Efficient implementation of cryptosystems based on non-maximal imaginary quadratic orders

Detlef Hühnlein

secunet Security Networks AG
Mergenthalerallee 77-81
D-65760 Eschborn, Germany
huehnlein@secunet.de

Abstract. In [14] there is proposed an ElGamal-type cryptosystem based on non-maximal imaginary quadratic orders with trapdoor decryption. The trapdoor information is the factorization of the non-fundamental discriminant $\Delta_p = \Delta_1 p^2$. The NICE-cryptosystem (**N**ew **I**deal **C**oset **E**n-cryption) [24, 12] is an efficient variant thereof, which uses an element $\mathfrak{g}^k \in \mathrm{Ker}(\phi_{Cl}^{-1}) \subseteq Cl(\Delta_p)$, where k is random and $\phi_{Cl}^{-1} : Cl(\Delta_p) \to Cl(\Delta_1)$ is a map between the class groups of the non-maximal and maximal order, to mask the message in the ElGamal cryptosystem. This mask simply "disappears" during decryption, which essentially consists of computing ϕ_{Cl}^{-1}. Thus NICE features quadratic decryption time and hence is very well suited for applications in which a central server has to decrypt a large number of ciphertexts in a short time. In this work we will introduce an *efficient batch decryption* method for NICE, which allows to speed up the decryption by about 30% for a batch size of 100 messages.

In [17] there is proposed a NICE-Schnorr-type signature scheme. In this scheme one uses the group $\mathrm{Ker}(\phi_{Cl}^{-1})$ instead of $\mathbb{F}_p^*$. Thus instead of modular arithmetic one would need to apply standard ideal arithmetic (multiply and reduce) using algorithms from [5] for example. Because every group operation needs the application of the Extended Euclidean Algorithm the implementation would be very inefficient. Especially the signing process, which would typically be performed on a smartcard with limited computational power would be too slow to allow practical application. In this work we will introduce an *entirely new arithmetic* for elements in $\mathrm{Ker}(\phi_{Cl}^{-1})$, which uses the generator and ring-equivalence for exponentiation. Thus the signer essentially performs the exponentiation in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$, which turns out to be about *twenty* times as fast as conventional ideal arithmetic. Furthermore in [17] it is shown, how one can further speed up this exponentiation by application of the Chinese Remainder Theorem for $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$. With this arithmetic the signature generation is about *forty* times as fast as with conventional ideal arithmetic and *more than twice* as fast as in the original Schnorr scheme [26].

1 Introduction

The utilization of imaginary quadratic class groups in cryptography is due to Buchmann and Williams [4], who proposed a key agreement protocol analogue to [7] based on class groups of imaginary quadratic fields, i.e. the class group of the *maximal order*. Since the computation of discrete logarithms in the class group of the imaginary quadratic number field is at least as difficult as factoring the corresponding discriminant (see [4, 27]) these cryptosystems are very interesting from a theoretical point of view. In practice however these cryptosystems seemed to be less efficient than popular cryptosystems based on computing discrete logarithms in $\mathbb{F}_p^*$, like [7, 9] or factoring integers, like [25]. Furthermore the computation of the group order, i.e. the class number, is in general almost as hard as computing discrete logarithms itself by application of the algorithm of Hafner / McCurley [10] or more practical variants like [8, 19], which is subexponential with $L[\frac{1}{2}]$. Hence it seemed to be impossible to set up signature schemes analogue to [9, 22] or [25]. In [14] however it was shown how the application of *non-maximal* imaginary quadratic orders may be used to construct an ElGamal-type cryptosystem with fast decryption and that it is in principle possible to set up ElGamal and RSA-type signature schemes.

In [24] there is proposed an ElGamal-type cryptosystem, later on called NICE for **N**ew **I**deal **C**oset **E**ncryption [12], with very fast decryption. It was shown that the decryption process only needs quadratic time, which makes NICE unique in this sense. First implementations show that the time for decryption is comparable to the time for RSA-*encryption* with $e = 2^{16}$. The central idea of this scheme is to use an element $\mathfrak{g}^k$ of the kernel $\mathrm{Ker}(\phi_{Cl}^{-1})$ of the surjective map $\phi_{Cl}^{-1} : Cl(\Delta_p) \to Cl(\Delta_1)$ to mask the message in the ElGamal-type cryptosystem [14]. The map ϕ_{Cl}^{-1} is induced by the isomorphic map $\varphi^{-1} : \mathcal{I}_{\Delta_p}(p) \to \mathcal{I}_{\Delta_1}(p)$ which maps $\mathcal{O}_{\Delta_p}$-ideals which are prime to the conductor p to $\mathcal{O}_{\Delta_1}$-ideals which are also prime to p. Hence this mask simply "disappears" during the trapdoor-decryption, which just consists of applying ϕ_{Cl}^{-1}, reducing the resulting ideal in the maximal order (and possibly going back to the non-maximal order using φ). The most time consuming step in the decryption is to compute the map ϕ_{Cl}^{-1}, which is essentially the computation of a modular inverse (modulo p) using the Extended Euclidean Algorithm, which needs $O(\log^2(p))$ bit operations.

It is clear that because of this feature NICE is very well suited for applications where a central server has to decrypt a large number of ciphertexts in a short time. Thus it is natural to search for an efficient batch decryption method. In Section 4 we will introduce a simple yet efficient method for batch decryption, which speeds up the system in this scenario even further. The timings in Section 4 show that it is possible to speed up the decryption process for 100 messages by about 30%.

While the main application of the novel arithmetic for $\mathrm{Ker}(\phi_{Cl}^{-1})$ to be introduced in Section 5 might be in the signing procedure of the NICE-Schnorr-type signature scheme [17], its development was actually motivated by cryptosystems based on *totally non-maximal* orders. Due to the very recent result [16] how-

ever, which reduces the DL-problem in these totally non-maximal orders to the DL-problem in finite fields, these cryptosystems seem to have lost much of its attractiveness.

In [15] it was proposed to use totally non-maximal imaginary quadratic orders $\mathcal{O}_{\Delta_{pq}}$, where $\Delta_{pq} = \Delta_1 p^2 q^2$ to set up RSA-type cryptosystems. Because one chooses Δ_1 such that $h(\Delta_1) = 1$ it is easy to compute $h(\Delta_{pq}) = (p-(\Delta_1/p))(q-(\Delta_1/q))$. It is clear that a similar strategy may be used to set up DSA analogues based on totally non-maximal imaginary quadratic orders. First implementations however have shown that these cryptosystems using standard ideal arithmetic are far to inefficient to be used in practice [11]. This lack of efficiency was the motivation for developing a more efficient arithmetic for $Cl(\Delta_p)$, or $\mathrm{Ker}(\phi_{Cl}^{-1})$ which is the same in the case of totally non-maximal orders.

In Section 5 we will introduce this entirely new method for efficient exponentiation of elements in $\mathrm{Ker}(\phi_{Cl}^{-1})$. Instead of using the standard ideal arithmetic (multiplication and reduction of *ideals*) in the non-maximal order we multiply and "reduce" the corresponding *generators* in the maximal order and later on lift the resulting principal ideal, which corresponds to the computed generator, to the non-maximal order. Thus one essentially reduces the arithmetic in $\mathrm{Ker}(\phi_{Cl}^{-1}) \subseteq Cl(\Delta_p)$ to arithmetic in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$ which turns out to be much more efficient.

