

THE RABIN CRYPTOSYSTEM OVER NUMBER FIELDS

ALESSANDRO COBBE, ANDREAS NICKEL, AND AKAY SCHUSTER

ABSTRACT. We extend Rabin’s cryptosystem to general number fields. We show that decryption of a random plaintext is as hard as the integer factorisation problem, provided the modulus in our scheme has been chosen carefully. We investigate the performance of our new cryptosystem in comparison with the classical Rabin scheme and a more recent version over the Gaussian integers.

INTRODUCTION

Rabin’s cryptosystem [7] was the first asymmetric public-key cryptosystem for which it was shown that any algorithm which finds one of the possible plaintexts for every Rabin-encrypted ciphertext can be used to factor the modulus. It is performed in the quotient ring $\mathbb{Z}/N\mathbb{Z}$ where $N = pq$ is the product of two large distinct prime numbers. The public key is N and the private key is the pair (p, q) . Encryption is very simple: To encrypt a message $m \in \mathbb{Z}/N\mathbb{Z}$ one just computes its square; so $c = m^2$ is the ciphertext. For decryption, one uses the knowledge of p and q to compute the square roots of $c \pmod{p}$ and $c \pmod{q}$. For this one can use the algorithm of Tonelli-Shanks (see [2, §1.5.1], for instance), but there is an easier method for primes which are congruent to 3 modulo 4: If y is a quadratic residue modulo such a p , then $\pm y^{(p+1)/4}$ are the roots of y . Finally, one uses the Chinese Remainder Theorem to obtain the (in general four) square roots of c modulo N . It remains to decide which of these roots is the original message m .

In this paper, we generalise Rabin’s scheme to rings of integers in number fields. So let us fix a number field K with ring of integers \mathcal{O}_K and choose two distinct non-zero prime ideals \mathfrak{p} and \mathfrak{q} in \mathcal{O}_K . Our scheme is performed in the quotient ring $\mathcal{O}_K/\mathfrak{n}$ where $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$. The public key is the modulus \mathfrak{n} and the private key is the pair $(\mathfrak{p}, \mathfrak{q})$. The ciphertext of a message $m \in \mathcal{O}_K/\mathfrak{n}$ is again $c = m^2$. For decryption one has to compute the square roots of c modulo \mathfrak{p} and modulo \mathfrak{q} . For this, we provide a generalisation of the Tonelli-Shanks algorithm. Finally, one uses the Chinese Remainder Theorem to obtain the (again up to four) square roots of c modulo \mathfrak{n} .

The special case of Gaussian integers has been treated by Awad, El-Kassar and Kadri [1]. They use the (extended) Euclidean algorithm in $\mathbb{Z}[i]$ to make the isomorphism in the Chinese Remainder Theorem explicit. This does not generalise to arbitrary number fields. We provide a version that only relies on the Euclidean algorithm in \mathbb{Z} .

In order to speed up decryption, we also provide an easy and fast way to compute square roots modulo primes \mathfrak{p} in \mathcal{O}_K under certain restrictions on \mathfrak{p} , thereby generalising the above method for primes $p \equiv 3 \pmod{4}$. This approach works particularly well in number fields of degree 3, but does not work in any quadratic number field.

Date: Version of 3rd June 2025.

2020 Mathematics Subject Classification. 94A60, 11T71, 11Y40.

This is one reason why we need to allow more freedom in choosing the field K . Indeed, our runtime analysis shows that decryption of ciphertexts of comparable size becomes a lot faster over carefully chosen cubic than over quadratic fields.

It turns out that it is a subtle question how to represent the public key \mathbf{n} . There are two reasonable ways to do this: either as a list of generators or in Hermite normal form. We show that for most choices of \mathfrak{p} and \mathfrak{q} this is not secure. It indeed allows us to factor \mathbf{n} . Our considerations led us to the conclusion that the best choices are prime ideals of the form $\mathfrak{p} = (p)$, where $p \in \mathbb{Z}$ is prime, i.e. we consider rational primes p which are inert in K . So we also need conditions that guarantee us that a given rational prime remains prime in K . We will provide a class of number fields, where this can be checked by a simple congruence condition. So the public key in our scheme is just $N = pq$ where $p, q \in \mathbb{Z}$ are primes that are inert in K . Then, as in Rabin's original scheme, an algorithm which finds one of the possible plaintexts for each ciphertext allows N to be factored.

We also address the question how to find the original message m among the usually four square roots of the ciphertext $c = m^2$. In the classical case, Williams [10] proposed a method whenever both primes p and q are congruent to 3 modulo 4. It suffices to add two extra bits to the ciphertext: the parity and the Legendre symbol $\left(\frac{m}{N}\right)$ of the message m . If we assume in addition that p and q are inert in K , we will show that it is still possible to uniquely identify the original message by adding two extra bits.

Finally, we note that Petukhova and Tronin [6] have generalised the RSA scheme to general Dedekind domains with finite residue fields. However, they have not addressed the following questions: (i) how to find such Dedekind domains, (ii) how to make the required computations explicit, (iii) when is the scheme secure? Though we focus on Rabin's cryptosystem in this paper, many of our considerations also apply to this generalisation of the RSA scheme. (i) Rings of integers in number fields are a natural source of Dedekind domains with finite residue fields. (ii) Our remarks on computations in residue rings of the form $\mathcal{O}_K/\mathfrak{n}$ also apply to the generalised RSA scheme. (iii) The public key in the generalised RSA scheme is a pair (\mathbf{n}, e) , where e is coprime with the cardinality of $(\mathcal{O}_K/\mathfrak{n})^\times$. Since we show that the knowledge of \mathbf{n} (if given as a list of generators or in Hermite normal form) often suffices to factor it, one needs to impose the same conditions on \mathfrak{p} and \mathfrak{q} also in the generalised RSA scheme.

1. THE PUBLIC KEY

1.1. Number fields and rings of integers. Let us first explain how we represent number fields. Let K be a number field of degree d over \mathbb{Q} . Let $\theta \in K$ be a primitive element so that $K = \mathbb{Q}(\theta)$, which means that $1 = \theta^0, \theta, \theta^2, \dots, \theta^{d-1}$ constitute a \mathbb{Q} -basis of K . The minimal polynomial of θ is the unique monic polynomial $g_\theta \in \mathbb{Q}[x]$ of lowest degree with root θ . It is irreducible of degree d and there is a field isomorphism

$$(1) \quad \mathbb{Q}[x]/(g_\theta) \xrightarrow{\simeq} K = \mathbb{Q}(\theta), \quad x \mapsto \theta.$$

So if we 'choose' a number field, we really pick a monic irreducible polynomial $g \in \mathbb{Q}[x]$ and consider the field $\mathbb{Q}[x]/(g)$. If we want to view this field as a subfield of the complex numbers, we choose a root $\theta \in \mathbb{C}$ of g and apply (1). Then one has $g_\theta = g$ and the set of embeddings of $\mathbb{Q}[x]/(g)$ into \mathbb{C} is in one-to-one correspondence with the roots of g .

