Security of Cryptosystems Based on Class Groups of Imaginary Quadratic Orders

Safuat Hamdy Bodo Möller

Fachbereich Informatik, TU Darmstadt

{hamdy,moeller}@cdc.informatik.tu-darmstadt.de

May 29, 2000

Abstract

In this work we investigate the difficulty of the discrete logarithm problem in class groups of imaginary quadratic orders. In particular, we discuss several strategies to compute discrete logarithms in those class groups. Based on heuristic reasoning, we give advice for selecting the cryptographic parameter, i.e. the discriminant, such that cryptosystems based on class groups of imaginary quadratic orders would offer a similar security as commonly used cryptosystems.

1 Introduction

Cryptosystems based on class groups of imaginary quadratic orders (IQC) have been first proposed by Buchmann and Williams [3, 4] in 1988 and 1990. Since then, there was no clear advice on how to select the cryptographic parameter, i.e. the discriminant of the quadratic order. The goal of this work is to close this gap. In particular, we demonstrate how large Δ must be selected such that computing logarithms in $Cl(\Delta)$ is as hard as factoring an integer n of given size. We consider several strategies to compute discrete logarithms in class groups, such as reductions to other computational problems, index-calculus algorithms, Pollard's λ algorithm, and the Pohlig-Hellman algorithm in connection with an algorithm similar to the $(p-1)$-factoring method. In particular, in order to get the same security with IQC as with RSA with 1024 bit moduli, the discriminant should have at least 687 bits.

The security of IQC is based on the apparent difficulty of computing discrete logarithms in class groups of imaginary quadratic orders (Cl-DLP). The Cl-DLP can be extended to class groups of orders of number fields with arbitrarily high degree, and in furthermore, there is a generalization of the discrete logarithm problem [2]. However, in this work we shall focus only on imaginary quadratic fields, and whenever the term "class groups" appears in the sequel, we actually mean class groups of imaginary quadratic orders.

It is well known that solving the Cl-DLP is at least as hard as solving the integer factorization problem (IFP, we shall describe the reduction later in this work), yet it is still unknown whether the Cl-DLP is really harder than the IFP. The Cl-DLP can be solved with

a subexponential index-calculus algorithm due to Hafner and McCurley [11]. This algorithm was improved by Düllmann [9]. Recently, in [28] it has been rigorously proven that for solving the Cl-DLP one can expect a running time proportional to $L_{|\Delta|}\left[\frac{1}{2}, \frac{3}{4}\sqrt{2} + o(1)\right]$, where Δ is the discriminant of the imaginary quadratic order. Moreover, Jacobson [15] has applied the ideas of the MPQS to class group computations. In fact, the machinery behind his algorithm is the same as that of the original MPQS, and although his algorithm hasn't been analyzed, empirical data suggest a running time proportional to $L_{|\Delta|}\left[\frac{1}{2}, 1 + o(1)\right]$.

The best known algorithm to solve the IFP is the GNFS with asymptotic expected running time proportional to $L_n\left[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right]$; the best known algorithm to solve the GF-DLP (DLP in multiplicative groups of Galois fields) is a variant of the GNFS with the same asymptotic expected running time. Thus, currently the IFP or the GF-DLP can be solved asymptotically faster than the Cl-DLP. This means that the Cl-DLP is apparently harder than the IFP or the GF-DLP.

Hence class groups form another potential alternative to finite fields for DL-based cryptographic protocols. Unfortunately, popular signature protocols such as DSA can't be used with class groups in a direct way, because DSA requires the knowledge of the group order. Computing the order of an arbitrary class group appears to be as hard as computing discrete logarithms in class groups, because there's no efficient algorithm known that computes the class number. In [21] a variant of the Schnorr signature scheme that doesn't require knowledge of the group order has been proposed.

Computing roots without knowing the class number also appears to be intractable. This makes the Guillou-Quisquater signature protocol [10] suitable for class groups, since in this protocol even the signer does not need to know the class number. Moreover, in [1] a variant of DSA was presented that is based on the intractability to compute roots in finite abelian groups.

This paper is organized as follows: In Section 2 we recall the background we need, and in Section 3 we give advice for selecting the security parameters.

2 Class groups

Recall that we consider class groups of imaginary quadratic fields only. We shall state only some necessary facts without proofs; for details we refer to [8, 5]. Let Δ be a negative integer such that $\Delta \equiv 0, 1 \pmod 4$. Then Δ is the discriminant of a unique order $\mathcal{O}_\Delta = \mathbb{Z} + \mathbb{Z}(\Delta + \sqrt{\Delta})/2$ of $\mathbb{Q}(\sqrt{\Delta})$. $\mathcal{O}_\Delta$ is maximal if and only if Δ is fundamental, i.e. Δ or $\Delta/4$ is square free if $\Delta \equiv 1 \pmod 4$ or $\Delta \equiv 0 \pmod 4$, respectively.

Let $\mathcal{O}_\Delta$ be any (not necessarily maximal) order. The class group of $\mathcal{O}_\Delta$ is denoted by $Cl(\Delta)$, its elements are equivalence classes of invertible ideals of $\mathcal{O}_\Delta$. The group order of $Cl(\Delta)$ is the class number $h(\Delta)$. Later in this work we shall need the odd parts of class groups. We denote the odd part of a class group $Cl(\Delta)$ by $Cl_{\mathsf{odd}}(\Delta)$ and its cardinality by $h_{\mathsf{odd}}(\Delta)$.

