General Impossibility of Group Homomorphic Encryption in the Quantum World

Frederik Armknecht[1], Tommaso Gagliardoni[2,⋆],
Stefan Katzenbeisser[2], and Andreas Peter[3,⋆⋆]

[1] Universität Mannheim, Germany
armknecht@uni-mannheim.de
[2] Technische Universität Darmstadt and CASED, Germany
tommaso.gagliardoni@cased.de
katzenbeisser@seceng.informatik.tu-darmstadt.de
[3] University of Twente, The Netherlands
a.peter@utwente.nl

Abstract. Group homomorphic encryption represents one of the most important building blocks in modern cryptography. It forms the basis of widely-used, more sophisticated primitives, such as CCA2-secure encryption or secure multiparty computation. Unfortunately, recent advances in quantum computation show that many of the existing schemes completely break down once quantum computers reach maturity (mainly due to Shor's algorithm). This leads to the challenge of constructing quantum-resistant group homomorphic cryptosystems.

In this work, we prove the *general* impossibility of (abelian) group homomorphic encryption in the presence of quantum adversaries, when assuming the IND-CPA security notion as the minimal security requirement. To this end, we prove a new result on the probability of sampling generating sets of finite (sub-)groups if sampling is done with respect to an arbitrary, unknown distribution. Finally, we provide a sufficient condition on homomorphic encryption schemes for our quantum attack to work and discuss its satisfiability in non-group homomorphic cases. The impact of our results on recent fully homomorphic encryption schemes poses itself as an open question.

Keywords: Public-Key Cryptography, Homomorphic Encryption, Semantic Security, Quantum Algorithms, Sampling Group Generators

1 Introduction

Since the introduction of public-key cryptography by Diffie and Hellman [12] in 1976, researchers strived to construct encryption schemes that are *group homomorphic*. This property can be characterized by requiring the encryption scheme

⋆ Supported by the German Federal Ministry of Education and Research (BMBF) within the EC-SPRIDE project.

⋆⋆ Supported by the THeCS project as part of the Dutch national program COMMIT.

to have a homomorphic decryption procedure, while the plaintext and ciphertext spaces form groups. Ever since, the topic of homomorphic encryption is of central importance in cryptography. The recent advances in fully homomorphic encryption (FHE) [6, 15, 16] constitute just one example of this trend. In practice, *group* homomorphic encryption schemes lie at the heart of several important applications, such as electronic voting [8], private information retrieval [23], or multiparty computation [7] to name just a few. Moreover, the group homomorphic property comes quite naturally, as witnessed by a number of encryption schemes, for example RSA [28], ElGamal [13], Goldwasser-Micali [19], where the homomorphic property was not a design goal, but rather arose "by chance".

So far, these cryptosystems were all analyzed in the classical model of computation. However, it is reasonable to assume that the *quantum* model of computation will become more realistic in the future. Unfortunately, in this model all aforementioned cryptosystems are insecure due to Shor's algorithm [29], which allows to efficiently solve the discrete logarithm problem and to factor large integers. That is, until today nobody has been able to come up with a group homomorphic encryption scheme that can withstand quantum attackers.

It seems that such a scheme would require other design approaches. For instance, when considering ElGamal-like encryption schemes, simply replacing the underlying computational hardness assumption by a supposedly quantum-resistant one, say code-based, is not enough [2]. In fact although there is a substantial number of classical cryptographic primitives that can be proven secure against quantum attackers, e.g. [21], we still know little about what classical primitives can be realized in the quantum world and what not. Indeed this applies to the case of group homomorphic encryption schemes as well: so far it was even undecided whether *group homomorphic encryption can exist at all in the quantum world.* In other words, does the absence of a quantum secure group homomorphic encryption scheme so far imply that the right approach has not been found yet (but may be in the future) or are there universal reasons that prevent the existence of such schemes?

1.1 Our Contributions

Basic Impossibility Result. The central contribution of this work is to give a *negative* answer to the above question:

> *It is impossible to construct secure group homomorphic encryption in the quantum world, if the plaintext and ciphertext spaces form abelian groups.*

More precisely, we prove that any such scheme[4] cannot meet the minimial security notion of IND-CPA security in the presence of quantum adversaries. Observe that this result not only re-confirms the insecurity of *existing* schemes, but shows that *all* group homomorphic encryption schemes (including all yet to come schemes) are inevitably insecure in the quantum world.

[4] Although we postulate that our result is extendible to arbitrary solvable groups, we focus on the abelian case, since it is the most important one for reasons of practicability in real-world applications.

Quantum Attack. In order to prove this impossibility, we start by exhibiting the fact that the IND-CPA security of any group homomorphic encryption scheme can be reduced to an abstract *Subgroup Membership Problem* (SMP), introduced by Cramer and Shoup [9], which is much easier to analyze. Roughly speaking, this problem states that given a group G with subgroup H and a randomly sampled (according to some arbitrary distribution $\mathcal{D}$) element $g \in G$, decide whether $g \in H$ or not. This reduction to the SMP tells us that in order to break the IND-CPA security of a given group homomorphic encryption scheme in the quantum world, it is sufficient to give a quantum algorithm that breaks the SMP. Now, the basic idea for breaking the SMP for groups (G, H) is to use Watrous' variant [30] of the famous group order-finding quantum algorithm, which will effectively decide membership.

Sampling Generators in Finite Groups. Unfortunately, this algorithm only works when given a set of generators of H which we commonly do not have. Hence we restrict to the generic case that an attacker has only access to an efficient *sampling algorithm* for H that samples according to some distribution $\mathcal{D}$. We distinguish between the following two cases:

- **Uniform Distribution.** If $\mathcal{D}$ is uniform, Erdös and Rényi [14] show that sampling polynomially many times from H will give a generating set with high probability—a result that has been improved by Pak and Bratus [26]: If $k = \lceil \log_2(|H|) \rceil$, then $k+4$ samples are enough to get a set of generators with probability $\geq 3/4$. After obtaining a generating set for H, we use Watrous' quantum algorithm to decide membership in H, and hence efficiently break the SMP for (G, H).
- **Arbitrary/Unknown Distribution.** In general, the distribution $\mathcal{D}$ does not have to be uniform, but can be arbitrary, or completely unknown. Interestingly, we prove that, even then, breaking the SMP is possible with (almost) linearly many samples only. Observe that as we do not make any restrictions on the sampling algorithm, we cannot exclude seemingly exotic cases where regions of H are hardly (or never) reached by the sampling algorithm. Thus, the best we can aim for is to find a generating set for a subgroup H^* of H such that the probability that a random sample (with respect to $\mathcal{D}$) does fall into H^* is above an arbitrarily chosen threshold δ. We call such subgroups to be *δ-covering*. It turns out that having a generating set for such a subgroup is enough to break the SMP for (G, H). The main challenge, however, is to find a generating set for a δ-covering subgroup. To this end, we prove a new result on the probability of sampling generating sets of finite (sub-)groups with unknown sampling distribution. More precisely, we show that for any chosen probability threshold δ^*, there exists a value N, which grows at most logarithmically in k and does not depend on $\mathcal{D}$, such that $N \cdot k + 1$ samples yield a generating set for a δ-covering subgroup with probability at least δ^*. This result represents one of the main technical contribution of our work. We believe that it is also applicable in other research areas, e.g., computational group theory, and hence might be of independent interest.

