Towards practical non-interactive public key cryptosystems using non-maximal imaginary quadratic orders

(Extended Abstract)

Detlef Hühnlein[1], Michael J. Jacobson, Jr.[2] and Damian Weber[3]

[1] secunet Security Networks AG
Mergenthalerallee 77-81, D-65760 Eschborn, Germany
`huehnlein@secunet.de`
[2] Centre for Applied Cryptographic Research
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
`mjjacobs@cacr.math.uwaterloo.ca`
[3] Märkische Fachhochschule Iserlohn und Hagen
Haldener Str. 182, D-58095 Hagen, Germany
`weber@mfh-iserlohn.de`

Abstract. We present a new non-interactive public key distribution system based on the class group of a non-maximal imaginary quadratic order $Cl(\Delta_p)$. The main advantage of our system over earlier proposals based on $(\mathbb{Z}/n\mathbb{Z})^*$ [17, 19] is that embedding id information into group elements in a cyclic subgroup of the class group is easy (straight-forward embedding into prime ideals suffices) and secure, since the entire class group is cyclic with very high probability.
In order to compute discrete logarithms in the class group, the KGC needs to know the prime factorization of $\Delta_p = \Delta_1 p^2$. We present an algorithm for computing discrete logarithms in $Cl(\Delta_p)$ by reducing the problem to computing discrete logarithms in $Cl(\Delta_1)$ and either $\mathbb{F}_p^*$ or $\mathbb{F}_{p^2}^*$. Our algorithm is a special case of that in the more general setting of ray class groups [3], but we present it in terms of ideals of quadratic orders without using class field theoretic language, and we prove – for arbitrary non-maximal orders – that the reduction to discrete logarithms in the maximal order and a small number of finite fields has polynomial complexity if the factorization of the conductor is known.
Keywords: discrete logarithm, non-maximal imaginary quadratic order, non-interactive cryptography, identity based cryptosystem

1 Introduction

Public-key cryptography is undoubtedly one of the core techniques used to enable authentic, non-repudiable and confidential communication. However, a general problem inherent in public-key systems is that one needs to ensure the authenticity of a given public key. The most common way to solve this problem is to introduce a trusted third party, called a *Certification Authority* (CA), which issues certificates for public keys[1]. While this approach is widely used in practice, it would be desirable to have an immediate binding between an identity ID_B and its corresponding public key $\mathfrak{b}$, which allows one to avoid the tedious verification of certificates. This leads to the notion of *identity based cryptosystems*.

Although the paradigm of identity based cryptography was already introduced by Shamir in 1984 [21], it seems that Maurer and Yacobi [17] were the first to propose a *non-interactive* identity based public key cryptosystem in which Bob's public key $\mathfrak{b}$ can be derived efficiently, solely from his public identity information ID_B, by computing a publicly-known embedding function $\mathfrak{b} = f(ID_B)$. The main idea is to use an (ideally cyclic) group G (generated by $\mathfrak{g}$) in which exponentiation is not only a one-way-function but a *trapdoor-one-way-function*. The key generation center (KGC), a trusted third party responsible for distributing the private keys, knows the trapdoor information and hence is able to compute discrete logarithms in G. Thus, the KGC computes Bob's private key b such that $\mathfrak{g}^b = \mathfrak{b} = f(ID_B)$. The KGC hands over the secret key b to Bob, who can use this key in a conventional ElGamal- or Diffie-Hellman setup. As soon as all users are equipped with their corresponding secret key, the KGC can destroy the trapdoor-information and may cease to exist.

Maurer and Yacobi's initial proposal was to set up a discrete logarithm based system in $G = (\mathbb{Z}/n\mathbb{Z})^*$, where $n = p_1 \cdots p_r$, p_i prime, such that only the KGC, which knows the factorization of n, is able to compute discrete logarithms in G. However, this approach has a number of drawbacks which render such a scheme impractical [18, 16, 15].

In this paper, we show that using the class group $Cl(\Delta_p)$ of a non-maximal imaginary quadratic order is much better suited for this purpose. As in the orig-

[1] We assume throughout this work that Alice (A) wants to encrypt a message $m \in \mathbb{Z}_{>0}$ intended for Bob (B). We denote Bob's unique identity, for example his email-address, by ID_B and his public key by $\mathfrak{b}$.

inal scheme, the KGC knows some trapdoor information (the prime factorization of Δ_p) which enables it to compute discrete logarithms, while for anybody else the discrete logarithm problem (DLP) is assumed to be intractable. We generalize the recent result from [11], valid for the very special case of totally non-maximal orders with prime discriminant, to arbitrary non-maximal imaginary quadratic orders. The resulting algorithm reduces the problem of discrete logarithm computation in the class group of a non-maximal order to computing discrete logarithms in the much smaller class group of the corresponding maximal order and a small number of finite fields. Only the KGC, which knows the factorization of Δ_p, can perform this reduction.

