exists with non-negligible probability. Intuitively, with polynomially many oracle queries, enc_i^A cannot distinguish redundant a^{n_i} from unique a^{n_i}. Refer to Appendix D for the full proof.

It is easy to see that if one of the two cases is always possible, then there exists no (ε, δ)-SRE for L_A, according to Definition 1 with $\varepsilon, \delta = 2^{-n_i}$.[9] The following Lemma establishes this.

Lemma 7. *If $s(a^{n_i}) \leq 2^{-n_i}$ for all redundant a^{n_i}, then fraction of unique a^{n_i} with $s(a^{n_i}) < 1 - 2^{-n_i}$ is at least 2^{-n_i}.*

Refer the next section for the full proof of Lemma 7. Then, we obtain the following main lemma.

Lemma 8. $L_A \notin \mathsf{SRE}^A$.

D.3 Proof of Lemma 7

In this section, we give a formal proof of Lemma 7. We prove a more general statement, for any two fixed negligible functions $d_1(\cdot), d_2(\cdot)$. Lemma 7 is a special case of this proof for $d_1(n) = d_2(n) = 2^{-n}$. Consider the contrapositive statement of the lemma, that is, suppose there exist negligible polynomials $d_1(\cdot)$ and $d_2(\cdot)$ such that $s(a^{n_i}) \leq d_1(n_i)$ for all redundant a^{n_i}, and fraction of unique a^{n_i} with $s(a^{n_i}) < 1 - d_2(n_i)$ is less than $1/\mathsf{poly}(n_i)$ for all polynomials $\mathsf{poly}(\cdot)$.

[9] Note that this can, in general, be proved for ϵ, δ set to any fixed negligible function $\mathsf{negl}(n_i)$. Our proof handles this general case.

We prove the lemma by deriving a contradiction for the above statement, in two parts. In the first claim, we show that any PPT encryptor that runs in time at most n_i and is executed 2ℓ (where ℓ is a fixed polynomial of n_i) times on a fixed input set, fails to generate very biased outputs when the oracle A is set according to a^{n_i} which is unique versus a^{n_i} which is redundant. Next, we claim that if the statistical distance between $\mathsf{enc}(0^{n_i}), \mathsf{enc}(1^{n_i})$ is very low for all redundant a^{n_i}, and the statistical distance between $\mathsf{enc}(0^{n_i}), \mathsf{enc}(1^{n_i})$ is very high for all unique a^{n_i}, then the outputs are indeed very biased. We denote the set of redundant a^{n_i} by R and the set of unique a^{n_i} by U.

Claim. Consider any PPT oracle machine enc_i^A that runs in time at most n_i, and is executed 2ℓ times alternately on inputs 0^{n_i} and 1^{n_i} and uniform randomness, and produces an output distribution. The output produced by enc_i^A is taken as input by an unbounded distinguisher $\mathcal{D}$ (which does not have oracle access to A), which outputs a single bit. Then for any such distinguisher $\mathcal{D}$,

$$\Pr_{a^{n_i} \in R}[\mathcal{D} = 1] \leq \Pr_{a^{n_i} \in U}[\mathcal{D} = 1] \leq \Pr_{a^{n_i} \in R}[\mathcal{D} = 1]/(1 - \ell^2 n_i^{2i+1/2} 2^{2-n_i/2})$$

Proof. Fix the randomness $r_1, r_2, \ldots r_{2\ell}$ of enc_i^A. Once the randomness of enc_i^A is fixed (for inputs fixed alternatingly to 0^{n_i} or 1^{n_i}), it becomes deterministic and depends only on a^{n_i}. Consider one set of 2ℓ computations of enc, such that on each computation it examines $k_1, k_2, \ldots k_{2\ell}$ segments respectively, and the distinguisher outputs 1. Let $k = k_1 + k_2, \ldots + k_{2\ell}$.

Let m^R be the number of redundant a^{n_i} on which enc_i^A would produce this computation for this input set (that is, which have the same values at those k segments), and let m^U be the number of unique a^{n_i} on which enc_i^A would produce this computation for this input set.

Assuming without loss of generality that all the $2\ell.k$ segments are unique, we have that

$$\frac{m^U}{|U|} = \Pi_{i=0}^{k-1}(2^{3n} - i)^{-1}, \text{ and}$$

$$\frac{m^R}{|R|} = \Pi_{i=0}^{l_1-1}(2^{3n} - i)^{-1} \cdot \Pr_{a^{n_i} \in R}[k \text{ specified segments have unique values}]$$

Using $k \leq 2\ell n^i$ we get that the last probability[10], is at least $(1-\ell^2 n_i^{2i+1/2} 2^{2-n_i/2})$.

Summing over all possible sets of computations where $\mathcal{D}$ outputs 1, then over the randomness r:

$$\Pr_{a^{n_i} \in R}[\mathcal{D} = 1] \leq \Pr_{a^{n_i} \in U}[\mathcal{D} = 1] \leq \frac{\Pr_{a^{n_i} \in R}[\mathcal{D} = 1]}{(1 - \ell^2 n_i^{2i+1/2} 2^{2-n_i/2})}$$

Claim. If $s(a^{n_i}) \leq d_1(n_i)$ for all $a^{n_i} \in R$, and $s(a^{n_i}) \geq 1 - d_2(n_i)$ for a fraction $> 1 - 1/\mathsf{poly}(n_i)$ of $a^{n_i} \in U$ for all polynomials $\mathsf{poly}(\cdot)$, then there exists an (unbounded) distinguisher $\mathcal{D}$ which takes as input the pairs $(\mathsf{enc}_i^A(0), \mathsf{enc}_i^A(1))_i$

[10] Refer [1] for details.

for $i \in [\ell]$ over uniform randomness of the encoders and randomly chosen a^{n_i}, such that $|\Pr_{a^{n_i} \in R}[\mathcal{D} = 1] - \Pr_{a^{n_i} \in U}[\mathcal{D} = 1]| \geq 0.5 - \mathsf{negl}(\ell)$ for a negligible function $\mathsf{negl}(\cdot)$. Note that the oracle A and hence a^{n_i} is fixed over all inputs to the distinguisher, also note that the distinguisher does not require oracle access to $\mathcal{A}$.

