ÖAW AUSTRIAN ACADEMY OF SCIENCES

RICAM – JOHANN RADON INSTITUTE FOR COMPUTATIONAL AND APPLIED MATHEMATICS

On homomorphic encryption using abelian groups: Classical security analysis

E. Agathocleous, V. Anupindi,
A. Bachmayr, C. Martindale,
R.Y.N. Nchiwo, M. Stanojkovski

RICAM-Report 2023-12

On homomorphic encryption using abelian groups: Classical security analysis

Eleni Agathocleous[1], Vishnupriya Anupindi[2], Annette Bachmayr[3], Chloe Martindale[4], Rahinatou Yuh Njah Nchiwo[5], and Mima Stanojkovski[6]

[1]CISPA Helmholtz Center for Information Security, [2]RICAM Austrian Academy of Sciences, [3]RWTH Aachen University, [4]University of Bristol, [5]Aalto University, [6]Università di Trento

Abstract. In [15], Leonardi and Ruiz-Lopez propose an additively homomorphic public-key encryption scheme whose security is expected to depend on the hardness of the *learning homomorphism with noise problem* (LHN). Choosing parameters for their primitive requires choosing three groups G, H, and K. In their paper, Leonardi and Ruiz-Lopez claim that, when G, H, and K are abelian, then their public-key cryptosystem is not quantum secure. In this paper, we study security for finite abelian groups G, H, and K in the classical case. Moreover, we study quantum attacks on instantiations with solvable groups.

Acknowledgements. We warmly thank the organizers of *Women in Numbers Europe 4* for putting together this team. We also wish to thank Chris Leonardi and Andrew Sutherland for helpful conversations around the contents of [15] and [23] respectively.

The first author was supported by the European Union's H2020 Programme under grant agreement number ERC-669891. The second author was supported by the Austrian Science Fund, Project P34808. The fifth author was supported by the Magnus Ehrnrooth grant 336005 and Academy of Finland grant 351271. The last author was partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 286237555 – TRR 195 and by the Italian program Rita Levi Montalcini for young researchers, Edition 2020.

Introduction

Homomorphic encryption is a method of encrypting plaintext that allows users to compute directly with the ciphertext. This has many interesting applications, including being able to engage in cloud computing without giving up your data to the owner of the cloud. Scientifically, the premise is easy to describe: Suppose that the plaintext and the ciphertext space both have a ring structure, and that we encrypt plaintext via a map between these spaces. If this map is a ring homomorphism, then this describes a *fully homomorphic encryption* scheme. Creating such a ring homomorphism that describes secure encryption (requiring, for example, that such a map should be efficiently computable and hard to invert) is, however, much harder than describing its properties. The closest the scientific community has come to constructing an example of fully homomorphic encryption is using maps based on (variants of) the *Learning With Errors* (LWE) problem from lattice-based cryptography [18, 16, 9, 6, 4]. However, all known LWE-based constructions are not truly fully homomorphic: Decrypting a message that was encrypted using LWE relies on the 'error' that was used in the encryption being small, and adding and multiplying encrypted messages together causes the error to grow. Once the error is too large, the data can

* Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf.
Date of this document: 28th February 2023.

no longer be decrypted, so methods such as bootstrapping need to be employed to correct this growth (see e.g. [6]). These methods may lead to practical fully homomorphic encryption in the future, but more research is needed.

In this paper, we explore an alternative approach for homomorphic encryption, introduced by Leonardi and Ruiz-Lopez in [15]. Their construction relies on the *Learning Homomorphisms with Noise* (LHN) problem introduced by Baumslag, Fazio, Nicolosi, Shpilrain, and Skeith in [1]: Roughly speaking, this is the problem of recovering a group homomorphism from the knowledge of the images of certain elements multiplied by noise. The focus of [15] is on the difficulty of constructing post-quantum secure instantiations of their primitive, but we believe the construction is interesting even in a classical setting. A big advantage of the LHN approach over a LWE approach is that the noise, which plays the role of the errors in LWE-based homomorphic encryption, does not grow with repeated computation in the way that the errors grow in lattice-based constructions. As such, there is no limitation on the number of additions that can be computed on encrypted data. However, it is less clear how to extend this construction to a multiplicative homomorphism. As such, the LHN approach is akin in some sense to the Benaloh [2] or Paillier [17] cryptosystems. In the nonabelian setting, Leonardi and Ruiz-Lopez's construction has some hopes of being post-quantum secure unlike the Benaloh or Paillier constructions, but as they explored already in their work this is nontrivial to instantiate, and our work only strengthens this claim as we show that even solvable groups may admit quantum attacks.

Our main contributions address **finite groups** and include:

1. Reducing the security of the abelian group instantiation of Leonardi and Ruiz-Lopez's public key homomorphic encryption scheme to the discrete logarithm problem in 2-groups (under certain plausible assumptions); this gives a polynomial-time classical attack if the 2-part of the relevant group is cyclic, and a practical classical attack if it is a product of a small number of cyclic groups.
2. Highlighting an abelian group instantiation of Leonardi and Ruiz-Lopez's homomorphic encryption scheme where there is no known practical classical attack, namely, the product of many cyclic 2-groups. This may be of interest to the community as a new example of unbounded additively homomorphic encryption.
3. Highlighting assumptions that need to be made in order to apply any discrete-logarithm derived attack, with a view to constructing (more) examples of groups on which there is no known practical attack on Leonardi and Ruiz-Lopez's homomorphic encryption scheme.
4. A description of a quantum attack on an instantiation with solvable groups, under certain assumptions.

The layout of this paper is as follows: In Section 1, we recap the public key homomorphic encryption scheme proposed by Leonardi and Ruiz-Lopez in [15]. In Section 2, we discuss some simple instantiations: The abelian case, the noiseless case, and we give some basic security requirements (including some recalled from [15]). In Section 3 we describe our reduction from the abelian group instantiation of the Leonardi Ruiz-Lopez primitive to the (extended) discrete logarithm problem, under certain assumptions. In Section 4, we describe some ways of instantiating the primitive with nonabelian groups to which our attack on abelian groups would also apply. In Section 5, we describe our quantum attack on instantiations with solvable groups, under certain assumptions on how such groups would be represented. In Section 6, we outline our plans for future work.

1 Preliminaries

In this section, we describe the public-key additive homomorphic encryption of Leonardi and Ruiz-Lopez [15, Sec. 5.2]; we will refer to this throughout this work as *Leonardi–Ruiz-Lopez encryption*.

Fix three finitely generated groups G, H, K and probability distributions ξ on G and χ on H. This data should be chosen in such a way that operations can be performed efficiently in the groups and we can sample from both distributions efficiently. A natural choice could be, for instance, to take G, H, K finite and ξ, χ to be uniform distributions. The groups G, H, K and the distributions ξ, χ are public. In the following sections we will mostly work with finite groups and we will always make it clear when this is the case.

For the **key generation**, Alice

- chooses efficiently computable secret homomorphisms $\varphi\colon G \to H$ and $\psi\colon H \to K$ such that she can efficiently sample from $\ker(\psi)$ and such that the center $\mathrm{Z}(H)$ of H is not contained in $\ker(\psi)$;
- chooses a natural number m;
- samples elements $g_1, \dots, g_m \in G$ via ξ and secret elements $h_1, \dots, h_m \in \ker(\psi)$ via χ;
- chooses an element $\tau \in \mathrm{Z}(H) \setminus \ker(\psi)$ of order 2.

