IQ-cryptography Report

January 24, 2018

Abstract

In this report the survey of IQ-cryptography algebra and algorithms is given. In particular the choice of secure parameters is discussed.

0.1 IQ-cryptography

The IQ-cryptography or quadratic number field cryptography (`QNFC`) is another cryptographic scheme based on the abelian group, the ideal class group of some imaginary quadratic extension. `QNFC` was proposed by Buchmann and Williams [1], [2]. The question whether `QNFC` is secure at all, and the choice of cryptographically secure key-sizes, if it is, requires the study of the difficulty of the discrete logarithm problem (`DLP`). The case of the probabilistic algorithms the analysis relies on the well-established Extended Riemann Hypothesis (ERH). The particularity of these groups consists in the fact that its subexponentially difficult to calculate the size of the groups, contrary to the elliptic case. The multiplication and factorization is more time consumming compared to the elliptic case. The key observation is that `IQ-RDSA` signature is eventually faster than a RSA signature. The crossover at a security level that is rather moderate (according to [7]). Since the efficiency of IQ arithmetic received comparably little attention in the past, it is reasonable to expect significant improvements in the future. These improvements will make IQ cryptosystems even more competitive to traditional cryptosystems. Jacobson [12] reports running times of less than an hour for the computation of the structure of class groups of random 40 digit discriminants, and running (CPU) times of less than 10 days for special 80 digit discriminants on a 296 MHz SUN UltraSPARC-II platform. The running time of his algorithms can be improved using the optimized linear algebra techniques of [13]. Vollmer [9] has presented an IQ-DL algorithm that assuming the extended Riemann hypothesis has running time bounded by $L_{\Delta}[\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1)]$. For a proof of this bound see [10]. An extension of this algorithm that also computes the class number and class group structure is shown to have the same complexity.

0.2 Basic definitions, algorithms and examples

It is convenient to start from lattices as a more intuitive object, later on pass to the ideals. Lattices correspond to forms. We begin by defining how to associate lattices to bases, points and forms. Another application of the lattices is in the domain of the Post Quantum cryptography where the search for the minimal length vector turns out to be a hard problem for a quantum computer, for the lattices of size $n : \mathbb{Z}^n$. In the case of forms, the lattices

are two-dimensional. There is an additional structure here that it is possible to multiply lattices on the elements of some ring, thus making the lattices into two dimensional modules. Let A is a two-dimensional commutative $\mathbb{R}-$algebra (i.e a module over $\mathbb{R}$, that is itself a commutative algebra) The following Lemma leads to classification of such algebras:

Lemma 1. *Let A be a two-dimensional commutative $\mathbb{R}-$algebra. Then exactly one of the following three statements holds.*

1. *There is an $\mathbb{R}-$basis $(1, i)$ of A with $i^2 = 1$.*
2. *There is an $\mathbb{R}-$basis $(1, i)$ of A with $i^2 = 0$.*
3. *There is an $\mathbb{R}-$basis $(1, i)$ of A with $i^2 = -1$.*

Proof. There is an $\mathbb{R}-$basis $(1, \alpha)$ of A and by two-dimensionality of A we have $\alpha^2 = x + y\alpha$ with $x, y \in \mathbb{R}$. This implies $(\alpha - (y/2))^2 = x + y^2/4$. Set

$$i = \begin{cases} \frac{\alpha - y/2}{\sqrt{|x+y^2/4|}} & x + y^2/4 \neq 0 \\ \alpha - y/2 & \text{otherwise} \end{cases}$$

Then $(1, i)$ is still an $\mathbb{R}-$basis of A and $i^2 \in 0, \pm 1$ as asserted. Assume that there is an $\mathbb{R}-$basis $(1, \alpha)$ of A with $\alpha^2 = 1$ and another $\mathbb{R}-$ basis $(1, \beta)$ of A with $\beta^2 = -1$. Then we can write $\alpha = x + y\beta$ with $x, y \in \mathbb{R}$. Hence,

$$1 = \alpha^2 = x^2 - y^2 + 2xy\beta. \tag{1}$$

It follows that $xy = 0$. Since α and 1 are linearly independent, this implies $x = 0$. By (1) we have $1 = -y^2$, a contradiction. In a similar way, it can be shown that A cannot have two $\mathbb{R}-$bases $(1, \alpha)$ and $(1, \beta)$ with $\alpha^2 = 1$ and $\beta^2 = 0$ or $\alpha^2 = -1$ and $\beta^2 = 0$. □

There are three types of algebras corresponding to the square $i^2 = j$, denote them by A_j. The algebra that is equivalent to $\mathbb{C}$ corresponds to the last case of Lemma 1. It follows as well that the only non-identical automorphism of A is the conjugation:

$$\sigma : A_j \longrightarrow A_j : x + i(j)y \longrightarrow x - i(j)y.$$

Definition 1. *Let $\alpha \in A$.*

1. *The norm of* α *is* $N(\alpha) = \alpha\sigma(\alpha)$.

2. *The trace of* α *is* $Tr(\alpha) = \alpha + \sigma(\alpha)$.

3. *The characteristic polynomial of* α *is* $c_\alpha(X) = X^2 - Tr(\alpha)X + N(\alpha)$.

Definition 2. *The discriminant of a pair* $(\alpha, \gamma) \in A^2$ *is*

$$\Delta(\alpha, \gamma) = (\sigma(\alpha)\gamma - \sigma(\gamma)\alpha)^2.$$

The discriminant of $\theta \in A$ *is* $\Delta(\theta) = \Delta(1, \theta)$.

Definition 3. *Binary quadratic for integer coefficients is a polynomial in two variables of degree* 2*:*

$$f(X, Y) = aX^2 + bXY + cY^2,$$

where $a, b, c \in \mathbb{Z}$. *We write* $f = (a, b, c)$, *and call* f *a form.*

Definition 4. *A form* $f(X, Y)$ *is positive definite if for any*

$$(X, Y) \in A^2 \wedge (X, Y) \neq (0, 0) \Rightarrow f(X, Y) > 0$$

Definition 5. *A form* f *with real coefficients is called irrational, if* $f(x, y) \neq 0$ *for all* $(x, y) \in \mathbb{Z}^2, (x, y) \neq (0, 0)$

Note that an integral form is irrational if it is irreducible in $\mathbb{Z}[X, Y]$. Also, any positive definite form is irrational.