The timings in Section 5 show that the naive variant of the new exponentiation technique, as proposed here, is already about *twenty* times as fast as classical ideal arithmetic. Very recently it was shown in [17] that one can even do twice as good by utilizing the Chinese Remainder Theorem for $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$. With this improvement the signature generation of the proposed NICE-Schnorr-variant is more than twice as efficient as in the original Schnorr-scheme [26].

This paper is organized as follows: In Section 2 we will provide the necessary basics of imaginary quadratic orders. We will concentrate on the relation between the maximal and non-maximal orders and explain the structure of $\mathrm{Ker}(\phi_{Cl}^{-1})$. In Section 3 we will briefly recall the NICE cryptosystem. In Section 4 we will introduce the new batch decryption for NICE and compare the running times of the implementation. The new exponentiation methods for elements in $\mathrm{Ker}(\phi_{Cl}^{-1})$ are explained in Section 5. We will give the initially proposed method in Section 5.1 and outline the even more efficient CRT - variant from [17] in Section 5.2. In Section 5.3 we will also provide a timing comparison between the new methods, conventional ideal- and modular arithmetic.

2 Imaginary quadratic orders

The basic notions of imaginary quadratic number fields may be found in [1, 13] or [5]. For a more comprehensive treatment of the relationship between maximal and non-maximal orders we refer to [6] or [14].

Let $\Delta \equiv 0,1 \pmod 4$ be a negative integer, which is not a square. The quadratic order of discriminant Δ is defined to be

$$\mathcal{O}_\Delta = \mathbb{Z} + \omega\mathbb{Z},$$

where

$$\omega = \begin{cases} \sqrt{\frac{\Delta}{4}}, & \text{if } \Delta \equiv 0 \pmod 4, \\ \frac{1+\sqrt{\Delta}}{2}, & \text{if } \Delta \equiv 1 \pmod 4. \end{cases} \tag{1}$$

The standard representation of some $\alpha \in \mathcal{O}_\Delta$ is $\alpha = x + y\omega$, where $x, y \in \mathbb{Z}$.

If Δ_1 is squarefree, then $\mathcal{O}_{\Delta_1}$ is the *maximal order* of the quadratic number field $\mathbb{Q}(\sqrt{\Delta_1})$ and Δ_1 is called a fundamental discriminant. The *non-maximal order* of conductor $f > 1$ with (non-fundamental) discriminant $\Delta_f = \Delta_1 f^2$ is denoted by $\mathcal{O}_{\Delta_f}$. In this work we will omit the subscripts to reference arbitrary (fundamental or non-fundamental) discriminants. Because $\mathbb{Q}(\sqrt{\Delta_1}) = \mathbb{Q}(\sqrt{\Delta_f})$ we also omit the subscripts to reference the number field $\mathbb{Q}(\sqrt{\Delta})$. The standard representation of an $\mathcal{O}_\Delta$-ideal is

$$\mathfrak{a} = q\left(\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2a}\mathbb{Z}\right) = (a, b), \tag{2}$$

where $q \in \mathbb{Q}_{>0}, a \in \mathbb{Z}_{>0}$, $c = (b^2 - \Delta)/(4a) \in \mathbb{Z}$, $gcd(a, b, c) = 1$ and $-a < b \leq a$. The norm of this ideal is $\mathcal{N}(\mathfrak{a}) = aq^2$. An ideal is called primitive if $q = 1$. A primitive ideal is called *reduced* if $|b| \leq a \leq c$ and $b \geq 0$, if $a = c$ or $|b| = a$. It can be shown, that the norm of a reduced ideal $\mathfrak{a}$ satisfies $\mathcal{N}(\mathfrak{a}) \leq \sqrt{|\Delta|/3}$ and conversely that if $\mathcal{N}(\mathfrak{a}) \leq \sqrt{|\Delta|/4}$ then the ideal $\mathfrak{a}$ is reduced. We denote the reduction operator in the maximal order by $\rho_1()$ and write $\rho_f()$ for the reduction operator in the non-maximal order of conductor f.

The group of invertible $\mathcal{O}_\Delta$-ideals is denoted by $\mathcal{I}_\Delta$. Two ideals $\mathfrak{a}, \mathfrak{b}$ are equivalent, if there is a $\gamma \in \mathbb{Q}(\sqrt{\Delta})$, such that $\mathfrak{a} = \gamma\mathfrak{b}$. This equivalence relation is denoted by $\mathfrak{a} \sim \mathfrak{b}$. The set of principal $\mathcal{O}_\Delta$-ideals, i.e. which are equivalent to $\mathcal{O}_\Delta$, are denoted by $\mathcal{P}_\Delta$. The factor group $\mathcal{I}_\Delta/\mathcal{P}_\Delta$ is called the *class group* of $\mathcal{O}_\Delta$ denoted by $Cl(\Delta)$. $Cl(\Delta)$ is a finite abelian group with neutral element $\mathcal{O}_\Delta$. Algorithms for the group operation (multiplication and reduction of ideals) can be found in [5]. The order of the class group is called the *class number* of $\mathcal{O}_\Delta$ and is denoted by $h(\Delta)$.

Our cryptosystems make use of the relation between the maximal and non-maximal orders. Any non-maximal order may be represented as $\mathcal{O}_{\Delta_f} = \mathbb{Z} + f\mathcal{O}_{\Delta_1}$. If $h(\Delta) = 1$ then $\mathcal{O}_{\Delta_f}$ is called a *totally non-maximal* imaginary quadratic order of conductor f. An $\mathcal{O}_\Delta$-ideal $\mathfrak{a}$ is called prime to f, if $gcd(\mathcal{N}(\mathfrak{a}), f) = 1$. It is well known, that all $\mathcal{O}_{\Delta_f}$-ideals prime to the conductor are invertible. In every class there is an ideal which is prime to any given number. The algorithm FindIdealPrimeTo in [14] will compute such an ideal. If we denote the (principal) $\mathcal{O}_{\Delta_f}$-ideals, which are prime to f by $\mathcal{P}_{\Delta_f}(f)$ and $\mathcal{I}_{\Delta_f}(f)$ respectively then there

is an isomorphism

$$\mathcal{I}_{\Delta_f}(f)\Big/\mathcal{P}_{\Delta_f}(f) \simeq \mathcal{I}_{\Delta_f}\Big/\mathcal{P}_{\Delta_f} = Cl(\Delta_f). \tag{3}$$

Thus we may 'neglect' the ideals which are not prime to the conductor, if we are only interested in the class group $Cl(\Delta_f)$. There is an isomorphism between the group of $\mathcal{O}_{\Delta_f}$-ideals which are prime to f and the group of $\mathcal{O}_{\Delta_1}$-ideals, which are prime to f, denoted by $\mathcal{I}_{\Delta_1}(f)$ respectively:

Proposition 1. *Let $\mathcal{O}_{\Delta_f}$ be an order of conductor f in an imaginary quadratic field $\mathbb{Q}(\sqrt{\Delta})$ with maximal order $\mathcal{O}_{\Delta_1}$.*

