

# The Hidden Subgroup Problem and MKTP

Nicollas M. Sdroievski<sup>\*†</sup>    Murilo V. G. da Silva<sup>\*‡</sup>  
 André L. Vignatti<sup>\*§</sup>

November 14, 2018

## Abstract

We show that the Hidden Subgroup Problem for black-box groups is in  $\text{BPP}^{\text{MKTP}}$  (where MKTP is the Minimum KT Problem) using the techniques of Allender et al [AGvM<sup>+</sup>18]. We also show that the problem is in  $\text{ZPP}^{\text{MKTP}}$  provided that there is a *pac overestimator* computable in  $\text{ZPP}^{\text{MKTP}}$  for the logarithm of the order of the given black-box group. This last result implies that for permutation groups, the dihedral group and many types of matrix groups the problem is in  $\text{ZPP}^{\text{MKTP}}$ . 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 [Lad75] showed that, assuming  $\text{P} \neq \text{NP}$ , then there are NP-intermediate problems, that is, problems that are neither in P nor NP-complete. While the problems shown to be NP-intermediate in [Lad75] 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.

---

<sup>\*</sup>Department of Computer Science, Federal University of Paraná, 81531-980, Curitiba, Brazil.

<sup>†</sup>nmsdroievski@inf.ufpr.br

<sup>‡</sup>murilo@inf.ufpr.br

<sup>§</sup>vignatti@inf.ufpr.br

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 [Sho97] also applies to the Abelian HSP [BL95], and as such, there is an interest in improving upper bounds for the Non-abelian HSP in the quantum computation model, specially in permutation and dihedral groups [KS05, Roe16]. However, no polynomial time quantum algorithm for the general HSP is known.

The Minimum **KT** Problem (MKTP) is also a **NP**-intermediate candidate problem. It is closely related to the Minimum Circuit Size Problem (MCSP) [KC00], a very well studied problem. Many hardness results for MKTP are known: the class  $\text{BPP}^{\text{MKTP}}$  contains the entirety of SZK [AD17] and the Graph Isomorphism, Integer Factorization and Discrete Logarithm problems are all in  $\text{ZPP}^{\text{MKTP}}$  [AGvM<sup>+</sup>18, ABK<sup>+</sup>06, Rud17]. As the last three problems are generalized by HSP, it is natural to wonder if HSP itself is in  $\text{ZPP}^{\text{MKTP}}$ .

Most of the hardness results shown for MKTP also apply to MCSP. Recently, however, [AGvM<sup>+</sup>18] 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  $\text{ZPP}^{\text{MKTP}}$ . 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 such techniques, we present two results relating HSP and MKTP. The first one is a direct adaptation of [AGvM<sup>+</sup>17, Lemma 5]. In this Lemma the authors show how to obtain a list of elements that with high probability generate the stabilizer of 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, and as such it can be adapted to find any hidden subgroup, showing that  $\text{HSP} \in \text{BPP}^{\text{MKTP}}$ . 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 a black-box group is enough to imply that  $\text{HSP} \in \text{ZPP}^{\text{MKTP}}$ . This implies that HSP for the dihedral group, permutation groups and many types of matrix groups is in  $\text{ZPP}^{\text{MKTP}}$ . Note that for permutation and matrix groups we assume the group is given as a list of generators, 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 [Vad99]. 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 [Rot06].

In this section we provide more details about KT complexity and the problem MKTP, zero knowledge proofs and the black-box group model, besides defining the computational problems of interest. We also provide definitions of various important statistical concepts and restate some results from [AGvM<sup>+</sup>18] 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 [ABK<sup>+</sup>06] 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  $\text{KT}_U(x)$  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  $d$ .

$\text{KT}(x)$  is defined to be equal to  $\text{KT}_U(x)$  for a fixed choice of Universal machine  $U$  with logarithmic simulation time overhead. The Minimum KT Problem is defined as  $\text{MKTP} = \{(x, \theta) \mid \text{KT}(x) \leq \theta\}$ . An oracle for MKTP is sufficient to invert on average any function that can be computed efficiently [ABK<sup>+</sup>06]. We present the following formulation due to [AGvM<sup>+</sup>18].

