2018 IEEE 59th Annual Symposium on Foundations of Computer Science

PPP-Completeness with Connections to Cryptography

Katerina Sotiraki
EECS and CSAIL
MIT
Cambridge, USA
katesot@mit.edu

Manolis Zampetakis
EECS and CSAIL
MIT
Cambridge, USA
mzampet@mit.edu

Giorgos Zirdelis
CCIS
Northeastern University
Boston, USA
zirdelis.g@husky.neu.edu

***Abstract*—Polynomial Pigeonhole Principle (PPP) is an important subclass of TFNP with profound connections to the complexity of the fundamental cryptographic primitives: collision-resistant hash functions and one-way permutations. In contrast to most of the other subclasses of TFNP, no complete problem is known for PPP. Our work identifies the first PPP-complete problem without any circuit or Turing Machine given explicitly in the input, and thus we answer a longstanding open question from [Papadimitriou1994]. Specifically, we show that constrained-SIS, a generalized version of the well-known Short Integer Solution problem (SIS) from lattice-based cryptography, is PPP-complete.**

In order to give intuition behind our reduction for constrained-SIS, we identify another PPP-complete problem with a circuit in the input but closely related to lattice problems. We call this problem BLICHFELDT and it is the computational problem associated with Blichfeldt's fundamental theorem in the theory of lattices.

Building on the inherent connection of PPP with collision-resistant hash functions, we use our completeness result to construct the first natural hash function family that captures the hardness of all collision-resistant hash functions in a worst-case sense, i.e. it is natural and universal in the worst-case. The close resemblance of our hash function family with SIS, leads us to the first candidate collision-resistant hash function that is both natural and universal in an average-case sense.

Finally, our results enrich our understanding of the connections between PPP, lattice problems and other concrete cryptographic assumptions, such as the discrete logarithm problem over general groups.

***Keywords*-complexity of total search problems; pigeonhole principle; collision resistant hash functions; short integer solutions;**

I. Introduction

The fundamental task of *Computational Complexity* theory is to classify computational problems according to their inherent computational difficulty. This led to the definition of *complexity classes* such as NP which contains the *decision* problems with *efficiently* verifiable proofs in the "yes" instances. The *search* analog of the class NP, called FNP, is defined as the class of *search* problems whose decision version is in NP. The same definition extends to the class FP, as the search analog of P. The seminal works of [1], [2] considered search problems in FNP that are *total*, i.e. their decision version is always affirmative and thus a solution must always exist. This totality property makes the definition of FNP inadequate to capture the intrinsic complexity of total problems in the appropriate way as it was first shown in [1]. Moreover, there were evidences for the hardness of total search problems e.g. in [3]. Megiddo and Papadimitriou [4] defined the class **TFNP** that contains the total search problems of FNP, and Papadimitriou [2] proposed the following classification rule of problems in TFNP:

> Total search problems should be classified in terms of the profound mathematical principles that are invoked to establish their totality.

Along these lines, many subclasses for TFNP have been defined. Johnson, Papadimitriou and Yannakakis [1] defined the class **PLS**. A few years later, Papadimitriou [2] defined the complexity classes **PPA**, **PPAD**, **PPADS** and **PPP**, each one associated with a profound mathematical principle in accordance with the above classification rule. More recently, the classes **CLS** and **PWPP** were defined in [5] and [6], respectively. In Section I-A we give a high-level description of all these classes.

Finding complete problems for the above classes is important as it enhances our understanding of the underlying mathematical principles. In turn, such completeness results reveal equivalences between total search problems, that seemed impossible to discover without invoking the definition of these classes. Since the definition of these classes in [1], [2] it was clear that the completeness results about problems that do not have explicitly a Turing machine or a circuit as a part of their input are of particular importance. For this reason it has been established to call such problems *natural* in the context of the complexity of total search problems (see [7]).

Many natural complete problems are known for PLS and PPAD, and recently natural complete problems for PPA were identified too (see Section I-A). However, no natural complete problems are known for the classes PPP, PWPP that have profound connections with the hardness of important cryptographic primitives, as we explain later in detail.

Our Contributions. Our main contribution is to provide the first natural complete problems for PPP and PWPP, and

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00023

thus solve a longstanding open problem from [2]. Beyond that, our PPP completeness results lead the way towards answering important questions in cryptography and lattice theory as we highlight below.

UNIVERSAL COLLISION-RESISTANT HASH FUNCTION. Building on the inherent connection of PWPP with *collision-resistant hash functions*, we construct a natural hash function family $\mathcal{H}_{\mathsf{cSIS}}$ with the following properties:

- **Worst-Case Universality.** No efficient algorithm can find a collision in every function of the family $\mathcal{H}_{\mathsf{cSIS}}$, unless worst-case collision-resistant hash functions do not exist.

  Moreover, if an (average-case hard) collision-resistant hash function family exists, then there exists an efficiently samplable distribution $\mathcal{D}$ over $\mathcal{H}_{\mathsf{cSIS}}$, such that $(\mathcal{D}, \mathcal{H}_{\mathsf{cSIS}})$ is an (average-case hard) collision-resistant hash function family.
- **Average-Case Hardness.** No efficient algorithm can find a collision in a function chosen *uniformly at random* from $\mathcal{H}_{\mathsf{cSIS}}$, unless we can efficiently find short lattice vectors in any (worst-case) lattice.

The first property of $\mathcal{H}_{\mathsf{cSIS}}$ is reminiscent of the existence of *worst-case* one-way functions from the assumption that $\mathsf{P} \neq \mathsf{NP}$ [8]. The corresponding assumption for the existence of worst-case collision-resistance hash functions is assuming $\mathsf{FP} \neq \mathsf{PWPP}$, but our hash function family $\mathcal{H}_{\mathsf{cSIS}}$ is the first natural definition that does not involve circuits, and admits this strong completeness guarantee in the worst-case.

The construction and properties of $\mathcal{H}_{\mathsf{cSIS}}$ lead us to the first candidate of a *natural* and *universal collision-resistant hash function family*. The idea of universal constructions of cryptographic primitives was initiated by Levin in [9], who constructed the first universal one-way function and followed up by [10], [11]. Using the same ideas we can also construct a universal collision-resistant hash function family as we describe in Appendix A. The constructed hash function though invokes in the input an explicit description of a Turing machine and hence it fails to be *natural*, with the definition of naturality that we described before. In contrast, our candidate construction is natural, simple, and could have practical applications.

COMPLEXITY OF LATTICE PROBLEMS IN PPP. The hardness of lattice problems in $\mathsf{NP} \cap \mathsf{coNP}$ [12] has served as the foundation for numerous cryptographic constructions in the past two decades. This line of work was initiated by the breakthrough work of Ajtai [13], and later developed in a long series of works (e.g. [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]). This wide use of search (approximation) lattice problems further motivates their study.

We make progress in understanding this important research front by showing that:

1) the computational problem BLICHFELDT associated with Blichfeldt's theorem, which can be viewed as a generalization of Minkowski's theorem, is PPP-complete,
2) the cSIS problem, a constrained version of the Short Integer Solution (SIS), is PPP-complete,
3) we combine known results and techniques from lattice theory to show that most approximation lattice problems are reducible to BLICHFELDT and cSIS.

These results create a new path towards a better understanding of lattice problems in terms of complexity classes.