(i.) *If $\mathfrak{A} \in \mathcal{I}_{\Delta_1}(f)$, then $\mathfrak{a} = \mathfrak{A} \cap \mathcal{O}_{\Delta_f} \in \mathcal{I}_{\Delta_f}(f)$ and $\mathcal{N}(\mathfrak{A}) = \mathcal{N}(\mathfrak{a})$.*
(ii.) *If $\mathfrak{a} \in \mathcal{I}_{\Delta_f}(f)$, then $\mathfrak{A} = \mathfrak{a}\mathcal{O}_{\Delta_1} \in \mathcal{I}_{\Delta_1}(f)$ and $\mathcal{N}(\mathfrak{a}) = \mathcal{N}(\mathfrak{A})$.*
(iii.) *The map $\varphi : \mathfrak{A} \mapsto \mathfrak{A} \cap \mathcal{O}_{\Delta_f}$ induces an isomorphism $\mathcal{I}_{\Delta_1}(f) \tilde{\to} \mathcal{I}_{\Delta_f}(f)$. The inverse of this map is $\varphi^{-1} : \mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_1}$.*

Proof: See [6, Proposition 7.20, page 144] . □

Thus we are able to switch to and from the maximal order. The algorithms $\mathsf{GoToMaxOrder}(\mathfrak{a}, f)$ to compute φ^{-1} and $\mathsf{GoToNonMaxOrder}(\mathfrak{A}, f)$ to compute φ respectively may be found in [14].

It is important to note that the isomorphism φ is between the ideal groups $\mathcal{I}_{\Delta_1}(f)$ and $\mathcal{I}_{\Delta_f}(f)$ and *not the class groups.*

If, for $\mathfrak{A}, \mathfrak{B} \in \mathcal{I}_{\Delta_1}(f)$ we have $\mathfrak{A} \sim \mathfrak{B}$, it is not necessarily true that $\varphi(\mathfrak{A}) \sim \varphi(\mathfrak{B})$.

On the other hand, equivalence *does* hold under φ^{-1}. More precisely we have the following:

Proposition 2. *The isomorphism φ^{-1} induces a* surjective homomorphism ϕ_{Cl}^{-1} : $Cl(\Delta_f) \to Cl(\Delta_1)$, *where $\mathfrak{a} \mapsto \rho_1(\varphi^{-1}(\mathfrak{a}))$.*

Proof: This immediately follows from the short exact sequence:

$$Cl(\Delta_f) \longrightarrow Cl(\Delta_1) \longrightarrow 1$$

(see [23, Theorem 12.9, p. 82]). □

In the following we will study the kernel $\mathrm{Ker}(\phi_{Cl}^{-1})$ of the above map ϕ_{Cl}^{-1} and hence the relation between a class in the maximal order and the associated classes in the non-maximal order in more detail. We start with yet another interpretation of the class group $Cl(\Delta_f)$.

Proposition 3. *Let $\mathcal{O}_{\Delta_f}$ be an order of conductor f in a quadratic field. Then there are natural isomorphisms*

$$Cl(\Delta_f) \simeq \mathcal{I}_{\Delta_f}(f)\Big/\mathcal{P}_{\Delta_f}(f) \simeq \mathcal{I}_{\Delta_1}(f)\Big/\mathcal{P}_{\Delta_1,\mathbb{Z}}(f),$$

where $\mathcal{P}_{\Delta_1,\mathbb{Z}}(f)$ denotes the subgroup of $\mathcal{I}_{\Delta_1}(f)$ generated by the principal ideals of the form $\alpha\mathcal{O}_{\Delta_1}$ where $\alpha \in \mathcal{O}_{\Delta_1}$ satisfies $\alpha \equiv a \pmod{f\mathcal{O}_{\Delta_1}}$ for some $a \in \mathbb{Z}$ such that $gcd(a, f) = 1$.

Proof: See [6, Proposition 7.22, page 145]. □

The following corollary is an immediate consequence.

Corollary 4. *With notations as above we have the following isomorphism*

$$\mathrm{Ker}(\phi_{Cl}^{-1}) \simeq {}^{\mathcal{P}_{\Delta_1}(f)}\!/_{\mathcal{P}_{\Delta_1,\mathbb{Z}}(f)}.$$

The next result explains the relation between $\mathrm{Ker}(\phi_{Cl}^{-1})$ and $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$.

Lemma 5. *The map $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^* \to \mathrm{Ker}(\phi_{Cl}^{-1})$, where $\alpha \mapsto \varphi(\alpha\mathcal{O}_{\Delta_1})$ is a surjective homomorphism.*

Proof: This is shown in the more comprehensive proof of Theorem 7.24 in [6] (page 147). □

Another immediate consequence of Proposition 3 allows to decide which principal ideals in the maximal order are mapped to principal ideals in the non-maximal order by applying φ:

Corollary 6. *Let $\alpha \in \mathcal{O}_{\Delta_1}$ be an element of the maximal order and $\mathcal{O}_{\Delta_f}$ be the order of conductor f. Then $\varphi(\alpha\mathcal{O}_{\Delta_1}) \sim \mathcal{O}_{\Delta_f}$ if and only if*

$$\alpha \equiv a \pmod{f\mathcal{O}_{\Delta_1}}$$

with $a \in \mathbb{Z}$ such that $gcd(a, f) = 1$

Thus we are able to "model" the equivalence relation in the non-maximal order by considering generators of principal ideals in the maximal orders. This fact is called *ring-equivalence*.

In Section 5 we will use the above results to formulate concrete algorithms for efficient exponentiation of elements in $\mathrm{Ker}(\phi_{Cl}^{-1})$.

Finally, we will give the exact relationship between the class numbers $h(\Delta_1)$ and $h(\Delta_f)$.

Theorem 7. *Let $\mathcal{O}_{\Delta_f}$ be the order of conductor f in a quadratic field $\mathbb{Q}(\sqrt{\Delta})$ with maximal order $\mathcal{O}_{\Delta_1}$. Then*

$$h(\Delta_f) = \frac{h(\Delta_1)f}{[\mathcal{O}_{\Delta_1}^* : \mathcal{O}_{\Delta_f}^*]} \prod_{p|f} \left(1 - \frac{\left(\frac{\Delta_1}{p}\right)}{p}\right) = nh(\Delta_1),$$

where $n \in \mathbb{N}$ and $\left(\frac{\Delta_1}{p}\right)$ is the Kronecker-symbol.

Proof: See [6, Theorem 7.24, page 146]. □

Because $\mathcal{O}^*_{\Delta_1} = \mathcal{O}^*_{\Delta_p} = \{\pm 1\}$, for $\Delta_p = \Delta_1 p^2$, p prime and $\Delta_1 < -4$ we have an immediate corollary of Theorem 7.

Corollary 8. *Let* $\Delta_1 < -4$, $\Delta_1 \equiv 0,1 \pmod 4$ *and* p *prime. Then* $h(\Delta_p) = h(\Delta_1)\left(p - \left(\frac{\Delta_1}{p}\right)\right)$ *and* $\left|\mathrm{Ker}(\phi_{Cl}^{-1})\right| = \left(p - \left(\frac{\Delta_1}{p}\right)\right)$, *where* $\left(\frac{\Delta_1}{p}\right)$ *is the Kronecker-symbol.*

Thus we are able to control the order of the kernel and consequently set up a Schnorr analogue using the group $\mathrm{Ker}(\phi_{Cl}^{-1})$ instead of $\mathbb{F}_p^*$ as proposed in [17].

3 The NICE cryptosystem

In this section we will briefly recall the setup of NICE. We refer to [24, 12, 18] for a more comprehensive treatment.