**Lemma 2.1.** (*follows from [ABK<sup>+</sup>06, 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,*

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

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 [AGvM<sup>+</sup>17]. 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  $H(p)$ , 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  $R : \{0, 1\}^\ell \rightarrow T$  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  $\ell$* . 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  $\delta$  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  $\text{poly}(|x|/\delta)$ . 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  $\text{poly}(|x|/\delta)$ , 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.** [AGvM<sup>+</sup>18] (Probably-Approximately-Correct Overestimator) Let  $g : \Omega \rightarrow \mathbb{R}$  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  $\Delta$  if, for every  $\omega \in \Omega$ ,  $|M(\omega) - g(\omega)| \leq \Delta$  holds with probability at least  $1/\text{poly}(|\omega|)$  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 [AGvM<sup>+</sup>18] is the Entropy Estimator Corollary, which shows that the amortized value  $\text{KT}(y)/t$ , 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.** [AGvM<sup>+</sup>17] (**Entropy Estimator Corollary**). Let  $\{p_x\}$  be an ensemble of distributions such that  $p_x$  is supported on string of the same length  $\text{poly}(|x|)$ . Consider a randomized process that on input  $x$  computes  $\text{KT}(y)/t$ , 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  $|x|$ ,  $\text{KT}(y)/t$  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 [GMR89]. 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 [Vad99] for a complete treatment on the subject.

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

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

When designing an algorithm for a promise problem  $\Pi$  we are only interested in inputs in  $\Pi_Y \cup \Pi_N$ . As such, 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 both and the random bits used by  $V$ . Note  $\langle P, V \rangle(x)$  is a random variable. Since we only present a *honest verifier perfect zero knowledge protocol*, we define this notion.

**Definition 2.4.** An interactive proof system  $(P, V)$  for a promise problem  $\Pi$  is said to be *honest verifier perfect zero knowledge* if there is a probabilistic polynomial time simulator  $S$  such that for all  $x \in L$  the following two conditions hold:

1. On input  $x$ , the simulator  $S$  outputs *fail* with probability at most  $1/2$ .
2. 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  $\text{HVPZK} \subseteq \text{SZK}$  [Vad99].

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 [Vad99].

**Definition 2.5.** Entropy Approximation is the problem  $\text{EA} = (\text{EA}_Y, \text{EA}, N)$  where

$$\begin{aligned} \text{EA}_Y &= \{(C, t) \mid \text{H}(C) \geq t + 1\} \\ \text{EA}_N &= \{(C, t) \mid \text{H}(C) \leq t - 1\} \end{aligned}$$

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

## 2.5 Black-Box Groups

We study the Hidden Subgroup Problem in the context of black-box groups. This model was introduced by Babai [BS84] and has since been widely used to study algorithmic problems in finite groups [Bab91, FZ05, AD08]. We present definitions that are similar to the ones present in [AV97].

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  $\text{poly}(n)$  and the order of each  $B_n$  is computable in time  $\text{poly}(n)$ , both for a fixed polynomial on  $n$ . The inverse, product and identity testing operations of  $B_n$  are performed at unit cost by black-boxes. When given a generator set  $T \subset B_n$ , we refer to  $\langle T \rangle$  (the subgroup generated by  $T$ ) as a *black-box group*. For all black-box groups, we let  $e$  denote the identity element.

It is not clear if it is possible to efficiently obtain uniform samples of a black-box group  $\langle T \rangle$ , as is the case of permutation groups [Ser03]. However, a fundamental result about black-box groups, based on Babai's seminal work [Bab91, Theorem 1], is that they are uniformly samplable within a factor of  $1 + \delta$ .

**Claim 2.1.** [AGvM<sup>+</sup>17] *Let  $n \in \mathbb{N}$ ,  $T \subset B_n$  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.*

Using Corollary 2.1, Allender et al [AGvM<sup>+</sup>17] also show that it is possible to pac underestimate the logarithm of the order of a black-box group in probabilistic polynomial time with oracle access to MKTP.

