Applications of Algebraic Number Theory to Cryptography

by

Renate Scheidler

A Thesis
Presented to the University of Manitoba
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-315-86064-2

Canada

Name RENATE SCHEIDLER

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

APPLIED SCIENCES – COMPUTER SCIENCE

SUBJECT TERM

| 0 | 9 | 8 | 4 |

SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture	0729
Art History	0377
Cinema	0900
Dance	0378
Fine Arts	0357
Information Science	0723
Journalism	0391
Library Science	0399
Mass Communications	0708
Music	0413
Speech Communication	0459
Theater	0465

EDUCATION

General	0515
Administration	0514
Adult and Continuing	0516
Agricultural	0517
Art	0273
Bilingual and Multicultural	0282
Business	0688
Community College	0275
Curriculum and Instruction	0727
Early Childhood	0518
Elementary	0524
Finance	0277
Guidance and Counseling	0519
Health	0680
Higher	0745
History of	0520
Home Economics	0278
Industrial	0521
Language and Literature	0279
Mathematics	0280
Music	0522
Philosophy of	0998
Physical	0523
Psychology	0525
Reading	0535
Religious	0527
Sciences	0714
Secondary	0533
Social Sciences	0534
Sociology of	0340
Special	0529
Teacher Training	0530
Technology	0710
Tests and Measurements	0288
Vocational	0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language	
General	0679
Ancient	0289
Linguistics	0290
Modern	0291
Literature	
General	0401
Classical	0294
Comparative	0295
Medieval	0297
Modern	0298
African	0316
American	0591
Asian	0305
Canadian (English)	0352
Canadian (French)	0355
English	0593
Germanic	0311
Latin American	0312
Middle Eastern	0315
Romance	0313
Slavic and East European	0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy	0422
Religion	
General	0318
Biblical Studies	0321
Clergy	0319
History of	0320
Philosophy of	0322
Theology	0469

SOCIAL SCIENCES

American Studies	0323
Anthropology	
Archaeology	0324
Cultural	0326
Physical	0327
Business Administration	
General	0310
Accounting	0272
Banking	0770
Management	0454
Marketing	0338
Canadian Studies	0385
Economics	
General	0501
Agricultural	0503
Commerce-Business	0505
Finance	0508
History	0509
Labor	0510
Theory	0511
Folklore	0358
Geography	0366
Gerontology	0351
History	
General	0578
Ancient	0579
Medieval	0581
Modern	0582
Black	0328
African	0331
Asia, Australia and Oceania	0332
Canadian	0334
European	0335
Latin American	0336
Middle Eastern	0333
United States	0337
History of Science	0585
Law	0398
Political Science	
General	0615
International Law and Relations	0616
Public Administration	0617
Recreation	0814
Social Work	0452
Sociology	
General	0626
Criminology and Penology	0627
Demography	0938
Ethnic and Racial Studies	0631
Individual and Family Studies	0628
Industrial and Labor Relations	0629
Public and Social Welfare	0630
Social Structure and Development	0700
Theory and Methods	0344
Transportation	0709
Urban and Regional Planning	0999
Women's Studies	0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture	
General	0473
Agronomy	0285
Animal Culture and Nutrition	0475
Animal Pathology	0476
Food Science and Technology	0359
Forestry and Wildlife	0478
Plant Culture	0479
Plant Pathology	0480
Plant Physiology	0817
Range Management	0777
Wood Technology	0746
Biology	
General	0306
Anatomy	0287
Biostatistics	0308
Botany	0309
Cell	0379
Ecology	0329
Entomology	0353
Genetics	0369
Limnology	0793
Microbiology	0410
Molecular	0307
Neuroscience	0317
Oceanography	0416
Physiology	0433
Radiation	0821
Veterinary Science	0778
Zoology	0472
Biophysics	
General	0786
Medical	0760

EARTH SCIENCES

Biogeochemistry	0425
Geochemistry	0996
Geodesy	0370
Geology	0372
Geophysics	0373
Hydrology	0388
Mineralogy	0411
Paleobotany	0345
Paleoecology	0426
Paleontology	0418
Paleozoology	0985
Palynology	0427
Physical Geography	0368
Physical Oceanography	0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences	0768
Health Sciences	
General	0566
Audiology	0300
Chemotherapy	0992
Dentistry	0567
Education	0350
Hospital Management	0769
Human Development	0758
Immunology	0982
Medicine and Surgery	0564
Mental Health	0347
Nursing	0569
Nutrition	0570
Obstetrics and Gynecology	0380
Occupational Health and Therapy	0354
Ophthalmology	0381
Pathology	0571
Pharmacology	0419
Pharmacy	0572
Physical Therapy	0382
Public Health	0573
Radiology	0574
Recreation	0575
Speech Pathology	0460
Toxicology	0383
Home Economics	0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry	
General	0485
Agricultural	0749
Analytical	0486
Biochemistry	0487
Inorganic	0488
Nuclear	0738
Organic	0490
Pharmaceutical	0491
Physical	0494
Polymer	0495
Radiation	0754
Mathematics	0405
Physics	
General	0605
Acoustics	0986
Astronomy and Astrophysics	0606
Atmospheric Science	0608
Atomic	0748
Electronics and Electricity	0607
Elementary Particles and High Energy	0798
Fluid and Plasma	0759
Molecular	0609
Nuclear	0610
Optics	0752
Radiation	0756
Solid State	0611
Statistics	0463

Applied Sciences

Applied Mechanics	0346
Computer Science	0984
Engineering	
General	0537
Aerospace	0538
Agricultural	0539
Automotive	0540
Biomedical	0541
Chemical	0542
Civil	0543
Electronics and Electrical	0544
Heat and Thermodynamics	0348
Hydraulic	0545
Industrial	0546
Marine	0547
Materials Science	0794
Mechanical	0548
Metallurgy	0743
Mining	0551
Nuclear	0552
Packaging	0549
Petroleum	0765
Sanitary and Municipal	0554
System Science	0790
Geotechnology	0428
Operations Research	0796
Plastics Technology	0795
Textile Technology	0994

PSYCHOLOGY

General	0621
Behavioral	0384
Clinical	0622
Developmental	0620
Experimental	0623
Industrial	0624
Personality	0625
Physiological	0989
Psychobiology	0349
Psychometrics	0632
Social	0451

APPLICATIONS OF ALGEBRAIC NUMBER THEORY TO CRYPTOGRAPHY

BY

RENATE SCHEIDLER

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Renate Scheidler

Abstract

If two communication partners wish to engage in a private conversation across a public channel, they need to encrypt their messages to prevent an eavesdropper from discovering the contents of their conversation. To achieve this, the two parties must first agree on a common cryptographic key. Such a key cannot be distributed across an open channel, as this would enable an adversary to obtain the key and thus decrypt all communicated information. The problem of key exchange can be overcome in two ways. The partners can employ a public-key cryptosystem, i.e. use different keys for encryption and decryption, where the encryption key is publicly known and the decryption key is known only to the decrypter. Alternatively, they can communicate a sequence of messages according to a specific protocol that allow them to agree on a common key without revealing it to an opponent. This dissertation offers solutions to both approaches.

The first part of this thesis presents a generalization of several existing public-key cryptosystems. The difficulty of breaking the new scheme is equivalent to the problem of factoring a large integer, a task believed to be very difficult. This information regarding the security of the scheme represents an improvement over the well-known RSA public-key system. We describe the number theoretic fundamentals, present the algorithms required for the system together with their computational complexity, analyze the scheme's security, and finally discuss an implementation.

All conventional protocols for key exchange rely strongly on the structure of a group. Recently, for the first time, a modification of the standard protocol was suggested which does not use a group, but is based instead on the infrastructure of a real quadratic field. This loss of structure in the underlying set may increase the security of the scheme over that of previously known protocols. Part II of this thesis introduces the specifics of the new protocol. As before, we give the necessary number theoretic background, describe the algorithms and their complexity, present a complete approximation and error analysis, briefly discuss the security, and conclude with some computational results.

Acknowledgements

The author was assisted by many individuals during the development of this dissertation. Although there is only room to mention a few by name, I am grateful for the help of all those who were involved.

First and foremost, I thank my advisor, Dr. Hugh C. Williams. He first introduced me to the fascinating topic of cryptography, and has taught me a great deal over the years. He continually challenged and motivated me during his supervision of this thesis. He was extremely sensitive not only to academic difficulties I encountered, but also to occasional times of self-doubt that I experienced. He provided generous financial support, including assistance in attending several international conferences. Hugh's time, patience, and commitment are greatly appreciated.

I would like to thank the members of my examining committee, Professor Johannes A. Buchmann, Dr. Jacek Fabrykowski, and Dr. Ralph G. Stanton, for their time and efforts in reading and commenting on this thesis.

Special thanks go to Dr. Hendrik W. Lenstra, Jr., for his assistance in the development of the Euclidean division algorithm given in Section 6.4.4.

Finally, I wish to acknowledge those many individuals who supported me throughout the years of work by lending a sympathetic ear when I needed it, offering words of encouragement, and simply showing their friendship. In particular, I thank Jim Vincent for patiently putting up with my frequent mood swings and for his never-ending support and faith in me.

Table of Contents

Frequently Used Notation

The following sets and structures are frequently used.

Symbol	*Description*
$\mathbf{Z}$	Ring of rational integers
$\mathbf{Q}$	Field of rationals
$\mathbf{R}$	Field of real numbers
$\mathbf{C}$	Field of complex numbers
$\mathbf{K}, \mathbf{L}$	Algebraic number fields
$\mathrm{GF}(q)$	Finite field of q elements
$\mathbf{F}^*$	Multiplicative group $\mathbf{F}$-$\{0\}$ of any field $\mathbf{F}$
$\mathbf{S}[x]$	Ring of polynomials in x with coefficients in $\mathbf{S}$ ($\mathbf{S}$ any ring)
$\mathrm{GL}_n(\mathrm{S})$	Ring of non-singular $n \times n$ matrices over a ring $\mathbf{S}$
$\mathbf{G}$	Any group
$\lvert\mathbf{M}\rvert$	Cardinality of a set $\mathbf{M}$

For a given algebraic number field $\mathbf{K}$, we use the following notation.

Symbol	*Description*
$\mathbf{O}$	Maximal order
$\mathbf{I}$	Set of integral ideals
$\mathbf{P}$	Set of principal integral ideals
$\mathfrak{R}$	Set of reduced principal ideals
$\mathrm{Cl}(\mathrm{K})$	Class group
$\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{r}$	Ideals
$\mathfrak{p}, \mathfrak{q}$	Prime ideals
$n = (\mathrm{K}:\mathrm{Q})$	Degree of $\mathbf{K}$ over $\mathbf{Q}$
s	Number of real conjugate mappings
t	Number of complex conjugate mappings, $s + 2t = n$
r	Unit rank, $r = s + t - 1$
$\sigma_1, \dots, \sigma_n$	Conjugate mappings
	(for $n = s = 2$, α' denotes the conjugate of α)
$\eta_1, \dots, \eta_r$	Fundamental units (for $r = 1$, write $\eta_1 = \eta$)
Δ	Field discriminant
R	Regulator
h	Class number, $h = \lvert\mathrm{Cl}(\mathrm{K})\rvert$

Symbol	*Description*
ε	Any unit
π, ψ	Primes in $\mathbf{O}$
$N(\alpha)$	Norm of $\alpha \in \mathbf{K}$, $N(\alpha) = \prod_{i=1}^{n} \sigma_i(\alpha)$
$Tr(\alpha)$	Trace of $\alpha \in \mathbf{K}$, $Tr(\alpha) = \sum_{i=1}^{n} \sigma_i(\alpha)$
$\bar{\alpha}$	Complex conjugate $a - bi$ of $\alpha = a + bi \in \mathbf{C}$
$L(\mathbf{a})$	The least positive rational integer in an ideal $\mathbf{a}$, $L(\mathbf{a}) = \min\{\mathbf{a} \cap \mathbb{Z}^{>0}\}$

The following special symbols occur frequently:

Notation	*Description*
$\mathbf{a} \sim \mathbf{b}$	$\mathbf{a}$ and $\mathbf{b}$ are equivalent ideals
$\alpha \simeq \beta$	α and β are associates ($\alpha, \beta \in \mathbf{O}$)
$a \mid b$	a is a divisor of b (used for rational integers, integers, and integral ideals)
$a^n \parallel b$	a^n is the exact power of a dividing b
$\lfloor a \rfloor$	the floor of $a \in \mathbf{R}$
$\lceil a \rceil$	the ceiling of $a \in \mathbf{R}$
$Ne(a)$	the integer nearest to $a \in \mathbf{R}$, $Ne(a) = \lfloor a + \frac{1}{2} \rfloor$

The following abbreviations and acronyms are used.

Abbreviation/Acronym	*Meaning*
CCA	Chosen Ciphertext Attack
COA	Ciphertext Only Attack
CPA	Chosen Plaintext Attack
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
ERH	Extended Riemann Hypothesis
KPA	Known Plaintext Attack
PKC	Public-Key Cryptosystem
RSA	Rivest-Shamir-Adleman cryptosystem
UFD	Unique Factorization Domain

The symbol ❑ denotes the end of a proof or algorithm, or the end of a theorem or lemma whose proof is omitted.

1. Introduction

1.1 Introduction to Cryptology

The art of *cryptology* consists of the combined art of cryptography and cryptanalysis. *Cryptography* (Greek, κρυπτόζ: hidden, γράφειν: to write) provides secure communication over insecure channels while *cryptanalysis* provides the means of breaking into these communications. Historically, cryptology has been almost exclusively of interest to the military and diplomats. However, this changed drastically with the invention and widespread use of computers. Today, vast amounts of information are transmitted over telephone lines, distributed across computer networks, or stored in electronic data banks by individuals, private companies, government agencies, and academic institutions. Much of this information is intended for the eyes of certain designated recipients only, and its disclosure to unauthorized individuals may be harmful, sometimes even to the point of posing a threat to national security. The need for privacy in sharing sensitive data across public-access systems lead to civilian cryptology. Cryptography and cryptanalysis are engaged in an on-going race that is as old as the idea of "disguising" one's messages, which in turn may well go back to the idea of writing itself. For a detailed history of cryptology, see Kahn [Ka67].

1.1.1 Private-Key Cryptosystems

Most of the material presented in Section 1.1 can be found in any cryptography text, such as Denning [De83] or Brassard [Br88].

The classic scenario of a cryptographic session consists of two communication partners (traditionally called *Alice* and *Bob* in the world of non-classical cryptologic research), who wish to engage in a private conversation across a public channel, usually any kind of high speed data transmission line. We will only consider the case where Alice (the *sender*) transmits a piece of confidential information to Bob (the *receiver*); if Bob wants to reply, the same procedure is applied with reversed roles, i.e. Bob becomes the sender and Alice the receiver.

To send a message M, also called the *plaintext* or *cleartext*, Alice uses a *cipher* to *encrypt* or *encipher* the message, i.e. she applies a transformation that renders the plaintext unintelligible to anyone but the intended recipient. She thus obtains a *ciphertext* or *cryptogram* C, which she communicates over the insecure channel to Bob. Upon receiving C, Bob *decrypts* or *deciphers* it, i.e. he converts it back to its original form.[1] Both encryption and decryption are performed by means of a *key* K, which the two communication partners need to agree upon ahead of time and which must not be revealed to anyone else. Only individuals who know the key are able to encipher and decipher messages.

An unauthorized individual tapping the communication line in an attempt to recover plaintext from intercepted ciphertext is referred to variously as a *cryptanalyst*, *adversary*, or *opponent*. The cryptanalyst's ultimate goal is to retrieve a cleartext message or even the key, but he may be content with recovering partial information about either. If he is successful, the cipher is considered *broken*. An opponent may pose a passive threat by simply eavesdropping on a conversation, or an active threat by injecting information into a channel, thus altering the transmitted ciphertext. The latter problem can be overcome if the communication partners can *authenticate* their messages.

The scenario described in the previous two paragraphs is called a *one-key* or *private-key cryptosystem* (short for *cryptographic system*). Figure 1.1 below shows the scheme of a one-key cryptosystem.

[1] Some works on cryptography distinguish between *decipherment* as the recovery of the plaintext by the intended recipient, and *decryption* as the same process attempted by an unauthorized individual. We will not make this distinction.

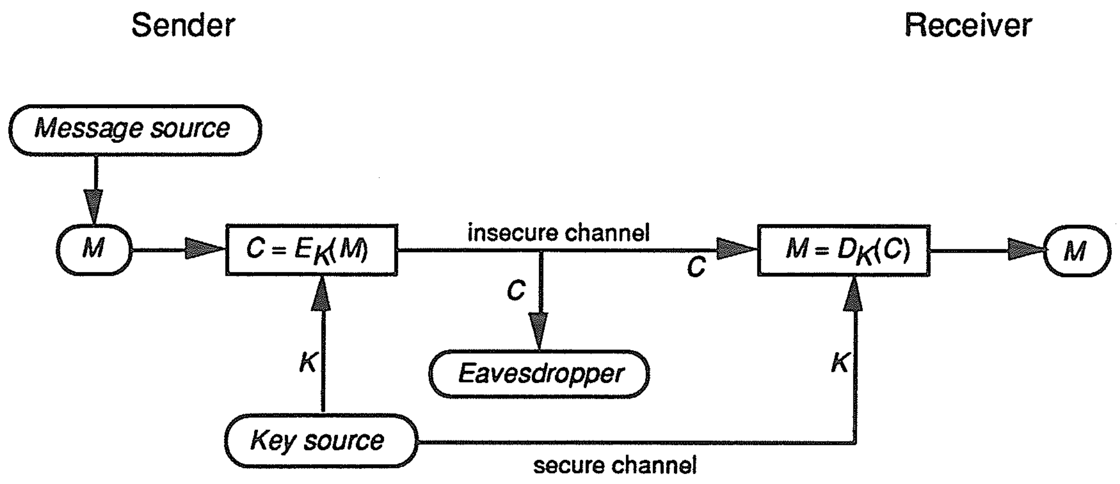

Figure 1.1

Formally, a private-key cryptosystem consists of a plaintext message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, a key space $\mathcal{K}$, and a family $(E_K)_{K\in\mathcal{K}}$ of encryption transformations with domain $\mathcal{M}$ and range $\mathcal{C}$, such that each encryption function E_K ($K \in \mathcal{K}$) has a left inverse D_K: $\mathcal{C} \to \mathcal{M}$, i.e. $D_K(E_K(M)) = M$ for each $M \in \mathcal{M}$. In addition, the system should be both practical and secure, i.e. it must satisfy the following requirements.

Practicality:

1. For all $K \in \mathcal{K}$, E_K and D_K should be easy to compute.
2. All keys should be small.

Security:

3. The secrecy of the system should depend entirely on the key K, not on the encryption/decryption methods E_K, D_K.
4. For any $K \in \mathcal{K}$, given the encryption algorithm E_K and a ciphertext $C = E_K(M)$, $M \in \mathcal{M}$, it should be infeasible to recover M or K

Clearly, these are not mathematically rigorous requirements. Usually, we will associate with any cryptosystem a security parameter $P \in \mathbb{Z}^{>0}$. Then requirement 1 states that the computation time for encryption and decryption is polynomial in log P, and condition 2 implies that the number of bits in the binary representation of any key be polynomial in log P. Requirement 4 can be formalized to mean that there is no polynomial-time algorithm (in log P) to deduce M or K from E_K and C. In particular, the key space $\mathcal{K}$ needs to be sufficiently large to prevent a cryptanalytic attack by *exhaustive key search.* From condition 3, we see that it is possible to publicize the enciphering and deciphering algorithms, i.e. their operations on M and K and C and K, respectively, without compromising the security of the scheme, as long as the key K is kept secret. In particular, the scheme must be secure against its own designer.

This somewhat more formal framework enables us to define cryptography as the application of transformations (ciphers) to information to effect the concealment (encryption) of that information. Cryptanalysis is the process by which an unauthorized interceptor of a cryptogram determines either its corresponding plaintext without prior knowledge of the key, or the key used in obtaining this ciphertext.

A very simple example of a private-key cryptosystem is the *Caesar cipher*, in which each letter is replaced by a letter which occurs a fixed number K of positions beyond it in the alphabet, with "wrap-around" if necessary.[2] More specifically, first the position P of a plaintext letter in the alphabet is determined, where A corresponds to 0, B to 1, etc., down to Z corresponding to 25. Then the position Z of the encrypted letter is obtained by computing $Z \equiv P + K \pmod{26}$, $0 \le Z \le 25$, where K is the number of positions skipped. Z uniquely identifies the ciphertext letter. As an example, consider the plaintext HAL, the computer's name in the well-known motion picture *Space Odyssey 2001*. If we choose $K = 1$, HAL encrypts to the familiar acronym IBM.

We distinguish between three different levels of cryptanalytic attack on a cryptosystem.

Ciphertext Only Attack (*COA*): The cryptanalyst is in possession of a number of distinct cryptograms enciphered under the same key and attempts to infer the key or, if this is impossible, find as many of the plaintexts corresponding to the cryptograms as possible.

[2] According to Suetonius [Su65, Julius 56, p. 42], this cipher was used by Julius Caesar with $K = 3$.

Known Plaintext Attack (*KPA*): The cryptanalyst is in possession of a number of distinct pairs of plaintext and corresponding ciphertexts, all encrypted under the same key. He attempts to infer the key or, if this is impossible, find the plaintext corresponding to some new ciphertext.

Chosen Plaintext Attack (*CPA*): The cryptanalyst chooses a number of distinct plaintexts and is given the corresponding ciphertexts, all encrypted under the same key. He attempts to infer the key or, if this is impossible, find the plaintext corresponding to some new ciphertext.

A COA is often successful if partial information about the plaintext is available to the cryptanalyst, such as redundancy in language (frequently occurring letters, sequences of letters, or words) which is reflected in the cryptogram. Our example of the Caesar cipher easily succumbs to such an attack. A KPA could be mounted if partial content of the plaintext is known. For example, an opponent could be intercepting a remote login message which he knows to contain the word LOGIN and the user's id. Encrypted programs are particularly vulnerable to this kind of attack because of a limited number of frequently occurring reserved words, such as BEGIN, END, IF, THEN, WHILE, etc. As an example for a CPA, assume that an adversary has partial read/write access to a system. He can implant changes into the system and observe the resulting changes in the stored ciphertext. This information may be used to mount an attack on the part of the system that he is not authorized to access. A CPA presents the most favourable circumstances to a cryptanalyst. Hence, a cryptosystem needs to be designed to be secure against this kind of attack.

There are two notions of security for a cryptosystem. A system is *unconditionally secure* if a cryptanalyst cannot gain enough information to break it, while it is *computationally secure* if he does not have sufficient resources (computing time and power). Unconditionally secure systems provide *perfect secrecy*, i.e. regardless of how much ciphertext is intercepted, it will never provide enough information to deduce the plaintext uniquely. Such systems are mathematically unbreakable in the sense that the probability that a cleartext M was sent (in encrypted form) is the same as the probability that M was transmitted, given that a cryptogram C was intercepted. An example of an unconditionally secure system is the *one-time pad*, in which any key is used only once and is bitwise XORed with the plaintext.

Unfortunately, such systems require the key space to be at least as large as the message space (Shannon [Sh49]). This means essentially that a key must be at least as long as the plaintext and that each key may be used no more than once. As a result, unconditionally secure systems are highly impractical. In fact, most available ciphers are theoretically breakable after intercepting only a few hundred bits of ciphertext, but the computational requirements needed to extract plaintext are beyond all available resources. This leads to the notion of computational security. A cryptosystem is computationally secure if it cannot be broken by systematic cryptanalysis with available resources under favourable conditions for an opponent (CPA).

Most available one-key ciphers use substitutions, transpositions, or both in their encryption algorithms. A *substitution cipher* is one in which each letter of plaintext is replaced by a character of ciphertext taken from one (or possibly several different) ciphertext alphabet(s). The Caesar cipher discussed earlier represents an example of such a cipher. In a *transposition cipher*, the plaintext characters are rearranged (permuted) according to a specific algorithm and key. For example, the cleartext could be arranged in a matrix of K columns (where K is the key), and the rows of the matrix represent the cryptogram.

Unfortunately, ciphers based on either substitution or transposition reflect redundancies in the language and are thus vulnerable to statistical attacks using frequency counts of letters, digrams, trigrams (consecutive pairs and triples of letters, respectively), and commonly occurring words. However, combining both substitution and transposition in a single cipher will foil such attacks [Sh49]. The well-known *Data Encryption Standard*

(DES) [NBS77] is based on this idea. DES was designed by IBM in the early 1970's and is widely used in commercial applications. So far, it remains unbroken.

The main problem arising from one-key cryptosystems is that of key distribution. How can a key generated by the encrypter (or a third party) be safely communicated to the decrypter (or all participants)? One possibility is to send the key along a secure channel. However, such channels often tend to be slow and hard to come by (after all, if the communication partners had a fast and secure channel available to them, they need not employ a cryptosystem!). Examples of a secure channel are a trusted courier or a personal meeting of the participants. Clearly, this approach is too cumbersome and inefficient, particularly if two individuals in geographically distant sites need to communicate frequently and wish to use a new key for each session for reasons of security. In the 1970's, Diffie and Hellman proposed two different solutions to the problem of secure key distribution. The first approach suggests using different keys for encryption and decryption, thus eliminating the need for distributing the same key to different users. This lead to the discovery of *public-key cryptosystems* [DH76] which will be discussed in the next section. The other solution proposed the exchange of partial information about the key across a public channel in a way that both partners can agree on a common key while an eavesdropper cannot gain any knowledge about this key from the intercepted information. This seemingly impossible goal was first achieved by the *Diffie-Hellman key exchange protocol* [DH76] and a different approach was independently suggested by Merkle [Me78]. Both these schemes will be briefly discussed in Section 1.1.4.

1.1.2 Public-Key Cryptosystems

The main difference between a public-key cryptosystem (PKC) and a private-key scheme is that the sender and receiver use different keys, where the enciphering key is publicly known, while the deciphering key is known only to the decrypter. More specifically, every user generates a pair of keys K_S, K_P, where K_S, the decryption key, is kept secret, and K_p, the encryption key, is publicized.[3] If Alice wishes to send a private message to Bob, she obtains his public key, encrypts the plaintext under this key, and transmits the ciphertext to Bob. Upon receiving the cryptogram, Bob uses his secret key to decipher it and recover Alice's original message. An eavesdropper tapping the line can intercept the ciphertext and obtain Bob's public key, but cannot recover the corresponding plaintext without knowledge of Bob's secret key. Furthermore, he should be unable to infer any information about Bob's secret key from the cryptogram and Bob's public key. The scheme of a PKC is given in Figure 1.2.

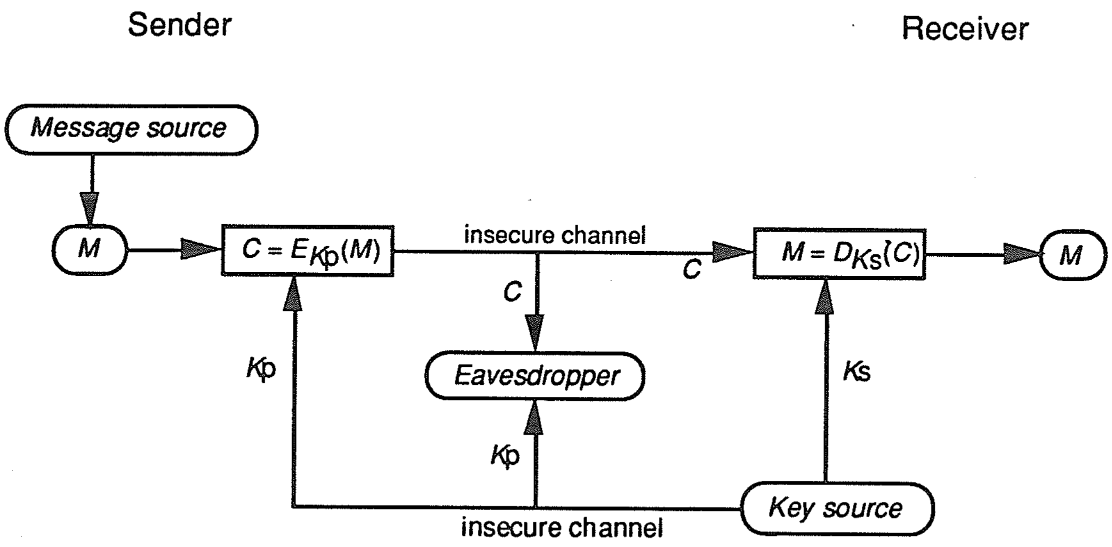

Figure 1.2

Formally, a PKC consists of a plaintext message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, a public-key space $\mathcal{K}_p$, a secret-key space $\mathcal{K}_s$, and pairs of functions $E_{K_p}: \mathcal{M} \rightarrow \mathcal{C}$ and $D_{K_s}: \mathcal{C} \rightarrow \mathcal{M}$, where $K_p \in \mathcal{K}_p$, $K_s \in \mathcal{K}_s$, such that D_{K_s} is the left-inverse of E_{K_p}, i.e. $D_{K_s}(E_{K_p}(M)) = M$ for all $M \in \mathcal{M}$. Again, the system is required to be both practical and secure and must satisfy the following conditions.

[3] The encryption key could be published, together with the user's name, on an electronic bulletin board or a key directory, the analogue of a phone book.

Practicality:

1. For all $K_p \in \mathcal{K}_p$, $K_s \in \mathcal{K}_s$, E_{K_p} and D_{K_s} should be easy to compute.
2. All public keys should be small.

Security:

3. Given a public key K_p, the encryption algorithm E_{K_p}, and a ciphertext $C = E_{K_p}(M)$, it should be infeasible to recover M or the secret key K_s.

Here the terms "easy to compute", "small ", and "infeasible" can again be formalized with respect to a security parameter P.

Note that a cryptanalyst can easily create the circumstances of a KPA or a CPA, since the recipient's encryption key and algorithm are public. Hence, we define a new level of attack for PKC's:

Chosen Ciphertext attack (CCA): The cryptanalyst chooses a number of distinct ciphertexts, all generated through encryption under the same public key, and obtains the corresponding plaintexts. He attempts to infer the secret key or produce an efficient algorithm for simulating decryption under the secret key or, if both these tasks are impossible, find the plaintext corresponding to some new ciphertext.

At the heart of public-key cryptography lies the notion of a one-way and a trapdoor one-way function. Informally, a *one-way function* is a function which is easy to compute and hard to invert, and a *trapdoor one-way function* is a function which is easy to compute and hard to invert unless some inside information is available, in which case the function is also easy to invert. Somewhat more formally, a one-way function is a function f: $X \rightarrow Y$ satisfying the following two conditions.

1. For all $x \in X, f(x)$ is hard to compute.
2. For almost all $y \in Y$, if there exists $x \in X$ such that $f(x) = y$, then it is hard to compute x.

An example for the use of a one-way function in cryptography is the safe storage of passwords. Instead of storing the passwords $P_1, P_2, \ldots$ of all users in the clear, we store

$f(P_1), f(P_2), \ldots$, where f is a one-way function. To log on, a user id i and a password P_i are entered. The system computes $f(P_i)$, looks up the user id i in the password file, and compares $f(P_i)$ with the password entry corresponding to id i. If the two match, the user may access the system, otherwise access is denied.

It is unknown whether one-way functions exist; in fact a proof of their existence (or non-existence!) would settle the famous "P = NP" question (Grollman & Sellman [GS88]). However, we have a number of candidates for one-way functions, i.e. functions which are easy to compute, but no efficient algorithm for their inversion is known. Two such candidates are integer multiplication and modular exponentiation for a fixed base. While it requires one arithmetic operation to multiply two integers, the number of arithmetic operations required for the fastest general factoring technique is subexponential in the number of bits of the integer to be factored. The difficulty of factoring is discussed in more detail in Section 2.1. Similarly, given $m \in \mathbb{Z}^{>0}$, $a \in \{0, \ldots, m-1\}$, it takes no more than 2log n multiplications and 2log n reductions (mod m) to compute $b \equiv a^n \pmod{m}$, $0 \leq b \leq m-1$ (see Algorithm 2.1 in Section 2.1). However, given $m \in \mathbb{Z}^{>0}$ and $a, b \in \{1, \ldots, m-1\}$ such that $a^n \equiv b \pmod{m}$ for some $n \in \mathbb{Z}^{>0}$, the *discrete logarithm problem* (DLP) of finding n is generally very difficult. More details on the DLP can be found in Section 8.2.

Clearly, one-way functions cannot be used for encryption, since even the intended recipient of a message would not be able to extract plaintext that has been enciphered using a one-way function. But if the decrypter were given certain *trapdoor information* which enabled him to invert the one-way function quickly, then such a function is a good candidate to be used for a PKC. A trapdoor one-way function is a function $f: X \rightarrow Y$ such that the following holds.

1. For all $x \in X$, $f(x)$ is easy to compute.

2. For almost all $y \in Y$, if there exists $x \in X$ such that $f(x) = y$, then it is hard to compute x, unless some special information I used in the design of f is known. In this case, there exists a function g_I such that $g_I(y) = x$ is easy to compute.

Note that the encryption transformation E_{K_p} of a PKC constitutes a trapdoor one-way function. Here, the special information I is the secret key K_s, and the inverse function g_I is the decryption transformation D_{K_s}.

A candidate for a trapdoor one-way function is squaring modulo a *Blum integer*, i.e. an integer $m = pq$ which is the product of two distinct primes $p, q \equiv 3 \pmod 4$. For $a \in \mathbb{Z}^{>0}$, $b \equiv a^2 \pmod m$ can be computed in one multiplication and reduction step, whereas extracting an integer square root $a \pmod m$ of $b \pmod m$ (provided one exists) is generally infeasible without extra information. However, if the factors p and q of m are given as trapdoor information, then the following procedure finds $a \pmod m$ using no more than a multiple of $\log_2 m$ arithmetic operations (we assume that there exists a square root $x \in \mathbb{Z}$ such that $x^2 \equiv b \pmod m$).

1. Compute $u \equiv b^{\frac{p+1}{4}} \pmod p$, $0 \le u \le p\text{-}1$, and $v \equiv b^{\frac{q+1}{4}} \pmod q$, $0 \le v \le q\text{-}1$
2. Find $a \in \mathbb{Z}$ such that $a \equiv u \pmod p$, $a \equiv v \pmod q$, $0 \le a \le m$, using the Chinese Remainder Theorem.

We have $a^2 \equiv u^2 \equiv \left(b^{\frac{p+1}{4}}\right)^2 \equiv b^{\frac{p+1}{2}} \equiv b^{\frac{p-1}{2}} b \equiv x^{p-1} b \equiv b \pmod p$, since by Euler's theorem $x^{p-1} \equiv 1 \pmod p$. Similarly, $a^2 \equiv b \pmod q$, hence $a^2 \equiv b \pmod m$.

