Generic Homomorphic Undeniable Signatures

Jean Monnerat * and Serge Vaudenay **

EPFL, Switzerland
http://lasecwww.epfl.ch

Abstract. We introduce a new computational problem related to the interpolation of group homomorphisms which generalizes many famous cryptographic problems including discrete logarithm, Diffie-Hellman, and RSA. As an application, we propose a generic undeniable signature scheme which generalizes the MOVA schemes. Our scheme is generic in the sense that we transform a private group homomorphism from public groups G to H (the order of H being public) into an undeniable signature scheme. It is provably secure in the random oracle model provided that the interpolation problem is hard and it offers the advantage of making the signature size arbitrarily short (depending on a security level). We (im)prove some security results from MOVA. We also propose a new example with complexity similar to RSA and with 3-byte signatures.

1 Introduction

An undeniable signature scheme is similar to a classical digital signature except that the recipient of a message cannot verify its validity alone: he needs to interact with the signer in order to be convinced of the validity of the signature. This opposes to the so called universal verifiability of classical digital signatures where anybody knowing the signer's public key is able to verify the signature at any time. In some applications such as signing a contract, it is desirable to keep the signer's privacy by limiting the ability to verify this signature. However, an undeniable signature does not abandon the non-repudiation property. Indeed, in case of a dispute, the signer could be compelled by an authority to prove the invalidity of a signature, otherwise this would be considered as an attempt of denying a valid signature. An undeniable signature scheme is composed of a signature generation algorithm, a confirmation protocol to prove the validity of a signature, and a denial protocol to prove the invalidity of an invalid signature.

Since the invention of the first undeniable signature scheme proposed by Chaum and van Antwerpen [9], a certain amount of work has been dedicated to its development and different improvements [5,7,8,11,12]. Until the proposition of an undeniable signature scheme based on RSA by Gennaro et al. [15], all

* Supported in part by a grant of the Swiss National Science Foundation, 200021-101453/1.

** Supported in part by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

previous undeniable signatures were based on the discrete logarithm problem. More recently, three undeniable signatures based on different problems have been proposed. The first one is based on pairings [18], the second one is based on a quadratic field [4], and the third one (MOVA) is based on characters [19].

In traditional digital signature schemes, the security collapses when the signature is too short because of universal verifiability: an attacker can try to guess a signature until it is valid in order to forge it. One advantage of undeniable signatures is that the security smoothly decreases with the signature length. As an example, we can think of 20-bit signatures which cannot be forged but with a probability of success of 2^{-20}. The forger can increase it in an on-line attack, but this can easily be detected and thwarted. So, undeniable signatures could in principle be arbitrarily small e.g. as small as a MAC, although no such signatures were proposed so far except MOVA signatures.

In this paper, we provide a new computational problem called Group Homomorphism Interpolation (GHI) problem whose solution consists in finding the image of a given point under an homomorphism which interpolates some given points. This generalizes and improves the MOVA scheme based on characters. Section 2 provides some theoretical results about the GHI problem. Section 3 contains several interactive proof protocols and some related security results that will be used for our undeniable signature from Section 4. Section 5 is devoted to a new example and further discussions. Finally, Section 6 concludes.

2 The Group Homomorphism Interpolation Problem

2.1 Problem Definitions

Given two Abelian groups G, H, and $S := \{(x_1, y_1), \ldots, (x_s, y_s)\} \subseteq G \times H$, we say that the set of points S *interpolates in a group homomorphism* if there exists a group homomorphism $f : G \longrightarrow H$ such that $f(x_i) = y_i$ for $i = 1, \ldots, s$. We say that a set of points $B \subseteq G \times H$ *interpolates in a group homomorphism with another set of points* $A \subseteq G \times H$ if $A \cup B$ interpolates in a group homomorphism. We state here the Group Homomorphism Interpolation problem (GHI problem) and its decisional problem (GHID problem).

S-GHI Problem (Group Homomorphism Interpolation Problem)
Parameters: two Abelian groups G and H, a set of s points $S \subseteq G \times H$.
Input: $x \in G$.
Problem: find $y \in H$ such that (x, y) interpolates with S in a group homomorphism.

S-GHID Problem (GHI Decisional Problem)
Parameters: two Abelian groups G and H, a set of s points $S \subseteq G \times H$.
Input: a point $(x, y) \in G \times H$.
Problem: does (x, y) interpolate with S in a group homomorphism?

We also consider the following problems.

d**-MGGD Problem** (Modular Group Generation Decisional Problem)
Parameters: an Abelian group G, an integer d.
Input: a set of values $S_1 = \{x_1, \dots, x_s\} \subseteq G$.
Problem: does S_1 modulo dG span G/dG.

(d, S_1)**-MSR Problem** (Modular System Representation Problem)
Parameters: an Abelian group G, a set $S_1 = \{x_1, \dots, x_s\} \subseteq G$, an integer d.
Input: $x \in G$.
Problem: find $a_1, \dots, a_s \in \mathbf{Z}$ such that $x \in a_1 x_1 + \dots + a_s x_s + dG$.

d**-Root Problem** (dth Root Problem)
Parameters: an Abelian group G, an integer d.
Input: $x \in G$.
Problem: find $r \in G$ such that $x = dr$.

2.2 Preliminaries

Here is a first straightforward condition to solve the GHID problem.

Lemma 1. *Let G, H be two finite Abelian groups. We denote by d the order of H. The set $S = \{(x_1, y_1), \dots, (x_s, y_s)\} \subseteq G \times H$ interpolates in a group homomorphism if and only if for any $a_1, \dots, a_s \in \mathbf{Z}$ such that $a_1 x_1 + \dots + a_s x_s \in dG$ we have $a_1 y_1 + \dots + a_s y_s = 0$.*

Let us now consider uniqueness criteria. We first notice that when the x-coordinates of points in S modulo dG generate G/dG (hence satisfy the MGGD problem), then there is at most one interpolating homomorphism. The following result says that this is a necessary condition as well.

Lemma 2. *Let G, H be two finite Abelian groups. We denote d the order of H. Let $x_1, \dots, x_s \in G$ which span G'. The following properties are equivalent. In this case, we say that $x_1, \dots, x_s$ H-generate G.*

1. *For all $y_1, \dots, y_s \in H$, there exists at most one group homomorphism $f : G \longrightarrow H$ such that $f(x_i) = y_i$ for all $i = 1, \dots, s$.*
2. *There exists a unique group homomorphism $\varphi : G \longrightarrow H$ such that $\varphi(x_i) = 0$ for $i = 1, \dots, s$, namely $\varphi = 0$.*
3. *The set $\mathrm{Hom}(G/G', H)$ of all group homomorphisms from G/G' to H is restricted to $\{0\}$.*
4. *$\gcd(\#(G/G'), d) = 1$.*
5. *$G' + dG = G$.*

Note that the criterion 4 suggests that H is only involved by the prime factors of its order. In what follows the smallest prime factor p will be important. Note that if $G = H$, these criteria mean that $x_1, \dots, x_s$ generate G.