**Lemma 2.2.** (follows from [AGvM<sup>+</sup>17]) *Let  $B_n$  be a group from a black-box group family,  $T \subset B_n$  and  $G = \langle T \rangle$ . The map  $(0^n, T) \mapsto \log |G|$  can be pac underestimated with any constant deviation  $\Delta > 0$  in  $\text{ZPP}^{\text{MKTP}}$ .*

## 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  $g_1, g_2 \in G$ ,  $f(g_1) = f(g_2) \iff g_1H = g_2H$ .

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.

**Definition 2.7.** The Hidden Subgroup Problem (HSP).

Input:  $(0^n, T, f)$ , where  $T \subset B_n$ ,  $G = \langle T \rangle$  and  $f$  is a polynomial time computable function that hides a subgroup  $H$  in  $G$ .

Output: a list of generators for  $H$ .

We also define two decision versions of HSP in the form of promise problems, dHSP, and GapHSP.

**Definition 2.8.** dHSP is the following promise problem.

$$\begin{aligned} \text{dHSP}_Y &= \{(0^n, T, f) \mid |H| = 1\} \\ \text{dHSP}_N &= \{(0^n, T, f) \mid |H| \geq 2\}, \end{aligned}$$

where  $T \subset B_n$ ,  $G = \langle T \rangle$  and  $f$  is a polynomial time computable function that hides a subgroup  $H$  in  $G$ .

While dHSP is an easier problem than HSP, 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 the problem is equivalent to HSP under oracle reductions [FZ08]. It is easy to see that GapHSP generalizes dHSP.

**Definition 2.9.** GapHSP is the following promise problem.

$$\begin{aligned} \text{GapHSP}_Y &= \{(0^n, T, f, k) \mid |H| \leq k\} \\ \text{GapHSP}_N &= \{(0^n, T, f, k) \mid |H| \geq 2k\}, \end{aligned}$$

where  $T \subset B_n$ ,  $G = \langle T \rangle$ ,  $f$  is a polynomial time computable function that hides a subgroup  $H$  in  $G$  and  $k \in \mathbb{N}$ .

### 3 The Hidden Subgroup Problem and MKTP

We show that [AGvM<sup>+</sup>17, Lemma 5] can be adapted in a rather straightforward way to obtain a  $\text{BPP}^{\text{MKTP}}$  algorithm for solving the Hidden Subgroup Problem. 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.**  $\text{HSP} \in \text{BPP}^{\text{MKTP}}$ .

*Proof.* Let  $(0^n, T, f)$  be an instance of HSP, where  $T \subset B_n$ ,  $G = \langle T \rangle$  and  $f$  is a polynomial time computable function that hides a 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 an 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,  $C_f$  be the circuit that computes function  $f$  and  $p_G$  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  $\sigma$  of length  $\ell = \text{poly}(n/\delta)$ . 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  $1/\text{poly}(n/\delta)$ .

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  $\text{poly}(n/\delta)$  many runs is exponentially close to 1. Since each run takes time  $\text{poly}(n/\delta)$  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.  $\square$  (claim)

Now it suffices to show that for some constant  $\delta > 0$ , a polynomial amount of samples  $h_1, h_2, \dots, h_k$  from  $H$  are sufficient to generate  $H$  with high probability. Denote by  $\Gamma_i$  the subgroup of  $H$  generated by  $L_i = \{h_1, h_2, \dots, h_i\}$ . 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  $k = O(\text{poly}(n))$  suffices to guarantee that  $\Gamma_k = H$  with probability exponentially close to 1.  $\square$

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  $(0^n, T, f) \mapsto \log |H|$  that is computable in  $\text{ZPP}^{\text{MKTP}}$ .

**Theorem 3.2.** *Let  $(0^n, T, f)$  be an instance of HSP. If there is a pac overestimator with deviation  $\Delta = 1/8$  for the map  $(0^n, T, f) \mapsto \log |H|$  that is computable in  $\text{ZPP}^{\text{MKTP}}$ , then  $\text{HSP} \in \text{ZPP}^{\text{MKTP}}$ .*

*Proof.* Let  $(0^n, T, f)$  be an instance of HSP where  $T \subset B_n$ ,  $G = \langle T \rangle$  and  $f$  is a polynomial time computable function that hides a subgroup  $H$  in  $G$ . By Claim 3.1 we can sample elements from  $H$  that are uniform within a factor of  $1 + \delta$ . As in Theorem 3.1, we build a list  $L$  by gradually adding elements  $h_1, h_2, \dots$  of  $H$  to it. What remains is certifying that the list  $L$  generates  $H$  before returning it.