Another example of a candidate for a trapdoor one-way function is modular exponentiation for a fixed exponent. As mentioned before, $b \equiv a^n \pmod m$ can be computed quickly, but no efficient algorithm is known to extract an n-th root of b (mod m) (provided one exists), unless the factorization of m is given as trapdoor information, in which case the n-th root can be computed in a multiple of $\log m$ steps. This trapdoor one-way function is the basis for the RSA public-key system discussed in Section 2.1.

PKC's have the advantage of eliminating the problem of key distribution. Their main disadvantage is the fact that they tend to be much slower than most one-key cryptosystems; for example, a secure hardware implementation of the RSA system is approximately 1000 times slower than a DES chip. Consequently, many practical applications use a *hybrid system*. These systems employ a PKC to communicate a private key for a one-key cryptographic session, and a private-key system is used for subsequent data exchange.

1.1.3 Digital Signatures

A *digital signature* is the cryptographic equivalent of a written signature, i.e. a means by which a recipient of a message can verify the identity of the sender. If Alice sends a digitally signed message to Bob, he not only knows that the message was signed by none other than Alice, but he can also convince any third party of this. A digital signature must satisfy three conditions:

1. Nobody but the sender can generate the signature.
2. The receiver can easily verify the signature.
3. If the sender should disavow signing a message, it must be possible for any judge or third party to resolve a dispute arising between sender and receiver.

A PKC provides signature capability if $\mathcal{M} = \mathcal{C}$ and if for any pair K_p, K_s of keys, the property $E_{K_p}(D_{K_s}(M)) = M$ holds for any message $\mathcal{M}$. To sign a (non-confidential) message M, Alice uses her secret key K_s to compute $S = D_{K_s}(M)$. She sends M along with S and her identity to Bob. Bob establishes the sender's identity to be that of Alice, uses Alice's public encryption key K_p to compute $\hat{M} = E_{K_p}(S)$, and declares the signature valid if $\hat{M} = M$.

Now assume that Alice wishes to send confidential information to Bob and he in turn wants to be sure that none other than Alice sent the information. Alice generates a signature $S = D_A(M)$ using her secret key. Then she uses Bob's public key to compute $T =$

$E_B(M)$. She sends T and her identity to Bob. From the identity tag, Bob identifies Alice as the sender, decrypts T to get S = $D_B(T)$, and finally recovers Alice's original message $M = E_A(S)$ by encrypting the signature under her public key.

Digital signatures can also be used to foil an attempt of impersonation by an opponent. Suppose a tamperer generates a random "signature" S, computes the "message" $M = D_A(S)$, and transmits M, S, and Alice's identity to Bob. Bob validates the signature by computing $E_A(M) = S$ and is lead to believe that Alice sent and signed M. Internal redundancy exposes this attack – Bob will notice that M is a meaningless message – but if the message being sent was supposed to be random (say, a key for a private-key cryptographic session), the tamperer will have succeeded. To prevent this attack, Alice could sign her message M with $S = D_A(f(M))$ instead of $S = D_A(M)$ where f is a public one-way function. To validate the signature, Bob computes $E_A(S)$ and $f(M)$. If the two are the same, the signature is valid and the message must have come from Alice.

1.1.4 Key Exchange

PKC's present one solution to the problem of safe key distribution. Secure key exchange is another approach. Suppose two communication partners wish to engage in a private-key cryptographic conversation, but have no secure channel available to communicate a key. In 1976, Diffie and Hellman [DH76] introduced the following *cryptographic protocol* (i.e. an algorithm for communications between different parties) which allows both partners to agree on a common key without revealing it to an eavesdropper.

1. Alice and Bob publicly agree on a large prime p and a *primitive root* g (mod p), i.e. a generator of the multiplicative group of residues a (mod p) where $p \nmid a$.
2. Alice generates a random integer $a \in \{1, \ldots, p\text{-}2\}$. She computes $x \equiv g^a \pmod{p}$, $1 \le x \le p\text{-}1$, and transmits x to Bob.
3. Bob generates a random integer $b \in \{1, \ldots, p\text{-}2\}$. He computes $y \equiv g^b \pmod{p}$, $1 \le y \le p\text{-}1$, and transmits y to Alice.

4. Alice computes $K \equiv y^a \pmod{p}$, $1 \leq K \leq p\text{-}1$.
5. Bob computes $K \equiv x^b \pmod{p}$, $1 \leq K \leq p\text{-}1$.

Note that $x^b \equiv g^{ab} \equiv y^a \pmod{p}$, so Alice and Bob indeed generate the same key. The protocol requires one round of communication and the quantities transmitted require no more than log p bandwidth. Furthermore, all powers can be computed in at most 2log p multiplications and 2log p reductions (mod p).

An opponent tapping the line can obtain p, g, x, and y, but does not know a or b. No efficient method for retrieving K without knowledge of a or b is known. If the cryptanalyst has a fast algorithm which for any $z \in \mathbb{Z}^{>0}$ and primitive root $g \pmod{p}$ computes $c \in \mathbb{Z}^{>0}$ such that $g^c \equiv z \pmod{p}$ or $c = \log_g z$ in the finite field GF(p) of p elements, then the key exchange protocol is broken. This is the already mentioned discrete logarithm problem which will be discussed in more detail in Section 8.2.

A different approach to secure key distribution without the use of a private channel was introduced by Merkle [Me78]. Here, two parties decide on a common key by exchanging a number of *puzzles*. The cryptanalytic cost of this scheme grows as n^2, where n is the cost to the legitimate users, hence the system is secure for n sufficiently large. However, the protocol requires large bandwidth, since n potential keys need to be communicated before one key can be agreed upon. In fact, Merkle points out that the high transmission overhead prevents his scheme from being practical.

1.1.5 Organisation of the Thesis

This thesis presents two cryptographic schemes that are based on the algebra and arithmetic in certain number theoretic structures. Part I (Chapters 2-7) discusses public-key cryptography based on modular exponentiation. We review previous work in Chapter 2, summarize the required mathematical ideas in Chapter 3, and introduce a new modular exponentiation-based cryptosystem in Chapter 4. The security of our scheme is analyzed in Chapter 5, and the algorithms used in our system are discussed in detail in Chapter 6.

Part I concludes with three specific cases and some computational results given in Chapter 7.

Part II (Chapters 8-13) of this dissertation introduces a new key exchange protocol. Chapter 8 presents the general Diffie-Hellman protocol and possible approaches for breaking it. In Chapter 9, we discuss the mathematical basis for the new protocol. The main algorithms of the new scheme are given in Chapter 10 and the problem of establishing a unique key is solved in Chapter 11. The final protocol is introduced in Chapter 11 as well; we point out that this is the first Diffie-Hellman protocol that does not require a group structure. Chapter 12 analyzes the scheme's security. We discuss our implementation and give some numerical examples in Chapter 13.

The dissertation concludes with some final remarks and a brief overview of open problems arising from our previous discussions in Chapter 14.

In the remainder of this Chapter, we will review some well-known mathematical facts that are repeatedly used throughout this document, and give a brief introduction to algebraic number theory.

1.2 Some Basics

We will briefly review a few basic and well-known mathematical facts and tools which we will use repeatedly throughout this dissertation.

1.2.1 Complexity

The performance of an algorithm is described in terms of its computational time and space requirements. Since all our procedures will be performing integer arithemtic, we will measure the time of an algorithm in terms of the number of basic integer arithmetic operations performed; these include addition, subtraction, multiplication, division, and comparison of two integers. We will not consider the computation time each such operation requires. Hence when referring to the *time complexity* of an algorithm, we mean the total number of integer operations performed. The *space complexity* of an algorithm is the size of its largest input throughout the computation, i.e. the number of bits in its largest input number, or the storage required by this number.

We will only be concerned with the asymptotic behaviour of the time and space complexity of an algorithm. Recall that if f, $g:\mathbb{Z}^{>0} \rightarrow \mathbb{R}^{>0}$ are two functions, we say that $f(n)$ is *big-Oh of* $g(n)$ and write $f(n) = O(g(n))$ if there exist $n_0 \in \mathbb{Z}^{>0}$ and $c \in \mathbb{R}^{>0}$ such that for all $n > n_0$, $f(n) < cg(n)$. We say that $f(n) = \Omega(g(n))$ if there exist $n_0 \in \mathbb{Z}^{>0}$ and $c \in \mathbb{R}^{>0}$ such that $f(n) > cg(n)$ for all $n > n_0$. In general, our algorithms will have both time and space complexity $O((\log P)^k)$ where $k \geq 1$ ($k = 1$ most of the time) and P is a parameter given by the cardinality of the mathematical structure underlying our algorithm.

1.2.2 Fermat's Little Theorem

In 1640, Fermat made the following observation (today referred to as his "Little Theorem"):

If p is a prime, then $a^{p-1} \equiv 1 \pmod{p}$ for any $a \in \mathbb{Z}$ such that $p \nmid a$.

This statement is an immediate consequence of the fact that $\mathrm{GF}(p)^* = \mathrm{GF}(p)-\{0\}$ is a cyclic multiplicative group, where $\mathrm{GF}(p)$ is the finite field of p elements. It is not to be confused with Fermat's well-known "(Last) Theorem" regarding the existence of solutions $(x, y, z) \in \mathbb{Z}^3$ of the equation $x^n + y^n = z^n$ $(n \in \mathbb{Z}^{>0})$.

1.2.3 Extended Euclidean Algorithm

It is well-known that we can compute the greatest comon divisor (gcd) $d = \gcd(a, b)$ of two integers a, b ($b > 0$) by performing repeated division with remainder

$$\begin{array}{ll} a = q_0 b + r_1, & 0 \leq r_1 < b, \\ b = q_1 r_1 + r_2, & 0 \leq r_2 < r_1, \\ r_1 = q_1 r_2 + r_3, & 0 \leq r_3 < r_2, \end{array}$$

$$\vdots \qquad\qquad\qquad \vdots$$

until we obtain $r_{n-1} = q_n r_n + r_{n+1}$ such that $r_{n+1} = 0$, in which case $r_n = \gcd(a, b)$. Then $n = O(\log \max\{|a|, b\})$ (see for example [Kn81, p. 343]) and all numbers computed by the algorithm require at most $O(\log \max\{|a|, b\})$ bits of storage.

By substituting backwards from $r_n = r_{n-2} - q_{n-1}r_{n-1} = r_{n-2} - q_{n-1}(r_{n-3} - q_{n-2}r_{n-2}) = \cdots$, we can obtain a representation of the gcd as a linear combination of a and b, i.e. we can find $x, y \in \mathbb{Z}$ such that $\gcd(a, b) = xa + yb$, in time $O(\log \max\{|a|, b\})$. This process is called the *Extended Euclidean Algorithm.*

If $k, m \in \mathbb{Z}$ are relatively prime, we can use the Extended Euclidean Algorithm to find x, $y \in \mathbb{Z}$ such that $xk + ym = 1 = \gcd(k, m)$. Then $xk \equiv 1 \pmod{m}$. Hence if we assume that $k < m$, then we can find the inverse $x \pmod{m}$ of $k \pmod{m}$ in time $O(\log m)$, and requiring space $O(\log m)$.

1.2.4 Chinese Remainder Theorem

The Chinese Remainder Theorem is used to solve systems of simultaneous linear congruences. Suppose $m_1, \ldots, m_n \in \mathbb{Z}^{>0}$ are pairwise relatively prime and $m = m_1 m_2 \cdots m_n$. We wish to find a solution $x \pmod{m}$ $(x \in \mathbb{Z})$ of the system of congruences

$$\begin{aligned} x &\equiv a_1 \pmod{m_1}, \\ x &\equiv a_2 \pmod{m_2}, \\ &\vdots \\ x &\equiv a_n \pmod{m_n}. \end{aligned}$$

For $1 \le i \le n$, we first find e_i such that $e_i \frac{m}{m_i} \equiv 1 \pmod{m_i}$, using the Extended Euclidean Algorithm, and set $x_i = e_i \frac{m}{m_i}$. Then $x_i \equiv 1 \pmod{m_i}$, $x_i \equiv 0 \pmod{m_j}$ for $j \neq i$. Set $x \equiv \sum_{k=1}^{n} a_k x_k$ (mod m). Clearly, x is our desired solution and can be found im time $O(\log m)$ (assuming that the number of congruences n is bounded), and requiring space $O(\log m)$.

1.3 Introduction to Algebraic Number Theory

Most of this material can be found in any introductory algebraic number theory text (see for example Stewart & Tall [ST79, Chapters 2, 4, 5, 9, 10.1, and 12]).

1.3.1 Algebraic Number Fields

Denote by $\mathbf{Q}$ the field of rationals, $\mathbf{Z}$ the ring of rational integers, $\mathbf{R}$ the field of real numbers, and $\mathbf{C}$ the field of complex numbers. For any ring S, we let S[x] be the ring of polynomials in x with coefficients in S. Let $\theta \in \mathbf{C}$ and let $f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0 \in \mathbf{Q}[x]$ be a monic irreducible polynomial with rational coefficients such that $f(\theta) = 0$ and $n = \deg(f(x))$ is minimal. Then we say that θ is *algebraic* over $\mathbf{Q}$. The set $\mathrm{K} = \mathbf{Q}(\theta) = \mathbf{Q} + \mathbf{Q}\theta + \mathbf{Q}\theta^2 + \cdots + \mathbf{Q}\theta^{n-1}$ is a subfield of $\mathbf{C}$ and a vector space of dimension n over $\mathbf{Q}$ for which the powers $1, \theta, \dots \theta^{n-1}$ form a basis. We call K an *algebraic number field* of *degree* $n = \dim_{\mathbf{Q}}\mathrm{K} = (\mathrm{K}:\mathbf{Q})$ over $\mathbf{Q}$. $f(x)$ is called a *generating polynomial* for K. If L is any subfield of K containing $\mathbf{Q}$, then K is also a vector space over L, i.e. an algebraic number field of degree $(\mathrm{K}:\mathrm{L}) = \dim_{\mathrm{L}}\mathrm{K} = \dfrac{(\mathrm{K}:\mathbf{Q})}{(\mathrm{L}:\mathbf{Q})}$ over L.

Denote by $\mathbf{O}$ the set of all $\alpha \in \mathrm{K}$ such that there exists a monic polynomial $g(x) \in \mathbf{Z}[x]$ such that $g(\alpha) = 0$. $\mathbf{O}$ is an *integral domain* in K, i.e. a commutative ring with no zero divisors and multiplicative identity 1; and $\mathbf{O}$ is a $\mathbf{Z}$-*lattice* of *rank* n, i.e. there exist $\omega_1, \omega_2, \dots, \omega_n \in \mathrm{K}$ such that $\mathbf{O} = [\omega_1, \dots, \omega_n] = \mathbf{Z}\omega_1 + \mathbf{Z}\omega_2 + \cdots + \mathbf{Z}\omega_n$ and $\omega_1, \omega_2, \dots, \omega_n$ is a $\mathbf{Q}$-basis of K. We call $\omega_1, \omega_2, \dots, \omega_n$ an *integral basis* of K and $\mathbf{O}$ the *maximal order* or the *ring of integers* in K. K is the *quotient field* of $\mathbf{O}$, i.e. every $\alpha \in \mathrm{K}$ can be written as $\alpha = \beta\gamma^{-1}$ where $\beta, \gamma \in \mathbf{O}$ and $\gamma \neq 0$. Furthermore $\mathbf{O} \cap \mathbf{Q} = \mathbf{Z}$.

Let $\theta = \theta_1, \theta_2, \dots, \theta_n$ be all the zeros of $f(x)$. The field homomorphisms σ_i: $\mathrm{K} \to \mathbf{C}$ defined by $\sigma_i(\theta) = \theta_i$ $(1 \leq i \leq n)$ are the *conjugate mappings* of K, and for any $\alpha \in \mathrm{K}$, the numbers $\sigma_i(\alpha)$ $(0 \leq i \leq n)$ are called the *conjugates* of α. After reordering the θ_i if necessary, we may assume that $\sigma_1, \dots, \sigma_s$ are the *real* conjugate mappings, i.e. $\sigma_i(\mathrm{K}) \subseteq \mathbf{R}$ for $0 \leq i \leq s$, and $\sigma_{s+1}, \overline{\sigma_{s+1}}, \dots, \sigma_{s+t}, \overline{\sigma_{s+t}}$ are the *complex* conjugate mappings, i.e. $\sigma_i(\mathrm{K})$

$\not\subseteq \mathbb{R}$ for $s+1 \le i \le s+t$ (here $\overline{\sigma_i}(\alpha) = \overline{\sigma_i(\alpha)}$ for $s+1 \le i \le s+t$, where $\overline{\beta}$ denotes the complex conjugate of β for any $\beta \in \mathbb{K}$). Then $s + 2t = n$. If $t = 0$, i.e. $s = n$, then we call $\mathbb{K}$ a *totally real* field; in the case where $s = 0$, i.e. $n = 2t$, $\mathbb{K}$ is said to be *totally complex*. $\mathbb{K}$ is a *normal* extension of $\mathbb{Q}$ if $\theta_1, \theta_2, \dots, \theta_n \in \mathbb{K}$ and hence $\sigma_i(\mathbb{K}) \subseteq \mathbb{K}$ and $\sigma_i(\mathbb{O}) \subseteq \mathbb{O}$ for $1 \le i \le n$. In this case, the conjugate mappings generate a group of order n, the *Galois group* of $\mathbb{K}$ over $\mathbb{Q}$. If $\mathbb{K}$ is normal over $\mathbb{Q}$, then $\mathbb{K}$ is a normal extension over any subfield $\mathbb{L}$ of $\mathbb{K}$ containing $\mathbb{Q}$. The Galois group of $\mathbb{K}$ over $\mathbb{L}$ consists of all those conjugate mappings which leave each elemement in $\mathbb{L}$ fixed. The order of the Galois group of $\mathbb{K}$ over $\mathbb{L}$ is $(\mathbb{L}:\mathbb{Q})$.

Let $\alpha \in \mathbb{K}$. We define $\mathrm{N}(\alpha) = \prod_{i=1}^{n} \sigma_i(\alpha)$ to be the *norm* of α and $\mathrm{Tr}(\alpha) = \sum_{i=1}^{n} \sigma_i(\alpha)$ to be the *trace* of α. Then $\mathrm{N}(\mathbb{K}), \mathrm{Tr}(\mathbb{K}) \subseteq \mathbb{Q}$ and $\mathrm{N}(\mathbb{O}), \mathrm{Tr}(\mathbb{O}) \subseteq \mathbb{Z}$. Forthermore $\mathrm{N}(\alpha\beta) = \mathrm{N}(\alpha)\mathrm{N}(\beta)$ and $\mathrm{Tr}(a\alpha + b\beta) = a\mathrm{Tr}(\alpha) + b\mathrm{Tr}(\beta)$ for $\alpha, \beta \in \mathbb{K}$ and $a, b \in \mathbb{Q}$. Note that for a totally complex field $\mathbb{K}$, $\mathrm{N}(\alpha) = \prod_{i=1}^{t} |\sigma_i(\alpha)|^2 \ge 0$ for any $\alpha \in \mathbb{K}$.

The *discriminant* Δ of $\mathbb{K}$ is the quantity $\Delta = (\det [\sigma_j(\omega_i)_{i,j=1,\dots,n}])^2 \in \mathbb{Z}$, where $\omega_1, \dots, \omega_n$ is any integral basis of $\mathbb{K}$. Δ is independent of the choice of the integral basis and is thus an invariant of $\mathbb{K}$. For $\alpha \in \mathbb{O}$, the *(element) discriminant* of α or the *(lattice) discriminant* of $\mathbb{Z}[\alpha] = \mathbb{Z} + \mathbb{Z}\alpha + \cdots + \mathbb{Z}\alpha^{n-1}$ is defined to be $d(\alpha) = d(\mathbb{Z}[\alpha]) = (\det [\sigma_j(\alpha^i)_{i,j=1,\dots,n}])^2$. Then $d(\alpha) = I(\alpha)^2\Delta$ where $I(\alpha) = (\mathbb{O} : \mathbb{Z}[\alpha])$ is the *index* of α, i.e. the index of $\mathbb{Z}[\alpha]$ in $\mathbb{O}$, or the cardinality of the factor ring $\mathbb{O}/\mathbb{Z}[\alpha]$. Hence Δ is divides the discriminant of any element $\alpha \in \mathbb{O}$. If $f(x) \in \mathbb{Z}[x]$ is a polynomial without multiple zeros such that $f(\alpha) = 0$, then $d(\alpha) = (-1)^{\frac{n(n-1)}{2}} d(f)$, where $d(f) = \mathrm{N}(f\,'(\theta))$ is the *(polynomial) discriminant* of $f(x)$ and $f\,'(\alpha)$ is the derivative of $f(x)$ at $x = \alpha$.

1.3.2 Units and Primes

There are two special kinds of elemens in every algebraic number field, namely units and primes. For two elements $\alpha, \beta \in \mathbf{O}$, $\beta \neq 0$, we say that β *divides* α, written $\beta \mid \alpha$, if there exists $\gamma \in \mathbf{O}$ such that $\beta\gamma = \alpha$.

A number $\varepsilon \in \mathbf{O}$ such that $\varepsilon \mid 1$ (i.e. ε is a divisor of 1 in $\mathbf{O}$) is called a *unit* in $\mathbf{K}$. If ε is a unit in $\mathbf{K}$, then ε^{-1} is a unit in $\mathbf{K}$ as well. $\varepsilon \in \mathbf{O}$ is a unit if and only if $N(\varepsilon) = \pm 1$.

Theorem 1.1: There exists a set of independent units $\eta_1, \dots, \eta_r$ such that every unit ε has a unique representation $\varepsilon = \zeta\eta_1^{e_1} \cdots \eta_r^{e_r}$ where $e_1, \dots, e_r \in \mathbf{Z}$, and ζ is a root of unity in $\mathbf{K}$. $\eta_1, \dots, \eta_r$ is called a set of *fundamental* units of $\mathbf{K}$. Here $r = s + t - 1$. r is called the *unit rank* of $\mathbf{K}$ and is an invariant of $\mathbf{K}$. □

Here, the term "independent" means that the equation $\eta_1^{e_1} \cdots \eta_r^{e_r} = -1$ has no solutions $(e_1, \dots, e_r) \in \mathbf{Z}^r$. In the case where $r = 1$, then we will always fix $\eta = \eta_1$ to be the unique fundamental unit exceeding 1. Note that η^{-1}, $-\eta^{-1}$, and $-\eta$ are also fundamental units satisfying $0 < \eta^{-1} < 1$, $-1 < -\eta^{-1} < 0$, and $-\eta < -1$.

If we define $l_i(\alpha) = \log |\sigma_i(\alpha)|$ for $1 \leq i \leq s$ and $l_i(\alpha) = \log\left(|\sigma_i(\alpha)|^2\right)$ for $s+1 \leq i \leq s+t$ for any $\alpha \in \mathbf{K}-\{0\}$, then the r vectors $(l_1(\eta_j), \dots, l_r(\eta_j))$ $(1 \leq j \leq r)$ can be shown to be linearly independent over the reals $\mathbf{R}$. The quantity $R = \left|\det\,[l_i(\eta_j)]_{i,j = 1, \dots, r}\right|$ is called the *regulator* of $\mathbf{K}$. R is independent of the choice of the system of fundamental units and is hence another invariant of $\mathbf{K}$.

If $\alpha, \beta \in \mathbf{O}$ are such that $\alpha = \varepsilon\beta$ for some unit ε, the α and β are said to be *associates*, and we write $\alpha \simeq \beta$. It is easy to see that the relation $\simeq$ is an equivalence relation on $\mathbf{O}$.

A *prime* in $\mathbf{O}$ is an element $\pi \in \mathbf{O}$ such that if $\pi \mid \alpha\beta$, then $\pi \mid \alpha$ or $\pi \mid \beta$ for any $\alpha, \beta \in \mathbf{O}$. Every non-zero $\alpha \in \mathbf{O}$ can be written as a product of prime powers in $\mathbf{O}$, but contrary to the case of prime decomposition in $\mathbf{Z}$, this representation need not be unique. For example, the number 21 in the field $\mathbf{Q}(\sqrt{-5})$ has two decompositions $21 = 3 \cdot 7 = (1+2\sqrt{-5})(1-2\sqrt{-5})$ into primes, none of which are pairwise associates.

1.3.3 Ideals

A subset $\mathbf{a}$ of $\mathbf{O}$ is an (*integral* $\mathbf{O}$-)*ideal* in $\mathbf{K}$ if both $\mathbf{a}+\mathbf{a}$ and $\mathbf{O}\mathbf{a}$ are subsets of $\mathbf{O}$. Any integral ideal $\mathbf{a}$ is a $\mathbf{Z}$-sublattice of $\mathbf{O}$ of rank n; if $\alpha_1, \dots, \alpha_n \in \mathbf{O}$ is a $\mathbf{Z}$-basis of $\mathbf{a}$, write $\mathbf{a} = [\alpha_1, \dots, \alpha_n]$. Denote by $\mathbf{I}$ the set of integral ideals in $\mathbf{K}$. A subset $\mathbf{b}$ of $\mathbf{K}$ is a (*fractional* $\mathbf{O}$-)*ideal* in $\mathbf{K}$ if $\mathbf{b} = \alpha\mathbf{a}$ where $\mathbf{a}$ is a non-zero integral ideal in $\mathbf{K}$ and $0 \neq \alpha \in \mathbf{K}$. For any fractional ideal $\mathbf{b}$, the set $\mathbf{b}^{-1} = \{\alpha \in \mathbf{K} \mid \alpha\mathbf{b} \text{ is an integral ideal in } \mathbf{K}\}$ is a fractional ideal in $\mathbf{K}$. If we define $\mathbf{ab} = \{\alpha\beta \mid \alpha \in \mathbf{a}, \beta \in \mathbf{b}\}$, then the fractional ideals form an Abelian group under multiplication with identity $\mathbf{O}$, and the integral ideals $\mathbf{I}$ form a semi-group.

If $\alpha_1, \dots, \alpha_k \in \mathbf{K}$, then the set $\mathbf{O}\alpha_1 + \dots + \mathbf{O}\alpha_k$ is a fractional ideal in $\mathbf{K}$ which is said to be *generated* by $\alpha_1, \dots, \alpha_k$; write $\mathbf{a} = (\alpha_1, \dots, \alpha_k)$. $\mathbf{a}$ is an integral ideal if and only if $\alpha_1, \dots, \alpha_k \in \mathbf{O}$. If $\mathbf{b} = (\beta_1, \dots, \beta_l)$, then $\mathbf{ab} = (\alpha_i\beta_j \mid 1 \leq i \leq k, 1 \leq j \leq l)$. Any ideal can be shown to require no more than two generators. An ideal $\mathbf{a} = (\alpha) = \alpha\mathbf{O}$ with a single generator is called *principal.* If $\alpha, \beta \in \mathbf{O}$, then $(\alpha) = (\beta)$ if and only if $\alpha \simeq \beta$. In particular, $\mathbf{O} = (\varepsilon)$ for any unit ε. The set of fractional principal ideals is a subgroup of the group of fractional ideals, and the set $\mathbf{P}$ of integral principal ideals is a sub-semigroup of the semi-group $\mathbf{I}$ of integral ideals.

The *norm* of an integral $\mathbf{a}$ is defined to be the cardinality $|\mathbf{O}/\mathbf{a}|$ of the factor ring $\mathbf{O}/\mathbf{a}$. The ideal norm is multiplicative, i.e. $N(\mathbf{ab}) = N(\mathbf{a})N(\mathbf{b})$ for any two integral ideals $\mathbf{a}$, $\mathbf{b}$ in $\mathbf{O}$. If $\mathbf{a} = (\alpha)$ is principal ($\alpha \in \mathbf{O}$), then $N(\mathbf{a}) = |N(\alpha)|$.

We say that an integral ideal $\mathbf{a}$ *divides* an integral ideal $\mathbf{b}$, written $\mathbf{a} \mid \mathbf{b}$, if there is an integral ideal $\mathbf{c}$ such that $\mathbf{ac} = \mathbf{b}$, or equivalently, if $\mathbf{b} \subseteq \mathbf{a}$. We write $\mathbf{a} \mid \alpha$ instead of $\mathbf{a} \mid (\alpha)$, $\alpha \in \mathbf{O}$, $\mathbf{a} \in \mathbf{I}$. $\alpha \equiv \beta \pmod{\mathbf{a}}$ is defined to mean $\mathbf{a} \mid \alpha - \beta$ for $\alpha, \beta \in \mathbf{O}$, $\mathbf{a} \in \mathbf{I}$.

Two integral ideals $\mathbf{a}$, $\mathbf{b}$ are *equivalent* if there exist non-zero α, $\beta \in \mathbb{O}$ such that $(\alpha)\mathbf{a} = (\beta)\mathbf{b}$, or equivalently, if there exists a non-zero $\gamma \in \mathbb{K}$ such that $\mathbf{a} = (\gamma)\mathbf{b}$; write $\mathbf{a} \sim \mathbf{b}$. This equivalence relation partitions the set of integral ideals into a set of equivalence classes which form a finite group under multiplication, called the *class group* Cl($\mathbb{K}$) of $\mathbb{K}$. The class group is exactly the factor group $\mathbb{I}/\mathbb{P}$. The order h of Cl($\mathbb{K}$) is called the *class number* of $\mathbb{K}$.

There are special representatives called reduced ideals in each ideal class. The idea of ideal reduction evolved out of the reduction theory for binary quadratic forms. It seems that reduced ideals were first mentioned by Berwick [Be28] for quadratic fields, later by Smadja [Sm73] for cubic fields, and finally by Williams [Wi85a]. For a fractional ideal $\mathbf{a}$, we define $m(\mathbf{a}) = \{m \in \mathbb{Z}_+ \mid m\mathbf{a} \subseteq \mathrm{O}\}$ and $\mathrm{L}(\mathbf{a}) = \min\{k \in \mathbb{Z}_+ \mid k \in m(\mathbf{a})\mathbf{a}\}$. When $\mathbf{a}$ is an integral ideal, we get $m(\mathbf{a}) = 1$ and $\mathrm{L}(\mathbf{a}) = \min\{\mathbf{a} \cap \mathbb{Z}_+)$, i.e. $\mathrm{L}(\mathbf{a})$ is the least positive rational integer in $\mathbf{a}$. An integral ideal $\mathbf{a}$ is said to be *primitive* if it has no rational integer divisors except 1, i.e. if $k \in \mathbb{Z}$ is such that $(k) \mid \mathbf{a}$, then $k = 1$. A number $0 \neq \alpha \in \mathbf{a}$ is a *minimum* in $\mathbf{a}$ if there is no $\beta \in \mathbf{a}$ such that $|\sigma_i(\beta)| < |\sigma_i(\alpha)|$ for $0 \leq i \leq s+t$.

Definition 1.2: An integral ideal $\mathbf{a}$ is said to be *reduced* if $\mathbf{a}$ is primitive and $\mathrm{L}(\mathbf{a})$ is a minimum in $\mathbf{a}$. ❑

The number of reduced ideals in $\mathbb{O}$ is finite and each ideal equivalence class contains at least one reduced ideal. Denote by $\mathfrak{R} \subseteq \mathbb{P}$ the set of reduced principal ideals in $\mathbb{K}$. $\mathfrak{R}$ need not be a group, since it is generally not closed under ideal multiplication.

An integral ideal $\mathbf{p}$ is called *prime* if $\mathbf{p} \mid \mathbf{ab}$ implies $\mathbf{p} \mid \mathbf{a}$ or $\mathbf{p} \mid \mathbf{b}$ for any pair of integral ideals $\mathbf{a}$, $\mathbf{b}$. It is easily shown that a principal integral ideal is prime if and only if its generator is a prime in $\mathbb{O}$.

Any non-zero integral ideal $\mathbf{a}$ has a unique representation (up to order) as a product of prime ideal powers. In particular, for every prime $p \in \mathbb{Z}$, the ideal (p) in $\mathbb{O}$ has a unique factorization $(p) = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_r^{e_r}$ where the $\mathfrak{p}_i$ are prime ideals. The factor ring $\mathbb{O}_{\mathfrak{p}_i} = \mathbb{O}/\mathfrak{p}_i$ is a finite field and a vector space over the finite field GF(p) of p elements. Denote by f_i the degree of $\mathbb{O}_{\mathfrak{p}_i}$ over GF(p), i.e. $f_i = \dim_{\mathrm{GF}(p)} \mathbb{O}_{\mathfrak{p}_i}$.

Theorem 1.3: Let p be a rational prime and let $(p) = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_r^{e_r}$ be the unique prime ideal factorization of the ideal (p) in $\mathbb{O}$. Let $f_i = (\mathbb{O}_{\mathfrak{p}_i}\text{:GF}(p))$. Then p satisfies the *decomposition law* $\sum_{i=1}^{r} e_i f_i = n$. ❑

If $\mathbb{K}$ is a normal extension over $\mathbb{Q}$, then $f_1 = \cdots = f_r$ and $e_1 = \cdots = e_r$, so the decomposition law reduces to $r_p e_p f_p = n$. A rational prime for which $e_i > 1$ for some i is said to be *ramified*. It can be shown that the ramified primes are exactly the prime divisors of Δ.

Theorem 1.4: Let $f(x) \in \mathbb{Q}[x]$ be monic and irreducible, θ a zero of $f(x)$, and let p be a rational prime such that $p \nmid I(\theta)$, the index of θ. Let $f(x) \equiv g_1(x)^{e_1} \cdots g_r(x)^{e_r} \pmod{p}$ be the unique decomposition of $f(x)$ into monic, modulo p irreducible polynomials. Then the prime ideal factorization and decomposition law of (p) are given by $(p) = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_r^{e_r}$ and $n = \sum_{i=1}^{r} e_i f_i$, where $f_i = \deg(g_i(x))$, and $\mathfrak{p}_i = (p, g_i(\theta))$ for $i \in \{1, \dots, r\}$. ❑

If $\mathfrak{p}$ is any prime ideal, then $\mathfrak{p} \cap \mathbb{Z} = p\mathbb{Z}$ for some prime p, hence there exists a unique rational prime p such that $\mathfrak{p} \mid p$. Let $f = (\mathbb{O}/\mathfrak{p}\text{:GF}(p))$, i.e. $|\mathbb{O}/\mathfrak{p}| = p^f$. Then $\mathrm{N}(\mathfrak{p}) = p^f$. If $\mathfrak{a} = \mathfrak{p}_1^{m_1} \cdots \mathfrak{p}_s^{m_s}$ is the unique prime ideal factorization of a non-zero integral ideal $\mathfrak{a}$, then $\mathrm{N}(\mathfrak{a}) = \mathrm{N}(\mathfrak{p}_1)^{m_1} \cdots \mathrm{N}(\mathfrak{p}_s)^{m_s}$.

