New public-key cryptosystems with fast decryption

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr.rer.nat)

Von
Tsuyoshi Takagi
(Master of Science)
aus Nagoya, Japan

Referenten: Prof. Dr. Johannes Buchmann
Prof. Dr. Kouichi Sakurai

Tag der Einreichung: 08.12.2000
Tag der mündlichen Prüfung: 30.01.2001

Darmstadt 2001
D 17

Acknowledgements

First of all, I would like to thank Prof. Johannes Buchmann for giving me the opportunity to join his research group, organizing the joint research between TUD and NTT, and promoting this doctor thesis as my supervisor. His suggestions and helpful support improve this work significantly. I would also like to thank Prof. Kouichi Sakurai for accepting the task of the second referee. He gives me several important comments for this doctor thesis.

Second I thank Dr. Sachar Paulus, Dr. Ingrid Biehl for inspiring this work and having everlasting discussions, Detlef Hühnlein, Dr. Michael Jacobson,Jr., Andreas Meyer, Michael Hartmann for collaborating the research of quadratic fields, Prof. Jean-Jacques Quisquater, Dr. Marc Joye for exchanging many interesting subjects about the RSA cryptosystem.

Third I acknowledge Dr. Haruhisa Ichikawa for giving me the chance to visit TUD, Shozo Naito for having helpful discussions on my work, Toru Kobayashi for supporting the joint research between TUD and NTT.

Finally I wish to thank my family and friends, especially my wife Julia, for all their encouragement and support, without which I would never have completed this work.

Contents

List of algorithms

Chapter 1

Preface

This thesis is about the construction, analysis and implementation of efficient public-key cryptosystems. The public-key cryptosystem which is used most frequently throughout the world is the RSA cryptosystem [RSA78] [PKCS]. As an alternative to the RSA cryptosystem, elliptic curve cryptosystems have been introduced [Kob87] [Mil86]. Here we are interested in constructing new public-key cryptosystems which are different from both the RSA cryptosystem and elliptic curve cryptosystems.

The theoretical foundation for the construction of a public-key cryptosystem is the following statement: A public-key cryptosystem has an encryption function $\mathcal{E}$ and a decryption function $\mathcal{D}$ which are published for all persons. Each person has a public key e and the corresponding secret key d. When person B wishes to send a message m to person A, then B uses the public-key e_A of A to encrypt m:

$$c = \mathcal{E}_{e_A}(m), \quad \mathcal{E}: \text{ a encryption function with } e_A. \tag{1.1}$$

Ciphertext c is sent to person B possibly through an open network. Then person A decrypts the ciphertext c using the secret key d_A which only person A knows:

$$m = \mathcal{D}_{d_A}(c), \quad \mathcal{D}: \text{ a decryption function with } d_A. \tag{1.2}$$

Here we have the relationship $m = \mathcal{D}_{d_A}(\mathcal{E}_{e_A}(m))$. All persons can compute $\mathcal{E}_{e_A}(\cdot)$ using the public key e_A, but the only person who can execute $\mathcal{D}_{d_A}(\cdot)$ is A. Using this mechanism, we can distribute secret keys securely over the open network.

From a computational point of view, it is an interesting problem to find public key cryptosystems. The following criteria should be satisfied:

1. Generating a pair of a public key e and a corresponding secret key d should be efficient, i.e., polynomial time.
2. Both to compute the secret key d for a given public key e and to compute the message m for a given ciphertext $\mathcal{E}_e(m)$ without knowing d should be computationally infeasible.

3. Encryption function $\mathcal{E}$ and decryption function $\mathcal{D}$ should be sufficiently efficient, i.e., polynomial time where the degree of the polynomial is ≤ 3.

The construction of a public-key cryptosystem currently requires the use of a number theoretical structure. For example, the RSA cryptosystem is constructed over the integer ring $\mathbb{Z}/n\mathbb{Z}$, where n is a product of two primes. Elliptic curve cryptosystems are constructed over elliptic curves over some finite fields $\mathbb{F}_n$, where usually n is either a prime or a power of 2. After the appearance of the RSA cryptosystem, many mathematicians and cryptographers have become interested in finding another suitable structure for a public-key cryptosystem, and many public-key cryptosystems have been proposed such as the McEliece cryptosystem [Mc78], the knapsack cryptosystems [MH78] [CR88], the quadratic field cryptosystem [BW88], or the Ajtai-Dwork cryptosystem [AD97]. However these public-key cryptosystems have over time become already impractical compared with the RSA cryptosystem or the elliptic curve cryptosystem. This is due to the invention of faster cryptanalysis algorithms, which have been derived from new results of computational number theory [MvOV97] [N00]. It is a very interesting problem to construct a public-key cryptosystem with a new trapdoor, which is based on a different number theoretical structure.

The decryption function of the RSA cryptosystem or an elliptic curve cryptosystem involve exponentiation of an element, which is a relatively slow computation, i.e., polynomial time with a polynomial of degree = 3. It has been said that computation of their decryption functions is generally more than 100 times slower than that of typical symmetry-key cryptosystem. In addition, for the sake of high security it is desirable to store the secret keys on a smart card and to carry out the decryption computation on the smart card. A smart card has limited computational resources, and one often needs a costly coprocessor which helps with calculation of the decryption functions. On the contrary we do not have to use a coprocessor to implement a symmetry-key cryptosystem on a smart card. Currently smart cards that are used for a large scale market such, as cash cards or SIM cards for mobile phones, do not use a public-key cryptosystem because of the high cost. Therefore, we desire a public-key cryptosystem with fast decryption, i.e., polynomial time with a polynomial of degree = 2.

Contributions of this Thesis

In this doctoral thesis, we propose three public-key cryptosystems with fast decryption function: the **NICE cryptosystem**, the **PkQ cryptosystem** and the **Nk cryptosystem**. The NICE cryptosystem is constructed over non-maximal quadratic orders [PT00]. The PkQ cryptosystem is constructed over $\mathbb{Z}/p^kq\mathbb{Z}$, where p, q are primes and k is a positive integer [Tak98]. The Nk cryptosystem is constructed over $\mathbb{Z}/n^k\mathbb{Z}$, where n is the RSA-modulus and k is a positive integer

[Tak97]. These three public-key cryptosystems are not only of theoretical interest but also practical one. The NICE cryptosystem and the PkQ cryptosystem are suitable for implementation on smartcards.

The NICE cryptosystem is constructed over quadratic orders [PT00]. The NICE cryptosystem has two interesting properties. The first property is that the trapdoor mechanism is different from previously reported public-key cryptosystems such as RSA cryptosystem [RSA78] and elliptic curve cryptosystems [Kob87] [Mil86]. The second property is that the decryption process of the NICE cryptosystem is very fast. It is of quadratic bit complexity in the length of the public key, i.e., polynomial time with a polynomial of degree = 2. The NICE cryptosystem is the only known public-key cryptosystem whose decryption has quadratic polynomial time. Our implementation shows that with regards to the decryption time, it is comparably as fast as the encryption time of the RSA cryptosystem with small encryption exponent $e = 2^{16} + 1$. The security of our cryptosystem is closely related to factoring the discriminant of a quadratic order. When we choose appropriate sizes of the parameters, the currently known fast algorithms for cryptanalysis such as the number field sieve [LL91], the elliptic curve method [Len87], or the Hafner-McCurley algorithm [HM89] are not applicable.

The PkQ cryptosystem is constructed over $\mathbb{Z}/p^k q\mathbb{Z}$, where p, q are primes and k is a positive integer [Tak98]. The prominent property of the PkQ cryptosystem is its decryption time. It is about three times faster than implementations of the RSA cryptosystem that use the Chinese remainder theorem [QC82] [PKCS]. Indeed, timings for implementations using LiDIA [LiD95] show that the PkQ cryptosystem is about 2.4 times faster than the RSA cryptosystem with the Chinese remainder theorem. The security of the PkQ cryptosystem is closely related to the RSA cryptosystem. We prove that standard attacks against the RSA cryptosystem, for examples, the cycling attack [WS79] and the low decryption exponent attack [Wie90] are not applicable to the PkQ cryptosystem. Moreover, to implement the PkQ cryptosystem we do not have to prepare extra cryptographic libraries; instead, we can use the standard one for the RSA cryptosystem. We can easily implement the PkQ cryptosystem in an environment designed for the RSA cryptosystem.

The Nk cryptosystem is constructed over $\mathbb{Z}/n^k\mathbb{Z}$, where n is the RSA modulus and k is a positive integer [Tak97]. The features of the Nk cryptosystem are as follows: We can encrypt a message which is several time larger than the RSA modulus. We can prove that breaking the second block of the Nk cryptosystems is as hard as breaking the original RSA cryptosystem. To implement the Nk cryptosystems, we used only ordinary and elementary mathematical techniques such as computation of greatest common divisors, so that it is easy to implement. Moreover, the decryption time of the first block is dominant, because after the first block we only calculate the modular multiplication of the encryption exponent and an extended Euclidean algorithm to decrypt blocks after the first one. Therefore the Nk cryptosystem is faster compared with the previously reported RSA-type cryptosystems [Dem94] [KMOV92] [Koy95] [LKBS92] [MM96] [SE96]. Even if a message is several times

longer than a public-key n, we can encrypt the message fast without additionally using a symmetry-key cryptosystem.

Recent related works

In the following we report several recent works related to my doctor thesis.

The NICE cryptosystem is constructed over quadratic orders $\mathcal{O}_{\Delta_q}$, where $\Delta_q = \Delta_1 q^2$ is a non-fundamental discriminant, Δ_1 is the fundamental discriminant of Δ_q, and q is a prime. The security of the NICE cryptosystem is based on the factoring problem of the non-fundamental discriminant $\Delta_q = \Delta_1 q^2$. On the other hand, a public-key cryptosystem based on the discrete logarithm problem of the class group $Cl(\Delta)$ has also proposed [BW88]. Vollmer rigorously proved that the discrete logarithm problem can be solved in time $L_{|\Delta|}[\frac{1}{2}, \frac{3}{4}\sqrt{2}]$ under the generalized Riemann hypothesis assumption [Vo00]. Hamdy and Möller investigated the required size of the discriminants for cryptographic purposes [HM00]. They estimate that the time to factor a 1024-bit RSA modulus by the number field sieve is equivalent to the time to solve the discrete logarithm problem of the class group $Cl(\Delta)$ with a 687-bit discriminant Δ. Hühnlein and Takagi proved that if a discriminant $\Delta_q = \Delta_1 q^2$ is a totally non-fundamental discriminant, i.e. $h(\Delta_1) = 1$ then the discrete logarithm problem of the class group $Cl(\Delta_q)$ can be reduced to that of the finite group $\mathbb{F}_{q^2}$ [HT99]. Using this reduction, Hühnlein proposed a digital signature scheme over $Cl(-8q^2)$ with a fast signature generation process [Hu99]. His implementation shows that it is faster than the Digital Signature Standard (DSS) [DSA94] with the same security parameter size.

The PkQ cryptosystem is constructed over $\mathbb{Z}/p^k q\mathbb{Z}$, where p, q are primes and $k = 2, 3, ...$ Recently, Lim at el. extended it to the modulus $p^r q^s (r, s = 2, 3, ...)$ [LKLY00]. RSA Laboratories presented Multi-Prime RSA in Public-Key Cryptography Standards #1 version 2.0 Amendment [PKCS]. Multi-Prime RSA uses the product of more than two primes as RSA modulus. Multi-Prime RSA with 3 primes is about 9 times faster than the original RSA cryptosystem (See also Section 3.4). Boneh, Durfee, and Howgrave-Graham present an algorithm for factoring integers of the form $n = p^k q$ for large k [BDH-G99]. The algorithm runs in time polynomial in $\log n$ if k is close to $\log p$. The algorithm dose not work for the 1024-bit modulus $n = p^2 q$ with 341-bit primes p and q, because 2 is much smaller than $\log p$. Boneh and Durfee extended the Wiener's attack (see Section 3.6) using lattice reduction theory [BD00]. The secret exponent d with $d < n^{0.292}$ can be heuristically detected. They conjecture that the bound would be eventually improved to $d < n^{0.5}$. It is an open problem how we apply the Boneh-Durfee method to the relation $ed \equiv 1 \bmod \varphi(p^k q)$, where $\varphi(p^k q) = p^{k-1}(p-1)(q-1)$.

The Nk cryptosystem is constructed over $\mathbb{Z}/n^k\mathbb{Z}$, where n is the RSA modulus and $k = 2, 3, ...$ The Nk cryptosystem is considered a multi-block mode usage of the

RSA cryptosystem. The most famous single-block mode of the RSA cryptosystem is the OAEP which is used in the Public-Key Cryptography Standards #1 version 2.0 [PKCS]. Using the OAEP we can prove that the RSA cryptosystem is semantically secure against adaptive chosen message attacks under the random oracle model. Recently Pointcheval proposed a multi-block mode of the RSA cryptosystem based on the dependent-RSA problem [Po99]. The dependent-RSA problem is a problem to distinguish two distributions in $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$:

$$(a^e \bmod n, b^e \bmod n), \quad (a^e \bmod n, (a+1)^e \bmod n)$$

where n is the RSA modulus, e is a public exponent of the RSA, and a, b are random elements of $\mathbb{Z}/n\mathbb{Z}$. Pointcheval's cryptosystem can be proven semantically secure against adaptive chosen message attacks under the dependent RSA problem and the random oracle model. Paillier proposed a public-key cryptosystem based on composite degree residuosity classes [Pa99]. The trapdoor of the cryptosystem is an extension of the EPOC cryptosystem [OU98] to the ring $\mathbb{Z}/n^2\mathbb{Z}$, where n is a product of two primes. The intractability to break this cryptosystem is based on the composite residuosity problem: to find the integer x for given $g, n, g^x r^n \bmod n^2$, where $g, r \in (\mathbb{Z}/n^2\mathbb{Z})^\times$ with $n | \mathrm{ord}_{n^2}(g)$. Paillier and Pointcheval proposed a variant of the Paillier cryptosystem that can be proven semantically secure against adaptive chosen message attacks [PP99]. For the Nk cryptosystem we proved that the oracle which computes the second block of a plaintext can also break the first block of it. However it is an open problem whether we can prove that the Nk cryptosystem is semantically secure against adaptive chosen attackers under some assumptions like the Pointcheval cryptosystem and the Paillier-Pointcheval cryptosystem.

This thesis is organized as follows.

In Chapter 2, we present the NICE cryptosystem. At first we review the arithmetic of non-maximal quadratic orders and we show several algorithms used in the NICE cryptosystem. In Section 2.3, key generation, encryption and decryption are explained. We also analyze the security of the NICE cryptosystem. In Section 2.5 we show the timings of implementation by software. In Section 2.6 we also show the timings of implementation by a smartcard.

In Chapter 3 we propose the PkQ cryptosystem. In Section 3.2 we describe the algorithm of the PkQ cryptosystem. Then the running time of its decryption algorithm is estimated. We also discuss the security of the PkQ cryptosystem, especially factoring algorithms for the modulus p^2q and the short secret exponent attack.

In Chapter 4 we present the Nk cryptosystem and analyze its security. Indeed we prove that deciphering the entire plaintext of this system is as intractable as breaking the RSA cryptosystem. We also analyze the efficiency of the Nk cryptosystem. The decrypting process is faster than any other multi-block RSA-type cryptosystem ever reported.

Chapter 2

NICE Cryptosystem

We present a new cryptosystem based on ideal arithmetic in quadratic orders. The method of our trapdoor is different from the Diffie-Hellman key distribution scheme or the RSA cryptosystem. The plaintext $\mathfrak{m}$ is encrypted by $\mathfrak{m}\mathfrak{p}^r$, where $\mathfrak{p}$ is a fixed element and r is a random integer, so our proposed cryptosystem is a probabilistic encryption scheme and has the homomorphy property. The most prominent property of our cryptosystem is the cost of the decryption, which is of quadratic bit complexity in the length of the public key. Our implementation shows that it is comparably as fast as the encryption time of the RSA cryptosystem with $e = 2^{16} + 1$. The security of our cryptosystem is closely related to factoring the discriminant of a quadratic order. When we choose appropriate sizes of the parameters, the currently known fast algorithms, for examples, the elliptic curve method, the number field sieve, the Hafner-McCurley algorithm, are not applicable.

2.1 Introduction

Plenty of public key cryptosystems have been proposed, and the Diffie-Hellman key distribution scheme or the RSA cryptosystem are mostly used throughout the world [DH76] [RSA78]. Typically, these public key cryptosystems involve a modular exponentiation with a large number, which is of cubic bit complexity in the bit length of the public key and its computation is relatively slow. On the other side, for the sake of high security the secret keys are stored on a smart card and the decryption computation is also carried out over the smart card. So a cryptosystem with fast decryption is desired. To our knowledge, there exists no practical public key cryptosystem which has quadratic decryption time. In this chapter, we present a new cryptosystem with fast decryption time. The decryption is of quadratic bit complexity; it involves an extended Euclidean algorithm computation, an ideal reduction and a few basic operations like multiplication and division with remainder of numbers.

By the experiment of our implementation, our cryptosystem is comparably as fast as the encryption time of the RSA cryptosystem with $e = 2^{16} + 1$.

Our cryptosystem is constructed over an imaginary quadratic field. Buchmann and Williams proposed the first algorithm which achieves the Diffie-Hellman key distribution scheme using the class group in an imaginary quadratic field [BW88]. Later, Hafner and McCurley discovered the sub-exponential algorithm against the discrete logarithm problem of the class group [HM89]. Since then cryptosystems over class groups have not gained much attention in practice. Recently, Hühnlein et. al. proposed an ElGamal-type public key cryptosystem with faster decryption process in class groups of imaginary quadratic fields [HJPT98]. Here we call it the HJPT cryptosystem. Denote by $Cl(\Delta_q)$ and $Cl(\Delta_1)$ the class group of the non-maximal order and that of the maximal order respectively. The technique used in HJPT cryptosystem is to "switch" the ideals between $Cl(\Delta_q)$ and $Cl(\Delta_1)$. Note that the arithmetic of the switching is fast i.e., has quadratic complexity in the bit length of the public key. Nevertheless, the HJPT cryptosystem has cubic decryption time because it is an ElGamal type public key cryptosystem and involves an exponentiation step. In our case, we encrypt the plaintext $\mathfrak{m}$ by $E(\mathfrak{m}, r) = \mathfrak{m}\mathfrak{p}^r$, where $\mathfrak{p}$ is an element in the kernel of the map $Cl(\Delta_q) \to Cl(\Delta_1)$ and r is a random integer. By this encryption, the decryption process only involves the switching arithmetic, so the decryption has quadratic complexity in the bit length of the public key. The encryption process $E(\mathfrak{m}, r) = \mathfrak{m}\mathfrak{p}^r$ induces that our cryptosystem uses a probabilistic encryption and the homomorphy property.