Possible Extensions to Fully Homomorphic Encryption Schemes. Finally, we provide a general sufficient condition on a homomorphic encryption scheme for our quantum attack to work and discuss the applicability in FHE schemes. The decision of whether our attack breaks any of the existing FHE schemes [6, 15, 16] proves itself to be a highly non-trivial task and lies outside the scope of this paper. We leave it as interesting future work.

1.2 Related Work

There are many papers dealing with the construction of IND-CPA secure group homomorphic encryption schemes [25, 17, 11, 2, 27]. Some of these works attempted to build such schemes using post-quantum primitives [1], which did not succeed (for a good reason as our results show). Also, for a restricted class of group homomorphic schemes, [2] shows the impossibility of using linear codes as the ciphertext group. Furthermore, we mention the impossibility (even in the classical world) of *algebraically homomorphic* encryption schemes [5], which are deterministic encryption schemes and thus do not fall into the class IND-CPA secure cryptosystems.

In the quantum world, there is an even more efficient algorithm for breaking such algebraically homomorphic schemes [10]. In this vein, there are many variants of Shor's algorithm [29] that are being used to solve different computational problems [24, 30], leading to the breakdown of certain cryptosystems. On the other hand, there are several papers dealing with the analysis of classical primitives in the presence of quantum adversaries [20, 21]. However, none of these works show a general impossibility of group homomorphic cryptosystems.

With respect to the sampling from finite groups, there are many papers that are concerned with the improvement of probability bounds on finding generating sets when sampling uniformly at random [14, 4, 26]. Similar strong results for the arbitrary sampling from finite groups are not known.

Finally, we mention the recent advances in fully homomorphic encryption (FHE) [6, 15, 16]. These schemes are not classified as being group homomorphic, as they follow a different design approach. Rather than having a group homomorphic decryption algorithm, the decryption is only guaranteed to run correctly for polynomially many evaluations of the group operation. Interestingly enough, our results show that since current FHE schemes are based on post-quantum hardness assumptions, they had to follow a different approach than the group homomorphic one.

1.3 Outline

We recall standard notation in Section 2 and show some basic observations on group homomorphic encryption and the Subgroup Membership Problem (SMP) in Section 3. Section 4 covers the main Theorem, showing the impossibility of group homomorphic encryption in the quantum world, thereby giving our new insights in the sampling of group generators. We discuss non-group homomorphic encryption, such as somewhat and (leveled) fully homomorphic encryption in Section 5.

2 Notation

Throughout the paper, we use some standard notation that we briefly want to recall. We write $x \longleftarrow X$ if X is a random variable or distribution and x is to be chosen randomly from X according to its distribution. In the case where X is solely a set, $x \overset{U}{\longleftarrow} X$ denotes that x is chosen uniformly at random from X. If we sample an element x from X by using a specific distribution $\mathcal{D}$, we write $x \overset{\mathcal{D}}{\longleftarrow} X$ (or $x \longleftarrow X$ when there is no doubt about the distribution $\mathcal{D}$). For a distribution $\mathcal{D}$ on X, the term $\mathcal{D}(x)$ for $x \in X$ expresses the probability with which x is sampled according to $\mathcal{D}$, i.e., the probability mass function at $x \in X$.

For an algorithm $\mathcal{A}$ we write $x \longleftarrow \mathcal{A}(y)$ if $\mathcal{A}$ outputs x on fixed input y according to $\mathcal{A}$'s distribution. Sometimes, we need to specify the randomness of a probabilistic algorithm $\mathcal{A}$ explicitly. To this end, we interpret $\mathcal{A}$ in the usual way as a deterministic algorithm $\mathcal{A}(y;r)$, which has access to values $r \longleftarrow \mathsf{Rnd}$ that are randomly chosen from some randomness space Rnd. Moreover, two distribution ensembles $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ taking values in a finite set S_λ (indexed by a parameter λ) are said to be *computationally indistinguishable*, if for all probabilistic polynomial time (PPT) algorithms $\mathcal{A}$ there exists a negligible function `negl` such that

$$\mathrm{Adv}_{\mathcal{A}}^{X,Y}(\lambda) := \left| \Pr_{x \longleftarrow X_\lambda}[\mathcal{A}(x) = 1] - \Pr_{y \longleftarrow Y_\lambda}[\mathcal{A}(y) = 1] \right| \leq \mathtt{negl}(\lambda).$$

We denote this by $X \overset{c}{=} Y$.

For a group G, we denote the neutral element by 1, and denote the binary operation on G by "$\cdot$", i.e., G is written in *multiplicative notation*. We recall that a subgroup H of a group G is said to be *normal* if $z \cdot h \cdot z^{-1} \in H$ for all $z \in G, h \in H$. In particular, this means that if G is an abelian group, then every subgroup H is normal.

In general, we will consider sequences of abelian groups $(G_\lambda)_\lambda$ indexed by a parameter λ, where any element of every G_λ admits a representation of size at most polynomial in λ. We might assume, without loss of generality, that the choice of this polynomial is the identity, and in particular that every G_λ has order upper bounded by 2^λ. We will just write G instead of G_λ for any fixed choice of λ.

By a *description* of a finite group G we mean an efficient (i.e., PPT in λ) sampling algorithm (where sampling is denoted by $x \longleftarrow G$), the neutral element 1, an efficient algorithm for performing the group operation on G, and one for the inversion of group elements. Notice that the output distribution of the sampling algorithm does not have to be necessarily uniform. We abuse notation and write G both for the description and for the group itself. Furthermore, for elements $x_1, \ldots, x_k \in G$, we write $\langle x_1, \ldots, x_k \rangle$ for the subgroup generated by $x_1, \ldots, x_k$.

3 Group Homomorphic Encryption

We recall the notion of public-key *group homomorphic* encryption, which roughly can be described as usual public-key encryption where the decryption algorithm is a group homomorphism.

Definition 1 (Group Homomorphic Encryption [2, 22]). *A public key encryption scheme* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is called* group homomorphic*, if for every output* $(\mathsf{pk}, \mathsf{sk})$ *of* $\mathsf{KeyGen}(\lambda)$*, the plaintext space* $\mathcal{P}$ *and the ciphertext space* $\widehat{\mathcal{C}}$ *are non-trivial groups such that*

- *the* set of all encryptions $\mathcal{C} := \{\mathsf{Enc}_{\mathsf{pk}}(m; r) \mid m \in \mathcal{P}, r \in \mathsf{Rnd}\}$ *is a non-trivial subgroup of* $\widehat{\mathcal{C}}$
- *the decryption* $\mathsf{Dec}_{\mathsf{sk}}$ *is a group homomorphism on* $\mathcal{C}$*, i.e.*

$$\mathsf{Dec}_{\mathsf{sk}}(c \cdot c') = \mathsf{Dec}_{\mathsf{sk}}(c) \cdot \mathsf{Dec}_{\mathsf{sk}}(c'), \text{ for all } c, c' \in \mathcal{C}.^{5}$$