Proof. On input three samples $(\mathsf{enc}_i^A(0), \mathsf{enc}_i^A(1))_1, (\mathsf{enc}_i^A(0), \mathsf{enc}_i^A(1))_2, (\mathsf{enc}_i^A(0), \mathsf{enc}_i^A(1))_3$ where all three correspond to the same (unique or redundant) A, consider a distinguisher $\mathcal{D}'$ which fixes any subset S of the union of the supports of $(\mathsf{enc}_i^A(0), \mathsf{enc}_i^A(1))$ over all unique and redundant a^{n_i}. The distinguisher outputs 0 if for all three pairs, one of $\mathsf{enc}_i^A(0)$ and $\mathsf{enc}_i^A(1)$ is in the subset S, and the other is outside the subset S. Otherwise, the distinguisher $\mathcal{D}'$ outputs 1.

Now, (for redundant A - low statistical distance), the probability of a single instance having both $\mathsf{enc}_i^A(0)$ and $\mathsf{enc}_i^A(1)$ in or outside the subset S is $\Pr_{a^{n_i} \in R}[\mathcal{D}' = 1] \geq 2\mathsf{min}_p(p^2 + (1-p)^2) \pm O(d_1) \geq 0.25 \pm O(d_1)$.

Therefore, the probability of this event happening in at least one out of 3 independent samples is, at least $1 - (0.75^3 \pm O(d_1) \geq 0.5 + O(d_1)$.

However, (for unique A), $\Pr_{a^{n_i} \in U}[\mathcal{D}' = 1] \leq O(d_2) = \mathsf{negl}(n_i)$.

But when $\ell = 3$, by Claim D.3 and our setting of n_i (refer Section 4), we have that for all distinguishers $\mathcal{D}$,

$$\Pr_{a^{n_i} \in R}[\mathcal{D} = 1] \leq \Pr_{a^{n_i} \in U}[\mathcal{D} = 1] \leq 2 \Pr_{a^{n_i} \in R}[\mathcal{D} = 1]$$

Whereas, by Claim D.3, we have a distinguisher $\mathcal{D}$ such that: $|\Pr_{a^{n_i} \in R}[\mathcal{D} = 1] - \Pr_{a^{n_i} \in U}[\mathcal{D} = 1]| \geq 0.5 - \mathsf{negl}(n)$. This gives a contradiction.

D.4 Statistical Zero Knowledge

It remains to show that L_A admits an oracle-SZK proof. Let $L_A^{(1)}$ denote the language L_A restricted to strings of the form 1^n for all n. Then following lemma can be imported from [1].

Imported Lemma 2. *[1]* $L_A^{(1)} \in \mathsf{SZK}[2]$.

Note that all inputs x such that $x \neq 1^n$ for some n, are trivially not in L_A. On inputs of the form 1^n for some n, we can invoke the interactive proof of [1] via Imported Lemma 2. With overwhelming probability over the choice of a^n, this proof is shown by [1] to have the statistical zero knowledge property[11].

Thus, we obtain the following main lemma.

Lemma 9. $L_A \in \mathsf{SZK}^A[2]$.

This completes the proof of Theorem 2.

[11] A slight modification to the diagonalization described above [1] makes it possible to always pick a unique a^{n_i} which satisfies the conditions of the diagonalization *and* the extra SZK condition. This ensures that $L_A \in$ oracle-SZK always, while also diagonalizing over all oracle-SRE encoders.

E ASM-based promise problem

We consider (a family of) promise problems, which is a special case of the Abelian Subgroup Membership problem. We devise an efficient SRE for this class of problems, and specify instances of it that are likely to be outside of $\mathsf{P/poly}$.

E.1 Notation and preliminaries.

We denote by $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_t]$ the matrix formed by taking $\mathbf{x}_i$ as rows and by $[\mathbf{x}_1 \mid \mathbf{x}_2 \mid \ldots \mid \mathbf{x}_t]$ the matrix formed by taking $\mathbf{x}_i$ as columns. $\mathbf{X}_i$ denotes the i'th row in a matrix $\mathbf{X}$.

Next, we formulate a lemma about distributions that will be useful in our ASM construction.

Lemma 10. *Let A, B denote a pair of distributions with finite supports. Let $C = pA + (1-p)B$ (that is, a distribution that samples A with probability $p > 0$, and samples B with probability $1-p$). Then, for every integer $k_1 > 0$ and $\epsilon \in (0, 1]$, there exists $k_2(k_1, p, \varepsilon) = poly(k_1, 1/p, 1/\varepsilon)$, monotonically increasing in $k_1, 1/p, 1/\varepsilon$, such that any pair of distributions D_1, D_2 as described below satisfies $SD(D_1, D_2) \leq \varepsilon$. Let $k_2' \geq k_2$.*

- *D_1 : Generate $k_2' + k_1$ iid. samples from C. Output a random permutation of the samples.*
- *D_2 : Generate k_1 iid. samples $s_1, \ldots, s_{k_1}$ from A, and k_2' iid. samples $s_{k_1+1}, \ldots, s_{k_1+k_2'}$ from C. Output a random permutation of the s_i's.*

Proof. The high level intuition is that replacing a "small" number of samples from A by a large number of samples from C when C contains some sufficiently large "component" of A does not change the distribution by much. The total number of samples $k_2 + k_1$ (for a fixed k_1) would need to grow the smaller p is (the less likely it is to "run into A" in C - the worst case is when A, B have disjoint supports).