We can often meet the GHI and GHID problems in cryptography as the following examples suggest.

Example 3. We take a cyclic group G of order q, $H = \mathbf{Z}_q$, and a generator g of G. The set $S = \{(g, 1)\}$ interpolates in a unique group homomorphism, and the GHI problem is exactly the discrete logarithm problem.

Example 4. We take a cyclic group $G = H$, and a generator g of G. For any $a \in \mathbf{Z}$, $S = \{(g, ag)\}$ interpolates in a unique group homomorphism: the exponentiation to the power a. The GHI and GHID problems are exactly the Diffie-Hellman problem [13] and the Diffie-Hellman Decisional problem.

Example 5. Let $n = pq$ such that p, q are different odd primes and $H = \{-1, +1\}$. We let $x_1, x_2 \in \mathbf{Z}_n^*$ be such that x_1 is a quadratic residue modulo p and not modulo q, and that x_2 is a quadratic residue modulo q, and not modulo p. We notice that $S = \{(x_1, 1), (x_2, -1)\}$ interpolates in a unique group homomorphism which is $(\cdot/p)$. Since it is easy to compute $(\cdot/n)$, the quadratic residuosity problem [16] with the information x_1 and x_2 is equivalent to the GHI and GHID problems.

Example 6. Here, we consider the well known RSA cryptosystem [21]. Let $n = pq$ be an RSA modulus and $G = H = \mathbf{Z}_n^*$. Let $f : \mathbf{Z}_n^* \to \mathbf{Z}_n^*$ be defined by $f(x) = x^e \bmod n$ for an exponent e such that $\gcd(e, \varphi(n)) = 1$ [21]. Given enough many pairs $(x_i^e \bmod n, x_i) \in \mathbf{Z}_n^* \times \mathbf{Z}_n^*$, $i = 1, \ldots, s$, for the first coordinates to generate $\mathbf{Z}_n^*$, the RSA decryption problem is solved by a GHI oracle.This application of GHI problem to the decryption problem can be adapted to every homomorphic encryption scheme, e.g. Paillier [20].

Example 7. Given $d \in \{2, 3, 4\}$ and given an integer n such that d divides $\varphi(n)$, we let $G = \mathbf{Z}_n^*$ and $H = \mathbf{Z}_d$. The GHI problem is the MOVA^d problem [19].

Example 8. We show here how we can apply the GHI problem to the Bilinear Diffie-Hellman Problem (BDHP). Let $\hat{e} : \mathbf{G}_1 \times \mathbf{G}_1 \to \mathbf{G}_2$ be a bilinear, non-degenerate and computable mapping, where $\mathbf{G_1}$ and $\mathbf{G_2}$ are cyclic groups of order a large prime p. Let P be a generator of $\mathbf{G}_1$, we can state the BDHP as follows: given three random elements aP, bP and $cP \in \mathbf{G}_1$, compute $\hat{e}(P, P)^{abc}$. ($\mathbf{G_1}$ resp. $\mathbf{G}_2$ is written additively resp. multiplicatively.) BDHP is equivalent to GHI problem with $S = \{(P, \hat{e}(aP, bP))\}$ and $x_1 = cP$.

Note that Examples 4,5,6,7,8, include trapdoors in order to interpolate the group homomorphism. Except Examples 4,8, they further include trapdoors in order to solve the MSR problem. Also note that the order d of H is publicly known in Examples 3,4,5,7,8. It can further be quite small in Examples 5,7. In what follows we focus on publicly known d and on trapdoor homomorphisms. We will also consider the following example inspired by [1].

Example 9. Let $n = pq$ such that $p = rd + 1$ and q are prime, $\gcd(r, d) = 1$, $\gcd(q - 1, d) = 1$, with d small prime. We take $G = \mathbf{Z}_n^*$ and $H = \mathbf{Z}_d$. We can easily compute a group homomorphism by first raising to the power $r(q - 1)$ then computing a discrete logarithm in a small subgroup.

We finally provide a useful lemma to sample group elements.

Lemma 10. *Let G, H, d be defined as in Lemma 2. Let $x_1, \ldots, x_s \in G$ which H-generate G. The following mapping from $G \times \mathbf{Z}_d^s$ to G is balanced.*

$$g : (r, a_1, \ldots, a_s) \longmapsto dr + a_1x_1 + \ldots + a_sx_s$$

2.3 Problem Reductions

We assume that S interpolates in a group homomorphism. We notice that the S-GHI problem can be solved with a single oracle call to a (d, S_1)-MSR oracle where S_1 denotes the set of all x coordinates for points in S.

Similarly, the S-GHID problem can be probabilistically solved with a (d, S_1)-MSR oracle by using Lemma 1 and Lemma 10: we generate a random $x' = ax + dr + a_1x_1 + \cdots + a_sx_s$, we send it to the MSR oracle who will answer $a'_1, \ldots, a'_s$, and we check whether $ay + (a_1 - a'_1)y_1 + \cdots + (a_s - a'_s)y_s = 0$.

Note that once we have witnesses to find the group invariants of G and H, it becomes easy to solve all problems. So GHI and GHID are in NP∩co-NP.

2.4 Problem Approximations

In this section we present our most important results. They are inspired from the theory of checkable proofs [2,3] and linear cryptanalysis.

Lemma 11. *Given two finite Abelian groups G and H, and a set of s points $S = \{(x_i, y_i) \mid i = 1, \ldots, s\}$, we assume that $x_1, \ldots, x_s$ H-generate G. We let d be the order of H and p be its smallest prime factor. We assume that there exists a function $f : G \longrightarrow H$ such that*

$$\rho := \Pr_{(r,a_1,\ldots,a_s) \in_U G \times \mathbf{Z}_d^s} [f(dr + a_1x_1 + \cdots + a_sx_s) = a_1y_1 + \cdots + a_sy_s] > \frac{1}{p}.$$

The set of points (x_i, y_i) interpolates in a group homomorphism. Furthermore, given a random $x \in_U G$, the value $y = f(x)$ matches the unique interpolation with probability ρ.

This improves Theorem 13 from [19] where we have $1/2$ instead of $1/p$.

Proof. Let $K \subseteq \mathbf{Z}_d^s$ be the set of all $(a_1, \ldots, a_s)$ such that $a_1x_1 + \cdots + a_sx_s \in dG$. We notice that the representation of any G element as a combination of $x_1, \ldots, x_s$ is uniquely defined modulo K. Following Lemma 1, we only have to prove that we have $a_1y_1 + \cdots + a_sy_s = 0$ for any $(a_1, \ldots, a_s) \in K$. This way, the value $g(x) = a_1y_1 + \cdots + a_sy_s$ is uniquely defined by $x = dr + a_1x_1 + \cdots + a_sx_s$ and g is a group homomorphism which corresponds to f with probability ρ.