Let \mathcal{O}_K be the ring of integers in K . Then one can always choose $\theta \in \mathcal{O}_K$ so that g_θ has integral coefficients. The isomorphism (1) then induces an isomorphism of rings

$$\mathbb{Z}[x]/(g_\theta) \simeq \mathbb{Z}[\theta].$$

The ring $\mathbb{Z}[\theta]$ is always contained in \mathcal{O}_K , but we do not have equality in general. This will be crucial in the following, as we get the ring $\mathbb{Z}[\theta]$ for free once we have g_θ , but it might be rather expensive to compute \mathcal{O}_K in general. For instance, Zassenhaus's Round 2 Algorithm [2, §6.1.4] first factors the discriminant of g_θ , which is a hard problem when the discriminant is large. Luckily, it will not be necessary to determine \mathcal{O}_K for our purposes (this will be explained below). Alternatively, one can stick to (classes of) number fields, where the ring of integers has been determined.

Example 1.1 (Quadratic number fields). Let $\delta \in \mathbb{Z}$ be square-free, $\delta \neq 0, 1$. Then $x^2 - \delta \in \mathbb{Z}[x]$ is irreducible and $\mathbb{Q}[x]/(x^2 - \delta) \simeq \mathbb{Q}(\sqrt{\delta})$. The ring of integers in $\mathbb{Q}(\sqrt{\delta})$ is $\mathbb{Z}[\omega]$, where $\omega = \sqrt{\delta}$ if $\delta \equiv 2$ or $3 \pmod{4}$, whereas $\omega = (1 + \sqrt{\delta})/2$ if $\delta \equiv 1 \pmod{4}$ (see [2, Proposition 5.1.1], for example). So in the latter case the index of $\mathbb{Z}[\sqrt{\delta}]$ in $\mathcal{O}_{\mathbb{Q}(\sqrt{\delta})}$ is 2.

Example 1.2 (Cyclotomic fields). Let m be a positive integer such that $m \not\equiv 2 \pmod{4}$. Let $\zeta_m \in \mathbb{C}$ be a primitive m -th root of unity. Then $\mathbb{Q}(\zeta_m)$ has degree $\varphi(m)$ over \mathbb{Q} , where φ denotes Euler's totient function. Here one always has $\mathcal{O}_{\mathbb{Q}(\zeta_m)} = \mathbb{Z}[\zeta_m]$ by [2, Proposition 9.1.2].

Every $\alpha \in K$ can uniquely be written as $\alpha = \sum_{i=0}^{d-1} \alpha_i \theta^i$ with rational coefficients $\alpha_0, \dots, \alpha_{d-1}$. If we use this representation for elements in K , addition becomes very easy. For multiplication one essentially has to compute once the representations for θ^{d+k} with $0 \leq k < d-1$.

We denote the norm of $\alpha \in K$ by $N(\alpha)$.

Example 1.3. In the case of quadratic number fields, computing the norm is very simple. Suppose that $\alpha = a + b\sqrt{\delta} \in \mathbb{Q}(\sqrt{\delta})$, where $\delta \neq 0, 1$ is square-free and $a, b \in \mathbb{Q}$. Then it is easy to see that $N(\alpha) = a^2 - \delta b^2$.

1.2. Prime ideals. For our scheme we need to work in residue fields of the form $\mathcal{O}_K/\mathfrak{p}$, where \mathfrak{p} is a non-zero prime ideal of \mathcal{O}_K . However, we want to avoid computing \mathcal{O}_K and to work with $\mathbb{Z}[\theta]$ instead.

Let p be the rational prime below \mathfrak{p} and $\mathfrak{p}_\theta := \mathfrak{p} \cap \mathbb{Z}[\theta]$ the prime ideal of $\mathbb{Z}[\theta]$ below \mathfrak{p} . The index of $\mathbb{Z}[\theta]$ in \mathcal{O}_K is finite. Whenever it is not divisible by p , the natural inclusion

$$\mathbb{Z}[\theta]/\mathfrak{p}_\theta \rightarrow \mathcal{O}_K/\mathfrak{p}$$

is an isomorphism. As p will be chosen to be a large prime, the probability that p divides the unknown index $[\mathcal{O}_K : \mathbb{Z}[\theta]]$ is negligibly small. However, there are even methods to guarantee this.

Let us denote the discriminant of a polynomial g by $\Delta(g)$ and the discriminant of K by Δ_K . If $K = \mathbb{Q}(\theta)$, then we have

$$\Delta(g_\theta) = \Delta_K \cdot [\mathcal{O}_K : \mathbb{Z}[\theta]]^2.$$

So it is sufficient to choose a prime p that does not divide $\Delta(g_\theta)$ and the latter is computable (see [2, §3.3]). However, we can also use Mahler's bound [5, Corollary on p. 261]

$$\Delta(g_\theta) < C_\theta := d^d L(g_\theta)^{2d-2},$$

where we recall that $d = \deg(g_\theta)$ and $L(g_\theta) := \sum_{i=0}^d |a_i|$ if $g = \sum_{i=0}^d a_i x^i$. This bound is very easy to compute and choosing p larger than C_θ will be sufficient for our purposes.

In order to choose a prime ideal \mathfrak{p} we first choose a rational prime $p > C_\theta$. Then the polynomial $g_\theta \pmod{p} \in \mathbb{F}_p[x]$ is square-free and we let

$$g_\theta \equiv \prod_{j=1}^r g_j(x) \pmod{p}$$

be its decomposition into irreducible factors in $\mathbb{F}_p[x]$. Here $g_j(x) \in \mathbb{Z}[x]$ are taken to be monic. Note that there are efficient algorithms for the factorisation of polynomials modulo p , which indeed simplify if the polynomial is square-free [2, §3.4]. Then by [2, Theorem 4.8.13] choosing a prime ideal above p is the same as choosing a factor g_j . Indeed the prime ideals are given by $\mathfrak{p}_j = (p, g_j(\theta))$ and the residue degree $[\mathcal{O}_K/\mathfrak{p}_j : \mathbb{F}_p]$ is equal to the degree of g_j .

We now fix such a choice and change notation slightly. We simply write \mathfrak{p} for the prime ideal we have obtained in that way and rename the chosen g_j as $g_\mathfrak{p}$ to make the dependence on \mathfrak{p} visible. So we have $\mathfrak{p} = (p, g_\mathfrak{p}(\theta))$, where $g_\mathfrak{p}(x) \pmod{p}$ is an irreducible factor of $g_\theta(x) \pmod{p}$. Since

$$\mathbb{Z}[\theta]/(p, g_\mathfrak{p}(\theta)) \simeq \mathcal{O}_K/(p, g_\mathfrak{p}(\theta))$$

by our choice of p , we abuse notation and write \mathfrak{p} for the ideal generated by p and $g_\mathfrak{p}(\theta)$ in both \mathcal{O}_K and $\mathbb{Z}[\theta]$. In the same way we choose a second large prime $q \neq p$ and a prime ideal $\mathfrak{q} = (q, g_\mathfrak{q}(\theta))$ above q . We write $f_\mathfrak{p}$ and $f_\mathfrak{q}$ for the residue degree $[\mathcal{O}_K/\mathfrak{p} : \mathbb{F}_p]$ and $[\mathcal{O}_K/\mathfrak{q} : \mathbb{F}_q]$, respectively.

