Journal de Théorie des Nombres
de Bordeaux **12** (2000), 293–307

Cryptography based on number fields with large regulator

par Johannes BUCHMANN, Markus MAURER
et Bodo MÖLLER

Résumé. Nous introduisons une variante du protocole de signature et d'identification de Fiat-Shamir, basée sur la difficulté pratique qu'il y a à calculer des générateurs des idéaux principaux dans les corps de nombres. Nous montrons en outre comment utiliser les heuristiques de Cohen-Lenstra-Martinet pour les groupes de classes dans le but de construire des corps de nombres dans lesquels le calcul de générateurs des idéaux principaux est encore hors d'atteinte.

Abstract. We explain a variant of the Fiat-Shamir identification and signature protocol that is based on the intractability of computing generators of principal ideals in algebraic number fields. We also show how to use the Cohen-Lenstra-Martinet heuristics for class groups to construct number fields in which computing generators of principal ideals is intractable.

1. Introduction

The security of public key cryptosystems is based on the intractability of computational problems in mathematics and in particular in number theory. Examples are the problems of factoring integers or computing discrete logarithms in certain finite abelian groups (see [24]). However, there is currently no such problem whose computational difficulty can be proved. On the contrary: Experience with the factoring problem shows that unexpected breakthroughs are always possible. To guarantee that public key cryptography is possible even if the currently used systems are broken, it is necessary to identify alternative computational problems that can be used as the basis of public key schemes.

In this paper, we consider the *principal ideal problem (PIP)*: Let O be an order of an algebraic number field F. Given a principal O-ideal I, find a generator of that ideal, i.e. an element $\alpha \in F$ such that $I = \alpha O$. We

Manuscrit reçu le 17 avril 2000.

show how to use the heuristics of Cohen, Lenstra, and Martinet ([13], [14], [15], and [16]) for class groups of algebraic number fields to generate orders for which PIP is intractable, and we present PIP-FS, a variant of the Fiat-Shamir identification and signature scheme [17] whose security is based on PIP. We describe an implementation of PIP-FS in real quadratic fields, we discuss its security, and we present timings that show that PIP-FS has the potential to become practical.

PIP is a special case of the number field discrete logarithm problem (NFDL), which was introduced in [8]. There, it has been suggested to develop cryptographic primitives based on NFDL. For number fields with large class number, a new scheme has been presented in [2]; also some previously known cryptographic schemes such as [18] and [26] can be adopted to the class group of a number field since they do not require knowledge of the group order (cf. [2] and [19]). A bit commitment and an oblivious transfer scheme for real-quadratic number fields have been described in [3]. Here, we present first identification and signature schemes for number fields with large regulators. We show how to generate such fields that are suitable for cryptographic applications.

This paper is organized as follows: In Section 2, we present a general version of the Fiat-Shamir protocol (FS). In Section 3, we explain PIP-FS, an FS variant based on PIP in number fields. A detailed description of the implementation of PIP-FS in real quadratic number fields is given in Section 4.

2. Fiat-Shamir identification

In this section, we present a fairly general version of the Fiat-Shamir identification protocol (FS). (For generalizations of the original Fiat-Shamir scheme [17], see also [12] and [11].)

The goal of the FS protocol is that one party, called the *prover*, convinces the other party, called the *verifier*, of his knowledge of a private key without revealing any relevant information concerning that private key. We will also explain how a digital signature scheme can be obtained from this protocol.

In the setup phase of the protocol, the prover and the verifier agree on two abelian groups G and H, on a homomorphism $\varphi : G \to H$, and on a positive integer k. The prover selects k group elements $g_i \in G$, $1 \leq i \leq k$. The sequence $(g_1, \ldots, g_k)$ is his private key. He then computes $h_i = \varphi(g_i)$, $1 \leq i \leq k$. The sequence $(h_1, \ldots, h_k)$ is his public key.

(In the original Fiat-Shamir protocol, a key issuing center is responsible for selecting G, H, and φ such that the center can efficiently invert φ using certain additional information, which must be kept secret. Then the components of any prover's public key can be derived from a description of the prover containing, for example, name and address information by applying a hash function; the key issuing center can compute an appropriate private key and pass it to the prover. We note that our variant of the Fiat-Shamir protocol does not have this property. Instead, public keys must be explicitly given, as e.g. in Schnorr's identification scheme [28].)

The FS identification protocol works as follows:

1. (Commitment and Witness) The prover randomly selects a *commitment* $g \in G$ and computes the *witness* $h = \varphi(g)$. The prover sends the witness h to the verifier.
2. (Challenge) The verifier selects a *challenge* $e \in \{0,1\}^k$ and sends it to the prover.
3. (Response) The prover computes the *response* $r = g \prod_{i=1}^k g_i^{e_i}$ and sends it to the verifier.
4. (Verification) The verifier checks whether $h \prod_{i=1}^k h_i^{e_i} = \varphi(r)$.

Clearly, a prover who knows the private key can convince the verifier of his identity.

Assume that φ has the following one way property: Without knowledge of the private key, it is intractable to compute, given $(f_1, \ldots, f_k) \in \{0, \pm 1\}^k$ where at least one f_i is not 0, an s with $\varphi(s) = \prod_{i=1}^k h_i^{f_i}$. We show that the probability to detect that a prover does not know the private key is at least $1 - 1/2^k$. To increase the probability, the basic protocol can be repeated several times.