Let us consider a random $(r, a_1, \ldots, a_s) \in_U G \times \mathbf{Z}_d^s$. ρ is the probability that $f(dr + a_1x_1 + \cdots + a_sx_s)$ equals $a_1y_1 + \cdots + a_sy_s$. This probability is also the average over all possible cosets of $\mathbf{Z}_d^s/K$ of the same probability when $(a_1, \ldots, a_s)$ is sampled in the coset only. Hence we deduce the existence of a coset $(a_1, \ldots, a_s) + K$ such that for $(r, b_1, \ldots, b_s) \in_U G \times K$ we have

$$\Pr[f(dr + (a_1 + b_1)x_1 + \cdots + (a_s + b_s)x_s) = (a_1 + b_1)y_1 + \cdots + (a_s + b_s)y_s] \geq \rho.$$

Note that $a_1x_1+\cdots+a_sx_s$ is now a constant x and that $dr+b_1x_1+\cdots+b_sx_s$ can be written dr' where r' is uniformly sampled in G and independent from $b_1,\ldots,b_s$. Hence, there exists r' such that

$$\Pr_{(b_1,\ldots,b_s)\in_U K}[f(dr'+x)=(a_1+b_1)y_1+\cdots+(a_s+b_s)y_s]\geq\rho.$$

So we have

$$\Pr_{(b_1,\ldots,b_s)\in_U K}[b_1y_1+\cdots+b_sy_s=\text{constant}]>\frac{1}{p}.$$

Since $(b_1,\ldots,b_s)\mapsto b_1y_1+\cdots+b_sy_s$ is a group homomorphism from K to a subgroup of H it must be a balanced function. Its kernel is either a subgroup of size at least p or the trivial subgroup $\{0\}$. Hence, the probability must actually be 1 and we have $b_1y_1+\cdots+b_sy_s=0$ for all $(b_1,\ldots,b_s)\in K$. □

The next result says that f can be used in order to solve the GHI problem.

Lemma 12. *Given two finite Abelian groups G and H, and a set of s points $S=\{(x_i,y_i)\mid i=1,\ldots,s\}$, we assume that $x_1,\ldots,x_s$ H-generate G. We assume that we are given the order d of H whose smallest prime factor is p and that we can sample elements in G with a uniform distribution. We assume that we have an oracle function $f:G\longrightarrow H$ such that*

$$\Pr_{(r,a_1,\ldots,a_s)\in_U G\times\mathbf{Z}_d^s}[f(dr+a_1x_1+\cdots+a_sx_s)=a_1y_1+\cdots+a_sy_s]=\frac{1}{p}+\theta$$

with $\theta>0$. Let $\varepsilon>0$ be arbitrarily small. There exists a group homomorphism which interpolates S and which is computable within $4\theta^{-2}\log(p/\varepsilon)$ oracle calls with an error probability less or equal to ε.

Note that this substantially improves Theorem 8 from [19] where we basically have $11/12$ instead of $1/p$. It was further conjectured in [19] that we could replace it by $1/2$. We made here a more precise result.

Proof (sketch). Due to Lemma 11, the homomorphism g exists and we have $\Pr_{x\in_U G}[f(x)=g(x)]=p^{-1}+\theta$. We use the same techniques which are used in linear cryptanalysis and consider the following algorithm.

Input: $x\in G$
repeat
  pick $r\in G,a_1,\ldots,a_s\in\mathbf{Z}_d$ at random
  $y=f(x+dr+a_1x_1+\cdots+a_sx_s)-a_1y_1-\cdots-a_sy_s$
  $c=0$
  for $i=1$ to n **do**
    pick $r\in G,a_1,\ldots,a_s,a\in\mathbf{Z}_d$ at random
    if $f(dr+a_1x_1+\cdots+a_sx_s+ax)=a_1y_1+\cdots+a_sy_s+ay$ (**T**) **then**
      $c=c+1$
    end if

10: **end for**
11: **until** $c > \tau n$
Output: y

We choose $n = 4\theta^{-2}(p^{-1} + \theta)\log(p/\varepsilon)$ and $\tau = p^{-1} + \frac{1}{2}\theta$ and we estimate the error probability of the acceptance test. We consider two types of error:

$$\varepsilon_1 = \Pr_{x \in_U G}[c \leq \tau n \mid y = g(x)] \qquad \varepsilon_2 = \Pr_{x \in_U G}[c > \tau n \mid y \neq g(x)]$$

We will now estimate these two values and show that they are negligible. If $y \neq g(x)$, then the test (**T**) works with probability $t_2 \leq 1/p$ due to Lemma 11. We also notice that if $y = g(x)$, the probability that the test works is $\frac{1}{p} + \theta$. Hence, using the central limit theorem we obtain

$$\varepsilon_1 \approx \Phi\left(\sqrt{n}\frac{\tau - p^{-1} - \theta}{\sqrt{(p^{-1}+\theta)(1-p^{-1}-\theta)}}\right) \qquad \varepsilon_2 \approx \Phi\left(-\sqrt{n}\frac{\tau - t_2}{\sqrt{t_2(1-t_2)}}\right),$$

when n is large enough and where Φ denotes the distribution function of the standard normal distribution. By looking at the logarithmic derivative of the function $f(t) = (\tau - t)/(\sqrt{t(1-t)})$ and noticing that this one is negative on the interval $[0, \tau]$ we deduce that

$$\varepsilon_2 \leq \Phi\left(-\sqrt{n}\frac{\tau - p^{-1}}{\sqrt{p^{-1}(1-p^{-1})}}\right).$$

Using $\tau = p^{-1} + \frac{1}{2}\theta$ provides

$$\varepsilon_2 \leq \Phi\left(-\sqrt{n}\frac{\theta}{2\sqrt{p^{-1}(1-p^{-1})}}\right) \approx \frac{1}{\sqrt{2\pi}}\left(e^{\frac{-n\theta^2}{4(p^{-1}(1-p^{-1}))}}\right),$$

where the last approximation holds when n is large enough (ε small). Now, we substitute the expression of n in the above inequality and we obtain

$$\varepsilon_2 \leq \frac{1}{\sqrt{2\pi}}\left(\frac{\varepsilon}{p}\right)^{\frac{p+p^2\theta}{p-1}}.$$

Since $\frac{p+p^2\theta}{p-1} \geq 1$ and $\frac{\varepsilon}{p} < 1$ when ε is small, we finally get $\varepsilon_2 \leq \varepsilon/(p\sqrt{2\pi}) \leq \rho\varepsilon/2$ where $\rho = p^{-1} + \theta$. In a similar way, we can show that $\varepsilon_1 \leq \varepsilon/2$. It remains to compute the complexity and the error probability of the algorithm. At first, we observe that the probability α that $c \leq \tau n$ in the algorithm is equal to $\rho\varepsilon_1 + (1-\rho)(1-\varepsilon_2)$. From the estimate of $\varepsilon_1, \varepsilon_2$, we see that $\alpha \approx 1-\rho$. Moreover, the number of iterations is equal to $\sum_{i=1}^{\infty} i\alpha^{i-1}(1-\alpha) = 1/(1-\alpha) \approx 1/\rho$. Hence, the complexity is $n/\rho = 4(\log(1/\varepsilon) + \log(p))/(\rho - \frac{1}{p})^2$. The probability of error is given by $\sum_{i=1}^{\infty} \alpha^{i-1}(1-\rho)\varepsilon_2 \approx (1-\rho)/\rho\varepsilon_2 \leq \varepsilon_2/\rho \leq \varepsilon/2$. □

3 Interactive Proof Protocol

3.1 Proof for the GHID Problem

Let G, H, and $S = \{(g_1, e_1), \ldots, (g_s, e_s)\}$ be parameters of a GHI problem, and let d be the order of H. We assume that we have a prover who wants to convince a verifier that he knows an interpolating group homomorphism $f : G \longrightarrow H$ for S. Let ℓ be an integer. He performs the following interaction with a verifier.