Any ideal of $\mathcal{O}_\Delta$ can be expressed as $\mathbb{Z}a + \mathbb{Z}(b + \sqrt{\Delta})/2$ such that $a, b \in \mathbb{Z}$, $a > 0$ and $4a \mid (b^2 - \Delta)$, that is, such that there exists a positive integer c such that $\Delta = b^2 - 4ac$. Thus we represent ideals as pairs (a, b) of integers. Observe that if $b = 0$ or $b = a$, then $\Delta = -4ac$

or $\Delta = a(a-4c)$, respectively, and if $a = c$, then $\Delta = (b-2a)(b+2a)$. Such ideals are called *ambiguous* and have order two in $Cl(\Delta)$.

An ideal is said to be *reduced* if $\gcd(a,b,c) = 1$, $-a < b \leq a \leq c$, and $b \geq 0$ if $a = c$. Each equivalence class of $\mathcal{O}_\Delta$ contains exactly one reduced ideal, thus the elements of $Cl(\Delta)$ can be represented by the reduced ideals of $\mathcal{O}_\Delta$, and checking equality of two ideal classes means to compare the representatives. The neutral element of $Cl(\Delta)$ is represented by $(1, \Delta \bmod 2)$. The group operation of $Cl(\Delta)$ is ideal multiplication with reduction (e.g. see [15] or [5, Chap. 5]). It can be shown that a group operation requires $O(\log^2 |\Delta|)$ bit operations. The inverse of an ideal (a,b) under this operation is $(a,-b)$. If an ideal (a,b) is reduced, then $a < \sqrt{|\Delta|/3}$, therefore $a, |b| = O(\sqrt{|\Delta|})$.

3 Selecting the class group

In this section we shall see that the discriminant is the cryptographic parameter. We shall discuss how to select a discriminant such that, based on heuristic grounds, computing discrete logarithms or the order of arbitrary elements in the corresponding class group is intractable. In particular,

- Δ must be chosen so that there is no efficient reduction of the CL-DLP to simpler problems,
- $|\Delta|$ must be large enough to preclude attacks with index-calculus algorithms,
- $h(\Delta)$ must be large enough to preclude attacks with λ algorithms,
- $h(\Delta)$ must not be smooth in order to preclude the computation of $h(\Delta)$ by an algorithm similar to the $(p-1)$-factoring algorithm with subsequent application of the Pohlig-Hellman algorithm.

It is tempting to ask whether the discriminant can be chosen such that its class number has properties selected à priori. However, we have no control over the class number, i.e. there's not even a probabilistic efficient algorithm known which outputs a fundamental discriminant whose class number has certain interesting properties (e.g. contains a large prime factor).

We shall show in the following subsections that if Δ is chosen appropriately, then the above conditions hold with high probability. In particular, in Sect. 3.1 we show that selecting $\Delta = -p$ or $\Delta = -8pq$ where p, q are primes precludes reductions to the GF-DLP and keeps the two-part of $Cl(\Delta)$ small. In Sect. 3.2 we show how large Δ must be to preclude index-calculus attacks. In Sect. 3.3 we show how large the class group must be to preclude attacks with the aid of Pollard's λ-method; based on the Brauer-Siegel theorem we deduce the size of the discriminant. In Sect. 3.4 we describe the relevance of the Pohlig-Hellman algorithm for class groups and discuss a possible application in conjunction with another algorithm, which is similar to the $(p-1)$-factoring method. Let a smoothness bound B be given; in Sect. 3.5, based on heuristic assumptions we show how Δ must be chosen so that the class number is B-smooth only with negligible probability.

It turns out that asymptotically the index-calculus methods dominate the selection of the discriminant with respect to order of magnitude. Moreover, since the best known algorithm

to compute class numbers of fundamental discriminants are again index-calculus methods, it is infeasible to compute the class number of fundamental discriminants if these are large. Therefore, the Pohlig-Hellman algorithm plays no role for class groups of maximal orders, unless the class number is smooth, because then an algorithm similar to the $(p-1)$-factoring algorithm can be applied to compute the class number.

3.1 Class group computation by reduction to other problems

Let Δ be a negative fundamental discriminant and let f be a positive integer. Then, if $\Delta \neq -3, -4$

$$h(\Delta f^2) = h(\Delta) f \prod_{p|f} \left(1 - \left(\frac{\Delta}{p}\right) \frac{1}{p}\right) \ ,$$

where (Δ/p) denotes the Kronecker symbol. For instance $h(-8) = 1$ and $h(-8p^2) = p - (-8/p)$. Since in general it is intractable to compute class numbers of large *fundamental* discriminants (see below), this could be a nice way to avoid it altogether.

However, the Cl-DLP in $Cl(-8p^2)$ can be reduced in polynomial time to the GF-DLP in $\mathbb{F}_p$ [13]. Currently no efficient reductions of this type for maximal orders are known, therefore we shall use only class groups of maximal orders, and in the sequel Δ will always be fundamental and thus $\mathcal{O}_\Delta$ will be maximal.