1.3. The public key. The public key consists of the monic irreducible polynomial $g_\theta \in \mathbb{Z}[x]$ and the ideal $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$. Alternatively, we may consider the polynomial g_θ and thus the field K as fixed. For the security of a generalised version of the Rabin cryptosystem it is crucial that it is not feasible to compute the decomposition into prime ideals of $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$, i.e. computing \mathfrak{p} and \mathfrak{q} assuming we know \mathfrak{n} .

First of all we observe that this is at most as hard as factoring the number $N = pq$, where p and q are the prime numbers below \mathfrak{p} and \mathfrak{q} , under the assumption that we represent \mathfrak{n} in a way which allows to compute $\mathfrak{n} \cap \mathbb{Z} = (N)$. Indeed from the factorisation of N , we easily compute all the primes of \mathcal{O}_K dividing p and q and we just need to pick the correct ones. To decide which of the factors actually divide \mathfrak{n} we can use the algorithm described in [2, §4.8.3] to compute the valuation of \mathfrak{n} with respect to the different prime ideals.

Before continuing, we need to clarify how we present \mathfrak{n} . Recall that p is called inert in K if $\mathfrak{p} = (p)$. If both p and q are inert, we have $\mathfrak{n} = (N)$, where $N = pq$, and we can publish the generator N . Now suppose that at least one of p and q is not inert. We claim that it is not secure to publish the four generators $N := pq$, $pg_\mathfrak{q}(\theta)$, $qg_\mathfrak{p}(\theta)$ and $g_\mathfrak{q}(\theta)g_\mathfrak{q}(\theta)$. To see this, we may and do assume that p is not inert. Then the coefficients of $g_\mathfrak{p}(\theta)$ are not all divisible by p , as otherwise $\mathfrak{p} = (p, g_\mathfrak{p}(\theta)) = (p)$. Hence the greatest common divisor of the coefficients of $qg_\mathfrak{p}(\theta)$ and N is q and we have factored N .

Let us assume that $g_\mathfrak{p}(\theta) \in \mathfrak{p} \setminus \mathfrak{p}^2$ and that $g_\mathfrak{p}(\theta)$ is coprime to q , and the same with the roles of p and q reversed. Both conditions can be achieved in polynomial time by [3, Proposition 1.3.11]. In practice, we can replace $g_\mathfrak{p}(\theta)$ by $g_\mathfrak{p}(\theta) + p$ if $g_\mathfrak{p}(\theta) \in \mathfrak{p}^2$. Then the first condition holds. The probability that the second condition fails is

negligibly small as q has been chosen independently of p and the probability of a random $x \in \mathcal{O}_K$ to lie in a prime ideal above q is at most d/q .

Under these conditions one has $\mathfrak{n} = (N, h(\theta))$ with $N = pq$ and $h = g_{\mathfrak{p}} \cdot g_{\mathfrak{q}}$. The next result shows that this is still not secure unless we choose $g_{\mathfrak{p}}$ and $g_{\mathfrak{q}}$ of the same degree, i.e. $f_{\mathfrak{p}} = f_{\mathfrak{q}}$.

Proposition 1.4. *Let $K = \mathbb{Q}(\theta)$, where θ is a root of a monic irreducible polynomial $g_{\theta} \in \mathbb{Z}[x]$. Let \mathfrak{p} and \mathfrak{q} be two prime ideals of \mathcal{O}_K lying over two distinct unramified primes $p, q \in \mathbb{Z}$. Let us assume that the inertia degrees $f_{\mathfrak{p}}$ and $f_{\mathfrak{q}}$ of \mathfrak{p} and \mathfrak{q} are distinct. Suppose that $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$ is given by a set of generators as described above or in Hermite normal form, and $\mathfrak{n} \cap \mathbb{Z} = (N)$. Then it is possible to efficiently compute p and q .*

Proof. We first show that we can factor N if we can compute the norm of \mathfrak{n} , namely $\mathcal{N}(\mathfrak{n}) := [\mathcal{O}_K : \mathfrak{n}] = p^{f_{\mathfrak{p}}}q^{f_{\mathfrak{q}}}$. Indeed if N^a is the largest power of N that divides $\mathcal{N}(\mathfrak{n})$, then $\mathcal{N}(\mathfrak{n})/N^a$ is divisible by exactly one of the prime numbers p and q and $\gcd(\mathcal{N}(\mathfrak{n})/N^a, N)$ is a prime factor of N .

If \mathfrak{n} is described by its Hermite normal form, which is a square matrix, then the norm of \mathfrak{n} is the ideal of \mathbb{Z} generated by the determinant of the matrix [2, Proposition 4.7.4]. So we are done in this case.

Assume now that $\mathfrak{n} = (N, h(\theta))$ as above. We actually need to compute $\mathcal{N}(\mathfrak{n})$ only up to some factor which is coprime to N , so it is enough to consider

$$\mathcal{N}((h(\theta))) = \mathcal{N}(\mathfrak{p})\mathcal{N}(\mathfrak{q})\mathcal{N}(\mathfrak{a}) = p^{f_{\mathfrak{p}}}q^{f_{\mathfrak{q}}}\mathcal{N}(\mathfrak{a}),$$

where $\mathcal{N}(\mathfrak{a})$ is prime to N . As the index $[\mathcal{O}_K : \mathbb{Z}[\theta]]$ is not divisible by either p or q , it is enough to compute $[\mathbb{Z}[\theta] : (h(\theta))]$, which can be done for instance via computing the Smith normal form (see [2, §2.4.4]) of the quotient $\mathbb{Z}[\theta]/(h(\theta))$. \square

By the above proposition, there is an efficient algorithm to factor \mathfrak{n} if $f_{\mathfrak{p}} \neq f_{\mathfrak{q}}$. Since this is not desirable, we focus on the remaining case $f_{\mathfrak{p}} = f_{\mathfrak{q}}$.

Let us consider the special case in which \mathfrak{p} and \mathfrak{q} are inert over \mathfrak{q} , i.e. $f_{\mathfrak{p}} = f_{\mathfrak{q}} = [K : \mathbb{Q}]$. Then $\mathfrak{p} = (p)$, $\mathfrak{q} = (q)$ and $\mathfrak{n} = (pq)$ and factoring \mathfrak{n} is clearly equivalent to factoring its generator pq . Moreover, restricting to the case of inert primes has some further positive side effects: the amount of information that can be sent with our protocol becomes maximal and the computations which have to be performed are easier.

In the case $f_{\mathfrak{p}} = f_{\mathfrak{q}} < [K : \mathbb{Q}]$ we can neither provide a proof that the factorisation of \mathfrak{n} is as difficult as the factorisation of N nor we can show that there is an efficient algorithm to factor \mathfrak{n} . In the case of quadratic number fields, we can offer the following result.