> **GHIproof**$_\ell(S)$
> **Parameters:** G, H, d
> **Input:** ℓ, $S = \{(g_1, e_1), \ldots, (g_s, e_s)\} \subseteq G \times H$
> 1: The verifier picks $r_i \in G$ and $a_{i,j} \in \mathbf{Z}_d$ at random for $i = 1, \ldots, \ell$ and $j = 1, \ldots, s$. He computes $u_i = dr_i + a_{i,1}g_1 + \cdots + a_{i,s}g_s$ and $w_i = a_{i,1}e_1 + \cdots + a_{i,s}e_s$ for $i = 1, \ldots, \ell$. He sends $u_1, \ldots, u_\ell$ to the prover.
> 2: The prover computes $v_i = f(u_i)$ for $i = 1, \ldots, \ell$. He sends a commitment to $v_1, \ldots, v_\ell$ to the verifier.
> 3: The verifier sends all r_i's and $a_{i,j}$'s to the prover.
> 4: The prover checks that the u_i's computations are correct. He then opens his commitment.
> 5: The verifier checks that $v_i = w_i$ for $i = 1, \ldots, \ell$.

From a practical point of view, the verifier can generate the r_i's and $a_{i,j}$'s in a pseudorandom way from a seed and simply disclose the seed in the third step of the protocol. Further note that if d^s is large enough, then the verifier can send $h(w_1, \ldots, w_s) \oplus$ seed (where h is a hash function) in his first message so that the complete protocol can run in 2 moves instead of 4. In the second move, the prover simply sends seed.

Note that we need a commitment scheme here, e.g. the trapdoor commitment scheme proposed by Bresson et al. [6]. Note that using trapdoor commitment with the verifier's public key strengthens our protocols by providing the non-transferability property [17].

Theorem 13. *Assuming that $g_1, \ldots, g_s$ H-generate an Abelian group G, let d be an integer and $e_1, \ldots, e_s \in H$, where H is an Abelian group of order d. Let p be the smallest prime factor of d. We consider the GHIproof$_\ell(S)$ protocol with $S = \{(g_1, e_1), \ldots, (g_s, e_s)\} \subseteq G \times H$.*

i. Completeness: assuming that the prover and the verifier are honest, the protocol always succeeds.
ii. Zero-knowledge: assuming that the commitment scheme is perfectly hiding, the above protocol is perfectly black-box zero-knowledge against any verifier.
iii. Proof of membership: assuming that the protocol succeeds with probability greater than $p^{-\ell}$ with a honest verifier, then S interpolates in a group homomorphism.

iv. *Proof of knowledge: for any $\theta > 0$, assuming that the protocol succeeds with probability greater than $(p^{-1} + \theta)^\ell$ with a honest verifier and that the commitment scheme is extractable, for any $\varepsilon > 0$ there exists an extractor with a time complexity factor $\mathcal{O}(\log(1/\varepsilon))$ which can compute an interpolating group homomorphism from the prover with probability at least $1 - \varepsilon$.*

Proof (sketch). Property i is quite clear. Property ii is proven by constructing a simulator for the transcript of the protocol without the secret of the prover. Property iii directly follows from Lemma 11. For Property iv, we use Lemma 11 and Lemma 12. □

3.2 Proof for the co-GHID Problem

Let G, H, and $S = \{(g_1, e_1), \ldots, (g_s, e_s)\} \subseteq G \times H$ be parameters of a GHI problem, and let d be the order of H. Let $T = \{(x_1, z_1), \ldots, (x_t, z_t)\} \subseteq G \times H$ be a set of t inputs of the GHID problem. We assume that we have a prover who wants to convince a verifier that for at least one k the answer to the GHID problem with (x_k, z_k) is negative. Let ℓ be an integer. He performs the following interaction with a verifier.

coGHIproof$_\ell(S, T)$
Parameters: G, H, d
Input: ℓ, $S = \{(g_1, e_1), \ldots, (g_s, e_s)\}$, $T = \{(x_1, z_1), \ldots, (x_t, z_t)\}$

1: The verifier picks $r_{i,k} \in G$, $a_{i,j,k} \in \mathbf{Z}_d$, and $\lambda_i \in \mathbf{Z}_p^*$ for $i = 1, \ldots, \ell$, $j = 1, \ldots, s$, $k = 1, \ldots, t$, where p is the smallest prime dividing d. He computes $u_{i,k} := dr_{i,k} + \sum_{j=1}^{s} a_{i,j,k} g_j + \lambda_i x_k$ and $w_{i,k} := \sum_{j=1}^{s} a_{i,j,k} e_j + \lambda_i z_k$. Set $u := (u_{1,1}, \ldots, u_{\ell,t})$ and $w := (w_{1,1}, \ldots, w_{\ell,t})$. He sends u and w to the prover.
2: The prover computes $v_{i,k} := f(u_{i,k})$ for $i = 1, \ldots, \ell$, $k = 1, \ldots, t$. Since $w_{i,k} - v_{i,k} = \lambda_i(z_k - y_k)$, he should be able to find every λ_i if the verifier is honest since $w_{i,k} \neq v_{i,k}$ for all i and at least one k. Otherwise, he sets λ_i to a random value. He then sends a commitment to $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ to the verifier.
3: The verifier sends all $r_{i,k}$'s and $a_{i,j,k}$'s to the prover.
4: The prover checks that u and w were correctly computed. He then opens the commitment to λ.
5: The verifier checks that the prover could find the right λ.

This protocol is inspired from denial protocol of Gennaro et al. [15]. We can also transform it into a 2-move protocol.

We notice that λ_i was chosen such that it can be uniquely retrieved for every nonzero values of $\mathbf{Z}_d$ that can be taken by the elements $z_k - y_k$'s. Namely, this is done by the following result.

Lemma 14. *Let H be an Abelian group of order d, and $a, b \in H$ such that $b \neq 0$. Let λ be in $\{1, \ldots, p-1\}$, where p is the smallest prime dividing d. Then, if the equation $a = \lambda b$ has a solution in λ, then this one is unique.*

3.3 Proof for the MGGD Problem

Inspired by [19], we propose here a proof that $S_1 = \{g_1, \ldots, g_s\}$ H-generate G. However, the signer needs expert knowledge about G since he has to be able to solve the (d, S_1)-MSR and d-Root problems. Let ℓ be an integer. He performs the following protocol.