The security of our cryptosystem is based upon a new number theoretic problem over quadratic orders which is closely related to factoring the discriminant $\Delta_q = -pq^2$. When we choose appropriate sizes of the parameters, the currently known fast algorithms like the elliptic curve method [Len87], the number field sieve [LL91] and the Hafner-McCurley algorithm [HM89] are not applicable. We also discuss the chosen ciphertext attack. In our cryptosystem, the surjective one way map $Cl(\Delta_q) \to Cl(\Delta_1)$ plays an important role. As related public key cryptosystems which uses such a surjective map, two public key cryptosystems are known: Shamir's *RSA for paranoids* which uses $(\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^* (n = pq)$ [Sha95] and the Okamoto-Uchiyama cryptosystem which uses $(\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^+ (n = p^2q)$ [OU98]. The chosen ciphertext attack is applicable to both public key cryptosystems, and the attacker can easily factor the public modulus n (See, for example, [GGOQ98]). Against our cryptosystem, Jaulmes and Joux also proposed a chosen-message attack for known two pair of messages [JJ99].

The chapter is organized as follows: we first recall the basic notions of class groups of quadratic orders and describe how to "switch" from the non-maximal order to the maximal order and vice-versa. We then present the new cryptosystem and analyze its security. Finally, we give some timings comparing our new cryptosystem with RSA.

2.2 Quadratic Orders

There are plenty of cryptographic primitives using quadratic fields and several public key cryptosystems are proposed [BW90]. We briefly explain the class group of a quadratic order. A more complete treatment may be found in [Cox89].

Let $\Delta \in \mathbb{Z}$ not a square such that $\Delta \equiv 0, 1 \pmod 4$. We call Δ a *(quadratic) discriminant*. Δ is called a *fundamental* discriminant if $\Delta \equiv 1 \pmod 4$ and is square-free, or $\Delta/4 \equiv 2, 3 \pmod 4$ and is square-free. Every discriminant Δ can be represented by $\Delta_1 f^2$, where Δ_1 is a fundamental discriminant and f is an integer, and we denote $\Delta_f = \Delta_1 f^2$. We consider only negative discriminants in this chapter. Let $\sqrt{\Delta_f} = i\sqrt{|\Delta_f|}$ be the square root of Δ_f on the upper half plane. Then we call $\mathcal{O}_{\Delta_f} = \mathbb{Z} + \frac{\Delta_f + \sqrt{\Delta_f}}{2}\mathbb{Z}$ the *quadratic order* of discriminant Δ_f. It is an integral domain. If Δ_f is not a fundamental discriminant then $\mathcal{O}_{\Delta_f} \subset \mathcal{O}_{\Delta_1}$ and $\mathcal{O}_{\Delta_f}$ has finite index f in $\mathcal{O}_{\Delta_1}$. Moreover, we have $\mathcal{O}_{\Delta_f} = \mathbb{Z} + f\mathcal{O}_{\Delta_1}$. The order $\mathcal{O}_{\Delta_f}$ is called the *non-maximal order* with *conductor* f, and the order $\mathcal{O}_{\Delta_1}$ is called the *maximal order*. Every element $\alpha \in \mathcal{O}_{\Delta_f}$ is represented by $\alpha = (x + y\sqrt{\Delta_f})/2$, $x, y \in \mathbb{Z}$. For $\alpha = (x + y\sqrt{\Delta_f})/2$, we denote by $\alpha' = (x - y\sqrt{\Delta_f})/2$ its (complex) conjugate. The *norm* of α is defined as $N(\alpha) = \alpha\alpha' = (x^2 - y^2\Delta_f)/4$. A subset $\mathfrak{a}$ of $\mathcal{O}_{\Delta_f}$ is an (integral) ideal of $\mathcal{O}_{\Delta_f}$ if $\alpha + \beta \in \mathfrak{a}$ whenever $\alpha, \beta \in \mathfrak{a}$, and $\alpha(\Delta_f + \sqrt{\Delta_f})/2 \in \mathfrak{a}$ whenever $\alpha \in \mathfrak{a}$. Every ideal $\mathfrak{a}$ of $\mathcal{O}_{\Delta_f}$ is given by

$$\mathfrak{a} = m\left(a\mathbb{Z} + \frac{b + \sqrt{\Delta_f}}{2}\mathbb{Z}\right), \tag{2.1}$$

where $m \in \mathbb{Z}$, $a \in \mathbb{Z}_{>0}$, and $b \in \mathbb{Z}$ such that $b^2 \equiv \Delta_f \pmod{4a}$. This expression is unique if we choose $-a < b \leq a$. Then (m, a, b) is called the *standard representation* of $\mathfrak{a}$. The *norm* of an ideal $\mathfrak{a}$ is defined by $N(\mathfrak{a}) = am$. $\mathfrak{a}$ is said to be *primitive* if $m = 1$. In that case, we represent $\mathfrak{a}$ by (a, b). For two given ideals $\mathfrak{a}, \mathfrak{b}$, we can define their product $\mathfrak{ab}$ (see, for example, [BW88]). The computation of a representation of $\mathfrak{ab}$ needs $O((\log(\max\{N(\mathfrak{a}), N(\mathfrak{b})\}))^2)$ bit operations.

We describe the class group of $\mathcal{O}_{\Delta_f}$. An ideal $\mathfrak{a}$ is called *prime* to f if $\mathrm{GCD}(N(\mathfrak{a}), f) = 1$ holds. The ideals of $\mathcal{O}_{\Delta_f}$ prime to f form an Abelian group; denote it by $\mathcal{I}_{\Delta_f}(f)$. Two ideals $\mathfrak{a}$ and $\mathfrak{b}$ are called *equivalent* if there is a $\alpha \in \mathcal{O}_{\Delta_f}$ such that $\mathfrak{a} = \alpha\mathfrak{b}$. Denote by $\mathfrak{a} \sim \mathfrak{b}$ this equivalence relation. For an element $\gamma \in \mathcal{O}_{\Delta_f}$ the ideal $\gamma\mathcal{O}_{\Delta_f}$ is called a *principal* ideal. The principal ideals $\mathcal{P}_{\Delta_f}(f)$ which are prime to f form a subgroup of $\mathcal{I}_{\Delta_f}(f)$. The quotient group $\mathcal{I}_{\Delta_f}(f)/\mathcal{P}_{\Delta_f}(f)$ is called the *class group* of $\mathcal{O}_{\Delta_f}$; denote it by $Cl(\Delta_f)$. The order of this group is denoted by $h(\Delta_f)$. For a primitive ideal $\mathfrak{a}$ in $\mathcal{I}_{\Delta_f}(f)$, we say that $\mathfrak{a} = (a, b)$ is *reduced* if $|b| \leq a \leq c = (b^2 - \Delta_f)/4a$ and additionally $b \geq 0$ if $a = c$ or $a = |b|$. There is only one reduced ideal in every equivalence class. Denote by $Red_{\Delta_f}(\mathfrak{a})$ the reduced ideal equivalent to $\mathfrak{a} \in \mathcal{I}_{\Delta_f}(f)$. An algorithm to compute $Red_{\Delta_f}(\mathfrak{a})$ from $\mathfrak{a}$ is described in [BW88] and requires $O((\log(N(\mathfrak{a})))^2)$ bit operations. We identify each class of the

class group with the unique reduced ideal. It is easy to verify that $N(\mathfrak{a}) < \sqrt{|\Delta_f|/3}$ holds for every reduced ideal $\mathfrak{a} \in \mathcal{I}_{\Delta_f}(f)$. Conversely, a primitive ideal $\mathfrak{a} \in \mathcal{I}_{\Delta_f}(f)$ with small norm such that $N(\mathfrak{a}) < \sqrt{|\Delta_f|/4}$ is always a reduced ideal. It turns out that we can compute the representation of the product of two classes of the class group in $O((\log \sqrt{|\Delta_f|})^2)$ bit operations. See [BB97].

2.2.1 The map $Cl(\Delta_q) \to Cl(\Delta_1)$

In [Cox89], the relationship between ideals in the maximal order $\mathcal{O}_{\Delta_1}$ and in the non-maximal order $\mathcal{O}_{\Delta_f}$ is investigated. If $\mathfrak{a}$ is an ideal in $\mathcal{I}_{\Delta_f}(f)$ then $\mathfrak{A} = \mathfrak{a}\mathcal{O}_{\Delta_1}$ is an ideal in $\mathcal{I}_{\Delta_1}(f)$ and $N(\mathfrak{a}) = N(\mathfrak{A})$. Similarly, if $\mathfrak{A}$ is an ideal in $\mathcal{I}_{\Delta_1}(f)$ then $\mathfrak{a} = \mathfrak{A} \cap \mathcal{O}_{\Delta_f}$ is an ideal in $\mathcal{I}_{\Delta_f}(f)$ and $N(\mathfrak{A}) = N(\mathfrak{a})$. The map $\phi : \mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{\Delta_1}$ induces an isomorphism $\mathcal{I}_{\Delta_f}(f) \tilde{\to} \mathcal{I}_{\Delta_1}(f)$. The inverse of this map is $\phi^{-1} : \mathfrak{A} \mapsto \mathfrak{A} \cap \mathcal{O}_{\Delta_f}$. Let $f = q$ be a prime and $\sqrt{|\Delta_1|/3} < q$. Then all the reduced ideals in $Cl(\Delta_1)$ are prime to the conductor q [HJPT98]. Thus we can consider the following map based on ϕ:

$$\begin{array}{rccc} \varphi_q : & Cl(\Delta_q) & \longrightarrow & Cl(\Delta_1) \\ & \mathfrak{a} & \longmapsto & Red_{\Delta_1}(\mathfrak{a}\mathcal{O}_{\Delta_1}), \end{array}$$

where we identify a class of both class groups with the unique reduced ideal in that class. (Note that if $q > \sqrt{|\Delta_1|/3}$ we also can define this map; we possibly have to compute an ideal equivalent to $\mathfrak{a}$ which is prime to q. See [HJPT98].) A practical algorithm to compute φ_q is as follows:

1. Algorithm (GoToMaxOrder)
Input: A reduced ideal $\mathfrak{a} = (a, b) \in Cl(\Delta_q)$, the discriminant Δ_q the fundamental discriminant Δ_1, and the conductor q
Output: A reduced ideal $\mathfrak{A} = \varphi_q(\mathfrak{a}) = (A, B)$.

1. $A \leftarrow a$
2. $b_{\mathcal{O}} \leftarrow \Delta_q \bmod 2$
3. Solve $1 = \mu q + \lambda a$ for $\mu, \lambda \in \mathbb{Z}$ using the extended Euclidean algorithm
4. $B \leftarrow b\mu + a b_{\mathcal{O}} \lambda \bmod 2a$
5. $(A, B) \leftarrow Red_{\Delta_1}(A, B)$
6. RETURN (A, B)

Note that the map GoToMaxOrder is different from the map described in [HJPT98]. Every step of this algorithm requires $O((\log \sqrt{|\Delta_q|})^2)$ bit operations, thus the complexity of this algorithm is quadratic.

We discuss the "inverse" map φ_q^{-1}. The map $\varphi_q : Cl(\Delta_q) \to Cl(\Delta_1)$ is surjective and we have $h(\Delta_q) = h(\Delta_1)\,(q - (\Delta_1/q))$, where (Δ_1/q) is the Kronecker-symbol (See,

for example, [Cox89]). Denote by $Ker(\varphi_q)$ the kernel of the map $\varphi_q : Cl(\Delta_q) \to Cl(\Delta_1)$ which is a cyclic subgroup of $Cl(\Delta_q)$ with order $q - (\Delta_1/q)$. So there is a $(q - (\Delta_1/q))$-fold ambiguity for the inverse of the map φ_q. We will distinguish a unique reduced ideal from these preimages using the size of the norm of an ideal. The norm of any reduced ideal in $Cl(\Delta_1)$ is smaller than $\sqrt{|\Delta_1|/3}$. By our assumption $\sqrt{|\Delta_1|/3} < q$ all ideals in $Cl(\Delta_1)$ are prime to the conductor q. Therefore for a reduced ideal $\mathfrak{A}$ in $Cl(\Delta_1)$ $\mathfrak{a} = \phi^{-1}(\mathfrak{A}) = \mathfrak{A} \cap \mathcal{O}_{\Delta_q}$ is a primitive ideal in $\mathcal{I}_{\Delta_q}(q)$, and $N(\mathfrak{A}) = N(\mathfrak{a})$. If the primitive ideal $\mathfrak{a}$ in $\mathcal{I}_{\Delta_q}(q)$ satisfies $N(\mathfrak{a}) < \sqrt{|\Delta_1|/4}$, then both $\mathfrak{a}$ an $\mathfrak{A}$ are reduced ideals. Consequently, if we restrict ourselves to ideals $\mathfrak{a}$ in $Cl(\Delta_q)$ such that $N(\mathfrak{a}) < \sqrt{|\Delta_1|/4}$, then $\varphi_q(\mathfrak{a}) \cap \mathcal{O}_{\Delta_q}$ is reduced (in $\mathcal{I}_{\Delta_q}(q)$) and so we can compute a distinguished inverse of the map φ_q, namely $\mathfrak{a}$. Note that the cardinality of this set is smaller than that of $Cl(\Delta_1)$. We denote by φ_q^{-1} this restricted inverse map and the practical algorithm to compute the map φ_q^{-1} is as follows:

2. Algorithm (Inverse)
Input: A reduced ideal $\mathfrak{A} = (A, B) \in Cl(\Delta_1)$ such that $N(\mathfrak{A}) < \sqrt{|\Delta_1|/4}$, the conductor q
Output: A reduced ideal $\mathfrak{a} \in Cl(\Delta_q)$ such that $\varphi_q^{-1}(\mathfrak{A}) = \mathfrak{a} = (a, b)$.

1. $a \leftarrow A$
2. $b \leftarrow Bq \bmod 2a$
3. RETURN (a, b)

This algorithm obviously requires only $O((\log(\sqrt{|\Delta_1|}))^2)$ bit operations.

2.3 NICE cryptosystem

Generate two random primes $p, q > 4$ such that $p \equiv 3 \pmod 4$ and let $\Delta_1 = -p$. Let $Cl(\Delta_1)$ be the class group of the maximal order with discriminant Δ_1 and $Cl(\Delta_q)$ be the class group of the non-maximal order with conductor q. Δ_q will be public, whilst its factorization into Δ_1 and q will be kept private. The discriminant Δ_1 and the conductor q are large primes to prevent breaking the cryptosystem by factoring Δ_q.

In the key generation, we choose an ideal $\mathfrak{p}$ from the kernel $Ker(\varphi_q)$ and make $\mathfrak{p}$ public. The message ideal $\mathfrak{m}$ is an reduced ideal in $Cl(\Delta_q)$ with norm smaller than $\lfloor\sqrt{|\Delta_1|/4}\rfloor$. The encryption is carried over the class group $Cl(\Delta_q)$ by computing $Red_{\Delta_q}(\mathfrak{m}\mathfrak{p}^r)$, where r is a random integer smaller than $q - (\Delta_1/q)$. Then, by the knowledge of the conductor, we can go to the maximal order and the image of the message ideal $\varphi_q(\mathfrak{m})$ in the maximal order is revealed, since $\varphi_q(\mathfrak{m}\mathfrak{p}^r) = \varphi_q(\mathfrak{m})\varphi_q(\mathfrak{p}^r) = \varphi_q(\mathfrak{m})\mathcal{O}_{\Delta_1} = \varphi_q(\mathfrak{m})$. We can recover the message by computing the unique preimage of $\varphi_q(\mathfrak{m})$, namely $\mathfrak{m} = \varphi_q^{-1}(\varphi(m))$.

1. **Key generation:** Generate two random primes $p, q > 4$ with $p \equiv 3 \pmod 4$ and $\sqrt{p/3} < q$. Let $\Delta_1 = -p$ and $\Delta_q = \Delta_1 q^2$. Let k and l be the bit lengths of $\lfloor\sqrt{|\Delta_1|/4}\rfloor$ and $q - (\Delta_1/q)$ respectively. Choose an ideal $\mathfrak{p}$ in $Cl(\Delta_q)$, where

$$\varphi_q(\mathfrak{p}) \text{ is a principal ideal in } \mathcal{O}_{\Delta_1}. \tag{2.2}$$

   Then $(\mathfrak{p}, \Delta_q, k, l)$ are the system parameters, and Δ_1, q are the secret keys.

2. **Encryption:** Let $\mathfrak{m}$ be the plaintext, where $\mathfrak{m}$ is a reduced ideal in $Cl(\Delta_q)$ with $\log_2 N(\mathfrak{m}) < k$. Pick up a random $l-1$ bit integer and we encrypt the plaintext as follows using binary exponentiation techniques:

$$\mathfrak{c} = Red_{\Delta_q}(\mathfrak{m}\mathfrak{p}^r) \tag{2.3}$$

   Then $\mathfrak{c}$ is the cipher ideal.

3. **Decryption:** Using the secret keys Δ_1, q, we compute $\mathfrak{K} = \text{GoToMaxOrder}(\mathfrak{c})$. The plaintext $\mathfrak{m}$ can be recovered by computing $\mathfrak{m} = \text{Inverse}(\mathfrak{K})$.

The embedding of a number into a message ideal may be simply done as follows: let x be a message and t be a random number of length $k - 2 - \lfloor\log_2 x\rfloor$. Denote by $x.t$ the concatenation of x and t as bitstrings. We determine the smallest prime l larger than $x.t$ with $(\Delta_q/l) = 1$. It follows $\log_2 l < k-1$. This can be done effectively using a few trials of a primality test and Jacobi symbol computations. Then, compute b such that $\Delta_q \equiv b^2 \bmod 4l, -l < b \leq l$. This can also be done effectively using the RESSOL algorithm of Shanks [Sha89]. Then $\mathfrak{a} = (l, b)$ is a reduced ideal with $\log_2 N(\mathfrak{a}) < k$.

The key generation simply works as follows: choose a number $\alpha \in \mathcal{O}_{\Delta_1}$ with norm less than $\sqrt{|\Delta_q|/4}$, compute the standard representation of the ideal $\alpha\mathcal{O}_{\Delta_1}$ and compute $\mathfrak{p} = \phi^{-1}(\alpha\mathcal{O}_{\Delta_1})$. This is explained in [HJPT98]. Then $\mathfrak{p} \in \ker\varphi_q$. The encryption takes $O((\log\sqrt{|\Delta_q|})^3)$ bit operations because of the binary exponentiation. The decryption involves two algorithms of quadratic complexity, so it requires only $O((\log\sqrt{|\Delta_q|})^2)$ bit operations.

