Undeniable Signatures Based on Characters: How to Sign with One Bit

Jean Monnerat * and Serge Vaudenay

Swiss Federal Institute of Technology (EPFL) - LASEC
`http://lasecwww.epfl.ch`

Abstract. We present a new undeniable signature scheme which is based on the computation of characters. Our signature scheme offers the advantage of having an arbitrarily short signature. Its asymptotic complexity is attractive: the asymptotic complexity of all algorithms (even the key setup) are quadratic in the size of the modulus n in bits when the other parameters are fixed. The practical complexity can be quite low depending on parameter and variant choices. We present also a proof of security of our scheme containing the standard security requirements of an undeniable signature.

Key words: Undeniable Signatures, Residue Characters.

1 Introduction

The concept of undeniable signature has been first introduced in 1989 by Chaum and van Antwerpen [6]. This kind of signature is similar to a classical digital signature except that one has to interact with the signer in order to be convinced of the validity of this one. This property offers the advantage of avoiding that any entity can verify the validity of a signature. In fact, limiting this universal verifiability (as it is in the case of a classical digital signature) is desirable in certain circumstances e.g. for privacy reasons. Here, the signer can control how the verification spreads in a community.

To be complete, an undeniable signature should be composed of three main components that are the signature generation, the confirmation protocol and the denial protocol. The role of the confirmation protocol is to allow the signer to prove the validity of a given signature. Conversely, the denial protocol allows a signer (prover) to prove the invalidity of a given signature. It is important to keep in mind that a failure in the confirmation protocol is not a proof of the invalidity of a signature but could be only due to a lack of cooperation from the prover. A similar argument holds also for the denial protocol. So, the confirmation resp. denial protocol is only used to prove the validity resp. invalidity of a signature.

Since their introduction, undeniable signatures received a certain attention and several papers related to them have been published. We give here a list

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

of some of them, [3–5, 8–10, 15]. It turns out that almost all of the undeniable signature schemes are based on the discrete logarithm. In [10], Gennaro et al. presented an undeniable signature based on RSA. In this paper, we propose a new undeniable signature that is based on another type of problems, namely the ability of computing a character on $\mathbb{Z}_n^*$. This corresponds actually to a generalization of the quadratic residuosity problem. In the present work, we focus our study on the characters of order $2, 3$ and 4. Note that the characters of order 3 have already been used in some public-key cryptosystems, e.g. [17] as well as more general characters, e.g. [18].

In section 2, we survey the mathematical theory of the characters on $\mathbb{Z}_n^*$. Section 3 is dedicated to the study of some problems related to the security of our scheme, in particular for cases of order $d = 2, 3, 4$. The new scheme is presented in the section 4. Section 5 is devoted to the security of our scheme. We provide some proofs of some security properties such as the resistance against existential forgery of our scheme or the soundness of the confirmation and denial protocol. Section 6 concludes the article.

2 Characters on $\mathbb{Z}_n^*$

In this section, we introduce the notion of multiplicative characters. The order 2, 3 and 4 cases will be exposed in the following subsections.

Definition 1. *Let n be an integer. A character χ on $\mathbb{Z}_n^*$ is a map from $\mathbb{Z}_n^*$ to $\mathbb{C} - \{0\}$ satisfying $\chi(ab) = \chi(a)\chi(b)$ for all $a, b \in \mathbb{Z}_n^*$.*

From this definition, we can quickly deduce that $\chi(1) = 1$ and that the value $\chi(a)$ is always a $(\lambda(n))^{\text{th}}$ root of the unity for all $a \in \mathbb{Z}_n^*$, where $\lambda(n)$ denotes the Carmichael function. We can also define a group structure on the set of characters on $\mathbb{Z}_n^*$. In this group, the product (group operation) $\chi_1\chi_2$ of the two characters χ_1 and χ_2 represents the map $a \longmapsto \chi_1(a)\chi_2(a)$ and the inverse χ^{-1} maps each element a to $\chi(a)^{-1}$.

Proposition 2. *Let p be a prime and d an integer such that $d|p-1$.*

1. *The group of characters defined on $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$.*
2. *The characters on $\mathbb{Z}_p^*$ of order dividing d form a cyclic subgroup of order d.*

A proof of this proposition can be found at the beginning of the chapter 8 of [12].

The second part of this proposition is especially interesting for us because we will consider characters of small order (e.g. 2, 3, 4) defined on $\mathbb{Z}_n^*$ for n large. We notice also that a character of order d maps the elements of $\mathbb{Z}_p^*$ to the set $\{\zeta_d^j \mid 0 \leq j \leq d-1\}$ where ζ_d denotes the unit $e^{2\pi i/d}$ and $i := \sqrt{-1}$.

We provide a way to define certain multiplicative characters on $\mathbb{Z}_n^*$ for a n being the product of two special primes. Since $\mathbb{Z}_n^*$ is not cyclic, using the above definition to this case is not suitable. It is more natural for our purposes to define such characters in the similar way as the Jacobi symbol is defined from the Legendre symbol. First, assume we are given an integer d and two different primes p, q such that $d|p-1$ and $d|q-1$. From two characters χ_1 and χ_2 of order d defined on $\mathbb{Z}_p^*$ respectively $\mathbb{Z}_q^*$, we define a character η of order d in the following way $\eta(a) := \chi_1(a \bmod p) \cdot \chi_2(a \bmod q)$.

For each character χ of order d we will sometimes associate a logarithm function denoted as $\log_\chi$. For an element $a \in \mathbb{Z}_n^*$, we know that $\chi(a)$ is of the form ζ_d^j for a $j \in \{0, 1, \ldots d-1\}$. We define $\log_\chi(a)$ equal to this j.

In the following subsections we present some complements that are specific to the cases $d = 2, 3, 4$. For more details, we refer to Ireland and Rosen [12].

2.1 Characters of order 2

Let p be an odd prime number. By Proposition 2, we know that there are only two characters of order 2, namely the trivial character ϵ that maps every elements to 1 and the Legendre symbol. We recall that the Legendre symbol (a/p) for an integer a with $(a, p) = 1$ is 1 if a is congruent to a square modulo p (quadratic residue) and -1 if it is not the case (quadratic non-residue). It turns out that there are $\frac{p-1}{2}$ quadratic residues resp. non quadratic residues in $\mathbb{Z}_p^*$.

For an odd integer n, the Jacobi symbol (a/n) for an $a \in \mathbb{Z}$ s.t. $(a, n) = 1$ is defined as $(a/n) = (a/p_1)^{i_1} \cdot (a/p_2)^{i_2} \cdots (a/p_k)^{i_k}$ where the factorization into primes of n is $p_1^{i_1} \cdots p_k^{i_k}$. Some additional properties are given below.

Proposition 3. *Let p be an odd prime, a, $b \in \mathbb{Z}$ and an odd $n \in \mathbb{Z}$. Then*

1. $a^{(p-1)/2} \equiv (a/p) \pmod{p}$.
2. $(ab/n) = (a/n)(b/n)$.
3. *If* $a \equiv b \pmod{n}$, *then* $(a/n) = (b/n)$.
4. *(Quadratic Reciprocity)* $(a/b)(b/a) = (-1)^{(\frac{a-1}{2})(\frac{b-1}{2})}$ *for a and b odd.*
5. $(2/n) = (-1)^{\frac{(p^2-1)}{8}}$.