MGGDproof$_\ell(S_1)$
Parameters: G, H, d
Input: ℓ, $S_1 = \{g_1, \ldots, g_s\} \subseteq G$

1: **for** $i = 1$ to ℓ **do**
2: The prover picks a $\delta_1 \in G$ at random and sends a commitment to δ_1 to the verifier.
3: The verifier picks a $\delta_2 \in G$ at random and sends δ_2 to the prover.
4: The prover solves (d, S_1)-MSR on $\delta_1 + \delta_2$ and d-Root and finds $r \in G$, $a_1, \ldots, a_s \in \mathbf{Z}_d$ such that $\delta_1 + \delta_2 = dr + \sum_{j=1}^{s} a_j g_j$. He sends $r, a_1, \ldots, a_s$ to the verifier and opens the commitment to δ_1.
5: The verifier checks that $\delta_1 + \delta_2 = dr + \sum_{j=1}^{s} a_j g_j$ really holds.
6: **end for**

We can prove as in Lemma 11 that if a honest verifier is convinced with probability greater than $p^{-\ell}$, then S_1 solves the d-MGGD problem.

Note that this can be transformed into a non-interactive proof following standard techniques [14]. An efficient way consists of generating pseudorandom $\delta_1, \ldots, \delta_\ell$ from the same seed then solving the (d, S_1)-MSR and d-Root problems on those elements.

4 Undeniable Signature

4.1 Description

We now describe our undeniable signature scheme.

Domain parameters. We let integers Lkey, Lsig, Icon, Iden be security parameters as well as "group types" for Xgroup and Ygroup. (The group types should define what groups and which sizes to use in order to achieve security.) An optional parameter Ival is used in Setup Variants 3 and 4 below.

Primitives. We use two deterministic random generators Gen_1 and Gen_2 which produce elements of Xgroup and a commitment scheme.

Setup Variant 1. (signer without expert group knowledge)
The signer selects Abelian groups Xgroup and Ygroup of given types together with a group homomorphism Hom : Xgroup $\longrightarrow$ Ygroup. He computes the order d of Ygroup. He then picks a random string seedK and computes the Lkey first values $(\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}})$ from $\mathrm{Gen}_1(\mathrm{seedK})$ and $\mathrm{Ykey}_j := \mathrm{Hom}(\mathrm{Xkey}_j)$, $j = 1, \ldots, \mathrm{Lkey}$.
The main problem of Setup is that the choice for $(\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}})$ must Ygroup-generate Xgroup in order to ensure non-repudiation of signatures. In Variant 1, Lkey must be large enough so that it is impossible to maliciously select a key which does not guaranty this condition.

Setup Variant 2. (signer with a Registration Authority (RA))
We use here a RA whose role consists of making sure that a key was randomly selected. (Note that, the RA does not check if the key is valid.)

1. The signer selects Abelian groups Xgroup and Ygroup of given type together with a group homomorphism Hom : Xgroup $\longrightarrow$ Ygroup. He computes the order d of Ygroup. He submits his identity Id together with Xgroup, Ygroup and d to RA.
2. RA first checks the identity of the signer and that he did not submit too many registration attempts. He then picks a random string seedK that is sent to the signer together with a signature C for

$$(\mathrm{Id}, \mathrm{Xgroup}, \mathrm{Ygroup}, d, \mathrm{seedK}).$$

3. The signer computes the Lkey first values $(\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}})$ from $\mathrm{Gen}_1(\mathrm{seedK})$ and $\mathrm{Ykey}_j := \mathrm{Hom}(\mathrm{Xkey}_j)$, $j = 1, \ldots, \mathrm{Lkey}$.

Here the RA basically selects the random key so Lkey can be reduced.

Setup Variant 3. (signer with an expert group knowledge)
In this variant we assume that the signer can solve the MSR and Root problems in Xgroup. It works exactly like in the Setup Variant 1, but the signer can further run a $\mathrm{MGGDproof}_{\mathrm{Ival}}$ in order to validate the public key so that Lkey can be further reduced to the smallest possible one.

Setup Variant 4. (signer with an expert group knowledge, non-interactive)
This variant is the same as Variant 3 except that MGGDproof is transformed into a non-interactive proof.

Public Key. $K_P = (\mathrm{Xgroup}, \mathrm{Ygroup}, d, \mathrm{seedK}, (\mathrm{Ykey}_1, \ldots, \mathrm{Ykey}_{\mathrm{Lkey}}))$ with an optional (Id, C) for Variant 2, an optional Ival for Variants 3,4, and an optional non-interactive proof for Variant 4. We say that K_P is valid if $\{\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}}\}$ Ygroup-generate Xgroup.

Secret Key. $K_S = \mathrm{Hom}$.

Signature generation. The message M is used to generate $\mathrm{Xsig}_1, \ldots, \mathrm{Xsig}_{\mathrm{Lsig}}$ from $\mathrm{Gen}_2(M)$. The signer computes $\mathrm{Ysig}_k = \mathrm{Hom}(\mathrm{Xsig}_k)$ for $k = 1, \ldots, \mathrm{Lsig}$. The signature is $(\mathrm{Ysig}_1, \ldots, \mathrm{Ysig}_{\mathrm{Lsig}})$. It consists of $\mathrm{Lsig}.\log_2 d$ bits.

Confirmation Protocol. Compute $\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}}$ from the public key, $\mathrm{Xsig}_1, \ldots, \mathrm{Xsig}_{\mathrm{Lsig}}$ from the message, run $\mathrm{GHIproof}_{\mathrm{Icon}}$ on the set

$$S = \{(\mathrm{Xkey}_j, \mathrm{Ykey}_j) | j = 1, \ldots, \mathrm{Lkey}\} \cup \{(\mathrm{Xsig}_k, \mathrm{Ysig}_k) | k = 1, \ldots, \mathrm{Lsig}\}.$$

Denial Protocol. Compute $\mathrm{Xkey}_1, \ldots, \mathrm{Xkey}_{\mathrm{Lkey}}$ from the public key as well as $\mathrm{Xsig}_1, \ldots, \mathrm{Xsig}_{\mathrm{Lsig}}$ from the message, run $\mathrm{coGHIproof}_{\mathrm{Iden}}$ on the sets

$$S = \{(\mathrm{Xkey}_j, \mathrm{Ykey}_j) | j = 1, \ldots, \mathrm{Lkey}\},\ T = \{(\mathrm{Xsig}_k, \mathrm{Zsig}_k) | k = 1, \ldots, \mathrm{Lsig}\}$$

where $(\mathrm{Zsig}_1, \ldots, \mathrm{Zsig}_{\mathrm{Lsig}})$ is the alleged non-signature.