2.4 Security considerations

The security of our cryptosystem depends on the difficulty of factoring the discriminant Δ_q. If the discriminant Δ_q can be factored, our proposed cryptosystem is completely broken. At first, we consider the size of the secret parameters Δ_1 and q to prevent breaking the cryptosystem by factoring Δ_q. On the other hand, an attacker may somehow compute the image $\phi(\mathfrak{a})$ in the maximal order $\mathcal{O}_{\Delta_1}$ for some ideal $\mathfrak{a}$ of $\mathcal{O}_{\Delta_q}$. We prove that to compute the map $\phi(\mathfrak{a})$ is as intractable as factoring Δ_q. In our cryptosystem, we make public an ideal $\mathfrak{p}$ in $Ker(\varphi_q)$. We discuss that

according to current knowledge the knowledge of such an ideal does not bring any advantage for factoring the discriminant. Finally we argue that a chosen ciphertext attack as presented in [GGOQ98] against Shamir's RSA variant [Sha95] will not give us substantially more knowledge than previously known.

2.4.1 The size of the secret parameters Δ_1, q

We discuss the size of the secret parameters $\Delta_1 = -p$ and q which prevents attacks by the known factoring algorithms. Let $L_N[s,c] = exp((c+o(1))\log^s(N)\log\log^{1-s}(N))$. The number field sieve [LL91] and the elliptic curve method [Len87] are the different types of factoring algorithms which have to be taken care of; other factoring algorithms are more or less slower [MvOV97] [RS97] [Si00]. The number field sieve is the fastest factoring algorithm, and the running time depends on the total bit length of the composite number $|\Delta_q|$; it is of the order of $L_{|\Delta_q|}[1/3, (64/9)^{1/3}]$. Currently the fastest implementation for the number field sieve factored a 155-digit ($\approx$ 512-bit) RSA modulus [RSA155]. If we choose Δ_q to be larger than 1024 bits, the number field sieve becomes infeasible. On the other side, the elliptic curve method depends on the size of the primes p or q and the expected running time is $L_r[1/2, 2^{1/2}]$, where r is p or q. The fastest implementation for the elliptic curve method found a 53-digit ($\approx$ 175-bit) prime factor [ECM98]. If we choose p and q to be larger than 341 bits, the elliptic curve method becomes infeasible. Therefore the 1024 bit discriminant Δ_q with 341 bit p, q is secure for cryptographic purposes.

We wonder if there exists a special algorithm for factoring a composite number with squared a prime factor. To our knowledge, the only one algorithm is presented for this problem by Peralta-Okamoto [PO96]. They improve the elliptic curve method by a constant factor by considering the distribution of the Jacobi symbol. For examples, for finding the 40-digit ($\approx$ 133-bit) prime factor, the algorithm is 25-time faster than the original elliptic curve method. Its improvement is negligible and is not a real threat.

2.4.2 Security of ϕ

Only the one who knows the conductor q can compute the map φ_q and then recover any message ideal. The map φ_q consists of $\varphi_q = Red_{\Delta_1} \circ \phi$. If attackers somehow can compute the ideal $\phi(\mathfrak{a})$ in the maximal order which is the image of an ideal $\mathfrak{a}$ in $Cl(\Delta_q)$, then the message ideal $\mathfrak{m}$ may be recovered. Here, we can prove that the discriminant Δ_q can be factored using few iterations of any algorithm which computes the image of ϕ.

2.1. Theorem *Assume that there exists an algorithm* $\mathbf{AL}_\phi$ *which computes for the primitive ideal* $\mathfrak{a} = (a_1, a_2) \in \mathcal{I}_{\Delta_q}(q)$ *a primitive ideal* $\mathfrak{A} = (A_1, A_2) \in \mathcal{I}_{\Delta_1}(q)$ *such*

that $\mathfrak{A} = \phi(\mathfrak{a})$ without knowing the conductor q. By using the algorithm $\mathbf{AL}_\phi$ as an oracle, the discriminant $\Delta_q = \Delta q^2$ can be factored in random polynomial time.

Proof: Let $\mathfrak{a} = (a, b)$ the a primitive ideal in $\mathcal{I}_{\Delta_q}(q)$. By using the algorithm $\mathbf{AL}_\phi$, we can compute $\mathfrak{A} = (A, B)$ such that $\mathfrak{A} = \phi(\mathfrak{a})$. The relation between the ideals $\mathfrak{a}$ and $\mathfrak{A}$ is as follows:

$$a = A, \quad B \equiv bq^{-1} \pmod{a}. \tag{2.4}$$

Therefore, we can compute $q \equiv bB^{-1} \pmod{a}$ because $(B, a) = (b, a) = 1$. We apply this algorithm for several prime ideals $\mathfrak{p}_i = (p_i, b_{p_i})$, where p_i is prime with $(p_i/\Delta_q) = 1$ which require the random polynomial time in generating them. After polynomially many iterations in $\log_2 \Delta_q$, we can recover the conductor q using the Chinese Remainder Theorem. It is easy to check the right q by computing the greatest common divisor with Δ_q. ■

This theorem means that nobody can "switch" the primitive ideal (a, b) to the maximal order without the knowledge of the conductor q.

2.4.3 Knowledge of $\mathfrak{p}$

Let $\mathfrak{p}$ be the public key which is the element in $Ker(\varphi_q)$. We will argue that the knowledge of $\mathfrak{p}$ does not substantially help to factor Δ_q using currently known fast algorithms. For simplicity, we assume $\mathfrak{p}$ is the generator of the group $Ker(\varphi_q)$, so the order of $\mathfrak{p}$ is $q - (q/\Delta_1)$. A non-trivial ambiguous ideal is an ideal $\mathfrak{f}$ in $Cl(\Delta_q)$ such that $\mathfrak{f}^2 \sim 1$ and $\mathfrak{f} \not\sim 1$. If a non-trivial ambiguous ideal in the order $\mathcal{O}_{\Delta_q}$ is known, we can factor the discriminant Δ_q [Sch83]. For the discriminant Δ_q of our cryptosystem, there is only 1 non trivial ambiguous ideals in $Cl(\Delta_q)$. Moreover, the non trivial ambiguous ideal lies in the group $Ker(\varphi_q)$, so the probability that $\mathfrak{p}^r$ for a random r will be a non trivial ambiguous ideal is negligible. It is unknown that other ideals in $Ker(\varphi_q)$ except the ambiguous ideals can be used for factoring the discriminant Δ_q.

In our cryptosystem, we publish the ideal $\mathfrak{p}$. A possible attack to find a non-trivial ambiguous ideal for a given $\mathfrak{p}$ is to compute the order of $\mathfrak{p}$ in the group $Cl(\Delta_q)$. The fastest algorithm to compute the order of $\mathfrak{p}$ in the group $Cl(\Delta_q)$ is Hafner-McCurley algorithm [HM89]. Its running time is $L_{|\Delta_q|}[1/2, 2^{1/2}]$ which is much slower than factoring Δ_q. This shows that with the currently known algorithms, the knowledge of $\mathfrak{p}$ does with high probability not help in factoring Δ_q. The same reasoning applies for polynomially many elements of $Ker(\varphi_q)$.

2.4.4 Chosen ciphertext attack

Let G_1, G_2 be finite abelian groups and consider a surjective homomorphism $\varphi : G_1 \to G_2$. If two elements g, h in G_1 satisfy $\varphi(g) = \varphi(h)$, then we call them *to be in the same coset*. Our cryptosystem is constructed using the surjective homomorphism $\varphi_q : Cl(\Delta_q) \to Cl(\Delta_1)$. The message ideal $\mathfrak{m}$ is encrypted by $\mathfrak{c} = Red_{\Delta_q}(\mathfrak{m}\mathfrak{p}^r)$. Then the ciphertext $\mathfrak{c}$ "represents" all elements of the coset of $\mathfrak{m}$ in the group $Cl(\Delta_q)$.

Similarly, Shamir proposed an RSA type public key cryptosystem using the homomorphism $\varphi_S : (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^*$, where $n = pq$ and p, q are primes [Sha95]. In the key generation, e, d are generated by the relation $ed \equiv 1 \pmod{p-1}$. The message M must be smaller than p. For the encryption we compute $C \equiv M^e \pmod{n}$, and the message can be recovered by $M \equiv C^d \pmod{p}$. For an element $a \in (\mathbb{Z}/n\mathbb{Z})^*$, all elements of the coset of a for the map φ_S are represented by $\{a, a+p, a+2p, \ldots, a+(q-1)p\}$. Therefore, if we know two elements a_1, a_2 in the same coset, we can factor the modulus n by computing $\mathrm{GCD}(a_1 - a_2, n) = p$. This is equivalent to the fact that n can be factored if we know an element in the kernel of φ_S. Using this Gilbert et al. proposed the following attack against this cryptosystem [GGOQ98]. Let M' be a message larger than p, and C be the ciphertext corresponding to M'. If an attacker can know the plaintext corresponding to C, say M, then the modulus n can be factored by computing $\mathrm{GCD}(M - M', n)$. We call this attack the *chosen ciphertext attack*. [1] Note that this chosen ciphertext can be achieved because $\mathbb{Z}/n\mathbb{Z}$ is not only a group but also a ring.

Consider the chosen ciphertext attack against our proposed cryptosystem. Let $\mathfrak{m}'$ be a message ideal such that $N(\mathfrak{m}') > \sqrt{|\Delta_1|/3}$. If $\mathfrak{m}$ is the regular message ideal which is a reduced ideal with norm smaller than $\sqrt{|\Delta_1|/4}$ and in the same coset of $\mathfrak{m}'$, then we have $\mathfrak{m} \sim \mathfrak{m}'\mathfrak{p}^s$ for some integer $s \geq 0$. This yields the knowledge of some other $\mathfrak{p}' \in Ker(\varphi_q)$. As shown above, no algorithm is known to compute the factorization of Δ_q when polynomially many elements of the kernel are known.

Next, we discuss the case where the chosen ciphertext attack is applied several times. Jaulmes and Joux proposed the chosen-message attack for known two pair of messages [JJ99]. We explain the attack in the following. It is known that if the norm of the primitive ideal $\mathfrak{A} = (A, B) \in Cl(\Delta_1)$ is smaller than $\sqrt{|\Delta_1|}$ then $\mathfrak{A}$ is already a reduced ideal or ideal (C, R) where $C = (B^2 - \Delta_1)/4A$ and $-B = 2CQ + R, -C < R \leq C$ is a reduced ideal (See, [Coh93], p.239). The ideal (C, R) is obtained after exactly one reduction step. Here we choose two ideals $\mathfrak{a}_1 = (a_1, b_1), \mathfrak{a}_2 = (a_2, b_2)$ such that $\sqrt{|\Delta_1|/3} < N(\mathfrak{a}_1), N(\mathfrak{a}_2) < \sqrt{|\Delta_1|}$, and we encrypt them by $\mathfrak{c}_1 \sim \mathfrak{a}_1\mathfrak{p}^{r_1}, \mathfrak{c}_2 \sim \mathfrak{a}_2\mathfrak{p}^{r_2}$ for some integers r_1, r_2. We assume that an attacker know the regular message ideals of $\mathfrak{c}_1, \mathfrak{c}_2$ which are different from $\mathfrak{a}_1, \mathfrak{a}_2$. We can choose such ideals because the bit length of $\lfloor\sqrt{|\Delta_1|/4}\rfloor$ is public. Here, let $\phi(\mathfrak{a}_1) = \mathfrak{A}_1 = (A_1, B_1), \phi(\mathfrak{a}_2) =$

[1] Okamoto and Uchiyama proposed the public-key cryptosystem using the homomorphism $\varphi_{OU} : (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^+$, where $n = p^2q$ and p, q are primes [OU98]. This chosen ciphertext attack is also applicable against the Okamoto-Uchiyama cryptosystem.

$\mathfrak{A}_2 = (A_2, B_2)$, then $A_1 = a_1, A_2 = a_2$ hold and B_1, B_2 are unknown. From the above lemma the ideals $\mathfrak{A}_1, \mathfrak{A}_2$ will be reduced in exact one reduction step. Denote by $\mathfrak{B}_1, \mathfrak{B}_2$ the reduced ideals in $Cl(\Delta_1)$ which are equivalent to $\mathfrak{A}_1, \mathfrak{A}_2$, respectively. Then we have relations $\mathfrak{B}_1 = ((B_1^2 - \Delta_1)/4, R_1), \mathfrak{B}_2 = ((B_2^2 - \Delta_1)/4, R_2)$ for some integer R_1, R_2. The ideal $\mathfrak{B}_1, \mathfrak{B}_2$ coincide with the regular message ideals of $\mathfrak{c}_1, \mathfrak{c}_2$, respectively. Therefore we know the relation $E_1 = (B_1^2 - \Delta_1)/4, E_2 = (B_2^2 - \Delta_1)4$, where B_1, B_2, Δ_1 are unknown. Then we transform them $(B_2 + B_1)(B_2 - B_1) = 4(E_2 - E_1)$. We can find B_1, B_2 by factoring $4(E_2 - E_1)$ which can be found easily because $4|E_2 - E_1| < 8|\Delta_1|$ holds and we can try the different pairs in case that it is the product of two large primes. Consequently, we can find Δ_1.

2.5 Running time of NICE cryptosystem

The prominent property of the proposed cryptosystem is the running time of the decryption. Most prominent cryptosystems require decryption time $O((\log_2 n)^3)$, where n is the size of the public key. The total running time of the decryption process of our cryptosystem is $O((\log_2 \Delta_q)^2)$ bit operations. In order to demonstrate the improved efficiency of our decryption, we implemented our scheme using the LiDIA library [LiD95]. It should be emphasized here that our implementation was not optimized for cryptographic purposes — it is only intended to provide a comparison between decrypting in the non-maximal order and using our trapdoor decryption. The results are shown in table 2.1.

Observe that we separated the fast exponentiation step of the encryption as a "pre-computation" stage. Indeed, if we can securely store the values $\mathfrak{p}^r$, then the actual encryption can be effected very rapidly, since it requires only one ideal multiplication and one ideal reduction. Of course, one can use one of the well-known fixed-base exponentiation techniques completely analogously as for ElGamal type protocols as mentioned e.g. in [MvOV97], section 14.6.3.

It should be mentioned that the size of a message for our cryptosystem is smaller than the size of a message for the RSA encryption (e.g. 256 bit vs. 768 bit, or 341 bit vs. 1024 bit). In connection with the very fast decryption time, an excellent purpose for our cryptosystem could be (symmetric) key distribution. In that setting, the short message length is not a real drawback. On the other side, the message length is longer than for ElGamal encryption on "comparably" secure elliptic curves (e.g. 341 bit vs. 180 bit).

$\log_2(n)$	1024	1536	2048
RSA encryption	10 ms	19 ms	31 ms
RSA classical decryption	1032 ms	3045 ms	7006 ms
NICE precomputation for $p \approx n^{1/3}$	7650 ms	21682 ms	36166 ms
NICE encryption for $p \approx n^{1/3}$	4 ms	8 ms	12 ms
NICE decryption for $p \approx n^{1/3}$	13 ms	22 ms	30 ms
NICE precomputation for $p \approx n^{1/4}$	8766 ms	24673 ms	36276 ms
NICE encryption for $p \approx n^{1/4}$	4 ms	6 ms	10 ms
NICE decryption for $p \approx n^{1/4}$	12 ms	22 ms	32 ms

Table 2.1: Average timings for the new cryptosystem compared to RSA ($e = 2^{16}+1$) over 100 randomly chosen pairs of primes of the specified size on a SPARC station 4 (110 MHz) using the LiDIA library

2.6 A smart card implementation and its problems

We implemented the NICE decryption on a smart card. More precisely, we implemented the NICE decryption algorithm using the Siemens development kit for chip card controller ICs based on Keil PK51 for Windows. As assembler we used A51, as linker L51 to generate code for the 8051 microcontroller family. The software simulation were made using dScope-51 for Windows and the drivers for SLE 66CX80S. Thus, we realized a software emulation of an assembler implementation of the NICE decryption algorithm to be run on the existing Siemens SLE 66CX80S. Unfortunately, the timings of this software simulation were unrealistic. So, we did run some timings on a hardware simulator for SLE 66CX80S. Thanks to Deutsche Telekom AG, Produktzentrum Telesec in Siegen and Infineon/Siemens in Munich for letting us use their hardware simulator. See the timings for decryption in table 2.2 for a smart card running at 4.915 MHz.

$\log_2(n)$	1024
RSA decryption with CRT	490 ms
New CS decryption for $p \approx q \approx \Delta_q^{1/3}$	1242 ms
Improved version	1035 ms

Table 2.2: Timings for the decryption of the new cryptosystem compared to RSA using the hardware simulator of Siemens 66CX80S at 4.915 MHz

The very first implementation was very inefficient; the straightforward algorithms used in the software comparison proved to be much slower on the smart card than the existing RSA on the card. This was surprising, but after a while this could be easily explained: the cryptographic coprocessor has been optimized for modular exponentiation. On the other side, NICE uses mostly divisions with remainder and

comparisons. These operations are slow on the coprocessor, so we had to modify the decryption algorithm to speed it up in hardware. We change two parts of the decryption aglorithm, namely step 3 and step 5 of GoToMaxOrder.

In step 3 of GoToMaxOrder, the computation of the inverse of a modulo q using the extended Euclidean algorithm took (with $q \approx p \approx 341$ bit) about 9 seconds (!), whereas computing the inverse using Fermat's little theorem - by using fast exponentiation mod q - took less than 1 second. Note that the decryption time using this method is no longer of quadratic complexity.