3.1.1 Selection of a fundamental discriminant

In order to check whether an arbitrary discriminant Δ is fundamental, it must be checked whether Δ (if $\Delta \equiv 1 \pmod 4$) or $\Delta/4$ (if $\Delta \equiv 0 \pmod 4$) is square free. This can be achieved by factoring the discriminant, but this is infeasible if the discriminant under consideration is large. A better method is to construct D from distinct prime factors, and set $\Delta = -D$ if $D \equiv 3 \pmod 4$ and $\Delta = -4D$ otherwise.

Some of the simplest cases are

1. $\Delta = -p$ where $p \equiv 3 \pmod 4$ is prime; and
2. $\Delta = -8pq$ where p and q are primes such that $p \equiv 1 \pmod 8$, $p + q \equiv 8 \pmod{16}$, and $(p/q) = -1$, where (p/q) denotes the Legendre symbol.

Discriminants selected like this have the additional advantage that the two-part of the class number is known to be small: In case 1, $h(\Delta)$ is odd; in case 2, the even part of $h(\Delta)$ is exactly 8 (see [16, Proposition B'_9]).

Observe that $\Delta = -8pq$ is attractive by a complexity theoretic argument, because if Δ is composite, then $Cl(\Delta)$ has non-trivial ambiguous elements, whose components lead immediately to a factorization of Δ; these ambiguous elements can be obtained by computing discrete logarithms in $Cl(\Delta)$, therefore IFP $\leq$ Cl-DLP.

3.2 Class group computations by index-calculus techniques

Let $L_x[e,c]$ be defined as usual, that is

$$L_x[e,c] \stackrel{\text{def}}{=} \exp\left(c(\log x)^e(\log\log x)^{1-e}\right)$$

for real positive x, real positive c, and $0 \leq e \leq 1$. In practice, instead of the term $L_x[e,c]$ we often see $L_x[e,c+o(1)]$, but in the sequel we shall ignore the $o(1)$ term.

We want to compare the expected computational work for solving the IFP and the Cl-DLP. In the following, we assume the expected running time for factoring an integer n by the GNFS to be proportional to $L_n\left[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right]$. For the Cl-DLP, an expected running time proportional to $L_{|\Delta|}\left[\frac{1}{2}, \frac{3}{4}\sqrt{2}\right]$ has been shown in [28]. However, Jacobson [15] showed that one can use a variant of the MPQS for DL-computations in $Cl(\Delta)$. The MPQS factoring algorithm has a conjectured expected running time proportional to $L_n\left[\frac{1}{2}, 1\right]$, while the MPQS DL-computation algorithm hasn't been analyzed, yet (not even heuristically). Empirical data suggests an expected running time of $L_{|\Delta|}\left[\frac{1}{2}, 1\right]$, so we shall base our arguments on this running time. In terms of security and efficiency, this will yield slightly larger keys: If we underestimate the running time of the Cl-MPQS, we overestimate the size of the security relevant parameters. This "conservative" approach is quite common practice.

The usual approach to estimate running times of an algorithm for large input parameters is to start from the empirical running time for smaller input parameters. If x_1 and x_2 are inputs for an algorithm with expected running time $L_x[e,c]$ and t_1 is the running time of the algorithm when executed with x_1, then (see [20] or [17]) the running time t_2 of the algorithm with input x_2 can be estimated by the equation

$$\frac{L_{x_1}[e,c]}{L_{x_2}[e,c]} = \frac{t_1}{t_2} \ . \tag{1}$$

However, this holds only if the sizes of x_1 and x_2 do not differ too much, otherwise it can't be ignored that $o(1) \to 0$. Thus, if x_2 is much larger than x_1, then t_2 will be a significant overestimate. To obtain more precise estimates a finer expression for the running time must be used or the $o(1)$ term must be taken into account by modifying (1) as in [12]. We stick to (1), since the estimates presented here differ only slightly from those given in [12].

magnitude of n	expected no. of MIPS-years to factor n
2^{512}	8.00×10^3
2^{768}	4.91×10^7
2^{1024}	5.99×10^{10}
2^{1280}	2.68×10^{13}
2^{1536}	5.97×10^{15}
2^{1792}	7.91×10^{17}
2^{2048}	6.98×10^{19}
2^{2560}	2.16×10^{23}
2^{3072}	2.64×10^{26}
2^{3584}	1.63×10^{29}
2^{4096}	5.87×10^{31}

Table 1: Estimated expected computational work of the GNFS for larger inputs

Table 1 shows some extrapolated running times for the GNFS. They are based on data points of the factorization of RSA-155 (155 decimal digits, 512 bits) with the GNFS [26]. In particular, it was estimated that about 8000 MIPS-years were spent.