Notice that the scheme does *not* include a membership testing algorithm (i.e., an algorithm to test whether a group element is a valid encryption or not). The standard security notion for such homomorphic encryption schemes is that of *indistinguishability under chosen-plaintext attack*, denoted by IND-CPA [2]. Informally, this notion states whenever an adversary picks two plaintext messages of his choosing and gets to see an encryption of either of them, it should be computationally infeasible for him to decide which of the two messages was encrypted. Formally, for a given security parameter λ, group homomorphic encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, and PPT adversary $\mathcal{A}$, we consider the experiment $\mathbf{Exp}^{\text{ind-cpa}}_{\mathcal{A},\mathsf{KeyGen}}(\lambda)$, where $\mathcal{A}$ chooses two different plaintexts m_0, m_1 and is then provided an encryption $\mathsf{Enc}_{\mathsf{pk}}(m_b)$ for a randomly chosen bit b and a public key pk output by $\mathsf{KeyGen}(\lambda)$. The experiment succeeds (outputs 1) if b is guessed correctly. We say that $\mathcal{E}$ is IND-CPA *secure* if the advantage

$$\left| \Pr\left[\mathbf{Exp}^{\text{ind-cpa}}_{\mathcal{A},\mathsf{KeyGen}}(\lambda) = 1\right] - \frac{1}{2} \right| \text{ is negligible for all PPT adversaries } \mathcal{A}.$$

Moreover, we recall a fact showing the strong group-theoretic structure of the set of encryptions of $1 \in \mathcal{P}$ for *any* group-homomorphic encryption scheme. For this, we introduce the *set of all encryptions of* $m \in \mathcal{P}$

$$\mathcal{C}_m := \{c \in \mathcal{C} \mid \mathsf{Dec}_{\mathsf{sk}}(c) = m\}.$$

Fact 1 (Basic Properties [2]) *Let* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an arbitrary group homomorphic encryption scheme. It holds that*

1. $\mathcal{C}_m = \mathsf{Enc}_{\mathsf{pk}}(m; r) \cdot \mathcal{C}_1$ *for all* $m \in \mathcal{P}$ *and all* $r \in \mathsf{Rnd}$*, and*

[5] Note that the decryption might output an error $\perp$ on inputs in $\widehat{\mathcal{C}} \setminus \mathcal{C}$. Therefore, requiring it to be homomorphic on $\mathcal{C}$ is as general as possible since we do not give any restriction on its behaviour outside of $\mathcal{C}$.

2. $\mathcal{C}_1$ *is a proper* normal *subgroup of* $\mathcal{C}$ *such that* $|\mathcal{C}_1| = |\mathcal{C}_m|$ *for all* $m \in \mathcal{P}$.

It follows that the set $\{\mathsf{Enc}_{\mathsf{pk}}(m; r) \mid m \in \mathcal{P}\}$ *for a fixed* r *is a system of representatives of* $\mathcal{C}/\mathcal{C}_1$.

With this notation, the IND-CPA security of $\mathcal{E}$ is equivalent to saying that the distribution on $\mathcal{C}_{m_0}$ (induced by the encryption algorithm $\mathsf{Enc}_{\mathsf{pk}}(m)$) is computationally indistinguishable from the distribution on $\mathcal{C}_{m_1}$ for any two messages m_0 and m_1 [18, Ch. 5.2], i.e., $\mathcal{C}_{m_0} \overset{c}{=} \mathcal{C}_{m_1}$.

Necessary Security Condition. We briefly recall the *Subgroup Membership Problem* (SMP) which was introduced by Cramer and Shoup in [9].

Definition 2 (Subgroup Membership Problem). *Let* Gen *be a PPT algorithm that takes a security parameter* λ *as input and outputs descriptions* (G, H) *where* H *is a non-trivial, proper subgroup of a finite group* G. *Additionally, we assume here that there is an algorithm that allows for the efficient sampling from* $G \setminus H$. *We consider the following experiment for a given algorithm* Gen*, algorithm* $\mathcal{A}$ *and parameter* λ*:*

Experiment $\mathbf{Exp}^{\mathrm{smp}}_{\mathcal{A},\mathsf{Gen}}(\lambda)$:

1. $(G, H) \longleftarrow \mathsf{Gen}(\lambda)$
2. Choose $b \overset{U}{\longleftarrow} \{0, 1\}$. *If* $b = 1$*:* $z \longleftarrow G \setminus H$. *Otherwise:* $z \longleftarrow H$.
3. $d \longleftarrow \mathcal{A}(G, H, z)$ *where* $d \in \{0, 1\}$
4. The output of the experiment is defined to be 1 if $d = b$ *and 0 otherwise.*

We say that the SMP *is* hard for (G, H) *(or* relative to Gen*) if the advantage*

$$\left|\Pr\left[\mathbf{Exp}^{\mathrm{smp}}_{\mathcal{A},\mathsf{Gen}}(\lambda) = 1\right] - \frac{1}{2}\right| \text{ is negligible for all PPT algorithms } \mathcal{A}.$$

We stress the fact that the efficient sampling from $G \setminus H$ does not have to be uniform. Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a group homomorphic encryption scheme with the group $\mathcal{C}$ of all encryptions and the subgroup $\mathcal{C}_1$ of all encryptions of the neutral element 1. In fact, the hardness of SMP for $(\mathcal{C}, \mathcal{C}_1)$ (i.e., relative to KeyGen) is a necessary condition for $\mathcal{E}$ to be IND-CPA secure. Recall that the sampling algorithms for the groups $\mathcal{C}$ and $\mathcal{C}_1$ are the ones inherited from the encryption algorithm of $\mathcal{E}$. In particular, sampling an element c from $\mathcal{C} \setminus \mathcal{C}_1$ is done by choosing a random message $m \in \mathcal{P}$ with $m \neq 1$ and then computing c as $\mathsf{Enc}_{\mathsf{pk}}(m; r)$ for $r \longleftarrow \mathsf{Rnd}$. We have the following immediate result:

Theorem 1 (Necessary Condition on IND-CPA Security). *For a group homomorphic encryption scheme* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *we have:*

$$\mathcal{E} \textit{ is } \text{IND-CPA } \textit{secure} \implies \text{SMP } \textit{is hard (relative to } \mathsf{KeyGen}\textit{)}.$$

The above holds regardless of the type of adversary (i.e., classical vs quantum) taken into account. A straightforward proof of this Theorem can be found in

Appendix A. Since it is a popular belief (and for reasons of completeness), we want to point out that the converse of the Theorem does *not* hold in general. This can be seen by considering a somewhat pathological example, which we present in Appendix B. Note that the converse of Theorem 1 *does*, however, hold for so-called *shift-type homomorphic encryption schemes* [3], which describe a certain subclass of group homomorphic encryption schemes that actually encompasses all existing instances. Furthermore, it also holds for bit encryption schemes, since there are only two messages, 0 and 1.

4 General Impossibility in the Quantum World

Let Gen be a PPT algorithm that takes a security parameter λ as input and outputs descriptions (G, H) where H is a non-trivial, proper subgroup of a finite group G with an additional algorithm for the efficient sampling from $G \setminus H$ (cf. Section 3). Now, assume that for any such algorithm Gen, we can construct a quantum algorithm $\mathcal{A}_Q$ that breaks the hardness of SMP relative to Gen. In particular, for a given group homomorphic encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ this means that we have a quantum algorithm $\mathcal{A}_Q$ that breaks the hardness of SMP relative to KeyGen. However, by Theorem 1, this implies that we can construct an algorithm that breaks the IND-CPA security of $\mathcal{E}$. Since we had no restriction on the encryption scheme $\mathcal{E}$, this would imply that *any* group homomorphic encryption scheme $\mathcal{E}$ is insecure in terms of IND-CPA in the quantum world. This is the result we want to prove in this section, at least for the abelian case, i.e., when G is an abelian group. Therefore, let Gen be as above but with G being abelian.

It is well-known that a modification of the famous order-finding quantum algorithm [30] can efficiently find the order of an abelian group, given that we have its description by a set of generators.