Choose two primes p, q, $p > 2\sqrt{q}$ and set $\Delta_1 = -q$ if $q \equiv 3 \pmod 4$, $\Delta_1 = -4q$ otherwise and $\Delta_p = \Delta_1 p^2$. Then $\mathcal{O}_{\Delta_1}$ is a maximal order and $\mathcal{O}_{\Delta_p}$ is a non-maximal order of conductor p. Note that by [14, Lemma 8] all reduced $\mathcal{O}_{\Delta_1}$-ideals are guaranteed to be prime to p, because $p > \sqrt{|\Delta_1|}$. Furthermore choose a reduced $\mathcal{O}_{\Delta_p}$-ideal $\mathfrak{g} \in \mathrm{Ker}(\phi_{Cl}^{-1})$. In [18] there is given a simple algorithm which computes such a kernel element $\mathfrak{g}$.

The *secret* key is just

- the conductor p.

The *public* key consists of

- the non-fundamental discriminant Δ_p and
- the ideal $\mathfrak{g}$.

Because the system is entirely broken if one is able to factor Δ_p one should, as explained in [14], at least choose $p, q > 2^{200}$.

To *encrypt* a message $1 \leq m < \sqrt{|\Delta_1|/4}$ one proceeds as follows:

1. Choose a random $k \in \mathbb{Z}$ with $1 < k < 2^{80}$.
2. Compute the reduced $\mathcal{O}_{\Delta_p}$-ideal $\mathfrak{k} = \rho_p(\mathfrak{g}^k)$.
3. Embed the message $m \in \mathbb{Z}$ in a $\mathcal{O}_{\Delta_p}$-Ideal $\mathfrak{m}$ with $\mathcal{N}(\mathfrak{m}) < \sqrt{|\Delta_1|/4}$.
4. Compute the ciphertext $\mathfrak{c} = \rho_p(\mathfrak{m}\mathfrak{k})$.

For the message embedding one may use the algorithm given in [18]. It is clear that the ideal $\mathfrak{k}$ is simply used to "mask" the message in the ElGamal-type scheme. Furthermore note that $k < 2^{80}$ can be chosen to be "unusually small", because in contrast to the classical ElGamal cryptosystem the ciphertext consists

of just *one* element and hence one would have to apply a brute force strategy to determine the message. It is just not possible to compute some discrete logarithm using more sophisticated e.g. (baby-step-giant-step) techniques if one is only given the cipher text. We refer to [18] for a detailed treatment of this issue.

To *decrypt* the ciphertext $\mathfrak{c}$ one proceeds as follows:

1. Compute $\mathfrak{C} = \varphi^{-1}(\mathfrak{c})$ using algorithm GotoMaxorder$(\mathfrak{c}, p)$ from [14].
2. Reduce $\mathfrak{C}$, i.e. compute $\mathfrak{M} = \rho_1(\mathfrak{C})$.
3. Compute $\mathfrak{m} = \varphi(\mathfrak{M})$ using algorithm GotoNonMaxorder$(\mathfrak{M}, p)$ from [14].

Note that the computation in Step 1.-2. is just the computation of ϕ_{Cl}^{-1}.

The correctness of the decryption procedure is easy to see. Because $\mathfrak{g} \in \mathrm{Ker}(\phi_{Cl}^{-1})$ we have $\varphi^{-1}(\mathfrak{c}) = \varphi^{-1}(\mathfrak{m}\mathfrak{k}) = \varphi^{-1}(\mathfrak{m})(\alpha)\mathcal{O}_{\Delta_1} = \mathfrak{M}(\alpha)\mathcal{O}_{\Delta_1} \sim \mathfrak{M}$, where $\alpha \in \mathcal{O}_{\Delta_1}$.

Because $\mathcal{N}(\mathfrak{m}) < \sqrt{|\Delta_1|/4}$ we know that $\mathfrak{m} = \varphi(\mathfrak{M}) = \varphi(\rho_1(\mathfrak{C}))$ is a reduced $\mathcal{O}_{\Delta_p}$-ideal - the message-ideal $\mathfrak{m}$.

Note that if the message is embedded in the norm of the ideal $\mathfrak{m}$ only, as proposed in [18], then the step back to the non-maximal order (Step 3.) may be omitted, because we have $\mathcal{N}(\mathfrak{m}) = \mathcal{N}(\mathfrak{M})$.

For the readers convenience we will recall the algorithm GotoMaxOrder from [14]:

Algorithm 9. *(GoToMaxOrder)*

Input: *A primitive $\mathcal{O}_{\Delta_p}$-ideal $\mathfrak{a} = (a, b)$, the fundamental discriminant Δ_1 and the conductor p*

Output: *A primitive $\mathcal{O}_{\Delta_1}$-ideal $\mathfrak{A} = \varphi^{-1}(\mathfrak{a}) = \mathfrak{a}\mathcal{O}_{\Delta_1}$*

1. $b_{\mathcal{O}} \leftarrow \Delta_1 \pmod 2$
2. *Solve* $1 = \mu p + \lambda a$ *for* $\mu, \lambda \in \mathbb{Z}$
3. $B \leftarrow b\mu + a b_{\mathcal{O}} \lambda \pmod{2a}$
4. *RETURN* (a, B)

4 Efficient batch decryption for NICE

It is clear that because of its very fast decryption NICE is very well suited for applications in which a central server has to decrypt a large number of ciphertexts in a short time. Thus it is desireable to have an efficient batch decryption procedure at hand. In the following we will introduce a simple method which decrypts n ciphertexts $\mathfrak{c}_i$, $1 \leq i \leq n$ in one step, which turns out to be much faster than the sequential processing.

If we have a closer look at the decryption procedure above we recognize that the most time consuming operation is the computation of GotoMaxOrder. This

step is essentially the computation of a modular inverse modulo the conductor. Thus we can speed up the decryption process by applying a batch-gcd-strategy, like proposed in [21][1]. The central idea is to replace all but one costly inversions with the Extended Euclidean Algorithm by a few modular multiplications.

If one is asked to compute $b_1 \equiv a_1^{-1} \pmod{p}$ and $b_2 \equiv a_2^{-1} \pmod{p}$. Then instead of performing two inversions one can compute $a \equiv a_1 a_2 \pmod{p}$, $b \equiv a^{-1} \pmod{p}$, $b_1 \equiv b a_2 \pmod{p}$ and $b_2 \equiv b a_1 \pmod{p}$. Thus one replaces one inversion by three modular multiplications, which are usually faster, because in most implementations one inversion is "about" 15 modular multiplications.

It is an easy matter to generalize this strategy to n inversions. This immediately leads to the following algorithm for batch decryption, where we assume that the message is entirely encoded in the norm of the message-ideal, like proposed in [18].