A field $\mathbb{K}$ has class number 1 if and only if every ideal is principal. In this case the prime ideals are exactly those ideals whose generator is a prime in $\mathbb{O}$. If $(\alpha) = (\pi_1)^{e_1} \cdots (\pi_r)^{e_r}$ is the unique prime ideal decomposition of an integral ideal (α) $(0 \neq \alpha \in \mathbb{O})$, then it follows that $\alpha \simeq \pi_1^{e_1} \cdots \pi_r^{e_r}$ and this representation is unique up to order and unit factors. In this case we say that the maximal order is a *Unique Factorization Domain* (UFD). Thus the fields of class number 1 are exactly those whose ring of integers is a UFD.

1.3.4 Euclidean Division

Let $\mathbb{D}$ be an integral domain. As for rings of integers, we define an element $b \in \mathbb{D}$-$\{0\}$ to be a divisor of $a \in \mathbb{D}$ ($b \mid a$) if there exists $c \in \mathbb{D}$ such that $bc = a$. A unit is again a divisor of 1, and two elements $a, b \in \mathbb{D}$ are associates ($a \simeq b$) if $a = \varepsilon b$ for a unit ε.

A *Euclidean function* is a mapping f: $\mathbb{D}$-$\{0\} \rightarrow Z^{>0}$ such that the following two conditions hold.

i) If $a \mid b$, then $f(a) \leq f(b)$ for any $a, b \in \mathbb{D}$-$\{0\}$,

ii) For any $a, b \in \mathbb{D}$-$\{0\}$, there exist $q, r \in \mathbb{D}$ such that $a = qb+r$ and $r = 0$ or $f(r) < f(b)$.

$\mathbb{D}$ is said to be *Euclidean (for the function f)* if a Euclidean function f exists for $\mathbb{D}$. The process of finding q and r is called *Euclidean division.* For example, $\mathbb{Z}$ is Euclidean for the identity, and Euclidean division in $\mathbb{Z}$ is simply division with remainder.

There is a variety of algebraic number fields whose ring of integers is Euclidean for the absolute value of the norm $|\mathrm{N}|$. If $\mathbb{K}$ is an algebraic number field with maximal order $\mathbb{O}$, then it is easy to prove, using the fact that the norm is multiplicative, that $\mathbb{O}$ is Euclidean for the absolute value of the norm $|\mathrm{N}|$ if and only if for any $x \in \mathbb{K}$, there exists $y \in \mathbb{O}$ such that $|\mathrm{N}(x - y)| < 1$. If $\mathbb{O}$ is Euclidean for $|\mathrm{N}|$, then $\mathbb{O}$ is a UFD.

For any integral domain $\mathbb{D}$ and elements $a, b \in \mathbb{D}$, we call a number $d \in \mathbb{D}$ a *greatest common divisor* (gcd) of a and b if $d \mid a$, $d \mid b$, and for any $d_0 \in \mathbb{D}$ such that $d_0 \mid a$, $d_0 \mid b$, we have $d_0 \mid d$. It is easy to see that d is unique up to multiplication by a unit. a and b are said to be *relatively prime* if $\gcd(a, b) \simeq 1$. For any $a, b \in \mathbb{D}$, $d \simeq \gcd(a, b)$ can be found using the *Euclidean Algorithm*, i.e. by applying repeated Euclidean division

$$\begin{array}{ll}
a = q_0 b + r_1, & r_1 = 0 \text{ or } f(r_1) < f(b), \\
b = q_1 r_1 + r_2, & r_2 = 0 \text{ or } f(r_2) < f(r_1), \\
r_1 = q_1 r_2 + r_3, & r_3 = 0 \text{ or } f(r_3) < f(r_2), \\
\vdots & \vdots
\end{array}$$

until we obtain $r_{n-1} = q_n r_n + r_{n+1}$ such that $r_{n+1} = 0$. Then $r_n \simeq \gcd(a, b)$, and as for the Euclidean Algorithm in $\mathbb{Z}$, it follows that $n = O(\log \max\{f(a), f(b)\})$, and all numbers x computed by the algorithm satisfy $f(x) \leq \max\{f(a), f(b)\}$.

Part I

A Public-Key Cryptosystem Utilizing Cyclotomic Fields

2. Public-Key Cryptosystems Based on Modular Exponentiation

2.1 The RSA Public-Key Cryptosystem

The most widely used and tested PKC is the RSA system, named after its designers Rivest, Shamir, and Adleman [RSA78]. It is commercially available and is used by both the private sector and the US Government. The mathematical basis for this scheme is the problem of factoring – while it is easy to multiply two primes p, q, it appears to be very hard to extract the factors p, q from a given product $N = pq$.

Before presenting the details of RSA, we require the following definition. For any integer N, we define *Euler's totient function* $\phi(N)$ to be the number of residues a (mod N) such that a is relatively prime to N. If N is a prime, then clearly $\phi(N) = N\text{-}1$. If $N = pq$ is the product of two distinct primes p, q, then all the elements $a \in \{0, 1, \ldots, N\text{-}1\}$ are relatively prime to N except for 0, the p-1 multiples of q, and the q-1 multiples if p. Hence $\phi(N) = pq - (1 + (p\text{-}1) + (q\text{-}1)) = (p\text{-}1)(q\text{-}1)$.

2.1.1 Key Generation, Encryption, and Decryption

To generate a pair K_{p}, K_{s} of RSA keys, the designer chooses two distinct odd large primes p, q. (In [RSA78], it is suggested that p and q be roughly 100 decimal digits each to make the scheme sufficiently secure. For more details on prime generation, see Section 2.1.3). Set $N = pq$, then $\phi(N) =(p\text{-}1)(q\text{-}1)$. Next, the designer selects a random integer e such that $0 < e < N$ and $\gcd(e, \phi(N)) = 1$. He solves the congruence $ed \equiv 1 \pmod{\phi(N)}$ for $d \in \mathbb{Z}$, $0 < d < N$, using the Extended Euclidean Algorithm. The public key is the pair $K_{\mathrm{p}} = \{N, e\}$, the secret key is $K_{\mathrm{s}} = \{d\}$. N is the *modulus* and e and d are the *encryption* and *decryption exponents*, respectively, of the scheme. The primes p and q are discarded once the keys are generated. The size of the public key is bounded by $2\log N$.

Henceforth, we assume that our text which we wish to encrypt is represented numerically. This could be achieved by assigning a two digit integer to each character (A = 0, B = 1, ... , Z = 25, with further numbers for special characters and possibly lower case letters), or by associating with each character the binary value used to for its internal computer representation (i.e. the character's ASCII or EBCDIC code). We then divide our plaintext into numerical blocks M bounded by the modulus, i.e. $0 < M < N$, and treat each of these blocks separately with regards to encryption and decryption.

We may further assume that any such message block M is relatively prime to N. For suppose that $\gcd(M, N) > 1$, then since $0 < M < N$, $\gcd(M, N)$ must be p or q. The probability P of this event is $\frac{1}{p} + \frac{1}{q} \approx \frac{2}{\sqrt{N}}$ if p and q are of approximately equal size. Hence for N sufficiently large, we have $P < 10^{-100}$, a quantity which is obviously negligibly small[1].

If a sender Alice wishes to encrypt a message M, $0 < M < N$, under RSA, and transmit it to a receiver Bob, she first obtains Bob's public key $K_{\mathrm{p}} = \{N, e\}$. She computes $\mathrm{E}_{K_{\mathrm{p}}}(M) = C$, where $C \equiv M^e \pmod N$ and $0 < C < N$, and sends C to Bob. Bob can decrypt the ciphertext C, using his secret key $K_{\mathrm{s}} = \{d\}$, by calculating $\mathrm{D}_{K_{\mathrm{s}}}(C) = M'$, where $M' \equiv C^d \pmod N$ and $0 < M' < N$. Then it follows that $M' = M$ from the following theorem due to Euler which is a generalization of Fermat's Little Theorem.

Theorem 2.1 (*Euler*): If $M, N \in \mathbb{Z}^{\geq 0}$ are such that $\gcd(M, N) = 1$, then $M^{\phi(N)} \equiv 1 \pmod N$.

Proof: The residues $a \pmod N$ such that $\gcd(a, N) = 1$ form a multiplicative group $\mathbb{G}$ of order $\phi(N)$ with identity $1 \pmod N$. Since $M \pmod N \in \mathbb{G}$, we must have $M^{|\mathbb{G}|} \equiv M^{\phi(N)} \equiv 1 \pmod N$. ❑

[1] For comparison, the estimated number of hydrogen atoms in our galaxy is 10^{68} (Smith & Jacobs [SJ73, p. 537].

Corollary: Let $ed \equiv 1 \pmod{\phi(N)}$. If $C \equiv M^e \pmod N$, then $C^d \equiv M \pmod N$.

Proof: Let $ed = 1 + k\phi(N)$, $k \in \mathbb{Z}$. Then $C^d \equiv M^{ed} \equiv M^{1+k\phi(N)} \equiv M\left(M^{\phi(N)}\right)^k \equiv M \pmod N$ by Euler's Theorem. ❑

The Corollary implies $D_{K_s}(E_{K_p}(M)) = M$ for any message M. Since $E_{K_p}(D_{K_s}(C)) = C$ also holds for any ciphertext C, RSA can be used as a signature scheme as well.

To perform RSA encryption and decryption efficiently, the following well-known exponentiation technique (see for example Knuth [Kn81, pp. 441f.]) can be used.

Algorithm 2.1: For $M, N, n \in \mathbb{Z}^{>0}$, $0 < M < N$, compute $X \equiv M^n \pmod N$, $0 < X < N$.

```
1. Obtain the binary decomposition $n = \sum_{i=0}^{r} b_i 2^{r-i}$ of $n$, $b_i \in \{0,1\}$, $b_0 = 1$.
2. Set $X \leftarrow M$.
3. For $i = 1$ to $r$ do
       Set $X \leftarrow X^2 \pmod N$, $0 < X < N$.
       If $b_i = 1$, then set $X \leftarrow XM \pmod N$, $0 < X < N$. ❑
```

Clearly, no input of this algorithm requires more that $O(\log N)$ bits of storage, and the time complexity of the method is $O(r) = O(\log n)$, so since $e, d < N$, the time and space complexity of RSA encryption and decryption is $O(\log N)$.

2.1.2 Security

The security of RSA is at most as hard as factoring the modulus. For if a cryptanalyst knows p and q, he can easily compute $\phi(N) = (p\text{-}1)(q\text{-}1)$ and obtain the secret key by solving the congruence $ed \equiv 1 \pmod{\phi(N)}$ using the Extended Euclidean Algorithm. In fact, to find the secret key d, it suffices to know $\phi(N)$. However, knowledge of $\phi(N)$ is equivalent to knowledge of the factorization of N.

Lemma 2.2: Let $N = pq$. Given N and $\phi(N)$, the factors p and q of N can be computed in constant time.

Proof: Since $\phi(N) = (p - 1)(q - 1) = N - p - q + 1$, we have $p + q = N - \phi(N) + 1$. Let $b = \frac{N - \phi(N) + 1}{2}$. Then $p + q = 2b$ and $pq = N$, hence p and q must be equal to the roots $b \pm \sqrt{b^2 - N}$ of the quadratic equation $x^2 - 2bx + N = 0$. ❑

In fact, even recovering d will enable an adversary to factor N. For if d is known, then $ed-1$ is a multiple of $\phi(N)$, and Miller [Mi86] shows how knowledge of such a multiple enables an adversary to factor N. Hence recovering the secret key is equivalent to factoring N.

No way of breaking RSA is known other than retrieving the secret key. It is unknown whether breaking RSA is equivalent in difficulty to the factoring of N, i.e. whether plaintext messages can be retrieved illegitimately, without factoring N. Both exhaustive key search and factoring N are infeasible if N is sufficiently large.

The difficulty of an attack on RSA by factoring depends on the complexity of the factoring technique used. No polynomal-time algorithm for factoring is known; furthermore, it is not known whether such an algorithm exists. The best known general purpose factoring algorithms (Morrison & Brillhart [MB75], Pomerance [Po85], Coppersmith, Odlyzko & Schroeppel [COS86], Seysen [Se87], Lenstra [Le87], Lenstra & Pomerance [LP92]) all have a typical computation time of $L(N)^{c+o(1)}$, where $L(N) = \exp\left(\sqrt{\log N \log \log N}\right)$ and $c \geq 1$. The function $L(N)^{c+o(1)}$ is subexponential in $\log N$, since $(\log N)^k < L(N)^{c+o(1)} < N$ for any $k \in \mathbb{Z}^{>0}$ if N is sufficiently large. The asymptotic complexity of any of these procedures cannot be proved rigorously, but is based on heuristic arguments, except for the method in [LP92], where the bound is a rigorous one for the expected running time. In [Le87], Lenstra introduced a method using elliptic curves whose time complexity is $L(P)^{2+o(1)}$, where P is the largest prime factor of N, with a space requirement of $O(\log N)$

(as opposed to a polynomial in $L(N)$ for all other methods); this reduces to $L(N)$ in the case where $N = pq$ and p and q are approximately the same size.

Recently, an $L'(N)^{c+o(1)})$ factoring method, where $L'(N) = \exp\left(\sqrt[3]{\log N\ (\log\log N)^2}\right)$, was developed by Lenstra, Lenstra, Manasse, and Pollard [LLMP90] for numbers of the form $N = n^e \pm s$ where $n, s > 0$ and small. It is called the *number field sieve* and has been generalized to factor arbitrary integers by Buhler, Lenstra, and Pomerance [BLP93].

Pollard [Po75] gave a method for finding a factor of N having complexity $O(\sqrt{P})$ where P is the smallest prime factor of N. He also developed a practical method for discovering a prime divisor P of N when P-1 has only small prime factors [Po74], and Williams extended his technique to the case where P+1 has only small prime divisors [Wi82]. Hence, primes p, q such that any one of p-1, p+1, q-1, q+1 has only small prime factors should be avoided.

The ninth Fermat number $2^{2^9} + 1$, a 155 digit number, was factored in June 1990 by Lenstra, Lenstra, Manasse, and Pollard [LLMP93], using the number field sieve. It took a total of four months of computing time on a distributed network of workstations plus one supercomputer. Factoring a 200 digit number is still far beyond our present computational technology, and presumably this will not change for some time.

2.1.3 Choice of Safe Parameters

In their original paper [RSA78], the authors suggested generating random integers of approximately 100 digits and testing them for primality using the Solovay & Strassen test [SS77] to find primes p, q. However, it should be pointed out that it is possible to illegitimately retrieve plaintext messages and thus break RSA without factoring the modulus, if the primes p, q or the encryption exponent e are chosen carelessly.

As pointed out above, some factoring methods are quite efficient if one of the primes is chosen relatively small, hence p and q should be roughly of the same size. On the other hand, a difference of squares factoring attack is successful if $|p$-$q|$ is too small. If $y = \frac{p+q}{2}$,

$x = \frac{p-q}{2}$, then $N = y^2 - x^2$. If $|x|$ is known to be small, say $x < B \in \mathbb{Z}^{>0}$, then a linear search through the values of $x = 1, 2, \ldots, B$ will yield a non-trivial factorization of $N = (y-x)(y+x)$, if a value of x is found such that $N - x^2$ is a perfect square. Furthermore, Lehmer [Le07] proved that if $|p-q| < 2\sqrt[4]{N}$, then p and q can be found directly from the continued fraction expansion of $\sqrt{N}$. For other factoring attacks, see Schnorr [Sc83b].

In addition, $\delta = \gcd(p-1, q-1)$ should be small. Let the least common multiple of $p-1$ and $q-1$ be $\lambda(N)$, then $\phi(N) = \delta\lambda(N)$. The decryption exponent also satisfies the congruence $ed \equiv 1 \pmod{\lambda(N)}$, so if δ is large, then $\lambda(N)$ is small compared to $\phi(N)$, and it will be easy to find d using the molulus $\lambda(N)$.

It can be shown (DeMillo et al [DDDHL83, p. 47]) that any exponent e will leave at least nine messages unchanged when used for encryption, but a bad choice of e leaves up to half the messages unconcealed (Blakley & Blakley [BBl79], Blakley & Borosh [BBo79], Smith & Palmer [SP79]). The number of unconcealed messages can be kept to a minimum if both primes are chosen to be *safe*, i.e. $p = ip' + 1$ for some $i \in \mathbb{Z}^{>0}$ even and p' is a large prime (similarly for q); this guarantees that both $p-1$ and $q-1$ have at least one large factor. In addition, we should ensure that $p+1$ and $q+1$ also have at least one large factor each. For efficient techniques for choosing suitable primes see Gordon [Go85].

Cycling attacks using repeated encryption have been studied by Simmons & Norris [SN77], Herlestam [He78], Rivest [Ri78], [Ri79], Williams & Schmid [WS79], Berkovitz [Be82], and Jamnig [Ja88], but such attacks can be foiled by choosing the primes to be *doubly-safe*, i.e. $p = ip' + 1$ for some $i \in \mathbb{Z}^{>0}$ even and p' is safe; again similarly for q.

There are also successful attacks for small exponents. In [DDDHL83, pp. 58f.], it is shown how in the case where $e = 2$ a plaintext M can easily be retrieved if it is encrypted under two different moduli, and how two message which differ in only a few bits can be recovered if they are enciphered using the same modulus. Hasted [Ha86] showed that for any given encryption exponent, sending more than $\frac{e(e-1)}{2}$ linearly related messages encrypted under different moduli is insecure. Finally, Wiener [Wi90] introduced an attack which is based on the continued fraction expansion of $\frac{e}{N} \in \mathbb{Q}$ and is successful if $d < \sqrt[4]{N}$. To foil these attacks, e should be chosen to exceed the larger of p and q and d should be sufficiently large.

2.1.4 Modifications of RSA

Many modifications have been suggested for RSA. The scheme has been modified to be used in structures other than the multiplicative group of residues $a \pmod N$ where $\gcd(a, N) = 1$ (see Ecker [Ec83], or the extensions of RSA to matrix rings by Varadharajan [Va85] and Chuang & Dunham [CD91]). Kravitz & Reed [KR82] suggested replacing the primes p, q by irreducible binary polynomials $p(x)$, $q(x)$. Mueller & Noebauer [MN81] and Lidl & Mueller [LM84] proposed substituting the polynomial $f(x) = x^e \pmod N$ by more complicated functions, and recently, the use of Lucas sequences was suggested for an RSA-like scheme (Smith [Sm93], Smith & Lennon [SL93]).

Pohlig & Hellman [PH78] developed a secret-key cryptosystem similar to RSA. Here, instead of a composite number N, a prime modulus is used, and the exponents e and d are both kept secret as the private key. The security of this scheme depends on the difficulty of extracting discrete logarithms $\pmod p$ (see Section 8.2, Part II).

2.2 Rabin's Signature Scheme

Our analysis in the previous section leaves two open questions with regard to the security of RSA.

- How difficult is factoring the modulus?
- How difficult is breaking RSA?

The answers to both these questions are unknown. However, the problem of factoring has been known and studied for centuries, and up to now no polynomial-time factoring algorithm has been found, nor has the question of the existence of such an algorithm been answered. Factoring is widely believed to be hard, and for our purposes, we will be content to share this belief.

The situation with respect to the second problem is more hopeful. While we cannot give an answer to the question of RSA's security, it is nevertheless possible to modify RSA to obtain a PKC whose security is *equivalent* to the difficulty of factoring its modulus. This is achieved essentially by replacing the encryption exponent e by λe, where λ is a small prime. The receiver, upon decrypting $C \equiv M^{\lambda e}$ (mod N) using decryption exponent d, obtains not M, but M^{λ}, and needs to extract λ-th roots of M^{λ} in order to retrieve the original message M. This results in a λ-fold ambiguity for the value of M. While a message with internal redundancy (such as English text) can easily be distinguished, the ambiguity causes a problem if the plaintext is random, such as a key for a single-key cryptographic session. So the encrypter needs to provide information indicating the correct root of M^{λ} to the decrypter.

The idea of raising a text to the λ-th power before transmission, and extracting λ-th roots upon receipt was first introduced by Rabin [Ra79] for the case $\lambda = 2$. His system is a signature scheme, in which the signature S to a message M is essentially any one of the four square roots of M^2 (mod N), so here the ambiguity does not cause a problem. Specifically, a person wishing to sign a message selects two distinct odd large primes p, q, computes $N = pq$, and chooses a random integer b such that $0 < b < N$ and $\gcd(b, N) = 1$. N and b are made public. To sign a message M, $0 < M < N$, the signer computes a solution S, $0 < S < N$, of the congruence $S'(S' + b) \equiv M$ (mod N), i.e. a root S' of the congruence $x^2 + bx - M \equiv 0$ (mod N). To verify the signature S', the receiver computes $V' \equiv S'(S' + b)$ (mod N), $0 < V < N$. The signature is valid if $V = M$. If we set $S \equiv S' + \frac{b}{2}$ (mod N) and $V \equiv V' + \frac{b^2}{4}$ (mod N), $0 < S, V < N$, then the message M is signed with S such that $S^2 \equiv M$ (mod N) and verification is done by computing $V \equiv S^2$ (mod N) and checking whether $V = M$. (Here, $\frac{b}{2} \in \mathbb{Z}$ if b is even, and $\frac{b}{2} \equiv tb$ (mod N) where $2t \equiv 1$ (mod N), if b is odd.)

To compute a square root $S \pmod N$, the signer computes $x, y \in \mathbb{Z}$ such that $x^2 \equiv M \pmod p$ and $y^2 \equiv M \pmod q$. Since $S \equiv \pm x \pmod p$, $S \equiv \pm y \pmod q$, the Chinese Remainder Theorem yields four different possibilities for S, anyone of which may be used as a signature to M.

If $p \equiv 3 \pmod 4$, then $x \equiv M^{\frac{p+1}{4}} \pmod p$ is a square root of $M \pmod p$ as pointed out in Section 1.1.2. If $p \equiv 1 \pmod 4$, then one can use the methods by Shanks/Tonelli (Shanks [Sh73], Dickson [Di66, p. 215]) or Lehmer/Cipolla (Lehmer [Le69], [Di66, p. 218]) to compute a square root $x \pmod p$. Rabin also gives an $O(\log p)$ probabilistic algorithm for finding $x \pmod p$. Hence we can sign messages in time $O(\log N)$ and verify signatures in constant time.

Clearly, a message M can only be signed if $M \equiv S^2 \pmod N$ has a solution $S \in \mathbb{Z}$. This is the case if and only if the two congruences $x^2 \equiv M \pmod p$ and $y^2 \equiv M \pmod q$ have solutions, i.e. if and only if M is a *quadratic residue* (both (mod p) and (mod q) (opposite: *quadratic non-residue*). There is an efficient algorithm for checking whether or not $x \in \mathbb{Z}$ is a quadratic residue (mod p) by computing its *Legendre symbol.*

Definition 2.3: Let $p \in \mathbb{Z}$ be a prime and let $a \in \mathbb{Z}$. We define the *Legendre symbol* $\left(\frac{a}{p}\right)$ of a over p as follows.

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue } (\text{mod } p) \\ -1 & \text{if } a \text{ is a quadratic non-residue } (\text{mod } p) \\ 0 & \text{if } p \mid a \end{cases}$$

If $n \in \mathbb{Z}^{>0}$ and $n = p_1^{e_1} \cdots p_r^{e_r}$ is the unique prime factorization of n, then we define the *Jacobi symbol* $\left(\frac{a}{n}\right)$ of a over n as $\left(\frac{a}{n}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{e_i}$. ❑

Lemma 2.4: Let $p \in \mathbb{Z}$ be a prime and $a, b \in \mathbb{Z}$. Then the following holds.

a) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$

b) $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$ if $a \equiv b \pmod p$

c) $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$ ❑

So $S^2 \equiv M \pmod N$ has a solution S if and only if $\left(\frac{M}{p}\right) = \left(\frac{M}{q}\right) = 1$. If a message M is such that $\left(\frac{M}{p}\right) = -1$ or $\left(\frac{M}{q}\right) = -1$, then Rabin suggests appending a random string U to M to obtain M', $0 < M' < N$, and checking whether $\left(\frac{M'}{p}\right) = \left(\frac{M'}{q}\right) = 1$. Since the number of quadratic residues (mod p) is equal to the number of quadratic non-residues (mod p) (namely $\frac{p-1}{2}$), one expects to find M' such that $\left(\frac{M'}{p}\right) = \left(\frac{M'}{q}\right) = 1$ after four trials at a random string U. An $O(\log p)$ procedure for evaluating $\left(\frac{M}{p}\right)$ is given in Section 7.1.

The security of this signature scheme can be shown to be equivalent to the difficulty of factoring N as follows. Suppose an adversary is able to extract a square root S' (mod N) of a message M (mod N) signed with signature S. If $S' \equiv S \pmod N$ or $S' \equiv -S \pmod N$, then no knowledge is gained. However, if $S' \not\equiv S, -S \pmod N$, i.e. $S' \equiv S \pmod p$ and $S' \equiv -S \pmod q$ or $S' \equiv -S \pmod p$ and $S' \equiv S \pmod q$, then it follows from $S'^2 \equiv M \equiv S^2 \pmod N$ that $N \mid (S' - S)(S' + S)$, $N \nmid S' - S$, and $N \nmid S' + S$. Hence $S' - S$ is a multiple of p or q, so computing $\gcd(S' - S, N) = p$ or q will reveal the factorization of N. Hence an algorithm for extracting signatures requiring time $T(N)$ will factor N in time $T(N) + O(\log N)$ with a 50 percent likelihood.

Unfortunately, any constructive proof of the equivalence of breaking the scheme and factoring its modulus, such as the one above, gives rise to a *chosen message attack*. This is an attack by which a cryptanalyst can break the scheme if he is in possession of a particular message of his choice and the corresponding signature (a similar chosen ciphertext attack can be mounted on the cryptosystems discussed below, see Chapter 5). Suppose an adversary generates a random integer S' and computes $M \equiv S'^2 \pmod N$. He then asks the signer to sign M and reveal the corresponding signature S to him. If $S \not\equiv S', -S' \pmod N$, then the attacker can factor N and the system is broken. As indicated above, this attack has a 50 percent chance of success.

Rabin points out that his scheme can be extended to the cubic case where signatures S must satisfy $S^3 \equiv M \pmod{N}$. Here, each message M has nine possible signatures.

2.3 Public-Key Systems Equivalent to Factoring

Since Rabin's scheme, a number of PKCs based on modular exponentiation have been designed whose security is equivalent to the problem of factoring the system's modulus. Williams was the first to incorporate Rabin's idea of squaring and extracting square roots into a cryptosystem. In Williams [Wi80], he introduced a quadratic scheme which is generalized in Williams [Wi86]. A cubic system is also given in [Wi86]. Both these schemes will be discussed in more detail in Chapter 7. Harn and Kiesler [HK89] presented a different modification of Rabin's scheme which identifies the correct square root in the encryption and decryption algorithms.

A different extension of [Wi80] to the cubic case was recently presented by Loxton, Bird, Khoo, and Seberry [LKBS92]. In their system, encryption and decryption employ arithmetic in the quadratic field $\mathbf{Q}(\sqrt{-3})$ rather than the rational integers. Finally, another quadratic scheme is given in [Wi85b] (see also Salomaa [Sa90, pp. 159-166]). Here, encryption and decryption are performed in the quadratic field $\mathbf{Q}(\sqrt{c})$ where $c \in \mathbb{Z}$ is chosen such that $p \equiv -\left(\frac{c}{p}\right) \pmod{4}$ and $q \equiv -\left(\frac{c}{q}\right) \pmod{4}$. The system [Wi85b] is the only one which does not place any restrictions on the primes p and q (other than that they both be odd primes).

Chapters 3-7 of this dissertation present an RSA-like PKC which can be used as a quadratic and a cubic variant as well as with prime exponents exceeding 3 (see also Scheidler & Williams [SW92]). The scheme is a generalization of Williams' extended quadratic and cubic schemes [Wi86]. Like all the previous systems, it solves the problem of ambiguity in the decryption and its security is equivalent to the difficulty of factoring its modulus. Chapter 3 discusses the mathematical preliminaries, namely the algebra of cyclotomic fields. The system itself is given in Chapter 4 and its security is analyzed in Chapter 5. Chapter 6 presents in more detail the algorithms required for key generation, encryption, and decryption. In particular, we give methods for Euclidean division in cyclotomic fields, a generalization of the well-known division with remainder in $\mathbb{Z}$, and for the evaluation of residue symbols – these represent an extension of Legendre and Jacobi symbols to prime exponents other than 2. Finally, Chapter 7 discusses the quadratic and cubic case as well as, for the first time for a modular exponentiation-based system, the quintic case.

As mentioned before, all of the schemes discussed above address and solve the ambiguity problem in decryption. This can be done in two ways. The information required for the decrypter to distinguish the correctly deciphered message can be either given in the decryption algorithm as done in [Wi80] and [LKBS92], or transmitted together with the ciphertext, as done in [Wi86], [Wi85b], and our scheme.

Finally, it should be noted that for any system of this kind, there is a price to be paid for the additional information regarding its security over that of RSA.

- All schemes, with the exception of [Wi85b], need to place certain restrictions on the primes p, q, and thus on the modulus N. It is conceivable, though unlikely, that a modulus $N = pq$ of this special form is easier to factor than an arbitrary product of two distinct primes.
- The mechanisms for key generation, encryption, and decryption are more complicated than those for RSA and require more computation. However, the overall asymptotic complexity is the same as that of RSA, namely $O(\log N)$.
- The public key is larger than an RSA public key, up to twice as large ($4\log N$ bits rather than $2\log N$ bits in the quintic case).
- As mentioned in the discussion of Rabin's signature scheme, the proof of the equivalence of breaking the system to the problem of factoring its modulus is

constructive and can hence be used to mount a chosen ciphertext attack, similar to the chosen message attack against Rabin's scheme. However, a decrypter can foil such an attack by preventing any potential adversary from obtaining the decrypted message corresponding to a ciphertext of the adversary's choice.

It should be noted that our observations in Sections 2.1.3 and 2.1.4 regarding factoring and the choice of safe paramters also apply to the cryptosystems discussed in this section.

3. Introduction to Cyclotomic Fields

Much of this material can be found in Washington [Wa82, Chapters 1 & 2]. Let $\lambda \in \mathbb{Z}$ be a prime and let $\zeta \in \mathbb{C}$ be a *primitive λ-th root of unity*, i.e. $\zeta \in \mathbb{C}$ such that $\zeta \neq 1$ and $\zeta^\lambda = 1$. Then $f_\lambda(x) = \frac{x^\lambda - 1}{x - 1} = x^{\lambda-1} + \cdots + x + 1$ is the polynomial with rational coefficients and minimal degree such that $f_\lambda(\zeta) = 0$. In fact $f_\lambda(x) = \prod_{i=1}^{\lambda-1}(x-\zeta^i)$, so the powers ζ^i $(1 \leq i \leq \lambda-1)$ of ζ are exactly the zeros of $f_\lambda(x)$. The field $\mathbb{K} = \mathbb{Q}(\zeta)$ generated by ζ is called a *cyclotomic* field; its degree over $\mathbb{Q}$ is λ-1, and $f_\lambda(x)$ is a generating polynomial for $\mathbb{K}$.

It can be shown (see [Wa82, Proposition 1.2, pp. 1ff.]) that the powers ζ^i $(1 \leq i \leq \lambda-1)$ of ζ form an integral basis of $\mathbb{K}$, so $\mathbb{O} = \mathbb{Z}\zeta + \cdots + \mathbb{Z}\zeta^{\lambda-1} = \mathbb{Z}[\zeta]$. If $\lambda = 2$, then $\zeta = -1$ and $\mathbb{K} = \mathbb{Q}$, so assume $\lambda \geq 3$ for the remainder of this section. The conjugate mappings of $\mathbb{K}$ are given by $\sigma_i(\zeta) = \zeta^i$ for $1 \leq i \leq \lambda-1$, so $\mathbb{K}$ is a totally complex field. It follows that $\mathrm{N}(\alpha) \geq 0$ for all $\alpha \in \mathbb{K}$. Note that the last conjugate mapping $\sigma_{\lambda-1}$ is exactly the compex conjugation $\zeta \rightarrow \zeta^{-1}$ on $\mathbb{K}$. Since $\zeta^i \in \mathbb{K}$ for $1 \leq i \leq \lambda-1$, $\mathbb{K}$ is a normal extension over $\mathbb{Q}$ whose Galois group is isomorphic to the cyclic group $\mathrm{GF}(\lambda)^* = \mathrm{GF}(\lambda) - \{0\}$ of order λ-1.

Any unit $\varepsilon \in \mathbb{O}$ has a unique reprentation $\varepsilon = \pm\zeta^e\eta_1^{e_1} \cdots \eta_r^{e_r}$ where $e, e_1, \dots, e_r \in \mathbb{Z}$, $0 \leq e \leq \lambda-1$, and $\eta_1, \dots, \eta_r$ is a set of fundamental units in $\mathbb{K}$. Since the number s of real conjugate mappings is zero, it follows that $t = \frac{\lambda-1}{2}$, and the unit rank of $\mathbb{K}$ is $r = \frac{\lambda-3}{2}$.

Henceforth, let $\omega = 1-\zeta$. Then $\omega^{-1} = \frac{1}{\lambda}\alpha$ where $\alpha = \sum_{i=1}^{\lambda-1} i\zeta^i \in \mathbb{O}$.

Lemma 3.1: $\mathrm{N}(\omega) = \lambda$.

Proof: $\mathrm{N}(\omega) = \prod_{i=1}^{\lambda-1}(1-\zeta^i) = f_\lambda(1) = \lambda$. ❑

Lemma 3.2: The discriminant of K is $\Delta = (-1)^{\frac{(\lambda-1)}{2}} \lambda^{\lambda-2}$.

Proof: Since $(K:Q) = \lambda-1$, we have $\Delta = (-1)^{\frac{(\lambda-1)(\lambda-2)}{2}} d(f_\lambda(x))$. Now

$$f_\lambda'(x) = \frac{\lambda x^{\lambda-1}(x-1) - (x^\lambda-1)}{(x-1)^2} = \frac{\lambda x^{\lambda-1} - f_\lambda(x)}{x-1},$$

so $f_\lambda'(\zeta) = \dfrac{\lambda\zeta^{\lambda-1}}{\zeta-1}$ and $d(f_\lambda(x)) = N(f_\lambda'(\zeta)) = \dfrac{N(\lambda)N(\zeta)^{\lambda-1}}{N(\zeta-1)} = \dfrac{\lambda^{\lambda-1}}{\lambda} = \lambda^{\lambda-2}$, using the previous lemma. The rest follows from the fact that $\lambda-2$ is odd. ❑

Lemma 3.3 (*Decomposition of rational primes in $\mathcal{O}$*): Let p be a rational prime. If $p = \lambda$, then $p = \varepsilon\omega^{\lambda-1}$ where $\varepsilon = \prod_{i=1}^{\lambda-1}\varepsilon_i$, $\varepsilon_i = \dfrac{1-\zeta^i}{1-\zeta}$ is a unit for $1 \leq i \leq \lambda-1$, and ω is a prime. If $p \neq \lambda$, then $(p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{r_p}$ where $f_p = \mathrm{ord}(p \pmod{\lambda}) = \min\{n \in \mathbb{Z}_+ \mid p^n \equiv 1 \pmod{\lambda}\}$ and $r_p = \dfrac{\lambda-1}{f_p}$.