Theorem 2 (Quantum Order-Finding Algorithm with Generators [30]). *Let G be a finite abelian group with $k = \lceil \log_2(|G|) \rceil$. Then, there exists a quantum algorithm which, given a generating set of G and an error probability ε as an input, outputs the order of G with probability at least $1 - \varepsilon$ in time $o(\mathtt{poly}(k + \log_2(1/\varepsilon)))$.*

This Theorem already is sufficient to break the hardness of SMP (relative to Gen), *if* the description of H contains a set of generators, as the next Theorem shows.

Theorem 3 (Quantum Attack on SMP with Generators). *Let (G, H) be the output of $\mathsf{Gen}(\lambda)$, for some security parameter λ, such that H contains a set of generators $g_1, \ldots, g_r$. Since Gen is a PPT algorithm, this implies that $k = k(\lambda) = \lceil \log_2(|H|) \rceil$ is a polynomial in λ. There exists a quantum algorithm which, given $g_1, \ldots, g_r$ (i.e., the description of H), breaks the hardness of* SMP *with probability at least $(1 - \varepsilon)^2$ in time $o(\mathtt{poly}(k + \log_2(1/\varepsilon)))$.*

Proof. Let z denote the challenge in the SMP game (Def. 2), i.e., $z \in G \setminus H$ if $b = 1$, and $z \in H$ otherwise. Since H contains a set of generators $g_1, \ldots, g_r$, we can run the quantum algorithm in Theorem 2 twice: the first time on the generating set and the second time on the generating set plus the element z. Provided that both runs succeed, we have that $z \in H$ (i.e., $b = 0$) if and only if the two subgroup orders, obtained from the two algorithm runs, are the same. But both runs succeed with probability $(1 - \varepsilon)^2$. This proves the Theorem. □

Recall that the original definition of SMP gives no set of generators for H a priori, since the description of a group only contains standard algorithms for the group operations and a sampling algorithm (cf. Section 2). However, we show that the previous Theorem extends to this case, i.e., when only having a sampling algorithm. For the sake of readability, we will first treat the case of sampling *uniformly at random* from H (Section 4.1), and will then show the general case with arbitrary (possibly unknown) sampling from H (Section 4.2).

4.1 Breaking SMP with Uniform Sampling

It is well-known that if we have a sampling algorithm for H that samples *uniformly at random*, we can obtain a set of generators by sampling polynomially (in the base-2 logarithm of the order of H) many times from H. If $k = \lceil \log_2(|H|) \rceil$, Pak and Bratus [26] show that $k + 4$ samples are sufficient to generate the whole group with probability $> 3/4$. This result is an improvement over a result by Erdös and Rényi [14]. We recall it in the following Theorem:

Theorem 4 (Probability of Finding a Generating Set with Uniform Sampling [26]). *Let H be a finite abelian group of order n where $k = \lceil \log_2(n) \rceil$. Then:*

$$\Pr_{x_1, \ldots, x_{k+4} \xleftarrow{U} H} [\langle x_1, \ldots, x_{k+4} \rangle = H] > \frac{3}{4}.$$

As an immediate corollary of this Theorem and Theorem 3 we have the main result of this section.

Theorem 5 (Quantum Attack on SMP with Uniform Sampling). *Let (G, H) be the output of $\mathsf{Gen}(\lambda)$ with $k = \lceil \log_2(|H|) \rceil$, for some security parameter λ, such that the sampling algorithm in the description of H samples* uniformly at random *from H. Then, there exists a quantum algorithm which breaks the hardness of* SMP *with probability at least $\frac{3}{4}(1 - \varepsilon)^2$ in time $o(\mathtt{poly}(k + \log_2(1/\varepsilon)))$, and by sampling only $k + 4$ times from H.*

We remark that the constant $\frac{3}{4}$ can be greatly improved by increasing the number of samples we take from H, approximating 1 very quickly. In general, by performing $k + l$ random sampling, the success probability approximates 1 exponentially fast in l.

4.2 Breaking SMP with Arbitrary/Unknown Sampling

In this section, we show an extension of Theorem 5 to the general case, where the description of H only contains a sampling algorithm with *unknown/arbitrary distribution* $\mathcal{D}$. Since we do not make any restrictions on the sampling algorithm, we cannot exclude seemingly exotic cases where parts of H are hardly (or not at all) reached by the sampling algorithm. Consider the following example:

Example 1. Let $\lambda \geq 1$ be the security parameter. We define a family of groups by $G_\lambda := GF(2)^\lambda$ together with sampling distributions $\mathcal{D}_\lambda$ on G_λ as through the probability mass function

$$\mathcal{D}_\lambda(v_1, \ldots, v_\lambda) := \begin{cases} \frac{1}{2^{\lambda-1}} - \frac{1}{2^{\lambda\cdot(\lambda-1)}} & \text{, if } v_1 = 0 \\ \frac{1}{2^{\lambda\cdot(\lambda-1)}} & \text{, otherwise.} \end{cases} \tag{1}$$

Here, $(v_1, \ldots, v_\lambda)$ denotes an arbitrary element from $GF(2)^\lambda$. Observe that the probability of sampling one vector $(v_1, \ldots, v_\lambda)$ with $v_1 = 1$ is $2^{-\lambda}$. However, at least one such sample is necessary for a generating set of the whole group. This shows that the probability of sampling a generating set for the whole group is negligible in λ.

As the examples illustrates, the best we can aim for (in general) is to find a generating set for a subgroup of H such that the probability that a random sample (with respect to $\mathcal{D}$) does fall into this group is sufficiently large. This motivates the following definition:

Definition 3 (Covering Subgroup). *Let a finite group H be given, together with a sampling distribution $\mathcal{D}$. For a value $0 \leq \delta \leq 1$, we say that a subgroup $H^* \leq H$ is a δ-*covering subgroup *of H with respect to $\mathcal{D}$ if*

$$\Pr_{x \xleftarrow{\mathcal{D}} H}[x \in H^*] \geq \delta. \tag{2}$$

Example 2. Observe that the whole group H is trivially a δ-covering subgroup. A less trivial example is the following. We order the elements $h \in H$ in descending order according to their probabilities of being sampled, that is $h_1, h_2, \ldots$ with $\mathcal{D}(h_i) \geq \mathcal{D}(h_{i+1})$ for all i. Now, let b denote the smallest index such that $\sum_{i=1}^{b} \mathcal{D}(h_i) \geq \delta$. Then $\langle h_1, \ldots, h_b \rangle$ is for sure a δ-covering subgroup.

Obviously, it follows directly from Theorem 3 that given generators of a δ-covering subgroup, there exists a quantum attack on SMP with success probability at least $\delta \cdot (1-\varepsilon)^2$ in time $o(\texttt{poly}(k + \log_2(1/\varepsilon)))$. Thus in the remainder of this section, we consider the task of finding, with probability $\geq \sigma$, a generating set for a δ-covering subgroup (for fixed, but arbitrary values δ, σ) if only a sampling algorithm Sample is given which samples according to an arbitrary (possibly unknown) distribution $\mathcal{D}$. To this end, we prove the following new result on the probability of finding a δ-covering subgroup (with generators) of a finite group with arbitrary/unknown sampling distribution and a given value δ.

Algorithm 1 Sample generating set of a δ-covering subgroup

Given: A group H with sampling algorithm Sample, an integer $k = \lceil \log_2 |H| \rceil$, a membership testing procedure that efficiently tests for any subset $S \subseteq H$ and any $x \in H$ whether $x \in \langle S \rangle$, two real values $0 \leq \delta, \sigma \leq 1$.