COMPLEXITY OF OTHER CRYPTOGRAPHIC ASSUMPTIONS. Besides lattice problems, we discuss the relationship of other well-studied cryptographic assumptions and PPP. Additionally, we formulate a white-box variation of the *generic group model* for the discrete logarithm problem [27]; we observe that this problem is in PPP and is another natural candidate for being PPP-complete.

A. Related Work

In this section we discuss the previous work on the complexity of total search problems, that has drawn attention from the theoretical computer science community over the past decades. We start with a high-level description of the total complexity classes and then discuss the known results for each one of them.

PLS. The class of problems whose totality is established using a potential function argument.
Every finite directed acyclic graph has a sink.

PPA. The class of problems whose totality is proved through a parity argument.
Any finite graph has an even number of odd-degree nodes.

PPAD. The class of problems whose totality is proved through a directed parity argument.
All directed graphs of degree two or less have an even number of degree one nodes.

PPP. The class of problems whose totality is proved through a pigeonhole principle argument.
Any map from a set S to itself either is onto or has a collision.

Using the same spirit two more classes were defined after [2], in [5] and [6].

CLS. The class of problems whose totality is established using both a potential function argument and a parity argument.

PWPP. The class of problems whose totality is proved through a weak pigeonhole principle.
Any map from a set S to a strict subset of S has a collision.

Recently, a syntactic analog PTFNP of the semantic class TFNP has been defined in [28], and a complete problem for

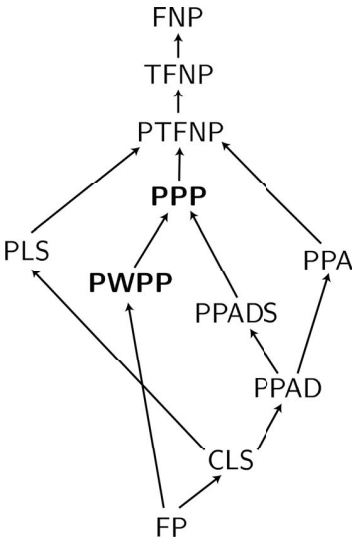

Figure 1: The classes PPP and PWPP in the TFNP world.

this class has been identified. It has also been shown that all the classes we described above are subsets of PTFNP. Oracle separations between all these classes are known [29], with the only exception of whether PLS is contained in PPAD.

PLS-completeness. The class PLS represents the complexity of local optimization problems. Some important problems that have been shown to be PLS-complete are: Local Max-Cut [30], Local Travelling Salesman Problem [31], and Finding a Pure Nash Equilibrium [32]. Recently, important results for the smoothed complexity of the Local Max-Cut problem were shown in [33], [34].

PPAD-completeness. Arguably, the most celebrated application of the complexity of total search problems is the characterization of the computational complexity of finding a Nash equilibrium in terms of PPAD-completeness [35], [36]. This problem lies in the heart of game theory and economics. The proof that Nash Equilibrium is PPAD-complete initiated a long line of research in the intersection of computer science and game theory and revealed connections between the two scientific communities that were unknown before (e.g. [37], [38], [39], [40], [41], [42], [43], [44], [45]).

PPA-completeness. PPA-complete problems usually arise as the undirected generalizations of their PPAD-complete analogs. For example, Papadimitriou [2] showed that Sperner's Lemma in a 3-D cube is PPAD-complete and later Grigni [46] showed that Sperner's Lemma in a 3-manifold consisting of the product of a Möbius strip and a line segment is PPA-complete. Since Möbius strip is non-orientable, this indeed is a non-directed version of the Sperner's Lemma. Similarly, other problems have been showed to be PPA-complete, all involving some circuit as an input in their definition [47], [48], [49]. Recently, the first natural PPA-complete problem, without a circuit as part of the input, has been identified in [7]. This illustrates an interesting relation between PPA and complexity of social choice theory problems.

CLS-completeness. The CLS class was defined in [5] to capture the complexity of problems such as P-matrix LCP, computing KKT-points, and finding Nash equilibria in congestion and network coordination games. Recently, it has been proved that the problem of finding a fixed point whose existence invokes Banach's Fixed Point Theorem, is CLS-complete [50], [51].

TFNP and cryptography. The connection of TFNP and cryptography was illustrated by Papadimitriou in [2], where he proved that if PPP $=$ FP then *one-way permutations* cannot exist. In [52], a special case of integer factorization was shown to be in PPA $\cap$ PPP. This was generalized in [6] by proving that the problem of factoring integers is in PPA $\cap$ PPP under randomized reductions. Recently, strong cryptographic assumptions were used to prove the average-case hardness of PPAD and CLS [53], [54], [55]. In [56] it was shown that average-case PPAD hardness does not imply one-way function, whereas in [57] it was shown that any hard on average problem in NP implies the average case hardness of TFNP. Finally, in [58] it is proved that the existence of *multi-collision resistant hash functions* is equivalent with a variation of the total search problem RAMSEY, which is not known to belong to any of the above complexity classes. Interestingly, they prove that another variation of RAMSEY called *colorful-Ramsey* (C-RAMSEY) is PWPP-hard. Although this an important result, the problem C-RAMSEY still invokes a circuit in the input and in not known to be in PWPP, hence does not resolve the problem of identifying a natural complete problem for PWPP.

TFNP and lattices. In [59] it was shown that the computational analog of Minkowski's theorem (namely MINKOWSKI) is in PPP, was conjectured that it is also PPP-complete. The authors justified their conjecture by showing that EQUAL-SUMS, a problem from [2] that is conjectured to be PPP-complete, reduces to MINKOWSKI. Additionally, they show that a number theoretic problem called DIRICHLET reduces to MINKOWSKI, and thus is in PPP. In [60] it is proven that the problem NUMBER-BALANCING is equivalent to a polynomial approximation of Minkowski's theorem in the ℓ_2 norm (via Cook reductions for both directions).

B. Roadmap of the paper

We give here a brief description of the results contained in this paper. First we briefly describe the PPP-completeness of BLICHFELDT that illustrates some of the basic ideas behind our main result that cSIS is PPP-complete. Then, we present a brief description of our main theorem and its proof. We also describe the PPP-completeness of a weaker

version of cSIS and its relation with the definition of the first natural universal collision resistant hash function family in the worst-case sense. This proof also provides the first candidate for a collision resistant hash function family that is both natural and universal in the average-case sense.

Finally we mention, for completeness of our exposition, other lattice problems that are already known to belong to PPP and PWPP and more general other cryptographic assumptions that belong to PPP and PWPP.

For the complete proofs of the statement that we present in this paper we refer to the full version of our paper.

II. Our Results

Before we describe our results in more detail we define the class PPP more formally. The class PPP contains the set of problems that are reducible to the PIGEONHOLE CIRCUIT problem. The input to PIGEONHOLE CIRCUIT is a binary circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ and its output is either an $\underline{\boldsymbol{x}} \in \{0,1\}^n$ such that $\mathcal{C}(\underline{\boldsymbol{x}}) = \underline{\mathbf{0}}$, or a pair $\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \in \{0,1\}^n$ such that $\underline{\boldsymbol{x}} \neq \underline{\boldsymbol{y}}$ and $\mathcal{C}(\underline{\boldsymbol{x}}) = \mathcal{C}(\underline{\boldsymbol{y}})$.