The detail of the reduction algorithm in step 5 of GoToMaxOrder is as follows:

3. Algorithm (Reduction)
Input: A discriminant Δ_1 and a primitive ideal $\mathfrak{A} = (A, B) \subset \mathcal{O}_{\Delta_1}$
Output: A reduced ideal $Red_{\Delta_1}(A, B)$

5. */* Compute $Red_{\Delta_1}(A, B)$ */*
   5.1. $C \leftarrow (\Delta_1 - B^2)/4A$
   5.2. WHILE $\{-A < B \leq A < C\}$ or $\{0 \leq B \leq A = C\}$ DO
      5.2.1. Find $\lambda, \mu \in \mathbb{Z}$ s.t. $-A \leq \mu = B + 2\lambda A < A$ using division with remainder
      5.2.2. $(A, B, C) \leftarrow (C - \lambda\frac{B+\mu}{2}, \mu, A)$
   5.3. IF $A = C$ AND $B < 0$ THEN $B \leftarrow -B$
   5.4. RETURN (A, B)

In the reduction process the quotient in the division with remainder step is most of the time very small (say ≤ 10, see Appendix 2.6); to effect a division is this case is much more time consuming than subsequent subtractions. We did replace

5.2.1 Find $\lambda, \mu \in \mathbb{Z}$ s.t. $-A \leq \mu = B + 2\lambda A < A$ using division with remainder
5.2.2 $(A, B, C) \leftarrow (C - \lambda\frac{B+\mu}{2}, -\mu, A)$

by

5.2.1 WHILE $B \leq -A$ OR $B > A$ DO
5.2.1.1 IF $B < 0$ THEN $B \leftarrow B + 2A$; $C \leftarrow C - (B + A)$
5.2.1.2 ELSE $B \leftarrow B - 2A$; $C \leftarrow C - (B - A)$;

every time that the bitlength of B was exceeding the bitlength of B by at least 3. Using this improvement, we could decrease the running time from about 1.8 s to 1.2 s. This is already faster than a 1024 bit RSA decryption without Chinese remainder (approx. 1.6 ms).

The timings in table 2.2 were made including these two improvements. Moreover, a detailed timing analysis in Siegen showed that both our static memory management and as well as the cryptographic coprocessor are not optimal for this algorithm. We discuss this in the sequel. An average overview of the most time consuming parts is given in table 2.3.

Function	Average time
`mul`, multiplications on the coprocessor	170 ms
`div`, divisions on the coprocessor	231 ms
`left_adjust`, length correction of the variables	376 ms
`C2XL`, moving numbers into the coprocessor	118 ms
`XL2C`, moving numbers out of the coprocessor	223 ms
others (comparisons, small operations)	114 ms
Overall time	1242 ms

Table 2.3: Detailed timings for different functions in the decryption of the new cryptosystem

One major difference between RSA and NICE is the number of variables needed during the computation of the decryption algorithm. In our implementation, we need to store 11 variables of length at most 2048 bit. Computations of the cryptographic coprocessor are shortening these variables. Thus, we had to adjust the length of the variables after each important operation. This was done by moving the top nonzero bytes of the number to the fixed address of the number and so "erasing" leading zero bytes. To do this, we used the cryptographic coprocessor. Now the exact timings showed that about 33 % of the running time is spent by the function `left_adjust`, which effects this correction.

Now changing the memory management from static to dynamic (i.e. in the `XL2C` and `C2XL` functions making the appropriate changes and having additionally some registers holding the starting address of the numbers), we got an improvement to **637 ms** using the software simulator. Both Infineon/Siemens and Deutsche Telekom reported the overall time of the hardware simulator now to be **1035 ms**. Note that the amount of memory required is **965 Bytes** and thus fits into a real SLE 66CX80S.

As one can see from table 2.3, another important time consuming operation is to move numbers into and out of the coprocessor. At this point, we would get a speedup of about 350 ms if we could leave the numbers in registers inside the coprocessor. It is clear that the currently used processor is not prepared for such operations, since it is optimized for RSA, thus operations with very few variables. At this point we ask the hardware community to present solutions to this problem.

Acknowledgements for the smartcard implemenation

We thank Deutsche Telekom AG, Produktzentrum Telesec for letting us testing NICE on the hardware simulator and Siemens AG/Infineon GmbH for their valuable help concerning the use of the development kit as well as running our code on their hardware simulator.

Appendix A: An example of the reduction step

Input: discriminant Δ_1, an ideal $\mathfrak{A} = (A, B)$ before the reduction step

```
D1 = -391955329881115736847552399099477748671287962016063668607498092668663577456528165441653561322531245266 3
A  =  14709196987236217470310344790336554582213450995580439354777614852261315534370729939175650447453125518926471909475 7
      65980819130165941950615573544141741449 1
B  = -16120786611173332936175433825318881027235314017722179961013037966005831645873381632911088594573172128409548203242276
      66120696958777850232980376265721150213
```

Reduction step 5.1.: quotient λ, remainder μ such that $B = 2A\lambda + \mu, (-a \leq \mu < a)$

```
(quotient, remainder) =
(-1,   13297607363299102004445255755354228137191587973438698748542191738516799425000764354924041494917452975375746178652 8
       76584094130155403366825077082271136787 69)
(-2,  -12761243590327174728420394096256304728831977979402420516873134568804685699281859725103664466875812641319914710949 0
       15928355646393488678950389641306963210 5)
(5,   -78528290011262457388456843941329248503454444074743493793783111760006835153040692321204584715201859717103940903082 6
       18686617579070735519374115992714040 65)
(3,    10244818933510891787370810612951117737019488809536816914420866112436574171352321538119941956085359074784832443404 7
       44032975209160972759289660851786 54669)
(-2,  -10224129845590147501382365118494266252021893092244629712061058909068680852197387614386894226954222864793077231352 3
       57416590511833810655070898694329260 5)
(5,   -11105296432786254324801445410868102997262161184053297146125015494247765187360954811606805469093183807780471092809 0
       93473562855601861999949481990650315)
(2,    22472176875075749176609304144458704509990853674790256950472930566758992023189046610292013458995337873101127300307 1
       26536968520513812928743328161115 1)
(-24, -19399768262238616074750925620345945700130240074364394015776939186812798812752098861599659305585438681964815901740 6
       46853690292952894987154672355 03)
(5,   -87214024755816152729750202158525445031303615368933392297827179212874361504535183941307866086816850522516735441849 1
       91048830504132818556443763927)
(5,   -65892561542204283487634848048402908205850705405268138620901150996210506067756390839423261605983040481101072730135 9
       35120696842084667841807743 33)
(3,   -35372540965036205934141340575440504523333071659818592716658193807183159467823011633885293333993006718946556400076 9
       28723389657964699502098981 5)
(7,   -24722228429459372714248962154023865024293254144916063323433706888146107111333799338478440615432732354638347537820 3
       76474826892936909497877111)
(2,    21123896082367134246190321628964774052168904073609642897109183311547987453027704896019427995726765091708200095299 2
       75660819814689206866945135)
(-5,  -13882849961083763068299825465068008772978539934410435686028126703692746197995757231193841008190833004541581261177 9
       61179681653871930740550 5)
(3,   -76581145098854742314404688330389466846231064125144951559075955741419778256647722945901688894028499123381476982860 8
       31385805486639937328 59)
(6,    45171597597333763623992365562270451301836956090008641386920496283515261555967185885324296557967735725107983930920 3
       84463045341798439293 1)
(-3,   57949399291153659951735032186608343843568674256297346337247982645939248022677985693922616046011747896784792467175 3
       92968596489710481213)
(-3,   13506775252588510548399455966472193855917717568054168943345013045530491969273737406572000625840454580937146559331 8
       9049738778828983289)
(-15,  33719331931229543174596901666464315138791547976559206823565738527321556044311098226988811614870540130943134313822 5
       96913769977980851)
(-3,   32387847383134384596896074551207436375450678276080867668177408072318238232294269933210033680515064228455159232690 4
       4296100266373941)
(-4,   16714419738146443196752822573871854231405166017209952769083137312359031623980857178233078983102193989180745595809 6
       732542876200243)
(-5,  -30939001159755959315976107374558197066820019463549929232211643659606767979683020394240493706340093182264070243126 5
       3906939082883)
(11,  -11531068580582167360450543647526376554517293415868320246276968612358966470144179822872871524178165755042181780892 0
       121862424193)
(3,   -21416477884929434181388631924023741361264598809816873076331133830318602596332088069733933047463884290695727838312 1
```

```
      52170109453)
(2,    1288852359116033410613791243994164241591905355073735634247165392805618377946014905015701508681255525992478185310146905542061)
(-8,   3163357815943921306495379915269786320411592267672643410223358096222549433470096155720769765128564997634716268579863745677 1)
(-5,  -1321001889705188081349892737410199533959581699250834689584960915480292822075788101381291893508778863463935154716199449891)
(5,   -1182076788551730602734619766770475116661074953562382333258507467774484339528870419622234444687351217744613298788422578 29)
(2,    2111927946171327759620701789350226653831314331196303085750441510781171013182825382234563175842409351590856581943897569 7)
(-2,  -2743285305114172425541110540802561105720020141938918426600397745755784059219379623215801476055360582029554774547226713)
(3,    2860748158891985401474429713136592782274727989378104287097773627056634739169980867676818174655427818291322493595179 75)
(-3,   1367532997912052465443644868991541513477548820779851985467270422519783676891382207571396439942634426120110281740330 9)
(-7,  -5733925658197840890157616477656092654752132021299440670896385313615178397931288947596754116299993867692880971575 73)
(3,    4642014124720289963275873549242830849165566501873785964876948824665393672719043131032957884408990038141451260148 11)
(-4,   2648752977947845396817832358670149346465038648366123949432443617527045265166163949185284933213945742646074855 73)
(-5,   2108559245863855227022919968753755644328349091487882633136164649958232365711910457701858011575588866966405009 37)
(-3,   2235894385477513774702853297152645208875741821470379502651453995593835452771449594075431944337959109115485481 1)
(-3,  -3066428918730050632325097660734163737851572230223614656315369265785635066318979992409078183621174963776878039)
(2,    1420865989874072755665282230298476831106299907933671932074879616218452312494744507627168783334527681703713 07)
(-10, -4013833186275051054261529002448266001854462865736283000730209262386311300176819714480584897953649188768987)
(3,    5133187827798145400999334148526572649747134934693745854533795436478351854566215608625239085628765532177 09)
(-2,  -6167743688256320381536911317831344419466606317713806120336190796903010939774300327391806056220720881742 1)
(4,   -5705343812063530411004779368065864956629980006638652578479437486915684727604688458627505624819542303243)
(3,   -9155818781739316346966742047891535700816738572323029795737035526883331905337014079460712753247612705 3)
(21,   4538685336850697187842499653670888781357885670866161805360478300528997110425684029899747272018435 37)
(-10,  2096400963120895452091855125365379202225167263766964847877535223464862248383150866105855804857668 83)
(-2,  -2452654164796263512959165565377597147123165793392501354290866811059027722610119847271723828796771)
(4,   -1470527936734615380865065306683575610942264782463158710865373829910403641681674649916472973816 5)
(4,    1396379881291112013027970946380118730029928835625361305509205527869439306688873128639700073563 7)
(-2,  -2243059469029860341733582498093820347780557194768743665298129222007849423064256685446565258585)
(3,   -3325369847370702933567816339534141091828891003805965525320260390920031496638268930678355279 79)
(3,   -5386967244970895407120103734398017221619479235338351726393439969384585546901687694925011 2145)
(2,    8809243662589899724441862302427489374063110417101583652380728682887905485821254651162082773)
(-3,   1523916850396224929683411116520822483473656939678806456889398425585339179937257634503878 43)
(-2,  -1754489364392193514749431904081409848801520964881650742860018735768133417632767916325280 51)
(4,   -7172119525392604758586912757900411673673727956088230007355764602327785267343320520772285)
(6,    4119835862177913257337980256296728495137802867857784860484730448194124783997117493312 93)
(-3,   3995237825404633164098524535985314848043967467204685269347794088687337119079150377 9555)
(-4,   2430418964222694480671212916136321892771426164807894584775183594905868264571397226917)
(-4,  -2004816389506792126513706962375499698367870035307726680846526580047822967554312546 61)
(3,   -1580903667060605871996966226809045877298060404628075628715566840184055844802721257)
(5,   -1523585718315717333817603063481778938222355282742962613553158680390832265726743003)
(2,    1843114936956169477779561058029154595977120008528310170603829047286098966387493 1)
(-4,   1860872135112964248738280512893474954558440050772629678744760840270585563475300 5)
(-3,   1869347445173273085208023229014781878050128548111311960144791632389450110085991)
(-4,   1783807300968843540335049091719016168942085295182686057073059993805288925377777)
(-3,   8087485289366444463103610663066728692803032028198886352664396077222417226 51)
(-8,   3309898705186043303215014982616604211642048557203333997938938487614396186 1)
(-3,  -1866453661596902048360161861366941356077297843206608429615574493135596039 3)
(6,   -1412504875497260093376398626643530047287435788085760178403662982434702419)
(2,    2199964927348448841457853722934605053489427558714791359395530729644637 95)
(-3,   2351604360566553921011925424106798286299309046100721528085455153587032 7)
(-3,  -3077511624656829785194435187496372328800530833832937512976380187853603)
(2,    2971521129174281864734874025301780049383610478829534623568875299744 35)
(-5,   2043047968784113838865178649033621466031951358193306410738753843972 5)
(-3,  -7157105225444615797643496386715830376878491722629619206484246416 23)
(9,    1417951347087597514948488495652560626163289875991704192430469253 1)
(-5,  -1282525333271264792977028780361879602218745339300046465268399291)
(2,    7133580613529858737331990335086812687125553016512866750439935)
(-89, -3140274313351805370923868077412836021349838155479525780249 1)
(3,   -5670014347280098278799124543796297689457759679846318408801)
(2,    4668843591720874170487843007845306559025103686844156105 49)
(-6,   3584612499799271885192020009484500568653151822992857983 5)
(-2,  -5174946979118686463949165359305774723028412521870150151)
(3,   -6386324325294959118689235547737361413294642450956512 5)
(27,   7425935781990473220367357982454273737732519281514 97)
(0,   -7425935781990473220367357982454273737732519281514 97)
```

Output: the reduced ideal $(A1, B1)$ equivalent to $\mathfrak{A}$

```
A1 = 9562398414327226521335535763343291861775258207781 49
B1 = -7425935781990473220367357982454273737732519281514 97
```

Chapter 3

PkQ cryptosystem

We propose a cryptosystem modulo $p^k q$ based on the RSA cryptosystem. We choose an appropriate modulus $p^k q$ which resists two of the fastest factoring algorithms, namely the number field sieve and the elliptic curve method. We also apply the fast decryption algorithm modulo p^k proposed in [Tak97]. The decryption process of the proposed cryptosystems is faster than the RSA cryptosystem using Chinese remainder theorem, known as the Quisquater-Couvreur method [QC82]. For example, if we choose the 1024-bit modulus $p^2 q$ for 341-bit primes p and q, then the decryption process of the proposed cryptosystem is about 3 times faster than that of RSA cryptosystem using Quisquater-Couvreur method.

3.1 Introduction

The RSA cryptosystem is one of the most practical public key cryptosystems and is used throughout the world [RSA78]. Let n be a public key, which is the product of two appropriate primes, e be an encryption key, and d be a decryption key. The algorithms of encryption and decryption consist of exponentiation to the e^{th} and d^{th} powers modulo n, respectively. We can make e small, but must consider low exponent attacks [CFPR96] [Cop96] [Ha88]. The encryption process takes less computation and is fast. On the other hand, the decryption key d must have more than one fourth the number of bits of the public key n to preclude Wiener's attack [Wie90] and its extension [VT97]. Therefore, the cost of the decryption process is dominant for the RSA cryptosystem.

In this chapter, we propose an RSA-type cryptosystem modulo $n = p^k q$. Even though the modulus is not of the form pq, we choose appropriate sizes for the secret primes p and q to preclude both the number field sieve and the elliptic curve method. Using this modulus $p^k q$, we construct a fast decryption public-key cryptosystem. In the key generation, we generate the public key e and secret key d using the relation $ed \equiv 1 \pmod{L}$, where $L = \text{LCM}(p-1, q-1)$. Note that L is not the same as

$\phi(n) = p^{k-1}(p-1)(q-1)$ or even $\lambda(n) = \mathrm{LCM}(p^{k-1}(p-1), q-1)$. Thus, the secret exponent d becomes much smaller than $n = p^k q$. Moreover, for decrypting $M_p \equiv M \pmod{p^k}$ we show that it is possible to apply the fast decryption algorithm proposed in [Tak97]. The running time for computing M_p is essentially equivalent to that for $C^d \pmod{p}$. Therefore, the decryption process is much faster than in the RSA cryptosystem using the Chinese remainder theorem [QC82].

The chapter is organized as follows. In Section 2, we describe the algorithm of the proposed cryptosystem. We discuss the size of the secret primes which prevents the use of both the number field sieve and the elliptic curve method in Section 3. Then, we show the running time of the proposed cryptosystem in comparison with the RSA cryptosystem using the Quisquater-Couvreur method in Section 4. We explain the effectiveness of Wiener's attack in Section 5. We show some properties of our cryptosystem related to some attacks in Section 6.

Notation: $\mathbf{Z}$ is an integer ring. $\mathbf{Z}_n$ is a residue ring $\mathbf{Z}/n\mathbf{Z}$ and its complete residue class is $\{0, 1, 2, \ldots, n-1\}$. $\mathbf{Z}_n^\times$ is a reduced residue group modulo n. $\mathrm{LCM}(m_1, m_2)$ is the least common multiple of m_1 and m_2. $\mathrm{GCD}(m_1, m_2)$ is the greatest common divisor of m_1 and m_2.

3.2 PkQ cryptosystem

In this section, we describe an RSA-type cryptosystem modulo $p^k q$, and discuss the size of its secret keys and the running time.

3.2.1 Algorithm

1. Generation of the keys: Generate two random primes p, q, and let $n = p^k q$. Compute $L = \mathrm{LCM}\,(p-1, q-1)$, and find e, d which satisfies $ed \equiv 1 \pmod{L}$ and $\mathrm{GCD}(e, p) = 1$. Then e, n are public keys, and d, p, q are the secret keys.

2. Encryption: Let $M \in \mathbf{Z}_n^\times$ be the plaintext. We encrypt the plaintext by the equation:

$$C \equiv M^e \pmod{n}. \tag{3.1}$$

3. Decryption: We decrypt $M_p \equiv M \pmod{p^k}$ and $M_q \equiv M \pmod{q}$ using the secret key d, p, q. The plaintext M can be recovered by the Chinese remainder theorem. Here, M_q is computed by $M_q \equiv C^d \pmod{q}$ and M_p is computed by the fast algorithm described in [Tak97].