Definition 6. *The discriminant of a form* $f = (a, b, c)$ *is* $f^2 = b^2 - 4ac$.

Definition 7. *The form* $f = (a, b, c)$ *is called normal if* $a < b \leq a$.

Definition 8. *The positive definite form* (a, b, c) *is called reduced if it is normal,* $a \leq c$, *and if* $b \geq 0$ *for* $a = c$.

Definition 9. *Let* r *be a real number. Then* $\lfloor r \rfloor$ *is the uniquely determined integer with*

$$0 \leq r - \lfloor r \rfloor < 1.$$

Also, $[r]$ *is the uniquely determined integer with*

$$1/2 \leq r - [r] < 1/2$$

If we choose

$$s = \left\lfloor \frac{a-b}{2b} \right\rfloor = \left\lceil \frac{-b}{2a} \right\rceil,$$

then we have

$$a < b + 2sa \leq a.$$

This choice of s minimizes the absolute value of b.

Definition 10. *By $\rho(f) = \rho(a,b,c)$ we denote the normalization of $(c,-b,a)$. We call ρ the reduction operator for positive definite forms.*

It is easy to check that

$$\rho(f) = (c, -b+2sc, cs^2 - bs + a)$$

The reduction algorithm is very simple. First, the form f is normalized. Then the algorithm proceeds iteratively. If f is reduced, then the algorithm returns f. Otherwise, f is replaced by $\rho(f)$. This is called a reduction step.

Algorithm 1. `rho`(f,T)

Input: *A positive definite form* $f = (a,b,c)$, $\quad T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \in \mathbb{Z}^{2\times 2}$.

Output: $(\rho(f), TU(f))$.

$s \leftarrow s(f)$

return $\left((c, -b+2sc, cs^2-bs+a), \begin{pmatrix} t_{12} & t_{11}+st_{12} \\ t_{22} & t_{21}+st_{22} \end{pmatrix}\right)$

Algorithm 2. `normalize`(f,T)

Input: *A positive definite form* $f = (a,b,c)$.

Output: *The normalization g of f and $U \in \Gamma$ such that $g = fU$.*

$s \leftarrow \lfloor (a-b)/(2a) \rfloor$

return $\left((a, b+2sa, as^2+bs+c), \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}\right)$

Algorithm 3. `reduce` (f)

Input: *A positive definite form* f.
Output: *A reduced form* g *and* $T \in GL(2,\mathbb{Z})$ *with* $fT = g$.
$(g,T) \leftarrow$ `normalize` *(f)*
while g *is not reduced* **do**
$(g,T) \leftarrow$ `rho` (g,T)
return (g,T)

0.2.1 Properties of reduced forms

Lemma 2. *if* (a,b,c) *is reduced, then* $a \leq \sqrt{|\Delta|/3}$

Proof. Assume that (a,b,c) is reduced. Then $|\Delta| = 4ac - b^2 \geq 4a^2 - a^2 = 3a^2$, which implies that $a \leq \sqrt{|\Delta|/3}$ □

Lemma 3. *If* f *is normal and* $a < \sqrt{|\Delta|}/2$*, then* f *is reduced.*

Proof. Let f be normal and $a < |\Delta|/2$. Since $\Delta < 0$ we have

$$c = \frac{b^2 + |\Delta|}{4a} \geq \frac{|\Delta|}{4a} > \frac{a^2}{a} = a$$

Thus f is reduced. □

Lemma 4. *if* f *is normal and* $a \geq \sqrt{|\Delta|}$*, then* $c \leq a/2$

Proof. Let f be normal. Then we have $b^2 \leq a^2$. Hence, it follows from $a \geq \sqrt{|\Delta|}$ that

$$c = \frac{b^2 + |\Delta|}{4a} \leq \frac{a^2 + a^2}{4a} = \frac{a}{2}.$$

□

Lemma 5. *Let* f *is normal but not reduced. If* $a < \sqrt{|\Delta|}$*, then* $\rho(f)$ *is reduced*

Proof. Assume that $a < \sqrt{|\Delta|}$. Since f is normal but not reduced we have either $a > c$ or ($a = c$ and $b < 0$). In the latter case $\rho(f)$ is obviously reduced. So assume that $a > c$. If $c < \sqrt{|\Delta|}/2$, then $\rho(f)$ is reduced by Lemma 3. Assume that $c \geq \sqrt{|\Delta|}/2$, that is $4c^2 \geq |\Delta|$. Since $|b| \leq a < \sqrt{|\Delta|}$ it follows

that $b^2/(4c^2) < 1$. Hence $|s(f)| \leq 1$. If $s(f) = 0$ then $\rho(f) = (c, -b, a)$ which is reduced since $c < a$. Assume that $|s(f)| = 1$. Then $\text{sign}(s(f)) = \text{sign}(b)$. Now $\rho(f) = (c, -b + 2s(f)c, a - |b| + c)$. If $a > |b|$, then $a - |b| + c > c$. Hence $\rho(f)$ is reduced. If $a = b$, hen $s(f) = 1$ since a is positive. So we have $\rho(f) = (c, -a + 2c, c)$. But $c \geq \sqrt{|\Delta|}/2 > a/2$, so $-a + 2c > 0$. This shows that $\rho(f)$ is reduced. □

Proposition 1. *The number of reduction steps performed by the reduction algorithm when applied to a positive definite form $f = (a, b, c)$ is at most*

$$\left\lfloor \log_2(a/\sqrt{|\Delta|}) \right\rfloor + 2.$$

Proof. In each reduction step, the form (a, b, c) is replaced by normalization of $(c, -b, a)$. If $a \geq \sqrt{|\Delta|}$ and if the resulting form is (a', b', c'), then, by Lemma 4, we have $a' = c \leq a/2$. This shows that after at most

$$\left\lfloor \log_2(a/\sqrt{|\Delta|}) \right\rfloor + 1$$

reduction steps the reducion algorithm finds a form (a, b, c) with $a < \sqrt{|\Delta|}$. From Lemma 5 it follows that plus another step is necessary to determine a reduced form. □

If we denote bit size of $f = (a, b, c)$ by $\text{size}\, f$ the it can be proven using the above results

Proposition 2. *The running time of Algorithm* `reduce` *when applied to a positive definite form f is $O((\text{size}\, f)^2)$.*

The consistency of the definition of form discriminant comes from

Definition 11. *Let $f = (a, b, c)$ be an irrational form, $\Delta = \Delta(f), j = \text{sign}\Delta$, and $i = i(j)$. Then $a \neq 0$ and we set*

$$B(f) = \left(a, \frac{b + i\sqrt{|\Delta|}}{2} \right)$$

and

$$\theta(f) = \frac{b + i\sqrt{|\Delta|}}{2a}$$

For an irrational form $f = (a, b, c)$ we have

$$\Delta(B(f)) = a^2\Delta(f), \quad \Delta(\theta(f)) = (1/a^2)\Delta(f).$$

Definition 12.