Let us consider a modulus $n = pq$. From the above discussion we deduce that the complete list of characters of order 2 on $\mathbb{Z}_n^*$ is $(\cdot/p)$, $(\cdot/q)$, $(\cdot/n)$ and the trivial character. Note that the properties given in Proposition 3 are used in order to compute the Jacobi symbol in a time complexity of $\mathcal{O}(\log(n)^2)$.

2.2 Characters of order 3

Here, we introduce the ring of Eisenstein integers. Indeed, this ring is the natural structure to study the characters of order 3 or the cubic residuosity. Most of the results below are taken from [12].

In what follows, ω will always denote the complex number $\frac{-1+\sqrt{-3}}{2}$. We define the ring of the Eisenstein integers as the set $\mathbb{Z}[\omega] := \{a + b\omega | a, b \in \mathbb{Z}\}$ with the classical operations (addition, multiplication) of $\mathbb{C}$. We notice that ω is a non trivial cubic root of 1 and satisfies $\omega^2 + \omega + 1 = 0$.

For an element $\alpha \in \mathbb{Z}[\omega]$, we define the norm $N(\alpha) = \alpha\bar{\alpha}$, where $\bar{\alpha}$ denotes the complex conjugate of α. This is the classical (squared) norm induced by the complex plane. From the definition, we have $N(a + b\omega) = a^2 - ab + b^2$.

It can be shown that $\mathbb{Z}[\omega]$ is a unique factorization domain i.e. every elements can be decomposed in a product of irreducible elements uniquely up to a unit element. We can also call the irreducible elements the prime elements of $\mathbb{Z}[\omega]$. To avoid some confusion a prime of $\mathbb{Z}$ will be called a rational prime if the context is not clear. The units are the invertible elements and in this case all have a norm equal to one. Hence, the units of $\mathbb{Z}[\omega]$ are ± 1, $\pm\omega$, $\pm\omega^2$. All prime numbers of $\mathbb{Z}[\omega]$ are classified below.

Proposition 4. *The following statements describe all primes of $\mathbb{Z}[\omega]$.*

1. *Let p be a rational prime s. t. $p \equiv 1 \pmod 3$. There exists a prime π s. t. $N(\pi) = \pi\bar{\pi} = p$.*
2. *If q is a rational prime s. t. $q \equiv 2 \pmod 3$, then q is also a prime in $\mathbb{Z}[\omega]$.*
3. *$1 - \omega$ is prime and $N(1 - \omega) = 3$.*

The ideal generated by a $\sigma \in \mathbb{Z}[\omega]$ is denoted by (σ) and is equal to $\sigma \cdot \mathbb{Z}[\omega]$.

Proposition 5. *Let π be a prime in $\mathbb{Z}[\omega]$. Then $\mathbb{Z}[\omega]/(\pi)$ is a finite field with $N(\pi)$ elements.*

We can also prove that the set $\{a + b\omega | 0 \leq a, b \leq q\}$ resp. $\{0, 1, 2 \ldots, p-1\}$ form all representatives of the residue class field in the case where $q \equiv 2 \pmod 3$ resp. $p \equiv 1 \pmod 3$. We can also prove that for a prime π s.t. $N(\pi) \neq 3$ and $\alpha \in \mathbb{Z}[\omega]$ s.t. $\alpha \not\equiv 0 \pmod \pi$, we have $\alpha^{\frac{N(\pi)-1}{3}} \equiv \omega^i \pmod \pi$ for an $i \in \{0, 1, 2\}$. Here, ω^i is called the cubic residue character of α modulo π and is denoted as $(\alpha/\pi)_3$ or as $\chi_\pi(\alpha)$. If $\alpha \equiv 0 \pmod \pi$, we set $\chi_\pi(\alpha) = 0$.

Let α and β be in $\mathbb{Z}[\omega]$. Suppose the prime factorization of β is $u \prod_{i=1}^{k} \pi_i^{e_i}$ where $N(\pi_i) \neq 3$ for all $1 \leq i \leq k$ and u is a unit. Then the Jacobi-like symbol $(\alpha/\beta)_3$ is defined as $\prod_{i=1}^{k} (\alpha/\pi_i)_3^{e_i}$. In order to formulate the law of cubic reciprocity, we have to introduce the concept of primary. We say that an element α of $\mathbb{Z}[\omega]$ is primary iff $\alpha \equiv -1 \pmod 3$. Note that the term "primary" does not only apply to prime number[1]. Every elements possess exactly one associate that is primary. (An associate of an element σ is an element that is of the form $u\sigma$ for a unit u.)

Proposition 6. *Let π be a prime s.t. $N(\pi) \neq 3$ and α, β, $\gamma \in \mathbb{Z}[\omega]$. Let $\sigma = 3(A + B\omega) - 1$ be a primary with $A, B \in \mathbb{Z}$.*

1. *$(\alpha/\pi)_3 = 1$ iff $x^3 \equiv \alpha \pmod \pi$ is solvable, i.e., iff α is a cubic residue.*
2. *$(\alpha\beta/\gamma)_3 = (\alpha/\gamma)_3(\beta/\gamma)_3$.*
3. *$\alpha \equiv \beta \pmod \gamma \Longrightarrow (\alpha/\gamma)_3 = (\beta/\gamma)_3$.*
4. *(Law of Cubic Reciprocity) If α and β are primary. Then $(\alpha/\beta)_3 = (\beta/\alpha)_3$.*
5. *$(\omega/\sigma)_3 = \omega^{A+B}$.*
6. *$(1 - \omega/\sigma)_3 = \omega^{2A}$.*

[1] The analog notion of "primary" in $\mathbb{Z}$ is the notion of "negative" number.

We are now in the position to define the characters of order 3 on $\mathbb{Z}_p^*$ for a rational prime p and their extensions on a composite modulus that is a Jacobi-like symbol. We consider only the case where $p \equiv 1 \pmod 3$, since the characters are not trivial only in this case. Set $p = \pi\bar{\pi}$. Recall first that the field $\mathbb{Z}[\omega]/(\pi)$ can be represented by $\mathbb{Z}_p^*$ since the set $\{0, 1 \dots p-1\}$ contains all representatives and the multiplications are equivalent in the two cases. Thus, the cubic residue characters χ_π is completely defined on $\mathbb{Z}_p^*$. We directly deduce that χ_π^2 is another non trivial character of order 3 and is even equal to $\chi_{\bar{\pi}}$ on the rational integers. Let p, q be two different rational primes such that $p \equiv q \equiv 1 \pmod 3$ and π, $\sigma \in \mathbb{Z}[\omega]$ such that $N(\pi) = p$ and $N(\sigma) = q$. Let $n = pq$, the character on $\mathbb{Z}_n^*$ produced by χ_π and χ_σ is denoted by $\chi_{\pi\sigma}$ and is defined as $\chi_{\pi\sigma}(a) = \chi_\pi(a) \cdot \chi_\sigma(a)$. The other characters are defined exactly in the same multiplicative way. There are 8 non trivial characters of order 3 defined on $\mathbb{Z}_n^*$, namely χ_π, $\chi_{\bar{\pi}}$, χ_σ, $\chi_{\bar{\sigma}}$, $\chi_{\pi\sigma}$, $\chi_{\bar{\pi}\sigma}$, $\chi_{\pi\bar{\sigma}}$ and $\chi_{\bar{\pi}\bar{\sigma}}$.

