Theoretical Computer Science 795 (2019) 204–212

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The Hidden Subgroup Problem and MKTP

Nicollas M. Sdroievski *, Murilo V.G. da Silva, André L. Vignatti

Department of Computer Science, Federal University of Paraná, 81531-980, Curitiba, Brazil

ARTICLE INFO

Article history:
Received 11 December 2018
Received in revised form 17 April 2019
Accepted 17 June 2019
Available online 2 July 2019
Communicated by P. Lu

Keywords:
Hidden Subgroup Problem
NP-intermediate problems
Statistical zero-knowledge
Time-bounded Kolmogorov complexity

ABSTRACT

We show that the Hidden Subgroup Problem for group families where products and inverses can be computed efficiently is in $\mathsf{BPP}^{\mathsf{MKTP}}$ (where MKTP is the Minimum KT Problem) using the techniques of Allender et al. (2018) [1]. We also show that the problem is in $\mathsf{ZPP}^{\mathsf{MKTP}}$ provided that there is a *pac overestimator* computable in $\mathsf{ZPP}^{\mathsf{MKTP}}$ for the logarithm of the order of the input group. This last result implies that for permutation groups, the dihedral group and many types of matrix groups the problem is in $\mathsf{ZPP}^{\mathsf{MKTP}}$. Lastly, we also show that two decision versions of the problem admit statistical zero knowledge proofs. These results help classify the relative difficulty of the Hidden Subgroup Problem.

1. Introduction

Ladner [2] showed that, assuming $\mathsf{P} \neq \mathsf{NP}$, there exist NP-intermediate problems, that is, problems in NP that are neither in P nor NP-complete. While the problems shown to be NP-intermediate in [2] are quite artificial, there are some "natural" candidates such as the Graph Isomorphism and Integer Factorization problems. In this paper we present results that relate two such candidates, the Hidden Subgroup Problem and the Minimum KT Problem.

The Hidden Subgroup Problem (HSP) is a well known candidate for **NP**-intermediate status. Many current cryptographic protocols rely on the hardness of HSP since both the Integer Factorization and Discrete Logarithm problems are reducible to the problem. It was shown that Shor's polynomial time quantum algorithm for Integer Factorization [3] also applies to the Abelian HSP [4]. Consequently, there is an interest in improving upper bounds for the Non-abelian HSP in the quantum computation model, especially in permutation and dihedral groups [5,6]. However, no polynomial time quantum algorithm for the general HSP is known.

The Minimum KT Problem (MKTP) is also an NP-intermediate candidate problem. It is closely related to the Minimum Circuit Size Problem (MCSP) [7], a very well studied problem. Many hardness results for MKTP are known: the class $\mathsf{BPP}^{\mathsf{MKTP}}$ contains the entirety of SZK [8] and the Graph Isomorphism, Integer Factorization and Discrete Logarithm problems are all in $\mathsf{ZPP}^{\mathsf{MKTP}}$ [1,9,10]. As the last three problems are generalized by HSP, it is natural to wonder if HSP itself is in $\mathsf{ZPP}^{\mathsf{MKTP}}$.

Most of the hardness results shown for MKTP also apply to MCSP. Recently, however, [1] presented a new reduction technique that, up until now, applies only to MKTP. They showed that Graph Isomorphism (GI) and a variety of "Isomorphism Problems" are in $\mathsf{ZPP}^{\mathsf{MKTP}}$. The result for GI is unconditional, while other problems must respect some basic conditions on efficiency and samplability. In doing so, they developed many technical results that we use in this work.

Using their techniques, we present two results relating HSP and MKTP. The first one is a direct adaptation of [1, Lemma 5.4]. In this Lemma the authors show how to obtain a list of elements that with high probability generate the stabilizer of

* Corresponding author.
E-mail addresses: nmsdroievski@inf.ufpr.br (N.M. Sdroievski), murilo@inf.ufpr.br (M.V.G. da Silva), vignatti@inf.ufpr.br (A.L. Vignatti).

https://doi.org/10.1016/j.tcs.2019.06.012
0304-3975/

an action point. We note that the result only depends on the fact that a group action with a fixed action point *hides* the stabilizer subgroup, so it can be adapted to find any hidden subgroup, showing that $\mathsf{HSP} \in \mathsf{BPP}^{\mathsf{MKTP}}$, providing operations on the underlying group can be computed efficiently. The second, and main result of this paper, is a strengthening of the first, where we show that having a *pac overestimator* for the logarithm of the order of the input group is enough to imply that $\mathsf{HSP} \in \mathsf{ZPP}^{\mathsf{MKTP}}$. This implies that HSP for the dihedral group, permutation groups and many types of matrix groups is in $\mathsf{ZPP}^{\mathsf{MKTP}}$. Note that, in the previous discussion, we assume the group is given as a list of generators for permutation and matrix groups, while for the dihedral group we assume the group D_N is given by N, a reflection and a rotation.

We also present two results relating decision versions of HSP with SZK, the class of problems admitting statistical zero knowledge proofs [11]. SZK is conjectured to be strictly contained in NP, while also containing hard problems. Specifically, we prove that the problem of deciding whether the hidden subgroup is trivial or not is in HVPZK, and that a gap version of the problem for permutation groups is in NISZK. Both HVPZK and NISZK are subclasses of SZK.