Set some k_2' to be determined later, and let Bin denote the binomial distribution. We have $SD(D_1, D_2) \leq SD(Bin(p, k_2' + k_1), Bin(p, k_2') + k_1)$. To see this, think of D_1', D_2' obtained from the same processes as above, but sampling from A, B is replaced by the constants $0, 1$. Now, apply the randomized function of replacing 0 by A and 1 by B in both, to obtain D_1, D_2 which can only reduce SD. Thus, it suffices to bound $SD(D_1', D_2')$. The random permutation of the samples insures that the SD between the distribution can be calculated from a "succinct" representation of the counts of 1's in the sample.

Observation 1. *For $t < k_1$, $Pr(A = t) = 0 < Pr(C = t)$. For $t \in [k_1, k_1 + k_2']$, we have $Pr(C = t)/Pr(A = t) = p^{k_1} \prod_{i=1}^{k_1} (k_2' + i)/(t - k_1 + i)$.*

We conclude that $Pr(C = t)/Pr(A = t)$ is monotone decreasing where it is defined, and thus the graphs $Pr(C)$ and $Pr(A)$ over $[k_1, k_1 + k_2']$ (viewed as

functions of t over $\mathbb{R}$) have a single (not necessarily integral) point t' of intersection. Thus, we have

$$SD(A,C) = \sum_{i \le t'} (Pr(C=i) - Pr(A=i)) + \sum_{i > t'} (Pr(A=i) - Pr(C=i)) = \tag{1}$$

$$(\sum_{i \le t'} Pr(C=i) - \sum_{i > t'} Pr(C=i)) + (\sum_{i > t'} Pr(A=i) - \sum_{i \le t'} Pr(A=i)) \tag{2}$$

We prove that for large enough k_2', t' is close to the mean of both distributions, both summands in Equation 2 are small. It is easy to see that

$$t \in [p(k_1 + k_2'), pk_2' + k_1 - 1] \tag{3}$$

From [10], for the binomial distribution $X = Bin(n,p)$, $Pr(X \le np), Pr(X \ge np) = 1/2 \pm \Theta((np)^{-0.5})$. That is, the distribution $Bin(n,p)$ is almost symmetric about its mean np. Here Θ hides a global constant, and $\Theta((np)^{-0.5})$ is the probability of obtaining the value T with maximal probability. It is known that T equals np rounded either up or down, or both. That is, we have.

Claim. The maximum weight of a single value t (may occur for 1 or two values around the mean) in $X = Bin(n,p)$ occurs with probability $\Theta((np)^{-0.5})$.

Proof. To prove the claim, observe that by Chernoff bounds $Pr(|X - np| \ge \gamma) \le e^{-\gamma^2/3np}$. Taking $\gamma = (np)^{0.5}$, we get a bound of 0.72. Thus, a T with maximal probability has weight at least $0.14(np)^{-0.5}$.

Thus, each of the two summands in Equation 2 is bounded by

$$\Theta(k_1(pk_2')^{-0.5}). \tag{4}$$

(the probability for each t in both distributions is bounded by $0.14(pk_2')^{-0.5}, 0.14(p(k_2' + k_1)^{-0.5})$ respectively). Let c denote the constant implicit in Equation 4. By Equation 3, there are at most $2k_1'$ elements of the form $Pr(C=t)$ ($Pr(A=t)$) over the two summands. Thus, by Claim E.1, taking k_2' so that $4ck_1p(k_2' + k_1)^{-0.5} \le \varepsilon$ suffices yields $SD(D_1', D_2') \le \varepsilon$. Letting $k_2' \ge k_1/3$ and rearranging we get $k_2' \ge \frac{64c^2k_1^2}{\epsilon^2 p}$.

E.2 Problem Specification.

The problem family $(\mathsf{ASM}_n^{G,H})_n$ is specified by G where $(G, \cdot)$ is a finite abelian group, and a subgroup H of G, fixed for every length parameter n. We have $G/H \cong \mathbb{Z}_q^t$ for a prime q. The input is an element $x \in G$ (that is, G_n, H_n is a sequence of groups specifying one instance of the problem, we often omit the subscript n when clear from the context). We assume:

1. The group element x is specified by an encoding (a one to one mapping, not necessarily efficient) $E : G_n \to \{0,1\}^{p(n)}$, for some monotonically increasing $n \leq p(n) \in \mathsf{poly}(n)$. There exists a PPT algorithm $\mathsf{Mul}(\cdot)$ to that multiplies encoded elements of G (giving in a valid encoding of the result).
2. There exists a PPT algorithm Gen that takes 1^n as input and outputs a generator set $(h_1, \ldots, h_l)$ (for arbitrary l defined according to H), for H. If $t > 1$, also compute a $g_1, \ldots, g_t \in G_n$, such that $I(g_1), \ldots, I(g_t)$ are independent vectors in $\mathbb{Z}_q^t$,[12] where I is some isomorphism from G/H to $\mathbb{Z}_q^t$, and the g_i's are viewed as coset representatives.
3. x is promised to be in the Image of E.

$$\mathsf{Yes}_n = \{x \ : \ x \in \mathsf{E}(H)\}$$
$$\mathsf{No}_n = \{x \ : \ x \in \mathsf{E}(G)/\mathsf{E}(H)\}$$

$\mathsf{Yes} = \cup_n \mathsf{Yes}_n, \mathsf{No} = \cup_n \mathsf{No}_n$.

Remark 1. Requirements 2,3 in the above definition are needed for our SRE to work. Settling for non-uniform SRE, these requirements can be dropped, as each set of generators is of size $\leq n$, and can be given as advice to the SRE encoder.

We stress that $\mathsf{ASM}^{G,H}$ is in fact a framework for defining promise problems, each possessing properties as above. Some of these problems are easy and some are hard (under certain computational assumptions). Our SRE will work for any instance of $\mathsf{ASM}^{G,H}$, while hardness for $\mathsf{P/poly}$ is proved for a specific instance (a language) in the sequel.