If the prover is able to give the correct answer for two different challenges $(e_1, \ldots, e_k)$ and $(e'_1, \ldots, e'_k)$, then he knows $r, r' \in G$ such that $\varphi(r) = h \prod_{i=1}^k h_i^{e_i}$ and $\varphi(r') = h \prod_{i=1}^k h_i^{e'_i}$. This implies that he can compute $s = r' r^{-1}$ such that $\varphi(s) = \prod_{i=1}^k h_i^{e'_i - e_i}$. Note that $e'_i - e_i \in \{0, \pm 1\}$. By assumption, computing s without the knowledge of the private key is intractable. Therefore, a cheating prover cannot know the correct answer for two different challenges, and the probability for him to be detected is at least $1 - 1/2^k$.

When we explain our variant PIP-FS in Section 3, we will also show that the verifier or an observer are not able to derive the private key from information transmitted during the protocol.

The FS identification protocol is efficient if multiplication in the groups G and H can be performed efficiently and if the homomorphism φ can be computed efficiently.

In the following way, the FS identification protocol can be transformed into a signature protocol. Suppose a document d is to be signed. The signer selects a commitment g and computes the witness h as in the above protocol. To generate the challenge, the signer uses a cryptographic hash function f (see [24]). He computes $f(d \circ h)$ where $\circ$ is concatenation and d, h are identified with the bit strings by which they are represented. The challenge is the sequence of the first k bits of the hash value. The signature consists of the witness and the response. The verifier computes the challenge from the witness and the document and proceeds as in the FS protocol.

3. PIP-FS

We explain PIP-FS, our FS variant that is based on the intractability of solving the principal ideal problem in number fields. Let F be an algebraic number field and let O be an order of F. In the Fiat-Shamir protocol, we use the multiplicative group F^* of all non-zero elements in F, the multiplicative group

$$P = \{\alpha O : \alpha \in F^*\},$$

of principal fractional O-ideals, and the homomorphism

$$\varphi : F^* \to P, \quad \alpha \mapsto \alpha O.$$

With this choice of G, H, and φ, the general Fiat-Shamir protocol described in the previous section can be implemented. We call the resulting protocol PIP-FS.

In order for PIP-FS to be secure, inverting φ must be intractable. Inverting φ means solving the principal ideal problem for O-ideals. We discuss the difficulty of this PIP. The two most efficient methods known for solving the principal ideal problem are the babystep-giantstep algorithm [7] [1] and the index calculus method [29] [6]. The running time of the babystep-giantstep algorithm is $n^{O(n)} R^{1/2} |\Delta|^{o(1)}$ where n is the degree of F, Δ is the discriminant of O, and R is the regulator of O. More precisely, the following is true. Let I be a principal O-ideal. Let α be a generator of I such that the euclidean length of the logarithmic embedding of α (see [4]) is minimal, and let a be that length. Then α can be computed in time $n^{O(n)} \min\{a, R\}^{1/2} |\Delta|^{o(1)}$. The running time of the index calculus algorithm is $\exp(O(n \log n)(\log \Delta \log\log(\Delta))^{1/2})$. Thus, in order for the principal ideal problem to be intractable, the discriminant, the regulator, and the minimal logarithmic length of the generators must be sufficiently large. We note that no Pohlig-Hellman attack (see [24]) is known for PIP. Therefore, no further condition for the order O appears to be necessary.

For the appropriate choice of the order O, we use the analytic class number formula [4] and the heuristics of Cohen, Lenstra, and Martinet ([13], [14], [15], [16]). The analytic class number formula tells us that the product of the class number h and the regulator R of the algebraic number field F is asymptotically proportional to $\sqrt{|\Delta|}$ where Δ is the discriminant of F. The heuristics of Cohen, Lenstra, and Martinet predict when the class number is small with very high probability. Thus, if we choose the number field F

1. with sufficiently large discriminant in order to make index calculus attack infeasible and
2. with small class number (using the heuristics of Cohen, Lenstra, and Martinet),

then the principal ideal problem appears to be intractable.

Next, we must answer the question how the keys, commitment, witness, challenge, and response are selected or computed. The idea is to use reduced principal O-ideals and their generators. We will explain this in detail for the case of real quadratic orders. The methods explained for these orders can be generalized to general orders. This will be explained in a forthcoming paper.

4. PIP-FS in real quadratic fields

In this section, we show how to implement PIP-FS using a real quadratic order in which the principal ideal problem is intractable. In particular, we explain

1. how the order O is selected,
2. how the private key and the commitment are selected and represented,
3. how the public key, the witness, and the response are computed and represented, and
4. how the verification is performed.

We let O be a real quadratic order of discriminant Δ, class number h, and regulator R. By F we denote the field of fractions of O.

4.1. The order. As explained in Section 3, we have to choose the order O such that both the index calculus algorithm and the babystep-giantstep algorithm cannot be used to solve the principal ideal problem in O. We will, in fact, choose O as a maximal order.

The most efficient variant of the index calculus algorithm that is currently known is due to Jacobson [21]. Extrapolating experiments with Jacobson's algorithm, Hamdy [19] found that the difficulty of applying the index calculus algorithm in an order with a 687 bit discriminant is the same as the difficulty of factoring a 1024 bit number with the number field sieve. Therefore, we require that

$$\Delta > 2^{687} \tag{1}$$

Further comparisons can be found in Table 1.