Proof: Assume $p = \lambda$. By Lemma 3.1, $\lambda = N(\omega) = \omega^{\lambda-1}\varepsilon$ where ε is given as above. Since $N(1-\zeta^i) = N(1-\zeta)$ for any i, it is clear that $\varepsilon_i = \dfrac{1-\zeta^i}{1-\zeta}$ is a unit for $1 \leq i \leq \lambda-1$. To prove that ω is a prime, assume that $\mathfrak{p}$ is a prime ideal such that $\mathfrak{p} \mid \lambda$. Then $\mathfrak{p} \mid \omega$ and hence $\mathfrak{p}^{\lambda-1} \mid \lambda$. By the decomposition law, we must have $(\lambda) = \mathfrak{p}^{\lambda-1}$, so in the decomposition law $e_p = \lambda-1, f_p = r_p = 1$, so $N(\mathfrak{p}) = \lambda$. Suppose $(\omega) = \mathfrak{p}\mathfrak{q}$ where $\mathfrak{q}$ is an integral ideal. Then $\lambda = N(\omega) = N(\mathfrak{p})N(\mathfrak{q}) = \lambda N(\mathfrak{q})$ and hence $N(\mathfrak{q}) = 1$, $\mathfrak{q} = \mathcal{O}$, and $\mathfrak{p} = (\omega)$. It follows that ω must be a prime.

Now suppose that $p \neq \lambda$, then p is not ramified by Lemma 3.2, hence by Theorem 1.3, $(p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{r_p}$. Let $\mathfrak{p} \in \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{r_p}\}$, i.e. $\mathfrak{p} \mid p$. Then $f_p = (\mathcal{O}/\mathfrak{p}:GF(p))$ and the field $\mathcal{O}/\mathfrak{p}$ is a normal extension over $GF(p)$ whose Galois group is a cyclic group generated by the *Frobenius automorphism* τ_p, characterized by $\tau_p(\alpha) \equiv \alpha^p \pmod{\mathfrak{p}}$ for all $\alpha \in \mathcal{O}$. In particular, $\tau_p(\zeta) \equiv \zeta^p \pmod{\mathfrak{p}}$. Let $m = \mathrm{ord}(p \pmod{\lambda})$, i.e. $p^m \equiv 1 \pmod{\lambda}$ and $m \in \mathbb{Z}^{>0}$ is minimal. Then $(\tau_p)^m(\zeta) \equiv \zeta^{p^m} \equiv \zeta \pmod{\mathfrak{p}}$, and if $\alpha = \sum_{i=1}^{\lambda-1} a_i\zeta^i \in \mathcal{O}$, $a_i \in \mathbb{Z}$ for

$1 \leq i \leq \lambda\text{-}1$, we have $(\tau_p)^m(\alpha) \equiv \sum_{i=1}^{\lambda-1} a_i(\tau_p)^m(\zeta)^i \equiv \alpha \pmod{\mathfrak{p}}$. Hence $(\tau_p)^m$ is the identity on $O/\mathfrak{p}$, and since m is minimal, we must have $m = \text{ord}(\tau_p) = f_p$. From the decomposition law, $r_p = \frac{\lambda-1}{f_p}$. ❑

Corollary 1: $p \equiv 1 \pmod{\lambda} \Rightarrow (p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{\lambda-1}$, so $e_p = f_p = 1$, $r_p = \lambda\text{-}1$.

Proof: Clear from $p \equiv 1 \pmod{\lambda} \Leftrightarrow \text{ord}(p \pmod{\lambda}) = 1$. ❑

Corollary 2: Let $p \neq \lambda$. Then $N(\mathfrak{p}) \equiv 1 \pmod{\lambda}$ for any prime ideal divisor $\mathfrak{p}$ of p.

Proof: Clear from $N(\mathfrak{p}) = p^{f_p}$ and $f_p = \text{ord}(p \pmod{\lambda})$. ❑

Corollary 3: Let $\alpha \in O$, $\alpha \neq 0$, and let $\mathfrak{p}$ be a prime ideal such that $\mathfrak{p} \nmid \alpha$. Then $\alpha^{N(\mathfrak{p})-1} \equiv 1 \pmod{\mathfrak{p}}$.

Proof: If $\mathfrak{p} = (\omega)$, then $N(\mathfrak{p}) = \lambda$. Now $\omega = 1\text{-}\zeta \mid 1\text{-}\zeta^i$, hence $\zeta^i \equiv 1 \equiv \zeta^{\lambda i} \pmod{\omega}$ for $1 \leq i \leq \lambda\text{-}1$, and by Fermat's little theorem, $a^\lambda \equiv a \pmod{\lambda}$ for all $a \in \mathbb{Z}$. Since $\lambda \mid \binom{\lambda}{i}$ for $1 \leq i \leq \lambda\text{-}1$, it follows that $(\beta + \gamma)^\lambda \equiv \beta^\lambda + \gamma^\lambda \pmod{\omega}$ for all $\beta, \gamma \in O$. Let $\alpha = \sum_{i=1}^{\lambda-1} a_i\zeta^i \in O$. Then $\alpha^\lambda \equiv \left(\sum_{i=1}^{\lambda-1} a_i\zeta^i\right)^\lambda \equiv \sum_{i=1}^{\lambda-1} a_i^\lambda\zeta^{\lambda i} \equiv \sum_{i=1}^{\lambda-1} a_i\zeta^i \equiv \alpha \pmod{\omega}$.

Assume now that $\mathfrak{p} \neq (\omega)$. Then if τ_p is the Frobenius automorphism generating the Galois group of $O/\mathfrak{p}$ over $GF(p)$, we have $\alpha \equiv \tau^{f_p}(\alpha) \equiv \alpha^{p^{f_p}} \equiv \alpha^{N(\mathfrak{p})} \pmod{\mathfrak{p}}$. ❑

Corollaries 2 and 3 of Lemma 3.3 enable us to define for any ideal $\mathfrak{a}$ in O a multiplicative mapping K → C as follows.

Definition 3.4: Let $\alpha \in O$ and let $\mathfrak{p}$ be a prime ideal in O. Then we define

$$\left[\frac{\alpha}{\mathbf{p}}\right] = \begin{cases} 0 \text{ if } \mathbf{p} \mid \alpha, \\ \zeta^k \text{ if } \mathbf{p} \nmid \alpha, \text{where } 0 \le k \le \lambda\text{-}1 \text{ and } \alpha^{\frac{N(\mathbf{p})-1}{\lambda}} \equiv \zeta^k \pmod{\mathbf{p}}. \end{cases}$$

$\left[\frac{\alpha}{\mathbf{p}}\right]$ is called the *λ-th residue symbol* of α over $\mathbf{p}$. For any non-zero integral ideal $\mathbf{a}$, we define $\left[\frac{\alpha}{\mathbf{a}}\right] = \prod_{i=1}^{r}\left[\frac{\alpha}{\mathbf{p}_i}\right]^{e_i}$, where $\mathbf{a} = \prod_{i=1}^{r}\mathbf{p}_i^{e_i}$ is the unique prime ideal factorization (up to order) of $\mathbf{a}$. ❑

For $\lambda = 2$, the residue symbol is simply the Legendre and Jacobi symbol as defined in Definition 2.3.

Lemma 3.5: Let $\alpha, \beta \in O - \{0\}$ and let $\mathbf{a}$, $\mathbf{b}$ be non-zero integral ideals in O.

a) $\left[\frac{\alpha}{\mathbf{a}}\right] = \left[\frac{\beta}{\mathbf{a}}\right]$ if $\alpha \equiv \beta \pmod{\mathbf{a}}$,

b) $\left[\frac{\alpha\beta}{\mathbf{a}}\right] = \left[\frac{\alpha}{\mathbf{a}}\right]\left[\frac{\beta}{\mathbf{a}}\right]$.

c) $\left[\frac{\alpha}{\mathbf{ab}}\right] = \left[\frac{\alpha}{\mathbf{a}}\right]\left[\frac{\alpha}{\mathbf{b}}\right]$.

Proof: By the definition of the residue symbol for composite denominators, there is nothing to prove for c), and it suffices to prove a) and b) for any prime ideal divisor of $\mathbf{a}$. Let $\mathbf{p}$ be such a divisor. Then $\alpha \equiv \beta \pmod{\mathbf{a}}$ implies $\alpha \equiv \beta \pmod{\mathbf{p}}$, so $\alpha^{\frac{N(\mathbf{p})-1}{\lambda}} \equiv \beta^{\frac{N(\mathbf{p})-1}{\lambda}} \pmod{\mathbf{p}}$ and hence $\left[\frac{\alpha}{\mathbf{p}}\right] = \left[\frac{\beta}{\mathbf{p}}\right]$. Similarly, $(\alpha\beta)^{\frac{N(\mathbf{p})-1}{\lambda}} \equiv \alpha^{\frac{N(\mathbf{p})-1}{\lambda}}\beta^{\frac{N(\mathbf{p})-1}{\lambda}} \pmod{\mathbf{p}}$, therefore $\left[\frac{\alpha\beta}{\mathbf{p}}\right] = \left[\frac{\alpha}{\mathbf{p}}\right]\left[\frac{\beta}{\mathbf{p}}\right]$. ❑

Definition 3.6: Let $\alpha \in O$ and $\beta \in O\text{-}\{0\}$. Then we define $\left[\frac{\alpha}{\beta}\right] = \left[\frac{\alpha}{(\beta)}\right]$. ❑

Note that this definition implies $\left[\frac{\alpha}{\beta}\right] = \left[\frac{\alpha}{\beta'}\right]$ for associates β, β'. For primes π, we have $N(\pi) = N\big((\pi)\big)$, hence $\left[\frac{\alpha}{\pi}\right] \equiv \alpha^{\frac{N(\pi)-1}{\lambda}} \pmod{\pi}$.

We will see in the next chapter that our cryptosystem requires an algorithm for evaluating λ-th residue symbols without factoring the denominator. Such an algorithm is only known for the case where the denominator of the residue symbol is an integer, rather than an integral ideal, as in the previous definition. In fact, currently we even require that O be Euclidean for the norm. This reduces our system to the cases where $\lambda \le 11$.

Lemma 3.7:

a) $2 \le \lambda \le 19 \quad \Leftrightarrow \quad$ O is a UFD.

b) $2 \le \lambda \le 11 \quad \Rightarrow \quad$ O is Euclidean for the norm N.

Proof: for a) see Masley & Montgomery [MM76]; for b) see Lenstra [Le75]. ❑

It is not known whether O is Euclidean in the cases λ = 13, 17, 19 (Lenstra [Le79]).

4. A New Public-Key Cryptosystem

Let $p, q \in \mathbb{Z}$ be rational primes such that $p \equiv q \equiv 1 \pmod{\lambda}$ and $p, q \not\equiv 1 \pmod{\lambda^2}$. Set $N = pq$ and $f = \frac{\phi(N)}{\lambda^2}$, where $\phi(N) = (p-1)(q-1)$. Let $e \in \mathbb{Z}^{>0}$, $\gcd(e, \phi(N) = 1$. Since $\lambda \nmid f$ and $\gcd(e, f) = 1$, the congruence $\lambda ed \equiv 1 \pmod{f}$ has a solution $d \in \mathbb{Z}$.

From Corollary 1 to Lemma 3.3, we obtain the unique prime ideal factorizations $(p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{\lambda-1}$, $(q) = \mathfrak{q}_1 \cdots \mathfrak{q}_{\lambda-1}$ for p and q, respectively. Let $\mathfrak{p} \in \{\mathfrak{p}_1, \dots, \mathfrak{p}_{\lambda-1}\}$, $\mathfrak{q} \in \{\mathfrak{q}_1, \dots, \mathfrak{q}_{\lambda-1}\}$, so $\mathrm{N}(\mathfrak{p}) = p$, $\mathrm{N}(\mathfrak{q}) = q$, and $\mathrm{N}(\mathfrak{pq}) = N$.

Let $r \in \mathbb{Z}^{>0}$ be such that $\gcd(r-1, N) = 1$, $r^\lambda \equiv 1 \pmod{N}$, and $\left[\frac{r}{\mathfrak{pq}}\right] = 1$. In Section 6.2, we will prove that r exists and give an algorithm for finding it.

Lemma 4.1: $\left[\frac{r}{\mathfrak{p}}\right] \neq 1$, $\left[\frac{r}{\mathfrak{q}}\right] \neq 1$.

Proof: Since $\gcd(r-1, N) = 1$, we have $r \not\equiv 1 \pmod{p}$, hence $r \not\equiv 1 \pmod{\mathfrak{p}}$. On the other hand, $r^\lambda \equiv 1 \pmod{N}$, hence $\mathfrak{p} \mid p \mid r^\lambda - 1 = (r-1)(r-\zeta)\cdots(r-\zeta^{\lambda-1})$, so $\mathfrak{p} \mid r - \zeta^i$ for some $i \in \{1, \dots, \lambda-1\}$. Therefore $\left[\frac{r}{\mathfrak{p}}\right] \equiv r^{\frac{p-1}{\lambda}} \equiv \zeta^{\frac{p-1}{\lambda}i} \pmod{\mathfrak{p}}$, and since $\lambda \nmid \frac{p-1}{\lambda}i$, we have $\left[\frac{r}{\mathfrak{p}}\right] \neq 1$. $\left[\frac{r}{\mathfrak{q}}\right] \neq 1$ follows from $\left[\frac{r}{\mathfrak{pq}}\right] = 1$. ❑

Just as the RSA scheme is based on Euler's theorem, a similar theorem gives rise to our cryptosystem.

Theorem 4.2: Let $X \in \mathbb{Z}$, $\gcd(X, N) = 1$, $\left[\frac{X}{\mathfrak{pq}}\right] = 1$. Then $X^f \equiv r^n \pmod{N}$ for some $n \in \{0, \dots, \lambda-1\}$.

Proof: Let $\left[\frac{X}{\mathbf{p}}\right] = \zeta^i$, $\left[\frac{r}{\mathbf{p}}\right] = \zeta^j$, $0 \le i \le \lambda\text{-}1$, $1 \le j \le \lambda\text{-}1$. Since $\left[\frac{X}{\mathbf{pq}}\right] = \left[\frac{r}{\mathbf{pq}}\right] = 1$, we have $\left[\frac{X}{\mathbf{q}}\right] = \zeta^{\lambda-i}$, $\left[\frac{r}{\mathbf{q}}\right] = \zeta^{\lambda-j}$. Then $r^{\frac{p-1}{\lambda}} \equiv \left[\frac{r}{\mathbf{p}}\right] \equiv \zeta^j$ (mod $\mathbf{p}$) and $r^{\frac{q-1}{\lambda}} \equiv \left[\frac{r}{\mathbf{q}}\right] \equiv \zeta^{-j}$ (mod $\mathbf{q}$). Define $n \in \mathbb{Z}$ such that $jn \equiv fi$ (mod λ) and $0 \le n \le \lambda\text{-}1$. Then $\zeta^{fi} \equiv \zeta^{jn} \equiv r^{\frac{p-1}{\lambda}n}$ (mod $\mathbf{p}$), so, after raising this congruence to the power $\left((p\text{-}1)/\lambda\right)^{-1}$ (mod λ), we get $\zeta^{\frac{q-1}{\lambda}i} \equiv r^n$ (mod $\mathbf{p}$). Similarly, $\zeta^{fi} \equiv r^{-\frac{q-1}{\lambda}n}$ (mod $\mathbf{q}$) and $\zeta^{-\frac{p-1}{\lambda}i} \equiv r^n$ (mod $\mathbf{q}$). Therefore $X^f \equiv \left[\frac{X}{\mathbf{p}}\right]^{\frac{q-1}{\lambda}} \equiv \zeta^{\frac{q-1}{\lambda}i} \equiv r^n$ (mod $\mathbf{p}$) and $X^f \equiv \left[\frac{X}{\mathbf{q}}\right]^{-\frac{p-1}{\lambda}} \equiv \zeta^{-\frac{p-1}{\lambda}i} \equiv r^n$ (mod $\mathbf{q}$), hence $X^f \equiv r^n$ (mod $\mathbf{pq}$). It follows that $X^f - r^n \in \mathbf{pq} \cap \mathbb{Z} = pq\mathbb{Z}$, so $X^f \equiv r^n$ (mod N). ❑

Corollary: If $Z \equiv X^{\lambda e}$ (mod N), then $Z^d \equiv r^k X$ (mod N) for some $k \in \{0, \dots, \lambda\text{-}1\}$.

Proof: Let $\lambda ed = 1 + lf$, $l \in \mathbb{Z}$, and let n be as in Theorem 4.2. Set $k \equiv nl$ (mod λ), $0 \le k \le \lambda\text{-}1$. Then $Z^d \equiv X^{\lambda ed} \equiv X^{1+lf} \equiv X(X^f)^l \equiv Xr^{nl} \equiv Xr^k$ (mod N). ❑

We are now ready to present our scheme.

Key Generation:

1. Choose two large primes p, q where $p \equiv q \equiv 1$ (mod λ), $p, q \not\equiv 1$ (mod λ^2).
2. Find prime ideals $\mathbf{p} \mid p$, $\mathbf{q} \mid q$. Compute $\mathbf{pq}$.
3. Set $N = pq, f = \frac{(p\text{-}1)(q\text{-}1)}{\lambda^2}$.
4. Chose $e \in \mathbb{Z}$, $0 < e < N$, $\gcd(e, \phi(N)) = 1$.
5. Solve the congruence $\lambda ed \equiv 1$ (mod f) for d, $0 < d < N$.
6. Find $S \in \mathbb{Z}$ such that $0 < S < N$ and $\left[\frac{S}{\mathbf{pq}}\right] = \zeta^{\lambda-1}$.
7. Set the public key to $K_p = \{r, S, N, e\}$ and the secret key to $K_s = \{d\}$.

As in RSA, the secret key d in Step 5 is computed using the Extended Euclidean Algorithm to find $d, x \in \mathbb{Z}$ such that $\lambda ed + xf = 1 = \gcd(\lambda e, f)$ and can be found in time $O(\log N)$, requiring space $O(\log N)$ for each input.

In Step 6, we merely require $\left[\frac{S}{\mathbf{pq}}\right] \neq 1$; the specification $\left[\frac{S}{\mathbf{pq}}\right] = \zeta^{\lambda-1}$ serves to simplify our arithmetic. In order to find S, generate a random integer T and compute $\left[\frac{T}{\mathbf{pq}}\right] = \zeta^k$ for some $k \in \{0, \dots, \lambda\text{-}1\}$. If $k = 0$, try another T, otherwise set $S \equiv T^l \pmod N$, $0 < S < N$, where $kl \equiv -1 \pmod \lambda$ and $1 \leq l \leq \lambda\text{-}1$. Then $\left[\frac{S}{\mathbf{pq}}\right] = \left[\frac{T}{\mathbf{pq}}\right]^l = \zeta^{kl} = \zeta^{\lambda-1}$.

Algorithms for finding r, $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{pq}$ and for evaluating residue symbols are given in Chapter 6. Note that our public key is up to twice as large as an RSA key (4log N bits), although S can usually be chosen small, resulting in a key size of approximately 3log(N) bits (50 percent larger than an RSA key).

Encryption: Let $M \in \mathbb{Z}$, $0 < M < N$ be a message, $\gcd(M, N) = 1$. Encrypt M as follows:

1. Determine $\left[\frac{M}{\mathbf{pq}}\right] = \zeta^m$ for some $m \in \{0, \dots, \lambda\text{-}1\}$.
2. Compute $M_0 \equiv MS^m \pmod N$, $M_i \equiv r^i M_0 \pmod N$ such that $0 < M_i < N$ for $0 \leq i \leq \lambda\text{-}1$.
3. Sort the M_i in ascending order to obtain $\hat{M}_0 < \cdots < \hat{M}_{\lambda-1}$ where $\{\hat{M}_0, \dots, \hat{M}_{\lambda-1}\} = \{M_0, \dots, M_{\lambda-1}\}$. (Note that all the M_i are pairwise distinct.) Find n such that $0 \leq n \leq \lambda\text{-}1$ and $M_0 = \hat{M}_n$.
4. Compute $C \equiv M_0^{\lambda e} \pmod N$, $0 < C < N$.
5. Transmit $\{C, m, n\}$.

Decryption: Upon receiving $\{C, m, n\}$:

1. Compute $L_0 \equiv C^d \pmod N$, $L_i \equiv r^i L_0 \pmod N$ such that $0 < L_i < N$ for $0 \leq i \leq \lambda\text{-}1$.
2. Sort the L_i in ascending order to obtain $\hat{L}_0 < \cdots < \hat{L}_{\lambda-1}$ where $\{\hat{L}_0, \dots, \hat{L}_{\lambda-1}\} = \{L_0, \dots, L_{\lambda-1}\}$. Find k such that $0 \leq k \leq \lambda\text{-}1$ and $L_k = \hat{L}_n$.

3. Compute S^{-1} (mod N). (This need only be done once for each modulus N).

4. Compute $\hat{M} \equiv S^{-m}L_k$ (mod N) such that $0 < \hat{M} < N$.

Lemma 4.3: $\hat{M} = M$.

Proof: For all $i \in \{0, \dots , \lambda\text{-}1\}$, we have $M_i^{\lambda} \equiv C$ (mod N) for $0 \leq i \leq \lambda\text{-}1$, and $\left[\frac{M_i}{\mathrm{pq}}\right] = \left[\frac{r}{\mathrm{pq}}\right]^i \left[\frac{M}{\mathrm{pq}}\right] \left[\frac{S}{\mathrm{pq}}\right]^m = \zeta^m\zeta^{(\lambda\text{-}1)m} = 1$, hence all M_i satisfy the requirements for Theorem 4.2 and its Corollary. It follows that for $0 \leq i \leq \lambda\text{-}1$: $L_i \equiv r^iL_0 \equiv r^iC^d \equiv r^{i+j \text{ (mod } \lambda)}M_0 \equiv M_{i+j \text{ (mod } \lambda)}$ (mod N) for some j, where all subscripts are taken to be between 0 and λ-1. Hence $\{L_0, \dots , L_{\lambda\text{-}1}\} = \{M_0, \dots , M_{\lambda\text{-}1}\}$, and after sorting, we get $\hat{L}_i = \hat{M}_i$ for $0 \leq i \leq \lambda\text{-}1$. Therefore $M_0 = \hat{M}_n = \hat{L}_n = L_k$, and finally $\hat{M} \equiv S^{-m}L_k \equiv S^{-m}M_0 \equiv M$ (mod N). Since $0 < M, \hat{M} < N$, we have $\hat{M} = M$. ❑

Lemma 4.3 implies that decryption is inverse to encryption. The security of our scheme is discussed in the next chapter, and efficient algorithms are given in Chapter 6.

5. Security

In order to prove that breaking our scheme is as difficult as factoring N, we first require three lemmas which are generalizations of results in [Wi86].

Lemma 5.1: Let $Y \in \mathbb{Z}$. Then there exists for any $i \in \{0, \dots , \lambda\text{-}1\}$ an integer X_i such that $X_i^{\lambda} \equiv Y^{\lambda}$ (mod N) and $\left[\frac{X_i}{\mathrm{pq}}\right] = \zeta^i \left[\frac{Y}{\mathrm{pq}}\right]$.

Proof: Let $i \in \{0, \dots , \lambda\text{-}1\}$ and let $j \in \mathbb{Z}$ be such that $\left[\frac{r}{\mathrm{p}}\right] = \zeta^j$. By Lemma 4.1, we can chose $j \in \{1, \dots , \lambda\text{-}1\}$. Let $k_i \in \mathbb{Z}$ be such that $jk_i \equiv i$ (mod λ). By the Chinese Remainder Theorem, there exists $X_i \in \mathbb{Z}$ such that $X_i \equiv r^{k_i}Y$ (mod p) and $X_i \equiv Y$ (mod q). Then $X_i^{\lambda} \equiv Y^{\lambda}$ (mod N) and $\left[\frac{X_i}{\mathrm{pq}}\right] = \left[\frac{X_i}{\mathrm{p}}\right]\left[\frac{X_i}{\mathrm{q}}\right] = \left[\frac{r}{\mathrm{p}}\right]^{k_i}\left[\frac{Y}{\mathrm{p}}\right]\left[\frac{Y}{\mathrm{q}}\right] = \zeta^{jk_i}\left[\frac{Y}{\mathrm{pq}}\right] = \zeta^i\left[\frac{Y}{\mathrm{pq}}\right]$. □

Lemma 5.2: Let $Y \in \mathbb{Z}$ such that $\gcd(Y, N) = 1$ and let $m, n \in \{0, \dots , \lambda\text{-}1\}$. If $C \equiv Y^{\lambda}$ (mod N) and $0 < C < N$, then there exists a unique $M \in \mathbb{Z}$, $0 < M < N$, such that encrypting M under key $K_{\mathrm{p}} = \{r, S, N, e\}$ yields $\{C, m, n\}$.

Proof: Since $\gcd(e, \phi(N)) = 1$, there exists $g \in \mathbb{Z}$ be such that $ge \equiv 1$ (mod $\phi(N)$). By the previous lemma, there exists $X \in \mathbb{Z}$ such that $X^{\lambda} \equiv (Y^g)^{\lambda}$ (mod N) and $\left[\frac{X}{\mathrm{pq}}\right] = 1$. For $0 \leq i \leq \lambda\text{-}1$, define $X_i \equiv r^iX$ (mod N), $0 < X_i < N$. Sort the X_i in ascending order, obtaining $\hat{X}_0 < \cdots < \hat{X}_{\lambda\text{-}1}$, where $\{X_0, \dots , X_{\lambda\text{-}1}\} = \{\hat{X}_0, \dots , \hat{X}_{\lambda\text{-}1}\}$, and let k be such that $X_k = \hat{X}_n$. Set $M \equiv S^{-m}X_k$ (mod N), $0 < M < N$. We need to prove that encrypting M under K_{p} yields $\{C, m, n\}$.

Step 1: $\left[\frac{M}{\mathrm{pq}}\right] = \left[\frac{S}{\mathrm{pq}}\right]^{-m}\left[\frac{r}{\mathrm{pq}}\right]^{k}\left[\frac{X}{\mathrm{pq}}\right] = \zeta^{-(\lambda-1)m} = \zeta^m$.

Step 2: $M_0 \equiv MS^m \equiv X_k$ (mod N), $M_i \equiv X_kr^i \equiv X_{i+k \text{ (mod } \lambda)}$ (mod N), $0 < M_i < N$ for $0 \leq i \leq \lambda\text{-}1$, where the subscript i+k (mod λ) is taken to be between 0 and λ-1.

Step 3: Since $\{M_0, \dots, M_{\lambda-1}\} = \{X_0, \dots, X_{\lambda-1}\}$, we have $\hat{M}_i = \hat{X}_i$ $(0 \le i \le \lambda-1)$ after sorting. Furthermore $M_0 = X_k = \hat{X}_n = \hat{M}_n$.

Step 4: $M_0^{\lambda e} \equiv X_k^{\lambda e} \equiv X^{\lambda e} \equiv Y^{\lambda e g} \equiv Y^{\lambda} \equiv C \pmod{N}$, using Euler's Theorem.

Hence encrypting M under key K_p gives $\{C, m, n\}$. Since decrypting $\{C, m, n\}$ under $K_s = \{d\}$ yields M, M must also be unique. □

Lemma 5.3: If $X, Y \in \mathbb{Z}$, $X^{\lambda} \equiv Y^{\lambda} \pmod{N}$, and $\left[\frac{X}{pq}\right] \neq \left[\frac{Y}{pq}\right]$, then $\gcd(X - r^iY, N) = p$ for some $i \in \{0, \dots, \lambda-1\}$.

Proof: $X^{\lambda} - Y^{\lambda} \equiv (X - Y)(X - rY) \cdots (X - r^{\lambda-1}Y) \equiv 0 \pmod{N}$. Assume that $X - r^iY \equiv 0 \pmod{N}$ for some $i \in \{0, \dots, \lambda-1\}$. Then $X \equiv r^iY \pmod{pq}$, hence $\left[\frac{X}{pq}\right] = \left[\frac{r}{pq}\right]^i \left[\frac{Y}{pq}\right] = \left[\frac{Y}{pq}\right]$ in contradiction our assumption. So there must exist $i \in \{0, \dots, \lambda-1\}$ such that $X - r^iY \equiv 0 \pmod{p}$ and $X - r^iY \not\equiv 0 \pmod{q}$. But then $\gcd(X - r^iY, N) = p$. □

Theorem 5.4: If A is an algorithm which, given any cipher $\{C, m, n\}$ will find the corresponding plaintext M, then the following algorithm will factor N:

1. Find $Y \in \mathbb{Z}$ such that $0 < Y < N$ and $\left[\frac{Y}{pq}\right] \neq 1$ (note that S is a possible choice for Y).

2. Put $C \equiv Y^{\lambda} \pmod{N}$, $0 < C < N$, and select any $m, n \in \{0, \dots, \lambda-1\}$.

3. Use A to decrypt $\{C, m, n\}$, obtaining M.

4. Put $M_0 \equiv MS^m \pmod{N}$, $X \equiv M_0^e \pmod{N}$.

5. For $0 \le i \le \lambda-1$, compute $\gcd(X - r^iY, N)$ until a nontrivial factor is found.

Proof: M in Step 3 is unique by Lemma 3.2. Since $\left[\frac{X}{pq}\right] = \left(\left[\frac{M}{pq}\right]\left[\frac{S}{pq}\right]^m\right)^e = (\zeta^m\zeta^{(\lambda-1)m})^e = 1$, $\left[\frac{Y}{pq}\right] \neq 1$, and $X^{\lambda} \equiv C \equiv Y^{\lambda} \pmod{N}$, by Lemma 5.3 we must have $\gcd(X - r^iY, N) = p$ for some $i \in \{0, \dots, \lambda-1\}$. □

Theorem 5.4 states that breaking the scheme is equivalent in difficulty to factoring the modulus N; hence, unless it is significantly easier to factor a number $N = pq$ such that $p \equiv q \equiv 1 \pmod{\lambda}$ and $p, q \not\equiv 1 \pmod{\lambda^2}$, compared to factoring the product $N = pq$ for arbitrary primes p, q, our system is secure, assuming that factoring is a hard problem.

Unfortunately, the algorithm given in Theorem 5.4 can be used to mount a chosen ciphertext attack similar to the chosen message attack which can be mounted against Rabin's scheme [Wi80]. Any constructive method for proving that the security of a cryptosystem is equivalent to factoring the modulus makes the system vulnerable to such an attack. An attacker need only generate Y and the cipher $\{C, m, n\}$ as in Theorem 5.4. Then he must convince his opponent to decrypt the triple $\{C, m, n\}$ and reveal the corresponding message M. By computing X and factoring N as in Theorem 5.4, he can then find the secret key d.

If A is such that it can only decrypt a fraction $\frac{1}{k}$ of all messages, then we expect to be able to find M and procede as above after k trials at a value of Y.

Finally, as pointed out in [Wi86], revealing r does not seem to compromise the security of the system. By Lemma 5.3, an adversary could factor N if he found a λ-th root of unity $X \pmod{N}$ such that $\left[\frac{X}{pq}\right] \neq 1$. But this corresponds to the case $C = 1$ in Theorem 5.4, so, unless the number 1 represents a special case, the problem of finding X is equivalent in difficulty to factoring N.

6. The Algorithms

As for RSA, we wish the overall complexity of our system to be $O(\log N)$. The key to our scheme is a fast method for evaluating the residue symbol $\left[\frac{M}{\mathbf{pq}}\right]$ for a message M, since the efficiency of such a method determines the overall efficiency of the encryption. Clearly, $\left[\frac{M}{\mathbf{pq}}\right]$ needs to be computed without factoring the denominator, as the knowledge of $\mathbf{p}$ or $\mathbf{q}$ would lead to the discovery of p and q, and would hence enable an adversary to break the system.

In addition, in order to generate our keys efficiently, we require fast methods for performing the following three tasks:

1. Given p, q, find prime ideals $\mathbf{p} \mid p$, $\mathbf{q} \mid q$ (or integer primes $\pi \mid p$, $\psi \mid q$).
2. Find $\mathbf{pq}$ (or $\pi\psi$).
3. Given p and q, find r such that $\gcd(r\text{-}1, N) = 1$, $r^{\lambda} \equiv 1 \pmod N$, and $\left[\frac{r}{\mathbf{pq}}\right] = 1$.

6.1 Prime Ideal Divisors and Integer Prime Divisors of Rational Primes

Lemma 6.1: Let $p \in \mathbb{Z}$ be a prime such that $p \equiv 1 \pmod \lambda$ and let $a \in \mathbb{Z}$ be a primitive λ-th root of unity (mod p), Then any $b \in \mathbb{Z}$ is a primitive λ-th root of unity (mod p) if and only if $b \equiv a^j \pmod p$ for some $j \in \{1, \ldots, \lambda\text{-}1\}$.

Proof: If $b \in \mathbb{Z}$, $b \equiv a^j \pmod p$ for some $j \in \{1, \ldots, \lambda\text{-}1\}$, then clearly $b \not\equiv 1 \pmod p$ and $b^{\lambda} \equiv a^{j\lambda} \equiv 1 \pmod p$. The polynomial $f_{\lambda}(x) = \frac{x^{\lambda}\text{-}1}{x\text{-}1}$ has at most λ-1 distinct roots (mod p), all of which are primitive λ-th roots of unity (mod p), Since the powers $a^j \pmod p$, $1 \leq j \leq \lambda\text{-}1$, are all distinct roots of $f_{\lambda}(x) \pmod p$, they must represent all the primitive λ-th roots of unity (mod p). ❑

Lemma 6.2: Let $p \in \mathbb{Z}$ be a prime such that $p \equiv 1 \pmod{\lambda}$. If a is a primitive λ-th root of unity (mod p), then $(p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{\lambda-1}$ where $\mathfrak{p}_i = (p, \zeta - a^i)$ for $1 \leq i \leq \lambda-1$.

Proof: Consider the generating polynomial $f_\lambda(x) = \prod_{i=1}^{\lambda-1}(x - \zeta^i)$ for $\mathbb{K}$. The zeros of $f_\lambda(x)$ (mod p) are exactly the primitive λ-th roots of unity (mod p), and by Lemma 6.1, these are exactly the powers of a (mod p). So $f_\lambda(x) \equiv \prod_{i=1}^{\lambda-1} g_i(x) \pmod{p}$, where $g_i(x) = x - a^i$. By Theorem 1.4, $(p) = \mathfrak{p}_1 \cdots \mathfrak{p}_{\lambda-1}$ and $\mathfrak{p}_i = (p, g_i(\zeta)) = (p, \zeta - a^i)$ for $1 \leq i \leq \lambda-1$. ❑

Theorem 6.3: Let $p \in \mathbb{Z}$ be a prime such that $p \equiv 1 \pmod{\lambda}$. Under the assumption of the Extended Riemann Hypothesis (ERH), a prime ideal divisor $\mathfrak{p}$ of p can be found in time $O\big((\log p)^3\big)$, but most likely in time $O(\log p)$, using no more than $O(\log p)$ bits of storage at each point in the algorithm.