E.3 SRE Construction

In this section we provide a non-uniform SRE for the Abelian Subgroup Membership problem.

Construction 1 (SRE_ε). *Fix some problem (sequence) $\mathsf{ASM}_n^{G,H}$ (with associated PPT algorithms E, Mul). In all the subsequent discussion, we consider a specific (sufficiently large) n. For $t > 1$, let I denote some isomorphism between G/H and $\mathbb{Z}_q^t$ for which Gen outputs $g_1, \ldots, g_t$ for which $I(g_1), \ldots, I(g_t)$ are independent vectors in $\mathbb{Z}_q^t$.*

Our SRE will make use of the following procedures:

- $\mathsf{samp}(y_1, \ldots y_l)$*: A PPT taking a generator set $y_1, \ldots, y_l$ of a subgroup Y of G, and outputs a random element of Y. Here and elsewhere we slightly abuse notation by writing elements of G, and products over G, meaning that these are encodings of elements of G. $x \cdot y$ is used as a shorthand for $\mathsf{Mul}(x, y)$.* [13]

[12] Here the g_i's are viewed as elements of G/H. At the cost of slightly modifying the construction, and complicating the analysis, the independence restriction may be replaced with merely requiring that each g_i is sampled at random from G (G/H).

[13] This can be done efficiently by returning $\prod_{j \in f} (y_j)^{i_j}$, where the i_j's are i.i.d uniform over $[|G|]$ (or $[|Y|]$, if known.

- $\mathsf{samp\text{-}ind}(l, g_1, \ldots, g_t)$: *A PPT taking an integer* l*, and* g_i*'s where* $I(g_1), \ldots, I(g_t) \in \mathbb{Z}_q^t$ *are independent, and outputs* l *elements* $s_1, \ldots, s_l$ *of* G*, so that the* $I(s_i)$*'s are random independent vectors in* $\mathbb{Z}_q^t$*.*[14]

$\mathsf{enc}_\varepsilon(x)$: *We begin by calling* $\mathsf{Gen}(1^n)$ *(where* n *is efficiently extracted from* x*). Let* $h_1, \ldots, h_l$ *denote the generators of* H*, and* $g_1, \ldots, g_f$ *the generators of* G *returned. If* $f = 0$*, set* $t = 1$*, otherwise, set* $t = f$*.*

We first address the case $t = 1$ *which is particularly simple. Here* $\mathsf{enc}_\varepsilon(x)$ *simply outputs* $x^i \cdot \mathsf{samp}(h_1, \ldots, h_l)$*, where* $i \xleftarrow{\$} [q-1]$*. This is easily seen to yield a perfectly correct and private* SRE *for the proble (or a negligible* ϵ *assuming we can only sample bits, and* $p-1$ *is not a power of 2). See analysis of a concrete instantiation of this case in Section E.4 for details. From now on we focus on the case* $t > 1$*.* SRE_ε *is specified as follows:*

We define the procedure $\mathsf{SampMat}(x)$ *as follows.*

1. *Sample* $(x_1, ..., x_{t-1}) \xleftarrow{\$} \mathsf{samp\text{-}ind}(g_1, \ldots, g_t)$*. Set* $x_t = x$*, and define* $\mathbf{x} = (x_1, \ldots, x_t)$*. Denote* $\mathbf{X} = I(\mathbf{x}) \triangleq [I(x_1); \ldots; I(x_t)] \in \mathbb{Z}_q^{t \times t}$ *(no need to actually compute* $\mathbf{X}$*, which may be inefficient).*
2. *Sample* $l_1, ..., l_t \in \mathbb{Z}_q^t$ *and denote* $\mathbf{R} = [l_1; ...; l_t] \in \mathbb{Z}_q^{t \times t}$*.*
   For $i \in [t]$*, let* $x'_i = \prod_{j \in [t]} (x_j)^{l_{i,j}} \cdot \mathsf{samp}(H)$*.*
3. *Output* $\mathbf{x}' = (x'_1, ..., x'_t)$*. Denote* $\mathbf{X}' = I(\mathbf{x}')$*.*

Now, let p_q *denote the fraction of invertible* $t \times t$ *matrices in* $\mathbb{Z}_q^{t \times t}$ *(as* t *goes to infinity). If* $q > 4 \cdot 2^n$*, set* $k_1 = 1, k_2 = 0$ *(this will result in a single call to* $\mathsf{SampMat}(x)$*). Otherwise, set* $k_1 = \lceil \log_{(1-p_q^2)}(2^{-n}) \rceil, k_2 = k_2(k_1, 0.14/q, \varepsilon)$ *as specified in Lemma 10.*

1. *For* w *in* $[k_1]$*, let* $\mathbf{x}'_w \xleftarrow{\$} \mathsf{SampMat}(x)$ *(with fresh randomness every time).*
2. *For* $w \in [k_2]$*, let* $\mathbf{x}'_{k_1+w} \xleftarrow{\$} \mathsf{SampMat}(1)$ *(again, with fresh randomness).*
3. *Output a uniformly picked permutation* $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_{k_1+k_2})$ *of* $\mathbf{x}'_1, \ldots, \mathbf{x}'_{k_1+k_2}$*.*

$\mathsf{dec}((\mathbf{y}_1, \ldots, \mathbf{y}_{k_1+k_2}))$*: Check whether some* $I(\mathbf{y}_i)$ *is of full rank (as an element of* $\mathbb{Z}_q^{t \times t}$*). If yes, reject, otherwise accept.*

Theorem 5. *For every promise problem (sequence)* $\mathsf{ASM}_n^{G,H}$ *(with associated algorithms* E, Mul*), and* $\varepsilon(n) = 1/\mathsf{poly}(n)$*,* SRE_ε *from Construction 1 is a* $(\varepsilon, 2^{-n})$-SRE *for the problem.*

In a search of instances of ASM separating SRE from P/poly, note that $\mathsf{ASM}_n^{G,H}$ is a candidate only if $\gcd(|G|/|H|, |H|) > 1$ for infinitely many n's, since otherwise $(x^{|H|} =^? 1)$ tests for membership in H.