1. *With an $\mathbb{R}-$basis $B = (\alpha, \beta)$ of A we associate the lattice*

$$L(B) = \mathbb{Z}\alpha + \mathbb{Z}\gamma.$$

2. *With a point in $A\backslash\mathbb{R}$ we associate the lattice*

$$L(\theta) = L(1, \theta) = \mathbb{Z} + \mathbb{Z}\theta.$$

3. *With a form f we associate the lattice $L(f) = L(B(f))$.*

Example 1. *If $f = (1, 0, 1)$, then $L(f) = \mathbb{Z} + \mathbb{Z}\sqrt{-1}$. This is the lattice of Gaussian integers. If $f = (1, 1, 1)$, then $L(f) = \mathbb{Z} + \mathbb{Z}(1 + \sqrt{-3})/2$. This is the hexagonal lattice.*

The discriminant of a $\mathbb{Z}-$basis of a two-dimensional lattice L in A is an invariant of L. This justifies the following

Definition 13. *The discriminant of a two dimensional lattice L in A is the discriminant of any $\mathbb{Z}-$basis of L. It is denoted by $\Delta(L)$.*

0.2.2 Multiplicative Lattices

Let L, M, and K be additive subgroups of A. We define sum, product and quotient of those groups.

Definition 14.

1. *The sum of L and M is $L + M = \{\alpha + \beta : \alpha \in L, \beta \in M\}$.*

2. *The product of L and M is the additive subgroup of A generated by all products $\alpha\beta, \alpha \in L, \beta \in L$.*

3. *The quotient of L and M is $L : M = \{\alpha \in A : \alpha M \subset L\}$.*

Example 2. *Let* $L = 2\mathbb{Z} + \sqrt{2}\mathbb{Z}, M = 3\mathbb{Z} + \sqrt{-2}Z$. *We claim that*

$$L + M = K = \mathbb{Z} + \sqrt{-2}Z.$$

We have $L, M \subset K$ *and therefore* $L + M \subset K$. *To show that* $K \subset L + M$ *it suffices to prove that* $1 \in L + M$. *Since* $1 = 3 - 2$, *this is true.*

We present an example of two lattices whose product is a lattice.

Example 3. *Let* $L_1 = L(2, 2, 1) = 2\mathbb{Z} + (1 + \sqrt{-1})\mathbb{Z}$ *and* $L_2 = (5, 4, 1) = 5\mathbb{Z} + (2 + \sqrt{-1})\mathbb{Z}$. *Then*

$$L_1L_2 = \quad 10\mathbb{Z} + 2(2 + \sqrt{-1})\mathbb{Z} + 5(1 + \sqrt{-1})\mathbb{Z} + (1 + 3\sqrt{-1}\mathbb{Z}) =$$

$$10\mathbb{Z} + (3 + \sqrt{-1})\mathbb{Z} = L(10, 6, 1).$$

So the product of L_1 *and* L_2 *is the lattice* $L = L(10, 6, 1)$.

We have seen two lattices whose product is a lattice. In general, the product of two lattices is not a lattice, as we will see in the next example.

Example 4. *Let* $L_1 = \mathbb{Z} + \sqrt{-1}\mathbb{Z}$ *and* $L_2 = \mathbb{Z} + \sqrt{-2}\mathbb{Z}$. *Then*

$$L_1L_2 = \mathbb{Z} + \sqrt{-1}\mathbb{Z} + \sqrt{-2}\mathbb{Z} + \sqrt{2}\mathbb{Z}.$$

We show that L_1L_2 *is not a lattice. Assume that* L_1L_2 *is a lattice. Let* $a = \min(L_1L_2 \cap \mathbb{R} > 0)$. *Then* $1 = xa$ *and* $\sqrt{2} = ya$ *with* $x, y \in \mathbb{Z}$. *Hence,* $\sqrt{2} = y/x$. *This contradicts the irrationality of* $\sqrt{2}$.

To characterize the irrational lattices in A whose product is a lattice we introduce quadratic orders.

Definition 15. *A quadratic order is a two-dimensional lattice* $\mathcal{O} \in A$ *which is also a unitary subring of* A *(i.e* $1 \in \mathcal{O}$*).*

We characterize the quadratic orders. We recall that for an integer Δ, $\Delta \equiv 0, 1 \bmod 4$, there is exactly one reduced form $(1, b, c)$ of discriminant Δ, which is called the principal form of discriminant Δ.

Any quadratic order can be related to some principal form. Indeed if $\mathcal{O}$ is a quadratic order. Then $1 \in \mathcal{O}$. Since $\mathcal{O}$ is a ring and discrete as a point set, it follows that $1 \in \min(\mathbb{R} \cap \mathcal{O})$. Hence there is a $\mathbb{Z}-$basis $(1, \theta)$ of $\mathcal{O}$ of positive orientation. Since $\mathcal{O}$ is a ring, we have

$$\theta^2 = -b\theta - c, \text{ where } b, c \in \mathbb{Z}.$$

It follows that $\theta = \theta(1, b, c)$ and $\mathcal{O} = L(1, b, c)$.