Proof: By Lemma 6.2, p and a represent $\mathfrak{p}$ uniquely, so if a is chosen such that $0 < a < p$, then all inputs are bounded by p. To compute a, find a λ-th non-residue v (mod p), i.e. $v^{\frac{p-1}{\lambda}} \not\equiv 1 \pmod{p}$, and set $a \equiv v^{\frac{p-1}{\lambda}} \pmod{p}$, where we can choose $2 \leq a \leq p-1$. Since the λ-th residues (mod p) form a proper multiplicative subgroup of GF(p)-{0}, by a theorem of Bach [Ba90], the least positive λ-th non-residue n (mod p) satisfies $n < 2\log(p)^2$, assuming ERH. Hence we can find v in $O((\log p)^2)$ steps, although we expect to find one much faster (in fact, in constant time) by trial as there are $\frac{(\lambda-1)(p-1)}{\lambda}$ possible values for v between 1 and p-1 (there are only $\frac{p-1}{\lambda}$ λ-th residues (mod p) in this range). Once we have found v, a can be computed in time $O(\log p)$ using fast modular exponentiation. ❑

Lemma 6.4: Let $p \in \mathbb{Z}$ be a prime such that $p \equiv 1 \pmod{\lambda}$ and let $\mathfrak{p} = (p, \zeta - a)$. Then $\{p, \zeta - a, (\zeta - a)^2, \ldots, (\zeta - a)^{\lambda-2}\}$ is a $\mathbb{Z}$-basis for $\mathfrak{p}$.

Proof: Let $\mathfrak{a} = [p, \zeta - a, (\zeta - a)^2, \ldots, (\zeta - a)^{\lambda-2}]$. We wish to prove $\mathfrak{a} = \mathfrak{p}$.

We first show that $\mathbf{a}$ is an ideal. Clearly, $\mathbf{a} + \mathbf{a} \subseteq \mathbf{a}$ holds, so it suffices to show $\mathbb{O}\mathbf{a} \subseteq \mathbf{a}$. Since $\mathbf{a} + \mathbf{a} \subseteq \mathbf{a}$, $\mathbb{Z}\mathbf{a} \subseteq \mathbf{a}$, and any $\alpha \in \mathbb{O}$ is a $\mathbb{Z}$-linear combination of the powers of ζ, it suffices to show $\zeta^i\mathbf{a} \subseteq \mathbf{a}$ for $1 \leq i \leq \lambda\text{-}1$, for which in turn it is sufficient to prove $\zeta\mathbf{a} \subseteq \mathbf{a}$. Let $f_\lambda(x) = \frac{x^\lambda\text{-}1}{x\text{-}1}$ and $g(x) = f_\lambda(x+a)$. Since $g(x) \in \mathbb{Z}[x]$ is a monic polynomial of degree $\lambda\text{-}1$ and $0 = f_\lambda(\zeta) = g(\zeta\text{-}a)$, we see that there exist $a_0, \dots, a_{\lambda\text{-}2} \in \mathbb{Z}$ such that $(\zeta\text{-}a)^{\lambda\text{-}1} = \sum_{i=0}^{\lambda\text{-}2} a_i(\zeta\text{-}a)^i$. Furthermore, $a_0 = g(0) = f_\lambda(a) = \frac{a^\lambda\text{-}1}{a\text{-}1} \equiv 0 \pmod{p}$. Therefore $(\zeta\text{-}a)^{\lambda\text{-}1}$ is a $\mathbb{Z}$-linear combination of p and $(\zeta\text{-}a)^i$ $(1 \leq i \leq \lambda\text{-}1)$ and is hence in $\mathbf{a}$. Since $\zeta(\zeta\text{-}a)^i = (\zeta\text{-}a)^{i+1} + a(\zeta\text{-}a)^i$ for $1 \leq i \leq \lambda\text{-}2$, we see that $\zeta(\zeta\text{-}a)^i \in \mathbf{a}$ for $1 \leq i \leq \lambda\text{-}2$. Furthermore, $\zeta p = ap + p(\zeta\text{-}a) \in \mathbf{a}$, so $\zeta\mathbf{a} \subseteq \mathbf{a}$.

To see that $\mathbf{p} = \mathbf{a}$, we first observe that $p, \zeta\text{-}a, (\zeta\text{-}a)^2, \dots, (\zeta\text{-}a)^{\lambda\text{-}2} \in \mathbf{p}$, so $\mathbf{a} \subseteq \mathbf{p}$. Now $\mathbb{O}\mathbf{a} \subseteq \mathbf{a}$, in particular $\mathbb{O}p \subseteq \mathbf{a}$ and $\mathbb{O}(\zeta\text{-}a) \subseteq \mathbf{a}$, and since $\mathbf{a} + \mathbf{a} \subseteq \mathbf{a}$, it follows that $\mathbf{p} = \mathbb{O}p + \mathbb{O}(\zeta\text{-}a) \subseteq \mathbf{a} + \mathbf{a} \subseteq \mathbf{a}$. ❑

Lemma 6.5: Let $p, q \equiv 1 \pmod{\lambda}$, $N = pq$. Let $\mathbf{p} = (p, \zeta\text{-}a)$ and $\mathbf{q} = (q, \zeta\text{-}b)$ be prime ideal divisors of p and q, respectively, according to Lemma 6.2. If $c \in \mathbb{Z}$ is such that $c \equiv a \pmod{p}$ and $c \equiv b \pmod{q}$, then $\mathbf{pq} = (N, \zeta\text{-}c)$.

Proof: $\mathbf{p} = (p, \zeta\text{-}c)$, $\mathbf{q} = (q, \zeta\text{-}c)$ by definition of c. Ideal multiplication yields $\mathbf{pq} = (N, p(\zeta\text{-}c), q(\zeta\text{-}c), (\zeta\text{-}c)^2)$. Let $\mathbf{a} = (N, \zeta\text{-}c)$. We need to show that $\mathbf{a} = \mathbf{pq}$.

Since $N \in \mathbf{a}$, $p(\zeta\text{-}c), q(\zeta\text{-}c) \in \mathbb{Z}\mathbf{a} \subseteq \mathbf{a}$, and $(\zeta\text{-}c)^2 = (\zeta\text{-}c)(\zeta\text{-}c) \in \mathbb{O}\mathbf{a} = \mathbf{a}$, we immediately see $\mathbf{pq} \subseteq \mathbf{a}$. For the other inclusion $\mathbf{a} \subseteq \mathbf{pq}$, it suffices to prove $N \in \mathbf{pq}$ and $\zeta\text{-}c \in \mathbf{pq}$. To see that $\zeta\text{-}c \in \mathbf{pq}$, note that since $\gcd(p, q) = 1$, there exist $x, y \in \mathbb{Z}$ such that $xp + yq = 1$, hence $\zeta\text{-}c = xp(\zeta\text{-}c) + yq(\zeta\text{-}c) \in \mathbf{pq}$. ❑

Corollary 1: $\{N, \zeta\text{-}c, (\zeta\text{-}c)^2, \dots, (\zeta\text{-}c)^{\lambda\text{-}2}\}$ is a $\mathbb{Z}$-basis for $\mathbf{pq}$.

Proof:The proof is very similar to the proof of Lemma 6.4. Let $\mathbf{a} = [N, \zeta\text{-}c, (\zeta\text{-}c)^2, \dots, (\zeta\text{-}c)^{\lambda\text{-}2}]$. To show that $\mathbf{a}$ is an ideal, if suffices again to prove $\zeta\mathbf{a} \subseteq \mathbf{a}$. Let $f_\lambda(x) = \frac{x^\lambda\text{-}1}{x\text{-}1}$

and $g(x) = f_\lambda(x+c)$. As before, we can conclude that $(\zeta - c)^{\lambda-1} = \sum_{i=0}^{\lambda-2} a_i(\zeta - c)^i$ for some $a_0, \ldots, a_{\lambda-2} \in \mathbb{Z}$ and $a_0 = g(0) = f_\lambda(c) = \frac{c^\lambda - 1}{c-1}$. Now since $c \equiv a \pmod{p}$ and $c \equiv b \pmod{q}$, it follows that $a_0 \equiv 0 \pmod{p}$ and $a_0 \equiv 0 \pmod{q}$, so $a_0 \equiv 0 \pmod{N}$. Using the same reasoning as in the proof of Lemma 6.4, we conclude that $\zeta\mathfrak{a} \subseteq \mathfrak{a}$ and that $\mathfrak{a} = \mathfrak{p}\mathfrak{q}$. ❑

Corollary 2: The product ideal $\mathfrak{p}\mathfrak{q}$ can be computed from $\mathfrak{p}$ and $\mathfrak{q}$ using $O(\log N)$ arithmetic operations on inputs requiring $O(\log N)$ bits of storage.

Proof: c is computed using the Extended Euclidean Algorithm. ❑

If $\lambda \leq 19$, then $\mathbb{O}$ is a UFD by Lemma 3.7, hence we can use prime elements instead of prime ideals in our system. In this case, all ideals are principal. If $\pi \in \mathbb{O}$ is a prime divisor of a rational prime $p \equiv 1 \pmod{\lambda}$, then the ideal (π) is a prime ideal divisor of p, so $(\pi) = (p, \zeta - a)$ for some primitive λ-th root of unity $a \pmod{p}$ as in Lemma 6.2. Hence to find π, it suffices to find a generator of the principal ideal $\mathfrak{p} = (p, \zeta - a)$. We will present two methods for finding π. The first method is a modification of the algorithm for principal ideal testing described in Buchmann & Williams [BW87a], [BW87b]. The second method uses Euclidean division and thus requires that $\mathbb{O}$ be Euclidean, i.e. $\lambda \leq 11$. Assume henceforth that $\lambda \leq 19$.

Algorithm 6.1: For $p \equiv 1 \pmod{\lambda}$, $\lambda \leq 19$, find a prime $\pi \in \mathbb{O}$ such that $\pi \mid p$.

1. Find a prime ideal divisor $\mathfrak{p} = (p, \zeta - a)$ of p as in Lemma 6.2.
2. Compute the set $\mathfrak{R} = \{(\alpha_1), \ldots, (\alpha_l)\}$ of all reduced ideals in $\mathbb{O}$.
3. Compute a reduced ideal $\mathfrak{a} \sim \mathfrak{p}$ and $\alpha \in \mathbb{O}$ such that $\mathfrak{p} = (\alpha)\mathfrak{a}$.
4. Find $i \in \{1, \ldots, l\}$ such that $\mathfrak{a} = (\alpha_i)$. Set $\pi = \alpha\alpha_i$. ❑

Theorem 6.6: Algorithm 6.1 generates a prime divisor π of p using $O(l + \log p)$ arithmetic operations on inputs requiring $O(\log p)$ bits of storage.

Proof: Since $\mathbf{O} \in \mathfrak{R}$, $\mathbf{a}$ in Step 3 exists and $\mathbf{a} = (\alpha_i)$ for some $i \in \{1, \dots, l\}$. Then $\mathbf{p} = (\alpha)\mathbf{a} = (\alpha\alpha_i)$, so $\pi = \alpha\alpha_i$ must be a prime divisor of p.

By Theorem 6.3, Step 1 requires $O(\log p)$ binary operations on inputs bounded by p. From Buchmann [Bu87a], it follows that all numbers generated in Step 2 are polynomially bounded by the absolute value of the discriminant of $\mathbb{K}$, i.e. by $|\Delta| = \lambda^{\lambda-2}$, and the number of binary operations required is $O(l)$. By [BW87b], the complexity of Step 3 is given by $O(\log \|B\| \, |\Delta|)$ where B is the transformation matrix obtained by expressing a $\mathbb{Z}$-basis of $\mathbf{p}$ in terms of an integral basis of $\mathbb{K}$ and $\|B\| = \max\{|b_{ij}|\}$. Furthermore, all inputs require space $O(\log p)$. By Lemma 6.4, the numbers p, $(\zeta\text{-}a)^i$ $(1 \le i \le \lambda\text{-}2)$ form a $\mathbb{Z}$-basis of $\mathbf{p}$. If we choose the powers $(\zeta\text{-}a)^i$ $(0 \le i \le \lambda\text{-}2)$ as an integral basis of $\mathbb{K}$ as in the proof of Lemma 6.2, then $B = [b_{ij}]_{i,j\,=\,0,\,\dots\,,\,\lambda\text{-}2}$ where $b_{00} = p$, $b_{ii} = 1$ for $1 \le i \le \lambda\text{-}2$, and $b_{ij} = 0$ for $i \neq j$, hence $\|B\| = p$, and the computation time required by Step 3 is $O(\log p)$. Finally, Step 4 performs a linear search requiring $O(\log l)$ comparisons. ❑

Corollary: If c is as in Lemma 6.5, then a generator ρ of the ideal $(N, \zeta\text{-}c)$ can be found in time $O(l + \log N)$ and requiring space $O(\log N)$. ❑

In general, $l = O(R)$ where R is the regulator of $\mathbb{K}$ (Buchmann [Bu87b]), so l could be very large and Algorithm 6.1 need not be polynomial in the size of p. For $\lambda = 2$ and $\lambda = 3$, we have $l = 1$, and computations by Buchmann & Williams [BW87a], [BW87b] show that the same is true for $\lambda = 5$ and $\lambda = 7$. Consequently, the complexity of Algorithm 6.1 is $O(\log p)$ for $\lambda \le 7$. For $11 \le \lambda \le 19$, the number of reduced ideals in $\mathbf{O}$ is unknown.

There is a much simpler method for computing π. However, in order for this algorithm to be practical, we require the ring of integers $\mathbf{O}$ to be Euclidean.

Lemma 6.7: If $p \equiv 1 \pmod{\lambda}$. Any prime divisor π of p in O satisfies $\pi \simeq \gcd(p, \zeta\text{-}a)$ for some primitive λ-th root of unity $a \pmod{p}$.

Proof: Let a be any primitive λ-th root of unity (mod p). By Lemma 6.2, $p = \pi_1 \cdots \pi_{\lambda\text{-}1}$ where $(\pi_i) = (p, \zeta\text{-}a^i)$ for $1 \leq i \leq \lambda\text{-}1$. Let $i \in \{1, \dots, \lambda\text{-}1\}$be fixed. Clearly, $\pi_i \mid p$ and $\pi_i \mid \zeta\text{-}a^i$. Now suppose that δ is a divisor of both p and $\zeta\text{-}a^i$. We want to show that $\delta \mid \pi_i$. Since $\zeta \equiv a^i \pmod{\pi_i}$, we have $\pi_i \mid \zeta\text{-}a^i$ and $\pi_i \nmid \zeta\text{-}a^j$ for $j \neq i$, implying $\pi_j \nmid \zeta\text{-}a^i$ for $j \neq i$. Now $\delta \mid \zeta\text{-}a^i$, so $\pi_j \nmid \delta$ for $j \neq i$, hence $\delta \mid \dfrac{p}{\prod_{j \neq i} \pi_j} = \pi_i$. It follows that $\pi_i \simeq \gcd(p, \zeta\text{-}a^i)$. □

Corollary 1: If $\lambda \leq 11$, then an integer prime divisor π of a rational prime $p \equiv 1 \pmod{\lambda}$ can be found in $O(\log p)$ arithmetic operations and the norms of all numbers computed throughout the algorithm are bounded by a polynomial in p.

Proof : We use the Euclidean Algorithm to find π. To prove the complexity result, it suffices to show that $\log(\mathrm{N}(\zeta\text{-}a)) = O(\log p)$. To see this, note that $\mathrm{N}(\zeta\text{-}a) = \prod_{i=1}^{\lambda\text{-}1}(\zeta^i\text{-}a)$ $= \sum_{i=0}^{\lambda\text{-}1} a^i < \lambda a^{\lambda\text{-}1} < \lambda p^{\lambda\text{-}1}$, using $a < p$. □

Corollary 2: A prime divisor ρ of N can be found in time $O(\log N)$ and the norms of all numbers computed throughout the algorithm are bounded by a polynomial in N.

Proof: Find $\pi \simeq \gcd(p, \zeta\text{-}c)$, $\psi \simeq \gcd(q, \zeta\text{-}c)$, where c is as in Lemma 6.5. Set $\rho = \pi\psi$. Clearly $\rho \mid N$ and $\rho \mid \zeta\text{-}c$. Suppose $\delta \mid N$, $\delta \mid \zeta\text{-}c$. Then $\delta \mid N = \pi_1 \dots \pi_{\lambda\text{-}1} \psi_1 \dots \psi_{\lambda\text{-}1}$ where $(\pi_i) = (p , \zeta\text{-}c^i)$, $(\psi_i) = (q, \zeta\text{-}c^i)$ for $1 \leq i \leq \lambda\text{-}1$. From the proof of Lemma 6.7, we see that $\pi_i \nmid \zeta\text{-}c$, $\psi_i \nmid \zeta\text{-}c$ for $2 \leq i \leq \lambda\text{-}1$, so $\delta \mid \pi_1\psi_1 = \rho$. Hence $\rho \simeq \gcd(N, \zeta\text{-}c)$. The time and space bounds are clear from the previous corollary. □

The public key $K_\text{p} = \{r, S, N, e\}$ describes the ideal $\mathfrak{pq}$ completely by Lemma 6.5. In the case where we employ prime elements, the encrypter needs to find a generator $\rho = \pi\psi$ of $\mathfrak{pq}$ as precomputation. Alternatively, K_p could be modified to be $K_\text{p} = \{r, S, c_1, \ldots, c_{\lambda\text{-}1}, e\}$ where $\rho = \sum_{i=1}^{\lambda-1} c_i \zeta^i$.

6.2 Roots of Unity (mod N)

Our next task is to generate a primitive λ-th root of unity r (mod N), $0 < r < N$, such that $\gcd(r\text{-}1, N) = 1$ and $\left[\frac{r}{\mathfrak{pq}}\right] = 1$.

Algorithm 6.2: Given $p, q \equiv 1 \pmod{\lambda}$, $p, q \not\equiv 1 \pmod{\lambda^2}$, $\mathfrak{p} = (p, \zeta\text{-}a)$, $\mathfrak{q} = (q, \zeta\text{-}b)$, find $r \in \mathbb{Z}$ such that

i) $2 \leq r \leq N\text{-}1$, ii) $\gcd(r\text{-}1, N) = 1$,

iii) $r^\lambda \equiv 1 \pmod{N}$, iv) $\left[\frac{r}{\mathfrak{pq}}\right] = 1$.

1. Solve $qx \equiv 1 \pmod{p}$, $py \equiv 1 \pmod{q}$ for $x, y \in \mathbb{Z}$, $1 \leq x \leq p$, $1 \leq y \leq q$.
2. Set $r \equiv qxa^{\lambda\text{-}1} + pyb \pmod{N}$, $2 \leq r \leq N\text{-}1$. ❑

Theorem 6.8: Algorithm 6.2 generates r such that conditions i) - iv) hold im time $O(\log N)$ and computes only numbers requiring $O(\log N)$ bits of storage.

Proof: We have $a \equiv \zeta \pmod{\mathfrak{p}}$, so $a^{\lambda\text{-}1} \equiv \zeta^{\lambda\text{-}1} \pmod{\mathfrak{p}}$, and $b \equiv \zeta \pmod{\mathfrak{q}}$. Then $r \equiv a^{\lambda\text{-}1} \not\equiv 1 \pmod{p}$ and $r \equiv b \not\equiv 1 \pmod{q}$, hence $\gcd(r\text{-}1, N) = 1$. In particular, $r \neq 1$, so $2 \leq r \leq N\text{-}1$. Furthermore, $r^\lambda \equiv a^\lambda \equiv 1 \pmod{p}$ and $r^\lambda \equiv b^\lambda \equiv 1 \pmod{q}$, so $r^\lambda \equiv 1$ (mod N). Finally, $\left[\frac{r}{\mathfrak{pq}}\right] = \left[\frac{a}{\mathfrak{p}}\right]^{\lambda\text{-}1} \left[\frac{b}{\mathfrak{q}}\right] = \zeta^{\lambda\text{-}1}\zeta = 1$.

The congruences in Step 1 are solved using the Extended Euclidean Algorithm, so Step 1 performs $O(\log \max \{p, q\})$ arithmetic operations on numbers bounded by $\max\{p, q\}$. Step 2 can be done in constant time and all numbers are bounded by N. ❑

If we replace the ideals $\mathfrak{p}$, $\mathfrak{q}$ in Algorithm 6.2 by prime divisors π, ψ of p and q, respectively, then condition iv) changes to $\left[\frac{r}{\pi\psi}\right] = 1$. Since π and ψ were obtained from ideals $\mathfrak{p} = (p, \zeta\text{-}a)$ and $\mathfrak{q} = (q, \zeta\text{-}b)$ using Algorithm 6.1 or the Euclidean Algorithm in $\mathbb{O}$, we can use the same algorithm.

The following lemma shows that one element $r \in \mathbb{Z}$ satisfying conditions i) - iv) of Algorithm 6.2 determines the rest of them.

Lemma 6.9: Let $p, q \in \mathbb{Z}$ be primes such that $p, q \equiv 1 \pmod{\lambda}$, $p, q \not\equiv 1 \pmod{\lambda^2}$, and let r be as in Algorithm 6.2. Then $s \in \mathbb{Z}$ satisfies conditions i) - iv) in Algorithm 6.2 if and only if $s \equiv r^i \pmod{N}$, $0 < s < N$, for some $i \in \{1, \dots, \lambda\text{-}1\}$.

Proof: Clearly, if $s \equiv r^i \pmod{N}$ for some $i \in \{1, \dots, \lambda\text{-}1\}$, then $s^\lambda \equiv 1 \pmod{N}$ and $\left[\frac{s}{\mathfrak{pq}}\right] = 1$. Also, if $p \mid s\text{-}1$, then $1 \equiv s \equiv r^i \pmod{p}$ in contradiction to Lemma 6.1 (similarly for q), so $\gcd(s\text{-}1, N) = 1$.

Now suppose $s \in \mathbb{Z}$ is such that s satisfies the conditions of Algorithm 6.2. By Lemma 6.1, $s \equiv r^i \pmod{p}$, $s \equiv r^j \pmod{q}$ for some $i, j \in \{1, \dots, \lambda\text{-}1\}$. Then $1 = \left[\frac{s}{\mathfrak{pq}}\right]$ $= \left[\frac{r}{\mathfrak{p}}\right]^i \left[\frac{r}{\mathfrak{q}}\right]^j = \left[\frac{r}{\mathfrak{pq}}\right]^i \left[\frac{r}{\mathfrak{q}}\right]^{j-i} = \left[\frac{r}{\mathfrak{q}}\right]^{j-i}$. But $\left[\frac{r}{\mathfrak{q}}\right] \neq 1$ by Lemma 2.1, so we must have $\lambda \mid j\text{-}i$, and $s \equiv r^i \pmod{N}$. ❑

6.3 Residue Symbols

Let $\alpha \in \mathbb{O}$ and let $\mathfrak{a}$ be an integral ideal in $\mathbb{O}$. It is unknown if or how the λ-th residue symbol $\left[\frac{\alpha}{\mathfrak{a}}\right]$ can be evaluated without finding the prime ideal factorization of $\mathfrak{a}$. Hence we will only discuss the computation of residue symbols $\left[\frac{\alpha}{\beta}\right]$ for $\alpha, \beta \in \mathbb{O}$, $\beta \neq 0$.

Furthermore, we will assume that α and β are relatively prime, since this holds for the residue symbols $\left[\frac{S}{\pi\psi}\right]$ and $\left[\frac{M}{\pi\psi}\right]$ which need to be computed in our cryptosystem.

Our algorithm for the evaluation of λ-th residue symbols is based on the properties given in Lemma 3.5, together with the law of reciprocity plus its complementaries which were first introduced by Kummer ([Ku75], see also Smith [Sm65]). Let $\lambda \in \mathbb{Z}$ be any prime and set $\omega = 1-\zeta$. Recall that for $\gamma \in \mathbb{K}$, $\bar{\gamma} = \sigma_{\lambda-1}(\gamma)$ denotes the complex conjugate of γ.

Definition 6.10: Let $\alpha \in \mathbb{O}$. α is said to be *primary* if one of the following holds.

a) Case $\lambda = 2$: $\alpha \equiv 1 \pmod 4$.

b) ([Ku75, p. 350], [Sm65, p. 118]) Case $\lambda \geq 3$: there exists $B \in \mathbb{Z}$ such that

i) $\alpha \not\equiv 0 \pmod{\omega}$ ii) $\alpha \equiv B \pmod{\omega^2}$ iii) $\alpha\bar{\alpha} \equiv B^2 \pmod{\lambda}$. ❑

Lemma 6.11: Let $\alpha = \sum_{i=1}^{\lambda-1} a_i\zeta^i \in \mathbb{O}$ and $b = \sum_{i=1}^{\lambda-1} a_i \in \mathbb{Z}$. Then $\mathrm{Tr}(\alpha) = -b$ and $\mathrm{Tr}(\alpha\zeta^j) = a_{\lambda-j}\lambda - b$ for $j \in \{1, \dots, \lambda-1\}$, so $\mathrm{Tr}(\alpha\zeta^j) \equiv -b \pmod{\lambda}$ for $j \in \{0, \dots, \lambda-1\}$.

Proof: $\mathrm{Tr}(\alpha) = \sum_{i=1}^{\lambda-1} a_i\,\mathrm{Tr}(\zeta^i) = -\sum_{i=1}^{\lambda-1} a_i = -b$, since $\mathrm{Tr}(\zeta^i) = \sum_{j=1}^{\lambda-1} \zeta^j = -1$ for $1 \leq i \leq \lambda-1$. Let $j \in \{1, \dots, \lambda-1\}$. $\mathrm{Tr}(\alpha\zeta^j) = \sum_{i=1}^{\lambda-1} a_i\,\mathrm{Tr}(\zeta^{i+j}) = a_{\lambda-j}\,\mathrm{Tr}(1) + \sum_{i\neq\lambda-j} a_i\,\mathrm{Tr}(\zeta^{i+j}) = a_{\lambda-j}\,(\lambda-1) - \sum_{i\neq\lambda-j} a_i$

$= a_{\lambda-j}\lambda - b$. ❑

Lemma 6.12: Let $\alpha = \sum_{i=1}^{\lambda-1} a_i\zeta^i$, $b = \sum_{i=1}^{\lambda-1} a_i$, $c = \sum_{i=1}^{\lambda-1} i a_i$. Then the following holds:

a) $\alpha \equiv 0 \pmod{\omega}$ $\Leftrightarrow$ $b \equiv 0 \pmod{\lambda}$.

b) $\alpha \equiv b \pmod{\omega^2}$ $\Leftrightarrow$ $c \equiv 0 \pmod{\lambda}$.

Proof: $\alpha = \sum_{i=1}^{\lambda-1} a_i(1-\omega)^i \equiv \sum_{i=1}^{\lambda-1} a_i(1-i\omega) \equiv b - c\omega \pmod{\omega^2}$ and $\alpha \equiv b \pmod{\omega}$. Hence

$\alpha \equiv 0 \pmod{\omega} \Leftrightarrow b \equiv 0 \pmod{\omega} \Leftrightarrow \omega \mid b \Leftrightarrow \mathrm{N}(\omega) \mid \mathrm{N}(b) \Leftrightarrow \lambda \mid b^{\lambda-1} \Leftrightarrow b \equiv 0 \pmod{\lambda}$. Similarly $\alpha \equiv b \pmod{\omega^2} \Leftrightarrow c \equiv 0 \pmod{\omega} \Leftrightarrow c \equiv 0 \pmod{\lambda}$. ❑

Lemma 6.12 shows that the number B in the definition of a primary number for $\lambda \geq 3$ is $B = b = \sum_{i=1}^{\lambda-1} a_i$. Furthermore, it provides a practical method for checking conditions i) and ii) of the definition. A practical test for condition iii) depends on the field, i.e on the value of λ.

Lemma 6.13: If $\alpha, \beta \in \mathbf{O}$ are primary, then so is $\alpha\beta$.

Proof: Clear for $\lambda = 2$. Assume $\lambda \geq 3$. Then, since ω is a prime and $\alpha, \beta \not\equiv 0 \pmod{\omega}$, we have $\alpha\beta \not\equiv 0 \pmod{\omega}$. If $\alpha \equiv b(\alpha) \pmod{\omega^2}$ and $\beta \equiv b(\beta) \pmod{\omega^2}$, then $\alpha\beta \equiv b(\alpha)b(\beta) \pmod{\omega^2}$ and $(\alpha\beta)(\overline{\alpha\beta}) \equiv (\alpha\overline{\alpha})(\beta\overline{\beta}) \equiv b(\alpha)^2 b(\beta)^2 \equiv \left(b(\alpha)b(\beta)\right)^2 \pmod{\lambda}$. ❑

Lemma 6.14: Every $\alpha \in \mathbf{O}$ such that $\alpha \not\equiv 0 \pmod{\omega}$ has a primary associate. Furthermore, if $\beta, \beta' \in \mathbf{O}$ are primary associates, then $\beta = \beta'\varepsilon^\lambda$ for some unit $\varepsilon \in \mathbf{O}$.

Proof: If $\lambda = 2$, then the condition $\alpha \not\equiv 0 \pmod{\omega}$ implies that α is odd, hence either $\alpha \equiv 1 \pmod{4}$ or $-\alpha \equiv 1 \pmod{4}$. Furthermore, if $\beta = \pm\beta'$ and $\beta, \beta' \equiv 1 \pmod{4}$, then $\beta = \beta' = \beta'(\pm 1)^2$. For $\lambda \geq 3$ the lemma is proved in [Ku75 , pp.349-351]. ❑

We will give explicit algorithms for obtaining a primary associate for a given number in the cases $\lambda = 2$, 3 and 5 in Chapter 7.

Theorem 6.15 (*Kummer's Law of Reciprocity* [Ku75, pp. 345ff.], [Sm65, pp. 120f.]): Let π, ψ be two distinct primary primes in $\mathbf{O}$. Then $\left[\frac{\pi}{\psi}\right] = \left[\frac{\psi}{\pi}\right]$. ❑

Corollary: Let $\alpha, \beta \in \mathbf{O}$ be relatively prime and primary. Then $\left[\frac{\alpha}{\beta}\right] = \left[\frac{\beta}{\alpha}\right]$.

Proof: Clear from Lemma 3.5 b) and c). ❑

In addition to the law of reciprocity, Kummer gave formulae for the residue symbols of ω and of a unit over a primary prime ([Ku75, pp. 485ff.], [Sm65, pp. 121ff.]). These are called *complementaries* to the reciprocity law and can be shown to hold for composite numbers as well. The following two complemetaries can be easily proven.

Lemma 6.16: Let π be a primary prime. Then $\left[\frac{\pm 1}{\pi}\right] = 1$ and $\left[\frac{\zeta}{\pi}\right] = \zeta^k$ where $0 \leq k \leq \lambda\text{-}1$ and $k \equiv \frac{N(\pi)-1}{\lambda} \pmod{\lambda}$.

Proof: The complementary for ζ follows immediately from the definition of the residue symbol. For the complementary for ± 1, note that $N(\pi)$ is odd, hence $\frac{N(\pi)-1}{\lambda}$ is even and $(\pm 1)^{\frac{N(\pi)-1}{\lambda}} = 1$. ❑

Corollary: Let $\beta \in \mathbf{O}$ be primary. Then $\left[\frac{\pm 1}{\beta}\right] = 1$ and $\left[\frac{\zeta}{\beta}\right] = \zeta^k$ where $0 \leq k \leq \lambda\text{-}1$ and $k \equiv \frac{N(\beta)-1}{\lambda} \pmod{\lambda}$.

Proof: It suffices to show the Corollary for $\beta \simeq \pi\psi$ where π and ψ are primary primes. By Lemma 3.5 c) $\left[\frac{\pm 1}{\pi\psi}\right] = \left[\frac{\pm}{\pi}\right]\left[\frac{\pm 1}{\psi}\right] = 1$. We have $\lambda^2 \mid (N(\pi)-1)(N(\psi)-1) = N(\pi\psi) - N(\pi) -$

$N(\psi) + 1$, so $\frac{N(\pi\psi)-1}{\lambda} \equiv \frac{N(\pi)-1}{\lambda} + \frac{N(\psi)-1}{\lambda} \pmod{\lambda}$ and $\left[\frac{\zeta}{\pi\psi}\right] = \left[\frac{\zeta}{\pi}\right]\left[\frac{\zeta}{\psi}\right] = \zeta^k$ where $0 \leq k \leq \lambda-1$, $k \equiv \frac{N(\pi\psi)-1}{\lambda} \pmod{\lambda}$. ❑

The idea for our residue symbol algorithm is as follows. Let $\alpha, \beta \in O\text{-}\{0\}$ be relatively prime. By Lemma 6.14, there exists a primary associate β' of β. Then from Defintion 3.6, $\left[\frac{\alpha}{\beta'}\right] = \left[\frac{\alpha}{\beta}\right]$. Let ω^k be the exact power of ω dividing α, i.e. $\omega^k \parallel \alpha$. Again by Lemma 6.14, we can write $\alpha = \varepsilon\omega^k\gamma$ where ε is a unit, $\omega \nmid \gamma$, and γ is primary. Then by the law of reciprocity, $\left[\frac{\alpha}{\beta}\right] = \left[\frac{\alpha}{\beta'}\right] = \left[\frac{\varepsilon}{\beta'}\right]\left[\frac{\omega}{\beta'}\right]^k\left[\frac{\beta'}{\gamma}\right]$ where $\left[\frac{\varepsilon}{\beta'}\right]$ and $\left[\frac{\omega}{\beta}\right]$ can be evaluated directly, using the complementaries, and we can repeat this procedure for the residue symbol $\left[\frac{\beta'}{\gamma}\right]$. This gives rise to the following algorithm.

Algorithm 6.3: For $\alpha, \beta \in O\text{-}\{0\}$ relatively prime, compute s such that $\left[\frac{\alpha}{\beta}\right] = \zeta^s$,

$0 \leq s \leq \lambda-1$.

```
1. Set $s = 0$.
2. Find a primary associate $\beta'$ of $\beta$.
3. Find $\gamma' \in O$ such that $\left[\frac{\alpha}{\beta'}\right] = \left[\frac{\gamma'}{\beta'}\right]$ and $N(\gamma') < N(\beta')$.
4. { Eliminate factors $\omega$ }
   a) Set $k = 0$.
   b) While $Tr(\gamma') \equiv 0 \pmod{\lambda}$ do
         set $\gamma' \leftarrow \frac{\gamma'}{\omega} = (-\gamma') \cdot \frac{1}{\lambda}\sum_{i=1}^{\lambda-1} i\zeta^i$,        $k \leftarrow k + 1$.
5. { Make $\gamma'$ primary }
   Find a unit $\varepsilon$ such that $\gamma' = \varepsilon\gamma$ and $\gamma$ is primary.
```

6. Use the complementaries to find $i, j \in \{0, \ldots, \lambda\text{-}1\}$ such that $\left[\frac{\omega}{\beta'}\right] = \zeta^i, \left[\frac{\varepsilon}{\beta'}\right] = \zeta^j$.