The undeniable signature scheme of Gennaro et al. [15] which is based on RSA corresponds to a special case of our scheme, namely with Xgroup = Ygroup = $\mathbf{Z}_n^*$, Lkey = Lsig = 1 and the classical RSA signing function as homomorphism

Hom. Another example with Lkey = Lsig = 1 is the undeniable signature of Chaum [7]. He considered Xgroup = Ygroup = $\mathbf{Z}_p^*$ for a prime p and the homomorphism consisting in raising an element to the power of the private key. In both examples the signature is quite large. The MOVA scheme [19] is another example with Xgroup = $\mathbf{Z}_n^*$, Hom is a character of order $d \in \{2,3,4\}$, and Ygroup is the subgroup of $\mathbf{C}^*$ spanned by $e^{\frac{2i\pi}{d}}$.

4.2 Security Analysis

Theorem 15 (Setup Variants 1,2). *We consider the above undeniable signature. Given a prime q, we let A_q be the subgroup of* Xgroup *of all terms whose orders are powers of q. Given q there is a unique k_q and $a_{q,1} \leq \ldots \leq a_{q,k_q}$ sequence such that A_q is isomorphic to $\mathbf{Z}_{q^{a_{q,1}}} \oplus \ldots \oplus \mathbf{Z}_{q^{a_{q,k_q}}}$. The probability P_{gen} that $\{\text{Xkey}_1, \ldots, \text{Xkey}_{\text{Lkey}}\}$ Ygroup-generate Xgroup satisfies*

$$P_{\text{gen}} \geq \prod_{q \in \mathbf{P}_d} \left(1 - \frac{k_q}{q^{\text{Lkey}}}\right),$$

where $\mathbf{P}_d$ is the set of all prime factors of $\gcd(\#\text{Xgroup}, d)$.

As an application, if d is prime and if Xgroup is a product of k cyclic groups, we have $P_{\text{gen}} \geq 1 - k.d^{-\text{Lkey}}$.

Theorem 16. *We consider the above undeniable signature scheme. Assuming that the public key is valid, we have the following security results.*

i. If the signer and the verifier are honest, the two protocols complete: a valid signature will always be accepted by the confirmation protocol, and an invalid signature will always be rejected by the denial protocol.
ii. Let $S = \{(\text{Xkey}_1, \text{Ykey}_1), \ldots, (\text{Xkey}_{\text{Lkey}}, \text{Ykey}_{\text{Lkey}})\}$. The scheme resists against existential forgery attacks provided that Gen_2 *is a random oracle and the S-GHI problem is intractable.*
iii. The confirmation (resp. denial) protocol is sound: if the signer is able to pass the protocol with probability $q > p^{-\text{Icon}}$ (resp. $q > p^{-\text{Iden}}$), then the alleged signature is valid (resp. invalid).
iv. The confirmation protocol is private when the commitment scheme is extractable: for any $\theta, \varepsilon > 0$, from a prover which is able to convince a honest verifier that a given signature is valid with probability $q > (p^{-1} + \theta)^{\text{Icon}}$, we can extract within a complexity factor of $\Omega(\theta^{-2} \log(p/\varepsilon))$ a group homomorphism which solves the GHI problem with success probability $1 - \varepsilon$.
v. The signatures are invisible: for any $\theta, \varepsilon > 0$, from a distinguisher of a valid signature from a random one with advantage $\theta > 0$, we can extract within a complexity factor of $\Omega(\theta^{-2} \log(1/\varepsilon))$ a GHID problem solver with success probability $1 - \varepsilon$.
vi. The confirmation (resp. denial) protocol is perfectly black-box zero-knowledge when the commitment scheme is perfectly hiding: we can build a simulator for the protocol without the secret key for any verifier.

In short, if we take Xgroup $= \mathbf{Z}_n^*$ where n is a product of two prime numbers, and Lsig $=$ Icon $=$ Iden $= s_{\text{online}}/\log_2 p$, we cannot contradict the confirmation or denial protocols but with a probability at most $2^{-s_{\text{online}}}$, and signatures are invisible provided that generators are random oracles and that the interpolation problem is hard. For Variant 2, we can take Lkey $= s_{\text{online}}/\log_2 p$ and this generates invalid keys with probability less than $2^{1-s_{\text{online}}}$. For Variant 1, we can take Lkey $= s_{\text{offline}}/\log_2 p$ so that the signer cannot create invalid keys within a complexity less than $2^{s_{\text{online}}}$. For Variants 3,4, Lkey can be as low as possible. We can take Ival $= s_{\text{online}}/\log_2 p$ for Variant 3 (so that invalid keys are accepted with probability less than $2^{-s_{\text{online}}}$), and Ival $= s_{\text{offline}}/\log_2 p$ for Variant 4 so that the signer cannot create invalid keys within a complexity less than $2^{s_{\text{online}}}$. We suggest $s_{\text{offline}} = 80$ and $s_{\text{online}} = 20$.

5 Example and Further Discussions

5.1 Setting Proposal

We consider Example 9 with a small prime d e.g. $d = 2^{20} + 7$. We take Xgroup $= \mathbf{Z}_n^*$, Ygroup $= \mathbf{Z}_d$, Lkey $=$ Lsig $=$ Icon $=$ Iden $= 1$ and we consider Variant 3 and 4 of the Setup protocol. If Xkey $\in$ Xgroup is not a dth power residue then it Ygroup-generates Xgroup. For any Ykey $\in \mathbf{Z}_d$ there is a unique group homomorphism Hom such that Hom(Xkey) $=$ Ykey. With this example we can sign with a single element of $\mathbf{Z}_d$ and a public key $(n, d, \text{seedK}, \text{Ykey})$.

Note that the group homomorphism computation requires raising to the power r in $\mathbf{Z}_p^*$ and computing the discrete logarithm in a cyclic group of about 2^{20} elements. This can be precomputed in a table of 2.5 MB as detailed below.

We first precompute a (large) table of all (Xsig_i, i) with $\text{Xsig}_i = \text{Xkey}^{ir} \pmod p$ for $i = 0, 1, \ldots, d-1$. Note that i can be encoded into 20 bits. Next we insert all (Xsig_i, i) pairs in a hash table of 2^{20} entries keyed by Xsig_i: put i at position $h(\text{Xsig}_i)$ unless there is a collision. Resolving collisions can be done by standard techniques, for instance see [10] Chapter 12, but note that resolving collisions is not necessary: if Xsig_i is not in the table, we can look for the smallest j such that Xsig_{i+j} is in the table.

Time/memory tradeoffs can also be considered. Remark also that such a tradeoff should not require more than the complexity of the Pollard's rho algorithm for the computation of the discrete logarithm in our example, i.e. approximately 3000 multiplications.

Depending on the application, the signature size of 20 bits may be considered as too small. Of course, we can easily enlarge it e.g. to 48 bits. Our point is that signature size versus security is fully scalable here.

The signature generation requires 1 homomorphism i.e. about one exponentiation in $\mathbf{Z}_p^*$. (Note that this is twice as fast as a 1024-bit RSA signature computation with Chinese remainders.) The complexity of the confirmation protocol is about 35 multiplications in $\mathbf{Z}_n^*$ for the verifier (which can be compared to 17 multiplications in $\mathbf{Z}_n^*$ for RSA if we take $e = 2^{16} + 1$) and 1 homomorphism for the prover. The denial protocol requires almost the same complexity.