Output: A set S of elements that generate a δ-covering subgroup of H with probability at least σ.

```

$x \leftarrow \mathsf{Sample}$, $S \leftarrow \{x\}$                    {Initial candidate for a generating set}
$N := \left\lceil \frac{\log(1-\sigma)-\log(k)}{\log(\delta)} \right\rceil$                    {Number of samples per round}

for $j = 1, \ldots, k$ do
    $x_i \leftarrow \mathsf{Sample}, i = 1, \ldots, N$                    {Sample $N$ elements from $H$}
    if $x_i \in \langle S \rangle$ for all $i = 1, \ldots, N$ then
        Abort for-loop                    {Abort as all samples are already in $\langle S \rangle$}
    else
        $S \leftarrow S \cup \{x_1, \ldots, x_N\}$                    {Extend candidate generating set}
    end if
end for

return $S$
```

Theorem 6 (Sampling a Generating Set for a δ-covering Subgroup). *Let H be a finite group, together with a sampling algorithm* Sample *that samples according to a (possibly unknown) distribution $\mathcal{D}$, and let $k = \lceil \log_2(|H|) \rceil$. Moreover, fix two values $0 \leq \delta, \sigma \leq 1$ and set $N := \left\lceil \frac{\log(1-\sigma)-\log(k)}{\log(\delta)} \right\rceil$.*

Let $x_1, \ldots, x_{N \cdot k+1} \in H$ be $N \cdot k + 1$ samples from H by invoking the sampling algorithm, i.e., $x_i \leftarrow$ Sample *for $i = 1, \ldots, N \cdot k + 1$. Then with probability at least σ, the group $H^* := \langle x_1, \ldots, x_{N \cdot k+1} \rangle$ is a δ-covering subgroup of H.*

Observe that like in the case of uniform sampling, a polynomial number of samples (almost linear in k) is sufficient. Interestingly, this number of samples is independent of the distribution.

For the sake of readability, we prove Theorem 6 in two steps. In the first step, we present an algorithm (Algorithm 1) that makes *at most* $N \cdot k + 1$ samples and outputs a set $S \subseteq H$. We prove that S is a generating set for a δ-covering subgroup with probability at least σ. The algorithm relies on the assumption of the existence of an efficient membership testing procedure. But in the second step we present a modification of the algorithm, Algorithm 2, that works *without* the membership testing procedure and has at least the same success probability. In fact, Algorithm 2 makes *exactly* $N \cdot k + 1$ samples, hence proving Theorem 6.

We start with Algorithm 1 and prove the following result:

Theorem 7 (Correctness of Algorithm 1). *With a probability of at least σ, the output S of Alg. 1 is a generating set for a δ-covering subgroup.*

Algorithm 2 Sample generating set of a δ-covering subgroup

```
Given: A group $H$ with sampling algorithm Sample, an integer $k = \lceil \log_2 |H| \rceil$, and
    two real values $0 \leq \delta, \sigma \leq 1$
Output: A set $S$ of elements that generate a $\delta$-covering subgroup of $H$ with probability
    $\geq \sigma$
 1:
 2: $x \leftarrow$ Sample, $S \leftarrow \{x\}$                {Initial candidate for a generating set}
 3: $N := \left\lceil \frac{\log(1-\sigma)-\log(k)}{\log(\delta)} \right\rceil$      {Number of samples per round}
 4:
 5: for $j = 1, \ldots, k$ do
 6:     $x_i \leftarrow$ Sample, $i = 1, \ldots, N$              {Sample $N$ elements from $H$}
 7:     $S \leftarrow S \cup \{x_1, \ldots, x_N\}$             {Extend candidate generating set}
 8: end for
 9:
10: return $S$
```

Proof. Let S denote the output of Alg. 1 and $H^* := \langle S \rangle$. There are two possibilities: (i) the algorithm aborted the **for**-loop for some value $j < k$ or (ii) the algorithm executed all k **for**-loops.

First, we consider case (i). At the same time, assume that H^* is *not* a δ-covering subgroup, that is

$$\delta^* := \Pr\left[x \in H^* | x \xleftarrow{\mathcal{D}} H\right] < \delta$$

(this would be a failure of the algorithm). As the algorithm aborted the **for**-loops for some value $j < k$ by assumption, this can only happen if $x_i \in \langle S \rangle =: H^*$ for all N samples made in round j although $\delta^* < \delta$. As the samples are made independently, the probability of this error event happening at a certain round is $(\delta^*)^N < \delta^N$; since there are at most $k-1$ independent rounds in case (i), the probability that an error occurs in any of them is at most $k \cdot \delta^N < 1 - \sigma$ by definition of N. Hence, the probability that no error happens and the output is correct, i.e., is a generating set of a δ-covering subgroup, is at least $1-(1-\sigma) = \sigma$. This concludes the first case.

Now, we consider case (ii), i.e., the algorithm has executed all k **for**-loops. For simplicity, we index the sets S according to the round number. More precisely, let S_0 denote the initial candidate for the generating set (line 2). Moreover, let S_ℓ denote the set S at the end of the while loop (after being extended - see line 10) and we define $H_\ell := \langle S_\ell \rangle$ for $\ell \geq 0$. Observe that $H_\ell \subseteq H$ for all ℓ by construction. The output of the algorithm is $S = S_k$. We make use of the following inequalities that we prove afterwards:

$$\mathrm{ord}(H_\ell) \geq 2^\ell \quad , \forall \ell \geq 0. \tag{3}$$

A consequence of (3) is that $\mathrm{ord}(H_k) \geq 2^k \geq \mathrm{ord}(H)$ which implies that $H_k = H$. Hence, $H^* = H_k = H$ is the whole group and trivially a δ-covering group for any value $0 \leq \delta \leq 1$.

It remains to prove the inequalities in (3), i.e., $\operatorname{ord}(H_\ell) \geq 2^\ell$ for all $0 \leq \ell \leq k$. Observe that H_ℓ is a proper subgroup of $H_{\ell+1}$ for every $\ell < k$. Thus, the number $\frac{|H_{\ell+1}|}{|H_\ell|}$ (which is an integer, by Lagrange's Theorem), must be strictly greater than 1. Hence $|H_{\ell+1}| \geq 2\,|H_\ell|$, and this proves (3) since $|H_0| = 1$. □

Observe that Alg. 1 runs at most k for-loops and uses the membership test procedure only for deciding if the algorithm can be stopped earlier. Hence, we consider a variant, namely Alg. 2, which simply drops this test and always runs all k loops. That is, the only difference between Algorithms 1 and 2, respectively, is that the latter may run longer (but still at most k loops) and outputs a superset S' of the output S of Alg. 1. Of course, if S is a generating set for a δ-covering subgroup, then this is certainly true for S' as well. This shows that Alg. 2 "inherits" the success probability of Alg. 1:

Corollary 1. *[Correctness of Algorithm 2] With a probability of at least σ, the output S of Algorithm 2 is a generating set for a δ-covering subgroup.*

Observe that Alg. 2 simply outputs $N \cdot k + 1$ samples. Hence, the proof of Theorem 6 is a direct consequence of Corollary 1. The remainder of this section is straightforward. Given a generating set S of a δ-covering subgroup, we can apply Theorem 3 in order to break the SMP for (G, H).