Proposition 1.5. *Let $N = pq$ be a product of two distinct prime numbers. Suppose that there is an algorithm with the following property: For a randomly chosen non-square δ in the range $0 < \delta < N$ such that the primes p and q split in $\mathbb{Q}(\sqrt{\delta})$, it computes the factorisation of a randomly chosen ideal $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$ with probability at least $\omega > 0$, where \mathfrak{p} and \mathfrak{q} are prime ideals of $\mathcal{O}_{\mathbb{Q}(\sqrt{\delta})}$ dividing p and q , respectively.*

Then there is an algorithm that computes the factorisation of N after k steps with probability at least $1 - (1 - \omega)^k$.

Proof. Choose a random non-square $0 < \delta < N$. If $\gcd(\delta, N) > 1$, we can factor N so that we will exclude this case in the following. Let us write $\delta = b^2\delta'$, where δ' is square-free. Then $\mathbb{Q}(\sqrt{\delta}) = \mathbb{Q}(\sqrt{\delta'})$ so that the index of $\mathbb{Z}[\delta]$ in $\mathcal{O}_{\mathbb{Q}(\sqrt{\delta})}$ is either b or $2b$ and thus coprime with N . As a consequence, p and q split in $\mathbb{Q}(\sqrt{\delta})$ if and

only if δ is a square modulo p and q and hence modulo N . So in this case we have $\delta \equiv a^2 \pmod{N}$ for some $0 < a < N$. Note that actually $a > \sqrt{N}$ as otherwise $\delta = a^2$ would be a perfect square. This observation holds for all four square roots of $\delta \pmod{N}$ and in particular for $N - a$ so that we must have $\sqrt{N} < a < N - \sqrt{N}$.

For the required algorithm, we first choose a random $\sqrt{N} < a < N - \sqrt{N}$ and take $0 < \delta < N$ to be a representative of the class of a^2 modulo N . If $\delta = b^2$ happens to be a square, then b is a root of δ , but $b \not\equiv \pm a \pmod{N}$ so that $\gcd(N, b - a)$ is either p or q . If δ is a non-square, we consider the ideal $\mathfrak{n} = (N, a + \sqrt{\delta})$ in the quadratic number field $\mathbb{Q}(\sqrt{\delta})$. We have seen above that p and q split in $\mathbb{Q}(\sqrt{\delta})$. The norm of $a + \sqrt{\delta}$ is $a^2 - \delta = kN$ for some $0 < k < N$, since $0 < \delta < N < a^2 < N^2$. Once more, if k happens to be a multiple of either p or q , we can factor N by computing $\gcd(N, k)$. So let us assume that this is not the case. Then we must have $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$ for prime ideals \mathfrak{p} above p and \mathfrak{q} above q . If we can factor \mathfrak{n} , we find \mathfrak{p} and \mathfrak{q} , and computing the gcd of their norms with N we finally obtain p and q . This event has probability at least ω if we can show that δ and \mathfrak{n} have been chosen at random.

For this, we observe that each non-square δ in the range $0 < \delta < N$ such that the primes p and q split in $\mathbb{Q}(\sqrt{\delta})$ is of the form $\delta \equiv a^2 \pmod{N}$ for exactly four choices of a in the range $\sqrt{N} < a < N - \sqrt{N}$. If our choice a leads to the ideal $\mathfrak{n} = (N, a + \sqrt{\delta}) = \mathfrak{p}\mathfrak{q}$, then $\mathfrak{p} = (p, a + \sqrt{\delta})$ and $\mathfrak{q} = (q, a + \sqrt{\delta})$. The second prime above p is $\bar{\mathfrak{p}} := (p, a - \sqrt{\delta}) = (p, N - a + \sqrt{\delta})$. We define $\bar{\mathfrak{q}}$ and $\bar{\mathfrak{n}}$ similarly. Then the root $N - a$ of $\delta \pmod{N}$ leads to $\bar{\mathfrak{n}} = \bar{\mathfrak{p}}\bar{\mathfrak{q}}$. The third root a' with $a' \equiv a \pmod{p}$ and $a' \equiv -a \pmod{q}$ leads to the ideal $\mathfrak{n}' := (N, a' + \sqrt{\delta})$ and $\mathfrak{n}' = \mathfrak{p}\bar{\mathfrak{q}}$, since we have that $a' + \sqrt{\delta} \equiv a + \sqrt{\delta} \equiv 0 \pmod{\mathfrak{p}}$, but $a' + \sqrt{\delta} \notin \mathfrak{q}$ as otherwise $q \mid (a - a')$. Finally, the root $N - a'$ leads to $\bar{\mathfrak{p}}\mathfrak{q}$. So choosing a at random is equivalent to choosing δ and \mathfrak{n} at random.

This finishes the proof as we can apply the above method k times for different choices of a . \square

Remark 1.6. Note that for a fixed N and a fixed δ , we are not able to find an ideal \mathfrak{n} as required in the proposition, since this would be equivalent to compute a square root of δ modulo N , which is known to be as hard as factoring N .

This means that for a fixed δ and for prime ideals \mathfrak{p} and \mathfrak{q} of $\mathbb{Q}(\sqrt{\delta})$ dividing split prime numbers p and q , we do not know whether $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$ is difficult to factor, but it seems likely that this is the case. Note, however, that if \mathfrak{n} is described by a set of generators, these generators might carry some information about the factorisation.

Example 1.7. Let us examine the case of quadratic fields more closely. Let $\delta \in \mathbb{Z}$ be square-free and let $\theta = \sqrt{\delta}$. An odd prime p not dividing δ can be either inert or totally split, depending on whether the polynomial $g(x) = x^2 - \delta$ is irreducible modulo p or not. This condition can easily be checked by computing the Legendre symbol $\left(\frac{\delta}{p}\right)$.

Let us take for example $\delta = -5$. The primes which are inert are exactly the primes $p \equiv 11, 13, 17, 19 \pmod{20}$; so the first two primes which split completely are $p = 3$ and $q = 7$. As explained above (up to the sign before $\sqrt{-5}$), we get $\mathfrak{p} = (3, 1 + \sqrt{-5})$ and $\mathfrak{q} = (7, 3 + \sqrt{-5})$. Since $1 + \sqrt{-5}$ is coprime to 7 and $3 + \sqrt{-5}$ is coprime to 3, we obtain that

$$\mathfrak{p}\mathfrak{q} = (21, (1 + \sqrt{-5})(3 + \sqrt{-5})) = (21, -2 + 4\sqrt{-5}).$$

When, like in this example, both primes split completely the factorisation algorithm described in the Proposition 1.4 does not apply, but still the factorisation is at most

as difficult as the factorisation of $N = pq$ in \mathbb{Z} . Since the security level remains at most the same, the involved computations are not easier to perform and the message space $\mathcal{O}_K/\mathfrak{p}\mathfrak{q}$ is isomorphic to $\mathbb{Z}/pq\mathbb{Z}$ (so, in particular, of the same size), this setting offers no advantages compared to the classical Rabin-cryptosystem and so we will not pursue this further.