Here, we explain how to find these characters and how they can be computed. The first statement consists of finding a prime $\pi \in \mathbb{Z}[\omega]$ such that $N(\pi) = p \equiv 1 \bmod 3$ for a rational prime p. We assume here some knowledge on the algorithms of Tonelli and Cornacchia (For more details see Cohen [7]).

For a given p, we have to find an element $a+b\omega \in \mathbb{Z}[\omega]$ such that $a^2-ab+b^2 = p$. This is equivalent to $(a-\frac{b}{2})^2 + \frac{3b^2}{4} = p$. By introducing the two new variables $s = a - \frac{b}{2}$ and $t = \frac{b}{2}$, we obtain $s^2 + 3t^2 = p$ for $s, t \in \mathbb{Z}$. Now, it suffices to apply the algorithm of Cornacchia to solve this equation in s and t. This algorithm consists of finding an $x \in \mathbb{Z}$ such that $x^2 \equiv -3 \pmod p$ (apply algorithm of Tonelli) and then applying the Euclid algorithm to x and p until we get the first rest term r_n such that $r_n^2 < p$. A solution is given by setting $s = r_n$.

Suppose we have a character χ_α where α can be for example $\pi\sigma$ or $\pi\bar{\sigma}$. The computation of a residue character $(\sigma/\alpha)_3$ can be done using a similar technique to the computation of the Jacobi symbol in the context of quadratic residuosity. Indeed, this consists of reducing $\sigma \bmod \alpha$ by an Euclidean division in $\mathbb{Z}[\omega]$ and then applying the cubic reciprocity law to exchange the two elements of the character. This last step can be done only after having extracted some units in order that α and σ become primary. Then by iterating this operation, we reduce the size of the elements involved in the cubic residue character until this one becomes trivial. Note that the asymptotic complexity of the computation is $\mathcal{O}(\log(n)^3)$ using standard arithmetic and $\mathcal{O}(\log(n)^2 \log\log(n) \log\log\log(n))$ using fast arithmetic. This is almost the same order of magnitude as the classical Jacobi symbol that is $\mathcal{O}(\log(n)^2)$ (See Cohen [7] p. 31). For more details about this algorithm and its complexity we refer to Scheidler [17].

2.3 Characters of order 4

Studying the characters of order 4 consists principally of the theory of biquadratic residuosity. This one is quite similar to that of cubic residuosity and is done in the ring of Gaussian integers $\mathbb{Z}[i]$. A rational prime p of the form $p \equiv 1 \pmod 4$ is the norm of a prime π in $\mathbb{Z}[i]$. The field $\mathbb{Z}[i]/(\pi)$ has the set of representatives $\{0, 1 \ldots p-1\}$ and is identical to $\mathbb{Z}_p$. The biquadratic residue character of an $\alpha \in \mathbb{Z}[i]$ is defined as $\chi_\pi(\alpha) := i^j$ where $j \in \{0, 1, 2, 3\}$ and such that $\alpha^{(N(\pi)-1)/4} \equiv i^j \pmod{\pi}$. Moreover, this character generates the two other nontrivial characters of order 4. Note also that the square of χ_π is equal to the quadratic residue character χ_p. We can also define a Jacobi-like symbol in this context similarly to that in the theory of characters of order 3. Moreover, there is also a law of reciprocity in a similarly way as before.

2.4 Characters of higher orders.

It is possible to extend our character constructions to some orders greater than 4. By introducing a power residue symbol defined on the integers of a cyclotomic field. A general treatment of these cases would be beyond the scope of this paper. Moreover, the computation seems to be more difficult to deal with and the ring of these integers becomes a non unique factorization domain when the order is large. Since such a ring is not a principal ideal domain, we should work with ideals that are generated by more than one element. However, we do not loose the existence of the reciprocity laws, namely there exists a so called Kummer's reciprocity law (see [14]).

3 On the Hardness of Related Problems

Here we expose some different computational problems that will be related with the security of our scheme. In particular, we focus this treatment to the case of characters of order $d \in \{2, 3, 4\}$.

For two problems $\mathbf{P}$ and $\mathbf{P}'$, we use the Karp reduction, i.e. we say that $\mathbf{P}$ is at most as hard as $\mathbf{P}'$ if the problem $\mathbf{P}$ can be solved in a polynomial time by using one access to an oracle $\mathcal{O}_{P'}$ that can solve $\mathbf{P}'$. We will denote this as $\mathbf{P} \leq \mathbf{P}'$. Moreover, this is also equivalent to say that $\mathbf{P}'$ is at least as hard as $\mathbf{P}$. We say also that two problems $\mathbf{P}$ and $\mathbf{P}'$ are equivalent if $\mathbf{P} \leq \mathbf{P}'$ and $\mathbf{P}' \leq \mathbf{P}$ are satisfied. We denote this property as $\mathbf{P} \equiv \mathbf{P}'$.

Let θ be a dth primitive root of 1 in $\mathbb{C}$, where d is typically equal to $2, 3, 4$. Below we expose the different problems.

FACT. For a given $n \in \mathbb{Z}$, find the factorization of n in $\mathbb{Z}$.

$\mathbf{CYCLOFACT}^d$. Let σ be an element of $\mathbb{Z}[\theta]$. Find the factorization of σ.

$\mathbf{ROOT}(-\mathbf{3})$. Let $n \in \mathbb{Z}$ be such that -3 is a quadratic residue modulo n. Given n, find an $u \in \mathbb{Z}$ such that $u^2 \equiv -3 \pmod{n}$.

$\mathbf{ROOT}(-\mathbf{1})$. Let $n \in \mathbb{Z}$ be such that -1 is a quadratic residue modulo n. Given n, find an $u \in \mathbb{Z}$ such that $u^2 \equiv -1 \pmod{n}$.

$\mathbf{FERMAT}^d$. Let $n \in \mathbb{Z}$ be such that $n = \pi\bar{\pi}$ for a $\pi \in \mathbb{Z}[\theta]$. Given n, find π.

$\mathbf{CHARACTER}^d$. Let $n \in \mathbb{Z}$. Devise an algorithm which given $x \in \mathbb{Z}_n^*$ computes $\chi(x)$ where χ is a *hard character* of order d on $\mathbb{Z}_n^*$.

$\mathbf{MOVA}^d$. Let $n \in \mathbb{Z}$, s be a positive integer and χ a *hard character* of order d on $\mathbb{Z}_n^*$. Given s pairs $(\alpha_i, \chi(\alpha_i))$, where $\alpha_i \in \mathbb{Z}_n^*$ for all $1 \leq i \leq s$ and $x \in \mathbb{Z}_n^*$

compute $\chi(x)$.

Remark. By "hard character" we mean a nontrivial character and for $d = 2$ we also exclude the Jacobi symbol $(\cdot/n)$.

Lemma 7. $\mathbf{FACT} \equiv \mathbf{CYCLOFACT}^d$ *and* $\mathbf{FERMAT}^d \leq \mathbf{CYCLOFACT}^d$ *for* $d = 2, 3, 4$. $\mathbf{FERMAT}^3 \equiv \mathbf{ROOT}(-\mathbf{3})$ *and* $\mathbf{FERMAT}^4 \equiv \mathbf{ROOT}(-\mathbf{1})$.

The proof is given in the appendix A. See also Landrock [13] for another cryptographic application of Fermat numbers (i.e. $\mathbf{FERMAT}^4$ and $\mathbf{ROOT}(-1)$).