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

Let  $L_i = \{h_1, h_2, \dots, h_i\}$  be the list obtained after sampling  $i$  elements from  $H$  and  $\Gamma_i = \langle L_i \rangle$ . The algorithm computes, at each step  $i$ ,  $\theta_{L_i} = M_{\text{under}}(0^n, L_i)$  and  $\theta_{H_i} = M_{\text{over}}(0^n, T, f)$ , then makes  $\theta_H = \min_{j \leq i} \{\theta_{H_j}\}$ . It then tests if  $|\theta_{L_i} - \theta_H| \leq 1/4$ , 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  $\theta_H \geq s - 1/8$ , and we have  $\theta_H \leq s + 1/8$  with high probability at each step  $i$ . We argue that if for step  $i$ ,  $|\theta_{L_i} - \theta_H| \leq 1/4$ , then  $\langle L_i \rangle = H$ . Assume  $\langle L_i \rangle \neq H$ , then  $|\langle L_i \rangle| \leq |H|/2$  by Lagrange's Theorem, and thus  $\theta_{L_i} \leq s - 7/8$  and  $|\theta_{L_i} - \theta_H| > 1/4$ . Now assume  $\langle L_i \rangle = H$ , in this case we have that with high probability  $\theta_{L_i} \geq s - 1/8$  and  $\theta_H \leq s + 1/8$ . In this case  $|\theta_{L_i} - \theta_H| \leq 1/4$ .

Together with the fact that both  $M_{\text{over}}$  and  $M_{\text{under}}$  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  $n$ .  $\square$

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 black-box group.

### 3.1 A pac overestimator for the order of $H$

Along the lines of [AGvM<sup>+</sup>17, 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 show in Claim 3.2 a generic way to pac underestimate the value of  $\log [G : H]$  using a MKTP oracle.

**Claim 3.2.** *Let  $B_n$  be a group from a black-box group family,  $T \subset B_n$ ,  $G = \langle T \rangle$  and  $f$  a polynomial time computable function that hides a subgroup  $H$  in  $G$ . The map  $(0^n, T, f) \mapsto \log [G : H]$  can be pac underestimated with any constant deviation  $\Delta > 0$  in  $\text{ZPP}^{\text{MKTP}}$ .*

*Proof.* Let  $f'$  be the function  $f$  with padded outputs so that every output has the same length  $\text{poly}(n)$ . 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  $\text{poly}(n/\delta)$ . Note that the difference between the max- and min- entropies of  $R_{f,\delta}$  is at most  $2 \log(1 + \delta)$ .

Let  $M_{f,\delta} = \text{KT}(y)/t$ , where  $y$  is the concatenation of  $t$  samples from  $R_{f,\delta}$ .  $M_{f,\delta}$  is computable in  $\text{ZPP}^{\text{MKTP}}$  since it is possible to compute the value of  $\text{KT}(y)$  in  $\text{P}^{\text{MKTP}}$ . 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  $\delta$  such that  $3 \log(1 + \delta) < \Delta$ , it follows that  $M_{f,\delta}$  is a pac underestimator

for the map  $(0^n, T, f) \mapsto \log [G : H]$  with deviation  $\Delta$  that is computable in  $\text{ZPP}^{\text{MKTP}}$ .  $\square$

From these results we obtain Corollaries 3.1 and 3.2.