As noted above there are a few advantages to our approach. Unlike the case of $(\mathbb{Z}/n\mathbb{Z})^*$, it is heuristically easy to find class groups $Cl(\Delta_p)$ which are *cyclic*, and hence the embedding of an identity ID_B into a group element $\mathfrak{b}$, for which the discrete logarithm exists, is straightforward. As the results from [18, 16] demonstrate, it seems to be no trivial task to find an embedding into a subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ which does not facilitate factoring n. In fact, the only secure embedding method for $(\mathbb{Z}/n\mathbb{Z})^*$ seems to restrict n to having *only two* large prime factors p_1 and p_2, and the workload for the KGC is consequently very high. Furthermore, since one chooses $p_i - 1$ smooth and uses Pohlig-Hellman's simplification together with Shank's Baby-Step Giant-Step algorithm, the time needed for generating k user keys is proportional to k.

In contrast, we use two different subexponential algorithms for the key generation. After the initial computation of relations over the factor bases, the workload for each individual key generation is very modest. For the computation of discrete logarithms in the class group of the maximal order, $Cl(\Delta_1)$, we use an analogue of the Self-Initializing Quadratic Sieve (SIQS) factoring algorithm [13, 12] and for the computation of discrete logarithms in $\mathbb{F}_p^*$ we use the Special Number Field Sieve, which recently was used for the solution of McCurley's challenge [22].

This paper is organized as follows: in Section 2 we provide the necessary background and notation for non-maximal imaginary quadratic orders. The next section contains the discrete logarithm algorithm for arbitrary non-maximal imaginary quadratic orders, and in Section 4 we present our new non-interactive public key cryptosystem. In order to save space, the proofs of most results have been omitted. These proofs, as well as computational results, will be given in the full paper [9].

2 Non-maximal imaginary quadratic orders

Let $\mathcal{O}_{\Delta_f}$ denote the non-maximal quadratic order of discriminant $\Delta_f = \Delta_1 f^2$ with conductor f, and let $\mathcal{O}_{\Delta_1}$ denote the corresponding maximal order. When the conductor is prime, we will use $\mathcal{O}_{\Delta_p}$ and Δ_p. By $Cl(\Delta_f)$ and $Cl(\Delta_1)$ we denote the ideal class groups of $\mathcal{O}_{\Delta_f}$ and $\mathcal{O}_{\Delta_1}$, respectively. The class numbers $h(\Delta_f)$ and $h(\Delta_1)$ are the orders of these groups. Lower-case Gothic letters $\mathfrak{a}, \mathfrak{b}, \ldots$ denote ideals in $\mathcal{O}_{\Delta_f}$ and upper-case Gothic letters denote ideals in $\mathcal{O}_{\Delta_1}$. Ideal equivalence is denoted by $\mathfrak{a} \sim \mathfrak{b}$, and the class of all ideals equivalent to $\mathfrak{a}$ is denoted by $[\mathfrak{a}]$. More background on imaginary quadratic number fields and non-maximal quadratic orders can be found in $[1, 2, 5]$.

Our cryptosystem makes use of the relationship between a non-maximal order of conductor f and its corresponding maximal order. Any non-maximal order can be represented as $\mathcal{O}_{\Delta_f} = \mathbb{Z} + f\mathcal{O}_{\Delta_1}$. If $h(\Delta_1) = 1$, then $\mathcal{O}_{\Delta_f}$ is called a totally non-maximal order. 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, and in every ideal equivalence class there is an ideal which is prime to any given number. We denote the principal $\mathcal{O}_{\Delta_f}$-ideals prime to f by $\mathcal{P}_{\Delta_f}(f)$ and all fractional ideals which are prime to f by $\mathcal{I}_{\Delta_f}(f)$. There is an isomorphism

$$\mathcal{I}_{\Delta_f}(f) / \mathcal{P}_{\Delta_f}(f) \simeq \mathcal{I}_{\Delta_f} / \mathcal{P}_{\Delta_f} = Cl(\Delta_f) \ , \qquad (1)$$

so we can "ignore" 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 $\mathcal{I}_{\Delta_f}(f)$, and $\mathcal{I}_{\Delta_1}(f)$, 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 $\mathcal{I}_{\Delta_1}(f) \tilde{\rightarrow} \mathcal{I}_{\Delta_f}(f)$. The inverse of this map is $\varphi^{-1} : \mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_1}$.*