To estimate the expected running time of the MPQS for DL-computations in class groups for large groups, we made extensive experiments where we computed discrete logarithms in 20 class groups of different negative discriminants for each magnitude tabulated below. The computations were carried out on a Sparc with ULTRA-170 processor. The results are summarized in Table 2.

magnitude of $\lvert\Delta\rvert$	mean running time (sec) $\overline{t_\Delta}$	standard deviation	$L_{\lvert\Delta\rvert}\left[\frac{1}{2},1\right] / \overline{t_\Delta}$ $(\sec^{-1})$
2^{140}	8.59×10^1	3.58×10^1	1.65×10^7
2^{142}	1.29×10^2	8.66×10^1	1.31×10^7
2^{144}	1.36×10^2	5.32×10^1	1.50×10^7
2^{146}	1.32×10^2	3.87×10^1	1.85×10^7
2^{148}	1.98×10^2	6.98×10^1	1.47×10^7
2^{150}	2.20×10^2	1.38×10^2	1.59×10^7
2^{152}	2.63×10^2	1.44×10^2	1.59×10^7
2^{154}	3.26×10^2	1.82×10^2	1.53×10^7
2^{156}	3.52×10^2	1.64×10^2	1.69×10^7
2^{158}	4.90×10^2	3.28×10^2	1.44×10^7
2^{160}	4.41×10^2	1.98×10^2	1.90×10^7
2^{162}	7.67×10^2	4.21×10^2	1.30×10^7
2^{164}	6.84×10^2	2.20×10^2	1.73×10^7
2^{166}	8.79×10^2	3.22×10^2	1.60×10^7
2^{168}	1.07×10^3	4.12×10^2	1.56×10^7
2^{170}	1.49×10^3	8.25×10^2	1.33×10^7
2^{172}	1.74×10^3	8.99×10^2	1.34×10^7
2^{174}	1.54×10^3	9.83×10^2	1.79×10^7
2^{176}	1.61×10^3	8.45×10^2	2.03×10^7
2^{178}	2.77×10^3	1.37×10^3	1.39×10^7
2^{180}	2.73×10^3	1.39×10^3	1.67×10^7
2^{184}	3.37×10^3	1.82×10^3	1.87×10^7
2^{188}	4.07×10^3	1.95×10^3	2.14×10^7
2^{192}	5.96×10^3	2.86×10^3	2.02×10^7
2^{196}	9.23×10^3	3.80×10^3	1.79×10^7
2^{200}	1.30×10^4	5.13×10^3	1.74×10^7
2^{210}	2.63×10^4	8.49×10^3	1.87×10^7
2^{220}	6.28×10^4	3.78×10^4	1.68×10^7

Table 2: Empirical computational work of the Cl-MPQS for relatively small inputs

Table 2 supports the conjectured running time of $L_{\lvert\Delta\rvert}\left[\frac{1}{2},1\right]$ for the MPQS. Note also that the standard deviation is almost always about half the running time. This shows that the

running times are pretty spread, which in turn confirms our suspicions of taking just a single sample.

All computations were performed on a SUN-workstation with a Sparc ULTRA-170 processor. SUN Microsystems does not publish MIPS ratings for its machines, and in fact, the unit MIPS-year is actually not appropriate [25]. However, it is widely used, so for simplicity we assume 100 MIPS, which is a value of reasonable order of magnitude for this machine. By Table 2 let us assume that $L_{|\Delta|}\left[\frac{1}{2},1\right]/t_\Delta = 1.8 \times 10^7 \text{ sec}^{-1}$. Then we get the extrapolations in Table 3.

When we align the parameters of the IFP and of the Cl-DLP in such a way that the expected running time for solving the Cl-DLP roughly equals the expected running time for solving the IFP for n of some particular magnitudes, we get Table 4.

magnitude of $\lvert\Delta\rvert$	expected no. of MIPS-years for solving the Cl-DLP in $Cl(\Delta)$
2^{256}	2.58
2^{348}	9.75×10^3
2^{512}	1.18×10^7
2^{640}	6.74×10^9
2^{768}	2.24×10^{12}
2^{896}	4.94×10^{14}
2^{1024}	7.79×10^{16}
2^{1280}	8.90×10^{20}
2^{1536}	4.56×10^{24}
2^{1792}	1.26×10^{28}
2^{2048}	2.13×10^{31}
2^{2560}	1.92×10^{37}
2^{3072}	5.30×10^{42}
2^{3584}	5.88×10^{47}
2^{4096}	3.15×10^{52}

Table 3: Estimated expected computational work of the Cl-MPQS for larger inputs

magnitude of n	magnitude of $\lvert\Delta\rvert$	expected no. of MIPS-years
2^{768}	2^{540}	4.99×10^7
2^{1024}	2^{687}	6.01×10^{10}
2^{1536}	2^{958}	5.95×10^{15}
2^{2048}	2^{1208}	7.05×10^{19}
2^{3072}	2^{1665}	2.65×10^{26}
2^{4096}	2^{2084}	5.87×10^{31}

Table 4: Estimated expected computational work of the GNFS and the Cl-MPQS aligned

3.3 Class group computations by Pollard's λ method

We now consider Pollard's λ method for computing discrete logarithms, orders of group elements and hence roots of group elements. From [27] it is known that the unparallelized version of this algorithm takes $\sqrt{\pi|G|/2}$ group operations (ignoring lower order terms) for cyclic groups G. Moreover, r-fold parallelization speeds the λ-method up by factor r.