Example 1.8. The next family of number fields we want to examine are cyclotomic fields. So let $m \in \mathbb{N}$ such that $m \not\equiv 2 \pmod{4}$ and let ζ_m be a primitive m -th root of unity. It is well-known that $\mathbb{Z}[\zeta_m]$ is the ring of integers in $\mathbb{Q}(\zeta_m)$, that the prime ideals which ramify in $\mathbb{Z}[\zeta_m]$ are those dividing m and for any other prime number p , the inertia degree of p is equal to its order modulo m . By Proposition 1.4 in our choice of p and q we need at least to make sure that their inertia degrees coincide; actually by the considerations in the above example, we would like to take both p and q to be inert, which is equivalent to p and q having order $\varphi(m)$ modulo m . This can be achieved (for infinitely many primes) if and only if m is a power of an odd prime or $m = 4$ (which is the case of the Gaussian integers $\mathbb{Z}[i]$).

Let us take for example $m = 7$. Then the prime numbers which are inert in $\mathbb{Z}[\zeta_7]$ are those congruent to 3 or 5 modulo 7, since 3 and 5 both have order 6 modulo 7.

2. SQUARE ROOTS MODULO IDEALS IN NUMBER FIELDS

Let K be a number field of degree d over \mathbb{Q} with ring of integers \mathcal{O}_K and let \mathfrak{p} and \mathfrak{q} be two distinct prime ideals of \mathcal{O}_K . Set $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$. As before we denote the rational primes below \mathfrak{p} and \mathfrak{q} by p and q , respectively. Since for our cryptographic applications p and q should be large, we can assume that they are both odd.

Note that we can compute p if the ideal \mathfrak{p} is given. Indeed in our applications \mathfrak{p} will be given by a set of generators, where the first is equal to p , but even if that is not the case, we can compute the norm $\mathcal{N}(\mathfrak{p})$, which is a (small) power of p . The exponent is at most d . From this, one can compute the prime p . If K is a Galois extension of \mathbb{Q} (as in any of the examples discussed above), we actually know that the exponent must be a divisor of d . Finally, if p is inert in K , the exponent is actually equal to d .

2.1. Chinese Remainder Theorem revisited. For our generalisation of Rabin's cryptosystem we need an algorithm that computes the roots of any square in $\mathcal{O}_K/\mathfrak{n}$, under the assumption that we know the prime factors \mathfrak{p} and \mathfrak{q} of \mathfrak{n} . The first tool is an explicit version of the Chinese Remainder Theorem for \mathcal{O}_K in which the involved isomorphism can be efficiently implemented on a computer. More precisely, we need to compute the inverse of the canonical isomorphism

$$(2) \quad \mathcal{O}_K/\mathfrak{n} \simeq \mathcal{O}_K/\mathfrak{p} \times \mathcal{O}_K/\mathfrak{q}, \quad m + \mathfrak{n} \mapsto (m + \mathfrak{p}, m + \mathfrak{q}).$$

In [1] the authors use the extended Euclidean algorithm in $\mathbb{Z}[i]$ to construct an inverse, but the ring of integers \mathcal{O}_K is not Euclidean in general. The following algorithm solves this problem by using only the fact that \mathbb{Z} is a Euclidean domain.

Algorithm 2.1 (Inverse of (2)). *Given $x + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ and $y + \mathfrak{q} \in \mathcal{O}_K/\mathfrak{q}$ this algorithm finds an $m \in \mathcal{O}_K$ which is mapped to the pair $(x + \mathfrak{p}, y + \mathfrak{q})$ under (2).*

- (1) *Compute the rational primes p and q below \mathfrak{p} and \mathfrak{q} , respectively.*
- (2) *Use the extended Euclidean algorithm in \mathbb{Z} to compute Bézout coefficients $a, b \in \mathbb{Z}$ of the equation $ap + bq = 1$.*
- (3) *Output $m = x bq + y ap \in \mathcal{O}_K$.*

2.2. Tonelli-Shanks for number rings. So we are left with the problem of computing square roots in $\mathcal{O}_K/\mathfrak{p}$ and $\mathcal{O}_K/\mathfrak{q}$, which can be done by a generalised version of the algorithm by Tonelli-Shanks. We let f be the inertia degree of \mathfrak{p} (the computation of f can be done efficiently). By definition, the residue field $\mathcal{O}_K/\mathfrak{p}$ has p^f elements, so $\mathcal{N}(\mathfrak{p}) = p^f$, and its group of units $(\mathcal{O}_K/\mathfrak{p})^\times$ has $p^f - 1$ elements.

Proposition 2.2 (Tonelli-Shanks for number rings). *Let $c + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ be a non-zero element which is a square. Choose a non-square $m + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ and $s, t \in \mathbb{N}$ such that $2^s \cdot t = p^f - 1$ and t is odd. Let $d + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ be the multiplicative inverse of $c + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$. Then there exists $i \in \{1, 2, \dots, 2^{s-1}\}$ such that*

$$(m + \mathfrak{p})^{2ti} = (d + \mathfrak{p})^t$$

and

$$((c + \mathfrak{p})^{(t+1)/2}(m + \mathfrak{p})^{ti})^2 = c + \mathfrak{p}.$$

Proof. Since $m + \mathfrak{p}$ is not a square and $(\mathcal{O}_K/\mathfrak{p})^\times$ is a cyclic group of order $2^s t$, the order of $(m + \mathfrak{p})^t$ is exactly 2^s . The order of $(d + \mathfrak{p})^t$ divides 2^{s-1} . The existence of i then follows from the fact that $(\mathcal{O}_K/\mathfrak{p})^\times$ is cyclic. The second formula is immediate. \square

We note that all the numbers required in the proposition can be computed efficiently, so that Proposition 2.2 provides an efficient way to compute a square root of $c + \mathfrak{p}$:

- (1) A non-zero element $x + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ is a square if and only if $(x + \mathfrak{p})^{(p^f-1)/2} = 1 + \mathfrak{p}$. This condition can be efficiently checked using the Square and Multiply Algorithm and since half of the elements are non-squares, it is easy to find a non-square element by very few trials (with a high probability).
- (2) To compute s and t we simply have to divide $p^f - 1$ by 2 as many times as possible and then stop. To find i efficiently by trials it is crucial that 2^s is not too large, which we expect to be the case for most of the choices of p .
- (3) To compute $d + \mathfrak{p}$ we first consider the product \tilde{c} of all the non-trivial Galois-conjugates of c in a Galois extension of \mathbb{Q} containing K . Then it is well-known that $c\tilde{c} \in \mathbb{Z}$ is prime to p and that $\tilde{c} \in \mathcal{O}_K$. Now we can solve Bezout's equation $c\tilde{c}x + py = 1$ for some $x, y \in \mathbb{Z}$. Then $d + \mathfrak{p} = \tilde{c}x + \mathfrak{p}$ is a multiplicative inverse of $c + \mathfrak{p}$.