To make the babystep-giantstep attack impossible, we choose the order such that

$$R > 2^{k_1} \ln \Delta, \tag{2}$$

where $k_1 \geq 160$ is a security parameter. The reason for this choice is given in Section 4.5.

Factoring	PIP-FS
1024	687
1536	958
2048	1208
3072	1665
4096	2084

TABLE 1. Comparison of factoring with the number field sieve and solving PIP-FS with index calculus.

To satisfy requirement (2), we choose Δ to be the product of two random primes $p_1, p_2 \equiv 3 \bmod 4$. We will show below that, assuming the Cohen-Lenstra heuristics [13] and the extended Riemann hypothesis, condition (2) is satisfied with probability $1 - 2^{-k_2}$, $k_2 \in \mathbb{N}$, if

$$\frac{\sqrt{\Delta}}{\ln \Delta \ln \ln \Delta} > 2^{k_1+k_2+3}. \tag{3}$$

For example, if $k_1 + k_2 = 240$, we obtain that $\Delta > 2^{509}$ is sufficient. So if (1) holds, then (2) is satisfied. Choosing Δ to be the product of two primes has additional appeal when these primes are not disclosed. Being able to solve PIP for arbitrary reduced principal ideals implies being able to factor the discriminant ([9], [27]). Thus in this case PIP is provably at least as hard as breaking cryptosystems such as RSA that rely on the assumption that factoring integers of this form is intractable.

So let us explain why it suffices to choose Δ according to (3). Since Δ is square-free, O is a maximal order. We can relate the regulator R to the class number h by using the analytic class number formula [4]: We have $2hR = \sqrt{\Delta} \cdot L(1,\chi)$ where $L(1,\chi) = \prod_{p \text{ prime}} \left(1 - \chi(p)/p\right)^{-1}$ is the value at 1 of the Dedekind L-series for the Kronecker symbol $\chi = \left(\frac{\Delta}{\cdot}\right)$. So our initial condition (2) translates to $\sqrt{\Delta} \cdot L(1,\chi)/(h \ln \Delta) > 2^{k_1+1}$. Assuming the ERH, we have, by a result of Littlewood [23], that $L(1,\chi) > (1+o(1))\left((12e^{\gamma}/\pi^2) \ln \ln \Delta\right)^{-1}$ where $\gamma = 0.5772\ldots$ is Euler's constant (cf. [25], where it is also discussed how $1 + o(1)$ can be replaced by explicit bounds). As $12e^{\gamma}/\pi^2 < 4$, certainly $L(1,\chi) > \frac{1}{4}(1+o(1))(\ln \ln \Delta)^{-1}$, and from this we obtain the new condition

$$\frac{(1+o(1))\sqrt{\Delta}}{h \ln \Delta \ln \ln \Delta} > 2^{k_1+3}. \tag{4}$$

We now examine the class number h in more detail. For the even part of h, we can use well-known theorems from genus theory [20]: Since Δ is the product of two primes $p_1, p_2 \equiv 3 \pmod 4$, the class number h is always odd.

For estimating the odd part of the class number, we apply the heuristics of Cohen and Lenstra [13] with the assumption that our restriction on the choice of Δ does not affect the statistical behaviour of the odd part of the class number. Then the probability that $h > x$ is asymptotic to $1/(2x)$ (cf. [13], § 9, (C12) a). With probability at least $1-2^{-k_2}$, we have $h \leq 2^{k_2}$. By substituting this into (4), we obtain the condition

$$\frac{(1+o(1))\sqrt{\Delta}}{2^{k_2}\ln\Delta\ln\ln\Delta} > 2^{k_1+3}.$$

Assuming that Δ is large enough such that $1+o(1)$ can be omitted, we arrive at (3).

4.2. Reduced O-ideals. To implement PIP-FS in O, we use reduced O-ideals, which we describe in this section. For more details on reduced ideals, we refer to [27] and [30].

Every fractional O-ideal I has a representation

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

where q is a positive rational number, a is a positive integer, and b is an integer. The numbers q and a are uniquely determined. The integer b is unique modulo $2a$. For $a > \sqrt{\Delta}$, we choose b such that $-a < b \leq a$; for $a < \sqrt{\Delta}$, we choose b such that $\sqrt{\Delta}-2a < b < \sqrt{\Delta}$. We represent I by (q,a,b,c) where $c=(b^2-\Delta)/(4a) \in \mathbb{Z}$. If $q=1$, then we write $I=(a,b,c)$.

The O-ideal $I=(a,b,c)$ is called *reduced* if $|\sqrt{\Delta}-2a| < b < \sqrt{\Delta}$. If $I=(a,b,c)$ is a reduced O-ideal, then $|a|+|c| < \sqrt{\Delta}$. This implies that the number of reduced O-ideals is finite. For example, the order O itself is a reduced principal O-ideal.

We explain the reduction operator ρ, which is the basic algorithmic primitive in reduction theory of O-ideals. If $I=(q,a,b,c)$ is an O-ideal, then

$$\rho(I) = (-at^2+bt-c, -b+2at, -a) \tag{5}$$

with

$$t = \begin{cases} \lfloor \frac{b}{2a} \rfloor & \text{for } a > \sqrt{\Delta} \\ \left\lfloor \frac{b+\lfloor\sqrt{\Delta}\rfloor}{2a} \right\rfloor & \text{for } a < \sqrt{\Delta}. \end{cases} \tag{6}$$

We can also write this as

$$\rho(I) = \alpha(I)I, \quad \alpha(I) = \frac{-b+2at+\sqrt{\Delta}}{2aq}. \tag{7}$$

If I is not reduced, then a reduced ideal J that is equivalent to I and an element $\gamma \in F^*$ with $J=\gamma I$ can be computed as follows.