By the heuristics of Cohen and Lenstra [7, 6], the probability that $Cl_{\mathsf{odd}}(\Delta)$ is cyclic is equal to $0.9775\ldots$. Moreover, it can be deduced from the heuristics that if $Cl_{\mathsf{odd}}(\Delta)$ is not cyclic, then with high probability $Cl_{\mathsf{odd}}(\Delta)$ has a cyclic subgroup G_{cyc} such that $|G_{\mathsf{cyc}}|$ has the same order of magnitude as $h_{\mathsf{odd}}(\Delta)$, and therefore, by our selection of Δ, the even part is 1 or 8 and thus $|G_{\mathsf{cyc}}|$ and $h(\Delta)$ have the same order of magnitude.

In order to provide a lower bound for Δ we need an (asymptotic) lower bound for $h(\Delta)$ that depends on Δ only. The best *proven* explicit lower bound is $h(\Delta) > 1/55 \ln|\Delta| \prod_{p|\Delta}(1 - 2\sqrt{p}/(p+1))$ [5, Sect. 5.10.1], which is too weak for our purposes. By the Brauer-Siegel Theorem we know that $\ln h(\Delta) \sim \ln\sqrt{|\Delta|}$ as $\Delta \to -\infty$, that is, $\sqrt{|\Delta|}^{1-\epsilon} \leq h(\Delta) \leq \sqrt{|\Delta|}^{1+\epsilon}$ for any positive real ϵ and sufficiently large Δ, but no explicit constants are known to make this statement effective. However, it is possible to show that $h(\Delta)$ is *on average* $c\sqrt{|\Delta|}$ with $c = 0.461559\ldots$ [5, Sect. 5.10.1]. This result has been proven for averages taken over class numbers of fundamental discriminants. In this work we make the assumption that this result is not affected by the restriction to the special discriminants given in section 3.1.1.

Example The time to perform a single group operation in $Cl(\Delta)$ depends on Δ, yet let us assume a fixed time of 1 ms on a machine with a computing power of 100 MIPS. Then the computational work of a single MIPS-year is equivalent to 2^{29} group operations. Based on this assumption and on the assumed average for the class number of a prime discriminant, in Table 5 we present some samples for (prime) discriminants, their average class number, and the expected computing amount to compute discrete logarithms by the λ-method; compare this with Table 1.

magnitude of $h(\Delta)$	magnitude of $\lvert\Delta\rvert$	expected no. of Group operations $\sqrt{\pi h(\Delta)/2}$	expected no. of MIPS-years
2^{108}	2^{218}	2^{54}	4.56×10^{7}
2^{129}	2^{260}	2^{64}	6.60×10^{10}
2^{162}	2^{326}	2^{81}	6.12×10^{15}
2^{189}	2^{380}	2^{94}	7.09×10^{19}
2^{233}	2^{468}	2^{116}	2.97×10^{26}
2^{268}	2^{538}	2^{134}	5.51×10^{31}

Table 5: Estimated expected computational work of the λ-method

3.4 Class group computations and the Pohlig-Hellman algorithm

The Pohlig-Hellman algorithm utilizes the prime factorization of the group order in order to simplify DL computations. However, the best known algorithm for computing the class

number is a variant of MPQS for DL computations in class groups and has the same expected asymptotic running time. Thus, if $|\Delta|$ is large, it is infeasible to compute $h(\Delta)$ or even odd multiples or factors (in particular the smooth part) of $h(\Delta)$. Moreover, there is no efficient method known that checks whether a particular odd prime divides $h(\Delta)$. Consequently, the Pohlig-Hellman algorithm is not applicable to class groups in general. There are also cryptographic protocols (e.g. the Guillou-Quisquater signature protocol) that depend explicitly on the fact that the group order is unknown.

We now consider the special case when $h(\Delta)$ is smooth. If the class number is smooth, there is a practical method to compute the order of an arbitrary element by a method similar to the $(p-1)$-factoring method. That is, given $\gamma \in Cl(\Delta)$, set $\alpha_0 = \gamma$ and successively compute $\alpha_i = \alpha_{i-1}^{p_i^{e(p_i,B)}}$ for all $p_i \leq B$, where p_i is the ith prime, B is a smoothness bound, and $e(p_i, B)$ depends only on p_i and B. For instance, if $e(p_i, B) = \log_{p_i} B$ for each p_i, then the algorithm will cover each possible prime power below the smoothness bound.

If $h(\Delta)$ is B-smooth, then this computation may yield $1_{Cl(\Delta)}$. If this happens, then there is an i such that $\alpha_i = 1_{Cl(\Delta)}$ but $\alpha_{i-1} \neq 1_{Cl(\Delta)}$, and we immediately know that p_i is the largest prime factor of $\mathrm{ord}_{Cl(\Delta)}\,\gamma$. If we set $\gamma' = \gamma^{p_i^{e(p_i)}}$ where $e(p_i)$ is the smallest positive integer such that $\alpha_{i-1}^{p_i^{e(p_i)}} = 1_{Cl(\Delta)}$ and repeat the complete procedure with γ', then we obtain the second largest prime factor, and eventually we get the complete prime factorization of $\mathrm{ord}_{Cl(\Delta)}\,\gamma$. Then we are able to compute roots as well as discrete logarithms in $\langle\gamma\rangle$ with the aid of the Pohlig-Hellman algorithm.