Complexities of this setting with all setup variants as well as those of the MOVA scheme with $d = 2$ and a 20-bit signature length are detailed in Table 1. The main advantage of using the above setting instead of MOVA is that the former strongly decreases the number of multiplications in $\mathbf{Z}_n^*$ for the confirmation.

Setup	d	Lsig, Icon, Iden	Lkey	Ival	Signature cost	Confirmation cost
1	2	20	80		20 Leg. symb.	20 Leg. symb., 730 mult.
2	2	20	20		20 Leg. symb.	20 Leg. symb., 280 mult.
3	2	20	2	20	20 Leg. symb.	20 Leg. symb., 145 mult.
4	2	20	2	80	20 Leg. symb.	20 Leg. symb., 145 mult.
1	$2^{20}+7$	1	4		1 Hom	1 Hom, 65 mult.
2	$2^{20}+7$	1	1		1 Hom	1 Hom, 35 mult.
3	$2^{20}+7$	1	1	1	1 Hom	1 Hom, 35 mult.
4	$2^{20}+7$	1	1	4	1 Hom	1 Hom, 35 mult.

Table 1. Implementation Examples.

5.2 On the MOVA Scheme

We point out here that our scheme generalizes the MOVA scheme [19] and improves the efficiency of the denial protocol of MOVA. An additional contribution to MOVA is also the improvement of some bounds related to the probability of a function approximating Hom from which we can compute Hom in a polynomial time. Our new bound with $1/p$ allows to formally prove the conjectured security level of MOVA.

5.3 Batch Verification and Selective Convertibility

We point out that our scheme allows a batch verification of signatures. Indeed, the confirmation protocol can be easily adapted in order to confirm several signatures at the same time by putting all $(\mathrm{Xsig}_k, \mathrm{Ysig}_k)$ in a single set S.

Note that the signer with expert group knowledge can selectively convert an undeniable signature into a classical one by solving the MSR and Root problems on all Xsig_k. The conversion consists of revealing the solution to those problems.

6 Conclusion

We have exposed an undeniable signature based on a generic group homomorphism interpolation and we have also analyzed the security in the random oracle model. The principal advantage is the size of the signature that can be chosen arbitrarily short depending on the required security level. Confirmation and denial can be run in a 2-move protocol. We can perform batch verification and have selective convertibility. From this general setting we have also proposed a practical example with 3-byte signatures and a complexity cost which is similar to RSA. We hope that this example will be completed by some various additional settings since group homomorphisms are common objects in cryptography.

As future work, we also aim at extending our techniques to other cryptographic algorithms such as the designated confirmer signatures [8].

Acknowledgments. We wish to thank Anna Lysyanskaya and Wenbo Mao for helpful discussions and comments.

References

1. R. Anderson, S. Vaudenay, B. Preneel, K. Nyberg, *The Newton Channel*, Proc. First International Workshop on Information Hiding, Cambridge, UK, LNCS **1174**, pp. 151–156, Springer, 1996.
2. S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof Verification and Hardness of Approximation Problems*, Proc. 33rd IEEE Symp. on Foundations of Computer Science, pp. 14–23, 1992.
3. L. Babai, L. Fortnow, L. Levin and M. Szegedy, *Checking Computations in Polylogarithmic Time*, Proc. 23rd ACM Symp. on Theory of Computing, pp. 21–31, 1991.
4. I. Biehl, S. Paulus and T. Takagi, *Efficient Undeniable Signature Schemes based on Ideal Arithmetic in Quadratic Orders*, Conference on The Mathematics of Public-Key Cryptography, Toronto, 1999.
5. J. Boyar, D. Chaum, I. Damgård and T. Pedersen, *Convertible Undeniable Signatures*, Advances in Cryptology - Crypto '90, LNCS **537**, pp. 189–205, Springer, 1990.
6. E. Bresson, D. Catalano and D. Pointcheval, *A Simple Public-Key Cryptosytem with a Double Trapdoor Decryption Mechanism and Its applications*, Advances in Cryptology - Asiacrypt '03, LNCS **2894**, pp. 37–54, Springer, 2003.
7. D. Chaum, *Zero-Knowledge Undeniable Signatures*, Advances in Cryptology - Eurocrypt '90, LNCS **473**, pp. 458–464, Springer, 1990.
8. D. Chaum, *Designated Confirmer Signatures*, Advances in Cryptology - Eurocrypt '94, LNCS **950**, pp. 86–91, Springer, 1994.
9. D. Chaum and H. van Antwerpen, *Undeniable Signatures*, Advances in Cryptology - Crypto '89, LNCS **435**, pp. 212–217, Springer, 1989.
10. T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw Hill, 1990.
11. I. Dåmgard and T. Pedersen, *New Convertible Undeniable Signatures Schemes*, Advances in Cryptology - Eurocrypt '96, LNCS **1070**, pp. 372–386, Springer, 1996.
12. Y. Desmedt and M. Yung, *Weaknesses of Undeniable Signature Schemes*, Advances in Cryptology - Crypto '91, LNCS **576**, pp. 205–220, Springer, 1991.
13. W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, vol. IT-22, pp. 644–654, 1976.
14. A. Fiat, A. Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, Advances in Cryptology - Crypto '86, LNCS **263**, pp. 186–194, Springer, 1987.
15. R. Gennaro, T. Rabin and H. Krawczyk, *RSA-Based Undeniable Signatures*, Journal of Cryptology, **13**, pp. 397–416, Springer, 2000.

16. S. Goldwasser and S. Micali, *Probabilistic Encryption*, Journal of Computer and System Sciences, 28, pp. 270–299, 1984.
17. M. Jakobsson, K. Sako and R. Impagliazzo, *Designated Verifier Proofs and Their Applications*, Advances in Cryptology - Eurocrypt '96, LNCS **1070**, pp. 143-154, 1996.
18. B. Libert and J.-J. Quisquater, *Identity Based Undeniable Signatures*, Proc. RSA Crypto Track '04, LNCS **2964**, pp. 112–125, Springer, 2004.
19. J. Monnerat and S. Vaudenay, *Undeniable Signatures Based on Characters: How to Sign with One Bit*, PKC '04, LNCS **2947**, pp. 69–85, Springer, 2004.
20. P. Paillier, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, Advances in Cryptology - Eurocrypt '99, LNCS **1592**, pp. 223–238, Springer, 1999.
21. R. L. Rivest, A. Shamir and L. M. Adleman, *A Method for Obtaining Digital Signatures and Public-key Cryptosystem*, Communications of the ACM, vol. 21, pp. 120–126, 1978.