$\mathbf{CHARACTER}^d$ plays an important role in the security of our signature. Indeed, the ability of signing will be related to the computation of hard characters when n cannot be factorized. Notice that this is a generalization of the quadratic residuosity problem on which the security of the probabilistic Goldwasser-Micali encryption is based [11]. In practice, we will consider a modulus of the form $n = pq$. For $d = 2$, such characters are simply the Legendre symbols modulo p and q. For $d = 3$, we can use the non trivial characters. For example, $\chi_{\pi\sigma}$ is a case where the security is related to $\mathbf{FERMAT}^3$ since $N(\pi\sigma) = n$. Indeed, an enemy that knows a square root of -3 modulo n would be able to retrieve this character by Lemma 7. Thus, $\mathbf{FERMAT}^3 \geq \mathbf{CHARACTER}^3$ and similarly $\mathbf{FERMAT}^4 \geq \mathbf{CHARACTER}^4$. Note also that $\mathbf{MOVA}^d \leq \mathbf{CYCLOFACT}^d$ but $\mathbf{MOVA}^d \leq \mathbf{CHARACTER}^d$ in some cases only, because the character devising in $\mathbf{CHARACTER}^d$ may be independent from the character required for $\mathbf{MOVA}^d$.

4 Description of the MOVA Scheme

We present here the components of our undeniable signature scheme called "MOVA"[2].

Public Parameters. Let s, t, k, ℓ be some positive integers whose size depend on the required security level of the scheme. We let θ denote a primitive dth root of 1 in $\mathbb{C}$, where $d \in \{2, 3, 4\}$.

Primitives. We assume the existence of two pseudorandom generators $G_1 : \{0,1\}^* \longrightarrow (\mathbb{Z}_n^*)^s$ and $G_2 : \{0,1\}^* \longrightarrow (\mathbb{Z}_n^*)^t$. We also assume the existence of a commitment scheme denoted as COMMIT : $x \mapsto (< x >, \mathrm{OPEN}_x)$ and CHECK$(x, < x >, \mathrm{OPEN}_x)$.

Setup. The signer generates an n and a hard character χ of order d on $\mathbb{Z}_n^*$. Then he takes a string $Id \in \{0,1\}^*$ and computes $G_1(Id) = (\alpha_1, \ldots, \alpha_s)$. Finally, he computes the logarithm of the character residues of the α_i's. We set $\Sigma_\alpha := (e_1, \ldots, e_s)$ an element of $\{0, 1 \ldots d-1\}^s$ where $e_i = \log_\chi(\alpha_i)$ for all $1 \leq i \leq s$. If the e_i's do not span $\mathbb{Z}_d$ or $e_i = (\frac{\alpha_i}{n})$ for all $1 \leq i \leq s$ in the $d = 2$ case then restart with another Id.[3] For $d = 3$ or 4 we can either start

[2] "MOVA" is related to the names of the authors of the present paper.

[3] As discussed in Subsection 5.6 an authority could be involved in this scheme in order to tolerate low s parameter.

by generating prime numbers p and q, take $n = pq$, get π such that $\pi\bar{\pi} = n$ and set $\chi = (./\pi)_d$, or directly generate $n = \pi\bar{\pi}$ from a random $\pi \in \mathbb{Z}[\theta]$. The latter is performed with smaller complexity but the factorization of n is unknown.

Public Key. $K_P = (n, Id, \Sigma_\alpha)$.

Secret Key. $K_S = \chi$.

Signature generation. Let $m \in \{0,1\}^*$ be a message to sign. The signer generates $G_2(m) = (\beta_1, \dots, \beta_t)$. Then the signer computes $c_i = \log_\chi(\beta_i)$. The signature of m is Σ, where Σ is defined as

$$\Sigma := (c_1, c_2, \dots, c_t).$$

Confirmation Protocol. We denote here the prover as P and the verifier as V. The signer is given (m, Σ) that is also public. Here is the sketch of the protocol.

Repeat k times :

1. V picks some values $a_1, a_2, \dots a_s, b_1, \dots b_t \in \{0, 1 \dots d-1\}$ and a $\gamma \in \mathbb{Z}_n^*$ randomly. Set $\delta := \gamma^d \cdot \prod_{i=1}^s \alpha_i^{a_i} \cdot \prod_{i=1}^t \beta_i^{b_i} \bmod n$. V then sends δ to P.
2. P computes $r = \log_\chi(\delta)$ and sends r to V.
3. V checks if $r = \sum_{i=1}^s a_i e_i + \sum_{i=1}^t b_i c_i \bmod d$. If this equality does not hold, V rejects the signature.

For some security reasons, this protocol must include a commitment function. Indeed, we notice that somebody could use this protocol several times in order to sign a message of his choice. This can be easily done by sending the β_i's instead of δ to the prover. A way to prevent against a such attack is to use a commitment function as mentioned in Gennaro and al. [10]. In our confirmation protocol, the modification works in the following way. After having computed r in Step 2., P runs COMMIT(r) and sends $< r >$ to V and then V sends $\gamma, a_1, \dots a_s, b_1, \dots b_s$ to P. The prover checks that $\delta = \gamma^d \cdot \prod_{i=1}^s \alpha_i^{a_i} \cdot \prod_{i=1}^t \beta_i^{b_i} \bmod n$ really holds. Finally, P sends r, OPEN_r to the verifier that can then effect Step 3 and do CHECK($r, < r >, \mathrm{OPEN}_r$).

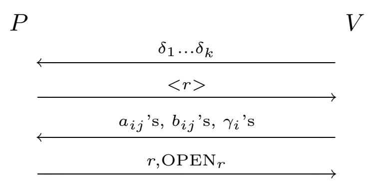

Fig. 1. Confirmation Protocol with commitment

Note that the confirmation protocol can be completely parallelized (see Figure 1). V sends $\delta_1, \dots \delta_k$ defined as $\delta_i = \gamma_i^d \cdot \prod_{j=1}^s \alpha_j^{a_{ij}} \cdot \prod_{j=1}^t \beta_j^{b_{ij}} \bmod n$ where the a_{ij}'s, b_{ij}'s and γ_i's are picked at random. This protocol continues similarly with $r := (r_1, \ldots, r_k)$ as the prover will commit. Finally, after V has sent the a_{ij}'s, b_{ij}'s and the γ_i's to P, this one opens the commitment of the values r_i's. Note that V can generate the a_{ij}'s, b_{ij}'s and the γ_i's in a pseudorandom way and send the seed of the pseudorandom generator. This method can considerably decrease the communication complexity.

Denial Protocol. Here, the verifier V is given a message $m \in \{0,1\}^*$ and an alleged non-signature Σ where $\Sigma = (c_1, \ldots, c_t)$. The protocol works as follows.