[14] This can be done efficiently by sampling an non-singular matrix $[m_1; \ldots; m_l] \in \mathbb{Z}_q^{l \times t}$, and output $\prod_{i=1}^l s_i^{m_{1,i}}, \ldots, \prod_{i=1}^l s_i^{m_{l,i}}$ (this procedure introduces a negligibly small error, due to the need to sample a non-singular matrix). This special procedure is only needed where q is small relatively to n. This is so, as making l independent calls to $\mathsf{samp}(g_1, \ldots, g_t)$, where the $I(g_i)$'s generate G/H already yields what we need w.p. $\Omega(1/q)$.

Proof. Before delving into proof details, let us provide a short overview of the ideas behind the construction for $t > 1$. The first idea is that given a group element $g \in G$, and a random element in $h \in H$, $g \cdot h$ reveals precisely the coset of x. We multiply by a random element of H where necessary. This idea has previously been used in the literature [2] for QR.

As to hiding the coset of x (except for whether it equals H), we start with the following basic construction (corresponds to $\mathsf{SampMat(x)}$ in our construction). One case is that q is very large. In this case, we observe that sampling random $t-1$ elements of G, $(x_1, \ldots, x_{t-1})$, and adding $x_t = x$ to the set, results in a basis $I(x_1), \ldots, I(x_t)$ of $\mathbb{Z}_q^{t \times t}$. We can view these vectors as a matrix $\mathbf{X} = [I(x_1); \ldots; I(x_t)]$ over $\mathbb{Z}_q^{t \times t}$ (although I is implicit, and enc_ε never actually evaluates it). We can effectively multiply this matrix by a vector $r \in \mathbb{Z}_q^t$ by computing $\prod_i x_i^{r_i}$, where r_i is viewed as an element of $\mathbb{Z}$. Similarly, we can multiply by a matrix R, viewed as a sequence of column vectors. Picking R to be a random matrix in $\mathbb{Z}_q^{t \times t}$, we obtain that $\mathbf{RX}$ is an SRE for large q. To see this, we observe:

- If $x \notin H$, $\mathbf{X}$ is non-singular and thus $\mathbf{RX}$ is a random matrix with probability $\geq 1 - 2/q$. Thus, a uniform matrix output by a simulator given $b = 0$ results in an error of at most $2/q$. As, again, a random matrix is non-singular with probability $1 - 2/q$, the decoder detects a full rank matrix $\mathbf{RX}$, and rejects with probability $\geq 1 - 4/q$.
- For $x \in H$, $I(x)$ is singular, so a matrix as above is distributed in the same way for all $x \in H$, so we get perfect privacy. Perfect correctness follows from the fact that $\mathbf{X}$ (and thus $\mathbf{RX}$) has rank $(t-1) < t$.

To conclude, we get a $(O(1/q), O(1/q))$-SRE using the above construction. However, the parameters are quite bad if q is small (worst for $q = 2$). In this case, we augment the above construction using Lemma 10. More precisely, we repeat $SampMat$ some $k_1 = O(n)$ times to amplify correctness correctness to, say, 2^{-n}. Now, to improve privacy, we "mix in" sufficiently many - k_2 instances of $SampMat(1)$, and randomly permuting all instances.

Simulator. The simulator Sim_ε takes as input the size parameter 1^n and a single bit b, where $b = 0$ denotes $x \notin H$, $b = 1$ denotes $x \in H$. It proceeds by.

- Input $b = 0$.
  1. If $q > 4 \cdot 2^n$, set $k_1 = 1, v = 1, k_2 = 0$. Otherwise, Let $p_S = \prod_{i=1}^{t-1} 1 - q^{i-t}$. Sample $v \xleftarrow{\$} Bin(k_1, p_S)$, $k_1 = \lceil \log_{1-p_q^2}(2^{-n}) \rceil$, $k_2 = k_2(k_1, 0.14/q, \varepsilon)$. For every $i \in [v]$.
     (a) Generate a pair $\mathbf{X}, \mathbf{R}$, where $\mathbf{X} = [I(x_1); \ldots; I(x_t)]$ is (isomorphic via I to) a random invertible matrix in $\mathbb{Z}_q^{t \times t}$, and $\mathbf{R}$ is a random matrix in $\mathbb{Z}_q^{t \times t}$.[15]
     (b) For $i \in [t]$, let $x'_i = \prod_{j \in [t]} (x_j)^{\mathbf{R}_{i,j}} \cdot \mathsf{samp}(H)$.

[15] This computation is generally inefficient, done by first picking a matrix $\mathbf{X} \in \mathbb{Z}_q^{t \times t}$, and then inverting it via I, which may not be efficiently computable, to obtain $\mathbf{x}$.

(c) Set $\mathbf{x}'_i = [x'_1; \ldots; x'_t]$.

2. For all $i \in [k_1 + k_2 - v]$, sample $\mathbf{x}'_{v+i} \xleftarrow{\$} Mat(1)$ uniformly at random.
3. Output a random permutation of the $\mathbf{x}'_i$'s.

- Input $b = 1$. Output $\mathsf{enc}_\varepsilon(1)$.

Analysis of simulation and correctness.

The case $x \in H$. In this case, the simulator simply runs $\mathsf{enc}_{\mathsf{SRE}}(1)$. To observe perfect privacy for this case, we note that the output of $\mathsf{enc}_{\mathsf{SRE}}(x)$ depends only on the coset of G/H falls in, rather then the concrete coset member. This is due to multiplying by $\mathsf{samp}(H)$ Correctness in this case is perfect, as $I(x) = \overline{0}$ for all isomorphisms, and thus $rank(y_i) < t$ for all i with probability 1 over the choices of $\mathsf{enc}_\varepsilon(x)$ and dec rejects.