2. Preliminaries

We assume familiarity with the standard complexity classes, including probabilistic polynomial classes like BPP (two-sided error), RP (one sided error) and ZPP (zero-sided error), as well as interactive proofs. We also refer the reader to the standard texts about general group theory for basic definitions [12].

In this section we provide more details about KT complexity and the problem MKTP, zero knowledge proofs and our group model, besides defining the computational problems of interest. We also provide definitions of various important statistical concepts and restate some results from [1] in an effort to make this paper self-contained.

2.1. KT complexity and MKTP

KT-complexity is a time-bounded variant of Kolmogorov Complexity. We refer the reader to [9] for more details about KT and present only definitions and results that are relevant to our results.

Definition 2.1. Let U be a universal Turing machine. For each string x, define $\mathsf{KT}_U(x)$ to be

$$\min\{|d| + T : (\forall \sigma \in \{0,1,*\})\ (\forall i \leq |x|+1)\ U^d(i,\sigma) \text{ accepts in } T \text{ steps iff } x_i = \sigma\}.$$

We define $x_i = *$ if $i > |x|$; thus, for $i = |x|+1$ the machine accepts iff $\sigma = *$. The notation U^d indicates that the machine U has random access to the description d.

$\mathsf{KT}(x)$ is defined to be equal to $\mathsf{KT}_U(x)$ for a fixed choice of Universal machine U with logarithmic simulation time overhead. The Minimum KT Problem is defined as $\mathsf{MKTP} = \{(x,\theta) \mid \mathsf{KT}(x) \leq \theta\}$. An oracle for MKTP is sufficient to invert on average any function that can be computed efficiently [9]. We present the following formulation due to [1].

Lemma 2.1. *(follows from [9, Theorem 45]) There exists a polynomial-time probabilistic Turing machine using oracle access to MKTP so that the following holds. For any circuit C on n input bits,*

$$\Pr[C(M(C, C(\sigma))) = C(\sigma)] \geq 1/\mathsf{poly}(n)$$

where the probability is over the uniform distribution of $\sigma \in \{0,1\}^n$ and the internal coin flips of M.

2.2. Random variables and samplers

We restate some basic definitions from [1]. A *finite probability space* consists of a finite sample space S and a probability distribution p on S. A *random variable* R is a mapping from the sample space S to a set T. The random variable R with the uniform distribution on S induces a distribution p on T. R may also be used to designate this distribution.

The *support* of a distribution p on a set T is the set $\{t \in T \mid p(t) > 0\}$. A distribution is *flat* if it is uniform on its support. The *entropy* of a distribution p, denoted by $\mathrm{H}(p)$, is the expected value of $\log(1/p(t))$. The *min-entropy* of p is the largest real s such that $p(t) \leq 2^{-s}$ for every $t \in T$. The *max-entropy* of p is the least real s such that $p(t) \geq 2^{-s}$ for every $t \in T$. Note that the entropy is always between the min- and max-entropies. For a flat distribution all of these coincide and equal the logarithm of the size of the support. For two distributions p and q on the same set T, we say that q approximates p within a factor $1+\delta$ if $q(t)/(1+\delta) \leq p(t) \leq (1+\delta)q(t)$ for all $t \in T$. In that case, p and q have the same support, and if p has min-entropy s, then q has min-entropy at least $s - \log(1+\delta)$, and if p has max-entropy s, then q has max-entropy at most $s + \log(1+\delta)$.

A *sampler* within a factor of $1+\delta$ for a distribution p on a set T is a random variable $R : \{0,1\}^\ell \to T$ that induces a distribution on T that approximates p within a factor $1+\delta$. We say that R *samples T within a factor $1+\delta$ from length ℓ*. The choice of $\{0,1\}^\ell$ reflects the fact that distributions need to be generated from a source of random bits.

We consider ensembles of distributions $\{p_x\}$ where $x \in \{0,1\}^*$. We call the ensemble *samplable by polynomial-size circuits* if there exists an ensemble of random variables $\{R_{x,\delta}\}$ where δ ranges over the positive rationals such that $R_{x,\delta}$ samples p_x

within a factor $1+\delta$ from length $\ell_{x,\delta}$ and $R_{x,\delta}$ can be computed by a circuit of size $\mathsf{poly}(|x|/\delta)$. If in addition the mappings $(x,\delta)\mapsto \ell_{x,\delta}$ and $(x,\delta,\sigma)\mapsto R_{x,\delta}(\sigma)$ can be computed in time $\mathsf{poly}((|x|/\delta))$, we call the ensemble *uniformly samplable in polynomial time*.

2.3. Pac estimators and KT

We present the concept of a Probably-Approximately-Correct Overestimator.

Definition 2.2. [1] (Probably-Approximately-Correct Overestimator) Let $g:\Omega\to\mathbb{R}$ be a function and M a randomized algorithm that, on input $\omega\in\Omega$, outputs a value $M(\omega)\in\mathbb{R}$. We say that M is a probably-approximately-correct overestimator for g with deviation Δ if, for every $\omega\in\Omega$, $|M(\omega)-g(\omega)|\leq\Delta$ holds with probability at least $1/\mathsf{poly}(|\omega|)$ and $M(\omega)>g(\omega)$ otherwise. We can define a probably-approximately-correct underestimator by reversing the last inequality.

By taking the minimum (maximum) value of a polynomial number of evaluations of a pac overestimator (underestimator) we are able to increase its confidence to be exponentially close to 1.