Repeat ℓ times:

1. The prover picks a matrix $A = (a_{ij}) \in \mathbb{Z}_d^{t\times s}$ at random and a matrix $B = (b_{ij}) \in \mathbb{Z}_d^{t\times t}$ of rank t. He then computes $q_i := \sum_{j=1}^{s} a_{ij}e_j + \sum_{j=1}^{t} b_{ij}c_j$ and $r_i := \sum_{j=1}^{s} a_{ij}e_j + \sum_{j=1}^{t} b_{ij}\log_\chi(\beta_j)$ for all $1 \le i \le t$. Set $Q = (q_i)$ and $R = (r_i)$. P computes $\delta_i := \gamma_i^d \cdot \prod_{j=1}^{s} \alpha_j^{a_{ij}} \cdot \prod_{j=1}^{t} \beta_j^{b_{ij}} \bmod n$. He finally runs COMMIT$(\gamma, A, B)$, COMMIT$(R)$ and sends $<\gamma, A, B>$, $<R>$ and the values δ_i's, Q to V.
2. V picks a challenge $u \in \{0,1\}$ at random and sends u to P.
3. If $u = 0$, he sends $\gamma, A, B, \mathrm{OPEN}_{(\gamma,A,B)}$ to V. If $u = 1$, he sends R, OPEN_R to V.
4. If $u = 0$, V does CHECK$(\gamma, A, B, <\gamma, A, B>, \mathrm{OPEN}_{(\gamma,A,B)})$ and checks that $\delta_i = \gamma_i^d \cdot \prod_{j=1}^{s} \alpha_j^{a_{ij}} \cdot \prod_{j=1}^{t} \beta_j^{b_{ij}} \bmod n$ for all $1 \le i \le t$, $q_i = \sum_{j=1}^{s} a_{ij}e_j + \sum_{j=1}^{t} b_{ij}c_j$ for all $1 \le i \le t$. If $u = 1$, V does CHECK$(R, <R>, \mathrm{OPEN}_R)$ and checks that $Q \ne R$. He then checks that $r_i = \log_\chi(\delta_i)$ for all $1 \le i \le t$ by interacting with P in a confirmation protocol on the "signature" R of δ.

5 Security Analysis

Here we analyze the security of our proposed scheme. We do not recall here every security properties suitable for an undeniable signature and refer to [8] and [10].

5.1 Validity of the Public Key

We say that a public key is valid if

1. the set $\{e_1 \ldots e_s\}$ spans $\mathbb{Z}_d$,
2. when $d = 2$ there exists at least one j s.t. $e_j \ne (\frac{\alpha_j}{n})$,
3. the set $\{\alpha_1 \ldots \alpha_s\}$ spans $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$.

If these conditions are fulfilled, we can prove that there exists at most one character χ such that $\chi(\alpha_i) = e_i$ for $1 \le i \le s$ and that this character is a hard one of order d. Note that the third condition is the only one which cannot be checked by V. This will be probabilistically satisfied depending on s. The first two are already avoided in the Setup of the scheme. Assuming that G_1 behaves like a random oracle, an analysis of the probability shows that the third condition is not checked with probability $\frac{3}{2^s} - \frac{2}{4^s}$ for $d = 2$ and $\frac{4}{3^s} - \frac{3}{9^s}$ for $d = 3$. For $d = 4$, this probability has magnitude $\mathcal{O}(\frac{1}{2^s})$. See Appendix B for more details on this computation. So, for $d = 3$ and $s = 52$ this probability is approximately 2^{-80}. Thus invalid keys cannot be forged in practice.

5.2 Signature Forgery and Impersonation.

In this subsection we show that our signature scheme is resistant to an existential forgery attack and that nobody else than the prover can confirm or deny a given signature.

Let first consider an attacker $\mathcal{A}_1$ living in the model of security of an undeniable signature. In a such model, $\mathcal{A}_1$ is supposed to have access to an oracle able to sign some queried messages, to a second oracle playing the role of the prover in the confirmation protocol and to an oracle able to play the role of the prover in the denial protocol. In fact, by looking at the confirmation protocol and denial protocol and assuming that G_2 is a random oracle, we can see that $\mathcal{A}_1$ does not learn more information in this model than having a random source $\mathcal{S}$ generating some pairs $(\mu, \log_\chi(\mu)) \in \mathbb{Z}_n^* \times \mathbb{Z}_d$. Hence, this attacker reduces to a new attacker $\mathcal{A}_2$ having $\mathcal{S}$ to his disposal. Assuming now that the α_i's generate $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$, an attacker picking some random values $\gamma \in \mathbb{Z}_n^*$, a_i's in $\{0, 1 \ldots d-1\}$ and then computing $\gamma \cdot \prod_{i=1}^s \alpha_i^{a_i}$ is also able to simulate the source S. Thus, $\mathcal{A}_2$ can be replaced by an attacker $\mathcal{A}_3$ that possesses only the public key. We conclude by saying that any attacker of our scheme will be then considered as $\mathcal{A}_3$. Finally, notice that $\mathcal{A}_3$ is exactly in the situation that corresponds to the assumption of the problem $\mathbf{MOVA}^d$ (see section 3.) .

To prepare these security proofs we first need the following results.

Theorem 8. *Let $\varphi : G \longrightarrow \mathbb{Z}_d$ be a group homomorphism. If one can compute a f such that $\Pr_{x \in G}(f(x) \neq \varphi(x)) \leq \frac{\xi}{12}$ with a constant $\xi < 1$, then one can compute φ in a number of calls to f bounded by a polynomial in $\log(\#G)$.*

We have postponed the proof of this theorem to the appendix C.

Assuming that $\alpha_1 \ldots \alpha_s$ span $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$ and using Theorem 8, we show that an entity that is able to confirm or deny a given signature must be able to compute the character, i.e. he possesses the secret key. Indeed, in these two protocols, it is requested to the Prover to evaluate the logarithm of the character on different values (e.g. δ). Passing these tests corresponds to the ability of the computation of $\log_\chi$. More precisely, in the confirmation protocol we can see the Prover as a function that takes on input the value δ depending of the a_i's and b_i's and computes $\log_\chi(\delta)$. We can see this process in one function that is defined on the Abelian group $\mathbb{Z}_n^*$ and whose values lie in $\mathbb{Z}_d$. We see that we can directly apply our above general results to this function, since it satisfies the properties of the function φ of Theorem 8. Thus, an entity that can evaluate this function with a small error probability is able to compute the character χ by Theorem 8.

Corollary 9 (Privacy of Confirmation). *Let Σ be a valid signature associated to a valid public key K_P. If* $\mathbf{MOVA}^d$ *is hard, then no fake prover can pass the confirmation but with a probability bounded by $(1 - \frac{\xi}{12})^k$ for any $\xi < 1$.*

This corollary protects a user against an impersonation during the confirmation protocol. So, an enemy is not able to confirm a message signed by a given person without knowing his secret key. The case of the denial protocol is more subtle because the number of characters the prover has really to compute is not fixed. In fact, when $u = 1$ he has a huge probability to pass the test by answering at random. It can happen with probability $2^{-\ell}$, that the prover does not need to compute any character at all. In anyway, he will have to distinguish between $u = 0$ or $u = 1$ in order to pass the test. Thus the probability of success of the enemy is in anyway less than $2^{-\ell}$ since the prover cannot know the value u.

After this discussion and having exposed Theorem 8, we can obviously say that our scheme is resistant against existential forgery.

Corollary 10 (Hardness of existential forgery). *Assuming that* $\mathbf{MOVA}^d$ *is hard and that G_2 is a random oracle, then no attacker can forge a valid signature for a message m but with a probability bounded by $(1 - \frac{\xi}{12})^t$ for any $\xi < 1$.*

5.3 The Confirmation Protocol

We provide below some properties on the security of the confirmation protocol. From now on, $\mathrm{Sign}(m, K_P, P)$ denotes the signature of the message m of the user P possessing the public key K_P.

Proposition 11 (Confirmation protocol).