**Corollary 3.1.** *If there is a pac overestimator for the order of a black-box group with deviation  $\Delta = 1/16$  that is computable in  $\text{ZPP}^{\text{MKTP}}$ , then  $\text{HSP} \in \text{ZPP}^{\text{MKTP}}$ .*

**Corollary 3.2.** *For permutation groups, the Hidden Subgroup Problem is in  $\text{ZPP}^{\text{MKTP}}$ .*

Corollary 3.2 also applies to many cases where  $T \subset \text{GL}_n(\mathbb{F}(q))$  as noted by [AGvM<sup>+</sup>17]. We also note that if an instance of HSP has  $G = B_n$ , because the order of  $B_n$  is computable in polynomial time, then this instance can also be solved in  $\text{ZPP}^{\text{MKTP}}$ . This implies, for example, that the Dihedral Hidden Subgroup Problem is in  $\text{ZPP}^{\text{MKTP}}$ . It also provides alternate proofs that Integer Factorization and Discrete Logarithm are in  $\text{ZPP}^{\text{MKTP}}$  [ABK<sup>+</sup>06, AGvM<sup>+</sup>18, Rud17],

## 3.2 Pac overestimators for black-box group order

Pac overestimators with MKTP oracles for the logarithm of the order of black box groups are a requirement for obtaining ZPP-reductions not only from HSP, but also from a variety of group problems to MKTP [AGvM<sup>+</sup>18]. We present a simple result that partially solves this problem for cyclic groups.

The following is a well known result according to [BBS09].

**Claim 3.3.** *Given a black-box group  $G$  together with 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  $\text{ZPP}^{\text{MKTP}}$  [ABK<sup>+</sup>06] to obtain Theorem 3.3.

**Theorem 3.3.** *Let CHSP be the problem HSP with the additional promise that the set  $T$  contains only one element. Then  $\text{CHSP} \in \text{ZPP}^{\text{MKTP}}$ .*

*Proof.* Let  $T = \{g\}$ , where  $g \in B_n$ . By Corollary 3.1, it suffices to show that it is possible to compute the order of  $g$  in probabilistic polynomial time with access to a MKTP oracle. To do that, first compute  $N = |B_n|$  using the group oracle, then obtain the factorization of  $N$  in  $\text{ZPP}^{\text{MKTP}}$  and finally use Claim 3.3 to compute  $|\langle T \rangle| = |g|$ .  $\square$

It is not clear how to extend this result to compute the order of a generic black-box group. 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 [FZ05, 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 where the prover works by finding pre-images of  $f$ . This implies that the problem is in  $\text{HVPZK} \subseteq \text{SZK}$ .

**Theorem 4.1.**  $\text{dHSP} \in \text{HVPZK}$ .

*Proof.* We present an interactive protocol for dHSP. Given  $(0^n, T, f)$ , 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.

- 1:  $V$ : Using Claim 2.1, uniformly selects  $g \in G$  within a factor  $1 + \delta$ , 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  $f(h) = y$ , and as such,  $P$  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  $1/2$ . Completeness and Soundness then follow.

As for the Zero Knowledge property, we first argue that for a 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  $\delta$ . 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  $h = g$ .  $\square$

Note that  $\text{HVPZK} \subseteq \text{SZK}$ , and since  $\text{SZK}$  is closed under complement [Vad99], the following holds.

**Corollary 4.1.**  $\text{dHSP} \in \text{SZK}$  and  $\overline{\text{dHSP}} \in \text{SZK}$ .

Restricted to permutation groups, we also show that the  $\text{GapHSP}$  problem is in  $\text{NISZK}$  by showing a reduction to the complete problem Entropy Approximation. It may seem unnecessary to consider this gap version of HSP instead of dHSP, specially when the latter is as hard as HSP for permutation groups under oracle reductions. However, it is not known whether  $\text{NISZK}$  is closed under oracle reductions [Vad99], so by considering this gap version we actually present a slightly stronger result. The reduction makes use of Proposition 4.1.

**Proposition 4.1.** [Vad99] *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*

$$\begin{aligned} \text{H}(C) \geq t_1 &\rightarrow (C', t) \in \text{EA}_Y \\ \text{H}(C) \leq t_2 &\rightarrow (C', t) \in \text{EA}_N. \end{aligned}$$

*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 permutation groups,  $\text{GapHSP} \in \text{NISZK}$ .*

*Proof.* Let  $(T, f, k)$  be an instance of  $\text{GapHSP}$  with  $T \subset S_n$ ,  $G = \langle T \rangle$ ,  $f$  a function that hides a 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 [Ser03].