1. Set $\gamma = 1$ and $J = I$.
2. While J is not reduced, replace γ by $\gamma\alpha(J)$ and J by $\rho(J)$.

This algorithm terminates with the correct result in quadratic time. We write $\gamma = \gamma(I)$ and $J = \operatorname{reduce}(I)$.

The fact that reduction of O-ideals is possible in polynomial time implies the following theorem, which shows that using only reduced O-ideals does not harm the security of PIP-FS.

Theorem 4.1. *There is a polynomial time reduction from PIP for O-ideals to PIP for reduced O-ideals.*

Proof. Let I be a fractional O-ideal. Instead of solving PIP for I, we solve PIP for $J = \operatorname{reduce}(I) = \gamma(I)I$. If J is not principal, then I is not principal. If J is principal and $J = \alpha O$, then I is principal and $I = \alpha/\gamma(I)O$. □

Restricted to the set of reduced ideals in an equivalence class of O-ideals, the reduction operator ρ is a transitive permutation. This implies that there is a positive integer p such that the set of reduced principal ideals is $\{\rho^i(O) : 0 \le i < p\}$ and $\rho^i(O) = \rho^j(O)$ if and only if $i \equiv j \bmod p$. If $I = (a, b, c)$ is reduced, then

$$\rho^{-1}(I) = (|c|, -b + 2s|c|, a + bs + cs^2), \quad s = \left\lfloor \frac{\sqrt{\Delta} + b}{2|c|} \right\rfloor, \tag{8}$$

which can also be written as

$$\rho^{-1}(I) = \frac{1}{\beta(I)} I, \quad \beta(I) = \frac{b + \sqrt{\Delta}}{2|c|}. \tag{9}$$

We also have (see [27, Lemma 3.2]

$$0 < \ln \alpha(I) < (\ln \Delta)/2 \tag{10}$$

and

$$\ln \alpha(I) + \ln \alpha(\rho(I)) \ge \ln 2 \tag{11}$$

for reduced I.

From (10) and (11), we obtain the following lemma, which is used in the construction of the private and public key and the commitment and witness.

Lemma 4.2. *Let r be a real number. Then there is a reduced O-ideal that has a positive generator α with $|\ln \alpha - r| < (\ln \Delta)/4$.*

Proof. Let $r > 0$. Set $I_0 = O$, $\alpha_0 = 1$, $I_{i+1} = \rho(I_i)$, $\alpha_{i+1} = \alpha_i \alpha(I_i)$. Then I_i is a reduced O-ideal with generator α_i, $\alpha_i > 0$, $0 < \ln \alpha_{i+1} - \ln \alpha_i = \ln \alpha(I_i) < (\ln \Delta)/2$ by (10), and $\lim_{i \to \infty} \ln \alpha_i = \infty$ by (11). This implies the assertion. For $r < 0$, the proof is analogous and uses ρ^{-1}. □

4.3. Private key, commitment, public key, and witness. Assume that Δ is chosen as described in Section 4.1.

The private key and the commitment are chosen such that the corresponding principal ideal problems are difficult. We now explain how we can choose them as generators of reduced principal ideals.

It follows from Theorem 4.1 that is is not harder to solve PIP for O-ideals than for reduced O-ideals. Therefore, we may limit ourselves to generators of reduced O-ideals when choosing the private key and the commitment. We denote the set of all reduced principal O-ideals by P_0. The public key and the witnesses are computed using the function

$$\mathrm{close}\colon \mathbb{N} \to P_0,$$

which is described in Section 4.6. Here, we use the following properties of that function:

1. Given $n \in \mathbb{N}$, the value $\mathrm{close}(n)$ can be computed in polynomial time.
2. For every $n \in \mathbb{N}$, there is a positive generator α of $\mathrm{close}(n)$ with $|\ln \alpha - cn| < (\ln \Delta)/4 + 1$, where $c = \lceil (\ln \Delta)/2 + 2 \rceil$. It can be computed in polynomial time.
3. The restriction of close to any interval of 2^{k_1} consecutive integers is injective, where $k_1 \geq 160$ is chosen as in Section 4.1.

To generate the private key, the prover randomly chooses k integers $n_1, \ldots, n_k$ in $[0, 2^{k_1} - 1]$. The corresponding public key is $(I_1, \ldots, I_k)$ with $I_i = \mathrm{close}(n_i)$, $1 \leq i \leq k$. For the generation of the commitment, we need two more security parameters $k_2, k_3 \in \mathbb{N}$. The integer k_2 is chosen such that an event that happens with probability $1/2^{k_2}$ is considered practically impossible. Parameter k_3 is chosen such that 2^{k_3} is an upper bound for the number of applications of the PIP-FS protocol with one key pair. For example, $k_3 = 30$ allows to use the same key pair every second for more than 30 years. The commitment is a random integer $n \in [0, 2^\ell - 1]$, $\ell = k_1 + k_2 + k_3 + 1$. The witness is $I = \mathrm{close}(n)$.