Completeness. *Let $\Sigma = Sign(m, K_P, P)$ be a valid signature. If P and V follow the Confirmation Protocol, then V always accepts the validity of the signature Σ.*

Soundness. *Let $\Sigma \neq Sign(m, K_P, P)$ be an invalid signature with respect to K_P. Then a cheating Prover P can confirm the signature Σ with a probability not better than $\frac{1}{p^k}$, where p is the smallest prime factor of d.*

Zero-Knowledge. *The confirmation protocol is zero-knowledge.*

Proof (Sketch). The completeness is obvious by looking at the protocol.

For the proof of the soundness, we investigate what the behavior of the cheater P should be in order to bypass the confirmation protocol. For sake of simplicity, assume also that the signature Σ differs to $\mathrm{Sign}(m, K_P, P)$ at only one component. W.l.o.g. assume that $c_1 \neq \log_\chi(\beta_1)$, where the term β_1 is the first term of $G_2(m)$. Passing one round of the confirmation protocol is equivalent to be able to find the value $v := \sum_{i=1}^{s} a_i e_i + \sum_{i=1}^{t} b_i c_i \bmod d$ knowing the e_i's, $\log_\chi(\beta_i)$'s and $\log_\chi(\delta)$. Since $v - \log_\chi(\delta) = b_1(c_1 - \log_\chi(\beta_1))$, we deduce that the cheater passes the test iff he can find the value b_1. This is not possible because the value δ can be generated in several different ways, i.e. for several different

$\gamma \in \mathbb{Z}_n^*$, a_i's and b_i's. Thus, the d different distributions of the δ corresponding to the d different fixed values b_1 are indistinguishable when d is prime. Otherwise, the assertion remains true when we replace d by p in the worst case. Therefore, he cannot do better than supposing the correct v in a set of at least p elements.

Zero-knowledge: A honest verifier can easily simulate the transcript of the protocol. Since a dishonest verifier has a negligible probability to pass the protocol, our confirmation protocol is therefore zero-knowledge. □

5.4 The Denial Protocol

Proposition 12 (Denial protocol).

> ***Completeness.*** *Let $\Sigma \neq Sign(m, K_P, P)$ be an invalid signature. If P and V follow the Confirmation Protocol, then V always concludes the invalidity of the signature Σ.*
> ***Soundness.*** *Let $\Sigma = Sign(m, K_P, P)$ be a valid signature with respect to K_P. Then a cheating Prover P can deny the signature Σ with a probability not greater than $\frac{1}{2^\ell}$.*
> ***Zero-Knowledge.*** *The denial protocol is zero-knowledge.*

Proof (Sketch). Completeness: It is obvious by examining the denial protocol.

Soundness: First, notice that a cheating prover can easily pass the denial protocol if he would be able to find when $u = 0$ or $u = 1$. Conversely, if he has not this ability, he cannot pass the denial protocol with a probability greater than $\frac{1}{2^\ell}$ if we assume that the soundness of confirmation protocol is perfect.

Zero-knowledge: For $u = 0$ a verifier can trivially simulate the transcript of the protocol (assuming that $< R >$ can be simulated). For $u = 1$ he can pick some a_{ij}'s and γ_i's at random then set $q_i := \sum_{j=1}^{s} a_{ij} e_j$ and $\delta_i := \gamma_i^d \cdot \prod_{j=1}^{s} \alpha_j^{a_{ij}}$. He can pick $R \neq Q$ at random then simulate the protocol. One can easily prove that the generated (δ, Q, R) have the same distribution as in the protocol. He then needs to simulate the confirmation protocol. □

5.5 Complexity

The complexity of the signature generation is the computation of t characters. For the confirmation protocol, the verifier needs about $k \cdot (s+t) \cdot (d-1)/d$ multiplications in $\mathbb{Z}_n^*$ assuming that the values α_i^2, β_i^2, α_i^3, β_i^3 mod n are precomputed. In the same protocol, the prover has to perform k character computations. The denial protocol requires about $\ell \cdot t \cdot (s+t) \cdot (s-1)/s$ modular multiplications and $k \cdot \ell/2$ character computations to the prover. The verifier has to compute $1/2 \cdot (\ell t + k) \cdot (s+t) \cdot (d-1)/d$ modular multiplications[4]. Note that character computation is asymptotically comparable to multiplication in terms of complexity i.e. $\mathcal{O}((\log n)^2)$.

[4] Note that, it is possible to adapt the protocol of [10] in order to reduce the complexity of the denial protocol.

The setup protocol requires the computation of s characters as well as finding the hard character. This step can be realized in two different ways. The first one requires the generation of two primes p,q with a complexity of $\mathcal{O}((\log n)^4)$. The second way (for $d = 3, 4$ only) requires $\mathcal{O}((\log n)^2)$ since we have to pick a large $\pi \in \mathbb{Z}[\theta]$ and compute $n = \pi\bar{\pi}$.

5.6 Key Setup Variants

Here, we discuss some variants of the setup allowing to reduce the size of s. As we have seen, in the first variant the signer selects his own key without any help. The consequence is that s has to be large to ensure the security.

In the second variant, we propose that the signer selects his own key online with the participation of a certificate authority. This allows to reduce the value of s since the signer is limited with the number of attempts. Note also that the complexity of this key setup is similar to the first variant, i.e. the complexity can be quadratic with $d = 3, 4$ and the second way for generating n as discussed in the previous section.

The last variant allows to have a s even lower but requires a greater complexity of the key setup since the signer needs to know the factorization of the modulus n. Here, the signer generates the key itself and proves its validity to the certificate authority or to the verifier. Below, we describe the protocol in which the prover (signer) convinces a verifier (authority) that the α_i's generate $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$.

Repeat m times:

1. The prover picks $\delta_1 \in \mathbb{Z}_n^*$ at random and runs COMMIT(δ_1). He sends $< \delta_1 >$ to the verifier.
2. The verifier picks $\delta_2 \in \mathbb{Z}_n^*$ at random and sends δ_2 to the prover.
3. The prover computes some coefficients $\gamma \in \mathbb{Z}_n^*$, $a_1, \ldots, a_s \in \{0, \ldots d-1\}$ that satisfy $\delta_1\delta_2 \equiv \gamma^d \cdot \prod_{j=1}^{s} \alpha_j^{a_j} \pmod n$. He sends δ_1, OPEN_{δ_1}, $a_1, \ldots a_s$ to the verifier.
4. The verifier runs $\mathrm{CHECK}(\delta_1, < \delta_1 >, \mathrm{OPEN}_{\delta_1})$ and checks if $\delta_1 \in \mathbb{Z}_n^*$ and if the equality $\delta_1\delta_2 \equiv \gamma^d \cdot \prod_{j=1}^{s} \alpha_j^{a_j} \pmod n$ holds.

It can be shown that this protocol is complete, sound and zero-knowledge.

5.7 Parameters Choice

Note that our bounds are not tight and that we believe that $\frac{\xi}{12}$ can be replaced by $1 - \frac{1}{d}$ everywhere[5]. Hence, the probability of an impersonation is similar to that of soundness. Since an attacker cannot check the validity or invalidity of a signature offline, the minimal size of the suitable parameters should correspond to a probability of 2^{-20}. The signature can therefore have a length of 20 bits,

[5] At the time we are wrapping up this paper, we can prove that we can replace $\xi/12$ by $\xi/2$.

i.e. $t = 20/(\log_2(d))$. The same probability for the soundness of the confirmation resp. denial protocol, implies that $k = 20/(\log_2(p))$ resp. $\ell = 20$. If the public key is generated offline (first variant of setup), we have to consider a probability of 2^{-80}. Hence, the value of s is 80 for $d = 2, 4$ and $80/(\log_2(3))$ for $d = 3$. Finally, the size of n should be as in RSA, i.e. 1024 bits. For $d = 3$ we get the following size: $s = 52$, $t = 13$, $k = 13$ and $\ell = 20$. If the α_i's are generated online (second variant of setup) which registering the public key to an authority, we can reduce s to $s = 13$. If failure cases are strongly controlled by the authority we can even afford a security level of 2^{-10} and have $s = 6$. If we can further prove that the α_i's span $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$ to authority (third variant of setup) we can shorten s drastically to $s = 2$ using certificates.