Alice computes the public key as the set

$$\{(g_1, \varphi(g_1)h_1), \dots, (g_m, \varphi(g_m)h_m), \tau\}.$$

Note that, whereas the elements $g_1, \dots, g_m$ are public, both φ and $h_1, \dots, h_m$ are private (as are also ψ and $\ker(\psi)$).

For **encrypting** a one-bit message $\beta \in \{0, 1\}$, Bob chooses a natural number ℓ, then samples a word $w = w_1 \cdots w_\ell$ over the indices $\{1, \dots, m\}$ and using Alice's public key, he computes

$$(g, h') = (g_{w_1} \cdots g_{w_\ell}, \varphi(g_{w_1})h_{w_1} \cdots \varphi(g_{w_\ell})h_{w_\ell}).$$

He then sends $(g, h) = (g, h'\tau^\beta)$ to Alice.

For **decrypting** (g, h), Alice computes $\nu = \psi(\varphi(g))^{-1} \cdot \psi(h) \in K$ and deduces that the message β equals 0 if ν equals 1_K (and else, β equals 1).

To see that the decryption indeed produces the correct message β, recall that $h_1, \dots, h_m$ are contained in $\ker(\psi)$. Hence $\nu = \psi(\tau)^\beta$ and, since τ is not contained in $\ker(\psi)$, the element ν equals 1 if and only if β equals 0.

For the convenience of the reader, we give a schematic summary of the data described above:

Public information	Alice's private information	Bob's private information
G, H, K, ξ, χ	$\varphi : G \to H$	
$(g_1, \varphi(g_1)h_1), \dots, (g_m, \varphi(g_m)h_m)$	$\psi : H \to K$	$\beta \in \{0, 1\}$
$\tau \in \mathrm{Z}(H) \setminus \ker(\psi)$	$\ker(\psi)$	word w
$(g, h) = (g, h'\tau^\beta) \in G \times H$	$h_1, \dots, h_m \in \ker(\psi)$	

1.1 Homomorphic properties

The primary selling point of Leonardi–Ruiz-Lopez encryption is that it is *unbounded additive homomorphic*. We say that an encryption function E from plaintext to ciphertext space is *additive* if the plaintext space admits an additive operator $+$ and, given encryptions $E(\beta)$ and $E(\tilde{\beta})$ of messages β and $\tilde{\beta}$ respectively, one can compute a valid encryption $E(\beta+\tilde{\beta})$ of $\beta+\tilde{\beta}$ (without the knowledge of the plaintext $\beta+\tilde{\beta}$). We say that E is *unbounded additive homomorphic* if such additions can be performed an unbounded number of times without introducing systematic decryption failures.[1]

The reader may have observed that τ having order two is not necessary for successful decryption; this property is needed to make the encryption additive, as we now recall from [15].

Write the encryptions of β and $\tilde{\beta}$ sampled from $\{0,1\}$ as

$$(g, h'\tau^{\beta}) = (g_{w_1}\cdots g_{w_\ell}, \varphi(g_{w_1})h_{w_1}\cdots\varphi(g_{w_\ell})h_{w_\ell}\tau^{\beta})$$

and

$$(\tilde{g}, \tilde{h}'\tau^{\tilde{\beta}}) = (g_{\tilde{w}_1}\cdots g_{\tilde{w}_{\tilde{\ell}}}, \varphi(g_{\tilde{w}_1})h_{\tilde{w}_1}\cdots\varphi(g_{\tilde{w}_{\tilde{\ell}}})h_{\tilde{w}_{\tilde{\ell}}}\tau^{\tilde{\beta}})$$

respectively. Then, as τ is central in H and has order 2, we can construct a valid encryption of $\beta+\tilde{\beta}$ via the observation that

$$\begin{aligned}&\varphi(g_{w_1})h_{w_1}\cdots\varphi(g_{w_\ell})h_{w_\ell}\tau^{\beta}\varphi(g_{\tilde{w}_1})h_{\tilde{w}_1}\cdots\varphi(g_{\tilde{w}_{\tilde{\ell}}})h_{\tilde{w}_{\tilde{\ell}}}\tau^{\tilde{\beta}}\\ =&\varphi(g_{w_1})h_{w_1}\cdots\varphi(g_{w_\ell})h_{w_\ell}\varphi(g_{\tilde{w}_1})h_{\tilde{w}_1}\cdots\varphi(g_{\tilde{w}_{\tilde{\ell}}})h_{\tilde{w}_{\tilde{\ell}}}\tau^{\beta+\tilde{\beta}};\end{aligned}$$

this encryption is given by

$$(g, h'\tau^{\beta})(\tilde{g}, \tilde{h}'\tau^{\tilde{\beta}}) = (g\tilde{g}, h'\tau^{\beta}\tilde{h}'\tau^{\tilde{\beta}}) = (g\tilde{g}, h'\tilde{h}'\tau^{\beta+\tilde{\beta}}).$$

Remark 1 *Note that, for Leonardi–Ruiz-Lopez encryption to be fully homomorphic, it would also need to be* multiplicative*: That is, at the very least, given valid encryptions of $E(\beta)$ and $E(\tilde{\beta})$, we should be able to deduce a valid encryption of $\beta\tilde{\beta}$. It is not at all obvious if this is even possible. On an abstract level our encryption function from plaintext to ciphertext space maps*

$$E : \{0,1\} \to G \times H,$$

where the domain can be naturally endowed with a ring structure using addition and multiplication mod 2, but there seems to be no natural extra operation on $G \times H$ that would allow us to deduce a valid encryption of $\beta\tilde{\beta}$. We stress that in the case of LWE-based homomorphic encryption, both plaintext and ciphertext spaces come equipped with a ring structure, so the equivalent of our function E is generally taken to be a ring homomorphism.

The map E being a ring homomorphism is, however, not always strictly necessary to deduce a valid encryption of $\beta\tilde{\beta}$. If in future work we were to succeed in deducing a valid encryption of $\beta\tilde{\beta}$, we expect that this will only apply to a specific instantiation of Leonardi–Ruiz-Lopez encryption, not one for abstract groups, where there is more structure to be exploited.

1.2 Remarks on Leonardi–Ruiz-Lopez encryption

Some remarks on the construction above:

[1] The number of additions is bounded in, for example, LWE-based homomorphic encryption, where the error grows too large.

(R1) The underlying hard problem of this encryption scheme is described as the LHN-PKE problem, so named as it is based on the Learning Homomorphisms with Noise problem (LHN) but is adapted to this Public Key Encryption scheme (PKE).

Definition 2. *Let G, H, K, ξ, and χ be as above. We define the* LHN-PKE *problem for G, H, K, ξ, and χ to be: Given G, H, K, ξ, and χ, for any*

- *φ sampled uniformly at random from* $\mathrm{Hom}(G, H)$,
- *ψ sampled uniformly at random from the elements of* $\mathrm{Hom}(H, K)$ *whose kernel does not contain* $\mathrm{Z}(H)$,
- *$g_1, \ldots, g_m$ sampled from G using ξ,*
- *$h_1, \ldots, h_m$ sampled from* $\ker(\psi)$ *using χ,*
- *τ sampled from the order 2 elements of* $\mathrm{Z}(H) \setminus \ker(\psi)$ *using χ,*
- *β sampled uniformly at random from $\{0, 1\}$,*
- *small ℓ and word $w = w_1 \cdots w_\ell$ sampled uniformly at random from $\{1, \cdots, m\}^\ell$,*

recover β from the following information:

- $(g_1, \varphi(g_1)h_1), \ldots, (g_m, \varphi(g_m)h_m)$,
- τ,
- $g = g_{w_1} \cdots g_{w_\ell}$,
- $h = h'\tau^\beta = \varphi(g_{w_1})h_{w_1} \cdots \varphi(g_{w_\ell})h_{w_\ell}\tau^\beta$.