Our first and technically most challenging result is to identify and prove the PPP-completeness of two problems, both of which share similarities with lattice problems. For our exposition, a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ can be viewed as a finitely generated additive subgroup of $\mathbb{Z}^n$. A lattice $\mathcal{L} = \mathcal{L}(\mathbf{B}) := \mathbf{B} \cdot \mathbb{Z}^n$ is generated by a full-rank matrix $\mathbf{B} \in \mathbb{Z}^{n \times n}$, called *basis*. In the rest of this section we also use the *fundamental parallelepiped* of $\mathcal{L}$ defined as $\mathcal{P}(\mathcal{L}) := \mathbf{B} \cdot [0,1)^n$.

A. BLICHFELDT *is* PPP*-complete.*

We define the BLICHFELDT problem as the computational analog of Blichfeldt's theorem.

Theorem II.1 (Blichfeldt's Theorem [61])**.** *Let $\mathbf{B} \in \mathbb{Z}^{n \times n}$ be a set of n-dimensional linearly independent integer vectors and a measurable set $S \subseteq \mathbb{R}^n$. If $\mathrm{vol}(S) > \det(\mathcal{L}(\mathbf{B}))$, then there exist $\boldsymbol{x}, \boldsymbol{y} \in S$ with $\boldsymbol{x} \neq \boldsymbol{y}$ and $\boldsymbol{x} - \boldsymbol{y} \in \mathcal{L}(\mathbf{B})$.*

BLICHFELDT Problem.
INPUT: An n-dimensional basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$ and a set $S \subseteq \mathbb{Z}^n$ described by the value function representation $(s, \mathcal{V}_S)$.
OUTPUT: If $s < \det(\mathcal{L}(\mathbf{B}))$, then the vector $\mathbf{0}$. Otherwise, one of the following:

0) a number $z \in [s]$ such that $\mathcal{V}_S(z) \notin S$ or two numbers $z, w \in [s]$ such that $\mathcal{V}_S(z) = \mathcal{V}_S(w)$,
1) a vector $\mathbf{x}$ such that $\mathbf{x} \in S \cap \mathcal{L}$,
2) two vectors $\mathbf{x} \neq \mathbf{y}$, such that $\mathbf{x}, \mathbf{y} \in S$ and $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

The input of BLICHFELDT is a basis for a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ and a set $S \subseteq \mathbb{Z}^n$ of cardinality greater or equal to the volume of $\mathcal{P}(\mathcal{L})$. Its output is either a point in S that belongs to $\mathcal{L}$, or for two (different) points in S such that their difference belongs to $\mathcal{L}$. In the overview below, we explain why such an output always exists. Notice that finding a solution to BLICHFELDT becomes trivial if the input representation of S has length proportional to its size, i.e. one can iterate over all element pairs of S. The problem becomes challenging when S is represented succinctly. We introduce a notion for a succinct representation of sets that we call *value function*. Informally, a value function for a set S is a small circuit that takes as input $\lceil \log(|S|) \rceil$ bits that describe an index $i \in \{0, \ldots, |S| - 1\}$, and outputs $\mathbf{s}_i \in S$.

We give a proof overview of our first main theorem, and highlight the obstacles that arise, along with our solutions.

Theorem II.2. *The* BLICHFELDT *problem is* PPP*-complete.*

PPP Membership of BLICHFELDT Overview. We denote with $[n]$ the set $\{0, \ldots, n-1\}$. We define the map $\boldsymbol{\sigma} : \mathbb{Z}^n \to \mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n$ that reduces any point in $\mathbb{Z}^n$ modulo the parallelepiped to $\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n$, i.e. $(\mathrm{mod}\ \mathcal{P}(\mathcal{L}))$. Using $\boldsymbol{\sigma}$ we can efficiently check the membership of any $\mathbf{v} \in \mathbb{Z}^n$ in $\mathcal{L}$, by checking if $\boldsymbol{\sigma}$ maps $\mathbf{v}$ to the origin. Observe that if $\boldsymbol{\sigma}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{y})$ then $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

It is well known that $\mathrm{vol}(\mathcal{P}(\mathcal{L})) = |\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n|$, hence the input requirement for S is equivalent to $|S| \geq |\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n|$. Notice that the points of S after applying the map $\boldsymbol{\sigma}$, either have a collision in $\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n$ or a preimage of the origin exists in S. It follows by a pigeonhole argument that a solution to BLICHFELDT always exists. For the rest of this part we assume that $|S| = |\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n|$ and let $n = \lceil \log(|S|) \rceil$.

We construct a circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ that on input an appropriate index i, evaluates the value function of S to obtain $\mathbf{s}_i \in S$, and computes $\boldsymbol{\sigma}(\mathbf{s}_i)$. The most challenging part of the proof is to construct an efficient map from $\boldsymbol{\sigma}(S)$ to $[|\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n|]$ in the following way. We define an appropriate parallelepiped $D = [L_1] \times [L_2] \times \cdots \times [L_n]$ where the L_i are non-negative integers, and a bijection $\boldsymbol{\pi} : \mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n \to D$. Because D is a cartesian product, a natural efficient indexing procedure exists as described in the full version of our paper. This allows to map $\boldsymbol{\pi}(\boldsymbol{\sigma}(\mathbf{s}_i))$ to $j \in [|\mathcal{P}(\mathcal{L}) \cap \mathbb{Z}^n|]$. The circuit $\mathcal{C}$ outputs the binary decomposition of j. It follows that any $\underline{\boldsymbol{x}}$ such that $\mathcal{C}(\underline{\boldsymbol{x}}) = \underline{\mathbf{0}}$ corresponds to a vector $\mathbf{x} \in S$ such that $\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{0}$. On the other hand, a collision $\mathcal{C}(\underline{\boldsymbol{x}}) = \mathcal{C}(\underline{\boldsymbol{y}})$ with $\underline{\boldsymbol{x}} \neq \underline{\boldsymbol{y}}$ corresponds to a collision $\boldsymbol{\sigma}_S(\mathbf{x}) = \boldsymbol{\sigma}_S(\mathbf{y})$, where $\boldsymbol{\sigma}_S$ is the restriction of $\boldsymbol{\sigma}$ on S, and hence $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

PPP Hardness of BLICHFELDT Overview. We start with a circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ that is an input to PIGEONHOLE CIRCUIT. We construct a set S and a lattice $\mathcal{L}$ as input to BLICHFELDT in the following way. The set S contains the elements $\mathbf{s}_{\underline{\boldsymbol{x}}} = \begin{bmatrix} \underline{\boldsymbol{x}} \\ \mathcal{C}(\underline{\boldsymbol{x}}) \end{bmatrix} \in \{0,1\}^{2n}$ and is represented succinctly with the value function that maps $\underline{\boldsymbol{x}}$ to $\mathbf{s}_{\underline{\boldsymbol{x}}}$. Notice that $|S| = 2^n$. The lattice $\mathcal{L}$ consists of all $\underline{\boldsymbol{y}} \in \{0,1\}^{2n}$ that satisfy the equation $[\mathbf{0}_n \ \ \mathbf{I}_n] \cdot \underline{\boldsymbol{y}} = \mathbf{0}\ (\mathrm{mod}\ 2)$.

It is also known that we can efficiently obtain a basis from this description of $\mathcal{L}$ and in addition the volume of $\mathcal{P}(\mathcal{L})$ is at most 2^n. Thus, S and $\mathcal{L}$ is a valid input for BLICHFELDT.