4.4. Response and verification. Let Δ and c be as in Section 4.3. Let $(n_1, \ldots, n_k)$ be the private key and $(I_1, \ldots, I_k)$ the corresponding public key, both chosen as in the previous section. Assume that n is a commitment, $I = \mathrm{close}(n)$ is the corresponding witness, and $(e_1, \ldots, e_k) \in [0,1]^k$ is the challenge. Then the response is $r = n + \sum_{i=1}^{k} e_i n_i$. Using r, the verifier is able to verify that there is a generator of $I \prod_{i=1}^{k} I_i^{e_i}$ that is close to cr. This is explained in Section 4.7.

4.5. Security. We explain why PIP-FS as described is secure. Let $P_1 = \{\mathrm{close}(n) : n \in [0, \ldots, 2^{k_1} - 1]\}$. As close is injective on the interval

$[0, \ldots, 2^{k_1} - 1]$, it is impossible to tabulate the elements of P_1 with generators or to guess a private key yielding the same public key. We analyze the time required to compute a generator of an element I of P_1. Let $I = \text{close}(n)$ with $n \in [0, 2^{k_1} - 1]$. Since by (2) we have $R \geq c \cdot 2^{k_1}$, it follows that the logarithm of the smallest positive generator of I is approximately cn. Therefore, the babystep-giantstep algorithm [1] for computing a generator of I takes $2^{k_1/2}\Delta^{o(1)}$ bit operations. So it is impossible to compute a generator of I this way. Also, by choice of Δ, determining a generator of I by the index calculus algorithm is impossible. Hence, computing generators for elements of P_1 is intractable. Similar arguments apply to close restricted to any other interval $[m, \ldots, m + 2^{k_1} - 1]$, and hence to close on intervals $[0, \ldots, b]$ where $b \geq 2^{k_1}$.

Now we show that knowledge of values of r, collected from up to 2^{k_3} executions of the protocol, with overwhelming probability does not allow the verifier or an observer to deduce any sub-sum of the secrets $n_1, \ldots, n_k$. For simplicity, assume that an attacker targets only n_1, which is most easily done by always using the challenge $(1, 0, \ldots, 0)$, so that $r = n + n_1$. As $n \in [0, 2^\ell - 1]$, this response r only reveals that $n_1 \in [r - 2^\ell + 1, r]$. It is known beforehand that $n_1 \in [0, 2^{k_1} - 1]$, so r is helpful for determining n_1 only if $r - 2^\ell + 1 > 0$ or $r < 2^{k_1} - 1$, which implies that $n + 2^{k_1} - 2^\ell > 0$ or $n < 2^{k_1} - 1$, respectively. Thus, for uniformly chosen $n \in [0, 2^\ell - 1]$, the probability is less than $2 \cdot 2^{k_1 - \ell}$. Combined over 2^{k_3} iterations of the protocol, the probability still is less than $1 - (1 - 2 \cdot 2^{k_1 - \ell})^{2^{k_3}}$, i.e. under $2 \cdot 2^{k_1 - \ell} \cdot 2^{k_3} = 2^{-k_2}$, which by choice of k_2 is considered negligible.

4.6. The function close. Let k_1, k_2, k_3, ℓ, k, O, F, Δ, and R be as in Section 4.3. We explain the implementation of a function

$$\text{close} : \mathbb{N} \to P_0,$$

with the properties from Section 4.3 where P_0 is the set of all reduced principal O-ideals. We will show that this function restricted to any interval of 2^{k_1} consecutive integers is injective, and that for any $n \in \mathbb{N}$ the ideal $\text{close}(n)$ has a positive generator α with $|\ln \alpha - cn| < (\ln \Delta)/4 + 1$ where

$$c = \lceil (\ln \Delta)/2 + 2 \rceil. \tag{12}$$

This function is used for the generation of the public key, the computation of the witness, and in the verification.

Let $n \in \mathbb{N}$. The algorithm computes $b = \lfloor \log_2 n \rfloor + 1$, i.e. the binary length of n, and $t = cn/2^b$. Then it recursively computes, for $0 \leq i \leq b$, reduced O-ideals I_i that have positive generators α_i with

$$|\ln \alpha_i - 2^i t| < (\ln \Delta)/4 + 1, \tag{13}$$

and returns $I_b = \text{close}(n)$.

To initialize the recursion, an O-ideal I_0 is computed that has a positive generator α_0 with $|\ln \alpha_0 - t| < (\ln \Delta)/4 + 1$. This is done by the following procedure:

1. Set $I_0 = O$, $\alpha_0 = 1$.
2. While $|\ln \alpha_0 - t| \geq (\ln \Delta)/4$, set $\alpha_0 = \alpha_0 \alpha(I_0)$ and $I_0 = \rho(I_0)$.

It follows from (10) and (11) that the algorithm terminates with the correct result after $O(\ln \Delta)$ iterations of the while loop. Since the implementation uses rational approximations to the logarithms, it can only guarantee that $|\ln \alpha_0 - t| < (\ln \Delta)/4 + 1$ when the while-condition is found to be false. The details are explained below.