(R2) In order for the encryption and decryption to work, the assumptions that τ is central or of order 2 are not necessary. The reason we work under these assumptions is, as explained in Section 1.1, that in this case, the cryptosystem is unbounded additive homomorphic.

(R3) Once Alice has fixed the elements $g_1, \ldots, g_m$ and determined the public key, all computations inside G actually take place inside the subgroup $\langle g_1, \ldots, g_m \rangle$ that is generated by $g_1, \ldots, g_m$. So for cryptanalysis, we may and will assume that G is generated by $g_1, \ldots, g_m$, i.e. that $G = \langle g_1, \ldots, g_m \rangle$.

(R4) The work of [15] was inspired by [1], which introduces the Learning Homomorphisms with Noise problem in order to construct a symmetric primitive. However, the noise accumulates in the construction of [1] in a manner akin to the error growth in LWE constructions. Leonardi and Ruiz-Lopez also introduce a symmetric primitive in [15], but we focus on the PKE construction in this work.

(R5) The noise consists of the elements $h_1, \ldots, h_m$ that are mixed into the product h' in the second component h of the ciphertext. These elements are chosen to be in the kernel of ψ and therefore get erased during decryption. 'Being contained in the kernel' of ψ can thus be thought of as an equivalent of 'the error being small' in the LWE-based encryption of [18] or 'the noise being small' in LHN-based encryption of [1]. The strength of Leonardi–Ruiz-Lopez encryption is that the noise does not accumulate and will not lead to systematic decryption errors, since in the decryption process, we can erase the noise neatly by applying ψ.

2 Simple instantiations and security

In this section, we describe some simple instantiations of Leonardi–Ruiz-Lopez encryption. The abelian case is a central focus of this paper as it is much simpler to describe than the general case; the description below is for the reader who wishes only to understand the abelian case. We also describe the noiseless case in order to highlight the role that the noise plays in the encryption. Finally we discuss the requirements on the setup parameters of Leonardi–Ruiz-Lopez encryption in order to achieve security against some naive classical attacks, concluding this section with a list of properties that the groups must have for any classically secure instantiation.

2.1 The abelian case

If H is abelian, we can rewrite $(g,h) = (g_{w_1}\cdots g_{w_\ell}, \varphi(g_{w_1})h_{w_1}\cdots\varphi(g_{w_\ell})h_{w_\ell}\cdot\tau^\beta)$ as

$$(g,h) = (g_{w_1}\cdots g_{w_\ell}, \varphi(g_{w_1})\cdots\varphi(g_{w_\ell})h_{w_1}\cdots h_{w_\ell}\cdot\tau^\beta) = (g, \varphi(g)h_{w_1}\cdots h_{w_\ell}\cdot\tau^\beta).$$

If both G and H are abelian, it makes sense to switch to the following notation: Instead of choosing indices $w_1,\dots,w_\ell$, Bob just chooses non-negative integers $r_1,\dots,r_m$ and encrypts β to

$$(g,h) = (g_1^{r_1}\cdots g_m^{r_m}, \varphi(g_1^{r_1})\cdots\varphi(g_m^{r_m})h_1^{r_1}\cdots h_m^{r_m}\cdot\tau^\beta) = (g, \varphi(g)h_1^{r_1}\cdots h_m^{r_m}\cdot\tau^\beta).$$

This system has been claimed to not be quantum secure in [15, Section 8.2], cf. also Section 3.4, while we discuss security in the classical sense in Section 3.2. Moreover, in Section 4 we describe instantiations of the LHN-PKE problem in which G and H are nonabelian but the security reduces to the case in which they are.

2.2 The noiseless case

Let us assume that $h_1 = h_2 = \cdots = h_m = 1$. Then the public key consists of all pairs $(g_i, \varphi(g_i))$ together with τ and Bob would encrypt the message $\beta\in\{0,1\}$ to

$$(g,h) = (g_{w_1}\cdots g_{w_\ell}, \varphi(g_{w_1})\cdots\varphi(g_{w_\ell})\tau^\beta) = (g, \varphi(g)\tau^\beta).$$

Let Eve be an attacker who is aware of the fact that Alice decided to work in a noiseless setting. Then Eve knows all g_i's as well as their images $\varphi(g_i)$ from the public key. If she can write g as a product in $g_1,\dots,g_m$, then she can compute $\varphi(g)$. Knowing τ from the public key, Eve can then decrypt $(g,h) = (g,\varphi(g)\tau^\beta)$. Note that even if Eve did not use the same word $w_1\cdots w_\ell$ as Bob to write g as a product in $g_1,\dots,g_m$, she would nonetheless obtain the correct value of $\varphi(g)$.

Of course, finding such a word might still be a hard problem, even if $m=1$. For example, if G is the multiplicative group of a finite field and g_1 is a generator, finding a word in g_1 defining g is the same as solving the discrete logarithm problem, which is known to be hard for classical computers (though there are quantum algorithms to solve it, see [19, 20]). Alice, on the other hand, will probably have a closed form describing φ that does not require to write elements as products in $g_1,\dots,g_m$ when she applies φ in the decryption process (and similarly for ψ). In the $m=1$ example above, she would choose φ to take every element to a certain power.

For attacking the cryptosystem in the general case, a possible strategy is to construct attacks that reduce to the noiseless case. We will come back to such attacks in Section 2.3.

2.3 Security

In all that follows, let λ be the security parameter.[2]

2.3.1 Many homomorphisms First of all, note that an attacker who can guess both φ and ψ can decrypt in the same way Alice does. Denote the set of all possible choices for ψ by

$$\mathrm{Hom}(H,K)^- = \{\psi\in\mathrm{Hom}(H,K) : \mathrm{Z}(H) \not\subseteq \ker(\psi)\}.$$

To avert brute force attacks, the groups G,H,K should be chosen in such a way that $\mathrm{Hom}(G,H)$ and $\mathrm{Hom}(H,K)^-$ are of size at least $\Theta(2^\lambda)$,[3] and φ and ψ should be sampled uniformly at random from $\mathrm{Hom}(G,H)$ and $\mathrm{Hom}(H,K)^-$ respectively. This ensures that if an attacker guesses φ she succeeds with probability $2^{-\lambda}$, and similarly for ψ.

[2] Typically, we want any computations undertaken by the user to have complexity that is polynomial in λ, and an attacker who attempts to decrypt by guessing any unknowns should only succeed with probability at most $2^{-\lambda}$.

[3] We are using Bachmann-Landau notation for complexity, see for example Section 1.2.11.1 of [13].

2.3.2 Words As we already saw in the noiseless case, there are links between the LHN-PKE problem and the ability of an attacker to write g as an expression in the generators $g_1, \dots, g_m$. Assume for instance that Eve wants to decrypt the cyphertext (g, h). She knows that g is a product in $g_1, \dots, g_m$ and recall that $g_1, \dots, g_m$ are public. If she knows the exact expression of g as a product in $g_1, \dots, g_m$ that Bob used in the encryption process, then she can compute h' from the public key, erase it from $h = h'\tau^\beta$, and recover the message β. It is important to note that other than in the noiseless case, it is in general not enough to find *any* expression of g as a product in $g_1, \dots, g_m$, because that product will in general not produce the correct term h' yielding to a different accumulation of the noise. That is, the attacker needs to recover the *correct* word w, not just any expression of g in $g_1 \dots, g_m$. In the cryptanalysis we carry out in Section 3 for finite abelian G, H, and K, we give a reduction of LHN-PKE to the extended discrete logarithm problem for finite 2-groups; our reduction circumnavigates the issue of finding the correct word.