3.2.2 Details of the decryption algorithm

The order of the group $\mathbf{Z}_{p^k}^{\times}$ is $p^{k-1}(p-1)$. When $M_p \equiv M \pmod{p^k}$ is recovered using the standard algorithm of RSA, we have to compute $M_p \equiv C^d \pmod{p^k}$ for $d \equiv e^{-1} \pmod{\mathrm{LCM}(p^{k-1}(p-1), q-1)}$. Then the running time is slower than that of the method using the Chinese remainder theorem for $n = pq$ [QC82], so there are no significant advantages in using the modulus $p^k q$. Instead, we apply the method described in [Tak97], where the author presents a fast algorithm for computing RSA decryption modulo n^k using n-adic expansion. Then, the running time for computing M_p becomes essentially equivalent to computing $M_p \equiv C^d \pmod{p}$ for $d \equiv e^{-1} \pmod{\mathrm{LCM}(p-1, q-1)}$.

First, we modify the algorithm into a more efficient form. We denote the ciphertext reduced modulo p^k by C_p. Then the relationship between the ciphertext C_p and the plaintext is $C_p \equiv M_p^e \pmod{p^k}$. Note that M_p the plaintext modulo p^k, has the p-adic expansion such that

$$M_p \equiv K_0 + pK_1 + p^2K_2 + \ldots + p^{k-1}K_{k-1} \pmod{p^k}. \tag{3.2}$$

Here, we define the function $F_i(X_0, X_1, \ldots, X_i)$ as follows:

$$F_i(X_0, X_1, \ldots, X_i) = (X_0 + pX_1 + \ldots + p^iX_i)^e,$$

where $i = 0, 1, \ldots, k-1$. $F_{k-1}(X_0 + pX_1 + \ldots + p^{k-1}X_{k-1})^e$ is the same as the function that encrypts the plaintext M_p in equation (3.2). By reducing modulo p^{i+1}, we get the relationship

$$F_i(X_0, X_1, \ldots, X_i) \equiv F_{i-1} + p^iG_{i-1}X_i \pmod{p^{i+1}},$$

where $F_{i-1} = F_{i-1}(X_0 + pX_1 + \ldots + p^{i-1}X_{i-1})^e$ and $G_{i-1} = e(X_0 + pX_1 + \ldots + p^{i-1}X_{i-1})^{e-1}$ for $i = 0, 1, \ldots, k-1$. From this relationship, we can recursively calculate $K_1, \ldots, K_{k-1}$. For $i = 1$, K_1 is the solution of the following linear equation of X_1:

$$C \equiv F_0(K_0) + pG_0(K_0)X_1 \pmod{p^2}. \tag{3.3}$$

Assume we have already calculated $K_1, K_2, \ldots, K_{i-1}$. Using these values, we compute $F_{i-1}(K_0, K_1, \ldots, K_{i-1}), G_{i-1}(K_0, K_1, \ldots, K_{i-1})$ in $\mathbf{Z}$, and denote them by F_{i-1}, G_{i-1}, respectively. Then, K_i is the solution of the following linear equation of X_i:

$$C \equiv F_{i-1} + p^iG_{i-1}X_i \pmod{p^{i+1}}. \tag{3.4}$$

Note that $(G_{i-1}, p) = 1$, because $\mathrm{GCD}(K_0, p) = \mathrm{GCD}(e, p) = 1$, so we can uniquely decrypt K_i.

After computing $K_0, K_1, \ldots, K_{k-1}$, we can evaluate $M_p \pmod{p^k}$ from equation (3.2). Finally, the plaintext $M \pmod{p^k q}$ is also computed from the values $M_p \pmod{p^k}, M_q \pmod{q}$, and the Chinese remainder theorem.

Moreover, note that we do not have to use the secret exponent d for evaluating $K_1, K_2, \ldots, K_{k-1}$. Thus, when we compute the two values of $K_0 \equiv C^d \pmod{p}$ and $M_q \equiv C^d \pmod{q}$, the secret exponent d can be reduced modulo $p-1$ and $q-1$. Indeed, $C^d \equiv C^{d_p} \pmod{p}$ and $C^d \equiv C^{d_q} \pmod{q}$ hold, where $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$.

In Appendix A, we describe the decryption program written in pseudo-code. For $x \in \mathbf{Z}$ and a positive integer N, $[x]_N$ denotes the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$.

3.3 Size of secret parameters

Here, we discuss the size of the secret parameters p and q. The RSA cryptosystem uses a composite number of the symmetry type pq, where p and q are the same bit size. The cryptosystem proposed in this chapter depends on the security of factoring the modulus $p^k q$. We have to carefully choose the size of p and q.

There are two types of fast factoring algorithm to consider: the number field sieve [LL91] and the elliptic curve method [Len87]. Other factoring algorithms have the same or slower running times, so the size of the RSA-modulus can be estimated by these two factoring algorithms [KR95] [MvOV97] [RS97] [Si00]. Let $L_N[s, c] = exp((c+o(1)) \log^s(N) \log\log^{1-s}(N))$. The number field sieve is the fastest factoring algorithm, and the running time is estimated from the total bit size of the integer n to be factored, which is expected as $L_n[1/3, (64/9)^{1/3}]$. If we choose n to be larger than 1024 bits, the number field sieve becomes infeasible. In our case, we have to make the modulus $n = p^k q$ larger than 1024 bits. The elliptic curve method is effective for finding primes which are divisors of the integer n to be factored. The running time is estimated in terms of the bit size of the prime divisor p. Its expected value is $L_p[1/2, 2^{1/2}]$. Note that the running time of the elliptic curve method is different from that of the number field sieve, and the order is much different. If we choose p to be larger than 341 bits, the elliptic curve method becomes infeasible. In our case, we have to make the primes p and q of the modulus larger than 341 bits.

The factoring algorithm strongly depends on the implementation. In my knowledge, the fastest implementation record for the number field sieves factored a 155-digit ($\approx$ 512-bit) RSA modulus [RSA155] and that for the elliptic curve method found a 53-digit ($\approx$ 175-bit) prime factor [ECM98]. Here, we again emphasize that there is a big difference in the cost between the number field sieve and the elliptic curve method. Therefore, if we choose the 1024-bit modulus $p^2 q$ with 341-bit primes p and q, neither of the factoring algorithms is feasible, so the scheme is secure for cryptographic purposes. But the size of secret primes must be thoroughly discussed for the practical usage of our proposed cryptosystem, and this is work in progress.

Here, we wonder if there exists factoring algorithms against the modulus with a square factor p^2q. This factoring problem appeared in the list of the open problems in number theoretic complexity by Adleman and McCurley [AM94], and it is unknown whether there exists $L_p[1/3]$-type sub-exponential algorithm which finds the primes of the composite number p^2q. Recently, Peralta and Okamoto proposed a factoring algorithm against numbers of the form p^2q based on the elliptic curve method [PO96]. They focused on the fact the Jacobi symbol is equal to one for a square integer, and the running time becomes a little bit faster than that of the original elliptic curve method.

A digital signature scheme [Oka90] and two public key cryptosystems [HJPT98] [OU98] which rely on the difficulty of factoring numbers of the type p^2q have been proposed. These cryptosystems are fast and practical. For secure usage of these cryptosystems and our proposed cryptosystem, the research of factoring algorithms against a composite number with a square factor is desirable.

3.4 Running time of PkQ cryptosystem

In this section, we estimate the running time of the proposed cryptosystem. We assume that the public modulus $n = p^2q$ is 1024 bits for 341-bit primes p and q in the following. We also assume the running time for computing $Z^a \pmod b$ is $O(\log_2^2(b)\log_2(a))$. Below, we estimate the worst-case running time.

In the decryption process of the proposed cryptosystem, the algorithm does not depend on the secret exponent d except when we compute

$$C^d \pmod p, \quad C^d \pmod q. \tag{3.5}$$

After calculating $C^d \pmod p$, we compute only a few multiplications for obtaining $M_p \equiv M \pmod{p^k}$. This costs the same as the encryption process. If we choose a very small e, this algorithm is very efficient. For example, if the modulus be p^2q, then we only compute at most $\lfloor \log_2 e \rfloor$ multiplications modulo p^2 and one division of p, two multiplications modulo p, and one inversion modulo p. Moreover, when we compute the two values of equation (3.5), the secret exponent d can be reduced modulo $p-1$ and $q-1$. In other words, $C^d \equiv C^{d_p} \pmod p$ and $C^d \equiv C^{d_q} \pmod q$ hold, where $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$. Thus, the size of the secret exponent can be reduced.

Denote by T the running time for computing the decryption algorithm of the original RSA cryptosystem, i.e., $C^{d'} \pmod n$, where d' is as large as n. Then, the running time of the proposed cryptosystem for a 1024-bit modulus is about $(2(1/3)^3+\alpha_e)T = (0.074+\alpha_e)T$, where α_e depends only on the encryption exponent e. When we make the encryption exponent e very small, α_e becomes negligible.

A similar decryption algorithm for the RSA cryptosystem using Chinese remainder theorem, the Quisquater-Couvreur method, mainly computes $C^d \pmod p$ and $C^d \pmod q$, where $n = pq$ is the RSA modulus, both p and q are as large as $(\log_2 n)/2$ bits, and we assume d is as large as p and q. So, the running time of Quisquater-Couvreur method is about 4 times faster than the original RSA cryptosystem.

Here, we compare the running time of our proposed cryptosystem with that of Quisquater-Couvreur method. The comparison is carried out based on the common bit length of the modulus. The proposed cryptosystem with the small encryption exponent e is about 3 times faster than the RSA cryptosystem applying the Quisquater-Couvreur method for the 1024-bit modulus.

In addition, consider the RSA cryptosystem with the square-free modulus $n = p_1 p_2 \cdots p_l$, where we assume that p_i are as large as $(\log_2 n)/l$ bits for $i = 1, 2, \ldots, l$. As we discussed in Section 3.3, we can use a 1024-bit modulus $n = p_1 p_2 p_3$ with 341-bit primes $p_i (i = 1, 2, 3)$ for the cryptographic purpose. This version of RSA will be faster when we use the decryption technique using the Chinese remainder theorem. Indeed, the decryption time with this modulus is dominant for computing $C^{d_i} \pmod{p_i}$, where we assume d_i are as large as p_i for $i = 1, 2, 3$. So, the running time of this RSA variant is about 9 times faster than the original RSA cryptosystem. Here, we compare this RSA variant with our proposed cryptosystem. Our proposed cryptosystem is about 1.5 times faster for a 1024-bit modulus.

3.5 Implementation data of PkQ cryptosystem

In order to demonstrate the improved efficiency of our decryption, we implemented our scheme using the LiDIA library [LiD95]. It should be emphasized here that our implementation was not optimized for cryptographic purposes — it is only intended to provide a comparison between the RSA cryptosystem and the PkQ cryptosystem. We implement two different sizes of moduli, namely a 1024-bit modulus $n = p^2 q$ for 341-bit primes p, q and a 2048-bit modulus $n = p^3 q$ for 512-bit primes p, q. These moduli are secure for cryptographic purposes (see Section 3.3 or reference [Si00]). The results are shown in table 3.1 for the 1024-bit modulus and in table 3.2 for the 2048-bit modulus.

	RSA	RSA with CRT	PkQ cryptosystem (k = 2)
Key Generation	2,256.91 ms	—	580.41 ms
Encryption	1.14 ms	—	1.20 ms
Decryption	118.68 ms	36.43 ms	15.25 ms

Table 3.1: Average timings for the PkQ cryptosystem compared to RSA cryptosystem for a 1024-bit modulus and $e = 2^{16} + 1$ over 1000 randomly chosen pairs of primes of the specified size on a Celeron 500 MHz using the LiDIA library

	RSA	RSA with CRT	PkQ cryptosystem (k = 3)
Key Generation	28,759.11 ms	—	2,344.38 ms
Encryption	4.34 ms	—	4.54 ms
Decryption	798.51 ms	235.23 ms	42.36 ms

Table 3.2: Average timings for the PkQ cryptosystem compared to RSA cryptosystem with a 2048-bit modulus and $e = 2^{16} + 1$ over 1000 randomly chosen pairs of primes of the specified size on a Celeron 500 MHz using the LiDIA library

3.6 Short secret exponent d

A short secret exponent is desirable for the fast decryption algorithm. However, Wiener reported an attack based on the continued fraction algorithm which detects a short secret exponent d [Wie90]. This attack is effective for $d < n^{1/4}$.

The secret key d and the public key e of the proposed cryptosystem have the relation $ed \equiv 1 \pmod{\text{LCM}(p-1, q-1)}$, and the primes p and q are much smaller than n. So, we wonder if Wiener's attack is applicable to larger secret exponents d. Moreover, if the attacker can compute d' such that

$$ed' \equiv 1 \pmod{\text{LCM}(p^{k-1}(p-1), q-1)}, \tag{3.6}$$

then proposed cryptosystem will also be broken.

Here, we discuss Wiener's attack for relation (3.6). From $\text{LCM}(p^{k-1}(p-1), q-1) = p^{k-1}(p-1)(q-1)/\text{GCD}(p^{k-1}(p-1), q-1)$, we have $ed' = 1 + mp^{k-1}(p-1)(q-1)/\text{GCD}(p^{k-1}(p-1), q-1)$ for some integer m. Generally, $\text{GCD}(p^{k-1}(p-1), q-1)$ is very small compared with p and q. Let $m/\text{GCD}(p^{k-1}(p-1), q-1) = h/g$, where $\text{GCD}(h, g) = 1$. Then, we get the relation

$$\left| \frac{e}{p^k q} - \frac{h}{gd'} \right| = \delta', \tag{3.7}$$

where $\delta' = \frac{h}{gd'} \frac{p^k + p^{k-1}q - p^{k-1} - g/h}{p^k q}$. From $h/d'g \leq 1$, the upper bound of δ' is of the size $n^{-1/(k+1)}$. It is known that for a rational number x such that $|x - P/Q| < 1/2Q^2$, P/Q is a convergent in the continued fraction of x, where P and Q are relatively prime integers. Therefore, if $n^{-1/(k+1)} < 1/2(gd')^2$ holds, then Wiener's attack is applicable by computing the continued fraction of $e/p^k q$. Therefore, Wiener's attack is effective for $d' < n^{\frac{1}{2(k+1)}}$. During key generation one must ensure that $d' \equiv e^{-1} \pmod{\text{LCM}(p^{k-1}(p-1), (q-1))}$ is sufficiently large.

In the same manner, we can discuss the Wiener's attack for the relation $ed \equiv 1 \pmod{\text{LCM}(p-1, q-1)}$. In this case, we get the relation

$$\left| \frac{e}{p^k q} - \frac{h}{gdp^{k-1}} \right| = \delta, \tag{3.8}$$

where $\delta = \frac{h}{gd}\frac{p+q-1-g/h}{p^k q}$. The lower bound on δ is of the size $1/gdn^{k/(k+1)}$, and $1/gdn^{k/(k+1)}$ is larger than the upper bound $1/2(gdp^{k-1})^2 \sim 1/2(gdn^{(k-1)/(k+1)})^2$ which the continued fraction can detect. So, Wiener's attack seems infeasible for the relation $ed \equiv 1 \pmod{\text{LCM}(p-1, q-1)}$. Further work on this is in progress.

3.7 Other properties

In this section, we describe some attacks against our proposed cryptosystem and some other properties of it.

Permutation: Let S be a finite set, and let $F(x)$ be a function from S to S. The function $F(x)$ is called a permutation function if every pair $x, y \in S$ that satisfies $F(x) = F(y)$ also satisfies $x = y$. The encryption function must be a permutation function in order to have unique decryption. The encryption function of the proposed cryptosystem is $F(X) \equiv X^e \pmod{p^k q}$. This function is a permutation function if and only if $\text{GCD}(p-1, e) = \text{GCD}(q-1, e) = \text{GCD}(p, e) = 1$. The last condition is always satisfied for small e, so this condition becomes the same as that for the original RSA cryptosystem.

Message concealing: A function $F(x)$ is called unconcealed when $F(x) = x$ holds for some x. If the encryption function is unconcealed, some plaintexts are not encrypted. Blakley and Borosh showed that the encryption function of the RSA cryptosystem is unconcealed [BB79]. And they also estimated the number of unconcealed messages for a modulus having the form $p^k q$. They proved

$$N = (1 + \text{GCD}(e-1, p^{k-1}(p-1)))(1 + \text{GCD}(e-1, (q-1))).$$

This number is negligible because we choose e to be small in our proposed cryptosystem.

Cycling attack: The cycling attack is to find an integer s such that $C^{e^s} \equiv C \pmod{p^k q}$ [Mau95] [WS79]. If we find such an integer, then the modulus $p^k q$ can be factored with probability greater than 1/2. From a recent result by Rivest and Silverman, it is known that the probability of the cycling attack success is negligible [RS97]. This analysis is also true for our proposed cryptosystem, because p and q must be chosen to be more than 341-bit primes. Here, denote by $\text{ord}_m(Q)$ the order of the point Q in the group $\mathbf{Z}_m$ for some integer m, and $\text{ord}_{\text{ord}_n(C)}(e) | s$ holds. Note that $\text{ord}_m(Q) | \text{ord}_n(Q)$ for $m | n$ and Q in $\mathbf{Z}_n$. The probability that $p | \text{ord}_{p^k}(Q)$ for a random point Q in $\mathbf{Z}_{p^k}$ is $1 - 1/p$, so $p | \text{ord}_n(C)$ holds for a random ciphertext C in $\mathbf{Z}_n$ with high probability, and $\text{ord}_p(e)$ is greater than the largest prime of $p-1$, which is more than 50 bits with high probability. Therefore, the integer s is greater than 50 bits with high probability.

Other attacks: All other attacks are applicable, for example, the low exponent attacks [CFPR96] [Cop96] [Ha88], the common modulus attack, and the chosen message attack (See, for example, [KR95] [MvOV97]).

Digital signature: Of course, the proposed algorithm can be used for a digital signature.[1] The prominent property of our proposed cryptosystem is the running time for generating the signature, which it is faster than that of the RSA cryptosystem using Chinese remainder theorem.

Rabin-type cryptosystem: We can construct a Rabin-type cryptosystem by applying the algorithm proposed in this chapter. We can also prove that the extended Rabin-type cryptosystem is as intractable as factoring the modulus $p^k q$.

3.8 Conclusion

We proposed a RSA-type cryptosystem using the modulus $p^k q$. We choose the modulus to be for example 1024-bit $p^2 q$ for the 341-bit primes p and q, in order to make both the elliptic curve method and the number field sieve infeasible. So, this modulus are secure against the fast factoring algorithms. When we use this modulus, the proposed cryptosystem is about 3 times faster than the RSA cryptosystem using the Quisquater-Couvreur method.