Assume that the $(p-1)$-like method above succeeds for an element γ and a bound B, and let q denote the largest prime factor of $\mathrm{ord}_{Cl(\Delta)}\,\gamma$. It is obvious that if we use a fast exponentiation method, then we have to perform at least $\sum_{p<q} e(p, B) \log_2 p$ group operations to find q. In order to find a smoothness-bound, we must consider the easiest case, i.e. $e(p_i, B) = 1$ for all p_i. Now $\sum_{p<q} \log_2 p = \theta(q)/\ln 2$, where θ is the Chebyshev θ-function. In [23, 24] it was shown that $0.998697\,x < \theta(x) < 1.001093\,x$ for all $x \geq 1155901$ (under assumption of the Riemann hypothesis, it is even possible to show that $|\theta(x) - x| = 1/(8\pi)\sqrt{x}\ln^2 x$ for $x \geq 599$, cf. [23, 24]). Therefore, to find q we have to perform about $q/\ln 2$ group operations. Note that we get the same result even in the case $e(p_i, q) = \log_{p_i} q$, because $\sum_{p<q} \log_2 p_i \log_{p_i} q = \pi(q) \log_2 q = q/\ln 2$ as $q \to \infty$.

Example We continue the example from the previous section. By Table 5, 2^{64} group operations take about 6.6×10^{10} MIPS-years (similar to the computational work to factor a 1024 bit integer with the aid of the GNFS). If we assume that this amount of work is infeasible, then it is safe to select a 63 bit smoothness-bound. At the end of the next section we shall see that a smaller smoothness-bound is sufficient.

3.5 The smoothness probability of class numbers

The estimates in this section are based on the heuristics of Cohen and Lenstra [7, 6], although our derivation is not rigorous at all. A more rigorous derivation should be done as in [6]; this is work in progress, and we shall present the results in a future work. In this work we compare class numbers and ordinary integers with respect to smoothness, and we argue that under reasonable assumptions the probability to get a smooth class number of a random

fundamental discriminant is not much larger than the probability that a random integer is smooth.

Consider the set of all negative fundamental discriminants Δ such that $|\Delta| \leq N$ for some bound N. Based on the heuristics of Cohen and Lenstra we assume that, given an odd prime p much smaller than N and a positive integer i, the proportion of such discriminants satisfying $p^i \mid h(\Delta)$ (or the "probability" that $p^i \mid h(\Delta)$) is at most $1/p^i + 1/p^{i+1} = (1 + 1/p)/p^i$. (The conjectures of Cohen and Lenstra predict that for $N \to \infty$, the probability that $p \mid h(\Delta)$ converges to

$$1 - \prod_{j \geq 1} \left(1 - \frac{1}{p^j}\right) = \frac{1}{p} + \frac{1}{p^2} - \frac{1}{p^5} - \frac{1}{p^7} + \frac{1}{p^{12}} + \frac{1}{p^{15}} - \cdots$$

(see [6]). Our assumption for $i \geq 2$ is accordance with computational experiments.)

We cannot use similar heuristics for primes that are not small compared to N. However, we know by the Brauer-Siegel theorem that $\ln h(\Delta) \sim \ln \sqrt{|\Delta|}$ for $\Delta \to -\infty$, thus class numbers are usually not small themselves.

Which power of 2 divides $h(\Delta)$ depends on the factorization of Δ. As discussed in section 3.1.1, we will restrict to special discriminants in order to control the two-part of $h(\Delta)$. In extension to the heuristics of Cohen and Lenstra, we assume that such restrictions do not affect the probabilities discussed above.

For x uniformly chosen from a sufficiently large interval of integers, the probability that $p^i \mid x$ is only about $1/p^i$. Comparing this with the above estimates for class numbers, we obtain

$$\frac{\Pr\left(p^i \mid h(\Delta)\right)}{\Pr\left(p^i \mid x\right)} \leq 1 + \frac{1}{p}$$

for small odd primes, which suggests that it must be expected to occur more frequently for negative fundamental discriminants to have smooth class numbers than for uniformly chosen integers to be smooth. We will now argue, however, that this increase in smoothness does not imply that a significant proportion of class numbers will be smooth.

Let k be any odd smooth integer. We write k as $\prod_{p|k} p^{e_p(k)}$. If k is not so large that $k \mid h(\Delta)$ is actually impossible, then k will have only a few different prime factors. Thus, it is conceivable that the probabilities discussed above will be reasonably close to being statistically independent over the different p dividing k. Under this presumption, we obtain

$$\frac{\Pr\left(k \mid h(\Delta)\right)}{\Pr\left(k \mid x\right)} = \frac{\prod_{p|k} \Pr\left(p^{e_p(k)} \mid h(\Delta)\right)}{\prod_{p|k} \Pr\left(p^{e_p(k)} \mid x\right)} \leq \prod_{p|k} \left(1 + \frac{1}{p}\right) =: F_k \ .$$