Definition 16. *The class number $h(\Delta)$ is the number of proper equivalence classes of primitive integral forms of discriminant Δ.*

Let the prime form base $\mathcal{P}_l$, $l \in N$, be defined as follows:

$$P_l := \left\{ I_p \in \mathrm{Cl}(-\mathcal{D}) : p \text{ prime}, p \leq l, \left(\frac{-\mathcal{D}}{p}\right) = 1 \right\}.$$

Then assuming Extended Riemann Hypothesis it is possible to show

Theorem 1 (ERH)**.** *There is an absolute, effectively computable constant c_1 such that $\mathcal{P}_{c_1 \log^2 \mathcal{D}}$ generates $\mathrm{Cl}(-\mathcal{D})$.*

The class group of ideals of a quadratic imaginary extension $\mathcal{K} = \mathbb{Q}(\sqrt{-\mathcal{D}})$, denoted as $\mathrm{Cl}(\mathcal{O}_\mathcal{K})$, is used as an underlying abelian group for Diffie - Hellman cryptographic protocol. The discriminant $\Delta_\mathcal{K} = -\mathcal{D}$ is said to be maximal order if it is square free and

$$\Delta_\mathcal{K} \equiv 1, 0 \pmod 4.$$

IQ-cryptography restricts itself to imaginary extensions, where the class group of ideals is well defined. The class group is generated by the equivalence classes of ideals in the ring of algebraic integers $\mathcal{O}_\Delta$. There is one to one corrspondence between the binary quadratic forms and reduced ideals.

Definition 17. *It is said that the ideals are equivalent $I_1 \sim I_2$ if and only if there exist $\alpha, \beta \in \mathcal{O}_\Delta$ such that*

$$(\alpha) I_1 = (\beta) I_2$$

Each ideal can be represented in the form of a $\mathcal{O}_\Delta$ module:

$$I = q(\mathbb{Z}a + \mathbb{Z}(b + c\omega_\Delta)),$$

where

$$\omega_\Delta = (r - 1 + \sqrt{\Delta})/r, \quad r = \begin{cases} 2 & \text{if} \quad \Delta \equiv 1 \pmod 4 \\ 1 & \text{otherwise} \end{cases}$$

There is one to one correspondence between reduced ideals and positive primitive forms

$$f(x, y) = ax^2 + bxy + cy^2, \quad \gcd(a, b, c) = 1, \quad a > 0, \quad x, y \in \mathbb{Z}$$

of discriminant $\Delta_{\mathcal{K}} \equiv -\mathcal{D} = b^2 - 4ac$. The corresponding ideal has the following form

$$I = \left(a, \frac{b + \sqrt{-\mathcal{D}}}{2}\right)$$

Definition 18 (Principal and Proper Ideals)**.** *If R is an integral domain and I is an $R-$ideal, then I is called a principal $R-$ideal if there exists an element $\alpha \in I$ such that $I = (\alpha)$, where α is called a generator of I. If $I \neq R$, then I is called a proper ideal.*

Definition 19 (Prime ideal)**.** *If R is an integral domain, then a proper $R-$ideal P is called a prime $R-$ideal if it satisfies the property that whenever, $\alpha\beta \in P$, for $\alpha, \beta \in R$, then either $\alpha \in P$ or $\beta \in P$.*

Definition 20 (Product of Ideals)**.** *If R is an integral domain and I, J are $R-$ideals, then the product of I and J, denoted by IJ, is given by*

$$IJ = \left\{ r \in R : r = \sum_{j=1}^{n} \alpha_j \beta_j \quad \text{where} \quad n \in \mathbb{N}, \alpha_j \in I, \beta_j \in J \quad \text{for} \quad 1 \leq j \leq n \right\}$$

Multiplication Formula for Ideals in Quadratic Fields
Let $I_j = (a_j, (b_j + \sqrt{\Delta})/2)$, for $j = 1, 2$ be $\mathcal{O}_\Delta$ ideals, then

$$I_1 I_2 = (g)\left(a_3, \frac{b_3 + \sqrt{\Delta}}{2}\right),$$

where

$$a_3 = \frac{a_1 a_2}{g^2}, \text{ with } g = \gcd\left(a_1, a_2, \frac{b_1 + b_2}{2}\right),$$

and

$$b_3 \equiv \frac{1}{g}\left(\delta a_2 b_1 + \mu a_1 b_2 + \frac{\nu}{2}(b_1 b_2 + \Delta_{\mathcal{K}})\right) \pmod{2a_3},$$

where δ, μ and ν are determined by

$$\delta a_2 + \mu a_1 + \frac{\nu}{2}(b_1 + b_2) = g$$

0.3 Discrete logarithm problem

One of the generic methods to solve the Discrete logarithm problem (DLP) is to use the index calculus. The idea of index calculus algorithms is to reduce DLP to linear algebra. Let G be a cyclic group of order r, that we denote additively, and let P be a generator and Q be another element for which we want to compute the discrete logarithm. The simplest version of index calculus is as follows:

1. Define a subset $\mathcal{F}$ of G, called the factor base.
2. Collect relations:
   (a) Pick random integers a and b and compute $R = aP + bQ$;
   (b) Try to decompose R as a sum of elements of $\mathcal{F}$;
   (c) In case of success, $aP + bQ = \sum_{P_i \in \mathcal{F}} e_i P_i$, call it a relation, and store integers (a, b) and the vector (e_i) as a row of a matrix (the relation matrix);
   (d) Repeat the procedure until we have at least $\#\mathcal{F}$ relations.
3. Via linear algebra modulo r, compute a linear combination of the relations such that the right-hand-side vanishes; this leads to an equation $\lambda P + \mu Q = 0$ in G.
4. If is invertible modulo r, the discrete logarithm of Q is $\lambda/\mu \bmod r$.

For a given group G, the difficulty is to choose a factor base $\mathcal{F}$ that has the following properties, where the key difficulty resides in the decomposition step that must be fast and have a high success probability: The set $\mathcal{F}$ should not be too large, since we need to collect $\#\mathcal{F}$ relations. It should be the case that a large proportion of group elements can be written as a sum of elements in $\mathcal{F}$; otherwise Step 2b will fail too often. Given an arbitrary group element it should be efficient to decompose it as a sum in $\mathcal{F}$, or else decide that such a decomposition does not exist; otherwise each execution of Step 2b will take too long. In general, the decomposition of an element will involve only a small number of factor base elements. Therefore the matrix is usually quite sparse, and appropriate linear algebra algorithms must be used (see Chapter 3.4 [5]). The archetype of this algorithm is for the group $\mathbb{F}_p$ where p is a large prime. In that case, of course it would be easier to use

a multiplicative notation for the group law. One sets $\mathcal{F} = p_1, \ldots, p_k$ to be the set of the first k primes. One can consider any group element $R \in \mathcal{F}_p^*$ as an integer in the range $1 \leq R < p$ and try to factor it as a product of primes. Denoting $L_p(a)$ a subexponential function $\exp(\log(p)^a \log(\log(p))^{1-a})$ for some constant c, one takes $k = L_p(1/2)$. The set $\mathcal{F}$ has subexponential size and the probability that a random integer less than p can be written as a product of primes in $\mathcal{F}$ is $1/L_p(1/2)$. One therefore gets an algorithm with subexponential running time.