A major contribution of [1] is the Entropy Estimator Corollary, which shows that the amortized value $\mathsf{KT}(y)/t$, where y is the concatenation of t samples from a random variable R, is a pac underestimator for the entropy of R.

Corollary 2.1. *[1] (Entropy Estimator Corollary) Let $\{p_x\}$ be an ensemble of distributions such that p_x is supported on strings of the same length $\mathsf{poly}(|x|)$. Consider a randomized process that on input x computes $\mathsf{KT}(y)/t$, where y is the concatenation of t independent samples from p_x. If p_x is samplable by circuits of polynomial size, then for t a sufficiently large polynomial in $|x|$, $\mathsf{KT}(y)/t$ is a pac underestimator for the entropy of p_x with deviation $\Delta(x)+o(1)$, where $\Delta(x)$ is the difference between the min- and max-entropies of p_x.*

2.4. Statistical zero knowledge

Zero knowledge proofs were introduced by [13]. We say that an interactive proof is zero knowledge when the verifier gets no information other than the validity of the assertion being claimed by the prover. We refer the reader to [11] for a complete treatment on the subject.

Problems admitting zero knowledge proofs are better defined as promise problems.

Definition 2.3. A promise problem Π is a pair (Π_Y,Π_N) of two disjoint sets $\Pi_Y,\Pi_N\in\{0,1\}^*$. The set Π_Y contains the "yes instances" and the set Π_N contains the "no instances".

When designing an algorithm for a promise problem Π we are only interested in inputs in $\Pi_Y\cup\Pi_N$. As a consequence, there are no guarantees about the algorithm's behavior on inputs outside of this set.

Let (P,V) be an interactive protocol. Define the verifier's view $\langle P,V\rangle(x)$ of the interaction between P and V on a common input x as all messages exchanged between P and V, together with the random bits used by V. Note $\langle P,V\rangle(x)$ is a random variable. Since we only present an *honest verifier perfect zero knowledge protocol*, we define this notion.

Definition 2.4. Let (P,V) be an interactive protocol and Π a promise problem. We say (P,V) is an *interactive proof* for Π if the following conditions hold:

1. (Efficiency) V is computable in polynomial time. Also, on common input x, the number of messages exchanged between P and V, as well as the message size is at most $\mathsf{poly}(|x|)$.
2. (Completeness) If $x\in\Pi_Y$, then V accepts in $(P,V)(x)$ with probability at least $2/3$.
3. (Soundness) If $x\in\Pi_N$, then for any P^*, V rejects in $(P,V^*)(x)$ with probability at least $2/3$.

We say that (P,V) is an *honest verifier perfect zero knowledge proof* if, in addition to conditions 1 to 3, there is a probabilistic polynomial time simulator S such that for all $x\in\Pi_Y$ the following two conditions hold:

4. On input x, the simulator S outputs *fail* with probability at most $1/2$.
5. Let $\tilde{S}(x)$ be the random variable describing the distribution of $S(x)$ conditioned on S not failing. Then $\tilde{S}(x)$ and $\langle P,V\rangle(x)$ are identically distributed.

HVPZK is the class of promise problems that admit honest verifier perfect zero knowledge proofs. Note that HVPZK $\subseteq$ SZK [11].

NISZK is a subclass of SZK that contains problems for which there are *non-interactive statistical zero knowledge proofs*. In this paper, instead of explicitly presenting a non-interactive protocol for a decision version of HSP, we show a reduction to the complete problem Entropy Approximation [11].

Definition 2.5. Entropy Approximation is the problem $\mathrm{EA} = (\mathrm{EA}_Y, \mathrm{EA}_N)$ where

$$\mathrm{EA}_Y = \{(C, t) \mid \mathrm{H}(C) \geq t + 1\}$$

$$\mathrm{EA}_N = \{(C, t) \mid \mathrm{H}(C) \leq t - 1\}$$

Above, C is a circuit encoding a probability distribution and t is an integer.

2.5. The group model

We study the Hidden Subgroup Problem in a context similar to that of black-box groups. The black-box group model was introduced by Babai [14] and has since been widely used to study algorithmic problems in finite groups [15–17]. We present definitions that are similar to the ones present in [18].

A *group family* is a countable sequence $\mathcal{B} = \{B_n\}_{n \geq 1}$ of finite groups such that elements of each B_n are uniquely represented by strings of size $\mathrm{poly}(n)$ and the order of each B_n is computable in time $\mathrm{poly}(n)$, both for a fixed polynomial on n. The inverse, product and identity testing operations of B_n can be computed in $\mathrm{poly}(n)$-time. We let e denote the identity element of any B_n. The inputs for problems in this model are subsets $T \subset B_n$, and we are interested in the group $\langle T \rangle$ (the subgroup generated by T).

Note that our model is less general than the original black-box group model since we only consider group families where group operations can effectively be computed in polynomial time. Still, the model captures most group families where the Hidden Subgroup Problem is of interest (for instance, permutation, dihedral and matrix groups).

It is not clear if it is possible to efficiently obtain uniform samples of a group $\langle T \rangle$ given $T \subset B_n$, as is the case with permutation groups [19]. However, a fundamental result about black-box groups, based on Babai's seminal work [15, Theorem 1], is that they are uniformly samplable within a factor of $1 + \delta$. Due to the closure of polynomial time under composition, this result also applies to our setting.