We now want to estimate the maximum value that this product can take for k not exceeding the order of $\sqrt{|\Delta|}$ (as suggested by the Brauer-Siegel Theorem). For reaching the maximum, k obviously must be of the form $k = \prod_{p<t} p$, i.e. the product of the smallest primes up to some bound. We have $\prod_{p<t} p = e^t$ as t tends to infinity (e.g. see [22, Chap. 12]), i.e. $t \approx \ln k \approx \ln \sqrt{|\Delta|}$; and thus we estimate the maximum for F_k as

$$\prod_{p<t} \left(1 + \frac{1}{p}\right) \approx \prod_{p<\ln \sqrt{|\Delta|}} \left(1 + \frac{1}{p}\right) \approx \ln \ln \sqrt{|\Delta|} \ ,$$

where the latter approximation can be seen as follows: $(1 + 1/p) = (1 - 1/p^2)/(1 - 1/p)$, and $\prod_{p<t}(1 - 1/p) = e^{-\gamma}/\ln t + O(1/\ln^2 t)$ (Mertens' theorem, cf. [22, Chap. 12]), while $\prod_p(1 - 1/p^2) = 1/\zeta(2) = 6/\pi^2$, thus $\prod_{p<t}(1 + 1/p) = 6e^{\gamma}/\pi^2 \ln t \approx 1.08 \ln t$ as t tends to infinity.

Now if we choose $|\Delta|$ so large that random integers of the expected order of $h(\Delta)$ are smooth only with probability close to 0, then the modest maximum size of F_k indicates that the tendency of the class number towards having small factors does not mean it will be smooth with non-negligible probability.

Specifically, let $B = M^{1/u}$; then the probability that a random positive integer less than M is B-smooth is approximately $\rho(u)$, where ρ is Dickmann's ρ-function [14]. We arrive at an estimated probability of at most $\rho(u) \ln \ln M$ for the class number being B-smooth by requiring $M \approx \frac{h_{\mathsf{odd}}(\Delta)}{h(\Delta)} c\sqrt{|\Delta|}$ where $\frac{h(\Delta)}{h_{\mathsf{odd}}(\Delta)}$ is either 1 or 8 depending on how Δ is chosen (section 3.1.1) and where $c = 0.461559\ldots$ [5, Sect. 5.10.1]. I.e.,

$$|\Delta| \approx 2^2 B^{2u}$$

if $h(\Delta)$ is odd and

$$|\Delta| \approx 2^8 B^{2u}$$

if the even part of $h(\Delta)$ is 8. Note that if $|\Delta| < 2^{4600}$, then $\ln \ln \sqrt{|\Delta|} < 2^3$ so that $8\rho(u)$ is an upper bound for the probability estimate.

Assume that an attacker applies the algorithm from the preceding section to class groups of random discriminants of a certain length (chosen as described in Sect. 3.1.1). Further assume that he will spend at most W_{max} computational work for a single class group until he gives up, and that B is the smoothness-bound for which he can succeed with this amount of work. Then he can expect one case of success for an investment of computational work $W = W_{\mathsf{max}}/\Pr\big(h(\Delta) \text{ is } B\text{-smooth}\big)$. We will determine lower bounds for the size of Δ based on this attack scenario.

Recall that 1 MIPS-year is approximately equivalent to 2^{29} group operations. Let $W = 2^{64}$ group operations which, by Tables 1 and 5, is comparable to the expected computational work to factor a composite 1024 bit integer by the GNFS; then W is currently infeasible (see also [17] for extrapolations into the future). Let $W_{\mathsf{max}} = 2^{42}$ group operations (corresponding to a smoothness bound of approximately $2^{42}/\ln 2$, cf. section 3.4), which is comparable to the expected work to factor a 512 bit integer by the GNFS. Then a smoothness-probability of up to 2^{-22} is acceptable, thus we need u such that $\rho(u) \approx 8 \cdot 2^{-22}$, and this is satisfied by $u = 8$. Since $B \approx 2^{41.5}$, the discriminant should have at least 666 bits for case 1 of section 3.1.1, and at least 672 bits for case 2 of section 3.1.1.

If W_{max} is larger or if a smaller smoothness-probability is demanded, then the order of magnitude of the discriminant will increase accordingly. For instance, if we choose $\Pr\big(h(\Delta) \text{ is } B\text{-smooth}\big) = 2^{-30}$ with W_{max} (and hence B) as before, then $u = 9.6$, and thus the discriminant should have at least 799 (case 1) or 805 (case 2) bits.

4 Conclusion

Based on the investigation of several strategies to solve the CL-DLP and based on heuristic reasoning, we have shown how to select the discriminant such that the security of cryptosys-

tems based on class groups offer a comparable security as commonly used cryptosystems (such as RSA). In particular, we have shown that the size of the discriminant asymptotically depends only on index-calculus algorithms (see Table 4). Thus, since index-calculus algorithms for solving the Cl-DLP are asymptotically much slower than index-calculus algorithms to solve the IFP (such as the GNFS), the discriminant can be selected smaller than a RSA modulus.

In a future work we shall demonstrate the impact of this result on the efficiency and performance of IQC. As a further research project we would also like to replace the heuristic reasoning of Sect. 3.5 by a more rigorous reasoning.

References

[1] Biehl, I., Buchmann, J., Hamdy, S., and Meyer, A. Cryptographic Protocols Based on Intractability of Extracting Roots and Computing Discrete Logarithms. Tech. Rep. TI-1/00, Fachbereich Informatik, TU Darmstadt, 2000. `http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/` .

[2] Buchmann, J., and Paulus, S. A One Way Function Based on Ideal Arithmetic in Number Fields. In *Advances in Cryptology — CRYPTO '97* (1997), B. S. Kaliski, Ed., vol. 1294 of *Lecture Notes in Computer Science*, Springer–Verlag, pp. 385–394.