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  $\text{poly}(n)$ , for a constant  $\delta > 0$  to be defined later. Construct the circuit  $C_{f,\delta}$  that samples a permutation  $\pi$  from  $R_{G,\delta}$  and outputs  $f(\pi)$ .

If  $|H| \leq k$ , then  $\text{H}(C_{f,\delta}) \geq \log t - \log k - \log(1 + \delta)$ . If, however,  $|H| \geq 2k$ , then  $\text{H}(C_{f,\delta}) \leq \log t - \log k - 1 + \log(1 + \delta)$ . 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  $\text{GapHSP} \leq_p \text{EA}$  as long as  $\delta$  is a constant such that  $2 \log(1 + \delta) < 1$ . Therefore  $\text{GapHSP} \in \text{NISZK}$ .  $\square$

## 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 a black-box group that is computable in  $\text{ZPP}^{\text{MKTP}}$ . 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  $\text{ZPP}^{\text{MKTP}}$ , it remains an open problem to show a general pac overestimator.

Allender et al [AGvM<sup>+</sup>18] 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 [Rud17], 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  $\text{BPP}^{\text{MKTP}}$  [ABK<sup>+</sup>06].

## 6 Acknowledgements

The first author acknowledges a scholarship from the National Council for Scientific and Technological Development (CNPq).

## References

- [ABK<sup>+</sup>06] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from Random Strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.
- [AD08] V. Arvind and Bireswar Das. SZK Proofs for Black-Box Group Problems. *Theory of Computing Systems*, 43(2):100–117, 2008.
- [AD17] Eric Allender and Bireswar Das. Zero Knowledge and Circuit Minimization. *Information and Computation*, 256:2 – 8, 2017.

- [AGvM<sup>+</sup>17] Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Christopher Moore, and Andrew Morgan. Minimum Circuit Size, Graph Isomorphism, and Related Problems. Technical Report TR17-158, Electronic Colloquium on Computational Complexity (ECCC), 2017.
- [AGvM<sup>+</sup>18] Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Christopher Moore, and Andrew Morgan. Minimum Circuit Size, Graph Isomorphism, and Related Problems. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:20, Dagstuhl, Germany, 2018.
- [AV97] Vikraman Arvind and Variyam Vinodchandran. Solvable black-box group problems are low for pp. *Theoretical Computer Science*, 180(1):17 – 45, 1997.
- [Bab91] László 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*, pages 164–174, 1991.
- [BBS09] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of computing*, pages 55–64, 2009.
- [BL95] Dan Boneh and Richard J. Lipton. Quantum Cryptanalysis of Hidden Linear Functions. In *Advances in Cryptology — CRYPTO’ 95*, pages 424–437, 1995.
- [BS84] L. Babai and E. Szemerédi. On The Complexity of Matrix Group Problems I. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1984*, pages 229–240, 1984.
- [FZ05] Stephen A. Fenner and Yong Zhang. Quantum Algorithms for a Set of Group Theoretic Problems. In *Theoretical Computer Science*, pages 215–227, 2005.



- [FZ08] Stephen Fenner and Yong Zhang. On the Complexity of the Hidden Subgroup Problem. In *Theory and Applications of Models of Computation*, pages 70–81, 2008.
- [GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18:186–208, 1989.
- [KC00] Valentine Kabanets and Jin-Yi Cai. Circuit Minimization Problem. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, pages 73–79, 2000.
- [KS05] Julia Kempe and Aner Shalev. The Hidden Subgroup Problem and Permutation Group Theory. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1118–1125, 2005.
- [Lad75] Richard E. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, 22(1):155–171, 1975.
- [Roe16] Martin 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*, volume 61 of *Leibniz International Proceedings in Informatics*, pages 8:1–8:16, 2016.
- [Rot06] J.J. Rotman. *A First Course in Abstract Algebra: With Applications*. Pearson Prentice Hall, 2006.
- [Rud17] Michael Rudow. Discrete Logarithm and Minimum Circuit Size. *Information Processing Letters*, 128:1 – 4, 2017.
- [Ser03] Á. Seress. *Permutation Group Algorithms*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003.
- [Sho97] Peter W. Shor. Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.
- [Vad99] Salil P. Vadhan. *A Study of Statistical Zero-knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, 1999.