The output of BLICHFELDT is either an $\mathbf{s}_{\underline{\boldsymbol{x}}} = \begin{bmatrix} \underline{\boldsymbol{x}} \\ \mathcal{C}(\underline{\boldsymbol{x}}) \end{bmatrix} \in S \cap \mathcal{L}$ that (by construction of $\mathcal{L}$) implies $\mathcal{C}(\underline{\boldsymbol{x}}) = \mathbf{0}$, or two different elements of S, $\mathbf{s}_{\underline{\boldsymbol{x}}} = \begin{bmatrix} \underline{\boldsymbol{x}} \\ \mathcal{C}(\underline{\boldsymbol{x}}) \end{bmatrix}$, $\mathbf{s}_{\underline{\boldsymbol{y}}} = \begin{bmatrix} \underline{\boldsymbol{y}} \\ \mathcal{C}(\underline{\boldsymbol{y}}) \end{bmatrix}$ with $\mathbf{s}_{\underline{\boldsymbol{x}}} - \mathbf{s}_{\underline{\boldsymbol{y}}} \in \mathcal{L}$ that implies $\underline{\boldsymbol{x}} \neq \underline{\boldsymbol{y}}$ and (by construction of $\mathcal{L}$) $\mathcal{C}(\underline{\boldsymbol{x}}) = \mathcal{C}(\underline{\boldsymbol{y}})$.

B. cSIS *is* PPP*-complete.*

Part of the input to BLICHFELDT is represented with a value function which requires a small circuit. As we explained before this makes BLICHFELDT a non-natural problem with the respect to the definition of naturality in the context of the complexity of total search problems. We now introduce a natural problem that we call *constrained Short Integer Solution* (cSIS), and show that it is PPP-complete. The cSIS problem is a generalization of the well-known *Short Integer Solution* (SIS) problem. We first define the notion of *binary invertible matrices* and we state their basic properties, then we formally define the cSIS problem.

Definition II.3 (BINARY INVERTIBLE MATRIX)**.** Let $\ell \in \mathbb{Z}_+$, $q \leq 2^\ell$ and $d, k \in \mathbb{N}$. First, we define the ℓ-th *gadget vector* $\boldsymbol{\gamma}_\ell$ to be the vector $\boldsymbol{\gamma}_\ell = \begin{bmatrix} 1 & 2 & 4 & \dots & 2^{\ell-1} \end{bmatrix}^T \in \mathbb{Z}_q^\ell$. Second, let $\mathbf{U} \in \mathbb{Z}_q^{d \times (d \cdot \ell)}$ be a matrix with non-zero elements only above the $(\ell+1)$-diagonal and $\mathbf{V} \in \mathbb{Z}_q^{d \times k}$ be an arbitrary matrix. We define the matrix $\mathbf{G} = \begin{bmatrix} (\mathbf{I}_d \otimes \boldsymbol{\gamma}_\ell^T + \mathbf{U}) & \mathbf{V} \end{bmatrix} \in \mathbb{Z}_q^{d \times (d \cdot \ell + k)}$ to be a *binary invertible* matrix.

It is evident from the definition of a binary invertible matrix, that it is not a fixed matrix but rather a collection of matrices. That is, after we fix q, the exact values of $\mathbf{G}$ depend on the choice of $\mathbf{U}$ and $\mathbf{V}$. For example, a special case of a binary invertible matrix is the *gadget matrix* $\mathbf{G} = \mathbf{I}_d \otimes \boldsymbol{\gamma}_\ell^T$ with $\mathbf{U} = \mathbf{0}^{d \times (d \cdot \ell)}$ and $k = 0$, that was defined in [62] and used in many cryptographic constructions (e.g. [22], [63], [23], [64], [65], [66], [67], [68], [69]).

Next, we formalize the main property of binary invertible matrices that is in the core of our proof for the inclusion of cSIS in PPP, and also explains the name "binary invertible".

Proposition II.4. *Let* $\mathbf{G} = \begin{bmatrix} (\mathbf{I}_d \otimes \boldsymbol{\gamma}_\ell^T + \mathbf{U}) & \mathbf{V} \end{bmatrix} \in \mathbb{Z}_q^{d \times (d \cdot \ell + k)}$ *be a binary invertible matrix and* $\mathbf{r}'$ *be an arbitrary vector in* $\mathbb{Z}_q^k$*. Then, for every* $\mathbf{b} \in \mathbb{Z}_q^d$*, there exists a vector* $\mathbf{r} \in \{0,1\}^{d \cdot \ell}$ *such that* $\mathbf{G} \begin{bmatrix} \mathbf{r} \\ \mathbf{r}' \end{bmatrix} = \boldsymbol{b} \pmod q$*. Additionally, the vector* $\mathbf{r}$ *is computable by a polynomial-size circuit and it is guaranteed to be unique when* $q = 2^\ell$*.*

Proof: For a simple illustration of the proposition, for a moment assume that $\mathbf{G} = \begin{bmatrix} \mathbf{W} & \mathbf{V} \end{bmatrix} \in \mathbb{Z}_q^{d \times (d+k)}$, where q is prime, $\mathbf{W} \in \mathbb{Z}_q^{d \times d}$ is an upper triangular matrix, and $\mathbf{V} \in \mathbb{Z}_q^{k \times d}$ is arbitrary. Then, for every $\mathbf{b} \in \mathbb{Z}_q^d$, using backwards substitution we can efficiently compute a vector $\mathbf{x} \in \mathbb{Z}_q^d$ such that $\mathbf{G} \begin{bmatrix} \mathbf{x} \\ \mathbf{r}' \end{bmatrix} = \mathbf{b} \pmod q$.

For the general case where $\mathbf{G} = \begin{bmatrix} (\mathbf{I}_d \otimes \boldsymbol{\gamma}_\ell^T + \mathbf{U}) & \mathbf{V} \end{bmatrix}$, we use again backward substitution. But because we require $\mathbf{r}$ to be binary, we make $\mathbf{r}$ to be the the binary decomposition of the corresponding solution in $\mathbb{Z}_q$. More precisely, we divide $\mathbf{r}$ into d parts of ℓ coordinates each, such that $\mathbf{r} = [\mathbf{r}_1 \dots \mathbf{r}_d]^T$. We define $\mathbf{g}_i^T$ to be the i-th row of the matrix $\mathbf{G}$. Then, the d-th part of $\mathbf{r}$ is equal to $\mathbf{r}_d = \text{bitDecomposition}\left(b_d - \mathbf{g}_d^T \begin{bmatrix} \mathbf{0} \\ \mathbf{r}' \end{bmatrix} \pmod q \right)$. Next, we recursively compute the t-th part $\mathbf{r}_t$ of $\mathbf{r}$, assuming we have already computed the parts $\mathbf{r}_{t+1}, \dots, \mathbf{r}_d$. The recursive relation for $\mathbf{r}_t$ is

$$\mathbf{r}_t = \text{bitDecomposition}\left(b_t - \mathbf{g}_t^T \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{r}_{t+1} \\ \mathbf{r}_{t+2} \\ \dots \\ \mathbf{r}_d \\ \mathbf{r}' \end{bmatrix} \pmod q \right), \tag{II.1}$$

where $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \dots & b_d \end{bmatrix}^T$. The fact that $\mathbf{r}$ can be computed by a polynomial sized circuit, follows easily from (II.1).