0.4 Index calculus, example

In this section we give an example of simple application of index calculus to the case of the multiplicative group of residues modulo prime number. Let p be a prime number, g a primitive root $\bmod p$, and $a \in 1, \ldots, p-1$. We want to solve the discrete logarithm problem

$$g^x \equiv a \bmod p.$$

We choose a bound B and determine the set

$$\mathcal{F}(B) = \{q \in \mathbb{P} : q \leq B\}.$$

This is the factor base. An integer b is called $B-$smooth if it has only prime factors in $\mathcal{F}(B)$.
Let $B = 15$. Then $\mathcal{F}(B) = \{2, 3, 5, 7, 11, 13\}$. The number 990 is $15-$smooth. Its prime factorization is

$$990 = 2 * 3^2 * 5 * 11.$$

We proceed in two steps. First, we compute the discrete logarithms of the factor base elements; that is, we solve

$$g^{x(q)} = q \bmod p$$

for all $q \in \mathcal{F}(B)$. Then we determine an exponent $y \in \{1, 2, \ldots, p-1\}$ such that $ag^y \bmod p$ is $B-$smooth. We obtain

$$ag^y \equiv \prod_{q \in \mathcal{F}(B)} q^{e(q)} \bmod p$$

with nonnegative exponents $e(q), q \in \mathcal{F}(B)$. Equations imply

$$ag^y \equiv \prod_{q\in\mathcal{F}(B)} q^{e(q)} \equiv \prod_{q\in\mathcal{F}(B)} g^{x(q)e(q)} \equiv g^{\sum_{q\in\mathcal{F}(\mathcal{B})} x(q)e(q)} \bmod p,$$

and hence

$$a \equiv g^{\sum_{q\in\mathcal{F}(B)} x(q)e(q)-y} \bmod p.$$

Therefore,

$$x = \left(\sum_{q\in\mathcal{F}(B)} x(q)e(q) - y\right) \bmod (p-1)$$

is the discrete logarithm for which we were looking. To compute the discrete logarithms of the factor base elements, we choose random numbers $z \in 1, ..., p-1$ and compute $g^z \bmod p$. We check whether those numbers are $B-$smooth. If they are, we compute the decomposition

$$g^z \bmod p = \prod_{q\in\mathcal{F}(B)} q^{f(q,z)}$$

Each exponent vector $(f(q,z))_{q\in\mathcal{F}(B)}$ is called a relation. We choose $p = 2027, g = 2$ and determine relations for the factor base $2, 3, 5, 7, 11$. We obtain

$$\begin{array}{rcrcl} 3*11 & \equiv & 33 & \equiv & 21593 \bmod 2027 \\ 5*7*11 & \equiv & 385 & \equiv & 2983 \bmod 2027 \\ 27*11 & \equiv & 1408 & \equiv & 21318 \bmod 2027 \\ 32*7 & \equiv & 63 & \equiv & 2293 \bmod 2027 \\ 26*52 & \equiv & 1600 & \equiv & 21918 \bmod 2027. \end{array}$$

If we have found as many relations as there are factor base elements, then we try to find the discrete logarithms by solving a linear system.

$$g^z \equiv \prod_{q\in\mathcal{F}(B)} q^{f(q,z)} \equiv \prod_{q\in calF(B)} g^{x(q)f(q,z)} \equiv g^{\sum_{q\in\mathcal{F}(B)} x(q)f(q,z)} \bmod p$$

This implies

$$z \equiv \sum_{q\in\mathcal{F}(B)} x(q)f(q,z) \bmod (p-1)$$

for all z, so each relation yields one linear congruence.

It can be shown that the index calculus algorithm that was described has subexponential running time

$$L_p[1/2, c + o(1)],$$

where the constant c depends on the technical realization of the algorithm; for example, on the complexity of the algorithm for solving the linear system.

0.5 The imaginary quadratic case, algorithms

0.5.1 The factor base

Our algorithm seeks to compute the relations lattice of a large set generating the class group. These generators are given by representatives from a large set $\mathcal{F}$ of ideals. The algorithm obtains individual relations by finding two exponent vectors $\mathbf{v}$ and $\mathbf{w}$ such that $[\mathcal{F}]^{\mathbf{v}} = [\mathcal{F}]^{\mathbf{w}}$. The first vector, $\mathbf{v}$, is chosen randomly so that $[\mathcal{F}]^{\mathbf{v}}$ is a random element of the class group in a sense to be made precise later. The second vector $\mathbf{w}$ is obtained by finding a reduced representative of $[\mathcal{F}]^{\mathbf{v}}$ and factor it over $\mathcal{F}$ using its factorization into a power product of prime ideals. This is successful if all prime ideals occuring in the factorization actually are elements of $\mathcal{F}$. Thus, depending on the proportion of prime ideals in $\mathcal{F}$ within the set of all prime ideals occuring in the factorization of reduced ideals, the process may have to be repeated many times before it will yield a relation. Since $\mathcal{F}$ needs to contain the prime ideals which the found reduced ideals are factored into, it is called a factor base. We will choose our factor base to contain the set $\mathcal{F}_z$ of all prime ideals $\mathfrak{p}(\Delta, p)$ and their conjugates for which p is a prime number with $\left(\frac{\Delta}{p}\right) = 1$ and

$$p \leq L_{|\Delta|}\left[\frac{1}{2}, z\right]$$

for some positive real number

$$z \leq 1$$

that is specified later. Depending on the problem we will want to solve, $\mathcal{F}$ will just contain the specified prime ideals, or one or two ideals in addition to them. Denote by f the cardinality of the factor base. By the prime number theorem there will be aproximately $L_{|\Delta|}[1/2, z]/(z\sqrt{\log|\Delta| \log\log|\Delta|})$ prime

ideals in $\mathcal{F}$. Thus, in all cases f will be in $L_{|\Delta|}[1/2, z + o(1)]$.
We order $\mathcal{F}$ in some way and write

$$\mathcal{F} = (\mathfrak{p}_1, \ldots, \mathfrak{p}_f).$$

All but at most two of the ideals $\mathfrak{p}_i$ are prime. We need $[\mathcal{F}]$ to generate the class group. We will assume that the Extended Riemann Hypothesis (ERH) holds and apply Proposition. This proposition says that $[\mathcal{F}]$ will generate the class group if $|\Delta|$ is sufficiently large. Comparing $c_3(\Delta)$ with $L_{|\Delta|}[1/2, z]$, we see that this is certainly the case if $\Delta < 157$ or $\Delta > 41$ which we will subsequently assume. We call an $\mathcal{O}-$ideal $\mathcal{F}-$smooth if all the factors in the prime ideal factorization of $\mathfrak{a}$ are in $\mathcal{F}$.