Thus we are able to switch to and from ideals in the maximal and non-maximal orders via the map φ. The algorithms $\mathsf{GoToMaxOrder}(\mathfrak{a}, f)$ to compute φ^{-1} and $\mathsf{GoToNonMaxOrder}(\mathfrak{A}, f)$ to compute φ respectively can be found in [8]. If $\mathfrak{a} = (a, b)$ and $\mathfrak{A} = (A, B)$ are reduced ideals, then these algorithms need $O(\log(|\Delta_1|)^2)$ and $O(\log(|\Delta_f|)^2)$ bit-operations respectively.

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 $\phi_{Cl}^{-1} : Cl(\Delta_f) \rightarrow Cl(\Delta_1)$, *where* $[\mathfrak{a}] \mapsto [\rho_1(\varphi^{-1}(\mathfrak{a}))]$.

We now focus on the kernel $\mathrm{Ker}(\phi_{Cl}^{-1})$ of this map, which will turn out to be of central importance for the computation of discrete logarithms in $Cl(\Delta_f)$.

Proposition 3. *The map $\psi : (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^* \rightarrow \mathrm{Ker}(\phi_{Cl}^{-1})$, $\alpha \mapsto [\varphi(\alpha\mathcal{O}_{\Delta_1})]$, is a surjective homomorphism.*

This homomorphism suggests the following representation for ideal classes in the kernel:

Definition 1. *Let $\alpha = x + y\omega \in (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ and let $\mathfrak{a} = \varphi(\alpha) \in \mathrm{Ker}(\phi_{Cl}^{-1})$ be a reduced $\mathcal{O}_{\Delta_f}$-ideal. Then the pair (x, y) is called a* generator representation *for the class of $\mathfrak{a}$.*

Remark 1. Note that this generator representation (x, y) for the class of $\mathfrak{a}$ is *not unique*. It is easy to see that (kx, ky), $k \in (\mathbb{Z}/f\mathbb{Z})^*$, is also a generator representation for the class of $\mathfrak{a}$. This means that we have $\mathfrak{a} = \varphi(x + y\omega) \sim \varphi(kx + ky\omega)$.

Our reduction of the discrete logarithm problem in $Cl(\Delta_f)$ to $Cl(\Delta_1)$ and finite fields requires computing various preimages of elements in $\mathrm{Ker}(\phi_{Cl}^{-1})$ under the map ψ. Algorithm 1 (Std2Gen) accomplishes this task. The algorithm Reduce reduces an ideal $\mathfrak{A}$ given in standard representation and simultaneously computes a reducing number $\gamma \in \mathcal{O}_{\Delta_1}$ of the form $(x + y\sqrt{\Delta_1})/2$ such that $\mathfrak{A}/\gamma$ is reduced (see, for example, [13, Algorithm 2.6, p.16]).

Algorithm 1 Std2Gen

Require: The standard representation (a, b) of a reduced $\mathcal{O}_{\Delta_f}$-ideal $\mathfrak{a} = a\mathbb{Z} + \frac{b+\sqrt{\Delta_f}}{2}\mathbb{Z} \in \mathrm{Ker}(\phi_{Cl}^{-1})$ and the conductor f.

Ensure: A generator representation (x, y) of the class $\mathfrak{a} \sim \varphi(x + y\omega) \in \mathrm{Ker}(\phi_{Cl}^{-1})$, where $x + y\omega \in (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$.

$(a, b) \leftarrow$ GoToMaxOrder$(\mathfrak{a}, f)$
$(\mathfrak{G}, \gamma) \leftarrow$Reduce$(a, b)$
if $\mathfrak{G} \not\sim \mathcal{O}_{\Delta_1}$ **then**
  return('Error! $\mathfrak{a} \notin \mathrm{Ker}(\phi_{Cl}^{-1})$!')
end if
if $\Delta_1 \equiv 0 \pmod 4$ **then**
  $x \leftarrow x/2 \pmod f$
  $y \leftarrow y/2 \pmod f$
else
  $x \leftarrow (x - y)/2 \pmod f$
  $y \leftarrow y \pmod f$
end if
return$((x, y))$

3 The DLP for arbitrary $Cl(\Delta_f)$

In this section we generalize the result from [11]. We show that given the conductor f and its prime factorization one can reduce the DLP in an arbitrary

$Cl(\Delta_f)$ to the DLP in various smaller groups. More precisely, we first show that the computation of discrete logarithms in $Cl(\Delta_f)$ can be reduced to the computation of discrete logarithms in the class group $Cl(\Delta_1)$ of the maximal order and the computation of discrete logarithms in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$. Furthermore, we show that the latter problem boils down to the computation of discrete logarithms in a small number of finite fields.