[3] Buchmann, J., and Williams, H. C. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology 1* (1988), 107–118.

[4] Buchmann, J., and Williams, H. C. Quadratic fields and cryptography. In *Number Theory and Cryptography*, J. H. Loxton, Ed., vol. 154 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1990, pp. 9–25.

[5] Cohen, H. *A Course in Computational Algebraic Number Theory*, vol. 138 of *Graduate Texts in Mathematics*. Springer–Verlag, 1995.

[6] Cohen, H., and Lenstra, Jr., H. W. Heuristics on class groups of number fields. In *Number Theory*, vol. 1068 of *Lecture Notes in Mathematics*. Springer–Verlag, 1983, pp. 33–62.

[7] Cohen, H., and Lenstra, Jr., H. W. Heuristics on class groups. In *Number Theory*, vol. 1052 of *Lecture Notes in Mathematics*. Springer–Verlag, 1984, pp. 26–36.

[8] Cox, D. A. *Primes of the form $x^2 + ny^2$*. John Wiley & Sons, 1989.

[9] Düllmann, S. *Ein Algorithmus zur Bestimmung der Klassengruppe positiv definiter binärer quadratischer Formen*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1991.

[10] Guillou, L. C., and Quisquater, J.-J. A Practical Zero-Knowledge Protocol Fitted To Security Microprocessors Minimizing Both Transmission and Memory. In *Advances in Cryptology — EUROCRYPT '88* (1988), C. G. Günther, Ed., vol. 330 of *Lecture Notes in Computer Science*, Springer–Verlag, pp. 123–128.

[11] Hafner, J. L., and McCurley, K. S. A rigorous subexponential algorithm for computation of class groups. *Journal of the American Mathematical Society 2* (1989), 837–850.

[12] HÜHNLEIN, D. Quadratic orders for NESSIE — Overview and parameter sizes of three public key families. Tech. Rep. TI-3/00, FB Informatik, TU Darmstadt, 2000. `http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/` .

[13] HÜHNLEIN, D., AND TAKAGI, T. Reducing logarithms in totally non-maximal imaginary quadratic orders to logarithms in finite field. Tech. Rep. TI-8/99, Fachbereich Informatik, TU Darmstadt, 1999. `http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/` .

[14] HUNTER, S., AND SORENSON, J. Approximating the Number of Integers Free of Large Prime Factors. *Mathematics of Computation 66*, 220 (1997), 1729–1741.

[15] JACOBSON, JR., M. J. *Subexponential Class Group Computation in Quadratic Orders*. PhD thesis, Fachbereich Informatik, TU Darmstadt, Darmstadt, Germany, 1999.

[16] KAPLAN, P. Sur le 2-groupe des classes d'idéaux des corps quadratiques. *J. reine angew. Math. 283/284* (1976), 313–363.

[17] LENSTRA, A. K., AND VERHEUL, E. R. Selecting Cryptographic Keysizes. H. Imai and Y. Zheng, Eds., vol. 1751 of *Lecture Notes in Computer Science*, Springer–Verlag, pp. 446–465. Full version available from `http://www.cryptosavvy.com/` .

[18] LiDIA — A C++ Library For Computational Number Theory. `http://www.informatik.tu-darmstadt.de/TI/LiDIA/` . The LiDIA Group.

[19] MENEZES, A. J., VAN OORSCHOT, P. C., AND VANSTONE, S. A. *Handbook of Applied Cryptography*. CRC Press, 1997.

[20] ODLYZKO, A. M. The Future of Integer Factorization. *CryptoBytes 1*, 2 (1995). `http://www.rsa.com/rsalabs/pubs/cryptobytes/` .

[21] POUPARD, G., AND STERN, J. Security Analysis of a Practical "on the fly" Authentication and Siganture Generation. In *Advances in Cryptology – EUROCRYPT '98* (1998), K. Nyberg, Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer–Verlag, pp. 422–436.

[22] ROSE, H. E. *A Course in Number Theory*, 2 ed. Oxford University Press, 1994.

[23] ROSSER, J. B., AND SCHOENFELD, L. Sharper bounds for the Chebyshev Functions $\theta(x)$ and $\psi(x)$. *Mathematics of Computation 29*, 129 (1975), 243–269.

[24] SCHOENFELD, L. Sharper bounds for the Chebyshev Functions $\theta(x)$ and $\psi(x)$, II. *Mathematics of Computation 30*, 134 (1976), 337–360.

[25] SILVERMAN, R. D. Exposing the Mythical MIPS Year. *IEEE Computer 32*, 8 (1999), 22–26.

[26] TE RIELE, H. J. J. Factorization of a 512-bits RSA key using the Number Field Sieve. Announcment on the Number Theory List (`NMBRTHRY@listserv.nodak.edu`), August 1999.

[27] VAN OORSCHOT, P. C., AND WIENER, M. J. Parallel Collusion Search with Cryptanalytic Applications. *Journal of Cryptology 12*, 1 (1999), 1–28.

[28] VOLLMER, U. Asymptotically Fast Discrete Logarithms in Quadratic Number Fields. Accepted for ANTS IV, 2000.