Next we explain the recursion. When I_i has been found, $i < b$, then I_{i+1} is computed as follows. First, the ideal I_i^2 is determined. This is a principal O-ideal with generator α_i^2. Note that

$$(14) \qquad |\ln \alpha_i^2 - 2^{i+1} t| = 2|\ln \alpha_i^2 - 2^i t| < 2(\ln \Delta)/4 + 2.$$

So $\ln \alpha_i^2$ is pretty close to $2^{i+1}t$, but (13) may not be satisfied. Also, the ideal I_i^2 is, in general, not reduced. To make I_i^2 reduced, the reduction algorithm explained above is applied. It yields a reduced principal O-ideal I_{i+1} and an element $\gamma_{i+1} \in F^*$ such that $I_{i+1} = \gamma_{i+1} I_i^2$. If $\ln \gamma_{i+1}$ is too small, then we replace γ_{i+1} by $\gamma_{i+1}\alpha(I_{i+1})$ and I_{i+1} by $\rho(I_{i+1})$ until (13) holds for $\alpha_{i+1} = \gamma_{i+1}\alpha_i^2$. If $\ln \gamma_{i+1}$ is too large, then we replace γ_{i+1} by $\gamma_{i+1}/\beta(I_{i+1})$ and I_{i+1} by $\rho^{-1}(I_{i+1})$ until (13) holds for $\alpha_{i+1} = \gamma_{i+1}\alpha_i^2$. Note that α_{i+1} is a generator of the reduced principal ideal I_{i+1}.

As we can only work with approximations to the logarithms, things are somewhat more difficult. We explain the details. To obtain (13), we approximate $\ln \alpha_i$ by a rational number a_i with

$$(15) \qquad |\ln \alpha_i - a_i| < 1/4$$

for $0 \leq i \leq b$, and each γ_{i+1} is chosen such that

$$(16) \qquad |\ln \gamma_{i+1} - 2^{i+1}t + 2a_i| < (\ln \Delta)/4 + 1/2.$$

Then $\alpha_{i+1} = \gamma_{i+1}\alpha_i^2$ satisfies (13).

To find γ_{i+1} such that (16) holds, we work with rational approximations to the logarithms and modify the algorithm from above as follows: If, after reduction, the approximation to the logarithm γ_{i+1} is too small, we start with $\beta_0 = \gamma_{i+1}$ and determine ideals $J_j = \beta_j I_i^2$ by iterative application of ρ, such that $2^{i+1}t - 2a_i$ is larger than or equal to the approximation of $\ln \beta_{j-1}$, but smaller than the approximation of $\ln \beta_j$. If the logarithm of γ_{i+1} is too large, we use ρ^{-1} instead. At the end, γ_{i+1} is replaced by β_{j-1} or β_j, depending of which logarithm approximation is closer to $2^{i+1}t - 2a_i$. This is done by the following procedure.

1. $J_0 = \text{reduce}(I_i^2)$, $\beta_0 = \gamma(I_i^2)$, $|b_0 - \ln \beta_0| < 1/4$.

2. Set $T = 2^{i+1}t - 2a_i$.
3. If $b_0 \leq T$, do the following: Set $j = 0$. While $b_j \leq T$, set $J_{j+1} = \rho(J_j)$, $\beta_{j+1} = \alpha(J_j)\beta_j$, and compute b_{j+1} with $|b_{j+1} - \ln\beta_{j+1}| < 1/4$; replace j by $j+1$.

   Otherwise: Set $j = 1$. While $b_{j-1} > T$, replace j by $j-1$; set $J_{j-1} = \rho^{-1}(J_j)$, $\beta_{j-1} = \beta_j/\beta(J_j)$, and compute b_{j-1} with $|b_{j-1} - \ln\beta_{j-1}| < 1/4$.
4. If $T - b_{j-1} \leq b_j - T$, then set $I_{i+1} = J_{j-1}$ and $\gamma_{i+1} = \beta_{j-1}$. Otherwise, set $I_{i+1} = J_j$ and $\gamma_{i+1} = \beta_j$.

We prove that this procedure is correct.

Theorem 4.3. *γ_{i+1} determined by the procedure satisfies* (16).

Proof. Let T, β_j, b_j be as in the procedure and let j be with the value at the end of the procedure. It is

$$b_{j-1} \leq T < b_j.$$

If $T - b_{j-1} \leq b_j - T$, set $c = b_{j-1}$. Otherwise, set $c = b_j$.

First, we examine the case $\ln\beta_{j-1} \leq T < \ln\beta_j$. It is $|\ln\gamma_{i+1} - T| < |c - T| + 1/4 \leq \min\{|\ln\beta_{j-1} - T|, |\ln\beta_j - T|\} + 1/2$. So, (10) implies $|\ln\gamma_{i+1} - T| < (\ln\Delta)/4 + 1/2$.

The second case is $\ln\beta_j \leq T$. It is $|\ln\gamma_{i+1} - T| < |c - T| + 1/4 \leq |b_j - T| + 1/4 = b_j - T + 1/4 = b_j - \ln\beta_j + \ln\beta_j - T + 1/4 \leq |b_j - \ln\beta_j| + 1/4 < 1/2$. We can use the same arguments for the third case $T < \ln\beta_{j-1}$. $\square$

The initial value α_0 is computed by executing the procedure with $i = -1$, $I_{-1} = O$, and $a_{-1} = 0$. Then the procedure determines γ_0, and we set $\alpha_0 = \gamma_0$.

Finally, we remark that the approximations a_i can be computed as follows. Each generator is of the form $\alpha_i = \prod_{j=0}^{i} \gamma_j^{2^{i-j}}$. After γ_j has been computed by the procedure above, we approximate its logarithm by c_j such that $|c_j - \ln\gamma_j| < 2^{-n+j-b-3}$. We set $a_0 = c_0$ and $a_{i+1} = 2a_i + c_{i+1}$. Then a_i satisfies (15) for $0 \leq i \leq b$.