It should be noted that our method here is in essence a special case of the more general methods employed by Cohen et al. to compute discrete logarithms in ray class groups [3]. The class group of a non-maximal order in any number field, not only degree 2, can be viewed as a ray class group of the maximal order, where the modulus is simply an integer, the conductor of the non-maximal order. Our exposition here is a reformulation of these results in terms of the simpler, special case of non-maximal orders using the language of [11]. In addition, we prove that the reduction of the DLP in $Cl(\Delta_f)$ to computing discrete logarithm computations in $Cl(\Delta_1)$ and a small number of finite fields is of polynomial complexity.

We start with an algorithm which reduces the DLP in $Cl(\Delta_f)$ to the DLP in $Cl(\Delta_1)$ and $\mathrm{Ker}(\phi_{Cl}^{-1})$. Since the map $\psi : (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^* \rightarrow \mathrm{Ker}(\phi_{Cl}^{-1})$ given in Proposition 3 is a surjective homomorphism, the latter DLP is clearly reduced to the DLP in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$. Thus, our algorithm makes use of the following two methods:

- $\mathsf{DLPinCl}(\mathfrak{G}, \mathfrak{A})$

  Accepts two reduced $\mathcal{O}_{\Delta_1}$-ideals $\mathfrak{G}, \mathfrak{A}$ as input and returns $x \in \mathbb{Z}$ with $0 \leq x < h(\Delta_1)$ such that $\mathfrak{G}^x \sim \mathfrak{A}$, or $x = -1$ if no such x exists.
- $\mathsf{DLPinOmodfO}(\gamma, \alpha)$

  Accepts two elements $\gamma, \alpha \in (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ as input and returns $x \in \mathbb{Z}$ with $0 \leq x < |(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*|$ such that $\gamma^x \equiv \alpha \pmod{f\mathcal{O}_{\Delta_1}}$, or $x = -1$ if no such x exists.

Furthermore, we assume that $h(\Delta_1)$ is known. This is no practical restriction, since the best currently known algorithm [13] for computing discrete logarithms in $Cl(\Delta_1)$ needs to compute $h(\Delta_1)$ and the group structure of $Cl(\Delta_1)$ before the actual DL-computation starts. Secondly, if there were any other algorithm DLPinCl with the above properties, then one could compute $h(\Delta_1)$, as shown in the full paper [9].

Algorithm 2 (ReduceDLP) reduces the DLP in $Cl(\Delta_f)$ to the DLP in $Cl(\Delta_1)$ and $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$. The proof of correctness can be found in the full version of the paper [9].

Algorithm 2 ReduceDLP

Require: Two reduced $\mathcal{O}_{\Delta_f}$-ideals $\mathfrak{g}, \mathfrak{a}$, the conductor f, the class number $h(\Delta_1)$, and the order of the kernel $|\mathrm{Ker}(\phi_{Cl}^{-1})| = \frac{f}{[\mathcal{O}_\Delta^* : \mathcal{O}_{\Delta f^2}^*]} \prod_{p \mid f} \left(1 - \frac{\left(\frac{\Delta}{p}\right)}{p}\right)$

Ensure: The discrete logarithm x, such that $\mathfrak{g}^x \sim \mathfrak{a}$, with $0 \le x < h(\Delta f^2)$, or $x = -1$, if no such x exists.