For academic purposes, we can propose $d = 2$, $s = 2$, $t = 1$, $k = 20$, $\ell = 20$ (i.e. a signature of only one bit !). An enemy is able to forge a signature with a probability of 1/2 but he would not be able to confirm it. However, the true signer could not deny it.

6 Conclusion

We proposed a new undeniable signature and prove its security. Since the signature does not have to be an element of the size of a modulus, our scheme offers the advantage to sign with short signatures. Moreover, we can see that the complexity of the signature generation, the confirmation and denial protocol is quadratic in the size of n since the most costly operation is a character computation. Furthermore, some key setup variants allow to get quadratic complexity. Another nice property of our protocol is the possibility to confirm several signatures at the same time. For this batch verification, we only need to consider these signatures as a big one.

As a further research, we will extend our scheme to characters of higher order. It would be also worth studying if our scheme can be modified in order to offer some additional advanced properties such as the convertibility or the delegation. In our scheme, we already have a kind of delegation when $d = 3$ or 4. Indeed, the ability to sign, confirm an deny can be delegated by releasing one hard character (i.e. some $\pi \in \mathbb{Z}[\theta]$) to the proxy while the original signer can keep the complete list of characters (i.e. the factorization of n). This property holds for $d \neq 2$ since disclosing one π does not fully disclose the complete factorization of n. In the context of undeniable signature the delegation should not give the possibility for the proxy to sign but only to confirm or deny.

References

1. 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.
2. 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.

3. J. Boyar, D. Chaum, I. Damgård and T. Pedersen, *Convertible Undeniable Signatures*, Advances in Cryptology - Crypto '90, LNCS **537**, pp. 189-205, Springer, 1990.
4. D. Chaum, *Zero-Knowledge Undeniable Signatures*, Advances in Cryptology - Eurocrypt '90, LNCS **473**, pp. 458-464, Springer, 1990.
5. D. Chaum, *Designated Confirmer Signatures*, Advances in Cryptology - Eurocrypt '94, LNCS **950**, pp. 86-91, Springer, 1994.
6. D. Chaum and H. van Antwerpen, *Undeniable Signatures*, Advances in Cryptology - Crypto '89, LNCS **435**, pp. 212-217, Springer, 1989.
7. H. Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics **138**, Springer, 2000.
8. I. Dåmgard and T. Pedersen, *New Convertible Undeniable Signatures Schemes*, Advances in Cryptology - Eurocrypt '96, LNCS **1070**, pp. 372-386, Springer, 1996.
9. Y. Desmedt and M. Yung, *Weaknesses of Undeniable Signature Schemes*, Advances in Cryptology - Crypto '91, LNCS **576**, pp. 205-220, Springer, 1991.
10. R. Gennaro, T. Rabin and H. Krawczyk, *RSA-Based Undeniable Signatures*, Journal of Cryptology, **13**, pp. 397-416, Springer, 2000.
11. S. Goldwasser and S. Micali, *Probabilistic Encryption,* Journal of Computer and System Sciences, **28**, pp. 270-299, 1984.
12. K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory: Second Edition*, Graduate Texts in Mathematics **84**, Springer, 1990.
13. P. Landrock, *A New Concept in Protocols: Verifiable Computational Delegation*, Security of Protocols, LNCS **1550**, Springer 1998.
14. F. Lemmermeyer, *Reciprocity Laws*, Monographs in Mathematics, Springer, 2000.
15. M. Michels, H. Petersen and P. Horster, *Breaking and Repairing a Convertible Undeniable Signature*, In Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp- 148-152, 1996.
16. P. Nguyen, *La Géométrie des Nombres en Cryptologie*, Thèse de Doctorat.
17. R. Scheidler, *A Public-Key Cryptosystem Using Purely Cubic Fields*, Journal of Cryptology, **11**, pp. 109-124, Springer, 1998.
18. R. Scheidler and H. Williams, *A Public-Key Cryptosystem Utilizing Cyclotomic Fields*, Design, Codes and Cryptography, **6**, pp. 117-131, Kluwer Academic Publishers, 1995.

A Proofs of Some Equivalence Problems

FACT and CYCLOFACT. The case $d = 2$ is trivial. The cases $d = 3$ and $d = 4$ are similar. We concentrate on $d = 3$ here.
$\mathbf{FACT} \leq \mathbf{CYCLOFACT}^3$: Suppose we are given an oracle $\mathcal{O}_{\mathbf{CYCLOFACT}^3}$ that solves the problem $\mathbf{CYCLOFACT}^3$. We compute the factorization of a $n \in \mathbb{Z}$ by calling $\mathcal{O}_{\mathbf{CYCLOFACT}^3}$ on the input n. We then obtain a decomposition of the form $n = u \cdot (1-\omega)^{2i} \cdot \pi_1 \pi_2 \ldots \pi_k \cdot q_1 \cdot q_2 \ldots q_l$. By choosing the π_j's that have the same norm and by combining them with u we get some terms of the form $\pi_j \bar{\pi_j} = p_j$, where the p_j's are rational prime integers. Doing the same with $(1-\omega)^{2i}$, provides the term 3^i. After this process, only rational primes will remain in this decomposition, i.e. the factorization of n in $\mathbb{Z}$.
$\mathbf{CYCLOFACT}^3 \leq \mathbf{FACT}$: Here, we have access to the oracle $\mathcal{O}_{\mathbf{FACT}}$ and we have to factorize a $\sigma \in \mathbb{Z}[\omega]$. To this end, we compute $n = \sigma\bar{\sigma}$ and call the

oracle $\mathcal{O}_{\mathbf{FACT}}$ on n to obtain the factorization $n = \prod p_i$. Since the rational prime numbers p_i congruent to 2 modulo 3 are also prime in $\mathbb{Z}[\omega]$, it suffices to find the nontrivial primes π_i of the form $\pi_i\bar{\pi}_i \equiv 1 \pmod 3$. To this purpose, we apply the algorithm of subsection 2.2 to the rational primes p_i's congruent to 1 modulo 3. Hence, we obtain the decomposition $p_i = \pi_i\bar{\pi}_i$ of those primes. It remains to decide which one of π_i or $\bar{\pi}_i$ divides σ. This can be decided by an Euclidean division. Thus, all the non trivial prime divisors of σ are found and therefore its factorization.

FERMAT and ROOT. We can show that **FERMAT**^{3} is equivalent to solve the equation $n = s^2 + 3t^2$. Then, we can easily see that a solution of this equation gives a square root of -3 modulo n if $(t, n) = 1$, namely $s \cdot t^{-1}$. The converse assertion follows by the fact that a solution s, t is obtained by finding the shortest vector of the lattice $\{(s,t) \in \mathbb{Z}^2 | s \equiv tu \pmod n\}$. This can be done by a lattice reduction in dimension two using the reduction algorithm of Gauss (see [16]). Moreover, this algorithm has a polynomially complexity. $\square$

B Probability of generating $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$.