2.3.3 An attack on instances with few normal subgroups The idea behind the following attack is to replace Alice's secret $\psi : H \to K$ with some new $\bar{\psi} : H \to L$ erasing the noise without erasing τ^β (for instance L could be a quotient of H, as described below). A similar attack was also described in Section 7.2 of [15]. Assume that Eve knows a normal subgroup N of H that contains all elements $h_1, \dots, h_m$ but does not contain τ. She can then define $\bar{\psi}\colon H \to H/N$ as the natural projection and by applying $\bar{\psi}$ to all second coordinates of the elements $(g_i, \varphi(g_i)h_i)$ in the public key and to the second coordinate of the encrypted message (g, h) she can switch to the noiseless case; cf. Section 2.2. We deduce in particular, that there should be at least $\Theta(2^\lambda)$ normal subgroups in H, so that if an attacker guesses $\ker(\psi)$ she succeeds with probability $2^{-\lambda}$.

2.3.4 An attack on instances with weak normal subgroups Now suppose that an attacker can find a normal subgroup N of H that contains $\varphi(g_i)h_i$ for all $i = 1, \dots, m$ but does not contain τ (note that these elements are all public so it is easy to check these conditions). Then she can directly apply the projection $H \to H/N$ to the second coordinate in the encrypted message (g, h) and can deduce that β equals zero if and only if she obtained the neutral element in H/N. To avoid such an attack it seems that Alice should check, after the key generation process, whether the normal closure of $\langle \varphi(g_1)h_1, \dots, \varphi(g_m)h_m \rangle$ contains τ and if it doesn't she should choose a different key. For cryptanalysis, we may thus assume that H equals the normal closure of $\langle \varphi(g_1)h_1, \dots, \varphi(g_m)h_m \rangle$ (as otherwise, we can just work in this smaller group). In particular, if H is abelian, we may assume $H = \varphi(G) \ker \psi$.

2.3.5 A summary of the discussed security assumptions We conclude this section with a list of necessary properties for security deduced from the list of naive attacks above:

(S1) $\mathrm{Hom}(G, H)$ and $\mathrm{Hom}(H, K)^-$ are of size exponential in the security parameter;
(S2) finding the precise word w used to express g as a product in the g_i's in the encryption phase has complexity that is exponential in the security parameter;
(S3) the number of normal subgroups in H is exponential in the security parameter;
(S4) the normal closure of $\langle \varphi(g_1)h_1, \dots, \varphi(g_m)h_m \rangle$ contains τ.

3 Cryptanalysis in the finite abelian case

In this section, we discuss the hardness of LHN-PKE under the assumption that G and H are finite and that H is abelian. Leonardi and Ruiz-Lopez [15] dismissed the abelian instantiation due

to an argument that there should exist a polynomial-time quantum algorithm for the LHN-PKE problem; the reduction is more complex than is suggested in [15] but their statement is true as we show in Section 3.4. Nevertheless, the unbounded homomorphic property of the proposed cryptosystem is sufficiently powerful that a classically secure construction would also be of great interest to the cryptographic community. We show that, if H is abelian, under some mild assumptions that we introduce in Section 3.2, the LHN-PKE problem for G and H can be reduced to the extended discrete logarithm problem (cf. Definition 4) in some specific abelian 2-group.

Assume that G and H are finite and that H is abelian. Given the following public information

1. $\{(g_1, \varphi(g_1)h_1), \ldots, (g_m, \varphi(g_m)h_m), \tau\}$ and
2. $(g, h) = (g, h'\tau^\beta) \in G \times H$

one can proceed as follows:

- In case G is not abelian, the LHN-PKE problem for G, H, K, ξ, and χ is reduced to the LHN-PKE problem for $\overline{G} = G/[G, G], H, K, \xi$, and χ following Lemma 3: Here $[G, G]$ denotes the commutator subgroup of G, so $\overline{G}$ is abelian.
- In case G is also abelian and satisfies some additional assumptions presented in Section 3.2, the LHN-PKE problem for G, H, K, ξ, and χ is reduced to the eDLP problem in G, as defined in Section 3.2.
- In case G is also abelian and satisfies some additional assumptions presented in Section 3.2, the LHN-PKE problem for G, H, K, ξ, and χ reduces to the LHN-PKE problem for G_2, H_2, K, ξ, and χ, where G_2 and H_2 are the Sylow 2-subgroups of G and H, respectively.

To conclude, in Section 3.3 we discuss the genericity and limitations of the assumptions made in Section 3.2, as well as the impact of the reductions made.

3.1 A simplified setting for LHN-PKE when H is abelian

In this section we take H to be abelian and we show that, for cryptanalysis, one can can consider, instead of G, its abelianization $G/[G, G]$.

Lemma 3. *Assume H is abelian. Then the LHN-PKE problem for G, H, K, ξ, and χ is at most as hard as the LHN-PKE problem for $\overline{G} = G/[G, G], H, K, \xi$, and χ.*

Proof. Since H is abelian, the commutator subgroup $[G, G]$ of G is contained in the kernel of φ. Define $\overline{G} = G/[G, G]$. Then $\varphi\colon G \to H$ induces a well-defined homomorphism $\overline{\varphi}\colon \overline{G} \to H$ and any ciphertext (g, h) can be interpreted as the ciphertext $(\bar{g}, h)$ in the cryptosystem given by $\overline{\varphi}\colon \overline{G} \to H$, $\psi\colon H \to K$ as before and public key given by $(\overline{g_i}, \bar{\varphi}(\overline{g_i})h_i)$ for $i = 1, \ldots, m$ together with the same τ as before. Indeed, if Bob used the word $w_1 \cdots w_\ell$ to encrypt (g, h), then the same word gives rise to the encryption $(\bar{g}, h)$ in the new cryptosystem of the same message. □

3.2 Cryptanalysis for abelian groups

In Section 3.2 we discuss the LHN-PKE problem for G, H, and K finite and abelian. In each subsection, we explicitly mention under which assumptions from the following list we are working. For a group Γ and $\gamma_1, \ldots, \gamma_m \in \Gamma$, we are interested in the following properties:

(A1) Γ is abelian;
(A2) the largest positive odd factor of the order $|\Gamma|$ of Γ is known (or easily computable);
(A3) $\gamma_1, \ldots, \gamma_m \in \Gamma$ are such that $\Gamma = \langle\gamma_1\rangle \oplus \ldots \oplus \langle\gamma_m\rangle$;
(A4) The orders $|\gamma_1|, \ldots, |\gamma_m|$ of $\gamma_1, \ldots, \gamma_m$ are known (or easily computable).

Under Assumptions (A1), (A3), and (A4) the following problem is well-posed.

Definition 4. *The* extended discrete logarithm problem (eDLP) *for Γ is the problem of determining, for each $x \in \Gamma$, the unique vector*

$$(\alpha_1, \dots, \alpha_m) \in \mathbb{Z}/|\gamma_1|\mathbb{Z} \times \dots \times \mathbb{Z}/|\gamma_m|\mathbb{Z}$$

such that

$$x = \gamma_1^{\alpha_1} \cdots \gamma_m^{\alpha_m}.$$

Note that eDLP is just called *discrete logarithm problem* (DLP) in [23]. For a discussion of existing algorithms to solve it, we refer to Section 3.3.1.