{Compute DL in $Cl(\Delta_1)$}
$\mathfrak{G} \leftarrow$GoToMaxOrder$(\mathfrak{g}, f)$
$\mathfrak{A} \leftarrow$GoToMaxOrder$(\mathfrak{a}, f)$
$x_1 \leftarrow$DLPinCl$(\mathfrak{G}, \mathfrak{A})$
if $x_1 = -1$ **then**
  return(-1)
end if
{Compute DL in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$}
$\alpha \leftarrow$Std2Gen$(\mathfrak{a}/\mathfrak{g}^{x_1}, f)$
$\gamma \leftarrow$Std2Gen$(\mathfrak{g}^{h(\Delta_1)}, f)$
$x_f \leftarrow$DLPinOmodfO(γ, α)
if $x_f = -1$ **then**
  return(-1)
end if
{Combine partial results to get DL in $Cl(\Delta f^2)$}
$c \equiv x_f \pmod{|\mathrm{Ker}(\phi_{Cl}^{-1})|}$ {, where $0 \le c < |\mathrm{Ker}(\phi_{Cl}^{-1})|$}
$x \leftarrow c \cdot h(\Delta_1) + x_1$
return(x)

Proposition 4. *Given the conductor f, the class number $h(\Delta_1)$ and the order of the kernel $|\mathrm{Ker}(\phi_{Cl}^{-1})|$ one can reduce the DLP in $Cl(\Delta_f)$ in $O(\log(|\Delta_f|)^3)$ bit-operations to the DLP in $Cl(\Delta_1)$ and $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$.*

Thus, in order to compute discrete logarithms in $Cl(\Delta_f)$, we need efficient algorithms for computing discrete logarithms in $Cl(\Delta_1)$ and $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$. The subexponential algorithm described in [12] is the most efficient algorithm known for computing discrete logarithms in $Cl(\Delta_1)$. We now consider the DLP in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ more closely.

By the Chinese Remainder Theorem (see, for example, [14, p.11]), the DLP in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ boils down to DLPs in $(\mathcal{O}_{\Delta_1}/p_i^{e_i}\mathcal{O}_{\Delta_1})^*$ for prime powers $p_i^{e_i}$, where $f = \prod p_i^{e_i}$. Furthermore, this problem can be efficiently reduced to the prime case $(\mathcal{O}_{\Delta_1}/p_i\mathcal{O}_{\Delta_1})^*$. We give an algorithm (ReducePe2P) for this reduction in the full version of the paper [9].

Proposition 5. *The DLP in $(\mathcal{O}_{\Delta_1}/p^e\mathcal{O}_{\Delta_1})^*$ can be reduced in $O(e \cdot (\log p^e)^3)$ bit-operations to $2e$ DL-computations in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$.*

Corollary 1. *If $e = O((\log p)^\alpha)$ for some $\alpha = O(1)$, then the DLP in $(\mathcal{O}_{\Delta_1}/p^e\mathcal{O}_{\Delta_1})^*$ can be reduced in polynomial time (in $\log p$) to the DLP in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$.*

Using ReduceDLP and ReducePe2P allows us to reduce the DLP in $Cl(\Delta_f)$ to DLPs in $Cl(\Delta_1)$ and $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$. As shown in [11, 10], the DLP in $(\mathcal{O}_{\Delta_1}/p\mathcal{O}_{\Delta_1})^*$ can in turn be reduced to the DLP in the finite field $\mathbb{F}_p^*$ or $\mathbb{F}_{p^2}^*$. This immediately leads to the central result of this section.

Theorem 1. *If the prime factorization of the conductor $f = \prod_{i=1}^k p_i^{e_i}$ is known and $e_i = O((\log p_i)^\alpha)$ for some $\alpha = O(1)$ then one can reduce the discrete logarithm problem in $Cl(\Delta_f)$ in polynomial time (in $\log \Delta_f$) to the computation of logarithms in $Cl(\Delta_1)$ and the following groups ($1 \leq i \leq k$):*

$$\begin{array}{l} \mathbb{F}_{p_i}^*, \text{ if } \left(\frac{\Delta_1}{p_i}\right) \in \{0, 1\} \\ \mathbb{F}_{p_i^2}^*, \text{ if } \left(\frac{\Delta_1}{p_i}\right) = -1 \ . \end{array}$$

Note that the central result of [11] now is nothing more than an immediate corollary.

3.1 Example

We illustrate the reduction of discrete logarithm computations in $Cl(\Delta_f)$ via a small example. Suppose $\Delta_1 = -1019$, $f = 23$, and $\Delta_f = \Delta_1 f^2 = -539051$. In this case, both $Cl(\Delta_f)$ and $Cl(\Delta_1)$ are cyclic with $h(\Delta_1) = 13$ and $h(\Delta_f) = h(\Delta_1)(23-1) = 286$. The equivalence class represented by the reduced ideal

$$\mathfrak{g} = 15\mathbb{Z} + \frac{-7+\sqrt{-539051}}{2}\mathbb{Z} = (15, -7)$$

generates $Cl(\Delta_f)$.