Algorithm 10. *(NICE-Batch-Decryption)*

Input: *n ciphertexts, i.e. reduced $\mathcal{O}_{\Delta_p}$-ideals $\mathfrak{c}_i = (a_i, b_i)$, $1 \leq i \leq n$, the fundamental discriminant Δ_1 and the conductor p.*

Output: *The n corresponding plaintexts, i.e. the norms A_i of the corresponding ideals $\mathfrak{M}_i = (A_i, B_i)$, for $1 \leq i \leq n$.*

1. $b_{\mathcal{O}} \leftarrow \Delta_1 \pmod{2}$
2. $g_0 \leftarrow 1$
3. $g_1 \leftarrow a_1$
4. *FOR* i *FROM* 2 *TO* n *DO* $g_i \leftarrow g_{i-1} a_i \pmod{p}$
5. *Compute* $h_n \leftarrow g_n^{-1} \pmod{p}$
6. *FOR* i *FROM* n *TO* 1 *DO*
   - 6.1 $\lambda_i \leftarrow h_i g_{i-1} \pmod{p}$
   - 6.2 $h_{i-1} \leftarrow h_i a_i \pmod{p}$
   - 6.3 $\mu_i \leftarrow \frac{1-\lambda_i a_i}{p}$
   - 6.4 $B_i \leftarrow b_i \mu_i + a_i b_{\mathcal{O}} \lambda_i \pmod{2a_i}$
   - 6.5 $\mathfrak{M}_i = (A_i, B_i) \leftarrow \rho_1(a_i, B_i)$
7. *RETURN n plaintexts A_i, $1 \leq i \leq n$*

Thus instead of n inversions with the Extended Euclidean Algorithm we only have to perform one inversion, $3n-3$ modular multiplications, n integer multiplications and n integer divisions. Thus in typical implementations we are able to reduce the time for n decryptions, as shown in Table 1 below.

The implementation was done using the LiDIA-package [20] on a Pentium 133 MHz choosing random primes p, q of the respective bit-length. The timings are given in microseconds, averaged over a number of 100 randomly chosen messages. The first row shows how many modular multiplications are as costly as one inversion in LiDIA. The next rows give the time for a NICE-encryption using

[1] The author would like to thank V. Müller for pointing out the reference.

80bit exponents and the binary, usual BGMW-, and the signed BGMW-method [2] for exponentiation. This includes the time for the message-embedding. The last four rows give the decryption time (per message) for batch sizes of 1, 5, 10 and 100 messages respectively. This shows that for a batch size of 100 we are able to speed up the decryption by about 30%.

bitlength p, q	200		300		400		500	
mult / inv	13.9		15.4		16.2		15.6	
	ms	%	ms	%	ms	%	ms	%
NICE Enc. (binary)	1861.7	100	4065.2	100	7368.9	100	12182.1	100
NICE Enc. (BGMW)	669.7	35.97	1786.6	43.95	3556.5	48.26	6461.9	53.04
NICE Enc. (±-BGMW)	640.9	34.43	1732.6	42.62	3493.6	47.41	6315.5	51.84
NICE Dec. (1 mess.)	9.50	100	16.75	100	26.30	100	35.66	100
NICE Dec. (5 mess.)	8.20	86.32	13.16	78.57	20.00	76.05	26.93	75.52
NICE Dec. (10 mess.)	7.45	78.42	12.34	73.67	19.11	72.66	25.61	71.82
NICE Dec. (100 mess.)	6.70	70.53	11.64	69.49	18.30	69.58	24.61	69.01

Table 1. Timings for NICE with sequential and batch decryption

5 Efficient exponentiation for elements of $\mathrm{Ker}(\phi_{Cl}^{-1})$

In this section we will introduce a novel arithmetic for classes in $\mathrm{Ker}(\phi_{Cl}^{-1})$ which turns out to be much more efficient than standard ideal arithmetic.

Since we need to apply φ during our computation we will only consider ideals $\mathfrak{a}$ which are prime to the conductor f. Thus if we are considering principal (integral) ideals $\alpha\mathcal{O}_{\Delta_1}$, for some $\alpha \in \mathcal{O}_{\Delta_1}$, then we require $\gcd(N(\alpha), f) = 1$.

We start with providing the details of a naive generator arithmetic in Section 5.1. While an exponentiation of an ideal using this arithmetic turns out to be about *twenty* times (for the Schnorr-scheme and thirteen times for the DSA-scheme in totally non-maximal orders) as fast as conventional ideal arithmetic, we can do even twice as good by applying CRT in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ as proposed in [17]. For the readers convenience this method is briefly outlined in Section 5.2. With this arithmetic the signature generation in the Schnorr-analogue [17] is more than twice as fast as in the original scheme.

5.1 Arithmetic in $\mathrm{Ker}(\phi_{Cl}^{-1})$ using $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$

While we already know from Lemma 5 that the arithmetic in $\mathrm{Ker}(\phi_{Cl}^{-1})$ can be reduced to the arithmetic in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$, we will give a very elementary proof here, which ends up in a "ready to implement" algorithm.

It is clear that all integral ideals $\mathfrak{a} \in \mathrm{Ker}(\phi_{Cl}^{-1}) \subseteq Cl(\Delta_f)$ are of the form

$$\mathfrak{a} = \varphi(\alpha \mathcal{O}_{\Delta_1}), \tag{4}$$

for some $\alpha \in \mathcal{O}_{\Delta_1}$.

Now instead of multiplying and reducing the ideals in the non-maximal order we will work with the generators which are corresponding to principal ideals in the maximal order.

We will start with a simple lemma, which can easily be verified by straightforward calculation.

Lemma 11. *Let $\alpha_i = x_i + y_i\omega \in \mathcal{O}_{\Delta_1}$, $x_i, y_i \in \mathbb{Z}, i \in \{1,2\}$ and ω like given in (1). Then $\beta = x + y\omega = \alpha_1\alpha_2$ is given by*

$$x = x_1x_2 + y_1y_2\frac{\Delta_1}{4} \tag{5}$$

$$y = x_1y_2 + x_2y_1 \tag{6}$$

in the case that $\Delta_1 \equiv 0 \pmod 4$ and

$$x = x_1x_2 + y_1y_2\frac{\Delta_1 - 1}{4} \tag{7}$$

$$y = x_1y_2 + x_2y_1 + y_1y_2 \tag{8}$$

if $\Delta_1 \equiv 1 \pmod 4$.

Thus multiplying two generators α_i is more efficient than multiplying the two ideals $\alpha_i\mathcal{O}_{\Delta_1}$, because no application of the costly Extended Euclidean Algorithm is necessary.

It is clear however that we "somehow need to reduce" intermediate results during exponentiation to obtain a polynomial time algorithm. The central idea is to "model" reduction of ideals (in the non-maximal order) by manipulating the generator. This task will turn out to be surprisingly simple.

The following lemma is immediate.

Lemma 12. *Let $\alpha = x + y\omega, \alpha' = x' + y'\omega \in \mathcal{O}_{\Delta_1}$ and $f \in \mathbb{Z}_{>1}$. Then $\alpha \equiv \alpha' \pmod{f\mathcal{O}_{\Delta_1}}$ if and only if $x' \equiv x \pmod f$ and $y' \equiv y \pmod f$.*

Next we will consider the norm of an element $\alpha \in \mathcal{O}_{\Delta_1}$ under this congruence.

Lemma 13. *Let $\alpha, \beta \in \mathcal{O}_{\Delta_1}$ and $f \in \mathbb{Z}_{>1}$. If $\alpha \equiv \beta \pmod{f\mathcal{O}_{\Delta_1}}$ then $\mathcal{N}(\alpha) \equiv \mathcal{N}(\beta) \pmod f$.*

Proof: Let $\alpha = x + y\omega$. Then by Lemma 12 above we have $\beta = x' + y'\omega$, where $x' \equiv x \pmod f$ and $y' \equiv y \pmod f$.