4.7. Implementing the verification. In the verification step, the verifier knows the positive integer r, the principal O-ideals $I, I_1, \ldots, I_k$, and the exponents $e_1, \ldots, e_k \in \{0, 1\}$. He wants to verify that the principal O-ideal $I \prod_{i=1}^{k} I_i^{e_i}$ has a positive generator α such that $\ln\alpha$ is close to cr, where c is defined according to (12).

The verifier uses the reduction algorithm from Section 4.2 to compute a reduced O-ideal J and $\gamma \in F^*$ with $J = \gamma I \prod_{i=1}^{k} I_i^{e_i}$. This is done as follows:

1. Set $J = I$, $\gamma = 1$.

2. For $i = 1, \ldots, k$: If $e_i = 1$, then replace γ by $\gamma\gamma(JI_i)$ and J by $\operatorname{reduce}(JI_i)$.

If α is a generator of $I\prod_{i=1}^k I_i^{e_i}$, then $\alpha\gamma$ is a generator of J. Since $I\prod_{i=1}^k I_i^{e_i}$ has a positive generator whose logarithm is close to cr and since $\ln\gamma$ is small, it follows that the reduced O-ideal J has a generator whose logarithm is close to cr. The verifier can verify this by computing $K = \operatorname{close}(r)$ and by searching for J in the neighborhood of K. More precisely, he applies the following algorithm where n is chosen according to Theorem 4.4:

1. Compute $K = \operatorname{close}(r)$. Set $K_{\mathrm{r}} = K_{\mathrm{l}} = K$ and $i = 0$.
2. While $i < n$, $J \neq K_{\mathrm{l}}$ and $J \neq K_{\mathrm{r}}$ replace i by $i+1$, K_{r} by $\rho(K_{\mathrm{r}})$, and K_{l} by $\rho^{-1}(K_{\mathrm{l}})$.

In the next theorem, we give an upper bound on the number of steps that are necessary for the verification to succeed.

Theorem 4.4. *If r is chosen according to the protocol, then the verification succeeds after at most $n = \lceil 2(x + (\ln\Delta)/4 + 1)/\ln 2\rceil$ iterations, where $x = ((\ln\Delta)/4 + 1)\sum_{i=1}^k e_i + \ln(32\Delta)\sum_{i=1}^k e_i$.*

Proof. Let J_{i-1} denote the value of J before the i-th iteration of the for-loop used in the reduction of $I\prod_{i=1}^k I_i^{e_i}$. Then $|\ln\gamma(J_{i-1}I_i)| \leq \ln(32\Delta)$ holds for each i (see [30, Theorem 5.2]). This implies

$$|\ln\gamma| \leq \ln(32\Delta)\sum_{i=1}^{k} e_i. \tag{17}$$

Assume that the response is correct, i.e., there exists a positive generator $\alpha \in F^*$ with $I\prod_{i=1}^k I_i^{e_i} = \alpha O$ and $|\ln\alpha - cr| < ((\ln\Delta)/4+1)(1+\sum_{i=1}^k e_i)$. It follows from (17) that there is a generator $\beta = \alpha\gamma$ of J with $|\ln\beta - cr| < x$. By (13), we have $|\operatorname{close}(cr) - cr| < (\ln\Delta)/4 + 1$. Since we apply ρ and ρ^{-1} to find J and since by (11) the step width of two such applications is at least $\ln 2$, we obtain the assertion. $\square$

4.8. Timings. We have implemented the real quadratic order PIP-FS identification protocol in C++. The running times given in this section has been measured on a Pentium II, 300 MHz, 64 MB main memory, running SuSe Linux 2.0.35, using the compiler egcs-2.91.57 with option `-O2`, and using the LiDIA-2.0 library [22] with libI as underlying kernel arithmetic. All timings are given in seconds.

The setup is as follows. For each $m \in \{687, 968, 1208, 1665, 2084\}$, we choose several discriminants with m bits, run the protocol, including order and key generation, and measure the average time per discriminant. We have run the tests with $k = 30$, i.e., the probability that a cheating prover is detected is at least $1 - 1/2^{30}$ As challenge, we choose a bit string with 15 bits set to 1. Furthermore, the security parameters are chosen as $k_1 = 160$, $k_2 = 80$, and $k_3 = 30$ (see Section 4.3).

m	687	968	1208	1665	2084
Order	0.98	3.96	13.25	16.16	63.42
Key pair	54.87	84.56	122.74	196.4	291.76
Witness	3.23	4.86	7.03	10.7	16.03
Response	0	0	0	0	0
Verification	3.28	4.96	7.15	11.04	16.45

References

[1] I. BIEHL, J. BUCHMANN, *Algorithms for quadratic orders.* In: Mathematics of Computation 1943–1993: a half-century of computational mathematics, Vancouver 1993, W. Gautschi, Ed., vol. 48 of *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society (1995), pp. 425–449.

[2] I. BIEHL, J. BUCHMANN, S. HAMDY, A. MEYER, *A signature scheme based on the intractability of extracting roots.* Tech. Rep. TI-1/00, Technische Universität Darmstadt, Fachbereich Informatik, 2000. `http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/`.