We consider here a modulus of the form $n = pq$, where p and q are two rational primes s.t. $p \equiv q \equiv 1 \bmod d$. We study here the probability for s elements $\alpha_1 \ldots \alpha_s \in \mathbb{Z}_n^*$ picked at random to generate $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^d$. Observe that this group is isomorphic to $\mathbb{Z}_p^*/(\mathbb{Z}_p^*)^d \times \mathbb{Z}_q^*/(\mathbb{Z}_q^*)^d$ by Chinese Remainder Theorem. Finally, this is also isomorphic to $\mathbb{Z}_d \oplus \mathbb{Z}_d$. Thus, it suffices to compute the probability that s elements of $\mathbb{Z}_d \oplus \mathbb{Z}_d$ generate the whole group.

Case $d = 2$. First we observe that $\mathbb{Z}_2^2$ has 3 non trivial subgroups, namely $G_1 := \{(1,0),(0,0)\}$, $G_2 := \{(0,1),(0,0)\}$, $G_3 := \{(1,1),(0,0)\}$. The only possibility of elements to not generate the whole group is to stay always in exactly one of the above subgroup, i.e. to pick always the same nonzero elements and/or the zero elements. This probability is then $\Pr_2 = \frac{1}{4^s} + 3\left(\frac{1}{2^s} - \frac{1}{4^s}\right) = \frac{3}{2^s} - \frac{2}{4^s}$. The first term corresponds to the probability that all elements are equal to zero and the second corresponds that these elements lie in one of the three subgroup without being all equal to zero.

Case $d = 3$. This works similarly. The probability is $\frac{1}{9^s} + 4\left(\frac{1}{3^s} - \frac{1}{9^s}\right) = \frac{4}{3^s} - \frac{3}{9^s}$.

Case $d = 4$. Here, an exact computation would be more complicated, but the existence of subgroups of order 8 implies that the dominant term in the probability will be of magnitude $(\frac{8}{16})^s = 2^{-s}$. An example of subgroup of order 8 is $< (1,2),(2,2) >= \{(0,0),(1,2),(2,0),(3,2),(2,2),(3,0),(0,2),(1,0)\}$.

C Proof of Theorem 8.

We first have the following theorem. Its proof is freely inspired from [1, 2].

Theorem 13. *Let G be a finite Abelian group and $d|(\#G)$. Let $x_1, \dots x_r \in G$, $y_1, \dots y_r \in \mathbb{Z}_d$ and $f : G \longrightarrow \mathbb{Z}_d$. If*

$$\Pr_{\substack{a_1 \dots a_r \in \mathbb{Z}_d \\ x \in G}} \left(f\left(d \cdot x + \sum_{i=1}^{r} a_i \cdot x_i \right) = \sum_{i=1}^{r} a_i \cdot y_i \right) = 1 - \varepsilon > \frac{1}{2},$$

then there exists a morphism $\varphi : G \longrightarrow \mathbb{Z}_d$ such that $\varphi(x_i) = y_i$ for all $1 \le i \le r$ and $\Pr_{x \in G}(f(x) = \varphi(x)) = 1 - \varepsilon$.

Proof. Let $H := \{(b_1 \dots b_r) \in \mathbb{Z}_d^r \text{ s.t. } \sum_{i=1}^{r} b_i \cdot x_i \in d \cdot G\}$. Let ε' be such that $\frac{1}{2} > \varepsilon' > \varepsilon > 0$ and let A be the set of all $(a_1 \dots a_r)$ in $\mathbb{Z}_d^r/H$ such that

$$\Pr_{\substack{b_1 \dots b_r \\ x \in G}} [f(d \cdot x + \sum_{i=1}^{r} (a_i + b_i) \cdot x_i) = \sum_{i=1}^{r} (a_i + b_i) \cdot y_i] \ge 1 - \varepsilon'\}.$$

We have

$$\begin{aligned} 1 - \varepsilon &= \mathop{\mathrm{E}}_{(a_1 \dots a_r) \in \mathbb{Z}_d^r/H} \left(\Pr_{\substack{(b_1 \dots b_r) \in H \\ x \in G}} [f(d \cdot x + \sum_{i=1}^{r} (a_i + b_i) \cdot x_i) = \sum_{i=1}^{r} (a_i + b_i) \cdot y_i] \right) \\ &\le \frac{\#A}{\#\mathbb{Z}_d^r/H} + (1 - \varepsilon')\big(1 - \frac{\#A}{\#\mathbb{Z}_d^r/H}\big). \end{aligned}$$

From this, we deduce that $\varepsilon' - \varepsilon \le \varepsilon' \cdot \frac{\#A}{\#\mathbb{Z}_d^r/H}$ and thus $A \neq \emptyset$.
Let $(a_1 \dots a_r)$ be in A. We have

$$\mathop{\mathrm{E}}_{x \in G} \left(\Pr_{b \in H} \left[f(d \cdot x + \sum_{i=1}^{r} a_i \cdot x_i) - \sum_{i=1}^{r} a_i \cdot x_i = \sum_{i=1}^{r} b_i \cdot y_i \right] \right) \ge 1 - \varepsilon'.$$

Hence, there exists a $x \in G$ such that $\Pr_{b \in H}[cste = \sum_{i=1} b_i \cdot y_i] \ge 1 - \varepsilon' > \frac{1}{2}$. Therefore, for all $b \in H$ there holds $\sum_{i=1}^{r} b_i \cdot y_i = 0$. Finally, we can define φ such that $\varphi(d \cdot x + \sum_{i=1}^{r} a_i \cdot x_i) = \sum_{i=1}^{r} a_i \cdot y_i$. ◻

Lemma 14. *Assume we are able to compute f s. t. $\Pr_{x \in G}(f(x) \neq \varphi(x)) \le \varepsilon$. Then we can compute a function g such that $\Pr_{x \in G}(g(x) \neq \varphi(x)) \le 12\varepsilon^2$ with at most 6 calls to f.*

Proof. For an $x \in G$, we compute the function g at x as follows:

1. Pick y_1, y_2, $y_3 \in G$.
2. Compute $f(x + y_i)$, $f(y_i)$ for $i = 1, 2, 3$.
3. If $f(x + y_1) - f(y_1) = f(x + y_2) - f(y_2)$, let this be $g(x)$. Otherwise, we set that $g(x) = f(x + y_3) - f(y_3)$.

Set $P_x := \Pr_{y \in G}(f(y) \neq \varphi(y) \text{ or } f(x + y) \neq \varphi(x + y))$. By definition, we have $P_x \le 2\varepsilon$. We obtain $\Pr(g(x) \neq \varphi(x)) \le 2P_x^2(1 - P_x) + P_x^2 \le 12\varepsilon^2$. ◻

Proof (Theorem 8). By iterating n times, we get $\Pr(f(x) \neq \varphi(x)) \le \frac{1}{12} \cdot (12\varepsilon)^{2^n} \le \frac{1}{12} \cdot \xi^{2^n}$. For $n > \log_2(\frac{\log(\#G)}{\log(1/\xi)})$ we have $\Pr_{x \in G}(f(x) \neq \varphi(x)) < \frac{1}{\#G}$. Hence, this probability is equal to zero and the complexity is multiplied by a factor that is in the class $\mathrm{poly}(\log(\#G))$. ◻