Claim 2.1. *(follows from [1, Claim 5.6]) Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$, $T \subset B_n$ and $p_{0^n,T}$ be the uniform distribution on $\langle T \rangle$. The ensemble $\{p_{0^n,T}\}$ is uniformly samplable in polynomial time.*

Combining Claim 2.1 with the Entropy Estimator Corollary, Allender et al. [1] also show that it is possible to pac underestimate the logarithm of the order of a group given by a list of generators in probabilistic polynomial time with oracle access to MKTP.

Lemma 2.2. *(follows from [1, Lemma 5.5]) Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$, $T \subset B_n$ and $G = \langle T \rangle$. The map $(0^n, T) \mapsto \log |G|$ can be pac underestimated with any constant deviation $\Delta > 0$ in $\mathsf{ZPP}^{\mathrm{MKTP}}$.*

2.6. The Hidden Subgroup Problem

In order to define the Hidden Subgroup Problem, we first define what it means for a function f to *hide* a subgroup.

Definition 2.6. Let G be a group and H a subgroup of G. We say that a function f *hides* H in G if for all $g_1, g_2 \in G$, $f(g_1) = f(g_2) \iff g_1 H = g_2 H$.

That is, for a function to hide a subgroup H in G it has to be constant for the elements of G that are in the same coset of H on G, while being different for elements in different cosets. We now formally define the Hidden Subgroup Problem for a specific group family $\mathcal{B}$.

Definition 2.7. The Hidden Subgroup Problem for group family $\mathcal{B}$ (HSP-$\mathcal{B}$).

Input: $(0^n, T, C_f)$, where $T \subset B_n$ for $B_n \in \mathcal{B}$, $G = \langle T \rangle$ and C_f is a $\mathrm{poly}(n)$-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G.

Output: a list of generators for H.

Note that function f is input as a $\mathrm{poly}(n)$-size circuit C_f. Although one could define the problem without a restriction on the size of C_f, we are usually interested in the case where function f is not too hard to compute. In fact, that is the case for most instantiations of the Hidden Subgroup Problem, and allows, for example, the generalization of the Graph Isomorphism, Integer Factorization and Discrete Logarithm problems to HSP-$\mathcal{B}$ for the corresponding group family.

We also define two decision versions of HSP-$\mathcal{B}$ in the form of promise problems, dHSP-$\mathcal{B}$, and GapHSP-$\mathcal{B}$.

Definition 2.8. Let $\mathcal{B}$ be a group family. dHSP-$\mathcal{B}$ is the following promise problem.

$$\text{dHSP-}\mathcal{B}_Y = \left\{(0^n, T, C_f) \mid |H| = 1\right\}$$
$$\text{dHSP-}\mathcal{B}_N = \left\{(0^n, T, C_f) \mid |H| \geq 2\right\},$$

where $T \subset B_n$ for $B_n \in \mathcal{B}$, $G = \langle T \rangle$ and C_f is a $\mathrm{poly}(n)$-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G.

While dHSP-$\mathcal{B}$ is a somewhat easier problem than HSP-$\mathcal{B}$, it captures the difficulty of many problems in groups, such as the Graph Automorphism Problem and the problem of deciding whether the stabilizer of an efficiently computable group action is trivial or not. Moreover, for permutation groups (where $\mathcal{B} = \{S_n\}_{n\geq 1}$) the problem is equivalent to HSP-$\mathcal{B}$ under oracle reductions [20]. It is easy to see that GapHSP-$\mathcal{B}$ generalizes dHSP-$\mathcal{B}$.

Definition 2.9. Let $\mathcal{B}$ be a group family. GapHSP-$\mathcal{B}$ is the following promise problem.

$$\text{GapHSP-}\mathcal{B}_Y = \left\{(0^n, T, C_f, k) \mid |H| \leq k\right\}$$
$$\text{GapHSP-}\mathcal{B}_N = \left\{(0^n, T, C_f, k) \mid |H| \geq 2k\right\},$$

where $T \subset B_n$ for $B_n \in \mathcal{B}$, $G = \langle T \rangle$, C_f is a $\mathrm{poly}(n)$-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G and $k \in \mathbb{N}$.

3. The Hidden Subgroup Problem and MKTP

We show that [1, Lemma 5.4] can be adapted in a rather straightforward way to obtain a $\mathsf{BPP}^{\mathrm{MKTP}}$ algorithm for solving the Hidden Subgroup Problem for a fixed group family. This is possible because the original result only depends on the fact that a group action with a fixed point hides the stabilizer subgroup.

Theorem 3.1. *For all group families $\mathcal{B}$,* HSP-$\mathcal{B} \in \mathsf{BPP}^{\mathrm{MKTP}}$.

Proof. Let $(0^n, T, C_f)$ be an instance of HSP-$\mathcal{B}$, where $T \subset B_n$ for $B_n \in \mathcal{B}$, $G = \langle T \rangle$ and C_f is a $\mathrm{poly}(n)$-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G. We argue that the uniform distribution on H, which we denote by p_H, is uniformly samplable in polynomial time using a oracle for MKTP.

Claim 3.1. *p_H is uniformly samplable in polynomial time with oracle access to* MKTP.

Proof. Let M be the Turing machine from Lemma 2.1 and let p_G denote the uniform distribution of the elements of G.