Suppose we wish to compute the discrete logarithm of $[\mathfrak{a}]$ with respect to the base $[\mathfrak{g}]$ in $Cl(\Delta_f)$, where

$$\mathfrak{a} = 11\mathbb{Z} + \frac{9+\sqrt{-539051}}{2}\mathbb{Z} = (11, 9) \ .$$

That is, we want to find x such that $\mathfrak{g}^x \sim \mathfrak{a}$. Since $\mathfrak{g}$ generates $Cl(\Delta_f)$, we know that such an x exists. Following ReduceDLP (Algorithm 2), we first compute $[\mathfrak{G}] = [\phi_{Cl}^{-1}(\mathfrak{g})]$ and $[\mathfrak{A}] = [\phi_{Cl}^{-1}(\mathfrak{a})]$, and solve the discrete logarithm problem

$$\mathfrak{G}^{x_1} \sim \mathfrak{A}$$

in $Cl(\Delta_1)$. We have $\mathfrak{G} = 15\mathbb{Z} + \frac{1+\sqrt{-1019}}{2}\mathbb{Z} = (15, 1)$, $\mathfrak{A} = (11, 9)$, and we easily compute $x_1 = 9$.

At this point we know that x has the form $x = c \cdot h(\Delta_1) + x_1 = 13c + 9$, and it remains to compute c. Again following ReduceDLP (Algorithm 2), we compute $\alpha, \gamma \in (\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$ such that $\psi(\alpha) \sim \mathfrak{a}/\mathfrak{g}^{x_1}$ and $\psi(\gamma) \sim \mathfrak{g}^{h(\Delta_1)}$. Following Std2Gen (Algorithm 1), we first compute

$$\mathfrak{b} = \mathfrak{a}/\mathfrak{g}^{x_1} = \mathfrak{a}/\mathfrak{g}^9 = (311, 277)$$

and

$$\mathfrak{c} = \mathfrak{g}^{h(\Delta_1)} = \mathfrak{g}^{13} = (297, 295) \ .$$

To find α and γ we compute the principal ideals $\mathfrak{B} = \varphi^{-1}(\mathfrak{b})$ and $\mathfrak{C} = \varphi^{-1}(\mathfrak{c})$, and reduce them while simultaneously computing their modulo $f\mathcal{O}_{\Delta_1}$ reduced generators, which we take as α and γ. We obtain $\mathfrak{B} = (311, -15) = (\alpha)$ and $\mathfrak{C} = (297, -13) = (\gamma)$ where

$$\alpha = -8 + 1\omega, \quad \gamma = -7 + 1\omega$$

and $\omega = \frac{1+\sqrt{-1019}}{2}$.

To compute c, we need to solve the discrete logarithm problem

$$\gamma^c \sim \omega$$

in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$. For this example, we have $(\Delta_1/f) = (-1019/23) = 1$, and thus $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^* \simeq \mathbb{F}_{23}^* \otimes \mathbb{F}_{23}^*$ by [11, Lemma 8]. Since $\omega \equiv 14 \pmod{23}$ and $\overline{\omega} \equiv 10 \pmod{23}$, we obtain

$$\gamma \mapsto (-7 + 1\omega \bmod 23, -7 + 1\overline{\omega} \bmod 23) = (7, 3) \in \mathbb{F}_{23}^* \otimes \mathbb{F}_{23}^*$$

and

$$\alpha \mapsto (-8 + 1\omega \bmod 23, -8 + 1\overline{\omega} \bmod 23) = (6, 2) \in \mathbb{F}_{23}^* \otimes \mathbb{F}_{23}^* \ .$$

Since α and γ represent equivalence classes in $(\mathcal{O}_{\Delta_1}/f\mathcal{O}_{\Delta_1})^*$, we need to find c by solving the discrete logarithm problem $(7, 3)^c = l(6, 2)$ in $\mathbb{F}_{23}^* \otimes \mathbb{F}_{23}^*$ for some $l \in \mathbb{F}_{23}^*$ (see Remark 1). This yields

$$7^c \equiv 6l \pmod{23}, \quad 3^c \equiv 2l \pmod{23},$$

and we combine these two discrete logarithm problems to obtain one discrete logarithm problem in $\mathbb{F}_{23}^*$:

$$(7/3)^c \equiv (6/2) \pmod{23} \rightarrow 10^c \equiv 3 \pmod{23} \ .$$

Solving yields $c = 20$, and finally $x = 12 \cdot 20 + 9 = 269$. It is easy to verify that x is indeed the desired discrete logarithm: simply compute the reduced ideal $\mathfrak{g}^{269}$ and verify that it is equal to the reduced ideal $\mathfrak{a}$.