Algorithm 4. `kronecker`(m, n)

Input: *Integers m and n*
Output: $\left(\frac{m}{n}\right)$

if $2 \mid n$ *and* $2 \mid m$ **then** *return 0.*
if $\text{sign}(m) = \text{sign}(n) = -1$ **then** $j \leftarrow -1$ **else** $j \leftarrow 1$.
$m \leftarrow |m|, n \leftarrow |n|$.
while n *is even* **do**
  if $m \equiv 3, 5 \pmod 8$ **then** $j \leftarrow -j$
  $n \leftarrow n/2$
$m \leftarrow m \bmod n$
while $m \neq 0$ **do**
  while m *is even* **do**
    if $n \equiv 3, 5 \pmod 8$ **then** $j \leftarrow -j$
    $m \leftarrow m/2$
  if $m \equiv 3 \pmod 4$ *and* $n \equiv 3 \pmod 4$ **then** $j \leftarrow -j$
  interchange m and n
  $m \leftarrow m \bmod n$
if $n = 1$ **then** *return j*
else *return 0*

Algorithm 5. `numberOfPrimeForms`(Δ, p)

Input: *A discriminant* Δ *and a prime number* p
Output: $R(\Delta, p)$
return $\left(\frac{\Delta}{p}\right) + 1$

Algorithm 6. `primeForm`(Δ, p)

Input: *A discriminant* Δ *and a prime number* p *with* $R(\Delta, p) > 0$
Output: *The form* $(p, b(\Delta, p), c(\Delta, p))$
$b \leftarrow$ `sqrtMod4P`(Δ, p)
return $(p, b, (b^2 - \Delta)/(4p))$

Tonelli algorithm of calculating square root modulo prime:

Algorithm 7. `sqrtModP`(r, p)

Input: *A prime* p *and a square* r *modulo* p
Output: *A square root* s *of* r *modulo* p *or* `nil`

$m \leftarrow p - 1, t \leftarrow 0$
while m *is even* **do**
  $t \leftarrow t + 1$
  $m \leftarrow m/2$
Choose a random element $c \in \{1, \ldots, p-1\}$
if $\left(\frac{c}{p}\right) = 1$ **then** *return* `nil`
$r_m \leftarrow r^m \bmod p$
$c_m \leftarrow c^{-m} \bmod p$
$e \leftarrow 0, i \leftarrow 0$
while $r_m \neq 1$ **do**
  $i \leftarrow i + 1$
  $c_m \leftarrow c_m^2$
  if $r_m^{2^{t-i-1}} \bmod p \neq 1$ **then**
    $e \leftarrow e + 2^i$
    $r_m \leftarrow r_m c_m \bmod p$
$a \leftarrow rc^{-e} \bmod p$

return $c^{e/2}a^{(m+1)/2} \bmod p$

Proposition 3. *The algorithm* `sqrtModP` *has the following properties. It fails with probability* $1/2$*. If it does not fail, then it returns a square root of* $r \bmod p$ *provided that* r *is a square* $\bmod p$*. Its running time is* $O((\log p)^4)$*.*

Corollary 1. *Algorithm* `primeForm` *has success probability* $1/2$ *and running time* $O(size(\Delta)\text{size}(p) + (\text{size}(p))^4)$*.*

Algorithm 8. `factorBase`(Δ, z)

Input: *The discriminant* Δ*, the parameter* z*.*
Output: *The factor base* $\mathcal{F}_z$ *or* `nil`*.*
Set $L \leftarrow L_{|\Delta|}[1/2, z], k \leftarrow \lceil \log L \rceil$
Set $\mathcal{F} \leftarrow \emptyset$
for *all primes* $p < L$ **do**
if `kronecker`$(\Delta, p) = 1$ **then**
$i \leftarrow 0$
repeat
$i \leftarrow i + 1$
$g \leftarrow$`primeForm`(Δ, p)
until $i > 2k$ *or* $g \neq$`nil`
if $f \neq$ `nil` **then**
$\mathcal{F} \leftarrow \mathcal{F} \cup \{L(g), L(g)^\sigma\}$
else
Return `nil`
Return $\mathcal{F}$

Lemma 6. *Algorithm* `factorBase` *succeeds with probability exceeding* $1/4$*.*

Proof. Corollary 1 implies that Algorithm factorBase succeeds with probability larger than

$$(1 - 2^{-k})^L.$$

This probability larger than $1/4$ since $2k > \log L$ as if follows from

$$(1 - (1 - p)^l)^f \geq \frac{1}{4}, \text{where } pl > \log f, f > 1, 0 < p \leq 1,\ l \in \mathbb{N}.$$

□

Lemma 7. *If $z \leq 1$, then $f < 8|\Delta|/h_\Delta + 1$.*

0.5.2 Random relations

The calculation of the random relations is done by applying `randomRelation` algorithm. That algorithm selects an exponent vector $\mathbf{v} \in \mathbb{Z}^f_{0\ldots|\Delta|-1}$ randomly with the uniform distribution. It calculates the reduced ideal $\mathfrak{a}$ in the ideal class $[\mathcal{F}^{\mathbf{v}+\mathbf{w}}]$ where $\mathbf{w}$ is some fixed offset vector that is also input for `randomRelation`. If the reduced ideal happens to be $\mathcal{F}-$smooth, that is,

$$\mathfrak{a} = \mathcal{F}^{\mathbf{a}}$$

for some $\mathbf{a} \in \mathbb{Z}^f$, then

$$[\mathcal{F}^{\mathbf{v}+\mathbf{w}}] = [\mathfrak{a}] = [\mathcal{F}^{\mathbf{a}}].$$

Hence, the relation

$$\mathbf{z} = \mathbf{a} - \mathbf{w} - \mathbf{v} \in L([\mathcal{F}])$$

is found. Here is the algorithm.