By Claim 2.1 there is a circuit $C_{G,\delta}$ that samples p_G within a factor of $(1+\delta)$ from strings σ of length $\ell = \mathrm{poly}(n/\delta)$. Let $C_{f,\delta} = C_f \circ C_{G,\delta}$. Note that $C_{f,\delta}$ uniformly samples the image of f within a factor of $1+\delta$. We sample $\sigma \in \{0,1\}^\ell$ uniformly at random and compute $\tau = M(C_{f,\delta}, C_{f,\delta}(\sigma))$. Let $g = C_{G,\delta}(\sigma)$ and $g' = C_{G,\delta}(\tau)$. In case the inversion performed by machine M is successful we have that $f(g) = f(g')$ and then $g^{-1}g' \in H$. Since g is uniform within a factor of $1+\delta$, conditioned on the success of inverting $f(g)$, $g^{-1}g'$ is uniform on H within a factor of $1+\delta$. The probability of success is $1/\mathrm{poly}(n/\delta)$.

We run this procedure many times and retain the value $g^{-1}g'$ of the first successful run. A Chernoff bound guarantees that the probability of obtaining one success in $\mathrm{poly}(n/\delta)$ many runs is exponentially close to 1. Since each run takes time $\mathrm{poly}(n/\delta)$ and success can be determined by evaluating $f(g')$ and $f(g)$ in polynomial time, it follows that the uniform distribution on H is uniformly samplable in polynomial time with access to a MKTP oracle. □ (claim)

Now it suffices to show that for some constant $\delta > 0$, a polynomial amount of samples $h_1, h_2, \ldots, h_k$ from H are sufficient to generate H with high probability. Denote by Γ_i the subgroup of H generated by $L_i = \{h_1, h_2, \ldots, h_i\}$. For $i < k$, if $\Gamma_i \neq H$ then by Lagrange's Theorem $|\Gamma_i| \leq |H|/2$. Thus, with probability at least $1/2 \cdot 1/(1+\delta)$ we have that $h_{i+1} \notin \Gamma_i$, in which case $|\Gamma_{i+1}| \geq 2|\Gamma_i|$. It follows that a value $k = O(\mathrm{poly}(n))$ suffices to guarantee that $\Gamma_k = H$ with probability exponentially close to 1. □

It is possible to improve this reduction to a ZPP reduction by finding a way to certify that the partial list of elements L_i actually generates H. We show how to achieve that by combining the pac-underestimator of Claim 2.2 with a pac-overestimator for the map $(0^n, T, C_f) \mapsto \log|H|$ that is computable in $\mathsf{ZPP}^{\mathrm{MKTP}}$.

Theorem 3.2. *Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$, $T \subset B_n$, $G = \langle T \rangle$ and C_f a* poly(n)*-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G. If there is a pac overestimator with deviation $\Delta = 1/8$ for the map $(0^n, T, C_f) \mapsto \log|H|$ that is computable in* $\mathsf{ZPP}^{\mathrm{MKTP}}$*, then* HSP-$\mathcal{B} \in \mathsf{ZPP}^{\mathrm{MKTP}}$.

Proof. Let $(0^n, T, C_f)$ be an instance of HSP-$\mathcal{B}$, where $T \subset B_n$ for $B_n \in \mathcal{B}$, $G = \langle T \rangle$ and C_f is a poly(n)-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G. By Claim 3.1 we can sample elements from H uniformly within a factor of $1+\delta$. As in Theorem 3.1, we build a list L by gradually adding elements $h_1, h_2, \ldots$ of H to it. What remains is certifying that the list L generates H before returning it.

Let M_{over} be the pac overestimator from the theorem's condition and M_{under} the pac underestimator from Lemma 2.2 with deviation $\Delta = 1/8$. Note that M_{over} pac overestimates the value of $\log|H|$ while M_{under} pac underestimates the value of $\log|\langle L\rangle|$, where $L \subset B_n$.

Let $L_i = \{h_1, h_2, \ldots, h_i\}$ be the list obtained after sampling i elements from H and $\Gamma_i = \langle L_i \rangle$. The algorithm computes, at each step i, $\theta_{L_i} = M_{under}(0^n, L_i)$ and $\theta_{H_i} = M_{over}(0^n, T, C_f)$, then makes $\theta_H = \min_{j \le i}\{\theta_{H_j}\}$. It then tests if $|\theta_{L_i} - \theta_H| \le 1/4$, and in this case it returns $L = L_i$. If the test fails, the algorithm keeps running.

Let $s = \log|H|$. Note that we always have $\theta_H \ge s - 1/8$, and we have $\theta_H \le s + 1/8$ with high probability at each step i. We argue that if for step i, $|\theta_{L_i} - \theta_H| \le 1/4$, then $\langle L_i \rangle = H$. Assume $\langle L_i \rangle \ne H$, then $|\langle L_i \rangle| \le |H|/2$ by Lagrange's Theorem, and thus $\theta_{L_i} \le s - 7/8$ and $|\theta_{L_i} - \theta_H| > 1/4$. Now assume $\langle L_i \rangle = H$, in this case we have that with high probability $\theta_{L_i} \ge s - 1/8$ and $\theta_H \le s + 1/8$. In this case $|\theta_{L_i} - \theta_H| \le 1/4$.

Together with the fact that both M_{over} and M_{under} output results that are within their deviation with probability exponentially close to 1, a similar argument to that of Theorem 3.1 shows that the expected running time of the algorithm is polynomial in n. □

While the condition of Theorem 3.2 may seem strong, we show in the next section that it can be relaxed to that of a pac overestimator for the order of a group given by a list of generators.