4 Towards practical non-interactive cryptosystems

Before we explain our system setup we list the crucial properties:

Required Properties:

1. The discrete logarithm problem (DLP) in $Cl(\Delta_p)$ *without* knowing the factorization of $\Delta_p = \Delta_1 p^2$ is infeasible. The following bounds are based on the estimates in [7, Table 3], such that one expects to need more than 90.000 MIPS years to solve the problem.
   1.1 Δ_p is large enough that using the subexponential algorithm from [12] to directly compute discrete logarithms in $Cl(\Delta_p)$ is infeasible. $\Delta_p > 2^{423}$ is sufficient.
   1.2 Δ_p cannot be factored to reduce the DLP to DLPs in $Cl(\Delta_1)$ and $\mathbb{F}_p^*$ or $\mathbb{F}_{p^2}^*$.
      1.2.1 Δ_p is large enough so that the Number Field Sieve is infeasible. This yields $\Delta_p > 2^{576}$.
      1.2.2 Δ_1 and p are large enough that they cannot be found with the Elliptic Curve Method. This implies $\Delta_1, p > 2^{250}$.
2. Δ_1, p must be small enough to enable the KGC to compute discrete logarithms in $Cl(\Delta_1)$ and $\mathbb{F}_p^*$ or $\mathbb{F}_{p^2}^*$ using subexponential algorithms. $\Delta_1, p < 2^{300}$ seems to be feasible.
3. $Cl(\Delta_p)$ must be cyclic.

It is easy to see that the following setup satisfies *all* above requirements.

System Setup:

1. The KGC randomly chooses a prime $q \equiv 3 \pmod 4$, $q > 2^{260}$, sets $\Delta_1 = -q$ and computes $h(\Delta_1)$ and the group structure of $Cl(\Delta_1)$ with the algorithm from [13]. The Cohen-Lenstra heuristics [4] suggest that $Cl(\Delta_1)$ is cyclic with probability > 0.97. If $Cl(\Delta_1)$ is not cyclic, the KGC selects another prime q until it is cyclic.
2. The KGC chooses a prime $p > 2^{260}$ with $\gcd(p - (\Delta_1/p), h(\Delta_1)) = 1$ (here (Δ_1/p) denotes the Legendre symbol), such that the SNFS can be applied as in [22], and computes $\Delta_p = \Delta_1 p^2$. The gcd condition ensures that $Cl(\Delta_p)$ is cyclic.

3. The KGC computes a generator $\mathfrak{g}$ of $Cl(\Delta_p)$ and publishes it together with Δ_p.

Given a generator $\mathfrak{G}$ of $Cl(\Delta_1)$, which the KGC can easily obtain during the computation of $Cl(\Delta_1)$ [13, Algorithm 6.1], it is also easy in practice to find a generator $\mathfrak{g}$ of $Cl(\Delta_p)$ with the additional property that $\phi_{Cl}^{-1}(\mathfrak{g}) = \mathfrak{G}$. The KGC repeatedly selects random values of $\alpha \in \mathcal{O}_{\Delta_1}$ and takes the first $\mathfrak{g} = \phi(\alpha\mathfrak{G})$ such that $\mathfrak{g}^{h(\Delta_p)/2} \not\sim \mathcal{O}_{\Delta_p}$.

User Registration:

1. Bob requests the public key $\mathfrak{b}$ corresponding to his identity ID_B at the KGC.
2. The KGC verifies Bob's identity, for example, using a passport, and starts with the key generation.
3. The KGC computes the 128-bit hash $id = h(ID_B)$ using, for example, MD5 [20], of Bob's identity and embeds id into a group element of $Cl(\Delta_p)$ by taking the largest prime $p_B \leq id$, for which $\left(\frac{\Delta_p}{p_B}\right) = 1$ and computing the prime ideal $\mathfrak{b} = p_B\mathbb{Z} + \frac{b_B+\sqrt{\Delta_p}}{2}$. Note that $\mathfrak{b}$ is already reduced, since $\sqrt{|\Delta_p|} > 2^{128} > p_B$. If the KGC recognizes that $\mathfrak{b}$ is already assigned to another user it will ask Bob to choose another identity, for example, his postal address.
4. Finally, the KGC computes the discrete logarithm b such that $\mathfrak{g}^b \sim \mathfrak{b}$ using the secret knowledge of the conductor p and the reduction procedure described in the Section 3, and returns b to Bob.