3.2.1 From LHN-PKE to eDLP In this section, we work under Assumptions (A1), (A3), and (A4) of Section 3.2. More precisely, we assume that G and H are abelian, G is finite and satisfies $G = \langle g_1 \rangle \oplus \dots \oplus \langle g_m \rangle$, and the orders of $g_1, \dots, g_m$ are known.

Proposition 5. *The LHN-PKE problem for G, H, K, ξ, and χ is at most as hard as the* eDLP *problem in G.*

Proof. Let $r_1 \dots, r_m \in \mathbb{Z}$ be the non-negative integers chosen to write

$$g = g_1^{r_1} \cdots g_m^{r_m} \quad \text{and} \quad h = \varphi(g) h_1^{r_1} \cdots h_m^{r_m} \cdot \tau^{\beta}.$$

For each $i \in \{1, \dots, m\}$, set $\ell_i = \varphi(g_i) h_i$ and, given that the order of $\varphi(g_i)$ divides $|g_i|$, compute

$$\ell_i^{|g_i|} = (\varphi(g_i) h_i)^{|g_i|} = h_i^{|g_i|}.$$

As a consequence, the subgroup M of H that is generated by $Y = \{\ell_i^{|g_i|} : i = 1, \dots, m\}$ is contained in $\ker(\psi)$ and thus does not contain τ. Since the orders of the g_i's are known, the subgroup M can be easily determined.

We show that, eDLP being solvable in G yields also LHN-PKE being solvable in this context. To this end, let $s_1, \dots, s_m$ be such that, for any choice of i, one has $s_i \equiv r_i \bmod |g_i|$. From the public information, one easily computes

$$X = \prod_{i=1}^{m} (\ell_i^{s_i})^{-1} \cdot h = \prod_{i=1}^{m} \ell_i^{r_i - s_i} \cdot \tau^{\beta} = \prod_{i=1}^{m} \varphi(g_i)^{r_i - s_i} h_i^{r_i - s_i} \cdot \tau^{\beta}.$$

Since $r_i - s_i$ is a multiple of $|g_i|$, it is straightforward to see that $X \in M$ if and only if $\beta = 0$, i.e., working modulo the subgroup M, one can recover β from the public information. □

3.2.2 From abelian groups to 2-groups In this section, we assume that (A1) and (A2) from Section 3.2 hold, i.e. we assume that both G and H are finite abelian groups and that the odd parts of $|G|$ and $|H|$ are known. We let G_2 and H_2 denote the unique Sylow 2-subgroups of G and H, respectively. In the new language, τ and τ^{β} belong to H_2. In the following, we show that in the case of abelian groups, the LHN-PKE problem for the original data reduces to the LHN-PKE problem for the 2-parts G_2 and H_2 (with the induced data).

Theorem 6. *The LHN-PKE problem for G, H, K, ξ, and χ is at most as hard as the LHN-PKE problem for G_2, H_2, K, ξ, and χ.*

Proof. Write the orders of G and H as

$$|G| = 2^{n_G} q_G, \ |H| = 2^{n_H} q_H$$

where q_G and q_H are odd numbers. Define q as the least common multiple of q_G and q_H or any other odd common multiple (such as $q_G q_H$ which might be easier to compute). Then for any elements $g \in G$ and $h \in H$, the orders of g^q and h^q are both powers of 2, i.e. $g^q \in G_2$ and $h^q \in H_2$. Moreover, G and H being abelian, the assignment $x \to x^q$ defines homomorphisms $G \to G_2$ and $H \to H_2$.

Equip G_2 and H_2 with the induced distributions ξ and χ and write $\varphi_2 \colon G_2 \to H_2$ and $\psi_2 \colon H_2 \to K$ for the restrictions of φ and ψ to G_2 and H_2, respectively. Assume that β can be recovered from (G_2, H_2, K, ξ, χ). We show that β can be determined from (G, H, K, ξ, χ).

Recall that the pair $(g, h) = (g_1^{r_1} \cdots g_m^{r_m}, \varphi(g) h_1^{r_1} \cdots h_m^{r_m} \cdot \tau^\beta) = (g, h'\tau^\beta) \in G \times H$ is public. Raising both entries to their q-th power, one obtains

$$(g^q, h^q) = ((g_1^q)^{r_1} \cdots (g_m^q)^{r_m}, \varphi(g^q) \cdot (h_1^q)^{r_1} \cdots (h_m^q)^{r_m} \cdot (\tau^q)^\beta) = (g^q, (h')^q \tau^\beta) \in G_2 \times H_2.$$

From the last equation it is clear that the elements $g_1, \ldots, g_m \in G$ and $h_1, \ldots, h_m \in H$ are replaced with their q-th powers $g_1^q, \ldots, g_m^q \in G_2$ and $h_1^q, \ldots, h_m^q \in H_2$. Moreover, as $\tau = \tau^q$, it holds that $\tau \in H_2 \setminus \ker(\psi)$ and so the pair (τ, β) is preserved. By assumption β is determined from the data associated to the 2-parts and so the proof is complete. □

3.3 Assumptions and reductions

Assume in this section that G and H are finite and that H is abelian.

Thanks to (R3), for cryptanalysis purposes, we can replace G with $\tilde{G} = \langle g_1, \ldots, g_m \rangle$ and we do so. Moreover, in view of Lemma 3, the LHN-PKE problem on the pair $(\tilde{G}, H)$ is reduced to the LHN-PKE problem on the pair $(\overline{G}, H)$, where $\overline{G}$ denotes the abelianization of $\tilde{G}$. Assumption (A1) holds for the last pair and the elements $g_1, \ldots, g_m \in G$ are replaced with their images $\overline{g_1}, \ldots, \overline{g_m}$ in $\overline{G}$ and the homomorphism $\varphi : \tilde{G} \to H$ with the induced homomorphism $\overline{G} \to H$, which we identify with φ, for simplicity.

Assuming now that (A2) holds for H and $\overline{G}$, Theorem 6 allows to reduce the LHN-PKE problem on $(\overline{G}, H)$ to the LHN-PKE problem on $(\overline{G}_2, H_2)$, where $\overline{G}_2$ and H_2 denote the Sylow 2-subgroups of $\overline{G}$ and H. Here the elements $\overline{g_1}, \ldots, \overline{g_m}$ and $h_1, \ldots, h_m$ are replaced by

$$\overline{g_1}^q, \ldots, \overline{g_m}^q \quad \text{and} \quad h_1^q, \ldots, h_m^q$$

where q denotes the least common multiple of the odd parts of $|\overline{G}|$ and $|H|$.

Set $X_2 = \langle \overline{g_1}^q, \ldots, \overline{g_m}^q \rangle$ and suppose, at last, that (A3) and (A4) hold for X_2 namely that

$$X_2 = \langle \overline{g_1}^q \rangle \oplus \ldots \oplus \langle \overline{g_m}^q \rangle$$

and the sizes of the summands are known. Then, as a consequence of Proposition 5, the LHN-PKE problem for $(\overline{G}_2, H_2)$ is reduced to the eDLP problem in $\overline{G}_2$.

While assumptions (A1)–(A4) are natural assumptions to make when constructing groups that both admit efficient computation and satisfy the security requirements (S1)–(S4) of Section 2.3.5, there exist examples of groups where these assumptions are not satisfied or may be at odds with our security requirements. In this section we discuss assumptions (A1)–(A4) with a view towards constructing instantiations of Leonardi–Ruiz-Lopez encryption to which our classical attack does not apply, or at least is not polynomial-time.