2.3. Fast square root extraction. In the case of the classical Rabin scheme, i.e. in the case $K = \mathbb{Q}$, there is a more efficient way than the Tonelli-Shanks algorithm to compute square roots modulo a prime p provided that $p \equiv 3 \pmod{4}$. For general number fields, we provide the following generalisation.

Proposition 2.3. *If $|\mathcal{O}_K/\mathfrak{p}| \equiv 3 \pmod{4}$ and $c + \mathfrak{p} \in \mathcal{O}_K/\mathfrak{p}$ is a non-zero element which is a square, then*

$$(c + \mathfrak{p})^{(|\mathcal{O}_K/\mathfrak{p}|+1)/4}$$

is a square root of $c + \mathfrak{p}$.

Proof. Since $c + \mathfrak{p}$ is a square, its order divides $(|\mathcal{O}_K/\mathfrak{p}| - 1)/2$. Therefore we have that

$$((c + \mathfrak{p})^{(|\mathcal{O}_K/\mathfrak{p}|+1)/4})^2 = (c + \mathfrak{p})^{(|\mathcal{O}_K/\mathfrak{p}|+1)/2} = (c + \mathfrak{p})^{(|\mathcal{O}_K/\mathfrak{p}|-1)/2}(c + \mathfrak{p}) = c + \mathfrak{p},$$

as desired. \square

By the discussion in the previous section, we are mainly interested in primes which are inert and by the above proposition we have a gain in efficiency in the computations if we restrict to the case $|\mathcal{O}_K/\mathfrak{p}| \equiv 3 \pmod{4}$. Unfortunately, these two conditions are incompatible in quadratic number fields, since then for an inert odd prime p , $|\mathcal{O}_K/\mathfrak{p}| = p^2 \equiv 1 \pmod{4}$. This motivates us to examine cubic number fields as $p^3 \equiv p \pmod{4}$. Since we also want an efficient way of checking if a prime is inert, it turns out that taking fields of the form $\mathbb{Q}(\sqrt[3]{d})$ might not be the optimal choice.

Example 2.4. Some very interesting examples of cubic extensions with good characterisations for families of inert primes are subfields of cyclotomic extensions.

The first example we consider is the maximal real subfield of $\mathbb{Q}(\zeta_7)$, i.e. the field $\mathbb{Q}(\zeta_7 + \bar{\zeta}_7)$. By [9, Proposition 2.16] the ring of integers of that field is $\mathbb{Z}[\zeta_7 + \bar{\zeta}_7]$. We can easily compute the minimal polynomial of $\zeta_7 + \bar{\zeta}_7$, namely $x^3 + x^2 - 2x - 1$. The primes which are inert in $\mathbb{Q}(\zeta_7 + \bar{\zeta}_7)$ are those with inertia degree divisible by 3 in $\mathbb{Q}(\zeta_7)$. We have already seen that these are the primes whose classes modulo 7 have an order which is a multiple of 3, namely those congruent to 2, 3, 4 or 5 modulo 7. In order to use the simplified computation of square roots above, we restrict to primes congruent to 3 modulo 4. Summarizing, we want to take prime numbers congruent to 3, 11, 19 or 23 modulo 28.

Similarly, we can consider the maximal real subfield of $\mathbb{Q}(\zeta_9)$, i.e. $\mathbb{Q}(\zeta_9 + \bar{\zeta}_9)$. The minimal polynomial of $\zeta_9 + \bar{\zeta}_9$ is $x^3 - 3x + 1$ and a family of inert primes consists of those which are congruent to 2, 4, 5 or 7 modulo 9. This leads to considering the primes congruent to 7, 11, 23 or 31 modulo 36.

This can be done for every cyclotomic field whose degree is a multiple of 3. From $\mathbb{Q}(\zeta_{13})$ we obtain the extension given by a root of $x^3 + x^2 - 4x + 1$; from $\mathbb{Q}(\zeta_{19})$ we get $x^3 + x^2 - 6x - 7$. Note that in this last example, the largest power of 3 dividing the degree of the cyclotomic field over \mathbb{Q} is 9, so in order to make sure that a prime is inert in the subfield of degree 3 we need to require that the order of p modulo 19 is a multiple of 9, not just of 3, which leads to the primes congruent to 2, 3, 4, 5, 6, 9, 10, 13, 14, 15, 16, 17 modulo 19. Together with the usual requirement $p \equiv 3 \pmod{4}$, we get the acceptable congruences to 3, 15, 23, 35, 43, 47, 51, 55, 59, 63, 67, 71 modulo 76.

2.4. Square roots and factoring. We end this section with an examination of the converse problem, which will be relevant in the discussion about the security of our cryptosystem.

Proposition 2.5. *Let K be a number field and let $\mathfrak{n} = \mathfrak{p}q$ be the product of two distinct prime ideals in \mathcal{O}_K . Suppose $m_1 + \mathfrak{n}$ and $m_2 + \mathfrak{n}$ are both square roots of the same element $c + \mathfrak{n} \in \mathcal{O}_K/\mathfrak{n}$ and that $m_1 + \mathfrak{n} \neq \pm m_2 + \mathfrak{n}$. Then we can efficiently compute a factorisation of \mathfrak{n} .*

Proof. By the Chinese Remainder Theorem (2), m_1 and m_2 are congruent to each other modulo one of the prime divisors of \mathfrak{n} and to the opposite of each other modulo the other one. Hence $m_1 + m_2$ belongs to exactly one of the prime divisors of \mathfrak{n} , say \mathfrak{p} . Then the norm n of $m_1 + m_2$ is divisible by p but not by q . Hence the greatest common divisor of n and the norm $\mathcal{N}(\mathfrak{n})$ is a power of p . The exponent is bounded by the degree d so that we can find p . Since $\mathcal{N}(\mathfrak{n})$ is a product of a small power of p and a small power of q , we also obtain q . Once we have p and q , we can compute their decompositions into prime ideals in \mathcal{O}_K as described in §1.2. Therefore the two prime factors of \mathfrak{n} must belong to two families of prime ideals $\mathfrak{p}_1, \dots, \mathfrak{p}_{n_p}$ and $\mathfrak{q}_1, \dots, \mathfrak{q}_{n_q}$ with $n_p, n_q \leq d$. To decide which of the factors actually divide \mathfrak{n} , we can

use the algorithm described in [2, §4.8.3] to compute the valuation of \mathfrak{n} with respect to the different prime ideals. \square

3. CHOOSING THE RIGHT SQUARE ROOT

An important issue of the Rabin-cryptosystem, even in its classical version over \mathbb{Z} , is the missing unicity in the decryption process: While encrypting corresponds to squaring, decrypting means to compute a square root modulo $N = pq$, and in general there are four of them. This question has been addressed in [4], where the authors describe several ways of transmitting the required information in two extra bits. The easiest method consists in taking as extra bits the parity of the message m and its Legendre symbol over N . This system, which was originally proposed in [10], is shown to work when both p and q are congruent to 3 modulo 4, which, as seen before, is a good choice also in order to simplify the computation of the square roots. Of course, using some paddings with random bits, one has to make sure that the two extra bits do not leak any information about the message, which would compromise semantic security.