   Set $s \leftarrow s + ki + j \pmod{\lambda}, 0 \leq s \leq \lambda\text{-}1$.

7. *If* $N(\gamma) = 1$, *then*

   use the complementaries to find $l \in \{0, \ldots, \lambda\text{-}1\}$ such that $\left[\frac{\gamma}{\beta'}\right] = \zeta^l$,

   set $s \leftarrow s + l \pmod{\lambda}, 0 \leq s \leq \lambda\text{-}1$,

   else

   set $\alpha \leftarrow \beta', \beta' \leftarrow \gamma$. *Goto* step 3. ❑

Theorem 6.17: Algorithm 6.3 terminates after a finite number of iterations and computes the residue symbol $\left[\frac{\alpha}{\beta}\right] = \zeta^s, 0 \leq s \leq \lambda\text{-}1$.

Proof: By Lemmas 6.11 and 6.12, we have $\omega^k \parallel \gamma' \Leftrightarrow \lambda^k \parallel \mathrm{Tr}(\gamma')$, so after Step 4, we have $\gamma' \not\equiv 0 \pmod{\omega}$. The primary associate γ of γ' in Step 5 exists by Lemma 6.14. Then $\left[\frac{\alpha}{\beta}\right] = \left[\frac{\omega^k}{\beta'}\right]\left[\frac{\gamma'}{\beta'}\right] = \left[\frac{\omega}{\beta'}\right]^k\left[\frac{\varepsilon}{\beta'}\right]\left[\frac{\gamma}{\beta'}\right] = \zeta^{ik+j}\left[\frac{\beta'}{\gamma}\right]$, hence we need to add $ik+j$ to s as done in Step 6, and (in the case where $N(\gamma) > 1$) replace α by β' and β' by γ as done in Step 7, after which the procedure is repeated. Now $N(\gamma') = N(\gamma) \in \mathbb{Z}^{>0}$ and since the norm strictly decreases each time Step 3 is executed, we must eventually have $N(\gamma) = 1$, so $\left[\frac{\gamma}{\beta'}\right] = \zeta^l$ can be computed from the complementaries, and adding l to s yields the final value of s. ❑

In order to compute $\left[\frac{\varepsilon}{\beta}\right]$ for a unit ε as required in Steps 6 and 7 of Algorithm 6.3, we write $\varepsilon = \pm\zeta^m\eta_1^{j_1}\cdots\eta_r^{j_r}$ where $0 \leq m \leq \lambda\text{-}1$, $j_\nu \in \mathbb{Z}$ for $1 \leq \nu \leq r$, $\{\eta_1, \ldots, \eta_r\}$ is a system of fundamental units in $\mathbb{K}$, and $r = 0$ if $\lambda = 2$, $r = \frac{\lambda\text{-}3}{2}$ if $\lambda \geq 3$. If the complementaries for the fundamental units are given by $\left[\frac{\eta_\nu}{\beta}\right] = \zeta^{e_\nu(\beta)}$, $0 \le e_\nu(\beta) \le \lambda\text{-}1$, for $\nu \in \{1, \dots, r\}$, then by Lemma 6.16 $\left[\frac{\varepsilon}{\beta}\right] = \zeta^{\frac{\mathrm{N}(\beta)-1}{\lambda}m + \sum_{i=1}^{r} e_\nu(\beta) j_\nu}$.

Theorem 6.18: If each individual step of Algorithm 6.3 can be performed in constant time, then Algorithm 6.3 performs $O(\log \mathrm{N}(\beta))$ arithmetic operations on inputs whose norms are bounded by max $\{\mathrm{N}(\alpha), \mathrm{N}(\beta)\}$.

Proof: The space bound is clear since the initial values of α and β have norm at least as large as the norms of any of the numbers subsequently computed by the algorithm.

Suppose that $\mathrm{N}(\gamma) = 1$ after h iterations of Steps 3 - 7 of the algorithm. Let the loop in step 4 be executed m_μ times in iteration μ $(1 \le \mu \le h)$. Since each individual step only requires constant time, the total number of arithmetic operations performed in h iterations is $O(h + m_1 + \cdots + m_h)$. Let $\frac{1}{Q}$ $(Q \ge 1)$ be an upper bound on the ratio $\frac{\mathrm{N}(\gamma')}{\mathrm{N}(\beta')}$ in Step 3. Since division by ω reduces the norm by a factor of λ, $\mathrm{N}(\beta')$ is reduced by a factor of at least $Q\lambda^{m_\mu}$ in the μ-th iteration of Steps 3 - 7, hence h iterations reduce $\mathrm{N}(\beta')$ by a factor of at least $Q^h\lambda^{m_1+\cdots+m_h}$. It follows that $Q^h\lambda^{m_1+\cdots+m_h} \le \mathrm{N}(\beta') = \mathrm{N}(\beta)$, so the overall complexity of Algorithm 4.3 is $O(h + m_1 + \cdots + m_h) = O(\log \mathrm{N}(\beta))$. □

If π, ψ are prime divisors of $p, q \equiv 1 \pmod{\lambda}$, respectively, as in our cryptosystem, then under the conditions of Theorem 6.18, the residue symbols $\left[\frac{S}{\pi\psi}\right]$ for a partial key S and $\left[\frac{M}{\pi\psi}\right]$ for any message M using modulus $N = pq$ can be computed in time $O(\log \mathrm{N}(\pi\psi)) = O(\log N)$, and all norms are bounded by N.

In order for Theorem 6.18 to hold, we need to find constant time procedures for Steps 2, 3, 5, 6, and 7 of Algorithm 6.3, Furthermore, it is clear that Algorithm 6.3 converges more rapidly for small ratios $\frac{\mathrm{N}(\gamma')}{\mathrm{N}(\beta')}$, hence we will also seek to maximize the value of Q whose reciprocal bounds this ratio.

6.4 Euclidean Division

Let $x \in \mathbb{K}$. Recall that Euclidean division yields $y \in \mathbb{O}$ such that $\mathrm{N}(x - y) < 1$. For $\lambda = 2$, Euclidean division reduces to division with remainder, modified to allow for negative divisors. For $x \in \mathbb{Q}$, set $y = \lfloor x \rfloor$, or alternatively, $y = \mathrm{Ne}(x)$. Here, $\lfloor x \rfloor$ denotes the largest ineger not exceeding x and $\mathrm{Ne}(x)$ denotes the integer nearest to x, i.e. $\mathrm{Ne}(x) = \lfloor x+\frac{1}{2} \rfloor$. Then $0 \le x\text{-}y < 1$ for $y = \lfloor x \rfloor$, and $|x\text{-}y| \le \frac{1}{2}$ for $y = \mathrm{Ne}(x)$.

Assume that $\lambda \ge 3$ for the remainder of this section. We will find a simple sufficient condition guaranteeing $\mathrm{N}(x\text{-}y) < 1$. Let $z = \sum_{i=1}^{\lambda-1} z_i \zeta^i \in \mathbb{K}$, $z_i \in \mathbb{Q}$ for $1 \le i \le \lambda\text{-}1$. Define $\|z\|_1 = \sum_{i=1}^{\lambda-1} z_i$ and $\|z\|_2 = \sqrt{\sum_{i=1}^{\lambda-1} z_i^2}$.

Lemma 6.19: $\mathrm{Tr}(z\bar{z}) = \lambda\|z\|_2^2 - \|z\|_1^2$.

Proof:
$$\mathrm{Tr}(z\bar{z}) = \sum_{i=1}^{\lambda-1}\sum_{j=1}^{\lambda-1} z_i z_j \mathrm{Tr}(\zeta^{i-j}) = (\lambda - 1)\sum_{i=1}^{\lambda-1} z_i^2\, \mathrm{Tr}(1)) + \sum_{i=1}^{\lambda-1}\sum_{j\neq i} z_i z_j \mathrm{Tr}(\zeta^{i-j})$$
$$= (\lambda - 1)\sum_{i=1}^{\lambda-1} z_i^2 - \sum_{i=1}^{\lambda-1}\sum_{j\neq i} z_i z_j = \lambda\|z\|_2^2 - \sum_{i=1}^{\lambda-1}\sum_{j=1}^{\lambda-1} z_i z_j = \lambda\|z\|_2^2 - \|z\|_1^2. \square$$

Corollary: $\mathrm{Tr}(z\bar{z}) \le \lambda\|z\|_2^2$. □

Lemma 6.20: $\mathrm{N}(z) \le \left(\frac{\mathrm{Tr}(z\bar{z})}{\lambda\text{-}1}\right)^{\frac{\lambda-1}{2}}$.

Proof: Using the arithmetic-geometric mean inequality, we obtain

$$\mathrm{N}(z)^2 = \left(\prod_{i=1}^{\lambda-1}\sigma_i(z)\right)^2 = \prod_{i=1}^{\lambda-1}|\sigma_i(z)|^2 \leq \left(\frac{1}{\lambda-1}\sum_{i=1}^{\lambda-1}|\sigma_i(z)|^2\right)^{\lambda-1} = \left(\frac{\mathrm{Tr}(z\bar{z})}{\lambda-1}\right)^{\lambda-1}.$$

Since $\mathrm{N}(z) \geq 0$ for all $z \in \mathbb{K}$, we can extract square roots in this inequality. □

Corollary: $\mathrm{N}(z) \leq \left(\frac{\lambda}{\lambda-1}\|z\|_2^2\right)^{\frac{\lambda-1}{2}}$. □

Hence we obtain a Euclidean division method if we can find for any $x \in \mathbb{K}$, $y \in \mathbb{O}$ such that $\|x-y\|_2 < \sqrt{\frac{\lambda-1}{\lambda}}$.

6.4.1 Direct Euclidean Division

Let $x = \sum_{i=1}^{\lambda-1} x_i\zeta^i$, $x_i \in \mathbb{Q}$ for $1 \leq i \leq \lambda-1$. An obvious approach is to set $y_i = \mathrm{Ne}(x_i)$. Then $|x_i - y_i| \leq \frac{1}{2}$, hence $\|x - y\|_2^2 \leq \frac{\lambda-1}{4}$ and by the Corollary to Lemma 6.20, we have $\mathrm{N}(x - y) \leq \left(\frac{\lambda}{4}\right)^{\frac{\lambda-1}{2}}$. Hence this will yield Euclidean division only for $\lambda = 3$, and we obtain the bound $\mathrm{N}(x - y) \leq \frac{3}{4}$.

6.4.2 Uspensky's Euclidean Division

Uspensky ([Us09], see also [La69], pp. 228-231) introduced a Euclidean division method for the case $\lambda = 5$. We will present his method for arbitrary odd λ. For $0 \leq i \leq \lambda-1$, let $A_i = \lfloor \mathrm{Tr}(x\zeta^{-i}) \rfloor$. Then it follows that $\mathrm{Tr}(x\zeta^{-i}) = A_i + \delta_i$ where $0 \leq \delta_i < 1$ for $0 \leq i \leq \lambda-1$. From Lemma 6.11, $x_i = \frac{1}{\lambda}\left(\mathrm{Tr}(x\zeta^{-i}) - \mathrm{Tr}(x)\right) = \frac{1}{\lambda}(A_i - A_0 + \delta_i - \delta_0)$ for $1 \leq i \leq \lambda-1$; hence we obtain

$$0 = \lambda\left(\sum_{i=0}^{\lambda-1} x_i + \mathrm{Tr}(x)\right) = \sum_{i=0}^{\lambda-1}\left(\lambda x_i + \mathrm{Tr}(x)\right) = \sum_{i=0}^{\lambda-1}\mathrm{Tr}(x\zeta^{-i}) = \sum_{i=0}^{\lambda-1} A_i + \sum_{i=0}^{\lambda-1}\delta_i,$$

so $\sum_{i=0}^{\lambda-1}\delta_i \in \{0, \dots, \lambda-1\}$.

If $\delta_i = 0$ for all $i \in \{0, \dots, \lambda-1\}$, then set $y_i = \mathrm{Ne}(x_i)$ and $y = \sum_{i=1}^{\lambda-1} y_i\zeta^i \in \mathbb{O}$. Then the fact that λ is odd, together with $\lambda x_i = A_i - A_0 \in \mathbb{Z}$ and $|x_i - y_i| \le \frac{1}{2}$ implies $\lambda|x_i - y_i| \le \frac{\lambda-1}{2}$ or $|x_i - y_i| \le \frac{\lambda-1}{2\lambda}$ for $1 \le i \le \lambda-1$, hence in this case by the Corollary to Lemma 6.20:

$$\mathrm{N}(x\text{-}y) \le \left(\frac{(\lambda-1)^2}{4\lambda}\right)^{\frac{\lambda-1}{2}},$$

yielding the bounds $\mathrm{N}(x\text{-}y) \le \frac{1}{3}$ for $\lambda = 3$, $\mathrm{N}(x\text{-}y) \le \frac{16}{25}$ for $\lambda = 5$, and a bound exceeding 1 for $\lambda \ge 7$.

If $\delta_k > 0$ for some $k \in \{0, \dots, \lambda-1\}$, then set $n_i \equiv A_i - A_0 \pmod{\lambda}$ such that $|n_i| \le \frac{\lambda-1}{2}$ for $1 \le i \le \lambda-1$. Then $0 = \sum_{i=0}^{\lambda-1} A_i + \sum_{i=0}^{\lambda-1}\delta_i \equiv \sum_{i=1}^{\lambda-1}(A_i - A_0) + \sum_{i=0}^{\lambda-1}\delta_i \equiv \sum_{i=1}^{\lambda-1} n_i + \sum_{i=0}^{\lambda-1}\delta_i \pmod{\lambda}$.

Suppose $n_j = 0$ for some $j \in \{0, \dots, \lambda-2\}$. Then set again $y_i = \mathrm{Ne}(x_i)$ $(1 \le i \le \lambda-1)$ and $y = \sum_{i=1}^{\lambda-1} y_i\zeta^i$. Clearly $|x_i - y_i| \le \frac{1}{2}$ for $1 \le i \le \lambda-1$. Furthermore, since $n_j = 0$, we have $\frac{1}{\lambda}(A_j\text{-}A_0) \in \mathbb{Z}$ and $\left|\frac{1}{\lambda}(\delta_j\text{-}\delta_0)\right| < \frac{1}{\lambda} < \frac{1}{2}$, hence $y_j = \frac{1}{\lambda}(A_j\text{-}A_0)$ and $|x_j - y_j| = \frac{1}{\lambda}(\delta_j\text{-}\delta_0) < \frac{1}{\lambda}$.

Now suppose that $n_i \ne 0$ for all $i \in \{1, \dots, \lambda-1\}$. Then we must have $n_j = n_k$ for some $j, k \in \{1, \dots, \lambda-1\}$, $j < k$, for otherwise $\{n_1, \dots, n_{\lambda-1}\} = \left\{-\frac{\lambda-1}{2}, -\frac{\lambda-3}{2}, \dots, \frac{\lambda-1}{2}\right\}$ and $0 \equiv \sum_{i=1}^{\lambda-1} n_i + \sum_{i=0}^{\lambda-1}\delta_i \equiv \sum_{i=0}^{\lambda-1}\delta_i \pmod{\lambda}$, contradicting $0 < \sum_{i=0}^{\lambda-1}\delta_i < \lambda$. In this case, consider

$$\begin{aligned} X &= x\zeta^{-j} = \sum_{i=1}^{\lambda-1} x_i\zeta^{i-j} = \sum_{i=1}^{j-1} x_i\zeta^{i-j+\lambda} + x_j + \sum_{i=j+1}^{\lambda-1} x_i\zeta^{i-j} = \sum_{l=\lambda+1-j}^{\lambda-1} x_{j+l-\lambda}\zeta^l - x_j\sum_{l=1}^{\lambda-1}\zeta^l + \sum_{l=1}^{\lambda-1-j} x_{l+j}\zeta^l \\ &= \sum_{l=1}^{\lambda-1-j}(x_{l+j} - x_j)\zeta^l - x_j\zeta^{\lambda-j} + \sum_{l=\lambda+1-j}^{\lambda-1}(x_{j+l-\lambda} - x_j)\zeta^l = \sum_{l=1}^{\lambda-1} X_l\zeta^l, \end{aligned}$$

and set $Y_l = \mathrm{Ne}(X_l)$ for $1 \le l \le \lambda\text{-}1$, $Y = \sum_{l=1}^{\lambda-1} Y_l \zeta^l$, $y = \zeta^j Y$. Again we have $|X_l - Y_l| \le \frac{1}{2}$ and since $n_j = n_k$, we have $\frac{1}{\lambda}(A_j - A_k) \in \mathbb{Z}$ and $\left|\frac{1}{\lambda}(\delta_j - \delta_k)\right| < \frac{1}{\lambda} < \frac{1}{2}$. Since $1 \le k\text{-}j \le \lambda\text{-}1\text{-}j$, it follows that $X_{k-j} = x_k - x_j = \frac{1}{\lambda}(A_k - A_j) + \frac{1}{\lambda}(\delta_k - \delta_j)$, so $Y_{k-j} = \frac{1}{\lambda}(A_k - A_j)$ and $|X_{k-j} - Y_{k-j}| < \frac{1}{\lambda}$. Hence $x - y = \zeta^j(X - Y)$ has its k-th coefficient bounded in absolute value by $\frac{1}{\lambda}$ and all other coefficients by $\frac{1}{2}$.

From the Corollary to Lemma 6.20, we obtain for the case $\delta_k > 0$ for some k:

$$\mathrm{N}(x - y) < \left[\frac{\lambda}{\lambda-1}\left(\frac{\lambda-2}{4} + \frac{1}{\lambda^2}\right)\right]^{\frac{\lambda-1}{2}},$$

which yields the following bounds:

$\lambda = 3$: $\mathrm{N}(x - y) < \frac{13}{24}$. Computing the norm directly yields a slightly better bound of

$$\mathrm{N}(x - y) = (x_1 - y_1)^2 - (x_1 - y_1)(x_2 - y_2) + (x_2 - y_2)^2 < \frac{1}{2^2} + \frac{1}{2\cdot 3} + \frac{1}{3^2} = \frac{19}{36}.$$

$\lambda = 5$: $\mathrm{N}(x - y) < \left(\frac{79}{80}\right)^2 = \frac{6241}{6400}$.

For $\lambda \ge 7$, we obtain a bound exceeding 1, so this algorithm will not yield Euclidean division.

6.4.3 Kummer's Euclidean Division

An algorithm given by Kummer ([Ku75, p. 87]), details given by [Le79]) for the case $\lambda = 5$ slightly improves Uspensky's bounds. It uses a slightly stronger version of Lemma 6.19. For reasons of symmetry, we will represent field elements as linear combinations of all λ roots of unity $1, \zeta, \dots, \zeta^{\lambda-1}$ (note that this representation is not unique anymore). Let $z = \sum_{i=0}^{\lambda-1} z_i \zeta^i \in \mathrm{K}$, $z_i \in \mathbb{Q}$ for $0 \le i \le \lambda\text{-}1$.

Lemma 6.21: Let $c \in \mathbb{R}$. Then $\mathrm{Tr}(z\bar{z}) = \lambda \sum_{i=0}^{\lambda-1} (z_i - c)^2 - \left(\sum_{i=0}^{\lambda-1} (z_i - c)\right)^2$.

Proof: $\lambda\sum_{i=0}^{\lambda-1}(z_i - c)^2 - \left(\sum_{i=0}^{\lambda-1}(z_i - c)\right)^2 = \lambda\sum_{i=0}^{\lambda-1}z_i^2 - 2\lambda c\sum_{i=0}^{\lambda-1}z_i + (\lambda c)^2 - \left(\sum_{i=0}^{\lambda-1}z_i - \lambda c\right)^2$ $= \lambda\sum_{i=0}^{\lambda-1}z_i^2 - \left(\sum_{i=0}^{\lambda-1}z_i\right)^2 = \mathrm{Tr}(z\bar{z})$, where the last equality follows using the same reasoning as in the proof of Lemma 6.19. ❑

Using Lemma 6.20, we obtain the following

Corollary: Let $c \in \mathbb{R}$. Then $\mathrm{N}(z) \leq \left(\frac{\lambda}{\lambda-1}\sum_{i=0}^{\lambda-1}(z_i - c)^2\right)^{\frac{\lambda-1}{2}}$. ❑

Now let $x = \sum_{i=0}^{\lambda-1}x_i\zeta^i \in \mathbb{K}$, $x_i \in \mathbb{Q}$ for $0 \leq i \leq \lambda-1$. For $0 \leq i \leq \lambda-1$, set $y_i = \lfloor x_i \rfloor$. Then $y = \sum_{i=0}^{\lambda-1}y_i\zeta^i \in \mathbb{O}$ and if $z = x - y = \sum_{i=0}^{\lambda-1}z_i\zeta^i$, then $0 \leq z_i < 1$ for $i \in \{0, \ldots, \lambda-1\}$. Let $z_{\max} = \max\{z_i \mid 0 \leq i \leq \lambda-1\}$ and $z_{\min} = \min\{z_i \mid 0 \leq i \leq \lambda-1\}$. If $z_{\max} - z_{\min} \leq \frac{\lambda-1}{\lambda}$, then there exist $j, k \in \{0, \ldots, \lambda-1\}$ such that $|z_j - z_k| \leq \frac{1}{\lambda}$. Otherwise let i be such that $z_i = z_{\max}$. Replace z_i by z_i-1, i.e. replace z by z-ζ^i and y by y+$\zeta^i \in \mathbb{O}$. Then since $\frac{\lambda-1}{\lambda} < z_{\max} - z_{\min} < 1$, we have $-\frac{1}{\lambda} < (z_{\max} - 1) - z_{\min} < 0$, so again we have found $j, k \in \{0, \ldots, \lambda-1\}$ such that $|z_j - z_k| \leq \frac{1}{\lambda}$. Now let $c = \frac{z_j + z_k}{2}$. Then $|z_j - c| = |z_k - c| = \frac{1}{2}|z_j - z_k| \leq \frac{1}{2\lambda}$ and for $i \notin \{j, k\}$, we have $|z_i - c| \leq \frac{1}{2}(|z_i - z_j| + |z_i - z_k|) < 1$. If $|z_i - c| \geq \frac{1}{2}$, then replacing z_i by one of z_i1 or z_i+1, i.e. again replacing z by one of z-ζ^i or z+ζ^i and y by one of y+ζ^i or y-ζ^i will achieve $|z_i - c| \leq \frac{1}{2}$. From the Corollary to Lemma 6.20, we get

$$\mathrm{N}(x - y) \leq \left[\frac{\lambda}{\lambda-1}\left(\frac{\lambda-2}{4} + \frac{1}{2\lambda^2}\right)\right]^{\frac{\lambda-1}{2}}$$

which yields $N(x - y) \leq \frac{11}{24}$ for $\lambda = 3$, $N(x - y) \leq \left(\frac{77}{80}\right)^2 = \frac{5929}{6400}$ for $\lambda = 5$, and again a bound larger than 1 for $\lambda \geq 7$.

6.4.4 Lenstra's Euclidean Division

In [Le75], Lenstra proves that if ζ is a primitive m-th root of unity ($m \in \mathbb{Z}_+$) such that $m \neq 16$, $m \neq 24$, and $\varphi(m) \leq 10$, then $\mathbb{K} = \mathbb{Q}(\zeta)$ is Euclidean for the norm. Since his results can be used as the basis for a Euclidean division algorithm, we will repeat some of the ideas here.

Let $\mathbb{K}$ be an algebraic number field of degree n over $\mathbb{Q}$. We define $\mathbb{K}_\mathbb{R}$ to be the $\mathbb{R}$-*algebra* $\mathbb{K}\otimes_\mathbb{Q}\mathbb{R}$, i.e. if $\alpha_1, \ldots, \alpha_n$ is a $\mathbb{Q}$-basis of $\mathbb{K}$, then $\mathbb{K}_\mathbb{R} = \left\{ \sum_{i=1}^{n} a_i\alpha_i \mid a_i \in \mathbb{R} \text{ for } 1 \leq i \leq n \right\}$ is an integral domain and a vector space of dimension n over $\mathbb{R}$. All conjugate mappings $\sigma_i:\mathbb{K} \rightarrow \mathbb{C}$ have canonical extensions $\sigma_i:\mathbb{K}_\mathbb{R} \rightarrow \mathbb{C}$ $(1 \leq i \leq n)$. For $x \in \mathbb{K}_\mathbb{R}$, define $\mu(x) = \sum_{i=1}^{n} |\sigma_i(x)|^2$. Then $\mu:\mathbb{K}_\mathbb{R} \rightarrow \mathbb{R}$ is a positive quadratic form on the $\mathbb{R}$-vector space $\mathbb{K}_\mathbb{R}$. The *fundamental domain* $\mathbb{F}_\mathbb{R}$ with respect to $\mathbb{O}$ is $\mathbb{F}_\mathbb{R} = \{x \in \mathbb{K}_\mathbb{R} \mid \mu(x) \leq \mu(x-y) \text{ for all } y \in \mathbb{O}\}$.

Lemma 6.22: $\mathbb{F}_\mathbb{R} + \mathbb{O} = \mathbb{K}_\mathbb{R}$.

Proof Lenstra [Le92]: We only need to prove $\mathbb{K}_\mathbb{R} \subseteq \mathbb{F}_\mathbb{R} + \mathbb{O}$. It is possible to choose an $\mathbb{R}$-basis $\beta_1, \ldots, \beta_n$ of $\mathbb{K}_\mathbb{R}$ such that if $x = \sum_{i=1}^{n} a_i\beta_i \in \mathbb{K}_\mathbb{R}$ ($a_i \in \mathbb{R}$ for $1 \leq i \leq n$), then $\mu(x) = \sum_{i=1}^{n} a_i^2$. Set $\|x\| = \sqrt{\sum_{i=1}^{n} a_i^2}$. Then $\mu(x) = \|x\|^2$ is the square of the Euclidean norm of x with respect to the basis $\beta_1, \ldots, \beta_n$. Now $x \in \mathbb{F}_\mathbb{R}$ if and only if $\|x\| \leq \|x - y\|$ for all $y \in \mathbb{O}$, i.e. if and only if x is at least as close to 0 with respect to $\|\cdot\|$ as to any other point in $\mathbb{O}$.

Let $x \in \mathbb{K}_{\mathbb{R}}$. Since $\mathbb{O}$ is a lattice in $\mathbb{K}_{\mathbb{R}}$, there exists a point $y \in \mathbb{O}$ which is closest to x, i.e. $\|x - y\|$ is minimal. Then $x - y$ has 0 as its closest lattice point, so $x - y \in \mathbb{F}_{\mathbb{R}}$. Hence if we set $z = x - y$, then $x = z + y$ where $z \in \mathbb{F}_{\mathbb{R}}$ and $y \in \mathbb{O}$. ❑

Corollary: If $\mathbb{F} = \mathbb{F}_{\mathbb{R}} \cap \mathbb{K} = \{x \in \mathbb{K} \mid \mu(x) \leq \mu(x\text{-}y) \text{ for all } y \in \mathbb{O}\}$, then $\mathbb{F} + \mathbb{O} = \mathbb{K}$. ❑

Lemma 6.23: Let $x \in \mathbb{K}$ and let $y \in \mathbb{O}$ such that $\mu(x - y)$ is minimal. Then $x - y \in \mathbb{F}$.

Proof: We need to show that $\mu(x - y) \leq \mu(x - y + u)$ for all $u \in \mathbb{O}$. Let $u \in \mathbb{O}$ and let $v = y - u \in \mathbb{O}$. Since $\mu(x - y)$ is minimal, we have $\mu(x - y) \leq \mu(x - v) = \mu(x - y + u)$. ❑

Now let $\mathbb{K}$ be a cyclotomic field generated by a m-th primitive root of unity ($m \in \mathbb{Z}^{\geq 3}$). Then $\mu(z) = \mathrm{Tr}(z\bar{z})$ for any $z \in \mathbb{K}$. Let $x \in \mathbb{K}$. If we can choose $y \in \mathbb{O}$ such that $\mu(x - y)$ is sufficiently small, then Lemma 6.20 provides an upper bound on $\mathrm{N}(x - y)$, and the previous lemma shows that ring elements u with $z = x - u \in \mathbb{F}$ are possible candidates for y. Let $m = \lambda$ be a prime. Lenstra shows in [Le75] that in this case $\mu(z) \leq \frac{\lambda^2\text{-}1}{12}$ for all $z \in \mathbb{F}$. From Lemma 6.20, we obtain $\mathrm{N}(z) \leq \left(\frac{\mu(z)}{\lambda\text{-}1}\right)^{\frac{\lambda\text{-}1}{2}} \leq \left(\frac{\lambda+1}{12}\right)^{\frac{\lambda\text{-}1}{2}}$ for $z \in \mathbb{F}$, yielding the following values for a bound Q on $\mathrm{N}(x - y)$: $Q = \frac{1}{3}$ for $\lambda = 3$, $Q = \frac{1}{4}$ for $\lambda = 5$, $Q = \frac{8}{27}$ for $\lambda = 7$, and finally, $Q = 1$ for $\lambda = 11$.

Lenstra does not provide an explicit algorithm for generating for $x \in \mathbb{K}$ a ring element $y \in \mathbb{O}$ such that $\mu(x - y)$ is minimal. We will show that there are only λ possible candidates for y, all of which can be easily computed from x.

Lemma 6.24: Let $a_1, \ldots, a_n \in \mathbb{R}$, $a_1 \leq \cdots \leq a_n$. Let $b_i = a_i$ for $2 \leq i \leq n\text{-}1$, $b_1 = a_1 + 1$, $b_n = a_n - 1$. If $a_n - a_1 > 1$, then $\sum_{i=1}^{n} b_i^2 < \sum_{i=1}^{n} a_i^2$.

Proof: $\sum_{i=1}^{n} b_i^2 = (a_1+1)^2 + \sum_{i=2}^{n-1} a_i^2 + (a_n\text{-}1)^2 = \sum_{i=1}^{n} a_i^2 + 2(a_1 - a_n + 1) < \sum_{i=1}^{n} a_i^2$. ❑

Lemma 6.25: Let $x \in \mathbb{K}$ and let $y \in \mathbb{O}$ such that $\mu(x - y)$ is minimal. Let $z = x - y = \sum_{i=0}^{\lambda-1} z_i\zeta^{\mu_i}$ where $z_0 \leq \cdots \leq z_{\lambda-1}$. Then $z_{\lambda-1} - z_0 \leq 1$.

Proof: Set $y' = y - \zeta^{\mu_0} + \zeta^{\mu_{\lambda-1}} \in \mathbb{O}$, $z' = z + \zeta^{\mu_0} - \zeta^{\mu_{\lambda-1}} = x - y'$, i.e $z' = \sum_{i=0}^{\lambda-1} z_i'\zeta^{\mu_i}$ where $z_i' = z_i$ for $1 \leq i \leq \lambda\text{-}2$, $z_0' = z_0 + 1$, $z_{\lambda-1}' = z_{\lambda-1} - 1$. Suppose $z_{\lambda-1} - z_0 > 1$. Then $\sum_{i=0}^{\lambda-1} z_i' = \sum_{i=0}^{\lambda-1} z_i$ and by Lemma 6.24 $\sum_{i=0}^{\lambda-1} z_i'^2 > \sum_{i=0}^{\lambda-1} z_i^2$. Using Lemma 6.21 with $c = 0$, we obtain $\mu(z') < \mu(z)$, contradicting the minimality of $\mu(z)$. ❑

Theorem 6.26: Let $x = \sum_{i=0}^{\lambda-1} x_i\zeta^i \in \mathbb{K}$, $z_i' = x_i - \lfloor x_i \rfloor$ for $0 \leq i \leq \lambda\text{-}1$. Let $z_i = z_{\mu_i}$ where $0 \leq z_0 \leq \cdots \leq z_{\lambda-1} < 1$. Set $z^{(0)} = \sum_{i=0}^{\lambda-1} z_i'\zeta^i = \sum_{i=0}^{\lambda-1} z_i\zeta^{\mu_i}$, $z^{(j)} = z^{(j-1)} + \zeta^{\mu_j}$ for $1 \leq j \leq \lambda\text{-}1$. If $y \in \mathbb{O}$ is such that $\mu(x - y)$ is minimal, then $x - y = z^{(k)}$ for some $k \in \{0, \ldots, \lambda\text{-}1\}$.

Proof: Let $y \in \mathbb{O}$ be such that $\mu(w)$ is minimal where $w = x - y$. Set $u = \sum_{i=0}^{\lambda-1} \lfloor x_i \rfloor \zeta^i \in \mathbb{O}$, then $x = u + z^{(0)} = y + w$, so $w - z^{(0)} \in \mathbb{O}$. Hence if $w = \sum_{i=0}^{\lambda-1} w_i\zeta^{\mu_i}$, then $w_i - z_i = n_i \in \mathbb{Z}$ for all $i \in \{0, \ldots, \lambda\text{-}1\}$. If $z_i = z_j$ and $n_i < n_j$ for some $i < j$, then swap i and j to obtain $n_i \geq n_j$. This does not change the order of the z_ν $(0 \leq \nu \leq \lambda\text{-}1)$.

Let $0 \leq i < j \leq \lambda\text{-}1$. Then $0 \leq z_j - z_i < 1$, hence $w_i - w_j \leq n_j - n_i < 1 + w_i - w_j$. Now since $\mu(w)$ is minimal, by Lemma 6.25, we must have $|w_i - w_j| \leq 1$, so $-1 \leq n_j - n_i < 2$ or $n_j - n_i \in \{-1, 0, 1\}$. Suppose $n_j > n_i$, then $n_j - n_i = 1$ and $z_i < z_j$ by our renumbering of the z_ν hence $w_j = n_j + z_j > n_i + 1 + z_i = w_i + 1$ in contradiction to $|w_i - w_j| \leq 1$. Therefore $n_j \leq n_i$. It follows that $n_j - n_i \in \{0, -1\}$ and $0 \geq n_1\text{-}n_0 \geq \cdots \geq n_{\lambda-1}\text{-}n_0 \geq -1$. Define k such that $n_i - n_0 = 0$ for $i < k$ and $n_i - n_0 = -1$ for $i \geq k$. Then $w_i = z_i + n_0$ for $i < k$ and $w_i = z_i + n_0 - 1$ for $i \geq k$. Hence

$$w = \sum_{i=0}^{k-1}(w_i+n_0)\zeta^{\mu_i} + \sum_{i=k}^{\lambda-1}(w_i+n_0-1)\zeta^{\mu_i} = \sum_{i=0}^{k-1}(w_i+1)\zeta^{\mu_i} + \sum_{i=k}^{\lambda-1}w_i\zeta^{\mu_i} + (n_0-1)\sum_{i=0}^{\lambda-1}\zeta^{\mu_i}$$

$$= z^{(0)} + \sum_{i=0}^{k-1}\zeta^{\mu_i} = z^{(k)}. \quad \square$$

Corollary: $z^{(k)} \in \mathbb{F}$ for $0 \leq k \leq \lambda\text{-}1$, hence $\mu(x - y) \leq \frac{\lambda^2\text{-}1}{12}$. □

The previous theorem and its corollary give rise to the following Euclidean division algorithm.