3.1. A pac overestimator for the order of the hidden subgroup

Along the lines of [1, Section 5.2], we show how to obtain a pac overestimator for the logarithm of the order of the hidden subgroup H. Note that the order of H is $|G|/[G:H]$, where $[G:H]$ is the *index* of H in G. Note also that the size of the image of a function f that hides H in G is precisely $[G:H]$. In this case to pac overestimate $\log|H| = \log|G| - \log[G:H]$ it suffices to use the following approach:

1. Pac overestimate $\log|G|$ with deviation $1/16$.
2. Pac underestimate $\log[G:H]$ with deviation $1/16$.
3. Return the result of step 1 minus the result of step 2. This gives a pac overestimator for $\log|H|$ with deviation $1/8$.

To achieve step 2, we present in Claim 3.2 a generic procedure to pac underestimate the value of $\log[G:H]$ using a MKTP oracle.

Claim 3.2. *Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$, $T \subset B_n$, $G = \langle T \rangle$ and C_f a* poly(n)*-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G. The map $(0^n, T, C_f) \mapsto \log[G:H]$ can be pac underestimated with any constant deviation $\Delta > 0$ in* $\mathsf{ZPP}^{\mathrm{MKTP}}$.

Proof. Let R_f be the random variable that maps a uniform sample of $g \in G$ to $f(g)$. Note that the entropy of R_f is $\log[G:H]$. Since f is computable in polynomial time and G is uniformly samplable in polynomial time, it follows that R_f is uniformly samplable in polynomial time. Let $R_{f,\delta}$ for a $\delta > 0$ to be defined later be the random variable that samples R_f within a factor of $1+\delta$ from strings of length poly(n/δ). Note that the difference between the max- and min-entropies of $R_{f,\delta}$ is at most $2\log(1+\delta)$.

Let $M_{f,\delta} = \mathsf{KT}(y)/t$, where y is the concatenation of t samples from $R_{f,\delta}$. $M_{f,\delta}$ is computable in $\mathsf{ZPP}^{\mathrm{MKTP}}$ since it is possible to compute the value of $\mathsf{KT}(y)$ in $\mathsf{P}^{\mathrm{MKTP}}$. By the Entropy Estimator Corollary, for a sufficiently large polynomial t we have that $M_{f,\delta}$ is a pac underestimator for the entropy of $R_{f,\delta}$ with deviation $2\log(1+\delta) + o(1)$, and thus a pac underestimator for the entropy of R_f with deviation $3\log(1+\delta) + o(1)$. By picking a value of δ such that $3\log(1+\delta) < \Delta$, it follows that $M_{f,\delta}$ is a pac underestimator for the map $(0^n, T, C_f) \mapsto \log[G:H]$ with deviation Δ that is computable in $\mathsf{ZPP}^{\mathrm{MKTP}}$. □

From the results of Theorem 3.2 and Claim 3.2 we obtain Corollaries 3.1 and 3.2.

Corollary 3.1. *Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$ and $T \subset B_n$. If there is a pac overestimator for the map $(0^n, T) \mapsto \log|\langle T\rangle|$ with deviation $\Delta = 1/16$ that is computable in $\mathsf{ZPP}^{\mathrm{MKTP}}$, then HSP-$\mathcal{B} \in \mathsf{ZPP}^{\mathrm{MKTP}}$.*

Corollary 3.2. *For $\mathcal{B} = \{S_n\}_{n\geq 1}$ (permutation groups),* HSP-$\mathcal{B} \in \mathsf{ZPP}^{\mathrm{MKTP}}$.

Corollary 3.2 also applies to many cases where $\mathcal{B} = \{\mathrm{GL_n}(\mathbb{F}(q))\}_{n\geq 1}$ as noted by [1]. We also note that if an instance of HSP-$\mathcal{B}$ has $G = B_n$, because the order of B_n is computable in polynomial time, then this instance can also be solved in $\mathsf{ZPP}^{\mathrm{MKTP}}$. This implies, for example, that the Dihedral Hidden Subgroup Problem is in $\mathsf{ZPP}^{\mathrm{MKTP}}$. It also provides alternate proofs that Integer Factorization and Discrete Logarithm are in $\mathsf{ZPP}^{\mathrm{MKTP}}$ [9,1,10],

3.2. Pac overestimators for group order

Pac overestimators with MKTP oracles for the logarithm of the order of groups given as a list of generators are a requirement for obtaining ZPP-reductions to MKTP not only from HSP-$\mathcal{B}$, but also from a variety of group problems [1]. We present a simple result that partially solves this problem for cyclic groups.

The following is a well known result according to [21].

Claim 3.3. *Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$, $T \subset B_n$ and $G = \langle T\rangle$. Given the prime factorization of $|G|$, it is possible to determine the order of any element of G in polynomial time.*

We can combine this claim with the fact that factoring integers can be done in $\mathsf{ZPP}^{\mathrm{MKTP}}$ [9] to obtain Theorem 3.3.

Theorem 3.3. *Let* CHSP-$\mathcal{B}$ *be the problem* HSP-$\mathcal{B}$ *with the additional promise that the set T contains only one element. Then for all group families $\mathcal{B}$,* CHSP-$\mathcal{B} \in \mathsf{ZPP}^{\mathrm{MKTP}}$.