3.3.1 The eDLP in finite abelian p-groups Let p be a prime number and let G be a finite abelian p-group given as

$$G = \langle g_1 \rangle \oplus \ldots \oplus \langle g_m \rangle,$$

where the orders of the summands are known. Let $e \in \mathbb{Z}$ be such that p^e is the exponent of G. Then, according to [23, Cor. 1], the eDLP in G can be solved using

$$O\left(\frac{\log(e+1)}{\log\log(e+2)} \log|G| + \frac{\log_p|G|}{m} p^{m/2}\right) \tag{1}$$

group operations on a classical computer. In particular, when m is polynomial in $\log\log|G|$, the eDLP has complexity polynomial in $\log|G|$. When $p = 2$, setting $n = \log_p|G|$, we deduce the following from the performance result in [23, Table 1]:

- When $m = 1, 2, 4, 8$, the counts are dominated by the first term of (1). This explains the initial cost decrease when m increases for a fixed n.
- Using Shank's algorithm, there is a possibility of improving the factor n/m by $\sqrt{n/m}$ though this is not relevant for applications: these normally require that n/m is close to 1.

To the best of our knowledge, there is no existing work on the eDLP that beats [23].

3.3.2 On the security assumptions (A2) and (A4) In Section 3.2.1 above we described how and under which assumptions the LHN-PKE problem for G, H, and K can be reduced to solving the eDLP in G. The necessity for setting assumptions (A2) and (A4) in particular, came from the fact that there are known examples of abelian groups, some of them already being used successfully in existing cryptographic protocols, that do not have to obey them.

One type of such groups are the known RSA groups. These are given in the form $(\mathbb{Z}/N\mathbb{Z})^\times$ where $N = pq$ is hard to factor. In this case, Alice would know the factorisation and hence the order of the group, but an adversary should not be able to compute it.

Another type of such groups, where even the creators of the cryptosystem might not know the group's order, is that of ideal class groups of imaginary quadratic fields. This is an interesting category of groups for cryptography since it allows one to work in a *trustless setup*. In other words, we do not need a trusted third party to generate groups of secure order, in contrast to cryptosystems that employ RSA groups for example, where a trusted third party needs to generate a secure, i.e. hard to factor, modulus $N \in \mathbb{N}$ for the groups $(\mathbb{Z}/N\mathbb{Z})^\times$. Despite the fact that neither the structure nor the order of the ideal class group is known, the group operation is efficient and the elements of the group have a compact representation, via reduced binary quadratic forms. An excellent reference for trusted unknown-order groups is the paper by Dobson, Galbraith and Smith [5]; they also take into account Sutherland's algorithm [22] and they propose new security parameters for cryptosystems that employ ideal class groups. In the same paper the authors also discuss other groups of unknown order that can be used, namely genus-3 Jacobians of hyperelliptic curves, initially introduced by Brent [3]. Even though these groups appear to have some computational advantages when compared to ideal class groups, these advantages exist only in theory for now since these genus-3 Jacobians have not yet been implemented.

3.3.3 Regarding Assumption (A3) In this section, we discuss Assumption (A3). For the sake of simplicity and in view of Section 3.2.2, we restrict to 2-groups but everything can be said similarly for arbitrary finite abelian groups using Theorem 6.

The case where Assumption (A3) holds for $\langle g_1, \ldots, g_m \rangle$, i.e., $g_1, \ldots, g_m$ are independent and satisfy $\langle g_1, \ldots, g_m \rangle = \langle g_1 \rangle \oplus \ldots \oplus \langle g_m \rangle$, seems to be the key case out of the following reasons. First

of all, when sampling $g_1, \ldots, g_m$ from G uniformly, it is very likely that these are independent at least if m is small in comparison with the number of cyclic factors in G. For example, if $G = C_2^\lambda$ is a direct product of cyclic groups of order 2, then the probability of sampling $g_1, \ldots, g_m$ that are independent is

$$\frac{(2^\lambda - 1)(2^\lambda - 2)(2^\lambda - 2^2) \cdots (2^\lambda - 2^{m-1})}{2^{m\lambda}}$$

which is very close to 1 if $m \ll \lambda$. In addition, the case where $g_1, \ldots, g_m$ are independent can be seen as the generic case and we expect that an attacker could use similar strategies as developed in Section 3.2 and design an attack for the dependent case. Indeed, if $g_1, \ldots, g_m$ are dependent, the attacker will obtain more information from the public key as in the independent case.

There is also another strategy for an attack if (A3) does not hold for G and the number of cyclic factors in H is small. More precisely, assume that $|\operatorname{Hom}(H, \mathbb{F}_2)|$ is sub-exponential. If Eve can write any element $g \in G$ as a product in $g_1, \ldots, g_m$ then she can recover the secret message β from the ciphertext (g, h) as follows. For every maximal subgroup M of H not containing τ (by assumption, there are only sub-exponentially many of these) convert h into an element $\tilde{h}$ in H/M. Decrypt $(g, \tilde{h}) \in G \times \tilde{H}$ as if the noise was erased completely in H/M (as explained in Section 2.2) and check on a number of self-encrypted messages if this provides a correct decryption function. Since $\ker(\psi) \neq H$, there will always be a maximal subgroup M of H that contains $h_1 \ldots, h_m$ but does not contain τ, so eventually this procedure will indeed provide a decrpytion function.

3.4 Comparison with the quantum attack

Throughout this section, assume that (A3) holds for the abelian group G. Even though we only did cryptanalysis for finite groups until now, we show in the following two subsections how one can perform quantum attacks when G is either torsion-free or finite. The mixed case can be considered as a combination of the two cases: first dealing with the free part of the group and then recovering β as explained in the finite case.

3.4.1 The torsion-free case Assume as in [15, Sec. 8.2], that $G = \langle g_1 \rangle \oplus \ldots \oplus \langle g_m \rangle$ is isomorphic to $\mathbb{Z}^m$, i.e. the orders $|g_i|$ are all infinite. We briefly recall the discussion from [15, Sec. 8.2]. To this end, let $f : \mathbb{Z}^{m+1} \to G$ be defined by

$$(a_1, \ldots, a_{m+1}) \longmapsto g^{a_{m+1}} g_1^{a_1} \cdots g_m^{a_m}.$$

Then the kernel of f is equal to $\langle (r_1, \ldots, r_m, 1) \rangle$ and it can be determined, using Shor's algorithm [20], in quantum polynomial time in m; cf. [10, 24]. Once $(r_1, \ldots, r_m, 1)$ is known, it is easy to recover β from the encrypted message (g, h) and the public information.

3.4.2 The torsion case Assume in this section that G is finite and that (A4) holds. Let, moreover, $f : \mathbb{Z}^{m+1} \to G$ be defined by

$$(a_1, \ldots, a_{m+1}) \longmapsto g^{a_{m+1}} g_1^{a_1} \cdots g_m^{a_m}.$$

Then a set of generators of $\ker(f)$ can be determined in time polynomial in $\log |G|$ on a quantum computer [12, 20, 21]; see also [8]. Note that $\ker(f)$ contains $|g_1|\mathbb{Z} \times \ldots \times |g_m|\mathbb{Z} \times |g|\mathbb{Z}$.

Let now $(s_1, \ldots, s_{m+1})$ be one of the generators found. Then $g = g_1^{-s_1 s_{m+1}} \cdots g_m^{-s_m s_{m+1}}$ and so it follows that

$$g_1^{r_1 + s_1 s_{m+1}} \cdots g_m^{r_m + s_m s_{m+1}} = 1.$$

This is the same as saying that, for each $1 \leq i \leq m$, one has

$$r_i \equiv -s_i s_{m+1} \bmod |g_i|.$$

Modding out H by the subgroup generated by all $h_i^{|g_i|}$ one can recover β and thus solve the LHN-PKE problem; cf. Proposition 5.