The case $x \notin H$. The case $x \notin H$ is more complicated. As observed already, we may focus only on the coset of G/H that x belongs to, as the concrete member of the coset that x constitutes is completely "erased" by $\mathsf{enc}_\varepsilon(x)$. On a high level, there are two cases. In a simpler case, q is large ($q > 4/\varepsilon$). Then, $\mathbf{X}'$ generated in $\mathsf{SampMat}(x)$ is a random non-singular matrix $\mathbb{Z}_q^{t \times t}$ with probability $\geq 1 - 2/q$. Thus, letting the simulator output a random full-rank matrix $\mathbf{X}$ on input $(1^n, 0)$ results in privacy error of at most $2/q$. Correct decryption occurs when both $\mathbf{X}, \mathbf{R}$ generated by $\mathsf{SampMat(x)}$ are of full rank. By union bound, this occurs with probability $\geq 4/q$. Thus, we have obtained a $(4/q, 2^{-n}) - \mathsf{SRE}$.

However, the $4/q$ privacy error bound becomes significant for small q (and trivial for $q = 4$). More generally, even though a tighter analysis yields some non-trivial guarantees starting $q = 2$, the privacy error is $\Theta(1/q)$. Thus, for $4/q > \varepsilon$, we need a more complicated technique. The first idea is to repeat $\mathsf{SampMat(x)}$ k_1 times to amplify correctness to the proper level 2^{-n}. However, this can only increase the privacy error. Luckily, we can remedy this situation by using Lemma 6, and "mixing in" some $k_2 \gg k_1$ samples from a suitable distribution C, which is independent of x. Details follow.

Claim. There exist distributions $\mathcal{A}', \mathcal{D}, \mathcal{B}$, where $\mathcal{D}$ is the same for all $x \notin H$ and efficiently samplable, such that $\mathsf{SampMat}(1) = p_1\mathcal{A}' + (1 - p_1)\mathcal{B}$ and $\mathsf{SampMat}(x) = p_2\mathcal{D} + (1 - p_2)\mathcal{A}'$. The probabilities p_1, p_2 are also independent of x, and $p_1 \geq 0.14/q$.

The observations in Claim E.3 allow us to use Lemma 10. In particular, it is critical for encoding efficiency that p_1 is not too small. More precisely, to simulate $\mathsf{enc}_\varepsilon(x)$, take $k'_1 = \lceil \log_{1-p_q^2} \varepsilon \rceil$ samples $s_1, \ldots, s_{k'_1}$ of $Ber(p_2)$. Replace all 1's by independent samples of D, and 0's by samples of $\mathsf{SampMat}(1)$. Let $k_2 = k_2(k'_1, p_1, \varepsilon)$. Generate k_2 samples $s_{k'_1+1}, \ldots, s_{k'_1+k_2}$ of $\mathsf{SampMat}(1)$. Output a random permutation of $s_1, \ldots, s_{k_1+k_2}$. The locations and distribution of samples from $\mathcal{D}$ are distributed as in $\mathsf{enc}_\varepsilon(x)$. Applying Lemma 10 with $k_1 = v$, where v is the (unknown) number of 0's in the above experiment, $p = p_1$, $\mathcal{A} = \mathcal{A}'$ and $\mathcal{C} = \mathsf{SampMat}(1)$ results in a simulation error of ϵ.

In particular, not knowing k_1 precisely is not a problem, since k_2 is monotonically increasing in k_1 (and we could only overestimate k_1). Similarly, a bound on p_1, as we use above, rather then an exact value suffices for the same reason. Note that the simulation does not require knowing the distribution $\mathcal{A}'$, as Lemma 10 allows us to replace instances of $\mathcal{A}'$ with instances of $\mathsf{SampMat}(1)$!

It remains to prove Claim E.3. Let Good_x denote the event that $\mathbf{X}$ has rank t in $\mathsf{SampMat}(x)$. As $I(x_1), \ldots, I(x_{t-1})$ are independent vectors, its complement $\overline{\mathsf{Good}_x}$ is exactly the event where $I(x_1), \ldots, I(x_{t-1})$ span $I(x)$ for $x \notin H$. Let $\mathcal{A}'$ denote the distribution of $\mathsf{SampMat}(x)$ conditioned on Good_x, and $\mathcal{C} \triangleq Mat(1)$. It is easy to see that $Mat(x) = p_2\mathcal{D} + (1-p_2)\mathcal{A}'$, where $\mathcal{D}$ is the uniform distribution over all matrices in $\mathbb{Z}_q^{t\times t}$ ($\mathbf{X}' = \mathbf{R}\mathbf{X}$, conditioned on Good_x). Also, it is easy to see that $p_2 = Pr_{Mat(x)}(\mathsf{Good}_x) = \prod_{i=1}^{t-1} 1 - q^{i-t}$ (the precise value of p_2 is needed for the simulation).

Now, to see that $Mat(1) = p_1\mathcal{A}' + (1-p_1)\mathcal{B}$ for $p_1 \geq p_q$, we show that the distribution $\mathsf{SampMat}(1)$ conditioned on the event $\mathsf{Bad}_{Sim} = I(x) \in span(I(x_1), \ldots, I(x_{t-1}))$ equals A'.

As to bounding p_1, we show that Bad_{Sim} occurs with probability $p_1 \geq 0.14p_q$. Bad_{Sim} occurs if, for instance, the first $t-2$ $I(x_i)$'s are independent, and $I(x_{t-1})$ is of the form $aI(x) + \mathbf{b}$, for $\mathbf{b} \in span(I(x_1), \ldots, I(x_{t-2})), a \neq 0$. The probability of this event is $p_1 \geq p_q(q-1)/q^2 \geq p_q/2q \geq 0.14/q$ (for all $x \notin H$).