Appendix A: Decryption algorithm of the PkQ cryptosystem

In this appendix, we describe the decryption program written in pidgin ALGOL. For $x \in \mathbf{Z}$ and a positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$. The plaintext M is encrypted by $C \equiv M^e \pmod{p^k q}$. The relation between the encryption exponent e and the decryption exponent d is $ed \equiv 1 \pmod{\mathrm{LCM}(p-1, q-1)}$.

procedure **PkQ Decryption**:

INPUT: d, p, q, e, k, C

OUTPUT: M

(1) $d_p := [d]_{p-1}, d_q := [d]_{q-1}$;

(2) $K_0 := [C^{d_p}]_p, M_q := [C^{d_q}]_q$;

(3) $A_0 := K_0$;

FOR $i = 1$ **to** $(k-1)$ **do**

[1] Shamir proposed a variation of RSA cryptosystem with an unbalanced modulus [Sha95]. As he stated in the paper, Shamir's RSA can not be used for digital signatures.

$F_i := [{A_{i-1}}^e]_{p^{i+1}}$;

$E_i := [C - F_i]_{p^{i+1}}$;

$B_i := E_i/p^i$ in $\mathbf{Z}$;

$K_i := [(eF_i)^{-1} A_{i-1} B_i]_p$;

$A_i := A_{i-1} + p^i K_i$ in $\mathbf{Z}$;

(4) $M_{p^k} := A_{k-1}$;

(5) $p_1 := [(p^k)^{-1}]_q, v_1 := [(M_q - M_{p^k}) p_1]_q$

(6) $M \equiv M_{p^k} + p^k v_1 \bmod n$

Chapter 4

Nk cryptosystem

We propose two RSA-type cryptosystems using n-adic expansion, where n is the public key. These cryptosystems can have more than one block as a plaintext space, and the decrypting process is faster than any other multi-block RSA-type cryptosystem ever reported. Deciphering the entire plaintext of this system is as intractable as breaking the RSA cryptosystem or factoring. Even if a message is several times longer than a public key n, we can encrypt the message fast without repeatedly using the secret key cryptosystem.

4.1 Introduction

The RSA cryptosystem is one of the most practical public key cryptosystems and is used throughout the world [RSA78]. Let n be a public key, which is the product of two appropriate primes, e be an encryption key, and d be a decryption key. The algorithms of encryption and decryption consist of the e-th and d-th power modulo n, respectively. We can make e small by considering the low exponent attacks [CFPR96] [Cop96] [Ha88]. The encryption process uses less computation and is fast. On the other hand, we must keep the decryption key d up to the same size as the public key n to preclude Wiener's attack [Wie90]. Therefore, the cost of the decryption process is dominant for the RSA cryptosystem.

If a cryptosystem has more than one block of plaintexts, where each block is as large as the public-key n, we call it a multi-block cryptosystem. A lot of multi-block RSA-type cryptosystems have been proposed [Dem94] [KMOV92] [Koy95] [LKBS92] [MM96] [SE96]. Their advantage is that they allow us to encrypt data larger than the public-key at a time, and we can prove their security is equivalent to the original RSA cryptosystem or factoring. However, these algorithms are very slow and the attacks against the RSA cryptosystem are also applicable to them (See, for example, [Kal97] [TN96].). We cannot find significant advantage over using the original RSA cryptosystem for each block.

In this chapter, we propose two methods of constructing fast multi-block RSA-type cryptosystems. We express the plaintext as an n-adic expansion, where n is the public key. The features of this method are as follows. We can take an arbitrary number of blocks as a plaintext. To implement the proposed cryptosystems, we use only ordinary and elementary mathematical techniques i.e., the greatest common divisor, so the designer can easily make them. Deciphering the entire plaintext of the proposed cryptosystems is as hard as breaking the original RSA cryptosystem or factoring. Moreover, the decryption speed is much faster than any previously proposed multi-block RSA-type cryptosystems. Decryption time of the first block is dominant, because we calculate the modular multiplication of the encryption exponent and a greatest common divisor to decrypt blocks after the first one. Even if a message is several times longer than a public-key n, we can encrypt the message fast without repeatedly using the secret key cryptosystem.

Notation: $\mathbf{Z}$ is an integer ring. $\mathbf{Z}_n$ is a residue ring $\mathbf{Z}/n\mathbf{Z}$ and its complete residue class is $\{0, 1, 2, \ldots, n-1\}$. $\mathbf{Z}_n^{\times}$ is a reduced residue group modulo n. $\mathrm{LCM}(m_1, m_2)$ is the least common multiple of m_1 and m_2. $\mathrm{GCD}(m_1, m_2)$ is the greatest common divisor of m_1 and m_2. ${}_lC_m$ is permutation theory notation meaning the number of ways of choosing m from l.

4.2 Nk cryptosystem

In this section, we describe how to extend the RSA cryptosystem using n-adic expansion, and discuss its security and running time.

4.2.1 Algorithm

1. Generation of the keys: Generate two appropriate primes p, q, and let $n = pq$. Compute $L = \mathrm{LCM}\,(p-1, q-1)$, and find e, d which satisfies $ed \equiv 1 \pmod{L}$, $\mathrm{GCD}\,(e, L) = 1$ and $\mathrm{GCD}\,(e, n) = 1$. Then e, n are public keys, and d is the secret key.

2. Encryption: Let $M_0 \in {\mathbf{Z}_n}^{\times}$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$ be the plaintext. We encrypt the plaintexts by the equation:

$$C \equiv (M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^e \pmod{n^k}. \tag{4.1}$$

3. Decryption: First, we decrypt the first block M_0 by the secret key d:

$$M_0 \equiv C^d \pmod{n}. \tag{4.2}$$

This is the same decryption process as in the original RSA. For the remaining blocks $M_1, M_2, \ldots, M_{k-1}$, we can decrypt by solving the linear equation modulo n.

4.2.2 Details of decryption

Assume that we have already decrypted M_0 by the decryption method of the original RSA cryptosystem, and we write down the process to find $M_1, M_2, \ldots, M_{k-1}$ as follows.

Consider that the encryption function (4.1) is the polynomial of the variables X_0, $X_1, \ldots, X_{k-1}$ such that

$$E(X_0, X_1, \ldots, X_{k-1}) = (X_0 + nX_1 + \ldots + n^{k-1}X_{k-1})^e.$$

Expand the polynomial $E(X_0, X_1, \ldots, X_{k-1})$ by the polynomial theorem:

$$\sum_{\substack{0 \le s_0, s_1, \ldots, s_{k-1} \le e \\ s_0 + s_1 + \ldots + s_{k-1} = e}} \frac{e!}{s_0! s_1! \ldots s_{k-1}!} X_0^{s_0} (nX_1)^{s_1} \ldots (n^{k-1}X_{k-1})^{s_{k-1}}.$$

And let

$$\begin{aligned} \Gamma_i := \{(s_0, s_1, \ldots, s_i) \quad | \quad & s_1 + 2s_2 + \ldots + is_i = i, \\ & s_0 + s_1 + \ldots + s_i = e, 0 \le s_0, s_1, \ldots, s_i \le e\}, \end{aligned}$$

where $(0 \le i \le k-1)$. Let $D_i(X_0, X_1, \ldots, X_i)$ be the coefficient of $n^i (0 \le i \le k-1)$. For $i = 0, 1, \ldots, k-1$, we can find $D_i(X_0, X_1, \ldots, X_i)$ by calculating

$$D_i(X_0, X_1, \ldots, X_i) = \sum_{(s_0, s_1, \ldots, s_i) \in \Gamma_i} \frac{e!}{s_0! s_1! \ldots s_i!} X_0^{s_0} X_1^{s_1} \ldots X_i^{s_i}. \tag{4.3}$$

Here, we write them down with small i as follows:

$$\begin{aligned} & D_0(X_0) = M_0^e, \\ & D_1(X_0, X_1) = eM_0^{e-1}M_1, \\ & D_2(X_0, X_1, X_2) = {}_eC_2 M_0^{e-2}M_1^2 + eM_0^{e-1}M_2, \\ & D_3(X_0, X_1, X_2, X_3) = {}_eC_3 M_0^{e-3}M_1^3 + 2{}_eC_2 M_0^{e-2}M_1M_2 + eM_0^{e-1}M_3, \\ & D_4(X_0, X_1, \ldots, X_4) = {}_eC_4 M_0^{e-4}M_1^4 + 3{}_eC_3 M_0^{e-3}M_1^2M_2 + {}_eC_2 M_0^{e-2}M_2^2 + eM_0^{e-1}M_4, \\ & D_5(X_0, X_1, \ldots, X_5) = {}_eC_5 M_0^{e-5}M_1^5 + 4{}_eC_4 M_0^{e-4}M_1^3M_2 + 3{}_eC_3 M_0^{e-3}M_1M_2^2 \\ & \quad + 2{}_eC_2 M_0^{e-2}M_2M_3 + 2{}_eC_2 M_0^{e-2}M_1M_4 + eM_0^{e-1}M_5, \\ & D_6(X_0, X_1, \ldots, X_6) = {}_eC_6 M_0^{e-6}M_1^6 + 5{}_eC_5 M_0^{e-5}M_1^4M_2 + 4{}_eC_4 M_0^{e-4}M_1^3M_3 \\ & \quad + 3{}_eC_3 M_0^{e-3}M_1^2M_4 + {}_eC_3 M_0^{e-3}M_2^3 + {}_eC_2 M_0^{e-2}M_3^2 \\ & \quad + 2{}_eC_2 M_0^{e-2}M_2M_4 + 2{}_eC_2 M_0^{e-2}M_1M_5 + eM_0^{e-1}M_6, \\ & \ldots \\ & D_{k-1}(X_0, X_1, \ldots, X_{k-1}) = \{\text{ polynomial of } M_0, M_1, \ldots, M_{k-1}\}. \end{aligned}$$

We show the algorithm of decryption. Note that the terms that include X_i do not appear in $D_j\,(j < i)$, and the only term that includes X_i in D_i is $eX_0^{e-1}X_i$ for $i = 0, 1, \ldots, k-1$. We define

$$D_i'(X_0, X_1, \ldots, X_{i-1}) = D_i(X_0, X_1, \ldots, X_i) - eX_0^{e-1}X_i.$$

Therefore, the terms $D_0, D_1, \ldots, D_{i-1}, D_i'$ are the polynomial of $X_0, X_1, \ldots, X_{i-1}$ $(0 \leq i \leq k-1)$.

From this relation, we can inductively decrypt M_i after decrypting $M_0, M_1, \ldots, M_{i-1}$ $(0 \leq i \leq k-1)$. Indeed, $M_1, M_2, \ldots, M_{k-1}$ are calculated as follows. At first, let $i = 1$. The relations $D_1'(X_0) = 0$ and $D_0(X_0) = X_0^e$ hold. So, the solution of the linear equation

$$eM_0^{e-1}x \equiv B_1 \pmod{n}, \quad B_1 = E_1/n, \tag{4.4}$$

$$E_1 \equiv C - D_0(M_0) \pmod{n^2},$$

is M_1, because M_0 and e are in the reduced residue class modulo n such that ${\mathbf{Z}_n}^{\times}$. Provided that we decrypt $M_1, M_2, \ldots, M_{i-1}$, in the same manner we can uniquely decrypt M_i by solving the linear equation

$$eM_0^{e-1}x \equiv B_i \pmod{n}, \quad B_i = E_i/n^i, \tag{4.5}$$

$$E_i \equiv C - \sum_{j=0}^{i-1} n^j D_j(M_0, M_1, \ldots, M_j) - n^i D_i'(M_0, M_1, \ldots, M_{i-1}) \pmod{n^{i+1}}.$$

Inductively, we can decrypt all plaintexts $M_1, M_2, \ldots, M_{k-1}$.

Here, we describe the decryption program written in the pidgin ALGOL in the following. For $x \in \mathbf{Z}$ and positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$.

procedure **Nk Decryption**:

INPUT: $d, n, C(:= [(M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^e]_{n^k})$

OUTPUT: $M_0, M_1, \ldots, M_{k-1}$

(1) $C_0 := [C]_n$;

$M_0 := [C_0^d]_n$;

(2) $D_0 := [M_0^e]_{n^2}$;

$E_1 := [C - D_0]_{n^2}$;

$B_1 := E_1/n$ in $\mathbf{Z}$;

$A := [(eC_0)^{-1}M_0]_n$;

$M_1 := [AB_1]_n$;

(3) **FOR** $i = 2$ **to** $(k-1)$ **do**

begin

SUM := 0;

FOR $j = 0$ **to** $(i-1)$ **do**

begin

$D_j := [D_j(M_0, M_1, \ldots, M_j)]_{n^{i+1}}$;

SUM := $[\text{SUM} + n^j D_j]_{n^{i+1}}$

end

$D'_i := [D'_i(M_0, M_1, \ldots, M_{i-1})]_{n^{i+1}}$;

$E_i := [C - \text{SUM} - n^i D'_i]_{n^{i+1}}$;

$B_i := E_i/n^i$ in $\mathbf{Z}$;

$M_i := [AB_i]_n$

end

4.2.3 Permutation

Let S be a finite set, and let $F(x)$ be a function from S to S. The function $F(x)$ is called a permutation function if every pair $x, y \in S$ that satisfies $F(x) = F(y)$ also satisfies $x = y$. If the encryption function $F(x)$ is not a permutation, we cannot uniquely decrypt a ciphertext. It is known that the encryption function of the RSA cryptosystem is a permutation, if and only if the relation $\text{GCD}(e, L) = 1$ holds with the same notation as in section 4.2.1. In the previous section, we showed that if the conditions $\text{GCD}(e, L) = 1$ and $\text{GCD}(e, n) = 1$ are satisfied, the proposed cryptosystem can be uniquely decrypted i.e., it is a one-to-one function.

Here, the encryption function of the proposed cryptosystem is defined from ${\mathbf{Z}_{n^k}}^\times$ to ${\mathbf{Z}_{n^k}}^\times$. We can prove this function is a permutation if and only if the conditions $\text{GCD}(e, L) = 1$ and $\text{GCD}(e, n) = 1$ hold.

Actually, the reduced residue group modulo n^k such that ${\mathbf{Z}_{n^k}}^\times$ is decomposed into two products such that

$$ {\mathbf{Z}_{n^k}}^\times \cong {\mathbf{Z}_{p^k}}^\times \times {\mathbf{Z}_{q^k}}^\times. \tag{4.6} $$

Both groups are cyclic groups whose orders are $p^{k-1}(p-1)$ and $q^{k-1}(q-1)$, respectively. Therefore, the order of the group ${\mathbf{Z}_{n^k}}^\times$ is $n^{k-1}(p-1)(q-1)$. All elements in ${\mathbf{Z}_{n^k}}^\times$ are expressed by n-adic expansion such that

$$ M = M_0 + nM_1 + \ldots + n^{k-1}M_{k-1}, $$

where $M_0 \in \mathbf{Z}_n{}^{\times}$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$. This is the reason that the plaintext must have the form in equation (4.1).

Let $E(x) \equiv x^e \pmod{n^k}$ be the encryption function of the proposed cryptosystem. Suppose $E(x) \equiv E(y) \pmod{n^k}$, and we get $(x/y)^e \equiv 1 \pmod{n^k}$. By Chinese remainder theorem, we reduce the equation into modulo p^k. Let g be a primitive root of modulo p^k, and let $x/y \equiv g^j \pmod{p^k}$ for some j. We get $g^{je} \equiv 1 \pmod{p^k}$, and

$$je \equiv 0 \pmod{p^{k-1}(p-1)}.$$

If $E(x)$ is a permutation, this equation must be solvable and all solutions are different, so $\mathrm{GCD}(e, p) = 1$ and $\mathrm{GCD}(e, (p-1)) = 1$ holds. Therefore, we have to choose e such that $\mathrm{GCD}(e, n) = 1$ and $\mathrm{GCD}(e, L) = 1$. The criteria in the key generation of the proposed cryptosystem are necessary.

4.2.4 Security

4.1. Theorem *When plaintexts are uniformly distributed, finding the entire plaintext from the ciphertext for the RSA cryptosystem is as intractable as doing it for the proposed n-adic RSA-type cryptosystem.*

Proof: Using a black-box which can decipher the RSA cryptosystem, we can decipher the first block. Moreover, we can also decrypt blocks after the first one by using the decryption algorithm in section 4.2.2, so the entire plaintext is deciphered. Conversely, we are given ciphertext C, which is the result of encrypting a random $M \pmod{n}$ by the RSA cryptosystem. Let C' be a random n-adic ciphertext, whose plaintext M' satisfies $M' \equiv M \pmod{n}$. All the bits of M' are uniquely distributed since M is random, and we can use the black box for the n-adic system to recover M'. Hence, we can decipher the plaintext M. ■

All the attacks against the RSA cryptosystem (See, for example, [MvOV97] [KR95].) are also applicable to the proposed system, because if we can decipher the first block M_0, then we can recover all the following blocks using relationships (4.4) or (4.5).

Here, we wonder whether the proposed cryptosystem has extra flaws in terms of using a non-square modulo n^k. The attacks that might break it are the message concealing [BB79] and the cycling attacks [WS79]. In the following two sections, we show these attacks never work against the proposed cryptosystem.

4.2.5 Message concealing

A function $F(x)$ is called unconcealed when $F(x) = x$ holds for all x. If a function of a cryptosystem is unconcealed, then we cannot encrypt any message by it. G. R. Blakley and I. Borosh showed that the encryption function of the RSA cryptosystem is unconcealed [BB79]. Let N be the number of residue classes x modulo n^k such that $x^e \equiv x \pmod{n^k}$. They proved

$$N = (1 + \mathrm{GCD}(e-1, p^{k-1}(p-1)))(1 + \mathrm{GCD}(e-1, q^{k-1}(q-1))).$$

If $\mathrm{GCD}(e-1, pq) > 1$ holds, then N becomes very large. We have to choose the system parameters p, q and e described in section 4.2.1 to preclude this failure. It must be noted that if e is selected smaller than p and q, then $\mathrm{GCD}(e-1, pq) = 1$ holds.

Moreover, they also showed that if e is an odd integer larger than 2, then $N = 9$ if and only if

$$\mathrm{GCD}(e-1, \lambda) = 2, \quad \lambda = \mathrm{LCM}((p-1)p^{k-1}, (q-1)q^{k-1}).$$

For example, the RSA cryptosystem has only 9 unconcealed messages if $\mathrm{GCD}(e-1, L) = 2$. For small e, we have $\mathrm{GCD}(e-1, pq) = 1$, and N for the proposed cryptosystem is equal to that of the RSA cryptosystem.