4 An attack that reduces to the abelian case

A general strategy for an attacker to solve the LHN-PKE problem could be to convert the H-part of the ciphertext as follows. Assume that Eve has access to the ciphertext

$$(g, h) = (g_{w_1} \cdots g_{w_\ell}, \varphi(g_{w_1})h_{w_1} \cdots \varphi(g_{w_\ell})h_{w_\ell}\tau^\beta) \in G \times H$$

encrypted by Bob using Alice's public key

$$\{(g_1, \varphi(g_1)h_1), \ldots, (g_m, \varphi(g_m)h_m), \tau\}.$$

Eve can then choose a homomorphism $\vartheta\colon H \to \overline{H}$ for some group $\overline{H}$ such that $\vartheta(\tau) \neq 1$ and compute $\vartheta(h)$. Her new pair $(g, \overline{h}) = (g, \vartheta(h))$ is then of the form

$$(g_{w_1} \cdot \ldots \cdot g_{w_\ell}, \overline{\varphi}(g_{w_1})\overline{h}_{w_1} \cdot \ldots \cdot \overline{\varphi}(g_{w_\ell})\overline{h}_{w_\ell}\overline{\tau}^\beta)$$

where, for all i, we set $\overline{h_i} = \vartheta(h_i)$ and write $\overline{\varphi} = \vartheta \circ \varphi$ and $\overline{\tau} = \vartheta(\tau)$. Since $\overline{\tau} \neq 1$, this pair still contains the information on β and if ϑ is chosen cleverly, it might be much simpler to deduce β from (g, h'). Note that in general, we cannot define a suitable counterpart $\overline{\psi}$ of ψ here, so the information that τ is not contained in the kernel of ψ cannot be directly converted into a statement on $\overline{\tau}$ and has to be considered individually (if necessary).

A special case of this strategy is to reduce, if possible, to an abelian group $\overline{H}$, i.e., the goal is to eventually apply the attack that we describe in Section 3 even when G and H are nonabelian.

Suppose that, given τ and H, Eve is able to find an abelian group $\overline{H}$ and an efficiently computable homomorphism $\vartheta : H \to \overline{H}$ such that $\vartheta(\tau) \neq 1$ and such that $\vartheta(\tau)$ is not contained in the subgroup generated by certain powers of $\vartheta(\varphi(g_i)h_i)$. This condition can be checked using the public key and we will specify below which powers are sufficient. Let $\overline{G} = G/[G, G]$ be the abelianization of G and $\bar{\varphi}\colon \overline{G} \to \overline{H}$ be the homomorphism obtained from $\vartheta \circ \varphi\colon G \to \overline{H}$ by reducing modulo $[G, G]$. For all i, define $\bar{h}_i = \vartheta(h_i)$, set $\bar{\tau} = \vartheta(\tau)$ and, for all g in G, let $\bar{g} \in \overline{G}$ be the image of g in $\overline{G}$.

Eve first replaces the encrypted message by

$$(\bar{g}, \vartheta(h)) = (\bar{g}_{w_1} \cdots \bar{g}_{w_\ell}, \bar{\varphi}(\bar{g}_{w_1})\bar{h}_{w_1} \cdots \bar{\varphi}(\bar{g}_{w_\ell})\bar{h}_{w_\ell}\bar{\tau}^\beta) \in \overline{G} \times \overline{H}$$

in order to work inside abelian groups $\overline{G}$ and $\overline{H}$. Now she proceeds in a similar way as in the proof of Theorem 6: First, she computes the group orders of $\overline{G}$ and $\overline{H}$ and finds the least common multiple q of their odd parts. Then she converts the tuple $(\bar{g}, \vartheta(h))$ above into a tuple with entries inside the 2-Sylow subgroups $\overline{G}_2$, $\overline{H}_2$ of $\overline{G}$ and $\overline{H}$ by taking both entries to their q-th powers:

$$(\bar{g}^q, \vartheta(h)^q) = (\bar{g}^q_{w_1} \cdots \bar{g}^q_{w_\ell}, \bar{\varphi}(\bar{g}^q_{w_1})\bar{h}^q_{w_1} \cdots \bar{\varphi}(\bar{g}^q_{w_\ell})\bar{h}^q_{w_\ell}\bar{\tau}^\beta) \in \overline{G}_2 \times \overline{H}_2.$$

Finally, Eve can proceed in a similar way as in the proof of Proposition 5: For all i define $\bar{l}_i = (\bar{\varphi}(\bar{g}_i)\bar{h}_i)^q = \vartheta(\varphi(g_i)h_i)^q$ and let β_i be the order of $\varphi(\bar{g}_i)$. Eve can compute the elements $\bar{l}_i$ and the numbers β_i using the public key. Then $(\bar{l}_i)^{\beta_i} = (\bar{h}_i^q)^{\beta_i}$ can also be computed from the public key. If $\bar{\tau}$ is not contained in the subgroup M generated by $(\bar{l}_1)^{\beta_1}, \ldots, (\bar{l}_m)^{\beta_m}$, then working inside $\overline{H}/M$ can reveal the value of β as in the proof of Proposition 5 under similar assumptions as stated there.

Remark 7 *Observe that the above attack applies to any finite group H if $\tau \notin [H,H]$ and the image of τ in $H/[H,H]$ is not contained in the subgroup generated by the images of certain powers of $\varphi(g_i)h_i$. In particular, it is advisable (but likely not sufficient) to sample τ from $[H,H]$.*

5 Normal forms and the solvable case

In this section we consider the case where the platform groups are finite and solvable and give evidence of why, given the efficiency constraints attached to the system, the groups should not be expected to provide secure postquantum cryptosystems (in analogy to Section 3.4).

For the background on finite solvable groups we refer to the very friendly [11, Ch. 3].

Definition 8. *For a finite group Γ, the* derived series *of Γ is the series $(\Gamma^{(i)})_{i\geq 1}$ defined inductively by*

$$\Gamma^{(1)} = \Gamma \quad \text{and} \quad \Gamma^{(i+1)} = [\Gamma^{(i)}, \Gamma^{(i)}].$$

If for some index m the group $\Gamma^{(m)}$ is trivial, then Γ is said to be solvable.

In a finite solvable group Γ each quotient $\Gamma^{(i)}/\Gamma^{(i+1)}$ is abelian. Moreover, only finitely many such quotients are nontrivial and, if one is trivial, all subsequent ones are, too.

Until the end of this section, assume that H is solvable. Then, for cryptanalysis purposes and in analogy with Lemma 3, we assume without loss of generality that G is also solvable. We let s be the *derived length* of H, i.e. s is such that $H^{(s+1)} = 1$ but $H^{(s)} \neq 1$. Then, without loss of generality, we assume that $G^{(s+1)} = 1$. We remark that, if $s = 1$, then G and H are abelian.