Proof. Let $\mathcal{B}$ be a group family, $B_n \in \mathcal{B}$ and $T = \{g\}$, where $g \in B_n$. By Corollary 3.1, it suffices to show that it is possible to compute the order of g in $\mathsf{ZPP}^{\mathrm{MKTP}}$. To do that, first compute $N = |B_n|$ in polynomial time, then obtain the factorization of N in $\mathsf{ZPP}^{\mathrm{MKTP}}$ and finally use Claim 3.3 to compute $|\langle T\rangle| = |g|$. □

It is not clear how to extend this result to compute the order of a generic group given as a list of generators. This is true even if we are promised that the group is cyclic without knowing a generator.

4. The Hidden Subgroup Problem and zero knowledge

In [16, Theorem 15], the authors show that the Group Intersection Problem is in SZK by showing a protocol where the prover works by finding the factorization of a random product of elements from both groups. We generalize this result by showing a protocol for dHSP-$\mathcal{B}$ where the prover works by finding pre-images of f. This implies that the problem is in $\mathsf{HVPZK} \subseteq \mathsf{SZK}$.

Theorem 4.1. *For all group families $\mathcal{B}$,* dHSP-$\mathcal{B} \in \mathsf{HVPZK}$.

Proof. We present an interactive protocol for dHSP-$\mathcal{B}$. Given $(0^n, T, C_f)$, the prover P wants to convince the verifier V that f hides a subgroup H of size 1. Let $G = \langle T\rangle$ and fix a sufficiently small $\delta > 0$.

Protocol 1 for dHSP-$\mathcal{B}$.

1: V: Using Claim 2.1, uniformly (within a factor $1+\delta$) selects $g \in G$, computes $y = f(g)$ and sends y to P.
2: P: Computes $h \in G$ such that $f(h) = y$. Sends h to V.
3: V: Accepts if and only if $h = g$.

We now analyze the protocol. The key observation is that if $|H| = 1$, then there is a single element $h \in G$ such that $f(h) = y$. In this case, P can find this element and make V accept with probability 1. If, on the other hand, $|H| \geq 2$, then there are at least two such h's, and in this case, any P^* cannot make V accept with probability greater than $1/2$. We can make this probability smaller than $1/3$ by repeating the protocol a constant number of times. Completeness and Soundness then follow.

As for the Zero Knowledge property, we first argue that for an honest verifier V, it is possible to construct a simulator S that selects the element g with exactly the same distribution as V, even though this distribution is not exactly uniform. To do that, it suffices for both V and S to sample g using Claim 2.1 with the exact same δ. It is then enough to point that when $|H| = 1$ the distribution of h, the only message sent by P in the protocol, is the same as the distribution of g, so S can just make $h = g$. □

Note that HVPZK $\subseteq$ SZK, and since SZK is closed under complement [11], the following holds.

Corollary 4.1. *For all group families $\mathcal{B}$,* dHSP-$\mathcal{B} \in$ SZK *and* $\overline{\text{dHSP-}\mathcal{B}} \in$ SZK.

Restricted to permutation groups (where $\mathcal{B} = \{S_n\}_{n\geq 1}$), we also show that the GapHSP-$\mathcal{B}$ problem is in NISZK by showing a reduction to the complete problem Entropy Approximation. It may seem unnecessary to consider this gap version of HSP-$\mathcal{B}$ instead of dHSP-$\mathcal{B}$, especially when the latter is as hard as HSP-$\mathcal{B}$ for permutation groups under oracle reductions [20]. However, it is not known whether NISZK is closed under oracle reductions [11], so by considering this gap version we actually present a slightly stronger result. The reduction makes use of Proposition 4.1.

Proposition 4.1. *[11] There is an efficient transformation that takes a triple (C, t_1, t_2), where C is a distribution encoded by a circuit and $t_1 > t_2$ are rational numbers, and produces a new distribution C' and an integer t such that*

$$\mathrm{H}(C) \geq t_1 \rightarrow (C', t) \in \mathrm{EA}_Y$$
$$\mathrm{H}(C) \leq t_2 \rightarrow (C', t) \in \mathrm{EA}_N.$$

The transformation is computable in time polynomial in the input length and $1/(t_1 - t_2)$.

We now present the reduction.

Theorem 4.2. *For $\mathcal{B} = \{S_n\}_{n\geq 1}$ (permutation groups),* GapHSP-$\mathcal{B} \in$ NISZK.

Proof. Let $(0^n, T, C_f, k)$ be an instance of GapHSP-$\mathcal{B}$ with $T \subset S_n$, $G = \langle T \rangle$, C_f is a poly(n)-size circuit that takes as input encodings of group elements of B_n and outputs m-bit strings for some $m \in \mathbb{N}$, with the promise that the function f computed by C_f hides some subgroup H in G and $k \in \mathbb{N}$. Also let $t = |G|$ and note that the value of t is computable in polynomial time as G is a permutation group [19].

Note that, even though G is a permutation group, it does not really follow that we can *exactly* uniformly sample elements from G when we consider that the sampling has to be done from random bits. With this in mind, let $R_{G,\delta}$ be the random variable that samples the uniform distribution on G within a factor of $1 + \delta$ from strings of length poly(n), for a constant $\delta > 0$ to be defined later. Construct the circuit $C_{f,\delta}$ that samples a permutation π from $R_{G,\delta}$ and outputs $f(\pi)$.