We will now see how to handle the issue of identifying the right square root in the generalised setting treated in this paper. The requirement of taking inert primes p and q turns out to be very helpful also at this stage, so we will keep assuming it. Assume that the message we encrypt is of the form $m = \sum_{i=0}^{d-1} a_i \theta^i$, where $a_i \in \mathbb{Z}/N\mathbb{Z}$ for $0 \leq i < d$. Then the decryption process leads to four possible square roots m_1, m_2, m_3, m_4 of $c = m^2$. Write $m_j = \sum_{i=0}^{d-1} a_{i,j} \theta^i$ with $a_{i,j} \in \mathbb{Z}/N\mathbb{Z}$. Then each coefficient $a_{i,j}$ must be congruent to $\pm a_i$ modulo p and modulo q , possibly with opposite signs. In any case $a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}$ are exactly the four square roots of a_i^2 modulo N . Therefore to determine which of the possible messages m_1, m_2, m_3, m_4 is the original message m is equivalent to determining which of the coefficients $a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}$ coincides with a_i . This means that we can literally use the same methods discussed in [4] and apply them to a single component of the message, for example the one of θ^0 .

There is still one detail which needs to be addressed: some of the $a_{0,1}, a_{0,2}, a_{0,3}, a_{0,4}$ might coincide, which happens if and only if any one of the numbers $a_{0,j}$ is congruent to 0 modulo p or modulo q . If it is 0 modulo both p and q , then it is 0 modulo N and so are all of the $a_{0,1}, a_{0,2}, a_{0,3}, a_{0,4}$, as well as a_0 . If this is the case, then we should just look at the next coefficient a_1 and store the corresponding bits for this component. The second case is when $a_{0,j}$ is 0 modulo p but not modulo q , or vice versa. Having such a number is equivalent to knowing the factorisation of N , by just computing $\gcd(N, a_{0,j})$ with the Euclidean algorithm. This means that under our assumption that N can not be factored, we conclude that the described situation will never happen in practice.

4. THE RABIN CRYPTOSYSTEM IN NUMBER FIELDS

We can now introduce the specifications of our generalised Rabin cryptosystem to number fields.

4.1. Key generation. First of all we need to choose a number field. This choice could be part of the public key, but there is actually no reason for using a different field every time; indeed, fixing a field K once and for all permits to increase the efficiency of key generation by doing some preliminary computations.

We will distinguish and compare three different implementations for the following families of number fields: quadratic fields, (subfields of) cyclotomic fields, general

number fields. Since for the key generation algorithm, we want to choose prime numbers that are inert in K , it is useful to find some congruence conditions which are easy to verify. Let us see how to do this in the three mentioned cases.

Case 1. A quadratic field $K = \mathbb{Q}(\sqrt{\delta})$, where δ is a square-free integer, which is not too big. A sufficiently large prime p will not ramify in K and it is inert if and only if $x^2 - \delta$ is irreducible modulo p , i.e. if and only if $\left(\frac{\delta}{p}\right) = -1$. Using quadratic reciprocity and its supplements, we can find a list of congruences modulo δ , 4δ or 8δ which p must satisfy.

Case 2. A cyclotomic field $K = \mathbb{Q}(\zeta_m)$, where $m = \ell^k$ is a perfect power of an odd prime number ℓ . Then a prime p is inert if and only if its class modulo m is a primitive element modulo m . By computing one primitive element modulo m and taking all of its powers with exponents coprime to $\varphi(m) = (\ell - 1)\ell^{k-1}$, we find all the possible primitive elements and, as in case 1, we obtain a list of possible congruences modulo m which p has to satisfy.

If $\ell - 1 = 3^a b$ with $a > 0$ and $3 \nmid b$, then there is a unique subfield K_3 of K of degree 3 over \mathbb{Q} . A prime p is inert in K_3 if and only if its inertia degree in K is divisible by 3^a . Again this leads to a list of possible congruences modulo m .

Case 3. A generic number field K is defined by the minimal polynomial g of a primitive element θ of K . Then the prime numbers p which are inert are characterised as those primes modulo which g remains irreducible. This condition needs to be checked prime by prime in the key generation algorithm, making the computation less efficient. It might indeed happen that for a given number field we do not find any suitable primes.

Let us describe the key generation algorithm for a fixed field K . We need to choose two prime numbers p and q satisfying the conditions described above and such that their product is not factorisable efficiently with any known algorithm. In particular we need to make sure that p and q are sufficiently large and are such that $p - 1$ and $q - 1$ have some large primes among their factors. Here is a good way of achieving this if we have some congruence conditions modulo some integer D like in case 1 and case 2: choose a random value c among the allowed congruences, choose random numbers k larger than 2^λ , where λ is a fixed security parameter, until $\ell = 2kD + 1$ is prime. Finally take $h = 0$ and increase it by 1 until $p = (hD + c - 1)\ell + 1$ is prime. By construction p is a large prime congruent to c modulo D and such that $p - 1$ is divisible by the large prime ℓ . The same construction is used to choose q . The couple (p, q) is the private key, the product $N = pq$ is the public key. In the generic case 3 we can just choose random primes p and q and need to check prime by prime whether the polynomial g factorises modulo p and q ; this should be quite time consuming.

In the cases where we cannot apply the fast square root extraction of §2.3, the two non-squares which are required for the Tonelli-Shanks algorithm can be determined once and then stored for further use as part of the private key. This speeds up decryption slightly.

4.2. Rabin scheme over number rings. Suppose we have given a monic irreducible polynomial $g \in \mathbb{Z}[x]$ of degree d and a public key $N = pq$, where p and q are two distinct primes, which are inert in $K = \mathbb{Q}[x]/(g)$ and congruent to 3 modulo 4. We let $\theta := x \pmod{g}$ so that $\mathbb{Z}[\theta] = \mathbb{Z}[x]/(g)$. If $0 \neq m = \sum_{i=0}^{d-1} a_i \theta^i \in \mathbb{Z}[\theta]/(N)$ with $a_i \in \mathbb{Z}/N\mathbb{Z}$, $0 \leq i < d$, then we write $\alpha(m)$ for the first non-zero coefficient of

m , i.e. $\alpha(m) := a_{i_0}$, where $a_{i_0} \neq 0$ and $a_i = 0$ for all $i < i_0$. In our implementation we will assume for simplicity that $a_0 \neq 0$.

Encryption. Suppose Alice wants to encrypt a (non-zero) message $m = \sum_{i=0}^{d-1} a_i \theta^i$ with $a_i \in \mathbb{Z}/N\mathbb{Z}$, $0 \leq i < d$. Then she computes $c \equiv m^2 \pmod{N}$, the parity $b_0 = \alpha(m) \pmod{2}$ of $\alpha(m)$ and $b_1 = \frac{1}{2} \left(1 - \left(\frac{\alpha(m)}{N} \right) \right) \in \{0, 1\}$ and transmits the triple (c, b_0, b_1) to Bob.