Remark 9 (Efficient communication and computation) *Alice and Bob, as part of their message exchange, need to be able to communicate elements and perform operations in the groups efficiently. An often favourable approach (also proposed in [1, § 4.2]) is that of using* normal forms *of elements with respect to a* polycyclic presentation*, cf. [7, Ch. 2]. For instance, when working with G and H abelian, (A3) and (A4) holding for G is almost the same as saying that G is given by a polycyclic presentation and the expression $g = g_1^{r_1} \cdots g_m^{r_m}$ is the normal form of g with respect to this presentation. The word "almost" in the previous sentence is there to stress that the r_i's are not uniquely identified by their class modulo $|g_i|$, which in turn is what happens for normal forms (see below).*

We briefly explain here what it means for an element g of the solvable group G to be communicated in a *normal form* with respect to a polycylic presentation respecting the derived filtration. To do so, for each $i \in \{1, \ldots, s\}$:

(*a*) let m_i denote the minimum number of generators of $G^{(i)}/G^{(i+1)}$,

(*b*) let $g_{i1}, \ldots, g_{im_i}$ be elements of $G^{(i)}$ such that

$$G^{(i)}/G^{(i+1)} = \langle g_{i1}G^{(i+1)}\rangle \oplus \cdots \oplus \langle g_{im_i}G^{(i+1)}\rangle,$$

(*c*) for each $j \in \{1, \ldots, m_i\}$ let o_{ij} denote the order of g_{ij} modulo $G^{(i+1)}$.

Then any element $g \in G$ can be uniquely represented by a vector

$$\delta = (\delta_{11}, \ldots, \delta_{1m_1}, \delta_{21}, \ldots, \delta_{2m_2}, \ldots, \delta_{s1}, \ldots, \delta_{sm_s})$$

of integers $0 \leq \delta_{ij} < o_{ij}$ of length $n = m_1 + \ldots + m_s$ such that

$$g = \prod_{i=1}^{s} \prod_{j=1}^{m_i} g_{ij}^{\delta_{ij}},$$

which is precisely the normal form of g with respect to the chosen generators $\{g_{ij}\}$.

Note that the data (a)-(b)-(c) mentioned above should be public for Alice and Bob to be able to share the elements with each other. If the chosen generators fit into a *polycyclic sequence*, then the group operation is performed through the *collection process* [7, § 2.2]. It should be mentioned that, though polycyclic presentations generally yield a good practical performance, it is, to the best of our knowledge, not clear whether multiplication in these presentations can always be performed in polynomial time [14]. Note that the expression of h' also depends on the vector δ:

$$h' = \prod_{i=1}^{s} \prod_{j=1}^{m_i} (\varphi(g_{ij}) h_{ij})^{\delta_{ij}}.$$

Shor's algorithm, applied on each level $G^{(i)}/G^{(i+1)}$, is polynomial in the log of the size of this quotient. Moreover, Shor's algorithm really does determine the vector δ of g because of the condition $0 \leq \delta_{ij} < o_{ij}$. In particular Eve can recover the vector δ of g and use it to compute

$$\tau^{\beta} = \left(\prod_{i=1}^{s} \prod_{j=1}^{m_i} (\varphi(g_{ij}) h_{ij})^{\delta_{ij}} \right)^{-1} \cdot h.$$

The complexity of this algorithm on each quotient $G^{(i)}/G^{(i+1)}$ is polynomial in $\log |G^{(i)}/G^{(i+1)}|$, which makes the overall complexity to be polynomial in $\log |G|$. There is obviously no separation in complexity between the algorithm being run by the adversary, and the one being run by the user. This implies that for the system to be secure, we would need to have $|G| > 2^{2^{\lambda}}$, making it hard to say if it would even be possible to represent elements of G in a computer.

Remark 10 *Given that the quotients of consecutive elements of the derived series are abelian, it is natural to ask whether the classical attack we designed for abelian groups could be generalised to an attack in the solvable context. It seems, however, that the nonabelian solvable case is substantially different from the abelian one. Among the limitations are:*

- *τ could for instance belong to $H^{(s)}$ while $g_1, \ldots, g_m$ will typically live in $H^{(1)} \setminus H^{(2)}$ (in the general setting we are indeed not necessarily working with normal forms);*
- *without knowing φ, it is not at all clear at which depths in the derived filtration $\varphi(g_1), \ldots \varphi(g_m)$ are to be found in H (as publicly given only with noise);*
- *dealing with quotients is more delicate as one always has to consider normal closures of subgroups.*

6 Future work

In future work, we plan to consider several types of finite groups G, H, and K for instantiating Leonardi–Ruiz-Lopez encryption and explore whether we can either construct attacks for the corresponding cryptosystems or prove security results.

A first candidate would be the group C_2^{λ} for all groups G, H, K. As none of the classical attacks presented in this paper apply in this case, Leonardi–Ruiz-Lopez encryption might prove to be classically secure for this choice. Other abelian candidates are the RSA groups and ideal class groups mentioned in Section 3.3.2.

As a first nonabelian example, we plan to work with certain p-groups and use strategies that allow us to circumnavigate the attacks presented in Section 5. In particular, it would be beneficial to work with presentations of groups that are not based on normal forms and yet allow efficient computation. The advantage of working with nonabelian groups is that it may be possible to construct a post-quantum additive homomorphic cryptosystem.

References

[1] G. Baumslag, N. Fazio, A. R. Nicolosi, V. Shpilrain, and W. E. Skeith, III. Generalized learning problems and applications to non-commutative cryptography (extended abstract). In *Provable security*, volume 6980 of *Lecture Notes in Comput. Sci.*, pages 324–339. Springer, Heidelberg, 2011.

[2] J. Benaloh. Dense probabilistic encryption. In *Proceedings of the Workshop on Selected Areas of Cryptography*, page 120–128, 1994.

[3] R. P. Brent. *Public Key Cryptography with a Group of Unknown Order*. Tech. Rep. Oxford University, 2000.

[4] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Tfhe: Fast fully homomorphic encryptionover the torus. *Journal of Cryptology*, 33:34–91, 2020.

[5] S. Dobson, S. D. Galbraith, and B. Smith. Trustless unknown-order groups. Cryptology ePrint Archive, Paper 2020/196, 2020. https://eprint.iacr.org/2020/196.

[6] L. Ducas and D. Micciancio. Fhew: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology - EUROCRYPT 2015*, Lecture Notes in Computer Science, pages 617–640, 2015.

[7] B. Eick. Algorithms for polycyclic groups, 2000.

[8] M. Ettinger, P. Hø yer, and E. Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Inform. Process. Lett.*, 91(1):43–48, 2004.

[9] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - Crypto 2013*, Lecture Notes in Computer Science, pages 75–92, 2013.

[10] M. Imran and G. Ivanyos. An exact quantum hidden subgroup algorithm and applications to solvable groups. *Quantum Inf. Comput.*, 22(9-10):770–789, 2022.

[11] I. M. Isaacs. *Finite group theory*, volume 92 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2008.

[12] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Uspekhi Mat. Nauk*, 52(6(318)):53–112, 1997.

[13] D. E. Knuth. *The art of computer programming. Volume 1, Fundamental Algorithms*. 1997.

[14] C. R. Leedham-Green and L. H. Soicher. Symbolic collection using deep thought. *LMS J. Comput. Math.*, 1:9–24, 1998.

[15] C. Leonardi and L. Ruiz-Lopez. Homomorphism learning problems and its applications to public-key cryptography. Cryptology ePrint Archive, Report 2019/717, 2019. https://ia.cr/2019/717.

[16] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):1–35, 2013.

[17] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'99)*, volume 1592, pages 223–238, 1999.

[18] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[19] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science (Santa Fe, NM, 1994)*, pages 124–134. IEEE Comput. Soc. Press, Los Alamitos, CA, 1994.

[20] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[21] D. R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.

[22] A. V. Sutherland. Order computations in generic groups, 2007. PhD Thesis, Massachusetts Institute of Technology. https://math.mit.edu/~drew/sutherland-phd.pdf.

[23] A. V. Sutherland. Structure computation and discrete logarithms in finite abelian p-groups. *Math. Comp.*, 80(273):477–500, 2011.

[24] J. Watrous. Quantum algorithms for solvable groups. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 60–67. ACM, New York, 2001.