In the special case of $q = 2^\ell$, it is easy to see that for every $x \in [q]$ there exists a unique $\mathbf{r}_t \in \{0,1\}^\ell$ such that $\boldsymbol{\gamma}_\ell^T \mathbf{r}_t = x \pmod q$. Additionally, (when $q = 2^\ell$) for any $\mathbf{r}_t \in \{0,1\}^\ell$, it holds that $\boldsymbol{\gamma}_\ell^T \mathbf{r}_t < q$, and thus $\boldsymbol{\gamma}_\ell^T \mathbf{r}_t = x$ over $\mathbb{Z}$. But in this case, $\mathbf{r}_t$ is the binary decomposition of x, and it is unique. Because every $\mathbf{r}_t$ is unique, we get that $\mathbf{r}$ is also unique. ∎

Remark II.5. The property of Proposition II.4 is the only property of binary invertible matrices that we need for our proofs. We could potentially define binary invertible matrices in a more general way. For example, a binary invertible matrix could be a permutation of the columns of a matrix of Definition II.3. Our results follow immediately for the more general class of matrices that satisfy the properties of Proposition II.4. Let us denote by $\mathcal{S}$ this set of matrices. We focus on the more restrictive case of Definition II.3, not only for ease of the exposition, but also because given a matrix $\mathbf{A}$ there is no known efficient procedure to check whether $\mathbf{A} \in \mathcal{S}$. In fact, this problem is NP-complete, since we can encode a SUBSET-SUM instance in $\mathbf{A}$ and reduce SUBSET-SUM to checking whether $\mathbf{A} \in \mathcal{S}$. Alternatively, we could define a promise version of cSIS where $\mathbf{G}$, is promised to be in $\mathcal{S}$. However, this would deprive us from

a *syntactic* definition of cSIS.

We now define the Constrained Short Integer Solution problem.

cSIS Problem.
INPUT: A matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a binary invertible matrix $\mathbf{G} \in \mathbb{Z}_q^{d\times m}$ and a vector $\mathbf{b} \in \mathbb{Z}_q^d$ where $\ell \in \mathbb{Z}_+$, $q \leq 2^\ell$ and $m \geq (n+d)\cdot\ell$.
OUTPUT: One of the following:

1) a vector $\mathbf{x} \in \{0,1\}^m$ such that $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$ and $\mathbf{Gx} = \mathbf{b} \pmod q$,
2) two vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^m$ such that $\mathbf{x} \neq \mathbf{y}$ with $\mathbf{x}-\mathbf{y} \in \Lambda_q^\perp(\mathbf{A})$ and $\mathbf{Gx} = \mathbf{Gy} = \mathbf{b} \pmod q$.

The input is $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{G} \in \mathbb{Z}_q^{d\times m}$ and $\mathbf{b} \in \mathbb{Z}_q^d$, for some positive integer q and $m \geq (n+d)\lceil\log(q)\rceil$. The matrix $\mathbf{G}$ has the property that for every $\mathbf{b}$ we can efficiently find an $\mathbf{x} \in \{0,1\}^m$ such that $\mathbf{Gx} = \mathbf{b} \pmod q$. We define such matrices as *binary invertible*. The output is either a vector $\mathbf{x} \in \{0,1\}^m$ such that $\mathbf{Ax} = \mathbf{0} \pmod q$ and $\mathbf{Gx} = \mathbf{b} \pmod q$, or two different vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^m$ such that $\mathbf{A}(\mathbf{x}-\mathbf{y}) = \mathbf{0} \pmod q$ and $\mathbf{Gx} = \mathbf{Gy} = \mathbf{b} \pmod q$. We give a proof overview of the next theorem, and a full proof in the full version of the paper.

Theorem II.6. *The* cSIS *problem is* PPP*-complete.*

PPP MEMBERSHIP OF cSIS OVERVIEW. We show the membership of cSIS in PPP for the general class of binary invertible matrices $\mathbf{G}$ in the full version of the paper. In order to simplify the exposition, we assume that $q = 2^\ell$ and $\mathbf{G}$ to be the "gadget" matrix concatenated with a random matrix $\mathbf{V}$. That is, $\mathbf{G}$ has the form $\begin{bmatrix}\mathbf{I}_d \otimes \boldsymbol{\gamma}^T & \mathbf{V}\end{bmatrix}$ where $\boldsymbol{\gamma}^T = [1, 2, \ldots, 2^\ell]$.

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{G} \in \mathbb{Z}_q^{d\times m}$, and $\mathbf{b} \in \mathbb{Z}_q^d$ be the input to cSIS. We now explain why $m \geq (n+d)\ell$ suffices to always guarantee a solution to cSIS. First, observe that the first $\ell \cdot d$ columns of $\mathbf{G}$, corresponding to the gadget matrix $[\mathbf{I}_d \otimes \boldsymbol{\gamma}^T]$, are enough to guarantee that for every $\mathbf{r}' \in \mathbb{Z}_q^{m-\ell d}$ there exists an $\mathbf{r}$ such that $\mathbf{G}\begin{bmatrix}\mathbf{r}\\ \mathbf{r}'\end{bmatrix} = \mathbf{b} \pmod q$. Hence, there are at least $q^{m-\ell d}$ solutions to the equation $\mathbf{Gx} = \mathbf{b} \pmod q$. Also, there are $2^{\ell n}$ possible values for $\mathbf{Ax} \pmod q$. By a pigeonhole argument a solution to cSIS always exists. To complete the membership proof, issues similar to BLICHFELDT with the circuit representation of the problem instance appear, but we overcome them using similar ideas.

PPP HARDNESS OF cSIS OVERVIEW. We start with a circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ that is an input to PIGEONHOLE CIRCUIT. Since the input of PIGEONHOLE CIRCUIT is a circuit and the input of cSIS is a pair of matrices and a vector, we need to represent this circuit in an algebraic way. In particular, we device a way to encode the circuit in a binary invertible matrix $\mathbf{G}$ and a vector $\mathbf{b}$. To gain a better intuition of why this is possible, we note that a NAND gate $x \barwedge y = z$ can be expressed as the linear modular equation $x + y + 2z - w = 2 \pmod 4$, where $x, y, z, w \in \{0,1\}$. By a very careful construction, we can encode these linear modular equations in a binary invertible matrix $\mathbf{G}$. For further details we defer to the full version of the paper.

Since cSIS with $q = 4$ returns a vector such that $\mathbf{Ax} = \mathbf{0} \pmod 4$ and PIGEONHOLE CIRCUIT asks for a binary vector such that $\underline{\boldsymbol{x}} = \underline{\mathbf{0}}$, a natural idea is to let $\mathbf{A}$ be of the form $[\mathbf{0}\ \mathbf{I}_n]$, where the identity matrix corresponds to the columns representing the output of circuit $\mathcal{C}$ in $\mathbf{G}$. Finally, we argue that a solution to cSIS with input $\mathbf{A}, \mathbf{G}$ and $\mathbf{b}$ as constructed above, gives either a collision or a preimage of zero for the circuit $\mathcal{C}$ as required.

It can be argued that this reduction shares common ideas with the reduction of 3-SAT to SUBSET-SUM; this shows the importance of the input conditions for cSIS and hints to the numerous complications that arise in the proof. Without these conditions, we could end up with a trivial reduction to an NP-hard problem! Fortunately, we are able to show that our construction satisfies the input conditions of cSIS.

We provide an example of our reduction for very simple circuit $\mathcal{C}$ in Figure 2.