As soon as all users are registered this way the KGC can destroy the factorization of Δ_p and cease to exist. The users can obtain any other user's *authentic* public key simply by hashing that user's identity and computing the largest prime ideal whose norm is less than the hash value. Each user has a public/private key-pair $(\mathfrak{a}, a)$ with $\mathfrak{a} \sim \mathfrak{g}^a$, so discrete logarithm-based protocols such as Diffie-Hellman or El Gamal can be directly applied in the class group $Cl(\Delta_p)$.

Preliminary experiments, together with computational experience using the subexponential algorithms from [12] and [22], indicate that a KGC with modest computational resources, for example, a small network of Pentium processors, should be able to set up a key distribution system with using $p, q \approx 2^{300}$ at least. Computational results will appear in the full version of the paper [9].

References

1. Z.I. Borevich and I.R. Shafarevich. *Number Theory.* Academic Press, New York, 1966.
2. H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, Berlin, 1993.
3. H. Cohen, F. Diaz y Diaz, and M. Olivier. Computing ray class groups, conductors, and discriminants. *Math. Comp.*, 67(222):773–795, 1998.
4. H. Cohen and H.W. Lenstra, Jr. Heuristics on class groups of number fields. In *Number Theory, Lecture notes in Math.*, volume 1068, pages 33–62. Springer-Verlag, New York, 1983.
5. D.A. Cox. *Primes of the form* $x^2 + ny^2$. John Wiley & Sons, New York, 1989.
6. D. Hühnlein. Efficient implementation of cryptosystems based on non-maximal imaginary quadratic orders. In *Selected Areas in Cryptography - SAC'99*, volume 1758 of *LNCS*, pages 150–167, 1999.
7. D. Hühnlein. Quadratic orders for NESSIE – Overview and parameter sizes of three public key families. Submitted to ISSE 2000, preprint via http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/Welcome.html#2000.
8. D. Hühnlein, M.J. Jacobson, Jr., S. Paulus, and T. Takagi. A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption. In *Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *LNCS*, pages 294–307, 1998.
9. D. Hühnlein, M.J. Jacobson, and D. Weber. Towards practical non-interactive public key cryptosystems using non-maximal imaginary quadratic orders. CORR technical report, www.cacr.math.uwaterloo.ca, to appear.
10. D. Hühnlein and J. Merkle. An efficient NICE-Schnorr-type signature scheme. To appear at PKC2000 and in Springer LNCS, preprint via http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/Welcome.html#1999.

11. D. Hühnlein and T. Takagi. Reducing logarithms in totally non-maximal imaginary quadratic orders to logarithms in finite fields. In *Advances in Cryptology - ASIACRYPT '99*, LNCS, 1999.

12. M.J. Jacobson, Jr. Computing discrete logarithms in quadratic orders. To appear Journal of Cryptology, 1999.

13. M.J. Jacobson, Jr. *Subexponential Class Group Computation in Quadratic Orders.* PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, 1999.

14. S. Lang. *Algebraic number theory.* Springer, Berlin, 2nd edition, 1991. ISBN 3-540-94225-4.

15. C.H. Lim and P.J. Lee. Modified Maurer-Yacobi's scheme and its applications. In *Proceedings of Auscrypt'92*, LNCS, pages 308–323, 1992.

16. M. Maurer and D. Kügler. A note on the weakness of the Maurer-Yacobi squaring method. Technical report, Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany, 1999. To appear.

17. U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *LNCS*, pages 498–507, 1991.

18. U. Maurer and Y. Yacobi. A remark on a non-interactive public-key distribution system. In *Advances in Cryptology - EUROCRYPT'92*, volume 658 of *LNCS*, pages 458–460, 1993.

19. U. Maurer and Y. Yacobi. A non-interactive public-key distribution system. *Design Codes and Cryptography*, 9:305–316, 1996.

20. R. Rivest. The MD5 message-digest algorithm, 1992. RFC1321, Internet Activities Board, Internet Engineering Task Force.

21. A. Shamir. Identity based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *LNCS*, pages 47–53, 1985.

22. D. Weber and T. Denny. The solution of McCurley's discrete log challenge. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *LNCS*, pages 56–60, 1998.