It remains to prove that $Mat(1)$ conditioned on Bad_{Sim} equals A'. We observe that A' is of the form $\mathbf{R}\mathbf{X}$, where $\mathbf{X}$ is uniform over $\mathbb{Z}_q^{t\times t}$ where $\mathbf{X}_t = x$ and the other rows span x (denote the support $\mathbf{X}$ by S_x), and $\mathbf{R}$ is a random matrix. $Mat(1)|\mathsf{Bad}_{Sim}$ is uniform over $\mathbb{Z}_q^{t\times t}$ where $\mathbf{X}_t = 0$ and the other rows span x (denote the support of $\mathbf{X}$ by S_1), and $\mathbf{R}$ is a random matrix. We show that the exists a bijection M_x from S_x to S_1 such that $M_x(\mathbf{X}) = \mathbf{T}_{x,\mathbf{X}}\mathbf{X}$, where $\mathbf{T}_{x,\mathbf{X}}$ is an invertible matrix. This way, the conditional distributions conditioned on $\mathbf{X} = \mathbf{V}$ in $Mat(x), Mat(1)$ respectively $\mathbf{R}\mathbf{V}$ and $\mathbf{R}(\mathbf{T}_{x,\mathbf{V}}\mathbf{V}) = (\mathbf{R}\mathbf{T}_{x,\mathbf{V}})\mathbf{V}$ have the same distribution for all $\mathbf{V} \in S_x$. Thus, $Mat(1)|\mathsf{Bad}_{Sim}$ equals A', as required. The mapping $M_x(\mathbf{X})$ is defined by $\mathbf{T}_{x,\mathbf{X}} = [e_1; \ldots, e_{t-1}; l]$, where l satisfies $l_t = -1$, and $(l_1, \ldots, l_{t-1})$ are the coefficients of the $\mathbf{X}_i$'s for $i < t$ that yields $\mathbf{X}_t$. Clearly, $\mathbf{T}_{x,\mathbf{X}}$ is invertible, and M_x is indeed a bijection.

Remark 2. The above construction extends for general finite G and $H \lhd G$. The only required adaptation is a procedure $\mathsf{samp}(H)$ for H which is not necessarily Abelian. An Abelian H can be perfectly sampled given any generating set of it. However, we can use a more sophisticated algorithm of [7] for sampling general finite groups $\Omega(N)$-close to uniformly given a set of generators for the group, and a bound N on the group's size. The sampling procedure is efficient in $\log(N)$.

E.4 Candidate Language

We define a language L which is an (extension of an) instance of ASM.

Let q denote the smallest, say, $n/10$-bit prime. For a length parameter n, let $p_1 = m_1 q^{t_1} + 1, p_2 = m_2 q^{t_2} + 1$ denote the two smallest distinct n-bit primes of this form where both $t_i > 0$. We define $G = G_1 \times G_2$, where G_i is the order-q subgroup

of $\mathbb{Z}^*_{p_i}$. Elements of G are encoded by pairs of integers $(a_1, a_2) \in [p_1 - 1] \times [p_2 - 1]$ satisfying $a_i^q = 1 \bmod p_i$ for both $i \in [2]$ (that is, this set of (a_1, a_2) equals $\mathsf{Im}(E)$). Multiplication in G is done by modular element-wise multiplication, which is clearly efficient assuming finding p_1, p_2 is efficient (this is Mul). In particular, p_1, p_2 are re-computed given an input $x = (x_1, x_2)$, based on $n = |x_1|$, in order to evaluate $\mathsf{Mul}(x, y)$. H is generated by a fixed (g_1, g_2), where each g_i is a generator of G_i (determined in a fixed way, as we will see below). $\mathsf{Yes}_n, \mathsf{No}_n$ are as induced by the specification of $\mathsf{ASM}^{G,H}$.

To complete the specification of the promise problem, it remains to specify a PPT algorithm $\mathsf{GenParams}(1^n)$ for finding q, p_1, p_2, g_1, g_2 (to make Mul efficient), and $\mathsf{Gen}(1^n)$. To find q start from $2^{n/10-1}+1$ until finding primes with the suitable properties. To find p_1, p_2, start from the smallest $i \geq 2^{n+1}+1$ which is of the form $aq+1$, and iterate in steps of q, until finding two primes p_1, p_2 of the form above. These procedures are efficient based on strong, but widely believed conjectures on the maximal gaps between consecutive primes in arithmetic progressions starting at x being at most $\mathsf{polylog}(x)$ steps apart. Namely

Conjecture 1 (Cramer). Let $a, q > 0$ be integers with $gcd(a, q) = 1$. Then there exists a constant c, such that every pair of consecutive primes in the sequence starting at $qx + a$ are at distance $O(\log^c x)$. In an idealized model where primes in $[x]$ are distributed uniformly this gap can be proved with $c = 2$ [12] for all fixed (a, q). This model of [12] is very strong, but is widely believed to give the correct prediction, and has empirical support [25].[16]

We apply this conjecture with $(a, q, x) = (1, 1, 2^{n/10})$ to find q, and $(a, q, x) = (1, q, 2^n)$ to find p_1, p_2. Now, $g_i = \mathsf{find\text{-}gen\text{-}mod}(p_i, q)$, where $\mathsf{find\text{-}gen\text{-}mod}$ is defined as follows.

Let $p = q^t m + 1$ where $t > 0, gcd(m, q) = 1$. For each $i \in [\log p^8]$, check whether $g = i^{(p-1)/q} \neq 1$:

1. If so, output g.
2. Otherwise, continue.

Clearly, this procedure is efficient. It always returns a proper generator of G_i assuming the ERH. This holds, as the range $[\log p^8]$ always contains a generator i of $\mathbb{Z}^*_p$ for a prime p [31]. Thus, Gen outputting such (g_1, g_2) is efficient assuming the ERH.