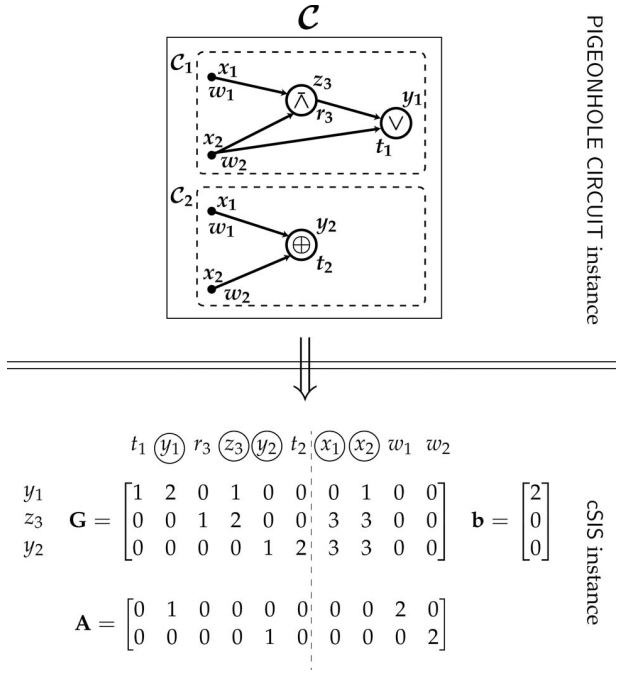

Figure 2: A simple example of the reduction from PPP to cSIS.

C. Towards a Natural and Universal Collision-Resistant Rash Family.

PWPP is a subclass of PPP in which a collision always exists; it is not hard to show that variations of both BLICHFELDT and cSIS are PWPP-complete. We tweak the parameters of valid inputs in order to guarantee that a collision always exists. The PWPP-complete variation of cSIS, which we denote by weak-cSIS, gives a function family which is a universal collision-resistant hash function family in a worst-case sense: if there is a function family that contains at least one function for which it is hard to find collisions, then our function family also includes a function for which it is hard to find collisions.

We now describe the differences of cSIS and weak-cSIS. As before we assume that $q = 2^\ell$. The input to weak-cSIS is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a binary invertible matrix $\mathbf{G} \in \mathbb{Z}_q^{d \times m}$. Notice that there is no vector $\mathbf{b}$ in the input, and the relation between n, m, d and ℓ is that m has to be *strictly* greater that $\ell(n+d)$. Namely, $m > \ell(n+d)$. This change in the relation of the parameters might seem insignificant, but is actually very important, as it allows us to replace $\mathbf{b}$ in cSIS by the zero vector. This transforms weak-cSIS into a pure lattice problem: on input matrices $\mathbf{A}, \mathbf{G}$ with corresponding bases $\mathbf{B_A}$ and $\mathbf{B_G}$, where $\mathbf{G}$ is binary invertible, find two vectors $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{x}, \mathbf{y} \in \mathcal{L}(\mathbf{B_G})$ and $\mathbf{x} - \mathbf{y} \in \mathcal{L}(\mathbf{B_A})$.

The great resemblance of weak-cSIS with SIS and its completeness for PWPP lead us to the first candidate for a universal collision-resistant hash function $\mathcal{H}_{\mathsf{cSIS}} = \{h_{\underline{\boldsymbol{s}}} : \{0,1\}^k \to \{0,1\}^{k'}\}$:

- The key $\underline{\boldsymbol{s}}$ is a pair of matrices $(\mathbf{A}, \mathbf{G})$, where $\mathbf{G}$ is binary invertible.
- Given a key $\underline{\boldsymbol{s}} = (\mathbf{A}, \mathbf{G})$ and a binary vector $\underline{\boldsymbol{x}} \in \{0,1\}^k$, $h_{\underline{\boldsymbol{s}}}(\underline{\boldsymbol{x}})$ is the binary decomposition of $\mathbf{Au} \pmod q$, where $\mathbf{u} = \begin{bmatrix} \mathbf{r} \\ \underline{\boldsymbol{x}} \end{bmatrix}$ such that $\mathbf{Gu} = \mathbf{0} \pmod q$.

Because lattice problems have worst-to-average case reductions and our hash family is based on a lattice problem, this gives hope for showing that our construction is universal in the average-case sense.

D. Other Lattice Problems Known to be in PPP.

We show that the computational analog of Minkowski's theorem, namely MINKOWSKI, is in PPP via a Karp-reduction to BLICHFELDT. We note that a Karp-reduction showing MINKOWSKI $\in$ PPP was shown in [59]. Based on these two problems and the known reductions between lattice problems, we conclude that a variety of lattice (approximation) problems belong to PPP; the most important among them are n-SVP, $\tilde{O}(n)$-SIVP and $n^{2.5}$-CVP (see Figure 3).

E. Other Cryptographic Assumptions in PPP.

By the definition, the class PWPP contains all cryptographic assumptions that imply collision-resistant hash functions. These include the factoring of Blum integers, the Discrete Logarithm problem over $\mathbb{Z}_p^*$ and over elliptic curves, and the SIS lattice problem (a special case of weak-cSIS). Also, Jeřábek [6] showed that the problem of factoring integers is in PWPP.

We extend the connection between PPP and cryptography by introducing a white-box model to describe *general groups*, which we define to be cyclic groups with a succinct representation of their elements and group operation (i.e. a small circuit). We show that the Discrete Logarithm over general groups is in PPP. An example of a general group is $\mathbb{Z}_q^*$. These connections are also summarized in Figure 3.

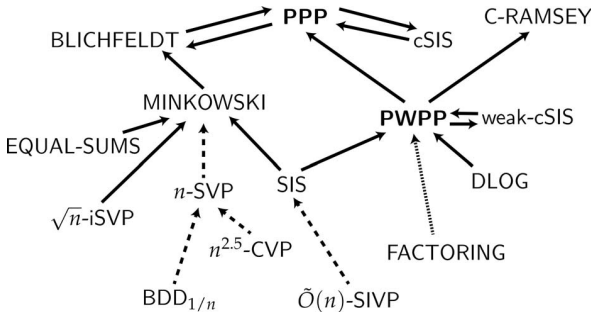

Figure 3: Solid arrows denote a Karp reduction, and dashed arrows denote a Cook reduction.

F. Open questions.

Numerous new questions arise from our work and the connections we draw between PPP, cryptography and lattices. We summarize here some of them.

Open Problem II.7. *Is there a worst-to-average case reduction from* weak-cSIS *to itself?*
This result will provide the first *natural*, in the sense that does not invoke explicitly a Turing machine in the input, and *universal* collision resistant hash function family.

Open Problem II.8. *Is* SIS *or* MINKOWSKI PPP*-hard?*

Open Problem II.9. *Is* γ-SVP *in* PPP *for* $\gamma = o(n)$*?*

Open Problem II.10. *Is* γ-CVP PPP*-hard for* $\gamma = \Omega(\sqrt{n})$*?*

Open Problem II.11. *Is the discrete logarithm problem in* PPP *for general elliptic curves?*

ACKNOWLEDGEMENTS

We thank the anonymous FOCS reviewers for their helpful comments. We thank an anonymous reviewer and Nico Döttling for bringing in our attention a universal CRH following Levin's paradigm. We thank Vinod Vaikuntanathan, Daniel

Wichs and Constantinos Daskalakis for helpful and enlightening discussions. We thank Christos-Alexandros Psomas and his coauthors for sharing their unpublished manuscript [59]. MZ also thanks Christos-Alexandros Psomas and Christos Papadimitriou for many fruitful discussions during his visit to Simons Institute at Berkeley at Fall 2015.

KS was partly supported by NSF grants CNS-1350619, CNS-1718161, CNS-1414119 and by the Chateaubriand Fellowship of the Office for Science and Technology of the Embassy of France in the United States. MZ was supported by NSF grants CCF-1551875, CCF-1617730, CCF-1650733. GZ was supported by NSF grants CNS-1314722, CNS-1413964, CNS-1750795.