Algorithm 6.4: Given $x = \sum_{i=0}^{\lambda-1} x_i\zeta^i \in \mathbb{K}$, find $y \in \mathbb{O}$ such that $x - y \in \mathbb{F}$.

```
1. For $0 \leq i \leq \lambda-1$ set $y_i = \lfloor x_i \rfloor$ and $z_i' = x_i - y_i$. Set $z = \sum_{i=0}^{\lambda-1} z_i'\zeta^i$, $y = \sum_{i=0}^{\lambda-1} y_i\zeta^i$.
2. Sort the $z_i'$ in non-descending order, i.e. let $z_i = z_{\mu_i}'$ and $0 \leq z_0 \leq z_1 \leq \cdots \leq z_{\lambda-1}$.
3. While $\mu(z) > \frac{\lambda^2-1}{12}$ do
        set $y \leftarrow y - \zeta^{\mu_0}$, $z \leftarrow z + \zeta^{\mu_0}$.
        sort the $z_i$ in non-descending order, i.e. set
            $t = z_0, z_0 = z_1, \ldots z_{\lambda-2} = z_{\lambda-1}, z_{\lambda-1} = t + 1$. □
```

Clearly $y \in \mathbb{O}$ in each step. By Theorem 6.26 and its Corollary, the algorithm terminates after at most λ iterations of step 3, after which we will have added $\sum_{i=0}^{\lambda-1}\zeta^i = 0$ to the value of z in step 1. Hence this algorithm produces $y \in \mathbb{O}$ such that $\mathrm{N}(x - y) < 1$ in constant time.

6.4.5 Summary

We summarize the bounds obtained for the four Euclidean division algorithms given in the previous four subsections in the following table.

Algorithm	$\lambda = 3$	$\lambda = 5$	$\lambda = 7$	$\lambda = 11$
Direct	$\frac{3}{4}$	–	–	–
Uspensky	$\frac{19}{36}$	$\left(\frac{79}{80}\right)^2$	–	–
Kummer	$\frac{11}{24}$	$\left(\frac{77}{80}\right)^2$	–	–
Lenstra	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{8}{27}$	1

7. The cases $\lambda = 2, 3, 5$

As mentioned in Section 2.3, Williams described a quadratic and a cubic scheme [Wi80], [Wi86]. Both schemes are slightly more restrictive than our system, since they require $p \equiv 1+\lambda \pmod{\lambda^2}$, $q \equiv 1-\lambda \pmod{\lambda^2}$ instead of the weaker conditions imposed on p and q by our scheme. Then $f \equiv -1 \pmod{\lambda}$, hence $\frac{f+1}{\lambda} \in \mathbb{Z}$. Let $e \in \mathbb{Z}$ be as in Chapter 2, and let $d \in \mathbb{Z}$, $0 < d < N$ satisfy $ed \equiv \frac{f+1}{\lambda} \pmod{\phi(N)}$. Again, we let $\pi, \psi \in \mathbb{O}$ be primes such that $\pi \mid p$, $\psi \mid q$. Finally, let $r \in \mathbb{Z}$ such that $\frac{r^\lambda - 1}{r - 1} \equiv 0 \pmod{N}$. For $\lambda = 2$, we have $r = N-1$ and by Lemma 2.4 c) $\left(\frac{r}{N}\right) = (-1)^{\frac{p-1}{2} + \frac{q-1}{2}} = 1$. For $\lambda = 3$, we will see in Section 7.2 that it is easy to compute $r \in \mathbb{Z}$ such that $r \equiv \zeta \pmod{\pi\psi}$. Then $\left[\frac{r}{\pi\psi}\right] = \zeta^{\frac{p-1}{3} + \frac{q-1}{3}} = 1$.

The encryption and decryption algorithm are based on a slightly different consequence of Theorem 4.2.

Corollary to Theorem 4.2: Let $X \in \mathbb{Z}$, $\gcd(X, N) = 1$, and $\left[\frac{X}{\pi\psi}\right] = 1$. If $Z \equiv X^{\lambda e} \pmod{N}$, then $Z^d \equiv r^k X \pmod{N}$ for some $k \in \{0, \dots, \lambda-1\}$.

Proof: Let $ed = \frac{f+1}{\lambda} + l\phi(N)$, $l \in \mathbb{Z}$. By Theorem 4.2, $X^f \equiv r^k \pmod{N}$ for some $k \in \{0, \dots, \lambda-1\}$ and by Euler's Theorem $X^{\phi(N)} \equiv 1 \pmod{N}$. Then $Z^d \equiv X^{\lambda ed} \equiv X^{f+1+\lambda l\phi(N)} = r^k X \pmod{N}$. ❑

7.1 The case $\lambda = 2$

This case is a modification of the generalized version of Williams' quadratic system [Wi80] described in [Wi86]. Here we have $p \equiv q \equiv 3 \pmod{4}$, $\pi = p$, $\psi = q$, $r = N - 1$, and the residue symbol reduces to the Jacobi symbol described in Definition 2.3. To generate a key, we need to find a quadratic non-residue S of N such that $0 < S < N$. The public key is then $K_p = \{S, N, e\}$ and its size is bounded by $3\log(N)$ at worst and $2\log(N)$ if S is small, in which case K_p has the same size as an RSA public key. If we specify $p \equiv 3 \pmod 8$, $q \equiv 7 \pmod 8$ as done in [Wi80], then the key generator can always use $S = 2$, since $\left(\frac{2}{pq}\right) = -1$. In this case, S need not be specified as part of the public key, so K_p has the same form as an RSA key.

To encrypt a message M, $0 < M < N$, we compute $\left(\frac{M}{S}\right) = \pm 1$. Then $M_0 = M$ if the plus sign holds, $M_0 \equiv MS \pmod N$, $0 < M_0 < N$, if the minus sign holds, and $M_1 = N - M_0$. The bit n determines if M_0 is the larger or the smaller of the two and could alternatively give the parity of M_0 as pointed out in [Wi86] (note that M_0 is even if and only if M_1 is odd).

Recall that p is said to be primary if $p \equiv 1 \pmod 4$. In this case, we have the quadratic law of reciprocity $\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)$ and the well-known quadratic complementaries $\left(\frac{\pm 1}{p}\right) = 1$ and $\left(\frac{2}{p}\right) = (-1)^{\frac{p-1}{4}}$, so Algorithm 6.3 reduces to the following standard algorithm for computing Legendre symbols.

Algorithm 7.1: For $\alpha, \beta \in \mathbb{Z}$, β odd, $\gcd(\alpha, \beta) = 1$, compute $\left(\frac{\alpha}{\beta}\right) = \delta = \pm 1$.

```
1. Set $\delta = 1$.
2. Set $\beta \leftarrow -\beta$ if $\beta \equiv 3 \pmod 4$.
3. Use division with remainder to find $\gamma \in \mathbb{Z}$ such that $\alpha \equiv \gamma \pmod \beta$ and $\gamma \leq |\beta|$
4. { Eliminate factors of 2 }
   a) Set $i \leftarrow 0$.
   b) While $\gamma$ is even do
          Set $\gamma \leftarrow \frac{\gamma}{2}$, $i \leftarrow i + 1$.
   c) If $i$ is odd and $\beta \equiv 5 \pmod 8$ then
          Set $\delta \leftarrow -\delta$
```

5. *If* $\gamma = 1$ then

Set $\alpha \leftarrow \beta$, $\beta \leftarrow \gamma$. *Goto* step 2. ❑

The algorithm terminates when $\gamma = \pm 1$. Step 2 insures that β is primary. For Step 4 c), note that $\beta \equiv 5 \pmod 8$ if and only if $\frac{\beta-1}{4}$ is odd, in which case $\left(\frac{2}{\beta}\right) = -1$. Since each step of Algorithm 7.1 requires only constant time, the complexity of this algorithm is $O(\log |\beta|)$ by Theorem 6.18. For a more detailed worst-case analysis of this and two other algorithms for computing Jacobi symbols see Shallit [Sh90].

7.2 The case $\lambda = 3$

7.2.1 Modification of Williams' Scheme

A modified version of this case is discussed in [Wi86]. The cubic scheme is more complicated than the quadratic case, since the cryptosystem requires arithmetic in the quadratic field $\mathbb{K} = \mathbb{Q}(\zeta) = \mathbb{Q}(\sqrt{-3})$ generated by a cube root of unity $\zeta = -\frac{1}{2} \pm \frac{1}{2}\sqrt{-3}$. However, since the unit rank of $\mathbb{K}$ is $\frac{3-3}{2} = 0$, the algorithms are much simpler than for higher order cases. We have $p \equiv q \equiv 1 \pmod 3$, and $p, q \not\equiv 1 \pmod 9$, hence $p, q \equiv 4, 7 \pmod 9$. Let $\pi = a_1\zeta + a_2\zeta^2$, $\psi = b_1\zeta + b_2\zeta^2$ be prime divisors in $\mathbb{O}$ of p and q, respectively. Then $N = c_1{}^2 - c_1c_2 + c_2{}^2$ where $\pi\psi = c_1\zeta + c_2\zeta^2$.

It is possible to find a primitive cube root of unity (mod N) such that $\gcd(r-1, N) = 1$ and $\left[\frac{r}{\pi\psi}\right] = 1$ directly from π and ψ without using Algorithm 6.2. Hence r need not be specified in the public key, and only needs to be computed once for each pair of keys K_p, K_s (recall that the same is true for $S^{-1} \pmod N$).

Algorithm 7.2: From $\pi = a_1\zeta + a_2\zeta^2$, $\psi = b_1\zeta + b_2\zeta^2$, compute r such that $2 \le r \le N$, $\gcd(r-1, N) = 1$, $r^3 \equiv 1 \pmod N$, and $\left[\frac{r}{\pi\psi}\right] = 1$.

```
If $p \equiv q \pmod 9$, then
        solve $r \equiv -a_1a_2^{-1} \pmod p$, $r \equiv -b_2b_1^{-1} \pmod q$, $2 \le r \le N-1$,
else
        compute $c_1 = a_2b_2 - a_2b_1 - a_1b_2$, $c_2 = a_1b_1 - a_2b_1 - a_1b_2$.
        Set $r \equiv -c_1c_2^{-1} \pmod N$, $2 \le r \le N-1$. ❑
```

Lemma 7.1: Algorithm 7.2 computes r satisfying all the conditions specified in the Algorithm in time $O(\log N)$, with all inputs bounded by N.

Proof: The complexity specifications are clear since all the inverses (mod p), (mod q), and (mod N) are computed using the Extended Euclidean Algorithm, and r can then be found in constant time in the case where $p \not\equiv q \pmod 9$, and in time $O(\log N)$ using the Chinese Remainder Theorem if $p \equiv q \pmod 9$.

A short calculation shows that $p = a_1^2 - a_1a_2 + a_2^2$, $q = b_1^2 - b_1b_2 + b_2^2$, $\pi\psi = c_1\zeta + c_2\zeta^2$, $N = c_1^2 - c_1c_2 + c_2^2$. Suppose that $p \mid a_1$, then $p \mid a_2^2 = p + a_1a_2 - a_1^2$, hence $p \mid a_2$, implying the contradiction $p^2 \mid p$. Hence $p \nmid a_1$, $p \nmid a_2$. Similarly $q \nmid b_1$, $q \nmid b_2$, and $p, q \nmid c_1$, $p, q \nmid c_2$. So all the inverses in the algorithm exist.

Suppose first that $p \equiv q \pmod 9$. Then $\frac{p-1}{3} \equiv \frac{q-1}{3} \not\equiv 0 \pmod 3$, so $\frac{p-1}{3} + 2\frac{q-1}{3} \equiv 0 \pmod 3$. If $r \equiv 1 \pmod p$, then $a_1 \equiv -a_2 \pmod p$, so $p \mid a_1+a_2$. Since $p = (a_1+a_2)^2 - a_1a_2$, it follows that $p \mid a_1a_2$, so $p \mid a_1$ or $p \mid a_2$ in contradiction to our previous observation. So $r \not\equiv 1$ (mod p), $r \not\equiv 1 \pmod q$, implying gcd $(r-1, N) = 1$ and $2 \le r \le N$. Now $\pi = a_2\zeta(a_1a_2^{-1} + \zeta)$, so $r \equiv -a_1a_2^{-1} \equiv \zeta \pmod \pi$ and $r^3 \equiv 1 \pmod \pi$. Similarly, $\psi = b_1\zeta^2(\zeta^2 + b_1^{-1}b_2\zeta)$, hence $r \equiv -b_2b_1^{-1} \equiv \zeta^2 \pmod \psi$ and $r^3 \equiv 1 \pmod \psi$. It follows that $r^3 \equiv 1 \pmod N$ and $\left[\frac{r}{\pi\psi}\right] =$

$$\left[\frac{\zeta}{\pi}\right]\left[\frac{\zeta}{\psi}\right]^2 = \zeta^{\frac{p-1}{3} + 2\frac{q-1}{3}} = 1.$$

Now assume $p \not\equiv q \pmod 9$, then $\frac{p-1}{3} \equiv -\frac{q-1}{3} \not\equiv 0 \pmod 3$, so $\frac{p-1}{3} + \frac{q-1}{3} \equiv 0 \pmod 3$. If $r \equiv 1 \pmod p$, then $c_1 \equiv -c_2 \pmod p$, so $p \mid c_1+c_2$, implying $p \mid c_1$ or $p \mid c_2$ as above and resulting in a contradiction. Hence $r \not\equiv 1$ (mod p), $r \not\equiv 1$ (mod q), implying again

$\gcd(r-1, N) = 1$ and $2 \le r \le N$. Now $\pi\psi = c_2\zeta(c_1c_2^{-1} + \zeta^2)$, so $r \equiv \zeta \pmod{\pi\psi}$, so $r^3 \equiv 1 \pmod N$ and $\left[\frac{r}{\pi\psi}\right] = \zeta^{\frac{p-1}{3}+\frac{q-1}{3}} = 1$. ❑

Lemma 7.2: Let $\alpha = a_1\zeta + a_2\zeta^2 \in \mathbb{O}$. Then α is primary if and only if $a_1 \equiv a_2 \not\equiv 0 \pmod 3$.

Proof: Assume that α is primary. Suppose that $a_1 \equiv 0 \pmod 3$, then $0 \equiv c(\alpha) \equiv 2a_2 \pmod 3$, so $a_2 \equiv 0 \pmod 3$, contradictory to $a_1 + a_2 \not\equiv 0 \pmod 3$. So $a_1 \not\equiv 0 \pmod 3$. Similarly, $a_2 \not\equiv 0 \pmod 3$. From $a_1 + a_2 \not\equiv 0 \pmod 3$, it follows that $a_1 \not\equiv -a_2 \pmod 3$, so $a_1 \equiv a_2 \not\equiv 0 \pmod 3$.

Now let $a_1 \equiv a_2 \not\equiv 0 \pmod 3$. By Definition 6.10 and Lemma 6.12, we need to prove $a_1 + a_2 \not\equiv 0 \pmod 3$, $a_1 + 2a_2 \equiv 0 \pmod 3$, and $\alpha\bar{\alpha} \equiv (a_1 + a_2)^2 \pmod 3$. Since $a_1 \equiv a_2 \equiv \pm 1 \pmod 3$, we have $a_1 + a_2 \equiv \pm 1 \not\equiv 0 \pmod 3$ and $a_1 + 2a_2 \equiv a_1 - a_2 \equiv 0 \pmod 3$. Finally, $\alpha\bar{\alpha} = \mathrm{N}(\alpha) = a_1^2 - a_1a_2 + a_2^2 \equiv a_1^2 + 2a_1a_2 + a_2^2 \equiv (a_1 + a_2)^2 \pmod 3$. ❑

Lemma 7.3: Let $\alpha = a_1\zeta + a_2\zeta^2 \in \mathbb{O}$ such that $a_1 + a_2 \not\equiv 0 \pmod 3$. Then one of α, $\zeta\alpha$, $\zeta^2\alpha$ is primary.

Proof: $\alpha = a_1\zeta + a_2\zeta^2$, $\alpha\zeta = a_1\zeta^2 + a_2 = -a_2\zeta + (a_1 - a_2)\zeta^2$, $\alpha\zeta^2 = a_1 + a_2\zeta = (a_2 - a_1)\zeta - a_1\zeta^2$. If $a_1, a_2 \not\equiv 0 \pmod 3$, then since $a_1 + a_2 \not\equiv 0 \pmod 3$, we must have $a_1 \equiv a_2 \not\equiv 0 \pmod 3$, so by Lemma 7.2, α is primary. If $a_1 \equiv 0 \pmod 3$, then $a_2 \not\equiv 0 \pmod 3$, hence $\alpha\zeta$ is primary. Finally, if $a_2 \equiv 0 \pmod 3$, then $a_1 \not\equiv 0 \pmod 3$, so $\alpha\zeta^2$ is primary. ❑

The law of reciprocity plus complementaries in $\mathbb{K}$ were first stated by Jacobi [Ja46] and explicitly proved by Eisenstein [Ei44a], [Ei44b].

Lemma 7.4: Let π, ψ be primary primes, $\pi = a_1\zeta + a_2\zeta^2$, $a_1 \equiv a_2 \equiv 1 \pmod 3$. Then

a) $\left[\frac{\pi}{\psi}\right] = \left[\frac{\psi}{\pi}\right]$

b) $\left[\frac{\pm 1}{\pi}\right] = 1,$

c) $\left[\frac{\zeta}{\pi}\right] = \zeta^{\frac{N(\pi)-1}{3}} = \zeta^{\frac{1+a_1-2a_2}{3}},$

d) $\left[\frac{\omega}{\pi}\right] = \zeta^{\frac{a_2-1}{3}}.$ □

Corollary: Lemma 7.4 holds if π and ψ are replaced by composite primary integers. □

This corollary enables us to compute cubic residue symbols as follows.

Algorithm 7.3: For $\alpha, \beta \in \mathbb{O}$, $\mathrm{Tr}(\beta) \not\equiv 0 \pmod 3$, $\gcd(\alpha, \beta) \simeq 1$, evaluate $\left[\frac{\alpha}{\beta}\right] = \zeta^s$, $0 \le s \le 2$.

```
1. Set $s = 0$.
2. Find a primary associate $\beta' = b_1\zeta + b_2\zeta^2$ of $\beta$.
3. Use Euclidean division to find $\gamma = c_1\zeta + c_2\zeta^2 \in \mathbb{O}$ such that $\gamma \equiv \alpha \pmod{\beta'}$ and
   $N(\gamma) < N(\beta')$.
4. { Factor out $\omega$ }    Set $i = 0$.
                              While $c_1 + c_2 \equiv 0 \pmod 3$ do
                                  Set $\gamma \leftarrow \frac{\gamma}{\omega}$, $i \leftarrow i + 1$.
5. { Factor out $\zeta$ }     Set $j = 0$.
                              If $c_1 \equiv 0 \pmod 3$ then
                                  set $j = 1$
                              If $c_2 \equiv 0 \pmod 3$ then
                                  set $j = 2$.
                              Set $\gamma \leftarrow \zeta^j\gamma$.
```

```
6. Set $s \leftarrow s + i\frac{b_2-1}{3} - j\frac{1+b_1-2b_2}{3} \pmod 3$, $0 \le s \le 2$.
7. If $\gamma \neq \pm 1$, then
        if $c_1 \equiv -1 \pmod 3$ then
            set $\gamma \leftarrow -\gamma$
        set $\alpha \leftarrow \beta'$, $\beta' \leftarrow \gamma$. Goto step 3. □
```

The algorithm halts when γ is a primary unit, i.e. when $\gamma = \pm 1$. For the computation of s in Step 6, we observe that i is the power of ω contained in γ whereas j is the power of ζ that γ needs to be multiplied by to make it primary. The change of the sign of γ in Step 7 does not change the residue symbol, since $\left[\frac{\gamma}{\beta}\right] = \left[\frac{-\gamma}{\beta}\right]$ by Lemma 7.4 b).

7.2.2 A Different Scheme

Recently, Loxton et al [LKBS92] presented a different cubic cryptosystem which is an extension of Williams' quadratic scheme [Wi80] to the case $\lambda = 3$. Field elements are written as a rational linear combination of 1 and $\zeta = \frac{1}{2}(-1+\sqrt{-3})$. The designer generates two primes $\pi, \psi \in \mathbb{O}$ such that $\pi \equiv 8 + 6\zeta \pmod 9$ and $\psi \equiv 5 + 6\zeta \pmod 9$, i.e. π and ψ are primary. Then $p = \mathrm{N}(\pi) \equiv 7 \pmod 9$ and $q = \mathrm{N}(\psi) \equiv 4 \pmod 9$, hence $f = \frac{(p-1)(q-1)}{9} \equiv -1 \pmod 3$. If we set $\rho = -\pi\psi$ and $\mathrm{N}(\rho) = N$, then $\left[\frac{\zeta}{\rho}\right] = 1$ and $\left[\frac{\omega}{\rho}\right] = \zeta$. The encryption exponent is $3e$ where $e \in \{1, \dots, N\}$ such that $\gcd(e, \phi(N)) = 1$, and the decryption exponent is $d \in \{1, \dots, N\}$ where $3ed \equiv \frac{f+1}{3} \pmod{\phi(N)}$. The public and private keys are $K_p = \{e, \rho\}$ and $K_s = \{d\}$.

Define the *fundamental region* A *with respect to* ρ as $\mathrm{A} = \{\gamma\rho \mid \gamma = c_0 + c_1\zeta;\ c_0, c_1 \in \mathbb{R};\ 0 \le c_0, c_1 < 1\}$. Then every $\beta \in \mathbb{O}$ is congruent (mod ρ) to exactly one element $\alpha \in \mathrm{A}$ which can be obtained as follows. Write $x = \frac{\beta}{\rho} = \frac{\beta\bar{\rho}}{N} = x_0 + x_1\zeta \in \mathbb{K}$, $x_0, x_1 \in \mathbb{Q}$. Set $y_i = \lfloor x_i \rfloor$, $z_i = x_i - y_i$ $(i = 0, 1)$, $y = y_0 + y_1\zeta$, $z = z_0 + z_1\zeta$, and $\alpha = z\rho$. Then $\beta = x\rho = y\rho + \alpha$ where

$\alpha \in \mathrm{A}$ and $y \in \mathbb{O}$. Set $\mathbb{H} = \mathrm{A} \cup \zeta\mathrm{A} \cup \zeta^2\mathrm{A}$. $\mathbb{H}$ can be thought of as a hexagon-shaped area in the compex plane with center 0 and corner points ρ, $(1+\zeta)\rho$, $\zeta\rho$, $-\rho$, $\zeta^2\rho$, and $-\zeta\rho$, and A is the rhombus-shaped subset of $\mathbb{H}$ with corners 0, ρ, $(1+\zeta)\rho$, and $\zeta\rho$ (see Figure 7.1).

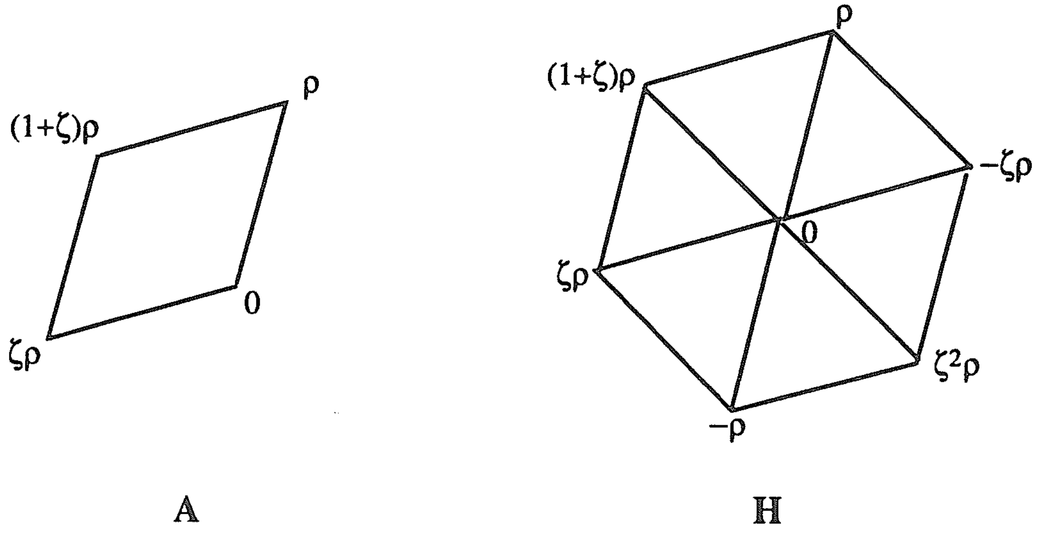

Figure 7.1

The message space $\mathcal{M}$ is obtained as follows. Let $\beta \in \mathbb{O}$ and find $\beta' \in \mathrm{A}$ such that $\beta' \equiv \beta \pmod{\rho}$. Set $\beta'' = \omega\beta' + 1$, so $\omega \nmid \beta''$. Let $\left[\frac{\beta''}{\rho}\right] = \zeta^m$, $0 \le m \le 2$. Set $E_1(\beta) = \omega^{3-m}\beta''$, so $\omega \mid E_1(\beta)$ and $\left[\frac{E_1(\beta)}{\rho}\right] = 1$. The message space $\mathcal{M}$ is defined to be $\mathcal{M} = \{M \in \mathrm{A} \mid E_1(M) \in \mathbb{H}\}$.

To encrypt $M \in \mathcal{M}$, compute $M_0 = E_1(M) = \omega^{3-m}(\omega M + 1)$ where $\left[\frac{\omega M+1}{\rho}\right] = \zeta^m$, $0 \le m \le 2$. Then find $C \in \mathrm{A}$ such that $C = E_2(M_0) \equiv M_0{}^{3e} \pmod{\rho}$. The ciphertext is $C \equiv E_2(E_1(M))$. To decipher C, the receiver first obtains $M_1 \in \mathrm{A}$ such that $M_1 \equiv D_2(C) \equiv C^d \pmod{\rho}$. Then $M_0 \equiv \zeta^k M_1 \pmod{\rho}$ for some $k \in \{0, 1, 2\}$. It is easy to identify the three points in $\mathbb{H}$ that are congruent to $M_1 \pmod{\rho}$, exactly one of which, say X, is a multiple of ω. Then we must have $X = E_1(M)$. Suppose $\omega^l \parallel X$, then set $D_1(X) = \omega^{-1}(\omega^{-l}X - 1) = M$, so $M = D_1(D_2(C))$.

Any algorithm A which finds the plaintext M corresponding to a cryptogram C can be shown to give rise to a probabilistic polynomial-time algorithm for factoring N requiring on average a bounded number of applications of A.

7.3 The case $\lambda = 5$

7.3.1 Quintic Residue Symbols

This is the smallest case with non-zero unit rank. Here, the unit rank is $\frac{5-3}{2} = 1$, so we have one fundamental unit $\eta = -(\zeta^2 + \zeta^3)$. Hence every unit ε in $\mathbb{K}$ has a unique representation $\varepsilon = \pm\zeta^j\eta^k$where $j, k \in \mathbb{Z}$ and $0 \leq j \leq 4$. If we choose $\zeta = \exp\left(\frac{2\pi i}{5}\right)$ then $\eta = \frac{1+\sqrt{5}}{2} > 1$.

For this case, we require Algorithms 6.1 - 6.3 in all their generality, i.e. there are no simple methods for finding a fifth primitive root of unity (mod N) or prime divisors in $\mathcal{O}$ of rational primes, as in the previous two cases. We will again give an explicit version of Algorithm 6.3 to compute quintic residue symbols.

Let $\alpha = \sum_{i=1}^{4} a_i\zeta^i \in \mathcal{O}$, $a_i \in \mathbb{Z}$ for $1 \leq i \leq 4$. Define the following quantities:

$$a = a(\alpha) = a_1 - a_2 - a_3 + a_4,$$

$$b = b(\alpha) = a_1 + a_2 + a_3 + a_4 = -\mathrm{Tr}(\alpha),$$

$$c = c(\alpha) = a_1 + 2a_2 + 3a_3 + 4a_4,$$

$$d = d(\alpha) = a_1 - 2a_2 + 2a_3 - a_4$$

Note that b and c are defined as in Section 6.3.

Lemma 7.5: Let $\alpha \in \mathcal{O}$ be such that $\alpha \equiv 0 \pmod{\omega}$ and $\alpha \equiv b \pmod{\omega^2}$. Then $\alpha\bar{\alpha} \equiv b^2 \pmod 5$ if and only if $a \equiv 0 \pmod 5$.

Proof: An easy calculation shows $5 = \omega^4\varepsilon$ where $\varepsilon = (\zeta + 1)^2\zeta^2$ is a unit. Furthermore $\bar{\omega} = 1 - \zeta^4 = -\zeta^4\omega$ and

$$\alpha + \bar{\alpha} = (a_1 + a_4)(\zeta + \zeta^4) + (a_2 + a_3)(\zeta^2 + \zeta^3) = a(\zeta + \zeta^4) - (a_2 + a_3) = a(\zeta + \zeta^4) + \frac{a - b}{2}.$$

From $\alpha \equiv b \pmod{\omega^2}$ and $\omega \simeq \bar{\omega}$, we obtain $\bar{\alpha} \equiv b \pmod{\omega^2}$, hence $\omega^4 \mid (\alpha - b)(\bar{\alpha} - b)$. Assume that $\alpha\bar{\alpha} \equiv b^2 \pmod 5$. Then, since $2^{-1} \equiv -2 \pmod 5$ and $5 \simeq \omega^4$:

$$\begin{aligned} 0 &\equiv (\alpha - b)(\bar{\alpha} - b) \equiv \alpha\bar{\alpha} - b(\alpha + \bar{\alpha}) + b^2 \equiv b^2 - b\left[a(\zeta + \zeta^4) - 2(a - b)\right] + b^2 \\ &\equiv b\left[2b - a(\zeta + \zeta^4) + 2(a - b)\right] \equiv ab(2 - \zeta - \zeta^4) \equiv -ab\zeta^4\omega^2 \pmod 5. \end{aligned}$$

Since $\omega \nmid b$ and ζ^4 is a unit, we have $\omega^2 \mid a$, so $a \equiv 0 \pmod 5$.

Conversely, assume $a \equiv 0 \pmod 5$. Then $\alpha + \bar{\alpha} \equiv -\frac{b}{2} \equiv 2b (\text{mod } 5)$, thus $0 \equiv (\alpha - b)(\bar{\alpha} - b)$ $\equiv \alpha\bar{\alpha} - b(\alpha + \bar{\alpha}) + b^2 \equiv \alpha\bar{\alpha} - b^2 \pmod 5$. ❑

Lemma 7.5 together with Lemma 6.12 gives rise to a practical test for a number to be primary as follows.

Corollary: $\alpha \in \mathbf{O}$ is primary if and only if

a) $b \not\equiv 0 \pmod 5$,

b) $c \equiv 0 \pmod 5$,

c) $a \equiv 0 \pmod 5$. ❑

Let $\alpha \in \mathbf{O}$. If $\mathrm{Tr}(\alpha) \not\equiv 0 \pmod 5$, then by Lemma 6.14, α has a primary associate. The following is a practical method for finding such an associate in constant time.

Lemma 7.6: Let $\alpha \in \mathbf{O}$ be such that $\mathrm{Tr}(\alpha) \not\equiv 0 \pmod 5$. Then α has a primary associate of the form $\alpha' = \zeta^j\eta^k\alpha$ where $0 \le j, k \le 4$.

Proof: Let $\alpha = \sum_{i=1}^{4} a_i\zeta^i$. We have $b = b(\alpha) \not\equiv 0 \pmod 5$. Then $\alpha\zeta^j \equiv b + (c+jb)\omega \pmod{\omega^2}$, so by Lemma 6.11, $b(\alpha\zeta^j) \equiv b \not\equiv 0 \pmod 5$ and $c(\alpha\zeta^j) \equiv c+jb \pmod 5$ for $0 \le j \le 4$. Since one of $c+jb$ $(0 \leq j \leq 4)$ must be divisible by 5, we have found an associate of α such that conditions a) and b) of the Corollary to Lemma 7.5 hold.

Assume now $b \not\equiv 0 \pmod 5$ and $c \equiv 0 \pmod 5$. Let $\alpha_j = \alpha\eta^j$ $(0 \leq j \leq 4)$. A straightforward calculation yields $\alpha_1 = \alpha\eta = (a_2-a_4)\zeta + (a_2+a_3-a_4)\zeta^2 + (a_2+a_3-a_1)\zeta^3 + (a_3-a_1)\zeta^4$ and $\eta^2 = \eta + 1$, hence $\alpha_j = \alpha_{j-2}\eta^2 = \alpha_{j-2}\eta + \alpha_{j-2} = \alpha_{j-1} + \alpha_{j-2}$ for $2 \leq j \leq 4$. If we let $n = \frac{a-b}{2} = -(a_2 + a_3)$, then

$a(\alpha_0) = a$, $a(\alpha_1) = (a_2-a_4) - (a_2+a_3-a_4) - (a_2+a_3-a_1) + (a_3-a_1) = -(a_2+a_3) = n$,

$a(\alpha_2) = n + a$, $a(\alpha_3) = 2n + a$, $a(\alpha_4) = 3n + 2a$.

Furthermore, since $\eta \equiv -2 \pmod{\omega^2}$, it follows that $b(\alpha_j) \equiv (-2)^j b \not\equiv 0 \pmod 5$ and $c(\alpha_j) \equiv (-2)^j c \equiv 0 \pmod 5$, so all the α_j satisfy conditions a) and b) of the Corollary to Lemma 7.5.

We need to prove that one of the α_j $(0 \leq j \leq 4)$ satisfies $a(\alpha_j) \equiv 0 \pmod 5$ and is hence primary. If $a \equiv 0 \pmod 5$, then α_0 is primary and if $n \equiv 0 \pmod 5$, then α_1 is primary. So suppose now that $an \not\equiv 0 \pmod 5$. Then if $a \equiv -n \pmod 5$, then α_2 is primary, if $a \equiv -2n \pmod 5$, then α_3 is primary, and if $a \equiv n \pmod 5$, then α_4 is primary. The only remaining case is $a \equiv 2n \pmod 5$, in which case $0 \equiv a - 2n \equiv b \pmod 5$, a contradiction. Hence we have found the required primary associate of α. ❑

For the evaluation of quintic residue symbols, we require complementaries for η and ω. These were explicitly stated by Williams [Wi76]). We summarize all quintic complementaries here.