A Technical proofs

Proof of Lemma 2. **1** $\Leftrightarrow$ **2** $\Leftrightarrow$ **3.** Straightforward.
3 $\Rightarrow$ **4.** Assume that there exists a common prime factor p of $\#(G/G')$ and d. Then, from the structure of Abelian groups G/G' and H we know that each of these two groups possesses one cyclic subgroup U and V respectively of order p. So, we define a non trivial homomorphism that is the composition of the isomorphism between the two cyclic subgroups and with the reduction modulo U. This contradicts 3.
4 $\Rightarrow$ **5.** If $x \in G$, then d must be invertible modulo the order k of x mod G' by 4. Let m such that $m \cdot d \equiv 1 \pmod{k}$. We have $m \cdot d \cdot x \equiv x \pmod{G'}$. Hence, $x - d(m \cdot x) \in G'$ and therefore $x \in G' + dG$.
5 $\Rightarrow$ **2.** If $\varphi \in \mathrm{Hom}(G, H)$ is such that $\varphi|_{G'} = 0$ and $x \in G$, we can write $x = a_1x_1 + \cdots + a_sx_s + dr$. Thus, $\varphi(x) = d\varphi(r) = 0$. This holds for all $x \in G$, i.e., $\varphi = 0$. □

Proof of Lemma 10. Let n be the order of G. Let $h : G \times \mathbf{Z}_{nd}^s \to G$ be a function defined by $h(r, a_1, \ldots, a_s) = dr + a_1x_1 + \ldots + a_sx_s$. Obviously, h is an homomorphism. It is onto G due to the property 5 of Lemma 2. Hence, it is balanced onto G. Let $\varphi : G \times \mathbf{Z}_{nd}^s \to G \times \mathbf{Z}_d^s$ be a function defined by $\varphi(r, a_1, \ldots, a_s) \to (r + q_1x_1 + \cdots + q_sx_s, a_1 \bmod d, \ldots, a_s \bmod d)$, where $a_i - (a_i \bmod d) = dq_i$. We have $g \circ \varphi = h$. Obviously, φ is balanced onto $G \times \mathbf{Z}_d^s$ since $\varphi^{-1}(r, a_1, \ldots, a_s) = \{(r - q_1x_1 - \ldots - q_sx_s, a_1 + dq_1, \ldots, a_s + dq_s) \,|\, (q_1, \ldots, q_s) \in \mathbf{Z}_n^s\}$. If $\#g^{-1}(x) = m$, we have $mn^s = \#\varphi^{-1}\left(g^{-1}(x)\right) = \#h^{-1}(x) = (dn)^s$. Hence, $m = d^s$ does not depend on x, so g is balanced. □

Proof of Theorem 15 (sketch). The decomposition of Xgroup comes from classical results on the structure of Abelian groups. We observe that we can handle each A_q independently because we can see that two elements generating two different A_q's generate the direct sum of these two groups, since the two respective group orders are coprime. We consider $B_q := A_q/dA_q$ and study the probability that elements generate this group. If $\gcd(d,q)=1$, then B_q is trivial. So, we focus only on the q's that divide d and denote e_q the largest integer such that $q^{e_q}|d$. We can also deduce that the structure of B_q satisfies

$$B_q \simeq \mathbf{Z}_{q^{a_{q,1}}} \oplus \ldots \oplus \mathbf{Z}_{q^{a_{q,r}}} \oplus \mathbf{Z}_{q^{e_q}} \oplus \ldots \mathbf{Z}_{q^{e_q}},$$

where r is the largest integer such that $a_{q,r} < e_q$. The probability that s elements does not generate B_q can be approximated by the probability that these elements stay in one of the largest non trivial subgroups of B_q, i.e. those of order $\#B_q/q$. The number of such subgroups is equal to k_q. Thus, this probability is greater or equal than $1-\frac{k_q}{q^s}$. Since these events are independent for the different B_q's, the final probability is obtained by the multiplication of these probabilities. □

Proof of Theorem 16 (sketch). i. The assertion i is straightforward.
ii. First, we show that an attacker $\mathcal{A}$ having access to a signing oracle can be simulated by an attacker without this access. Indeed, when $\mathcal{A}$ calls the signing oracle on a message M, the signing oracle will first produce a sequence of Lsig values $\mathrm{Xsig}_1,\ldots,\mathrm{Xsig}_{\mathrm{Lsig}} \in \mathrm{Xgroup}$ and then computes $\mathrm{Ysig}_i := \mathrm{Hom}(\mathrm{Xsig}_i)$ for $i=1,\ldots,\mathrm{Lsig}$. From the point of view of $\mathcal{A}$, this is completely equivalent to dispose of a random source generating pairs of the form $(x,\mathrm{Hom}(x))$ since Gen_2 is modelized as a random oracle. Assuming that S_1 Ygroup-generate Xgroup, we see that this source can be simulated by picking some random $r \in \mathrm{Xgroup}$, a_i's $\in \mathbf{Z}_d$, computing $x := dr + a_1\mathrm{Xkey}_1 + \cdots + a_{\mathrm{Lkey}}\mathrm{Xkey}_{\mathrm{Lkey}}$ and $\mathrm{Hom}(x) = a_1\mathrm{Ykey}_1 + \cdots + a_{\mathrm{Lkey}}\mathrm{Ykey}_{\mathrm{Lkey}}$ using Lemma 10. We denote now x, the challenged element of the GHI problem. We use our attacker $\mathcal{A}$ in order to compute the $\mathrm{Hom}(\mathrm{Xsig}_i)$'s as follows. We simulate Gen_2 by computing $u := dr + x + \sum_{j=1}^{\mathrm{Lkey}} a_j\mathrm{Xkey}_j$ for some random $r \in \mathrm{Xgroup}$, $a_j \in \mathbf{Z}_d$. This is indistinguishable from some uniformly picked element in Xgroup. By standard proofs we show that forged signatures are necessarily one of the Gen_2 queries, so we can deduce $\mathrm{Hom}(x)$ from $\mathrm{Hom}(u)$.
iii. For the confirmation, this directly comes from Theorem 13 property iii. For the denial, a cheating prover willing deny a valid signature has to find the value of λ_i at each round of the protocol. Since $\mathrm{Hom}(u_{i,k}) = w_{i,k}$, the prover does not learn additional information with $w_{i,k}$ and has to find λ_i from $u_{i,k}$ uniquely. He cannot find the λ_i since another distribution of the values $u_{i,k}$ with another λ_i is indistinguishable from the first one. Assuming that the commitment scheme is perfectly binding the cheating prover cannot do better than answering a random λ_i.
iv. This directly comes from Theorem 13 property iv.
v. This works like in Lemma 12. We count how many times (x',y') is accepted after having picked $x' = x + dr + a_1\mathrm{Xkey}_1 + \cdots + a_{\mathrm{Lkey}}\mathrm{Xkey}_{\mathrm{Lkey}}$ and $y' = y + a_1\mathrm{Ykey}_1 + \cdots + a_{\mathrm{Lkey}}\mathrm{Ykey}_{\mathrm{Lkey}}$. We use $n = \theta^{-2}\log(1/\varepsilon)$ iterations.
vi. For the confirmation, this comes from property ii in Theorem 13. For the denial, this is done as in [15]. □