Algorithm 9. `randomRelation` $(\Delta, \mathcal{F}, \mathbf{w})$

Input: *The discriminant Δ, the factor base $\mathcal{F}$ of length f, the offset vector $\mathbf{w} \in \mathbb{Z}^f$.*
Output: *A relation $\mathbf{z}$ for $[F]$ or* `nil`.
Select $\mathbf{v} \in \mathbb{Z}^f_{0\ldots|\Delta|-1}$ uniformly at random.
Calculate the reduced ideal $\mathfrak{a}$ in $[\mathcal{F}^{\mathbf{v}+\mathbf{w}}]$.
if $\mathfrak{a} = \mathcal{F}^{\mathbf{a}}$ *with* $\mathbf{a} \in \mathbb{Z}^f$ **then**
    Return $\mathbf{z} = \mathbf{v} + \mathbf{w} - \mathbf{a}$.
else
    Return `nil`

We explain how Algorithm `randomRelation` decides whether the reduced ideal $\mathfrak{a}$ factors over the factor base $\mathcal{F}$. Let $\mathfrak{a} = L(a, b, c)$ with a reduced form (a, b, c). We know from Proposition 8.6.11 that a factors over $\mathcal{F}$ if and only if $\mathfrak{a}$ factors into the norms of the prime ideals in $\mathcal{F}$. Proposition 8.6.11 also tells

---

Algorithm 10. `fullRank`$(\Delta, \mathcal{F}, z)$

Input: *Discriminant* Δ*, factor base* $\mathcal{F}$*, parameter* z*.*
Output: `nil` *or relations* $(\mathbf{z}_1, d_1), \ldots, (\mathbf{z}_f, d_f) \in \tilde{L}(\mathcal{F})$ *such that the matrix* $(\mathbf{z}_1, \ldots, \mathbf{z}_f)$ *is strictly diagonally dominant.*
$l = \lceil (\log f)/p(\Delta, z) \rceil$
for $i = 1, 2, \ldots, f$ **do**
    $j \leftarrow 0$
    repeat
        $j \leftarrow j + 1$
        $(z_i, d_i) \leftarrow$ `randomRelation`$(\Delta, \mathcal{F}, B_1(\Delta)\mathbf{e}_i)$
until $j = l$ *or* $z_i =$ `nil`
if $z_i =$ `nil` **then**
    return `nil`*.*
return $(\mathbf{z}_1, d_1), \ldots, (\mathbf{z}_f, d_f)$*.*

Algorithm 11. `relationLattice`$(\Delta, \mathcal{F}, z)$

Input: *The discriminant* Δ*, the factor base* $\mathcal{F}$ *with cardinality* f*, the parameter* z*.*
Output: `nil` *or relations* $\mathbf{z}_1, \ldots, \mathbf{z}_N \in L[\mathcal{F}]$*.*
$N \leftarrow 1, i \leftarrow 0, k \leftarrow \lceil f \log_2 B_2(\Delta) \rceil, n \leftarrow \lceil p_{new}(\Delta, z) \log k \rceil, l \leftarrow \lceil (\log kn)/p(\Delta, z) \rceil$
repeat
    $i \leftarrow i + 1$
    $z_{N+1} \leftarrow$ `randomRelation`$(\Delta, \mathcal{F}, 0)$
if $(\mathbf{z}_{N+1}, d_{N+1}) \neq$ `nil` *then*
    $N \leftarrow N + 1$
until $i \geq kln$ *or* $N \geq kn$
if $N < kn$ **then**
    return `nil`
else

return $(\mathbf{z}_1, d_1), \ldots, (\mathbf{z}_N, d_N)$

0.5.3 Fast Discrete Logarithm

Following [9] we describe the fast DL-algorithm The following algorithm does not require calculating the determinant of the relation matrix, i.e the size of $\mathrm{Cl}(-\mathcal{D})$. It has the running time

$$L_{|\Delta|}\left[\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1)\right].$$

Thus the algorithm does not certify that the DL problem is unsolvable, it just finds $[g]^l = [h]$.

Algorithm 12. DL-*algorithm in* $\mathrm{Cl}(-\mathcal{D})$

Input: *Discriminant* $-\mathcal{D}$ *of an imaginary quadratic field* $\mathcal{K}$,
two form classes $[g], [h] \in \mathrm{Cl}(-\mathcal{D})$, *error probability* ϵ
Output: *either natural* l *such that* $[g]^l = [h]$ *or* UnDef,
meaning that with probability $1 - \epsilon$ *there is no such* l.

IqDL$(-\mathcal{D}, g, h)$

1. *Construct the factor base* $\mathcal{F}$:
   $\mathcal{F} := \left\{[f] | f = (p, b, \cdot), p \in \mathcal{P}_{-\mathcal{D}}, p < L_d\left(\frac{1}{2}, \frac{1}{\sqrt{8}}\right)\right\}$
2. *Construct the generating set* $\mathcal{G}$:
   $\mathcal{G} := \left\{[f] | f = (p, b, \cdot), p < 6\log^2 \mathcal{D}\right\}$
3. *Construct the extended factor base* $\mathcal{E} := \mathcal{F} \cup \mathcal{G} \cup \{g, h\}$
4. **foreach** $f \in \mathcal{E}$
   $v^{(f)}$:= IqRelation$(f, 2n\mathcal{D}, \mathcal{G} \cup \{f, g\}, n^2\mathcal{D})$
5. **for** $i := 1$ **to** $3n\log\mathcal{D} - 3\log\epsilon$
   $v^{(i)}$:=IqRelation$(1, 0, \mathcal{E}, \mathcal{D}^2))$

6. *Collect relations* $v^{(i)}$ *and* $v^{(f)}$ *intp matrix* $A := \left(\frac{a}{A'}\right)$ *with first row* a *containing exponents of* g

7. `DiophantineSolver`$(A', e_1, \epsilon/2) =: (y, \mathcal{D})$

8. **if** $A'y = e_1$ **then return** $l := a \cdot y$

9. **else return** `UnDef`

We have the obvious group homomorphism

$$\phi : \mathbb{Z}^{\mathcal{E}} \to \mathrm{Cl}(-\mathcal{D}) \quad : \quad (e_f)_{f\in\mathcal{E}} \longrightarrow \prod_{f\in calE} [f]^{e_f}.$$