Decryption. Bob computes the square roots of $c \pmod{p}$ and $c \pmod{q}$ either with the Tonelli-Shanks algorithm (Proposition 2.2) or - whenever possible - via Proposition 2.3. He computes the Legendre symbols modulo p and modulo q of the first non-zero coefficients of these square roots. Via Algorithm 2.1 he combines two of the latter to a square root m_1 of $c \pmod{N}$ having the correct value of b_1 . Note that $-m_1$ then leads to the same value of b_1 and is also a square root of $c \pmod{N}$. The original message is the one among m_1 and $-m_1$ having the correct parity bit b_0 .

We provide an implementation of the described algorithms for the settings described above in PARI on GitHub. We also included an implementation of the classical Rabin cryptosystem in order to guarantee a fair comparison of running times.¹

5. SECURITY AND EFFICIENCY

In this section we examine some aspects of the implementation in more detail and compare running times of our algorithms in different cases.

As for the key generation algorithm we have three different scenarios: in case 1 we can use a list of congruences for p and q and we do the pre-calculations for the Tonelli-Shanks algorithm, in case 2 we still have a list of congruences but we do not need any pre-calculations, in case 3 we do not have a list of congruences and we do pre-calculations. Concerning the running time, the pre-calculations required for the Tonelli-Shanks algorithm are quite negligible and the number of candidates on which to perform a primality test is on average the same as for the classical Rabin cryptosystem or for RSA. The general case 3 is much slower because we have to discard quite a lot of primes because they are not inert.

The encryption function is literally the same in all cases, since it just has to compute a square in the relevant ring of integers modulo the public key. Note, however, that the running time increases with the degree of the number field, since we need to compute a square in a larger ring, whose elements are represented by polynomials. We performed experiments over the Gaussian integers and the cubic number field defined by the polynomial $x^3 + x^2 - 2x - 1$. The running time was in average about 1.1 and 1.25 times as long as over \mathbb{Z} , respectively. This is actually very favourable, since there are twice or even three times as many bits available in the plaintext space.

Concerning decryption there are only two different scenarios: one using the Tonelli-Shanks algorithm and one using the simplified version of Proposition 2.3, which in particular works in our degree 3 example. The running time of decryption over Gaussian integers turned out to be about 23 times slower than in the classical Rabin cryptosystem. Almost the whole time is required for the Tonelli-Shanks algorithm, which was our motivation to consider fields of degree 3. In our example, the factor we measured was slightly below 30. Taking the ratio with the number of bits of the

¹<https://github.com/alecobbbe/RabinNF>

message this is, as expected, better than over Gaussian integers, but it is still less efficient than the classical Rabin cryptosystem.

Let us explain what happens from a more theoretical point of view. Computing the e -th power of a number modulo p with the square and multiply algorithm means computing $c \log(e)$ multiplications within $\mathbb{Z}/p\mathbb{Z}$ for some constant c . If \mathcal{O}_K is the ring of integers of a number field of degree 3 over \mathbb{Q} , then the computation of an e -th power in $\mathcal{O}_K/(p)$ corresponds to $9c \log(e)$ multiplications in $\mathbb{Z}/p\mathbb{Z}$. However, the powers we need to compute over \mathbb{Z} and \mathcal{O}_K have exponents $\frac{p+1}{4}$ and $\frac{p^3+1}{4}$, respectively. Having a third power in the second exponent implies another factor of 3 in the required number of multiplications in $\mathbb{Z}/p\mathbb{Z}$. To conclude, this analysis would explain a factor of 27; the actually observed factor is just slightly larger.

To conclude, key generation takes about the same time for the classical Rabin cryptosystem and its generalisations, provided we have a list of congruences that guarantee that a prime p is inert in the considered field. The encryption is extremely fast and becomes even more efficient in number fields, while decryption becomes less efficient in general number fields, even though some extensions of degree 3 of the rational numbers perform better than Gaussian integers.

As for security, we should say that our implementation was mainly conceived to examine running times, but it uses the random function of PARI/GP, which is not cryptographically secure. Besides that issue, we have already discussed that being able to decrypt any ciphered message is equivalent to being able to factorise the public key into its prime factors, exactly as for the classical Rabin cryptosystem. In particular, it cannot be used for post quantum cryptography since an attacker having access to a quantum computer can factor the modulus via Shor's algorithm [8]. To achieve semantic security, one should also introduce some random paddings, just as for every public key scheme. We additionally point out that we are transmitting two extra-bits carrying information about the degree zero term of the polynomial representing our message; so that is where the padding should happen.

6. CONCLUSION

In this article we proposed a novel public-key cryptosystem which extends the classical Rabin scheme to general number fields. We have seen that encryption performs very fast, but decryption is less efficient than for the classical Rabin scheme. However, a more sophisticated variant for degree 3 number fields at least performs relatively better than the case of quadratic number fields. Because public key cryptosystems are generally used only to exchange keys for a more efficient symmetric encryption scheme, their slower performance is typically not a significant disadvantage.

Our new scheme is provably equivalent to the factorisation problem of a large integer N if the prime ideals \mathfrak{p} and \mathfrak{q} dividing the modulus \mathfrak{n} are chosen carefully. In particular, prime ideals generated by rational inert primes have this property and even further desirable side effects. These observations also hold for the RSA scheme over rings of integers in number fields.

REFERENCES

- [1] Y. Awad, A. N. El-Kassar, and T. Kadri. Rabin public-key cryptosystem in the domain of Gaussian integers. In *2018 International Conference on Computer and Applications (ICCA)*, pages 336–340, 2018.
- [2] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

- [3] H. Cohen. *Advanced topics in computational number theory*, volume 193 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.
- [4] M. Elia, M. Piva, and D. Schipani. The Rabin cryptosystem revisited. *Appl. Algebra Engrg. Comm. Comput.*, 26(3):251–275, 2015.
- [5] K. Mahler. An inequality for the discriminant of a polynomial. *Michigan Math. J.*, 11:257–262, 1964.
- [6] K. A. Petukhova and S. N. Tronin. RSA cryptosystem for Dedekind rings. *Lobachevskii J. Math.*, 37(3):284–287, 2016.
- [7] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, USA, 1979.
- [8] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [9] L. C. Washington. *Introduction to cyclotomic fields*, volume 83 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1997.
- [10] H. C. Williams. A modification of the RSA public-key encryption procedure. *IEEE Trans. Inform. Theory*, 26(6):726–729, 1980.

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN, FAKULTÄT FÜR INFORMATIK, WERNER-HEISENBERG
WEG 39, 85579 NEUBIBERG, GERMANY
Email address: `alessandro.cobbe@unibw.de`

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN, FAKULTÄT FÜR INFORMATIK, WERNER-HEISENBERG
WEG 39, 85579 NEUBIBERG, GERMANY
Email address: `andreas.nickel@unibw.de`
URL: <https://www.unibw.de/timor/mitarbeiter/univ-prof-dr-andreas-nickel>

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN, FAKULTÄT FÜR INFORMATIK, WERNER-HEISENBERG
WEG 39, 85579 NEUBIBERG, GERMANY
Email address: `akay.schuster@unibw.de`