Then we have

$$\begin{aligned}\mathcal{N}(\alpha) &= x^2 - y^2\omega^2 \\ &\equiv x'^2 - y'^2\omega^2 \pmod{f} \\ &= \mathcal{N}(\beta).\end{aligned}$$

□

The following corollary is immediate.

Corollary 14. *Let* $\alpha, \beta \in \mathcal{O}_{\Delta_1}$, $f \in \mathbb{Z}_{>1}$ *and* $\alpha \equiv \beta \pmod{f\mathcal{O}_{\Delta_1}}$. $\gcd(\mathcal{N}(\alpha), f) = 1$ *if and only if* $\gcd(\mathcal{N}(\beta), f) = 1$.

Lemma 15. *Let* $\alpha, \beta \in \mathcal{O}_{\Delta_1}$ *such that* $\gcd(\mathcal{N}(\alpha), f) = \gcd(\mathcal{N}(\beta), f) = 1$ *and* φ *as defined in Proposition 1. Furthermore let* $\gamma \equiv \alpha\beta \pmod{f\mathcal{O}_{\Delta_1}}$. *Then* $\gcd(\mathcal{N}(\gamma), f) = 1$ *and if* $\alpha \equiv \beta \pmod{f\mathcal{O}_{\Delta_1}}$ *then* $\varphi(\alpha\mathcal{O}_{\Delta_1}) \sim \varphi(\beta\mathcal{O}_{\Delta_1})$ *in* $Cl(\Delta_f)$.

Proof: That $\gcd(\mathcal{N}(\gamma), f) = 1$ is immediate by the multiplicativity of the norm and Corollary 14.

Because $\alpha \equiv \beta \pmod{f\mathcal{O}_{\Delta_1}}$ it follows, that $\alpha = \beta\delta$ for some $\delta \in \mathbb{Q}(\sqrt{\Delta})$, where $\delta \equiv 1 \pmod{f\mathcal{O}_{\Delta_1}}$. Thus by Proposition 6 we know that $\varphi(\delta\mathcal{O}_{\Delta_1}) \sim \mathcal{O}_{\Delta_f}$ and hence the assertion follows. □

Furthermore we need the following result, which is immediate because φ is an isomorphism.

Lemma 16. *Let* $\alpha \in \mathcal{O}_{\Delta_1}$, *such that* $\gcd(\mathcal{N}(\alpha), f) = 1$, $n \in \mathbb{Z}$ *and* φ *as defined in Proposition 1. Then we have* $\varphi(\alpha\mathcal{O}_{\Delta_1})^n = \varphi(\alpha^n\mathcal{O}_{\Delta_1})$.

By combining the above results we immediately obtain the following.

Lemma 17. *Let* $\alpha \in \mathcal{O}_{\Delta_1}$, *such that* $\gcd(\mathcal{N}(\alpha), f) = 1$, $n \in \mathbb{Z}$ *and* φ *as defined in Proposition 1. Then we have* $\varphi(\alpha\mathcal{O}_{\Delta_1})^n \sim \varphi(\gamma\mathcal{O}_{\Delta_1})$ *for some* $\gamma \equiv \alpha^n \pmod{f\mathcal{O}_{\Delta_1}}$.

The following lemma follows immediately from (5)-(8) and Lemma 12.

Lemma 18. *Let* $\alpha_i = x_i + y_i\omega \in \mathcal{O}_{\Delta_1}$, $x_i, y_i \in \mathbb{Z}, i \in \{1, 2\}$, ω *like given in (1) and* $f \geq 1$. *Then* $\beta = x + y\omega \equiv \alpha_1\alpha_2 \pmod{f\mathcal{O}_{\Delta_1}}$ *is given by*

$$x \equiv x_1x_2 + y_1y_2\frac{\Delta_1}{4} \pmod{f} \tag{9}$$

$$y \equiv x_1y_2 + x_2y_1 \pmod{f} \tag{10}$$

in the case that $\Delta_1 \equiv 0 \pmod 4$ *and*

$$x \equiv x_1x_2 + y_1y_2\frac{\Delta_1 - 1}{4} \pmod{f} \tag{11}$$

$$y \equiv x_1y_2 + x_2y_1 + y_1y_2 \pmod{f} \tag{12}$$

if $\Delta_1 \equiv 1 \pmod 4$.

This result enables us to "model" the conventional ideal arithmetic (multiplication and reduction) by simple calculations modulo f. This leads to the following algorithm for exponentiation, which is based on binary method for exponentiation. We denote the binary length of n by $\lambda(n) = \lfloor \log_2(n) \rfloor + 1$.

Algorithm 19. *(Gen-Exp)*

Input: $\alpha = x + y\omega \in \mathcal{O}_{\Delta_1}$, *the conductor* f *such that* $\gcd(\mathcal{N}(\alpha), f) = 1$ *and the exponent* $n \in \mathbb{Z}$.

Output: $\mathfrak{a} = (a, b) = \rho_f(\varphi((\alpha\mathcal{O}_{\Delta_1})^n))$.

1. *IF* $n = 0$ *THEN OUTPUT*$(1, \Delta_1 \pmod 2)$
2. *IF* $n < 0$ *THEN* $n \leftarrow -n$, $y \leftarrow -y$
3. $l \leftarrow \lambda(n) - 1$, $(n_l \ldots n_0)_2 \leftarrow$ *binary expansion of* n, *i.e.* $n_l = 1$
4. $x_h \leftarrow x \pmod f$
5. $y_h \leftarrow y \pmod f$
6. *IF* $\Delta_1 \equiv 0 \pmod 4$ *THEN* $D \leftarrow \Delta_1/4$ *ELSE* $D \leftarrow (\Delta_1 - 1)/4$
7. *FOR* $i = l - 1$ *DOWNTO* 0 *DO*
   - 7.1 $h \leftarrow x_h$
   - 7.2 $x_h \leftarrow h^2 + y_h^2 D \pmod f$
   - 7.3 *IF* $\Delta_1 \equiv 0 \pmod 4$ *THEN* $y_h \leftarrow 2hy_h \pmod f$ *ELSE* $y_h \leftarrow 2hy_h + y_h^2 \pmod f$
   - 7.4 *IF* $n_i = 1$ *THEN*
     - 7.4.1 $h \leftarrow x_h$
     - 7.4.2 $x_h \leftarrow hx + y_h y D \pmod f$
     - 7.4.3 *IF* $\Delta_1 \equiv 0 \pmod 4$ *THEN* $y_h \leftarrow hy + xy_h \pmod f$ *ELSE* $y_h \leftarrow hy + xy_h + y_h y \pmod f$
8. */* Compute the standard representation* $\mathfrak{A} = d(a, b) = \alpha_h \mathcal{O}_{\Delta_1}$ **/*
   - 8.1 */* Use* $\frac{x+y\sqrt{\Delta_1}}{2}$*-form */* $x_h \leftarrow 2x_h$ *IF* $\Delta_1 \equiv 1 \pmod 4$ *THEN* $x_h \leftarrow x_h + y_h$
   - 8.2 *Compute* $d \leftarrow \gcd(y_h, (x_h + y_h\Delta_1)/2) = \lambda y_h + \mu(x_h + y_h\Delta_1)/2$, *for* $\lambda, \mu \in \mathbb{Z}$
   - 8.3 $A \leftarrow |x_h^2 - \Delta_1 y_h^2|/(4d^2)$
   - 8.4 $B \leftarrow (\lambda x_h + \mu(x_h + y_h)\Delta_1/2)/d \pmod{2A}$
9. */* Lift* $\mathfrak{A}' = (1/d)\mathfrak{A}$ *to the non-maximal order and reduce it */* $b \leftarrow Bf \pmod{2A}$ $(a, b) \leftarrow \rho_f(A, b)$
10. *OUTPUT*(a, b)

Proof: By Lemma 17 we only have to compute $\gamma \equiv \alpha^n \pmod{f\mathcal{O}_{\Delta_1}}$. The correctness of the exponentiation algorithm is immediate because it is the well known binary method with the operation given in Lemma 18 as group operation.