To make the above promise problem a language, observe that it is easy to efficiently check that x is in $E(G)$ for some n. If not $x = (x_1, x_2)$, where the x_i's are integers of the same length n, then it is in $\overline{L}$. Otherwise, check whether $x = (x_1, x_2)$ is in $support(E)$, by computing q, p_1, p_2 from $1^{|x|}$, and checking that $x^q = 1 \bmod p_i$.

We conclude that the promise problem can be turned into a language by putting all malformed inputs x in $\overline{L}$, and setting $L = \cup_n \mathsf{Yes}_n$.

[16] This is to compare with the PNT for arithmetic progressions, that only states that the overall density of such primes is $\Omega(1/\log x)$. The maximal gaps assuming the ERH are only bounded by $\tilde{O}(\sqrt{x})$ [18].

Thus, it has a (tweaked) SRE based on our construction E.3 for the corresponding ASM instance. This (tweaked) ASM-based SRE for it checks whether $x \in E(G)$. If so, applies the SRE from E.3 to x. Otherwise, replaces x by $(1, g_2)$, say, which is in $\mathsf{No}_n = E(G)/E(H)$. Apply the SRE in E.3 to it.

The explicit construction for the special case is included here for completeness, and to build intuition by considering a special simple case.

Construction 2.

- $\mathsf{enc}(\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}))$*: Generate parameters* p_1, p_2, q, g_1, g_2 *from* $1^{|x_1|}$ *as explained above. Check that* $x_i^{(p_i-1)/q} = 1$ *mod* p_i *for* $i \in [2]$*. If not, replace* x *by* $(1, g_2)$*. Pick* $i \in [q-1], s \in [q]$ *at random. Output* $x^i \cdot (g_1, g_2)^s = ({g_1}^s x_1^i, {g_2}^s x_2^i)$*.*
- $\mathsf{dec}(\mathsf{out})$*: Check whether* $\mathsf{out} \in H$*: if* $\mathsf{out} = (g_1, g_2)^i$ *for some* i*, output* 1*. Otherwise, output* 0*.*

As explained in the following section, this is a perfect SRE for the problem. The case of a malformed x is handled by replacing it by $(1, g_2)$, which is in $E(G)/E(H)$.

It is perfectly correct, as the mapping x^i for $i \in [q-1]$ maps H onto itself, and permutes the other cosets of G/H. multiplying by $(g_1, g_2)^s$ (an element of H) does not change the coset. As to privacy, the mapping x^i already hides coset (of G/H) information except for whether $x \in H$ (by uniformity of i in $[q-1]$). Now, $(g_1, g_2)^s$ for a random $s \in [q]$ is uniform over H, so multiplying x^i by it wipes the information about which coset member x was.

This language is not in P/poly based on the following assumption, which is an instance of a modified co-DDH assumption [16], to which we refer as the **mod-co-DDH** assumption. Let p_1, p_2, g_1, g_2 be as above. Then, the language $L = \cup_n L_n$, where

$$L_n = \{x = (y, z) | \text{there exists } a \in [q], \text{ such that } {g_1}^a = y \bmod p_1, {g_2}^a = z \bmod p_2\}$$

is not in P/poly. This assumption is related to DDH, with p_1, p_2 being different. It is potentially weaker, as g_1, g_2 here are fixed, instead of being part of the input (observe that for $p_1 = p_2$, the assumption does not hold).

Theorem 6. *Assume the ERH, and Conjecture 1. Then there exists (a sequence of) groups* G, H *(with associated algorithms* Mul, E*), such that* $\mathsf{ASM}^{G,H}$ *extended to a language* $L_{G,H}$ *by placing malformed inputs in* $\overline{L_{G,H}}$ *has a* $(0,0)$-SRE*.* $L_{G,H}$ *is not in* P/poly *under the* ***mod-co-DDH*** *assumption.*

Remark 3. The above language falls into a simple case for our ASM construction, where G/H is cyclic ($t = 1$). The full(er) power of our construction comes out for a modified language where $G = L_{p_1} \times L_{p_2} \times L_{p_3}$, and H is generated by some (h_1, h_2, h_3). In this case $G/H \cong \mathbb{Z}_q^2$. In this case, a set of generators $\mathbf{g_1}, \mathbf{g_2}$ of G that generates $G/H (\cong \mathbb{Z}_q^2)$ (for some isomorphism I) is found by picking a pair of random elements in G (succeeds w.p $(1 - 1/q)(1 - 1/q^2) = 1 - o(n^{-1})$). Still, as q is large, even this is not the hardest case, which occurs when $G/H = \mathbb{Z}_q^t$ where $t > 1$ and q is small.

A non-uniform SRE *for an* ASM*-based language.* In the non-uniform setting, we have further freedom to pick q, p_1, p_2, g_1, g_2 as some numbers of $n/10$ and n bits respectively such that $p_i = qm_i + 1$, and g_i generating the corresponding G_i (there are $exp(n)$ sequences for a given n). Thus, hard-coding them into enc unconditionally results in a non-uniform SRE for the induced language L_n based on these parameters, as constructed above. The induced family of languages defined by each parameter sequence offers a broader range of candidates for a language L_n outside of P/poly. Thus, potentially, the assumption **mod-co-DDH** can be relaxed as follows. **enh-mod-co-DDH** states that there exists a sequence of (q, p_1, p_2, g_1, g_2) as above, where such that the corresponding language L_n is outside of P/poly. We obtain the following theorem.

Theorem 7. *There exists (a sequence of) groups G, H (with associated algorithms Mul, E), such that* $\mathsf{ASM}^{G,H}$ *extended to a language $L_{G,H}$ by adding malformed inputs to $\overline{L_{G,H}}$ has a $(0,0)$-*SRE*. $L_{G,H}$ is not in* P/poly *under the* ***enh-mod-co-DDH*** *assumption.*