REFERENCES

[1] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search?" *Journal of computer and system sciences*, vol. 37, no. 1, pp. 79–100, 1988.

[2] C. H. Papadimitriou, "On the complexity of the parity argument and other inefficient proofs of existence," *Journal of Computer and system Sciences*, vol. 48, no. 3, pp. 498–532, 1994.

[3] M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis, "Exponential lower bounds for finding brouwer fix points," *Journal of Complexity*, vol. 5, no. 4, pp. 379–416, 1989.

[4] N. Meggido and C. Papadimitriou, "A note on total functions, existence theorems, and computational complexity," Tech. report, IBM, Tech. Rep., 1989.

[5] C. Daskalakis and C. Papadimitriou, "Continuous local search," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2011, pp. 790–804.

[6] E. Jerábek, "Integer factoring and modular square roots." *J. Comput. Syst. Sci.*, vol. 82, no. 2, pp. 380–394, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/jcss/jcss82.html#Jerabek16

[7] A. Filos-Ratsikas and P. W. Goldberg, "Consensus halving is ppa-complete," *Proceedings of the 50th annual ACM symposium on Theory of computing (STOC)*, 2018.

[8] A. L. Selman, "A survey of one-way functions in complexity theory," *Mathematical systems theory*, vol. 25, no. 3, pp. 203–221, Sep 1992.

[9] L. A. Levin, "One-way functions and pseudorandom generators," *Combinatorica*, vol. 7, no. 4, pp. 357–363, 1987.

[10] ——, "The tale of one-way functions," *Problems of Information Transmission*, vol. 39, no. 1, pp. 92–103, Jan 2003.

[11] A. A. Kozhevnikov and S. I. Nikolenko, "On complete one-way functions," *Problems of Information Transmission*, vol. 45, no. 2, pp. 168–183, Jun 2009.

[12] D. Aharonov and O. Regev, "Lattice problems in NP cap coNP," in *45th FOCS*. IEEE Computer Society Press, Oct. 2004, pp. 362–371.

[13] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996, pp. 99–108.

[14] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, 1997, pp. 284–293.

[15] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on gaussian measures," *SIAM J. Comput.*, vol. 37, no. 1, pp. 267–302, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1137/S0097539705447360

[16] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[17] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *40th ACM STOC*, R. E. Ladner and C. Dwork, Eds. ACM Press, May 2008, pp. 197–206.

[18] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *41st ACM STOC*, M. Mitzenmacher, Ed. ACM Press, May / Jun. 2009, pp. 333–342.

[19] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," in *45th ACM STOC*, D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds. ACM Press, Jun. 2013, pp. 545–554.

[20] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *45th ACM STOC*, D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds. ACM Press, Jun. 2013, pp. 575–584.

[21] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[22] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *CRYPTO 2013, Part I*, ser. LNCS, R. Canetti and J. A. Garay, Eds., vol. 8042. Springer, Heidelberg, Aug. 2013, pp. 75–92.

[23] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," in *47th ACM STOC*, R. A. Servedio and R. Rubinfeld, Eds. ACM Press, Jun. 2015, pp. 469–477.

[24] R. Goyal, V. Koppula, and B. Waters, "Lockable obfuscation," in *58th FOCS*. IEEE Computer Society Press, 2017, pp. 612–621.

[25] D. Wichs and G. Zirdelis, "Obfuscating compute-and-compare programs under lwe," in *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*. IEEE, 2017, pp. 600–611.

[26] C. Peikert, O. Regev, and N. Stephens-Davidowitz, "Pseudorandomness of ring-LWE for any ring and modulus," in *49th ACM STOC*, H. Hatami, P. McKenzie, and V. King, Eds. ACM Press, Jun. 2017, pp. 461–473.

[27] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1997, pp. 256–266.

[28] P. W. Goldberg and C. H. Papadimitriou, "Towards a unified complexity theory of total functions," *Journal of Computer and System Sciences*, 2017.

[29] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi, "The relative complexity of np search problems," *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 3–19, 1998.

[30] A. A. Schäffer and M. Yannakakis, "Simple local search problems that are hard to solve," *SIAM journal on Computing*, vol. 20, no. 1, pp. 56–87, 1991.

[31] C. H. Papadimitriou, "The complexity of the lin–kernighan heuristic for the traveling salesman problem," *SIAM Journal on Computing*, vol. 21, no. 3, pp. 450–465, 1992.

[32] A. Fabrikant, C. Papadimitriou, and K. Talwar, "The complexity of pure nash equilibria," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 604–612.

[33] M. Etscheid and H. Röglin, "Smoothed analysis of local search for the maximum-cut problem," *ACM Transactions on Algorithms (TALG)*, vol. 13, no. 2, p. 25, 2017.

[34] O. Angel, S. Bubeck, Y. Peres, and F. Wei, "Local max-cut in smoothed polynomial time," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 429–437.

[35] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[36] X. Chen, X. Deng, and S.-H. Teng, "Settling the complexity of computing two-player nash equilibria," *Journal of the ACM (JACM)*, vol. 56, no. 3, p. 14, 2009.

[37] E. Elkind, L. A. Goldberg, and P. Goldberg, "Nash equilibria in graphical games on trees revisited," in *Proceedings of the 7th ACM Conference on Electronic Commerce*. ACM, 2006, pp. 100–109.

[38] X. Chen, D. Dai, Y. Du, and S.-H. Teng, "Settling the complexity of arrow-debreu equilibria in markets with additively separable utilities," in *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*. IEEE, 2009, pp. 273–282.

[39] V. V. Vazirani and M. Yannakakis, "Market equilibrium under separable, piecewise-linear, concave utilities," *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 10, 2011.

[40] S. Kintali, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, "Reducibility among fractional stability problems," *SIAM Journal on Computing*, vol. 42, no. 6, pp. 2063–2113, 2013.

[41] X. Chen, D. Durfee, and A. Orfanou, "On the complexity of nash equilibria in anonymous games," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 2015, pp. 381–390.

[42] A. Rubinstein, "Inapproximability of nash equilibrium," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 2015, pp. 409–418.

[43] ——, "Settling the complexity of computing approximate two-player nash equilibria," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*. IEEE, 2016, pp. 258–265.

[44] X. Chen, D. Paparas, and M. Yannakakis, "The complexity of non-monotone markets," *Journal of the ACM (JACM)*, vol. 64, no. 3, p. 20, 2017.

[45] S. Schuldenzucker, S. Seuken, and S. Battiston, "Finding clearing payments in financial networks with credit default swaps is ppad-complete," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 67. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[46] M. Grigni, "A sperner lemma complete for ppa," *Information Processing Letters*, vol. 77, no. 5-6, pp. 255–259, 2001.

[47] J. Aisenberg, M. L. Bonet, and S. Buss, "2-d tucker is ppa complete." in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 22, 2015, p. 163.

[48] X. Deng, J. R. Edmonds, Z. Feng, Z. Liu, Q. Qi, and Z. Xu, "Understanding ppa-completeness," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[49] A. Belovs, G. Ivanyos, Y. Qiao, M. Santha, and S. Yang, "On the polynomial parity argument complexity of the combinatorial nullstellensatz," in *Proceedings of the 32nd Computational Complexity Conference*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, p. 30.