If $|H| \leq k$, then $\mathrm{H}(C_{f,\delta}) \geq \log t - \log k - \log(1+\delta)$. If, however, $|H| \geq 2k$, then $\mathrm{H}(C_{f,\delta}) \leq \log t - \log k - 1 + \log(1+\delta)$. Thus, by taking $C = C_{f,\delta}$, $t_1 = \log t - \log k - \log(1+\delta)$ and $t_2 = \log t - \log k + \log(1+\delta) - 1$ in Proposition 4.1, we have that GapHSP-$\mathcal{B} \leq_p$ EA as long as δ is a constant such that $2\log(1+\delta) < 1$. Therefore GapHSP-$\mathcal{B} \in$ NISZK. □

5. Conclusion and open problems

The strongest result we show relating HSP and MKTP requires a pac overestimator for the logarithm of the order of the input group that is computable in $\mathsf{ZPP}^{\mathrm{MKTP}}$. While for some classes of groups there are polynomial time algorithms for computing order using oracles for the Factoring and Discrete Logarithm problems, both of which are in $\mathsf{ZPP}^{\mathrm{MKTP}}$, it remains an open problem to show a general pac overestimator that only depends on group operations being efficiently computable.

Allender et al. [1] show powerful techniques for obtaining zero-sided error reductions to MKTP. While we used these techniques to prove our results, it has been shown that they can also be used to improve known reductions to MKTP. Another line of investigation, as noted in [10], is improving reductions from the Shortest Independent Vector Problem, Unique Shortest Vector Problem, Closest Vector Problem and Covering Radius Problem to MKTP, as all these problems are in $\mathsf{BPP}^{\mathrm{MKTP}}$ [9].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The first author acknowledges a scholarship from the National Council for Scientific and Technological Development (CNPq) grant number 130654/2017-5. We also thank the anonymous referees for the very helpful and insightful suggestions.

References

[1] E. Allender, J. Grochow, D. van Melkebeek, C. Moore, A. Morgan, Minimum circuit size, graph isomorphism, and related problems, SIAM J. Comput. 47 (2018) 1339–1372, https://doi.org/10.1137/17M1157970.

[2] R.E. Ladner, On the structure of polynomial time reducibility, J. ACM 22 (1975) 155–171, https://doi.org/10.1145/321864.321877.
[3] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (1997) 1484–1509, https://doi.org/10.1137/S0097539795293172.
[4] D. Boneh, R.J. Lipton, Quantum cryptanalysis of hidden linear functions, in: Advances in Cryptology — CRYPTO' 95, 1995, pp. 424–437.
[5] J. Kempe, A. Shalev, The hidden subgroup problem and permutation group theory, in: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 1118–1125.
[6] M. Roetteler, Quantum algorithms for abelian difference sets and applications to dihedral hidden subgroups, in: 11th Conference on the Theory of Quantum Computation, Communication and Cryptography, in: Leibniz International Proceedings in Informatics, vol. 61, 2016, pp. 8:1–8:16.
[7] V. Kabanets, J.-Y. Cai, Circuit minimization problem, in: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, 2000, pp. 73–79.
[8] E. Allender, B. Das, Zero knowledge and circuit minimization, Inf. Comput. 256 (2017) 2–8, https://doi.org/10.1016/j.ic.2017.04.004.
[9] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, D. Ronneburger, Power from random strings, SIAM J. Comput. 35 (2006) 1467–1493, https://doi.org/10.1137/050628994.
[10] M. Rudow, Discrete logarithm and minimum circuit size, Inf. Process. Lett. 128 (2017) 1–4, https://doi.org/10.1016/j.ipl.2017.07.005.
[11] S.P. Vadhan, A Study of Statistical Zero-knowledge Proofs, PhD thesis, Massachusetts Institute of Technology, 1999.
[12] J. Rotman, A First Course in Abstract Algebra: With Applications, Pearson Prentice Hall, 2006.
[13] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. 18 (1989) 186–208, https://doi.org/10.1137/0218012.
[14] L. Babai, E. Szemeredi, On the complexity of matrix group problems I, in: Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1984, pp. 229–240.
[15] L. Babai, Local expansion of vertex-transitive graphs and random generation in finite groups, in: Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, 1991, pp. 164–174.
[16] S. Fenner, Y. Zhang, Quantum algorithms for a set of group theoretic problems, Int. J. Found. Comput. Sci. 26 (2015) 255–268, https://doi.org/10.1142/s012905411550015x.
[17] V. Arvind, B. Das, SZK proofs for black-box group problems, Theory Comput. Syst. 43 (2008) 100–117, https://doi.org/10.1007/s00224-007-9028-3.
[18] V. Arvind, V. Vinodchandran, Solvable black-box group problems are low for PP, Theor. Comput. Sci. 180 (1997) 17–45, https://doi.org/10.1016/S0304-3975(96)00100-4.
[19] Á. Seress, Permutation Group Algorithms, Cambridge Tracts in Mathematics, Cambridge University Press, 2003.
[20] S. Fenner, Y. Zhang, On the complexity of the hidden subgroup problem, Int. J. Found. Comput. Sci. 24 (2013) 1221–1234, https://doi.org/10.1142/S0129054113500305.
[21] L. Babai, R. Beals, Á. Seress, Polynomial-time theory of matrix groups, in: Proceedings of the Forty-First Annual ACM Symposium on Theory of computing, 2009, pp. 55–64.