[3] I. BIEHL, B. MEYER, C. THIEL, *Cryptographic protocols based on real-quadratic A-fields (extended abstract).* In Advances in Cryptology – ASIACRYPT '96, K. Kim and T. Matsumoto, Eds., vol. 1163 of *Lecture Notes in Computer Science*, Springer-Verlag (1996), pp. 15–25.

[4] Z. I. BOREVICH, I. R. SHAFAREVICH, *Number theory.* Academic Press, New York (1966).

[5] G. BRASSARD, Ed., *Advances in Cryptology – CRYPTO '89*, vol. 435 of *Lecture Notes in Computer Science*, Springer-Verlag (1990).

[6] J. BUCHMANN, *On the computation of units and class numbers by a generalization of Lagrange's algorithm.* Journal of Number Theory **26** (1987), 8–30.

[7] J. BUCHMANN, it Zur Komplexität der Berechnung von Einheiten und Klassenzahlen algebraischer Zahlkörper. Habilitationsschrift (1987).

[8] J. BUCHMANN, S. PAULUS, *A one way function based on ideal arithmetic in number fields.* In: Advances in Cryptology – CRYPTO '97, B. S. Kaliski, Ed., vol. 1294 of *Lecture Notes in Computer Science*, Springer-Verlag (1997), pp. 385–394.

[9] J. BUCHMANN, H. C. WILLIAMS, *A key exchange system based on real quadratic fields.* In Brassard [5], pp. 335–343.

[10] D. A. BUELL, *Binary quadratic forms.* Springer-Verlag, New York (1989).

[11] M. V. D. BURMESTER, Y. DESMEDT, F. PIPER, M. WALKER, *A general zero-knowledge scheme.* In: Advances in Cryptology – EUROCRYPT '89, J.-J. Quisquater and J. Vandewalle, Eds., vol. 434 of *Lecture Notes in Computer Science*, Springer-Verlag (1990), pp. 122–133.

[12] D. CHAUM, J.-H. EVERTSE, J. VAN DE GRAAF, *An improved protocol for demonstrating posession of discrete logarithms and some generalizations.* In: *Advances in Cryptology – EUROCRYPT '87*, D. Chaum and W. L. Price, Eds., vol. 304 of *Lecture Notes in Computer Science*, Springer-Verlag (1988), pp. 127–142.

[13] H. COHEN, J. W. LENSTRA, JR., *Heuristics on class groups of number fields.* In: Number Theory, Noordwijkerhout 1983, H. Jager, Ed., vol. 1068 of *Lecture Notes in Mathematics*. Springer-Verlag (1984), pp. 33–62.

[14] H. COHEN, J. MARTINET, *Class groups of number fields: numerical heuristics.* Mathematics of Computation **48** (1987), 123–137.

[15] H. COHEN, J. MARTINET, *Étude heuristique des groupes de classes des corps de nombres.* Journal für die reine und angewandte Mathematik **404** (1990), 39–76.

[16] H. COHEN, J. MARTINET, *Heuristics on class groups: Some good primes are not too good.* Mathematics of Computation **63** (1994), 329–334.

[17] A. Fiat, A. Shamir, *How to prove yourself: practical solutions to identification and signature problems.* In: Advances in Cryptology – CRYPTO '86 (1987), A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 186–194.

[18] L. C. Guillou, J.-J. Quisquater, *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.

[19] S. Hamdy, B. Möller, *Security of cryptosystems based on class groups of imaginary quadratic orders.* In: Advances in Cryptology – ASIACRYPT 2000 (2000), T. Matsumoto, Ed., Lecture Notes in Computer Science, Springer-Verlag. To appear.

[20] E. Hecke, *Vorlesungen über die Theorie der Algebraischen Zahlen.* Leipzig (1923).

[21] M. J. Jacobson, Jr., *Subexponential Class Group Computation in Quadratic Orders.* PhD thesis, Technische Universität Darmstadt, Fachbereich Informatik, Darmstadt, Germany, 1999.

[22] LiDIA – a C++ library for computational number theory. `http://www.informatik.tu-darmstadt.de/TI/LiDIA/`. The LiDIA Group.

[23] J. E. Littlewood, *On the class number of the corpus* $P(\sqrt{-k})$. Proceedings of the London Mathematical Society 2nd series **27** (1928), 358–372.

[24] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press (1997).

[25] R. A. Mollin, H. C. Williams, *Computation of the class number of a real quadratic field.* Utilitas Mathematica **41** (1992), 59–308.

[26] G. Poupard, J. Stern, *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.

[27] R. Scheidler, J. Buchmann, H. C. Williams, *A key-exchange protocol using real quadratic fields.* Journal of Cryptology **7** (1994), 171–199.

[28] C. P. Schnorr, *Efficient identification and signatures for smart cards.* In: Brassard [5], pp. 239–252.

[29] U. Vollmer, *Asymptotically fast discrete logarithms in quadratic number fields.* In: Algorithmic Number Theory, ANTS-IV (2000), W. Bosma, Ed., vol. 1838 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 581–594.

[30] H.C. Williams, M.C. Wunderlich, *On the parallel generation of the residues for the continued fraction factoring algorithm.* Mathematics of Computation **48** (1987), 405–423.

Johannes Buchmann, Markus Maurer, Bodo Möller
Technische Universität Darmstadt
Fachbereich Informatik
Alexanderstr. 10
64283 Darmstadt
Germany
E-mail : {`buchmann,mmaurer,moeller`}`@cdc.informatik.tu-darmstadt.de`