Theorem 8 (Quantum Attack on SMP with Arbitrary Sampling). *Let (G, H) be the output of $\mathsf{Gen}(\lambda)$ with $k = \lceil \log_2(|H|) \rceil$, for some security parameter λ. We denote the distribution of the sampling algorithm contained in the description of H by $\mathcal{D}$. Let $0 \leq \varepsilon^* \leq 1$ be an arbitrary fixed positive value. Then, there exists a value $N = N(k, \varepsilon^*)$ (which only grows at most logarithmically in k) and a quantum algorithm which breaks the hardness of* SMP *with probability at least $(1-\varepsilon^*)\,(1-\varepsilon)^2$ in time $o(\mathtt{poly}(k + \log_2(1/\varepsilon)))$, and by sampling only $N \cdot k + 1$ times from H (where ε is the error probability of Theorem 2).*

In particular, we can construct a quantum algorithm that breaks SMP *with probability at least $\frac{3}{4}(1-\varepsilon)^2$ in time $o(\mathtt{poly}(k + \log_2(1/\varepsilon)))$ while only sampling $7k \cdot (2 + \lceil \log(k) \rceil) + 1$ times from H.*

Proof. In principle, the attacker $\mathcal{A}$ is the same as described in Theorems 3 and 5, the only difference being the approach for finding an appropriate generating set. Given the value ε^*, the attacker chooses two positive values δ, σ such that $\delta \cdot \sigma \geq (1 - \varepsilon^*)$, for example $\delta = \sigma = \sqrt{1 - \varepsilon^*}$. Then, the attacker makes $N \cdot k + 1$ samples as explained in Theorem 6. Let H^* denote the subgroup of H that is generated by these $N \cdot k + 1$ samples. Due to Corollary 1, we know that H^* is a δ-covering subgroup of H with probability σ. From this point on, the attack continues as specified in Theorem 3, while using the $N \cdot k + 1$ samples as generators, i.e., we let z denote the challenge in the SMP game (Def. 2), so $z \in G \backslash H$ if $b = 1$, and $z \in H$ otherwise. If $b = 1$ (which happens with probability $\frac{1}{2}$), we know that $z \notin H^*$ and the attacker $\mathcal{A}$ will recognize this with probability $\geq (1-\varepsilon)^2$ (as in the proof of Theorem 3). If $b = 0$ (which also happens with probability $\frac{1}{2}$), several sub-cases do exist (depending on whether H^* is δ-covering and whether $z \in H^*$). In case that both properties are true (which happens with probability $\geq \sigma \cdot \delta$), the attacker recognizes that $z \in H^*$ again with probability $\geq (1-\varepsilon)^2$. As the success probabilities in the other sub-cases are at least zero, it follows that

$$\Pr\left[\mathbf{Exp}^{\mathrm{smp}}_{\mathcal{A},\mathsf{Gen}}(\lambda) = 1\right] \geq \frac{(1-\varepsilon)^2 + \delta\sigma(1-\varepsilon)^2}{2} \geq \delta\sigma(1-\varepsilon)^2 \geq (1-\varepsilon^*)(1-\varepsilon)^2$$

which concludes the proof of the first part of the Theorem. For the second part, we see that when choosing $\varepsilon^* = \frac{1}{4}$ and $\delta = \sigma = \frac{1}{2}\sqrt{3}$, the above attacker $\mathcal{A}$ has a success probability of at least $\frac{3}{4}(1-\varepsilon)^2$ by sampling only $N \cdot k + 1$ times from H where $N = \left\lceil \frac{\log(1-\sigma)-\log(k)}{\log(\delta)} \right\rceil \leq 7\left(\lceil \log(k) \rceil + 2\right)$. □

Finally, Theorems 8 and 1 together immediately imply our main result: the general impossibility of group homomorphic encryption in the quantum world, if the plaintext and ciphertext groups are abelian.

Theorem 9 (Impossibility of Group Homomorphic Encryption in the Quantum World). *Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an* IND-CPA *secure group homomorphic encryption scheme with abelian plaintext and ciphertext groups. Then, there exists a quantum PPT algorithm that breaks the security of $\mathcal{E}$ with non-negligible probability.*

5 Discussion

In this section, we provide an informal discussion about the applicability of our quantum attack to non-group homomorphic encryption schemes and elaborate on fully homomorphic encryption (FHE). In abstract terms, existing FHE schemes are standard public-key encryption schemes $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following extras [15]:

- the plaintext space $\mathcal{P}$ and ciphertext space $\widehat{\mathcal{C}}$ are rings,
- there is an algorithm Eval that takes as input a public key pk, a circuit C, a tuple $(c_1, \ldots, c_t)$ of ciphertexts (one for every input node of C), and outputs another ciphertext c, and
- for all outputs $(\mathsf{pk}, \mathsf{sk})$ by $\mathsf{KeyGen}(\lambda)$, all polynomials $p(\lambda)$ in λ, all $t \leq \mathtt{poly}(\lambda)$, all plaintexts $m_1, \ldots, m_t \in \mathcal{P}$ corresponding to fresh encryptions $c_i \longleftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i)$, $i = 1 \ldots t$, and all t-input circuits C of depth $\leq p(\lambda)$, we have the following *correctness* condition:

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(C, c_1, \ldots, c_t)) = C(m_1, \ldots, m_t). \tag{4}$$

Homomorphic encryption schemes for which the polynomial depth $p(\lambda)$ of the circuits C is bounded a priori (i.e., fixed in the public key pk) are called *leveled* FHE. For very small polynomials $p(\lambda)$, we say that the scheme is *somewhat* homomorphic. At a first glance, there a two main differences to the notion of *group* homomorphic encryption (see Fig. 1 for a pictorial explanation):

1. The set of all (fresh) encryptions $\mathcal{C} = \{\mathsf{Enc}_{\mathsf{pk}}(m; r) \mid m \in \mathcal{P}, r \in \mathsf{Rnd}\}$ is only a *subset* (and not necessarily a subgroup) of the ring $\widehat{\mathcal{C}}$.
2. The decryption is not necessarily a group homomorphism as it is only guaranteed to run correctly with circuits that are polynomially bounded in depth; this polynomial bound can be dynamically chosen in the "pure" FHE case, while it is fixed in the public key for leveled FHE and somewhat homomorphic schemes. But if the decryption is group homomorphic, it particularly must run correctly (at least theoretically) on all unbounded circuits consisting only of group-operation gates.

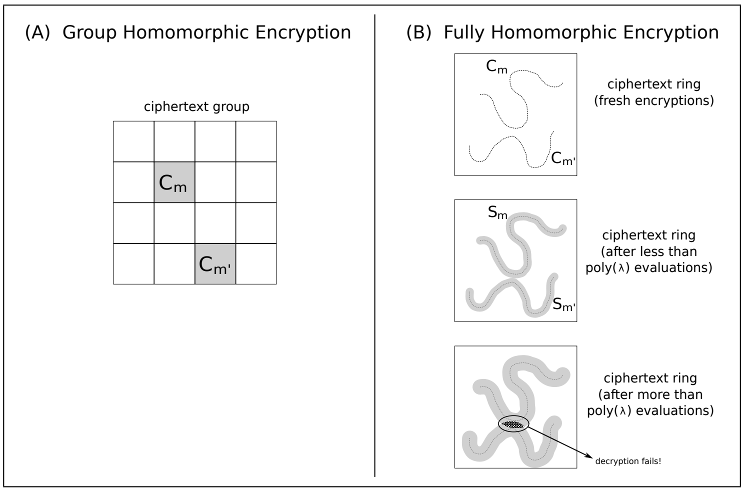
(A) Group Homomorphic Encryption
ciphertext group
Cm
Cm'
(B) Fully Homomorphic Encryption
Cm
Cm'
ciphertext ring (fresh encryptions)
Sm
Sm'
ciphertext ring (after less than poly(λ) evaluations)
ciphertext ring (after more than poly(λ) evaluations)
decryption fails!