4.2.6 Cycling attacks

It is known that the RSA cryptosystem is broken without factoring n when a ciphertext C has a period such that $C^{P(b)} \equiv 1 \pmod{n}$, where $P(t)$ is a polynomial and $t = b$ is an integer. Actually, if the relation holds, the plaintext can be recovered by computing $M \equiv C^Q \pmod{n}$, where Q satisfies $eQ \equiv 1 \pmod{P'}$ and $P' = P(b)/\mathrm{GCD}(e, P(b))$. Moreover, this analysis is true even if the modulo n is changed to n^k. To break the proposed n-adic RSA-type cryptosystem, an attacker would have to find the polynomial $P(t)$ and the value $t = b$, which have the relation $C^{P(b)} \equiv 1 \pmod{n^k}$. By decomposing of the group $\mathbf{Z}_{n^k}$ like (4.6), we reduce the relations to

$$P(t) \equiv 0 \pmod{p_i}, \quad P(t) \equiv 0 \pmod{q_i} \quad (i = 1, 2), \tag{4.7}$$

where $p_1 = p$; $p_2 = q$; and q_i is a large prime such that $q_i | p_i - 1 (i = 1, 2)$. H. C. Williams and B. Schmid [WS79] showed that the possibility of this polynomial satisfying equation (4.7) is very small, unless $P(t) = t \pm 1$ and $t = e^m$. Therefore, the designers must make m very large to preclude this attack. One method is to have $q_i - 1$ and $p_i - 1$ be divisible by large primes r_i and r_i' such that $r_i | p_i - 1$ and $r_i' | q_i - 1$; then $r_i | m$ and $r_i' | m$ hold for $i = 1, 2$ and m becomes very large. Since $p_i - 1 (i = 1, 2)$ must be divisible by a large prime to prevent the factoring algorithm called Pollard's $p-1$ method, we do not need worry about the equations

$e^m \equiv \pm 1 \pmod{p_i}$. Consequently, the proposed n-adic RSA-type cryptosystem is secure against this attack according to the same treatment as used for the original RSA cryptosystem.

4.2.7 Running time

Here, we discuss the running time of the proposed cryptosystem. In the encryption process, we have to compute the e-th power modulo $n^k (k \geq 2)$. As k increases, the running time becomes longer. However, it is possible to make the exponent of the encryption e small, since considering the low exponent attacks [CFPR96] [Cop96] [Ha88], the encryption cost is not so expensive.

Next, we consider the decryption process. The first block is decrypted by the same algorithm as in the RSA cryptosystem, and we should make the exponent d as large as the public modulus n to avoid Wiener's attack [Wie90]. Therefore, the decryption of the first block is the most expensive task. After the first block, we have to generate linear equation (4.4) and maybe also (4.5), and solve it/them. The ciphertext $C_i (i \geq 1)$ is expressed by the polynomial of $M_i (i \geq 1)$ and the task of computing the polynomial is essentially to calculate M_0^e. Therefore, it costs the same as the encryption process to generate the linear equations. Solving a linear equation is fast, so the decryption time after the first block also becomes as fast as the encryption process. If we choose a very small e, this algorithm becomes very efficient. For example, let the number of blocks be two. We can generate the linear equation to compute equation (4.4), which are at most $2\lfloor \log_2 e \rfloor$ multiplications modulo n^2 and one division of n^2, and to solve it, which are two multiplications modulo n and one inversion modulo n.

On the other hand, several multi-block RSA-type cryptosystems have been proposed [Dem94] [KMOV92] [LKBS92]. Their decryption time is l times slower than the original RSA cryptosystem, where l is the number of blocks. Our proposed cryptosystem is much faster than these cryptosystems, as showed by the above analysis. [1]

4.2.8 Effectiveness

As we discussed in the previous sections, the proposed n-adic RSA-type cryptosystem has several effective features. The most significant points are being as hard as breaking the original RSA cryptosystem and providing fast decryption for messages longer than the public key n.

[1] K. Koyama proposed a two-block cryptosystem having fast decryption by using singular cubic curves. But it only has two blocks [Koy95].

By the way, the RSA cryptosystem is slower than the secret-key cryptosystem, so the RSA cryptosystem is used to encrypt a session key of the secret-key cryptosystem to overcome this disadvantage. However, its theoretical security level must be estimated from the RSA cryptosystem and the secret-key. We do not have to use the secret-key cryptosystem, if the length of the data is shorter than a public-key n.

For a message that is several times longer than the public-key n, our proposed n-adic RSA-type cryptosystem is very efficient. We can encrypt such a message much faster.

Moreover, it is expected that the encryption speed of the RSA cryptosystem will reach 1 Mbits/second within a year or so [RSAFAQ]. The proposed method can contribute to the attainment of the fast encryption speed.

4.2.9 Implementation data of Nk cryptosystem

In order to demonstrate the improved efficiency of our decryption, we implemented our scheme using the LiDIA library [LiD95]. It should be emphasized here that our implementation was not optimized for cryptographic purposes — it is only intended to provide a comparison between the RSA cryptosystem and the Nk cryptosystem. The results are shown in table 4.1.

	RSA	Nk cryptosystem ($k = 2$)	Nk cryptosystem ($k = 3$)
Encryption	1.14 ms	4.27 ms	9.51 ms
Decryption	118.68 ms	125.10 ms	136.25 ms

Table 4.1: Average timings for the Nk cryptosystem compared to RSA cryptosystem with a 1024-bit modulus n and $e = 2^{16} + 1$ for over 1000 randomly chosen pairs of primes of the specified size on a Celeron 500 MHz using the LiDIA library

4.3 The Nk Rabin cryptosystem

In this section, we describe how to extend the Rabin cryptosystem using n-adic expansion. The discussion is similar to the extension of the RSA cryptosystem.

4.3.1 Algorithm

1. Generating keys: Generate two appropriate primes p, q, and let $n = pq$. Here, p and q are the secret keys, and n is the public key.

2. Encryption: Let $M_0 \in {\mathbf{Z}_n}^\times$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$ be the plaintext. We encrypt the plaintext by

$$C \equiv (M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^2 \pmod{n^k}. \tag{4.8}$$

And we send the ciphertext C.

3. Decryption: We solve the modular quadratic equation

$$x^2 \equiv C \pmod{n^k}. \tag{4.9}$$

Then the solutions are just plaintext $M_0, M_1, \ldots, M_{k-1}$.

4.3.2 Details of decryption

First, we decrypt the first block M_0. We solve the quadratic equation $C \equiv M_0^2$ modulo primes p and q. Here, several algorithms to solve the quadratic equation modulo a prime p are known, and the fastest one can be computed in sub-quadratic polynomial time [KS95]. Next, we decrypt the first block of the plaintext M_0 by the Chinese remainder theorem. The degree of ambiguity is 4 for the decryption modulo n, because we have two solutions of each quadratic equation. And we can eliminate the ambiguousness by adding redundancy bits, and we can get the true plaintext.

Next, we discuss the decryption of the remaining blocks $M_1, M_2, \ldots, M_{k-1}$. The process is similar to the case in the RSA cryptosystem. For M_1, we have the linear equation modulo n^2,

$$M_0^2 + 2nM_0x \equiv C \pmod{n^2}. \tag{4.10}$$

And this equation is solvable because $2M_0 \in (\mathbf{Z}/n\mathbf{Z})^\times$, and the solution is M_1. Here, assume that we already decrypt $M_0, M_1, \ldots, M_{i-1}$, and we can uniquely decrypt M_i by solving

$$2n^iM_0x \equiv C - \sum_{\substack{l+m \le i \\ 0 \le l,m \le i-1}} n^{l+m}M_lM_m \pmod{n^{i+1}}, \tag{4.11}$$

Therefore, we can decrypt all plaintext blocks $M_0, M_1, M_2, \ldots, M_{k-1}$.

We describe the decryption program written in the pidgin ALGOL in the following. For $x \in \mathbf{Z}$ and positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is $\{0, 1, \ldots, N-1\}$.

procedure **Nk Rabin Decryption**:

INPUT: $p, q, n, C(:= [(M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^2]_{n^k})$

OUTPUT: $M_0, M_1, \ldots, M_{k-1}$

(1) $C_0 := [C]_n$;

decrypt M_0 using p, q, C_0;

(2) **FOR** $i = 1$ **to** $(k-1)$ **do**

begin

SUM $:= 0$;

FOR $l = 0$ **to** $(i-1)$ **do**

FOR $m = 0$ **to** $(i-1)$ **do**

WHILE $l + m \leq i$ **do**

begin

$D := [n^{l+m} M_l M_m]_{n^{i+1}}$;

$\text{SUM} := [\text{SUM} + D]_{n^{i+1}}$

end

$E_i := [C - \text{SUM}]_{n^{i+1}}$;

$B_i := E_i / n^i$ in $\mathbf{Z}$;

$M_i := [(2M_0)^{-1} B_i]_n$

end

4.3.3 Security

4.2. Theorem *Completely breaking the proposed n-adic Rabin-type cryptosystem is as intractable as factoring.*

Proof: Let p, q be primes, and let $n = pq$. The complexities of the following three algorithms only differ by polynomial time.

(I) to factor $n = pq$

(II) to find the solution of the quadratic equation modulo n

(III) to find the solution of the quadratic equation modulo n^k,

where k is an integer greater than 2. (I) and (II) are clearly equivalent because the security of the Rabin cryptosystem is the same as factoring [Ra79]. (III) $\Rightarrow$ (II)

is true by reducing the solution in (II) modulo n. (II) $\Rightarrow$ (III) is true because it is just the decryption process after the first block in the previous section, and the algorithm only takes polynomial time to generate and solve linear equations. Here, (III) is just the algorithm deciphering the proposed n-adic system. ■

The exponent of the Rabin cryptosystem is only 2, so the low exponent attacks are applicable to it [CFPR96] [Cop96] [Ha88]. However, we can preclude these attacks by padding a plaintext with random bits.

4.3.4 Running time and effectiveness

Here, we discuss the running time of the proposed cryptosystem. In the encryption process, we only compute the second power modulo $n^k (k \geq 2)$, which is very fast. For the decryption process, the first block is decrypted by the same decryption method as for the Rabin cryptosystem. The decryption of the first block is the most expensive task. After the first block, we have to generate the linear equation (4.10) and maybe also (4.11), and solve it/them. These are computed very fast, and the cost is very small compared with the cost of decrypting the first block. Therefore, the total cost of the decryption is essentially the cost of the first block.

On the other hand, several multi-block Rabin-type cryptosystems have been proposed [MM96] [SE96]. We have to solve a polynomial with more than two degrees over the finite field of a prime order. Solving polynomials of higher degree is more expensive than solving a quadratic polynomial, and makes the decryption process ambiguous and restricts the form of the secret primes. These cryptosystems have few advantages.

From the above analysis, our proposed cryptosystem is much faster than these cryptosystems, and easy to implement. Designers do not have to code a complicated algorithm and can use only ordinary mathematical tools such as the greatest common divisor.

As we discussed in section 4.2.8, for messages that are several times longer than the public-key n, our proposal n-adic Rabin cryptosystem is very efficient. We can encrypt a message with the running time of the first block.

4.4 Open problems and a partial solution

A plaintext of the proposed n-adic cryptosystem modulo n^k has the form $M \equiv M_0 + nM_1 + \ldots + n^{k-1}M_{k-1}$. Theorems 4.1 and 4.2 show that breaking the entire plaintext M is as hard as breaking the RSA cryptosystem or factoring. Here, we mention some problems concerning the security of each block $M_0, M_1, \ldots, M_{k-1}$.

If we have an algorithm that breaks the first block M_0, we can decipher the RSA or Rabin cryptosystem. However, it is an open problem whether you can find the blocks after the first one without deciphering the first block. One strategy for finding such an algorithm is to seek some algebraic relations between a ciphertext and blocks after the first one. Indeed, the most trivial relation is linear equation (4.4) or (4.5) whose solutions are the remaining blocks after the first one. But, we have to compute the value M_0^{e-1} to construct them, which is as hard as deciphering the RSA cryptosystem.

W. Alexi et al. showed that we can find the whole plaintext by using an algorithm that deciphers certain bits of the plaintext [ACGS88]. This also means that the proposed n-adic system can be broken by an algorithm that deciphers certain bits of the first block of the plaintext. It is an open problem whether there exists an algorithm that can decipher certain bits after first block of the plaintext.

Against the RSA cryptosystem, D. Coppersmith et al. showed that we can recover the original plaintext by algebraic calculation, if we send two ciphertexts whose plaintexts have a polynomial relationship [CFPR96]. It might be possible to recover the plaintext of the proposed n-adic system using a variation of this technique. It is an open problem whether you can recover the plaintext if there is a polynomial relationship between some blocks of one plaintext or between blocks of two plaintexts.

4.4.1 Security of the second block

4.3. Theorem *Consider the n-adic RSA-type cryptosystem. Let $\mathcal{O}$ be an oracle which, given a ciphertext $C \equiv (M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^e \pmod{n^k}$, answers the second block of the plaintext M_1. The oracle $\mathcal{O}$ can be used to break the entire plaintext $(M_0, M_1, \ldots, M_{k-1})$.*

Proof: If we can decipher the first block M_0, then we can also do all the remaining blocks $M_2, \ldots, M_{k-1}$. Therefore, we can reduce the attack to the case of the two-block cryptosystem with modulo n^2. Let the plaintext $M = M_0 + nM_1\,(0 \leq M_0, M_1 < n)$, and $C \equiv M^e \pmod{n^2}$ be the ciphertext. For $i = 0, 1, 2, \ldots, h$, expand

$$2^i M \equiv M_0^{(i)} + nM_1^{(i)} \pmod{n^2}, \quad 0 < M_0^{(i)}, M_1^{(i)} < n,$$

where $h = \lfloor \log_2 n \rfloor$. Here, $2^{ie}C \equiv (M_0^{(i)} + nM_1^{(i)})^e \pmod{n^2}$ holds, and we can get each second block $M_1^{(i)} = \mathcal{O}(2^{ie}C)$ by using the oracle $\mathcal{O}$. Here, note that $M_0^{(i)} < n/2$ if and only if $2M_1^{(i)} \pmod{n} = M_1^{(i+1)}$ for $i = 0, 1, 2, \ldots, h$. Hence $2M_1^{(i)} \pmod{n} = \mathcal{O}(2^{(i+1)e}C)$ if and only if $M_0^{(i)} < n/2$ for $i = 0, 1, 2, \ldots, h$. On the other hand, let $C_0 \equiv C \pmod{n}$, and we have $2^{ie}C_0 \equiv (2^i M_0)^e \equiv (M_0^{(i)})^e \pmod{n}$ for $i = 0, 1, 2, \ldots, h$. This observation means that we can construct the half bit oracle

$\mathcal{O}_H$, which computes $\mathcal{O}_H(2^{ie}C_0) = 0$ if $M_0^{(i)} < n/2$ and $\mathcal{O}_H(2^{ie}C_0) = 1$ if $M_0^{(i)} > n/2$. Indeed, define that

$$\mathcal{O}_H(2^{ie}C_0) = \begin{cases} 0, & (2\mathcal{O}(2^{ie}C) \pmod n) = \mathcal{O}(2^{(i+1)e}C), \\ 1, & (2\mathcal{O}(2^{ie}C) \pmod n) \neq \mathcal{O}(2^{(i+1)e}C), \end{cases}$$

for $i = 0, 1, 2, \ldots, h$. It is well-known this half bit oracle $\mathcal{O}_H$ recovers the plaintext M_0 such that $C_0 \equiv M_0^e \pmod n$[GMT82]. Consequently, we can decipher the first block M_0. ■

4.5 Conclusion

Our proposed n-adic extensions of the RSA and Rabin cryptosystems perform decryption faster than any other multi-block RSA-type or Rabin-type cryptosystems ever reported. Deciphering the entire plaintext of this system is as intractable as breaking the original RSA cryptosystem or factoring. We also showed that the proposed n-adic RSA-type cryptosystem is a permutation function, and showed the criteria for message concealing and cycling attacks which are applicable to the RSA cryptosystem. Even if a message is several times longer than a public-key n, we can encrypt it fast without repeatedly using the secret-key cryptosystem.

Bibliography

[AM94] L. M. Adleman and K. S. McCurley, "Open problems in number theoretic complexity, II" Proceedings of ANTS-I, LNCS 877, (1994), pp.291-322.

[AD97] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," 29th ACM Symposium on Theory of Computing, (1997), pp.284-293.

[ACGS88] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr, "RSA and Rabin functions: certain parts are as hard as the whole," SIAM Journal of Computing, 17, (1988), pp.194-209.

[BB97] I. Biehl and J. Buchmann, "An analysis of the reduction algorithms for binary quadratic forms," Technical Report No. TI-26/97, Technische Universität Darmstadt, (1997).

[BPT99] I. Biehl, S. Paulus, and T. Takagi, "Efficient undeniable signature schemes based on ideal arithmetic in quadratic orders," Proceedings of Conference on the Mathematics of Public-Key Cryptography, The Fields Institute for Research in the Mathematical Sciences, (1999), pp. 1-17.

[BB79] G. R. Blakley and I. Borosh, "Rivest-Shamir-Adelman public key cryptosystems do not always conceal messages," Comput. & Maths. with Appls., 5, (1979), pp.169-178.

[BDH-G99] D. Boneh, G. Durfee, and N. Howgrave-Graham, "Factoring $N = p^r q$ for large r," Advances in Cryptology – CRYPTO'99, LNCS 1666, (1999), pp. 326-337.

[BD00] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," IEEE Transactions on Information Theory, Vol 46, No. 4, (2000), pp. 1339-1349.

[Bu99] J. Buchmann, *Einführung in die Kryptographie*, Springer, (1999).

[BW88] J. Buchmann and H. C. Williams, "A key-exchange system based on imaginary quadratic fields," Journal of Cryptology, 1, (1988), pp.107-118.

[BW90] J. Buchmann and H. C. Williams, "Quadratic fields and cryptography," London Math. Soc. Lecture Note Series 154, (1990), pp.9-26.

[BD90] J. Buchmann and S. Düllmann, "On the computation of discrete logarithms in class groups," Advances in Cryptology – CRYPTO'90, LNCS 537, (1991), pp.134-139.

[CR88] B. Chor and R.L. Rivest, "A knapsack-type public key cryptosystem based on arithmetic in finite fields," IEEE Transactions on Information Theory, 34 (1988), pp. 901-909.

[Coh93] H. Cohen, *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics 138, Springer, (1993).

[CFPR96] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with related messages," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.1-9.

[Cop96] D. Coppersmith, "Finding a small root of a univariate modular equation," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.155–165.