Its kernel Λ (i.e $\phi(Ker) = \{1\}1$) is a sub-lattice of full rank since $\mathrm{Cl}(-\mathcal{D})$. Here let A be the matrix whose column vectors are the $v \in \mathcal{H}$. We may arrange the rows of these vectors in such a way that the entries corresponding to the exponents of g and h appear in the first and second row, respectively. Then `DL` problem is solvable if and only if Λ contains some vector of the form $(l, 1, 0, \ldots, 0)^T$. Due to our assumption that $\mathcal{H}$ generates Λ this happens in turn iff

$$A'y = (1, 0, \cdots 0)^T =^{\mathrm{def}} e_1$$

is solvable, where A' is obtained from A by striking out the first row a. If y is a solution of the Diophantine linear system, then we have $[g]^l = [h]$ for $l = a \cdot y$ since $(l, 1, 0, \cdots, 0)^T \in \Lambda$.

Algorithm 13. *Generation of relations*

Input: *form* f*, exponent* u*, set of generators* $\mathcal{H}$*, radius* r
Output: *relation* $v = (v_e)_e \in \mathcal{E}$ *such that* $|v_f - a| < \log \mathcal{D}$*, and for* $e \neq f$ $|v_e| < r + \log \mathcal{D}$ *if* $e \in \mathcal{H}$*, or else* $|v_e| < \log \mathcal{D}$

`IqRelation`*(f, a, H, r)*

1. **repeat**

2. *Draw random* $(u)_e \in \mathcal{E}$ *from* $\mathbb{N}^{\mathcal{H}}_{<r}$ *with the uniform distribution*

3. *Let* $f' = (a, b, c)$ *be the reduced form in the class* $f^u \prod_{e\in\mathcal{H}} e^{u_e}$.

4. **until** *attempt to factor a with Algorithm 7.2 from [6] is successful where we choose* $y := L_{\mathcal{D}}[\frac{1}{2}, \frac{1}{\sqrt{8}}]$ *as upper bound for divisors of a.*

5. *Find with method (2.8) of [6]* $(t_e)_{e\in\mathcal{F}}$ *such that*

$$(a, b, c) = \prod_{e\in\mathcal{F}} e^{t_e},$$

and let $t_e = 0$ *for* $e \in \mathcal{E}\backslash\mathcal{F}$.

6. **return** $(s_e)_{e\in\mathcal{F}}$, *where*

$$s_e := \begin{cases} u + u_e - t_e & \text{if } e = f \\ u_e - t_e & \text{if } e \in \mathcal{H}, e \neq f \\ -t_e & \text{if } e \in \mathcal{E}\backslash\mathcal{H} \end{cases}$$

0.6 Choice of security parameters

Primitives for the Selection of a Class Group is first step in any $IQ-$scheme is to choose a particular class group to use. Any discriminant determines a unique class group. Therefore, we represent a class group by the discriminant Δ of the respective order. We give an algorithm for generating an appropriate group under the name `IQ`- PI. We assume that the implementer can check integers for primality. Choosing a new class group will usually be done very infrequently, and in most cases speed will not be an issue. Technically it is sufficient to choose one `IQ`-PI method once and forever.

Algorithm 14. `IQ-PI`

Input: *A bitlength* $l_\Delta > 3$, *Seq : normal sequence of numbers, i.e having uniform probability of subsequences* $1/2^{l_\Delta}$.
Output: *A fundamental imaginary quadratic discriminant* Δ *of length* l_Δ.

1. **repeat**

2. $p \leftarrow^{\text{rnd}} Seq$

3. $p \leftarrow 4p + 3$

4. **until** *p is prime*

5. $\Delta \leftarrow -p$

6. **return** Δ

The base point ideal is taken as random prime ideal $I = (p, b)$ where $p \in Seq$

Bibliography

[1] Buchmann, J. and Hugh C. W. A key-exchange sy- stem based on imaginary quadratic fields. Journal of Cryptology, 1(2):107118, 1988.

[2] Buchmann J. and Hugh C. W. A key exchange system based on real quadratic fields. extended abstract. In Gilles Brassard, editor, Advances in Cryptology CRYPTO 89, volume 435 of Lecture Notes in Computer Science, pages 335343. Springer-Verlag, 1990.

[3] Buchmann, J.: "Introduction to Cryptography", 2nd ed, 2004, Springer.

[4] Buchmann, J., Vollmer, U.: "Binary Quadratic Forms, An Algorithmic Approach", 2007, Springer.

[5] Joux, A.: Algorithmic cryptanalysis. Chapman & Hall/CRC (2009)

[6] Lenstra H.W. and Pomerance C.: A rigorous time bound for factoring integers. J. Amer. Math. Soc. , 5:483-516, 1992.

[7] Lenstra, A.K., and Verhueul, E.R.: Selecting cryptographic keysizes. In Practice and Theory in Public Key Cryptography, PKC 2000 (2000), H. Imai and Y. Zheng, eds., vol. 1751 of Lecture Notes in Computer Science, Springer-Verlag, pp. 446-465.

[8] Mollin, R. Advanced Number Theory with Applications Boca Raton, London, New York: Chapman&Hall/Crc Press, Taylor and Francis Group, 2010. 476. Print.

[9] Vollmer, U., Asymptotically fast discrete logarithms in quadratic number fields, Algorithmic Number Theory, ANTS-IV (Wieb Bosma, ed.), Lecture Notes in Computer Science, vol. 1838, Springer-Verlag, 2000, pp. 581594.

[10] Vollmer, U.: Invariant and discrete logarithm computation in quadratic orders, Ph.D. thesis, Technische Universitat Darmstadt, Fachbereich Informatik, 2003

[11] M. Seysen. A probablistic factorization algorithm with quadratic forms of negative discriminant. Math. Comp. , 48:757-780, 1987.

[12] Jacobson, M.:, Subexponential class group computation in quadratic orders, Ph.D. thesis, Technische Universitat Darmstadt, Fachbereich Informatik, Darmstadt, Germany, 1999.

[13] Giesbrecht M., Jacobson, M., and torjohann, A.: Algorithms for large integer matrix problems, Applied algebra, algebraic algorithms and error-correcting codes (Melbourne, 2001), Lecture Notes in Computer Science, vol. 2227, Springer, Berlin, 2001, pp. 297307.