[50] C. Daskalakis, C. Tzamos, and M. Zampetakis, "A converse to banach's fixed point theorem and its cls completeness," *Proceedings of the 50th annual ACM symposium on Theory of computing (STOC)*, 2018.

[51] J. Fearnley, S. Gordon, R. Mehta, and R. Savani, "Cls: New problems and completeness," *arXiv preprint arXiv:1702.06017*, 2017.

[52] J. Buresh-Oppenheim, "On the tfnp complexity of factoring," *Unpublished manuscript*, 2006, http://www.cs.toronto.edu/~bureshop/factor.pdf.

[53] N. Bitansky, O. Paneth, and A. Rosen, "On the cryptographic hardness of finding a nash equilibrium," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 2015, pp. 1480–1498.

[54] S. Garg, O. Pandey, and A. Srinivasan, "Revisiting the cryptographic hardness of finding a nash equilibrium," in *Annual Cryptology Conference*. Springer, 2016, pp. 579–604.

[55] P. Hubáček and E. Yogev, "Hardness of continuous local search: Query complexity and cryptographic lower bounds," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2017, pp. 1352–1371.

[56] A. Rosen, G. Segev, and I. Shahaf, "Can ppad hardness be based on standard cryptographic assumptions?" in *Theory of Cryptography Conference*. Springer, 2017, pp. 747–776.

[57] P. Hubácek, M. Naor, and E. Yogev, "The journey from np to tfnp hardness," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 67. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[58] I. Komargodski, M. Naor, and E. Yogev, "White-box vs. black-box complexity of search problems: Ramsey and graph property testing," in *58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Canada, 2017.

[59] F. Ban, K. Jain, C. Papadimitriou, C. A. Psomas, and A. Rubinstein, "Reductions in ppp," *Unpublished Manuscript*, 2015.

[60] R. Hoberg, H. Ramadas, T. Rothvoss, and X. Yang, "Number balancing is as hard as minkowski's theorem and shortest vector." in *IPCO*, ser. Lecture Notes in Computer Science, F. Eisenbrand and J. Könemann, Eds., vol. 10328. Springer, 2017, pp. 254–266. [Online]. Available: http://dblp.uni-trier.de/db/conf/ipco/ipco2017.html#HobergRRY17

[61] H. F. Blichfeldt, "A new principle in the geometry of numbers, with some applications," *Transactions of the American Mathematical Society*, vol. 15, no. 3, pp. 227–235, 1914.

[62] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2012, pp. 700–718.

[63] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *EUROCRYPT 2014*, ser. LNCS, P. Q. Nguyen and E. Oswald, Eds., vol. 8441. Springer, Heidelberg, May 2014, pp. 533–556.

[64] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Predicate encryption for circuits from LWE," in *CRYPTO 2015, Part II*, ser. LNCS, R. Gennaro and M. J. B. Robshaw, Eds., vol. 9216. Springer, Heidelberg, Aug. 2015, pp. 503–523.

[65] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *EUROCRYPT 2016, Part II*, ser. LNCS, M. Fischlin and J.-S. Coron, Eds., vol. 9666. Springer, Heidelberg, May 2016, pp. 735–763.

[66] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in *CRYPTO 2016, Part I*, ser. LNCS, M. Robshaw and J. Katz, Eds., vol. 9814. Springer, Heidelberg, Aug. 2016, pp. 190–213.

[67] D. Boneh, S. Kim, and H. W. Montgomery, "Private puncturable PRFs from standard lattice assumptions," in *EUROCRYPT 2017, Part I*, ser. LNCS, J. Coron and J. B. Nielsen, Eds., vol. 10210. Springer, Heidelberg, May 2017, pp. 415–445.

[68] Z. Brakerski, R. Tsabary, V. Vaikuntanathan, and H. Wee, "Private constrained PRFs (and more) from LWE," in *TCC 2017, Part I*, ser. LNCS, Y. Kalai and L. Reyzin, Eds., vol. 10677. Springer, Heidelberg, Nov. 2017, pp. 264–302.

[69] C. Peikert and S. Shiehian, "Privately constraining and programming prfs, the lwe way." in *PKC (2)*, ser. Lecture Notes in Computer Science, M. Abdalla and R. Dahab, Eds., vol. 10770. Springer, 2018, pp. 675–701. [Online]. Available: http://dblp.uni-trier.de/db/conf/pkc/pkc2018-2.html#PeikertS18

[70] O. Goldreich, *Foundations of Cryptography: Volume 1*. New York, NY, USA: Cambridge University Press, 2006.

APPENDIX

We sketch the construction of a universal (average-case hard) hash function, following Levin's paradigm [9] for a universal one-way function. Let h be a hash function family that takes two inputs a key k and a vector $x \in \{0,1\}^n$, and compresses the input x by one bit, i.e. $h(k,x) \in \{0,1\}^{n-1}$. Let $p(\cdot)$ be a polynomial that bounds the running time of $h(k,\cdot)$. First, using padding on the input we argue the existence of a hash function family h' that is defined as

$$h'(k, x \circ y) = h(k,x) \circ y$$

such that $|x \circ y| = p(|x|)$. This implies that $h'(k,\cdot)$ runs in quadratic time in $|x \circ y|$ (see [70, §2.4.1]). Second, using standard domain extension we argue the existence of a hash function family h'' that compresses the input for more than one bits, i.e. $h'' : \{0,1\}^{n'} \to \{0,1\}^{n-1}$ (we exclude the key k from the description of the domain). Specifically, we require that the compressing ratio is enough so that the concatenation of m copies of $h''(k,\cdot)$ is smaller than the input, meaning that $n' > m \cdot (p(n) - 1)$. Let $p''(\cdot)$ be a polynomial that bounds the running time of h''. The hash function $h'(k,\cdot)$ runs in time quadratic to its input, and using this fact, $p''(\cdot)$ can be made explicit depending on the length of the extended domain that we require from h'', once that length is explicitly specified. This is enough to get an upper bound for the running time of $h''(k,\cdot)$.

The universal hash function is described by a collection of $2 \cdot m + 1$ strings:

$$h_{\mathsf{uni}} = (i_1, \ldots, i_m, k_1, \ldots, k_m, x).$$

The numbers $i_1, \ldots, i_m$ (represented as strings) are the indices to Turing Machines (assume a canonical ordering of TMs) that describe m different hash function families $h''_{i_j} : \{0,1\}^{n'} \to \{0,1\}^{n-1}$, $j = 1, \ldots, m$. The keys k_j define the hash functions $h''_{i_j}(k_j,\cdot)$. Finally, for $j = 1, \ldots, m$, we run each $h''_{i_j}(k_j, x)$ for at most $p''(|x|)$ steps and output:

$$h_{\mathsf{uni}}(x) = h''_{i_1}(k_1, x) \circ \cdots \circ h''_{i_m}(k_m, x).$$

If $h''_{i_j}(k_j, x)$ does not terminate after $p''(|x|)$, we output $\perp$ for that j. We can see that if at least one hash function family h''_{i_j} is collision-resistant, then so is h_{uni}. At a high level, if

at least one hash function family h_{i_j} is collision-resistant then so is h'_{i_j}, and moreover so is h''_{i_j} by the security of domain extension (e.g. Merkle–Damgård). Without loss of generality we assume that all families h''_{i_j} are defined on the same domain and range. Finally, the simple hash function combiner in which we concatenate the output of m different hash functions $h''_{i_j}(k_j, \cdot)$, is collision-resistant as long as at least one hash function family h''_{i_j} is collision-resistant.