In step 8 we simply compute the standard representation of the ideal $\mathfrak{A} = \alpha_h\mathcal{O}_{\Delta_1} = d(a\mathbb{Z} + (b + \sqrt{\Delta_1})/2\mathbb{Z})$. By Corollary 14 we know that $\mathcal{N}(\alpha_h\mathcal{O}_{\Delta_1}) = ad^2$ is prime to f. This clearly implies that $\gcd(d, f) = 1$. Because $\mathfrak{A} = (d)\mathfrak{A}'$

for $d \in \mathbb{Z}$, $\mathfrak{A}' = a\mathbb{Z} + (b + \sqrt{\Delta_1})/2\mathbb{Z}$ we know from Proposition 6 that $\varphi(\mathfrak{A}) \sim \varphi(\mathfrak{A}')$. Finally it is clear that we can apply φ from Proposition 1, because $\gcd(a, f) = 1$ □

5.2 Even more efficient arithmetic in $\mathrm{Ker}(\phi_{Cl}^{-1})$ using CRT in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$

In the previous section we saw that the arithmetic in $\mathrm{Ker}(\phi_{Cl}^{-1})$ can be reduced to arithmetic in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$, which turns out to be much more efficient. In this section we *outline* yet another method for a further speed up. We refer to [17,16] for the details.

We will only concentrate on a special case which seems to be most important for practical application, as it is used in the Schnorr-analogue from [17]. That is we assume that the conductor is a prime p, where $\left(\frac{\Delta_1}{p}\right) = 1$.

Lemma 20. *Let $\mathcal{O}_{\Delta_1}$ be the maximal order and p be prime. Then there is an isomorphism*

$$(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^* \simeq \mathbb{F}_p[X]\Big/(f(X)),$$

where $(f(X))$ is the ideal generated by $f(X) \in \mathbb{F}_p[X]$ and

$$f(X) = \begin{cases} X^2 - \frac{\Delta_1}{4}, & \text{if } \Delta_1 \equiv 0 \pmod 4, \\ X^2 - X + \frac{1-\Delta_1}{4}, & \text{if } \Delta_1 \equiv 1 \pmod 4. \end{cases} \tag{13}$$

Proof: See [16, Proposition 5]. □

Theorem 21. *Assume that $\left(\frac{\Delta_1}{p}\right) = 1$ and the roots $\rho, \bar{\rho} \in \bar{\mathbb{F}}_p$ of $f(X) \in \mathbb{F}_p[X]$ as given in* (13) are known. Then the following isomorphism can be computed in time $O((\log p)^2)$:

$$(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^* \simeq \mathbb{F}_p^* \otimes \mathbb{F}_p^*$$

Proof: From Lemma 20 we know that there is an isomorphic map $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^* \to \mathbb{F}_p[X]/(f(X))$, where $f(X) \in \mathbb{F}_p[X]$ is given in (13). And that this isomorphism is trivial to compute.

Because $\left(\frac{\Delta_1}{p}\right) = 1$ the polynomial $f(X)$ is not irreducible, but can be decomposed as $f(X) = (X - \rho)(X - \bar{\rho}) \in \mathbb{F}_p[X]$ where $\rho, \bar{\rho} \in \mathbb{F}_p$ are the roots of $f(X)$. Thus if $\Delta_1 \equiv 0 \pmod 4$ and $D = \Delta_1/4$ we have $\rho \in \mathbb{F}_p$ such that $\rho^2 \equiv D \pmod p$ and $\bar{\rho} = -\rho$. In the other case $\Delta_1 \equiv 1 \pmod 4$ we have $\rho = (1 + b)/2$, where $b^2 \equiv \Delta_1 \pmod p$ and $\bar{\rho} = (1 - b)/2 \in \mathbb{F}_p$. Thus we have the isomorphisms

$$(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^* \simeq \left(\mathbb{F}_p[X]\Big/(X - \rho)\right)^* \otimes \left(\mathbb{F}_p[X]\Big/(X - \bar{\rho})\right)^* \simeq \mathbb{F}_p^* \otimes \mathbb{F}_p^*.$$

Let $\alpha = a + b\omega \in (\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$ then the mapping $\psi : (\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^* \to \mathbb{F}_p^* \otimes \mathbb{F}_p^*$ is given as $x_1 = \psi_1(\alpha) = a + b\rho \in \mathbb{F}_p^*$ and $x_2 = \psi_2(\alpha) = a + b\bar{\rho} \in \mathbb{F}_p^*$. The inverse map ψ^{-1} is computed by solving the small system of linear equations. I.e. one will recover $a, b \in \mathbb{F}_p^*$ by computing $b = \frac{x_2 - x_1}{\bar{\rho} - \rho}$ and $a = x_1 - b\rho$. Thus both transformations ψ and ψ^{-1} need time $O((\log p)^2)$. □

With this result we immediately obtain the of the following algorithm.

Algorithm 22. *(Gen-CRT)*

Input: $\alpha = x + y\omega \in \mathcal{O}_{\Delta_1}$, *the conductor* p, *such that* $\gcd(\mathcal{N}(\alpha), p) = 1$, $\left(\frac{\Delta_1}{p}\right) = 1$, *the roots* $\rho, \bar{\rho} \in \mathbb{F}_p^*$ *of* $f(X)$ *as given in (13) and the exponent* $n \in \mathbb{Z}$.

Output: $\mathfrak{a} = (a, b) = \rho_p(\varphi((\alpha\mathcal{O}_{\Delta_1})^n))$.

1. *IF* $n = 0$ *THEN OUTPUT*$(1, \Delta_1 \pmod 2)$
2. *IF* $n < 0$ *THEN* $n \leftarrow -n$, $y \leftarrow -y$
3. $x_1 \leftarrow (x + \rho y)^n \pmod p$
4. $x_2 \leftarrow (x + \bar{\rho} y)^n \pmod p$
5. $r \leftarrow (\bar{\rho} - \rho)^{-1} \pmod p$
6. $y_h \leftarrow (x_2 - x_1)r \pmod p$
7. $x_h \leftarrow x_1 - y_h\rho \pmod p$
8. *Compute standard representation, lift and reduce as in Algorithm 19 Step 8.-9.*
9. *OUTPUT*(a, b)

Note that the computation of r in Step 5 can be done in a precomputation phase, as is it independent of the current α.

5.3 Timings for different arithmetics

In this section we will give the timinings of a first implementation of the novel arithmetics for $\mathrm{Ker}(\phi_{Cl}^{-1})$. We will also include timings for standard-ideal arithmetic and modular arithmetic to allow comparison.