[CDELMZ96] J. Cowie, B. Dodson, R. Elkenbracht-Huizing, A. K. Lenstra, P. L. Montgomery, and J. Zayer, "A world wide number field sieve factoring record: on to 512 bits," Advances in Cryptology – ASIACRYPT '96, LNCS 1163, (1996), pp.382-394.

[Cox89] D. A. Cox, *Primes of the form* $x^2 + ny^2$, John Wiley & Sons, New York, (1989).

[De50] N. DeBruijn, "On the number of uncanceled elements in the sieve of Eratosthenes," Proc. Neder. Akad. Wetensch, vol. 53, (1950), pp.803-812.

[Dem94] N. Demytko, "A new elliptic curve based analogue of RSA," Advances in Cryptology – EUROCRYPT '93, LNCS 765, (1994), pp.40-49.

[DH76] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, 22, (1976), pp.472-492.

[DSA94] National Institute of Standards and Technology (NIST), Digital signature standard (DSS), Federal Information Processing Standards Publication, 186, (1994).

[ECM98] ECMNET Project, `http://www.loria.fr/~zimmerma/records/ecmnet.html`

[ElG85] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm in $GF(p)$," IEEE Transactions on Information Theory, 31, (1985), pp.469-472.

[Gar59] H. Garner, "The residue number system," IRE Transactions on Electronic Computers, EC-8 (6), (1959), pp. 140-147.

[GGOQ98] H. Gilbert, D. Gupta, A. M. Odlyzko, and J.-J. Quisquater, "Attacks on Shamir's 'RSA for paranoids', " preprint, (1998), `http://www.research.att.com/~amo/doc/recent.html`

[GMT82] S. Goldwasser, S. Micali, and P. Tong, "Why and how to establish a private code on a public network," Proc. of FOCS, (1982), pp.134-144.

[Ha88] J. Håstad, "Solving simultaneous modular equations of low degree," SIAM Journal of Computing, 17, (1988), pp.336-341.

[HM89] J. L. Hafner and K. S. McCurley, "A rigorous subexponential algorithm for computation of class groups, " J. Amer. Math. Soc., 2, (1989), pp.837-850.

[Hu99] D. Hühnlein, "Efficient implementation of cryptosystems based on non-maximal imaginary quadratic orders," Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, LNCS 1758, (1999), pp.294-307.

[HJPT98] D. Hühnlein, M. J. Jacobson, Jr., S. Paulus, and T. Takagi, "A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption," Advances in Cryptology – EUROCRYPT '98, LNCS 1403, (1998), pp.294-307.

[HT99] D. Hühnlein and T. Takagi, "Reducing logarithms in totally non-maximal imaginary quadratic orders to logarithms in finite fields," Advances in Cryptology – ASIACRYPT '99.

[HM00] S. Hamdy and B. Moeller, "Security of cryptosystems based on class groups of imaginary quadratic orders," to appear in ASIACRYPT '2000.

[HPT99] M. Hartmann, S. Paulus, and T. Takagi, "NICE - New Ideal Coset Encryption -, " Conference of Hardware Embedding System (CHES), LNCS 1717, (1999).

[Has88] J. Håstad, "Solving simultaneous modular equations of low degree," SIAM Journal of Computing, Vol.17, No.2, (1988), pp.336–341.

[Jac99] M. J. Jacobson, Jr. "Subexponential class group computation in quadratic orders," PhD Theses in Technical University of Darmstadt, (1999)

[JJ99] E. Jaulmes and A. Joux, "A NICE cryptanalysis," preprint, (1999).

[JQT00] M. Joye, J.-J. Quisquater, and T. Takagi, "How to choose secret parameters for RSA and its extensions to elliptic curves," to appear in Designs, Codes and Cryptography, (2000).

[Kal97] B. S. Kaliski Jr., "A chosen message attack on Demytko's elliptic curve cryptosystem," Journal of Cryptology, 10, (1997), pp.71-72.

[KR95] B. S. Kaliski Jr. and M. Robshaw, "Secure use of RSA," CRYPTOBYTES, 1 (3), (1995), pp.7-13.

[KS95] E. Kaltofen and V. Shoup, "Subquadratic-time factoring of polynomials over finite fields", Proc. of STOC, (1995), pp.398-406.

[Knu81] D. E. Knuth, *The art of computer programming*, vol.3: Sorting and searching, Addison-Wesley, 1981.

[Kob87] N. Koblitz, "Elliptic curve cryptosystems," Math. of Comp., 48(177), (1987), pp.203-209.

[Kob89] N. Koblitz, "Hyperelliptic cryptosystems," Journal of Cryptology, 1, (1989), pp.139-150.

[Koy95] K. Koyama, "Fast RSA-type schemes based on singular cubic curves," Advances in Cryptology – EUROCRYPT '95, LNCS 921, (1995), pp.329-340.

[KMOV92] K. Koyama, U. M. Maurer, T. Okamoto, and S. A. Vanstone, "New public-key schemes based on elliptic curves over the ring $\mathbb{Z}_n$," Advances in Cryptology – CRYPTO '91, LNCS 576, (1992), pp.252-266.

[LiD95] I. Biehl, J. Buchmann, and T. Papanikolaou, *LiDIA - A library for computational number theory.* The LiDIA Group, Universität des Saarlandes, Saarbrücken, Germany, 1995.

[LKLY00] S. Lim, S. Kim, H. Lee, and I. Yie, "A Generalised Takagi-Cryptosystem with a module of form $p^r q^s$," to appear in Indocrypt 2000.

[Len87] H. W. Lenstra, Jr., "Factoring integers with elliptic curves," Annals of Mathematics, 126, (1987), pp.649-673.

[LL91] A. K. Lenstra and H. W. Lenstra, Jr. (Eds.), *The development of the number field sieve.* Lecture Notes in Mathematics, 1554, Springer, (1991).

[LKBS92] J. H. Loxton, D. S. P. Khoo, G. J. Bird and J. Seberry, "A cubic RSA code equivalent to factorization," Journal of Cryptology, 5, (1992), pp.139-150.

[Mau95] U. M. Maurer, "Fast generation of prime numbers and secure public-key cryptographic parameters," Journal of Cryptology, 8, (1995), pp.123-155.

[McC98a] K. McCurley, "Cryptographic key distribution and computation in class groups," In: R.A. Mollin (ed.), *Number Theory and Applications*, Kluwer Academic Publishers, (1989), pp. 459-479.

[Mc78] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," DSN progress report 42-44, Jet Propulsion Laboratory, (1978).

[MvOV97] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*, CRC Press, 1997.

[MH78] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsack," IEEE Transaction on Information Theory, 24, (1978), pp.525-530.

[MM96] B. Meyer and V. Müller, "A public key cryptosystem based on elliptic curves over $\mathbf{Z}/n\mathbf{Z}$ equivalent to factoring," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.49-59.

[Mil86] V. Miller, "Use of elliptic curves in cryptography," Advances in Cryptology – CRYPTO '85, LNCS 218, (1986), pp.417-426.

[N00] P. Nguyen, "Lattice Reduction in Cryptology: An Update," to appear in ANTS IV, (2000).

[Oka90] T. Okamoto, "A fast signature scheme based on congruential polynomial operations," IEEE Transactions on Information Theory, IT-36, (1990), pp.47-53.

[OU98] T. Okamoto and S. Uchiyama, "A new public key cryptosystem as secure as factoring," Advances in Cryptology – EUROCRYPT '98, LNCS 1403, (1998), pp.308-318.

[Pa99] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," Advances in Cryptology – EUROCRYPT '99, LNCS 1592, (1999), pp.223-238.

[PP99] P. Paillier and Pointcheval, "Efficient public-key cryptosystems provably secure against active adversaries," Advances in Cryptology – ASIACRYPT '99, LNCS 1716, (1999), pp.165-179.

[PT00] S. Paulus and T. Takagi, "A new public-key cryptosystem over quadratic orders with quadratic decryption time", Journal of Cryptology, 13, (2000), pp.263-272.

[PT99] S. Paulus and T. Takagi, "A generalization of the Diffie-Hellman problem and related cryptosystems allowing fast decryption", 1998 International Conference on Information Security, ICISC'98, (1998), pp.211-220.

[PO96] R. Peralta and E. Okamoto, "Faster factoring of integers of a special form," IEICE Trans. Fundamentals, Vol.E79-A, No.4, (1996), pp.489-493.

[Po99] D. Pointcheval, "New public key cryptosystems based on the dependent-RSA problem," Advances in Cryptology – EUROCRYPT '99, LNCS 1592, (1999), pp.239-254.

[Pol78] J. M. Pollard, "Monte Carlo methods for index computation (mod p)," Math. Comp. Vol. 32, (1978), pp.918-924.

[PKCS] PKCS, Public-Key Cryptography Standards, RSA Laboratories
`http://www.rsalabs.com/pkcs/`

[QC82] J. -J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public-key cryptosystem," Electronic Letters, 18, (1982), pp.905-907.

[Ra79] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization", Technical Report No.212, MIT, Laboratory of Computer Science, Cambridge (1979), pp.1-16.

[Rie94] H. Riesel. *Prime Numbers and Computer Methods for Factorization.* Birkhäuser, 2nd ed., 1994.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21, (1978), pp.120-126.

[RS97] R. L. Rivest and R. D. Silverman, "Are 'strong' primes needed for RSA," The 1997 RSA Laboratories Seminar Series, Seminars Proceedings, (1997).

[RSAFAQ] RSA Laboratories, "Frequently asked questions about today's cryptography (Version 3.0)," `http://www.rsa.com/rsalabs/`, (1996).

[RSA155] RSA155, `http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa155.html`

[Sch83] R. J. Schoof, "Quadratic Fields and Factorization," Computational Methods in Number Theory, Math. Centrum Tracts 155, Part II, (1983), pp. 235-286.

[SE96] J. Schwenk and J. Eisfeld, "Public key encryption and signature schemes based on polynomials over $\mathbf{Z}_n$," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.60-71.

[Sha95] A. Shamir, "RSA for paranoids," CryptoBytes, v.1, n.3, (1995).

[Sha89] D. Shanks, "On Gauss and composition I, II," Proceedings of NATO ASI on Number Theory and Applications, Kluwer Academic Press, (1989), pp. 163-179.

[Si00] R. D. Silverman, "A cost-based security analysis of symmetric and asymmetric key lengths," RSA Laboratories, Bulletin #13, (2000).

[SN77] G.J. Simmons and M.J. Norris, "Preliminary comment on the M.I.T. public-key cryptosystem," Cryptologia, 1, (1977), pp.406–414.

[Tak97] T. Takagi, "Fast RSA-type cryptosystem using n-adic expansion," Advances in Cryptology – CRYPTO '97, LNCS 1294, (1997), pp.372–384.

[Tak98] T. Takagi, "Fast RSA-Type Cryptosystem Modulo $p^k q$," Advances in Cryptology - CRYPTO '98, LNCS 1462, (1998), pp.318-326.

[TN96] T. Takagi and S. Naito, "The multi-variable modular polynomial and its applications to cryptography," 7th International Symposium on Algorithm and Computation, ISAAC'96, LNCS 1178, (1996), pp.386-396.

[VT97] E. R. Verheul and H. C. A. van Tilborg, "Cryptanalysis of 'less short' RSA secret exponents," Applicable Algebra in Engineering, Communication and Computing, 8, (1997), pp.425-435.

[Vo00] U. Vollmer, "Asymptotically fast discrete logarithms in quadratic number fields," Algorithmic Number Theory, ANTS-IV, LNCS 1838, (2000), pp. 581-594.

[Wie90] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Transactions on Information Theory, IT-36, (1990), pp.553-558.

[WS79] H. C. Williams and B. Schmid, "Some remarks concerning the M.I.T. public-key cryptosystem," BIT 19, (1979), pp.525-538.

List of Tables

Frequently Used Notation

		page
$A \cap B$	intersection of A and B	
$A \subset B$	A is a subset of B	
$g \in G$	g is an element of G	
$\|G\|, \#G$	order of group G	
G/H	quotient group of G by subgroup H	
gH	coset of H	
$O(f)$	O-notation	
$o(f)$	o-notation	
log	logarithm base e	
$exp(x)$	e^x	
$L_n[s,c]$	$exp((c+o(1))(\log s)^s(\log\log n)^{1-s})$	
$\mathbb{Z}, \mathbf{Z}$	the integers	
$\mathbb{Q}, \mathbf{Q}$	the rationals	
$\mathbb{R}, \mathbf{R}$	the reals	
$\mathbb{Z}/m\mathbb{Z}, \mathbf{Z}_m$	the residue class modulo m: $\{0, 1, ..., m-1\}$	
$(\mathbb{Z}/m\mathbb{Z})^\times, \mathbf{Z}_m^\times$	the reduced residue class modulo m	
$a\|b$	a divides b	
$\gcd(a,b), \mathrm{GCD}(a,b)$	the great common divisor of a and b	
$\mathrm{lcm}(a,b), \mathrm{LCM}(a,b)$	the least common multiplier of a and b	
$a \equiv b \bmod m$	a is congruent b modulo m	
$\mathrm{ord}_m(z)$	the order z in $\mathbb{Z}/m\mathbb{Z}$	
$\lfloor x \rfloor$	greatest integer $\leq x \in \mathbb{R}$	
$\|x\|$	absolute value of $x \in \mathbb{R}$	
Δ_1	fundamental discriminant	9
f	conductor	9
Δ	non-fundamental discriminant $\Delta_f = \Delta_1 f^2$	9
$\mathcal{O}_\Delta$	quadratic order of discriminant Δ	9
γ'	complex conjugate of $\gamma \in \mathcal{O}_\Delta$	9
$\mathfrak{a}$	ideal in $\mathcal{O}_\Delta$	9

Biography and Publications

Biographical sketch

05.12.1969	Born in Nagoya, Japan
04.1985 - 03.1987	Showa High School, Nagoya, Japan
04.1987 - 03.1993	Nagoya University, Nagoya, Japan
	Degree of Bachelor of Science in Mathematics - B.Sc.
04.1993 - 03.1995	Graduate School of Science, Nagoya University, Nagoya, Japan
	Degree of Master of Science in Mathematics - M.Sc.
04.1995 - 09.1997	NTT Laboratories, Tokyo, Japan
	Nippon Telegraph and Telephone Corporation
10.1997 - 09.1998	Visiting researcher
	Darmstadt University of Technology
10.1998 -	NTT Laboratories, Düsseldorf, Germany
	Nippon Telegraph and Telephone Corporation

03.1997	IEICE Young Engineer Award
09.1998	NTT Laboratories President Award

Publications

International Journals

1. How to Choose Secret Parameters for RSA and its Extensions to Elliptic Curves, to appear in Designs, Codes and Cryptography, (2000), (with Marc Joye, Jean-Jacques Quisquater).

2. A New Public-key Cryptosystem over the Quadratic Order with Quadratic Decryption Time, Journal of Cryptology, 13, (2000), pp.263-272, (with Sachar Paulus).

3. Product Formula of the Cubic Gauss Sum Modulo the Product of the Primes, Journal of Number Theory, Vol.62, No.2, (1997), pp.298-306.

Japanese Transactions

4. Construction of RSA Cryptosystem over the Algebraic Field Using Ideal Theory and Investigation of its Security, IEICE Transactions, Vol.J81-A, No.1, (1998), pp.119-128. (with Shozo Naito). (In Japanese)
5. Low Exponent Attacks against the Schwenk-Eisfeld Cryptoscheme and Signature, IEICE Transactions, Vol.E81-A, No.3, (1998), pp.483-488, (with Shozo Naito).
6. Extension of Rabin Cryptosystem to Eisenstein and Gauss Fields, IEICE Transactions, Vol.E80-A, No.4, (1997), pp.753-760, (with Shozo Naito).

International Conferences by IACR

7. Fast RSA-Type Cryptosystem Modulo $p^k q$, Advances in Cryptology - CRYPTO '98, LNCS 1462, Springer, (1998), pp.318-326.
8. A Cryptosystem Based on Non-Maximal Imaginary Quadratic Orders with Fast Decryption, Advances in Cryptology - EUROCRYPT '98, LNCS 1403, Springer, (1998), pp.294-307, (with Detlef Huehnlein, Michael J. Jacobson, Sachar Paulus).
9. Fast RSA-Type Cryptosystems Using n-Adic Expansion, Advances in Cryptology - CRYPTO '97, LNCS 1294, Springer, (1997), pp.372-384.

International Conferences published as LNCS

10. NICE - New Ideal Coset Encryption -, Workshop on Cryptographic Hardware and Embedded Systems (CHES), LNCS 1717, Springer (1999), pp. 328-339, (with Michael Hartmann, Sachar Paulus).
11. Reducing Logarithms in totally Non-Maximal Imaginary Quadratic Orders to Logarithms in Finite Fields, Advances in Cryptology - AISACRYPT '99, LNCS 1716, Springer, (1999), pp.219-231, (with Detlef Huehnlein).
12. The Multi-Variable Modular Polynomial and its Applications to Cryptography, 7th International Symposium on Algorithm and Computation, ISAAC'96, LNCS 1178, Springer, (1996), pp.386-396. (with Shozo Naito).

Other International Conferences

13. Efficient Undeniable Signature Schemes Based on Ideal Arithmetic in Quadratic Orders, Proceedings of Conference on the Mathematics of Public-Key Cryptography, The Fields Institute for Research in the Mathematical Sciences, (1999), pp. 1-17, (with Ingrid Biehl, Sachar Paulus).
14. Rabin and RSA Analogues Based on Non-maximal Imaginary Quadratic Orders, 1998 International Conference on Information Security, ICISC'98, (1998), pp.221-240. (with Detlef Huehnlein, Andreas Meyer).
15. A Generalization of the Diffie-Hellman Problem and Related Cryptosystems Allowing Fast Decryption, 1998 International Conference on Information Security, ICISC'98, (1998), pp.211-220. (Sachar Paulus).

Academic Japanese Magazines

16. EC Security in Germany, Japan Electronic Industry Development Association, February 2000, pp.22-29, (with Toru Kobayashi). (In Japanese).
17. Mathematical Background of Public-key Cryptosystem, R&D Material No.1375, NTT Laboratories, April 1997, pp.1-24, (with Shozo Naito).
18. Public-key cryptosystem and number theory, Chubu Forum for Mathematical Sciences, Vol.1, December 1996, pp.19-27. (In Japanese)

Japanese Book

19. Public-key Cryptosystem (RSA and ElGamal), BIT, Special Volume, Information Security, Kyoritsu, January 2000, pp.222-227. (In Japanese)

Index