Fig. 1. Differences between group homomorphic encryption and FHE: (A) shows that each $\mathcal{C}_m$ is a coset of $\mathcal{C}_1$ in $\mathcal{C}$ (Fact 1), while the decryption is a group homomorphism; (B) shows first that $\mathcal{C}_m$ and $\mathcal{C}_{m'}$ are subsets and not necessarily cosets in $\mathcal{C}$, second that the decryption runs correctly on $\texttt{poly}(\lambda)$ evaluations of ciphertexts, and third that the decryption might fail if exponentially many evaluations have been performed, meaning that the decryption is not necessarily group homomorphic.

If the decryption is not a group homomorphism, the set of fresh encryptions of the neutral element in $\mathcal{P}$ is not necessarily a group, but only a subset of $\widehat{\mathcal{C}}$. However, the quantum order-finding algorithm of Theorem 2 only works on (solvable) *groups*. This immediately gives us the first important observation:

Observation 1 *Our quantum attack from Section 4 on group homomorphic encryption schemes is* not *immediately applicable to more general homomorphic encryption schemes, such as somewhat and (leveled) FHE schemes.*

A sufficient condition that we need a homomorphic scheme to have for our quantum attack to work is the following:

Sufficient Condition (Quantum Attack). For any output $(\mathsf{pk}, \mathsf{sk})$ by $\mathsf{KeyGen}(\lambda)$, there exist two plaintexts $m, m' \in \mathcal{P}$ and a subgroup G of $\widehat{\mathcal{C}}$ such that

1. there exists an efficient PPT algorithm which outputs a generating set for G of size at most $\texttt{poly}(\lambda)$,
2. the probability $\Pr_{c \longleftarrow \mathsf{Enc}_{\mathsf{pk}}(m)} [c \in G]$ is non-negligible in λ, and
3. the probability $\Pr_{c' \longleftarrow \mathsf{Enc}_{\mathsf{pk}}(m')} [c' \notin G]$ is non-negligible in λ.

In the setting of group homomorphic encryption schemes, the plaintext m would be the neutral element 1, while $m' \neq 1$ can be any other plaintext. The group G satisfying the above conditions would be a δ-covering subgroup of the group $\mathcal{C}_1$ of all (fresh) encryptions of 1, for a sufficiently small δ. For more general homomorphic encryption schemes, such as somewhat or (leveled) FHE schemes, the situation looks more like in Fig. 2.

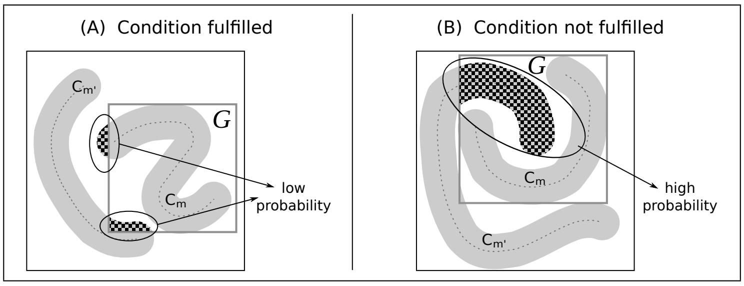

Fig. 2. Our condition in the FHE case: (A) shows pictorially when the condition is fulfilled; (B) shows the case when item 3 of the condition is not met and G intersects with a large part of encryptions of m'.

The important observation here is, that as long as only polynomially many evaluations of the ciphertexts have been performed, the decryption still runs correctly (cf. correctness condition in Equation (4)). But for any scheme to be IND-CPA secure, the set of encryptions of a given message m must be exponentially large, so in particular, a group G that fulfills condition 2 is required to be exponentially large. Hence, the decryption is not guaranteed to run correctly on G and might fail. More precisely, condition 3 for our attack to work will most likely be unsatisfied. However, proving or disproving that any of the existing somewhat or (leveled) FHE schemes satisfies our sufficient condition is a highly non-trivial task (due to the very general and abstract nature of the requirement) and lies outside the scope of this work. We leave it as interesting future work. Interestingly enough, since most of the existing FHE schemes base their security on supposedly quantum-resistant hardness assumptions (such as LWE), spotting a scheme that is susceptible to our quantum attack will effectively break the underlying hardness assumption and thereby disprove its quantum-resistance.

Acknowledgements. We would like to thank Richard Lindner and Pooya Farshim for helpful discussions. We are also grateful for the constructive comments by the anonymous reviewers.

References

1. Armknecht, F., Augot, D., Perret, L., Sadeghi, A.R.: On constructing homomorphic encryption schemes from coding theory. In: IMA Int. Conf. LNCS, vol. 7089, pp. 23–40. Springer (2011)
2. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, Codes and Cryptography pp. 1–24, 10.1007/s10623-011-9601-2
3. Armknecht, F., Katzenbeisser, S., Peter, A.: Shift-type homomorphic encryption and its application to fully homomorphic encryption. In: AFRICACRYPT. LNCS, vol. 7374, pp. 234–251. Springer (2012)
4. Babai, L.: Local expansion of vertex-transitive graphs and random generation in finite groups. In: STOC. pp. 164–174. ACM (1991)
5. Boneh, D., Lipton, R.J.: Algorithms for black-box fields and their application to cryptography (extended abstract). In: CRYPTO. LNCS, vol. 1109, pp. 283–297. Springer (1996)
6. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: ITCS. pp. 309–325. ACM (2012)
7. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: EUROCRYPT. LNCS, vol. 2045, pp. 280–299. Springer (2001)
8. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In: EUROCRYPT. pp. 103–118 (1997)
9. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT. LNCS, vol. 2332, pp. 45–64. Springer (2002)
10. van Dam, W., Hallgren, S., Ip, L.: Quantum algorithms for some hidden shift problems. SIAM J. Comput. 36(3), 763–778 (2006)
11. Damgård, I., Geisler, M., Krøigaard, M.: Homomorphic encryption and secure comparison. IJACT 1(1), 22–31 (2008)
12. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
13. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO. pp. 10–18 (1984)
14. Erdös, P., Rényi, A.: Probabilistic methods in group theory. J. Analyse Math. 14, 127–138 (1965)
15. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC. pp. 169–178. ACM (2009)
16. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the aes circuit. In: CRYPTO. LNCS, vol. 7417, pp. 850–867. Springer (2012)
17. Gjøsteen, K.: Homomorphic cryptosystems based on subgroup membership problems. In: Mycrypt. LNCS, vol. 3715, pp. 314–327. Springer (2005)
18. Goldreich, O.: The Foundations of Cryptography - Volume 2, Basic Applications. Cambridge University Press (2004)

19. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
20. Hallgren, S., Kolla, A., Sen, P., Zhang, S.: Making classical honest verifier zero knowledge protocols secure against quantum attacks. In: ICALP (2). LNCS, vol. 5126, pp. 592–603. Springer (2008)
21. Hallgren, S., Smith, A., Song, F.: Classical cryptographic protocols in a quantum world. In: CRYPTO. LNCS, vol. 6841, pp. 411–428. Springer (2011)
22. Hemenway, B., Ostrovsky, R.: On homomorphic encryption and chosen-ciphertext security. In: PKC. LNCS, vol. 7293, pp. 52–65. Springer (2012)
23. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: FOCS. pp. 364–373 (1997)
24. Mosca, M.: Quantum computing, cryptography and compilers. In: ISMVL. pp. 154–156. IEEE (2012)
25. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT. LNCS, vol. 1592, pp. 223–238. Springer (1999)
26. Pak, I., Bratus, S.: On sampling generating sets of finite groups and product replacement algorithm (extended abstract). In: ISSAC. pp. 91–96. ACM (1999)
27. Peter, A., Kronberg, M., Trei, W., Katzenbeisser, S.: Additively homomorphic encryption with a double decryption mechanism, revisited. In: ISC. LNCS, vol. 7483, pp. 242–257. Springer (2012)
28. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
29. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: FOCS. pp. 124–134. IEEE Computer Society (1994)
30. Watrous, J.: Quantum algorithms for solvable groups. In: STOC. pp. 60–67. ACM (2001)

A Proof of Theorem 1

We prove the theorem by contradiction and show that if we have a PPT algorithm $\mathcal{A}$ that breaks the hardness of SMP with non-negligible advantage $\Gamma(\lambda)$, we can construct (in PPT) an algorithm $\mathcal{B}$ that breaks the IND-CPA security with non-negligible advantage $\Gamma(\lambda)$. To this end, we fix an SMP-adversary $\mathcal{A}$ and construct an IND-CPA-adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$.

We start by letting $\mathcal{B}_1$ choose $m_0 = 1 \in \mathcal{P}$ and a random message $m_1 \longleftarrow \mathcal{P}$ with $m \neq 1$. Next, $\mathcal{B}_1$ sends the two messages m_0, m_1 to the IND-CPA-challenger. The challenger chooses a random bit $b \in \{0,1\}$ and returns the ciphertext $c \longleftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b)$. Then, $\mathcal{B}_2$ simply relays the ciphertext c to the SMP-adversary $\mathcal{A}$ who will output a bit $d \in \{0,1\}$, which in turn is forwarded by $\mathcal{B}_2$ to the IND-CPA-challenger.

It remains to be shown that $d = b$ with a non-negligible advantage, i.e., that

$$\left| \Pr\left[\mathbf{Exp}^{\text{ind-cpa}}_{\mathcal{B},\mathsf{KeyGen}}(\lambda) = 1 \right] - \frac{1}{2} \right| \text{ is non-negligible.}$$

By the assumption on $\mathcal{A}$, we know that $\mathcal{A}$'s advantage is non-negligible, namely $\Gamma(\lambda)$. Moreover, the ciphertext c is formatted as in SMP so $\mathcal{A}$ behaves as in the

SMP-game (it is either a fresh encryption of 1 or of a random message different from 1), meaning that $d = b$ with $\mathcal{A}$'s advantage $\Gamma(\lambda)$, i.e.,

$$\left|\Pr\left[\mathbf{Exp}^{\text{ind-cpa}}_{\mathcal{B},\mathsf{KeyGen}}(\lambda) = 1\right] - \frac{1}{2}\right| = \left|\Pr\left[\mathbf{Exp}^{\text{smp}}_{\mathcal{A},\mathsf{KeyGen}}(\lambda) = 1\right] - \frac{1}{2}\right| = \Gamma(\lambda).$$

This concludes the proof of the Theorem. □

B Example: Hardness of SMP does NOT imply IND-CPA Security

We construct a group homomorphic encryption scheme that is *not* IND-CPA secure, but whose corresponding SMP is hard. In a nutshell, the idea is to start with a IND-CPA secure scheme but to change the encryption process as follows. For a fixed message $m^* \neq 1$ the encryption process becomes deterministic for a significant probability (e.g., 1/2). An IND-CPA attacker can misuse this to easily distinguish encryptions of m^* from other ciphertexts. However, if the plaintext space is sufficiently large, the probability that the SMP-sampling algorithm chooses m^* is negligible, leaving the SMP still hard.

More precisely, let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure group homomorphic encryption scheme with a plaintext group $\mathcal{P}$ that is exponentially large in the security parameter such that the sampling algorithm, contained in the description of $\mathcal{P}$, samples according to the uniform distribution—for instance, this property is satisfied by the ElGamal cryptosystem [13]. By Theorem 1, we know that the SMP relative to KeyGen is hard. Now, the idea is to slightly modify $\mathcal{E}$ such that the corresponding SMP remains hard but the IND-CPA security can be easily broken. Therefore, we fix a public value $r^* \in \mathsf{Rnd}$, a public message $m^* \in \mathcal{P} \setminus \{1\}$, and construct a scheme $\mathcal{E}^*$ which is exactly the *same* as $\mathcal{E}$, except for the encryption algorithm. We denote the encryption algorithm of $\mathcal{E}^*$ by Enc^* and define it as follows:

Encryption. Enc^* takes the public key pk, a message m, and a random value $r \in \mathsf{Rnd}$ as input. Furthermore, it uniformly samples a random bit $b^* \in \{0, 1\}$. The output is defined as follows:

$$\mathsf{Enc}^*_{\mathsf{pk}}(m; r) := \begin{cases} \mathsf{Enc}_{\mathsf{pk}}(m; r^*) & \text{, if } m = m^* \text{ and } b^* = 0 \\ \mathsf{Enc}_{\mathsf{pk}}(m; r) & \text{, if } m = m^* \text{ and } b^* = 1 \\ \mathsf{Enc}_{\mathsf{pk}}(m; r) & \text{, otherwise.} \end{cases}$$

Recall that $r^* \in \mathsf{Rnd}$ and $m^* \in \mathcal{P}$ are fixed and public values corresponding to $\mathcal{E}^*$.

Our new scheme $\mathcal{E}^*$ certainly is *not* IND-CPA secure: Assume an adversary chooses two messages $m_0, m_1 \in \mathcal{P}$ where $m_0 = m^*$. Upon the retrieval of an encryption c of either of the two messages, the adversary checks whether $c = \mathsf{Enc}_{\mathsf{pk}}(m; r^*)$. If so, she knows that m_0 was encrypted. Otherwise she assumes that c is an encryption of message m_1. Her advantage is 1/4.

On the other hand, we see that the SMP corresponding to $\mathcal{E}^*$ is still hard: Recall that in the SMP game, the challenger flips a coin $b \in \{0, 1\}$. If $b =$

1, the challenger samples a randomly chosen message $m \xleftarrow{U} \mathcal{P}$ with $m \neq 1$ (recall that sampling from $\mathcal{P}$ is done according to the uniform distribution) and sends $c = \mathsf{Enc}^*_{\mathsf{pk}}(m)$ to an SMP-adversary. If $b = 0$, the challenger simply sends $c = \mathsf{Enc}^*_{\mathsf{pk}}(1)$ to the adversary. It is obvious that this SMP instance (using Enc^*) behaves exactly in the same way as our orginial SMP game (with Enc) corresponding to $\mathcal{E}$ if $b = 0$. But also if $b = 1$, it is clear that the advantage of an adversary in the SMP with Enc^* is negligibly close to the advantage of an adversary in the SMP with Enc. This is due to the fact that the plaintext space is exponentially large in the security parameter and the particular message m^* will only be chosen with a negligible probability. Therefore, the two games SMP with Enc^* and SMP with Enc are computationally indistinguishable, and so our SMP corresponding to $\mathcal{E}^*$ is hard.