For the RSA analogues in totally non-maximal orders [15] we fixed $\Delta_1 = -163$ and chose a random exponent $k < n = pq$. For all DL-based systems (DSA and Schnorr) we chose a random $k < 2^{160}$. For the DSA-analogue based on *totally non-maximal* orders we also fixed $\Delta_1 = -163$. Note that due to the recent result [16] this analogue with Δ_p is only as secure as the original scheme with p. Thus one needs to compare the lines for the 1200 bit DSA-analogue in $Cl(\Delta_p)$ with the time for 600 bit modular arithmetic.

For the NICE-Schnorr-analogue [17] we also chose a random $k < 2^{160}$ and $\Delta_p = \Delta_1 p^2$ where $\Delta_1 = -q$ (or $\Delta_1 = -4q$ if $q \equiv 1 \pmod 4$ respectively) and p, q with equal bitlength. Because factoring integers is about as hard as the computation of discrete logarithms (modulo p) one needs to compare the timings where Δ_p and the prime modulus have the same bitlength.

The timings are given in microseconds on a pentium 133 MHz using the LiDIA - package [20]. One should note that the implementation of neither variant is optimized. This is no problem, because we are interested in the comparison, rather than the absolute timings.

cryptosystem	Schnorr / DSA					RSA		
arithmetic	mod.	ideal	Gen-exp	Gen-exp	Gen-CRT	mod.	ideal	Gen-exp
bitlength of	p	Δ_p	$\Delta_p = -163p^2$	$\Delta_p = -qp^2$	$\Delta_p = -qp^2$	$n = pq$	$n = pq$	$n = pq$
600	188	3182	240	159	83	258	10490	994
800	302	4978	368	234	123	583	22381	2053
1000	447	7349	542	340	183	886	35231	3110
1200	644	9984	724	465	249	1771	68150	6087
1600	1063	15751	1156	748	409	3146	125330	10864
2000	1454	22868	1694	1018	563	5284	224799	18067

Table 2. Timings for exponentiation with different arithmetics

The timings in Table 2 show the impressive improvement. One can see that the exponentiation using Algorithm 19 is already about *thirteen* times as fast as an exponentiation using conventional ideal arithmetic, if $\Delta_p = -163p^2$ and *more than twenty* times as fast for the Schnorr-analogue.

If we apply Algorithm 22 as proposed in [17] and outlined in Section 5.2, we are about *forty times* as fast as conventional ideal arithmetic. Using this arithmetic the signature generation in the NICE-Schnorr-analogue is *more than twice* as fast as in the original scheme in $\mathbb{F}_p^*$.

On the other side we see that the RSA-analogue [15] in totally non-maximal orders is still far less efficient than the original scheme and although immune against low exponent and chosen ciphertext attack not preferable for practice.

Finally one should note that for the signature verification in the NICE-Schnorr-scheme one has to use standard ideal arithmetic, which is very inefficient. Thus an important task for the future will be to speed up the standard-ideal arithmetic as well, to enable practical application of the proposed Schnorr-analogue [17].

References

1. Z.I. Borevich and I.R. Shafarevich: *Number Theory* Academic Press: New York, 1966
2. E. Brickell, D. Gordon, K. McCurley, D. Wilson: *Fast Exponentiation with Precomputation*, Proceedings of Eurocrypt 1992, LNCS **658**, Springer, 1993, pp. 200-207

3. J. Buchmann, S. Düllmann: *On the computation of discrete logarithms in class groups*, Advances in Cryptology - CRYPTO '90, LNCS **773**, Springer, 1991, pp. 134-139
4. J. Buchmann and H.C. Williams: *A key-exchange system based on imagninary quadratic fields*. Journal of Cryptology, **1**, 1988, pp. 107-118
5. H. Cohen: *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics **138**. Springer: Berlin, 1993.
6. D.A. Cox: *Primes of the form $x^2 + ny^2$*, John Wiley & Sons, New York, 1989
7. W. Diffie and M. Hellman: *New directions in cryptography*, IEEE Transactions on Information Theory **22**, 1976, pp. 472-492
8. S. Düllmann: *Ein Algorithmus zur Bestimmung der Klassenzahl positiv definiter binärer quadratischer Formen*, PHD-thesis (in german), University of Saarbrücken: 1991
9. T. ElGamal: *A public key cryptosystem and a signature schem based on discrete logarithms*, IEEE Transactions on Information Theory **31**, 1985, pp. 469-472
10. J.L. Hafner, K.S. McCurley: *A rigorous subexponential algorithm for computation of class groups*, Journal of the American Mathematical Society, **2**, 1989, 837-850
11. S. Hamdy, A. Meyer: *personal communication*, 1999
12. M. Hartmann, S. Paulus and T. Takagi: *NICE - New Ideal Coset Encryption*, to appear in the proceedings of CHES, 1999
13. L.K. Hua: *Introduction to Number Theory*. Springer-Verlag, New York, 1982.
14. D. Hühnlein, M.J. Jacobson, S. Paulus and T. Takagi: *A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption*, Advances in Cryptology - EUROCRYPT '98, LNCS **1403**, Springer, 1998, pp. 294-307
15. D. Hühnlein, A. Meyer, T. Takagi: *Rabin and RSA analogues based on non-maximal imaginary quadratic orders*, Proceedings of ICICS '98, ISBN 89-85305-14-X, 1998, pp. 221-240
16. D. Hühnlein, T. Takagi: *Reducing logarithms in totally non-maximal imaginary quadratic orders to logarithms in finite fields*, submitted to Asiacrypt'99, 1999, preprint via http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/Welcome.html
17. D. Hühnlein, J. Merkle: *An efficient NICE-Schnorr-type signature scheme*, forthcoming
18. D. Hühnlein: *NICE - Ein neues Public Key Kryptosystem mit sehr schneller Entschlüsselung und seine potentiellen Anwendungen*, (in german) manuscript, 1999
19. M.J. Jacobson Jr.: *Subexponential Class Group Computation in Quadratic Orders*, PhD thesis, TU Darmstadt, to apear, 1999
20. LiDIA: *A c++ library for algorithmic number theory*, via http://www.informatik.tu-darmstadt.de/TI/LiDIA
21. P.L. Montgomery: *Speeding the Pollard and Elliptic Curve Methods for Factorization*, Mathematics of Computation, vol. **48**, nr. **177**, Jan. 1987, pp. 243-264
22. National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186, **FIPS-186**, 19th May, 1994
23. J. Neukirch, *Algebraische Zahlentheorie*, Springer, Berlin, 1992
24. S. Paulus, T. Takagi: *A new public-key cryptosystem over the quadratic order with quadratic decryption time*, to appear in Journal of Cryptology, 1998, preprint via http://www.informatik.tu-darmstadt.de/TI/Mitarbeiter/sachar.html
25. R. Rivest, A. Shamir, L. Adleman: *A method for obtaining digital signatures and public key-cryptosystems*, Communications of the ACM,**21**, 1978, pp. 120-126

26. C.P. Schnorr: *Efficient identification and signatures for smart cards*, Advances in Cryptology - CRYPTO '89, LNCS **435**, Springer, 1990, pp. 239-252
27. R.J. Schoof: *Quadratic Fields and Factorization*. In: H.W. Lenstra, R. Tijdeman, (eds.): *Computational Methods in Number Theory*. Math. Centrum Tracts **155**. Part II. Amsterdam, 1983. pp. 235-286.