Lemma 7.7: Let π, ψ be two distinct primary primes in $\mathbf{O}$. Then

a) $\left[\frac{\pi}{\psi}\right] = \left[\frac{\psi}{\pi}\right]$,

b) $\left[\frac{\pm 1}{\pi}\right] = 1$,

c) $\left[\frac{\zeta}{\pi}\right] = \zeta^{\frac{N(\pi)-1}{5}}$.

Furthermore, if $\pi = \sum_{i=1}^{4} a_i\zeta^i$, $N(\pi) = p \equiv 1 \pmod 5$, and $b^* \in \mathbb{Z}$ is such that $bb^* \equiv 1 \pmod 5$, then

d) $\left[\frac{\eta}{\pi}\right] = \zeta^{4db^*}$,

e) $\left[\frac{\omega}{\pi}\right] = \zeta^{4b^*\frac{c}{5}+3\frac{p+4}{5}}$. □

Corollary: Lemma 7.7 holds if π and ψ are replaced by composite primary integers. □

In order to perform Step 7 of Algorithm 6.3 in constant time, we need to be able to determine $\left[\frac{\varepsilon}{\beta}\right]$ quickly for a primary unit ε. Since $\eta \equiv -2 \pmod{\omega^2}$, η is primary, whereas ζ is not primary. Hence if ε is a primary unit, then we must have $\varepsilon = \pm\eta^i$ for some $i \in \mathbb{Z}$. Let $\varepsilon = \sum_{j=1}^{4} a_j\zeta^j$. An easy proof using induction on i shows that in this case $a_1 = a_4$ and $a_2 = a_3$, hence $\varepsilon = a_1(\zeta+\zeta^4) + a_2(\zeta^2+\zeta^3) = -\frac{a_1 + a_2}{2} + \frac{a_1 - a_2}{2}\sqrt{5}$. As usual, define the *Lucas* and *Fibonacci numbers*, respectively, by

$$\begin{array}{llll} L_0 = 2, & L_1 = 1, & L_{i+2} = L_{i+1} + L_i & (i \geq 0), \\ F_0 = 0, & F_1 = 1, & F_{i+2} = F_{i+1} + F_i & (i \geq 0). \end{array}$$

It is well-known that $\eta^\nu = \left(\frac{1+\sqrt{5}}{2}\right)^\nu = \frac{1}{2}(L_\nu + F_\nu\sqrt{5})$ for $\nu \geq 0$, hence if $\varepsilon = \pm\eta^i$ for $i \geq 0$, then $-(a_1 + a_2) = \pm L_i$ and $a_1 - a_2 = \pm F_i$. Now $L_{i+4} \equiv L_i \pmod 5$ and $F_{i+20} \equiv F_i \pmod 5$, and the pairs ($L_i \pmod 5$, $F_i \pmod 5$) ($0 \leq i \leq 19$) are exactly the pairs (m, n) ($1 \leq m \leq 4$, $0 \leq n \leq 4$). Since by Lemma 7.7 b), the sign of ε does not affect the value of $\left[\frac{\varepsilon}{\beta}\right]$, this gives rise to the following constant time algorithm.

Algorithm 7.4: For $\varepsilon = \sum_{j=1}^{4} a_j \zeta^j \in \mathbb{O}$ a primary unit, $\beta \in \mathbb{O}-\{0\}$, find i such that $\left[\frac{\varepsilon}{\beta}\right] = \left[\frac{\eta}{\beta}\right]^i$.

```
1. Compute a table containing the triples ($i$, $L_i$ (mod 5) $F_i$ (mod 5)) for $0 \le i \le 19$.
   (This table need only be computed once and can be re-used for any subsequent
   quintic residue symbol computation.)
2. Return $j \in \{0, \dots, 19\}$ such that $-(a_1 + a_2) \equiv L_j \pmod 5$, $a_1 - a_2 \equiv F_j \pmod 5$.
3. Set $i \equiv j \pmod 5$, $0 \le i \le 4$. ❑
```

We are now able to present our algorithm to evaluate the quintic residue symbol.

Algorithm 7.5: For $\alpha, \beta \in \mathbb{O}-\{0\}$, $\mathrm{Tr}(\beta) \not\equiv 0 \pmod 5$, $\gcd(\alpha, \beta) \simeq 1$, evaluate $\left[\frac{\alpha}{\beta}\right] = \zeta^s$, $0 \le s \le 4$.

```
1. Set $s = 0$.
2. Find a primary associate $\beta'$ of $\beta$.
3. Compute $N(\beta')$.
4. Use Euclidean division to find $\gamma \in \mathbb{O}$ such that $\gamma \equiv \alpha \pmod{\beta'}$ and $N(\gamma) < N(\beta')$.
5. Compute   $b \equiv b(\beta) \pmod 5$,     $b^* \equiv b^{-1} \pmod 5$,
             $c \equiv c(\beta) \pmod{25}$,   $d \equiv d(\beta) \pmod 5$.
6. { Factor out $\omega$ }  Set $i = 0$.
        While $b(\gamma) \equiv 0 \pmod 5$ do
            Set $\gamma \leftarrow \frac{\gamma}{\omega}$, $i \leftarrow i + 1$.
7. { Factor out $\zeta$ }  Set $j \equiv -c(\gamma)b^*(\gamma) \pmod 5$, $0 \le j \le 4$. Set $\gamma \leftarrow \gamma\zeta^j$.
8. { Factor out $\eta$ }  Compute $a(\gamma) \pmod 5$. Set $k = 0$.
        If $a(\gamma) \not\equiv 0 \pmod 5$, then
            compute $n(\gamma) \equiv \frac{a(\gamma) - b(\gamma)}{2} \equiv -a_2(\gamma) - a_3(\gamma) \pmod 5$.
```

```
                If $0 \equiv n(\gamma) \pmod 5$, then set $k = 1$.
                If $a(\gamma) \equiv -n(\gamma) \pmod 5$, then set $k = 2$.
                If $a(\gamma) \equiv -2n(\gamma) \pmod 5$, then set $k = 3$.
                If $a(\gamma) \equiv n(\gamma) \pmod 5$, then set $k = 4$.
            Set $\gamma \leftarrow \gamma\eta^k$.
9.  Set $s \leftarrow s + i\left(4b^*\frac{c}{5} + 3\frac{N(\beta')+4}{5}\right) - j\frac{N(\beta')-1}{5} - 4kdb^* \pmod 5$, $0 \le s \le 4$.
10. Compute $N(\gamma)$.
    If $N(\gamma) > 1$, then
        Set $\alpha \leftarrow \beta'$, $\beta' \leftarrow \gamma$. Goto Step 4.
    else
        Use Algorithm 7.4 to find $l$ such that $\left[\frac{\gamma}{\beta}\right] = \left[\frac{\eta}{\beta}\right]^l$.
        Set $s \leftarrow s + 4db^*l \pmod 5$ $0 \le s \le 4$. ❑
```

For the computation of s in Step 9, note again that i is the power of ω contained in γ, whereas j and k are the powers of ζ and η, respectively, which γ needs to be multiplied by to obtain $c(\gamma) \equiv 0 \pmod 5$ and $a(\gamma) \equiv 0 \pmod 5$. Hence we need to add the appropriate multiple of i given by Lemma 7.7 to s while subtracting the correct multiples of j and k.

7.3.2 Computational Results

We implemented a number of our algorithms for the quintic case. Our programs were written in C language using the GNU multiple precision integer arithmetic library (Granlund [Gr91]) and were run on a DECStation 5000. We wrote routines for the following algorithms.

- Uspensky's Euclidean division method described in Section 6.4.2.
- Kummer's Euclidean division method described in Section 6.4.3.
- Lenstra's Euclidean division method described in Section 6.4.4.

- Finding a prime divisor of a rational prime $p \equiv 1 \pmod 5$ using the gcd method described in Lemma 6.7.
- The Residue Symbol Algorithm 7.5, including Algorithm 7.4.
- Key generation, encryption, and decryption.

We now present some of our computational results.

Prime Divisors

Here, we used all three Euclidean division methods to compute prime divisors of rational primes $p \equiv 1 \pmod 5$. Despite the different upper bounds on the quotient of the norms of the remainder and the divisor given in Section 6.4, all three algorithms performed essentially the same. In particular, even though Lenstra's method gives a better bound than Uspensky's and Kummer's methods, it does not seem to run significantly faster in general. We ran the different gcd algorithms on three primes of approximately 100, 140, and 150 decimal digits, respectively. Each of the three algorithms produced a different prime divisor for each prime. The table below presents the results of our computation. Column 1 gives the rational prime p whose prime divisor π we computed and column 2 states the number of digits of the largest coefficient a_i of π, where $\pi = a_1\zeta+a_2\zeta^2+a_3\zeta^3+a_4\zeta^4$ (the other coefficients of π were always within a factor of 10 in absolute value of $|a_i|$). Column 3 specifies the Euclidean division algorithm used and columns 4 and 5 contain the number of Euclidean divisions performed and the CPU time (in seconds) required by the corresponding prime divisor computation.

p	$\lfloor \log_{10} \lvert a_i \rvert \rfloor + 1$	Eucl. Div. Alg.	# Eucl. Div.	CPU time (secs)
$10^{100} + 2911$	26	Uspensky	145	2.5
	26	Kummer	139	2.4
	26	Lenstra	134	2.4
$10^{140} + 2691$	36	Uspensky	185	5.7
	37	Kummer	195	5.7
	39	Lenstra	186	5.7
$10^{150} + 771$	39	Uspensky	202	7.4
	38	Kummer	210	7.3
	40	Lenstra	198	7.3

Residue Symbols

We computed the quintic residue symbol $\left[\frac{\alpha}{\beta}\right]$ for two pairs of integers $\alpha, \beta \in \mathcal{O}$ whose residue symbols are 2 and 1, respectively. Both numerator and denominator of the first pair had coefficients of 25 digits, for the second pair the number of digits of each coefficient was 50. Again, we give the number of Euclidean divisions performed and the computation time in seconds required for each of the three Euclidean division methods. As before, all three methods performed quite similarly.

# digits / coeff.	Eucl. Div. Alg.	# Eucl. Div.	CPU time (secs)
25	Uspensky	61	0.3
	Kummer	56	0.3
	Lenstra	57	0.3
50	Uspensky	119	1.2
	Kummer	133	1.2
	Lenstra	115	1.1

The Cryptosystem

Since we were mainly interested in encryption and decryption speeds (rather than generating a secure key for practical purposes), we did not attempt to find primes which are considered safe according to the criteria in Section 2.1. Instead, we used the following procedure to generate a modulus N of n digits.

Choose a starting value $I \approx \sqrt{N}$ (we used $I = 10^{\lfloor n/2 \rfloor}$). Find the smallest odd integer k satisfying $k \geq I$, $k \equiv 1 \pmod{5}$, $k \not\equiv 1 \pmod{25}$ (in our case $k = 10^{\lfloor n/2 \rfloor}+11$). k is the first candidate for a prime divisor of N. Trial-divide k by a few small primes (we used the first 12 primes except 2 and 5, i.e. 3, 7, 11, 13, 19, 23, 29, 31, 37, 41). If k has no small prime divisor, apply a simple probabilistic primality test to k, such as the base b Fermat test, which computes $f \equiv b^{k-1} \pmod{k}$ $0, \leq f \leq k-1$. If $f = 1$, then k satisfies Fermat's little theorem with base b and is hence a *base b probable prime*. Since the primality test is quite time-consuming, we generally apply no more than two rounds of the test (with $b = 2$ and $b = 3$). In the unlikely event where we obtain a *pseudo-prime* (i.e. a composite number which passes the probabilistic test) after our trial division and primality testing procedures, our enciphering and deciphering routines will produce random results, if the pseudo-prime is used as one factor of the modulus. Hence this situation will easily be detected after a few test encryptions and decryptions. For our implementation, we used two rounds of a built-in Fermat primality test routine which was part of the GNU mp library.

If k fails either the trial division or the primality test, repeat the procedure with k+50 (the next odd number k' such that $k' \equiv 1 \pmod 5$ and $k' \not\equiv 1 \pmod{25}$). Once a probabilistic prime p is found, repeat the process, starting with p+50 as the first candidate for q.

To find an encryption exponent e, we simply used the built-in GNU mp random number generator to find a random positive integer e' of a specified size (Usually $e' < N$ and $e' \approx N$). Then we computed $\delta = \gcd(e', \phi(N))$. Set $e = \frac{e'}{\delta}$, unless δ exceeds a certain bound (say $\delta > 10^{20}$), in which case we try again with $e'+1$.

Our implementation is much slower than a commercial RSA implementation for several reasons. Most importantly, it is entirely done in software, whereas many RSA applications have hardware available for their modular exponentiation. Furthermore, the residue symbol computation required for each cryptogram reduces the encryption speed of our system relative to RSA. This is also the reason why the rate of decipherment of our scheme is noticeably faster than the rate of encipherment. Finally, we were not able to optimize our multi-precision integer arithmetic routines mathematically or computationally, since they were third party software.

We conclude with two computational examples using moduli of approximately 100 and 200 digits, respectively. We give the parameters for the encipherment/decipherment keys and the corresponding encryption and decryption rates ($c_1, \ldots, c_4$ are the coefficients of $\pi\psi = \sum_{i=1}^{4} c_i\zeta^i$).

Example 1 (101 digit modulus):

$p = 10^{50} + 961,$ $q = 10^{50} + 1441,$ $N = 10^{100} + 2402 \cdot 10^{50} + 1384801,$

$r =$ 84931547665997521504413684614632156842168287791540
50855262544885509727236020916307125712929698277223 (100 digits)

$S = 4096$

$c_1 = -19136962415978924892222019$
$c_2 = -42309655017145551720914389$
$c_3 = -38328646001618243482946787$
$c_4 = -10340542598589099160896792$ (26 digits each)

$e =$ 81227992491335015952405420535534223326596719742406
9736020004351337230083194425751151396644016825 (96 digits)

$d =$ 38348577020019051005425929707268080592343243504439
0678618190196551875876635899059411355762319672201 (99 digits)

Key generation time: 2.9 seonds

Encryption rate: 55 characters/second (438 bits/second)

Decryption rate: 127 characters/second (1014 bits/second)

Example 2 (199 digit modulus):

$p = 10^{99} + 711,$ $q = 10^{99} + 2191,$ $N = 10^{198} + 2902 \cdot 10^{99} + 1557801,$

$r =$ 19841667173631890576222801178500689564530478358349
40317663963772812536888504019567615811442887807822
76455018845488367286185393410115682830003301374143
902351412626927276227869125489540038751002392398 (198 digits)

$S = 4096$

$c_1 = -336043338189666480827612322963380084482537477962065$
$c_2 = 130248364308303176166329013581338757407688634701424$
$c_3 = 753419296226214300427093232597651775544590428954738$
$c_4 = 673411902810673123492520704219434618383499989881146$ (51 digits each)

$e =$ 30602494974337378197188096303216409530262243108541
59786417980102017939673386985158775037178350502849
39824415736943251822538042894474495797284102421894
2601058905129605292730316607041189314666605 (193 digits)

$d =$ 57003408258214914795329755246327607369549435623247
54457842319295154706693519706595265877532330419585
23159031283744154859226925349579300493549878358868
1043707306420521443146236469401845238052076813 (196 digits)

Key generation time: 22 seconds

Encryption rate: 23 characters/second (183 bits/second)

Decryption rate: 38 characters/second (307 bits/second)

Part II

A Key Exchange Protocol Using Real Quadratic Fields

8. Key Exchange Using Finite Groups

8.1 The Diffie-Hellman Key Exchange Protocol

The first algorithm for exchanging a secret key across a public channel was given by Diffie and Hellman. The original version of their protocol given in [DH76] has already been outlined in Section 1.1.4. We will now repeat the scheme in more generality. Suppose that two communication partners Alice and Bob wish to establish a key for a private-key cryptographic conversation without making use of a secure channel. Then they perform the following steps.

1. Alice and Bob agree on a finite multiplicative group G of order $N = |G|$ and an element $g \in G$. Both G and g can be made public.
2. Alice generates a random integer $a \in \{1, \ldots, N\}$ and computes $x = g^a$. She transmits x to Bob, but keeps a secret.
3. Bob generates a random integer $b \in \{1, \ldots, N\}$ and computes $y = g^b$. He transmits y to Alice, but keeps b secret.
4. From a and y, Alice computes $k = y^a = g^{ba}$.
5. From b and x, Bob computes $k = y^b = g^{ab}$.

After the protocol has been executed, both parties are in possession of the same group element k, which can then be used to obtain their common key in whatever manner they have agreed upon ahead of time (e.g., for a DES key, k could be associated with a bit string S in some fashion, and the high order 56 bits of S represent the DES key).

The protocol requires one round of communication. Generally, the group should be chosen in such a way that its elements can be represented in binary, where each element requires no more than $O(\log N)$ bits of storage, so that the bandwidth of the communication channel can be bounded by $O(\log N)$. Then the computational cost for each party is $O(\log N)$, since both partners must perform two exponentiations involving exponents bounded by N.

For key distribution between multiple users, each user i generates a random number $a_i \in \{1, \ldots, N\}$, computes $x_i = g^{a_i}$, and stores x_i in a public directory. Now if Alice wishes to exchange a key with Bob, she recalls her secret number a_{A} which she used for her directory entry $x_{\mathrm{A}} = g^{a_{\mathrm{A}}}$, looks up Bob's entry x_{B} in the directory, and sends $x_{\mathrm{B}}^{a_{\mathrm{A}}}$ to Bob. Similarly, Bob looks up Alice's entry x_{A} and transmits $x_{\mathrm{A}}^{a_{\mathrm{B}}}$ to Alice. This reduces the computational effort for each user to $O(\log N)$ per key exchange.

The original version of the protocol given in [DH76] used $\mathbb{G} = \mathrm{GF}(p)^*$. Since then, numerous other groups have been suggested to serve as the basis for a Diffie-Hellman-like key exchange scheme, such as

i) the multiplicative group of an arbitrary finite field,

ii) the group of invertible $n \times n$ matrices over $\mathrm{GF}(p)$ (Odoni, Varadharajan & Sanders [OVS84]),

iii) the group of integers (mod n) relatively to n, where n is the product of two primes (Shmuely [Sh85], McCurley [Mc88]),

iv) the group of points on an elliptic curve over a finite field (Miller [Mi86], Koblitz [Ko87b]),

v) groups associated with hyperelliptic curves (Koblitz [Ko90], [Ko88]),

vi) the class group of an imaginary quadratic field (Buchmann & Williams [BW88a], Buchmann, Düllmann & Williams [BDW90]).

In order to break the scheme, an adversary needs to be able to infer $k = g^{ab}$ from $\mathbb{G}$, g, $x = g^a$, and $y = g^b$, without knowledge of a or b. A cryptanalyst could achieve this if he were able to compute for any $z \in \mathbb{G}$ the *discrete logarithm* or *index* of z in $\mathbb{G}$, i.e. an integer $c \in \{0, \ldots, N\text{-}1\}$ such that $z = g^c$, in time polynomial in $\log N$. It is unknown whether breaking the Diffie-Hellman protocol is equivalent in difficulty to solving the discrete logarithm problem (DLP), regardless of the choice of group.

8.2 The Discrete Logarithm Problem

For a comprehensive survey of algorithms for solving the DLP, see Odlyzko [Od84] and McCurley [Mc90]. Henceforth, assume that G is a finite multiplicative group of order N, and that $g \in$ G is a fixed element in G. We wish to find, for an arbitrary element $z \in$ G which is not the identity 1_G of G, the index $c = \log_g z$, i.e. the unique power $c \in \{1, \dots N\text{-}1\}$ such that $g^c = z$.

8.2.1 Deterministic Algorithms

The most direct approach to determining discrete logarithms in G is to precompute a table of the powers g^i $(1 \leq i \leq N\text{-}1)$ of g and find the index of any group element by table look-up. Clearly, this is infeasible, since the table requires N-2 group operations and storage of N-1 entries.

We can improve the running time to $O(\sqrt{N} \log N)$ arithmetic operations and reduce the storage requirement to $O(\sqrt{N})$ table entries by employing Shanks' "baby step-giant step" idea (Knuth [Kn73, pp. 9, 575-576]. Set $m = \lceil \sqrt{N} \rceil$ to be the least integer above $\sqrt{N}$. Shanks' method makes use of the fact that every discrete logarithm $c \in \{1, \dots, N\text{-}1\}$ can be written as $c = qm - r$ where $0 \leq r < m$ and $1 \leq q \leq m$. Assume that we can quickly enumerate the group elements $g_1, \dots, g_N$, i.e. there exists an easily computable function f: G $\rightarrow \{1, \dots, N\}$. To find $\log_g z$, compute the sets $S = \{(i, g_j) \mid g_j = zg^i, 0 \leq i \leq m\}$ and $T = \{(i, g_j) \mid g_j = g^{mi}, 0 \leq i \leq m\}$ (note that T need only be computed once whereas S must be recomputed for each logarithm). S and T require storage of a total of $O(m) = O(\sqrt{N})$ group elements and can be generated using $O(\sqrt{N} \log N)$ arithmetic operations ($O(\sqrt{N})$ exponentiations). Sort S and T according to the second coordinate of each element. This requires $O(\sqrt{N} \log N)$ comparisons. Now scan S and T until two elements $s = (r, g_j) \in S$ and $t = (q, g_j) \in T$ are found that match in their second coordinate. Then $g_j = zg^r$ and $g_j = g^{mq}$, hence $z = g^{mq-r}$.

A more practical algorithm was given by Pollard [Po78]. It's running time can be heuristically estimated to be similar to that of the previous method, while the space required seems to be much smaller. In the first stage of the algorithm, we find integers s and t such that $z^s = g^t$. In order to do this, we partition G into three sets S_1, S_2, S_3 of approximately equal size. Set $x_0 = 1_G$ and compute a sequence $x_0, x_1, \ldots$, where

$$x_{i+1} = \begin{cases} zx_i \text{ if } x_i \in S_1 \\ x_i^2 \text{ if } x_i \in S_2 \\ gx_i \text{ if } x_i \in S_3 \end{cases} \qquad (i \geq 0).$$

This defines a sequence of integers a_i and b_i such that $x_i = z^{a_i} g^{b_i}$, which can easily be shown to satisfy the following recurrences: $a_0 = b_0 = 0$,

$$a_{i+1} = \begin{cases} a_i{+}1 \ (\text{mod } N) & \text{if } x_i \in S_1 \\ 2a_i \ (\text{mod } N) & \text{if } x_i \in S_2 \\ a_i & \text{if } x_i \in S_3 \end{cases}, \quad b_{i+1} = \begin{cases} b_i & \text{if } x_i \in S_1 \\ 2b_i \ (\text{mod } N) & \text{if } x_i \in S_2 \\ b_i{+}1 \ (\text{mod } N) & \text{if } x_i \in S_3 \end{cases} \quad (i \geq 0),$$

where all remainders (mod N) are taken to be between 0 and N-1. If the sequence $(x_i)_{i \geq 0}$ were to behave like a random sequence – a reasonable assumption if the sets S_1, S_2, S_3 are chosen randomly – then we should expect that there exists $i \in \{0, \ldots \lfloor 3\sqrt{N} \rfloor\}$ such that $x_i = x_{2i}$, and we can find i by computing the 6-tuples $(x_i, a_i, b_i, x_{2i}, a_{2i}, b_{2i})$ recursively. If $x_i = x_{2i}$, then $z^s = g^t$ is satisfied with $s = a_i - a_{2i}$ (mod N) and $t = b_{2i} - b_i$ (mod N).

Let $d = \gcd(s, N)$. Then the Extended Euclidean Algorithm yields $u, v \in \mathbb{Z}$ such that $d = us + vN$, hence $z^s = g^t$ implies $z^d = g^{ut}$. If $d = 1$, then $\log_g z = ut$ (mod N), otherwise $d > 1$ and we must have $d \mid ut$. Extracting d-th roots from the identity $z^d = g^{ut}$ yields $z = g^{\frac{ut+iN}{d}}$ for some $i \in \{1, \ldots d\}$, so we can check this last equation for $i = 0, 1, \ldots$, until the correct value of i is found, in which case we have found $\log_g z$. Note that if s were a random residue (mod N), then we expect d to be small most of the time.

The last algorithm to be discussed in this section is due to Pohlig and Hellman [PH78], who credit its earlier independent discovery to Silver. It is very efficient if N is *smooth*, i.e.

if N has only small prime factors[1]. Let $N = p_1^{e_1} \cdots p_r^{e_r}$ be the unique prime factorization of N. It suffices to find the index of z modulo each $p_i^{e_i}$ $(1 \le i \le r)$; using the Chinese Remainder Theorem, we can then combine the indices modulo $p_i^{e_i}$ to find the index of z in G in time $O(r \log N)$.

Let $p \in \{p_1, \ldots, p_r\}$, $p^e \parallel N$ $(e \ge 1)$, and let $c \equiv \log_g z \pmod{p^e}$, $0 < c < p^e$. We compute the p-ary representation $c = \sum_{i=0}^{e-1} b_i p^i$, $0 \le b_i \le p-1$ for $i \in \{0, \ldots, e-1\}$ as follows. Set $\gamma = g^{\frac{N}{p}}$. From $\log_g z = c + kp^e$ for some $k \in \mathbb{Z}$, it follows that

$$\frac{N}{p} \log_g z = \frac{cN}{p} + Nkp^{e-1} = N\left(\frac{b_0}{p} + \sum_{i=1}^{e-1} b_i p^{i-1} + kp^{e-1}\right) = N\left(\frac{b_0}{p} + M\right)$$

for $M \in \mathbb{Z}$, hence $z^{\frac{N}{p}} = g^{\frac{N}{p}\log_g z} = g^{\frac{Nb_0}{p}} = \gamma^{b_0}$. To find b_0, we compute $z^{\frac{N}{p}}$ and γ^i for $i = 0, 1, \ldots,$ until we obtain $b_0 = i$ such that $\gamma^i = z^{\frac{N}{p}}$. (Alternatively, we could use the baby step-giant step method to search for the correct power of γ. This would reduce the running time of the algorithm while increasing the storage space as well as introducing some precomputation.)

If $e \ge 2$, then determine b_1 as follows. Compute $h = g^{-1} = g^{N-1}$ and $z_1 = zh^{b_0}$. Then again $\frac{N}{p^2} \log_g z = N\left(\frac{b_0}{p^2} + \frac{b_1}{p} + M\right)$ for $M \in \mathbb{Z}$, hence $z_1^{\frac{N}{p^2}} = z^{\frac{N}{p^2}} g^{-\frac{b_0 N}{p^2}} = g^{\frac{Nb_1}{p}} = \gamma^{b_1}$, and as before, we compute $z_1^{\frac{N}{p^2}}$ and search through the powers of γ to find b_1. If $e \ge 3$, we set $z_2 = z_1 h^{pb_1}$, compute $z_2^{\frac{N}{p^3}}$, and generate powers of γ until we find b_2 such that $\gamma^{b_2} = z_2^{\frac{N}{p^3}}$. Continuing in this fashion, we can find $b_3, \ldots, b_{e-1}$. This process is performed for all p_i $(1 \le i \le r)$. Pohlig and Hellman show that there is a time-memory trade-off that can be

[1] Pohlig and Hellman first introduced their algorithm using the group $G = \mathrm{GF}(p)^*$ of order p-1. Recall that Pollard gave a fast method for factoring an integer N when N-1 is smooth [Po75]. This is only one of a number of parallels between the problems of factoring and extracting discrete logarithms.

exploited to result in a running time of $O\left(\sum_{i=1}^{r} e_i(\log N + p_i^{1-\delta_i}(1+\log p_i^{\delta_i}))\right)$ group operations, using $O\left(\log N \sum_{i=1}^{r} (1+p_i^{\delta_i})\right)$ bits of memory, and requiring a precomputation of $O\left(\sum_{i=1}^{r}(p_i^{\delta_i} \log p_i^{\delta_i} + \log N)\right)$ group operations, where $\delta_i \in \mathbb{R}, 0 \leq \delta_i \leq 1$ for $i \in \{1, \dots, r\}$.

8.2.2 The Index Calculus Method

In contrast to the previous techniques, this algorithm is probabilistic rather than deterministic. It is quite similar to the factor base factoring method by Morrison & Brillhart [MB75]. The basic ideas are due to Western & Miller [WM68], although the approach first appeared in the work of Kraitchik ([Kr22], pp. 119-123, [Kr24], pp. 69-70, 216-267]) and Cunningham (see [WM68]). The actual algorithm was introduced independently by Adleman [Ad79], Merkle [Me79], and Pollard [Po78].

The method consists of two stages; a precomputation stage, in which a database of certain discrete logarithms in G is generated, and a second stage, in which the logarithm of an arbitrary group element is determined using this database. During the precomputation phase, we collect identities of the form $\prod_{j=1}^{B} a_j^{s_{ij}} = g^{t_i}$, where $a_1, \dots, a_B$ is a set of fixed elements in G (the *factor base*) and $s_{i1}, \dots, s_{iB}, t_i \in \{0, \dots, N-1\}$ for $i = 0, 1, \dots$ These relations give rise to a set of linear congruences $\sum_{j=1}^{B} s_{ij} \log_g a_j \equiv t_i \pmod{N}$, in which the unknowns are the $\log_g a_j$ $(1 \leq j \leq B)$. We solve this system of congruences, thus creating our database of known logarithms. Note that this stage need only be performed once.

To compute a particular index $\log_g z$ ($z \in \mathbb{G}$), we construct an identity of the form $\prod_{j=1}^{B} a_j^{e_j} = zg^e$ in $\mathbb{G}$, whence it follows that $\log_g z \equiv \sum_{j=1}^{B} e_j \log_g a_j - e \pmod{N}$. Usually, this process takes considerably less time than the first stage of the algorithm.

Clearly, this rough outline of the algorithm leaves many questions unanswered. First, it is not at all obvious how to efficiently construct relations of the form $\prod_{j=1}^{B} a_j^{s_{ij}} = g^{t_i}$ and in fact, there are only a few groups for which it is known how to generate such identities. Secondly, the number B of factor base elements $a_1, \ldots, a_B$ must be chosen carefully. In order to solve the system of linear congruences $\sum_{j=1}^{B} s_{ij} \log_g a_j \equiv t_i \pmod{N}$ ($i = 0, 1, \ldots$) uniquely for $\log_g a_j$ ($1 \leq j \leq B$), we require a sufficient number of congruences (at least B). If B were chosen too large, then it would be difficult to find enough factor relations, and in addition, the effort of solving the linear system of congruences would be considerable. On the other hand, if B were picked too small, then it is unlikely that a "typical" group element would factor over our factor base. The optimal choice of B requires a sophisticated analysis of probabilities.

We will now give more details with regard to the method and its complexity for some specific groups.

The case $\mathbb{G} = \mathrm{GF}(p)^*$: Here the factor base consists of the first B primes $p_1, \ldots, p_B$. To generate the relations for stage one, we choose a random integer $t \in \{1, \ldots, p\text{-}1\}$ and compute $r \equiv g^t \pmod{p}$, $0 < r < p$. We then try to factor r over $\{p_1, \ldots, p_B\}$, say by using trial division. If r factors over our base, then we are successful in obtaining a factor relation. To compute an index $\log_g z$ in $\mathrm{GF}(p)^*$, pick a random integer e, compute $r \equiv zg^e \pmod{p}$, $0 < r < p$, and see if r factors as a product of $p_1, \ldots, p_B$. If $r = \prod_{i=1}^{B} p_i^{e_i}$, then $\log_g z \equiv \sum_{i=1}^{B} e_i \log_g p_i - e \pmod{p\text{-}1}$. More details of this case are given in Koblitz [Ko87a].

The running time of this algorithm is subexponential in $\log p$ and is comparable to the time required for factoring an integer of magnitude p. Initially, the complexity was analyzed to be $L(p)^{2+o(1)}$ for stage one and $L(p)^{\frac{3}{2}+o(1)}$ for stage two, where $L(p) = \exp\left(\sqrt{\log p \log \log p}\right)$. Pomerance [Po87] improved this estimate $L(p)^{\sqrt{2}+o(1)}$, and this is the smallest asymptotic bound that is rigorously proved for GF(p)*. Coppersmith, Odlyzko, and Schroeppel [COS86] described three index calculus algorithms whose running time they heuristically estimate to be $L(p)^{1+o(1)}$ for the precomputation and $L(p)^{\frac{1}{2}+o(1)}$ for the index computation. Recent results by Gordon [Go93a] suggest that a technique similar to the number field sieve for factoring large integers could be used for computing discrete logarithms over GF(p)*, resulting in a method with heuristic complexity $L'(p)^c$ where $L'(p) = \exp\left(\sqrt[3]{\log p\ (\log \log p)^2}\right)$.

The case G = GF(p^n), *p a fixed small prime*: For details, see again [Ko87a]. It is well-known that GF(p^n) is isomorphic to the residue ring GF(p)[x]/$f(x)$, where $f(x) \in$ GF(p)[x] is any irreducible polynomial of degree n, so if we fix $f(x)$, then any element in GF(p^n) can be represented uniquely as a polynomial in GF(p)[x] of degree at most n-1. In particular, $g = g(x) \in$ GF(p)[x]. In this case, the factor base is usually chosen to consist of all monic irreducible polynomials over GF(p) of degree $\leq m$, where $m < n$ is chosen such that the size B of the factor base is optimal. It is possible to significantly improve the complexity of this case over that of the case G = GF(p)*. This is due to the fact that factoring polynomials is much easier than factoring integers (see for example Berlekamp's method [Be70]). The first subexponential DLP algorithm for GF(p^n) (p fixed) was given by Hellman and Reyneri [HR83]. The running time for the specific case GF(2^n) (often a popular choice for cryptographic schemes, since the arithmetic in this field can be nicely implemented in hardware) was analyzed by Blake et al [BFMV84] and in detail by Odlyzko [Od84]. Coppersmith's heuristic arguments for GF(p^n) suggest a running time of $L'(p^n)^c = \exp\left(\sqrt[3]{n\ (\log n)^2}\right)^c$ [Co84].

The case $\mathbf{G} = \mathrm{GF}(p^n)$, $n \in \mathbf{Z}^{>0}$ *fixed*: Here, the special case $n = 2$ was analyzed by ElGamal [El85a]. Lovorn gave a subexponential method for solving the DLP if $\log p \leq n^{0.98}$ [Lo92]. Recently, Gordon announced that there exists a $L'(p^n)^c$ algorithm for computing indices in $\mathrm{Gf}(p^n)$ (n fixed), which again uses the technique of the number field sieve [Go93b].

Finally, we point out that the first subexponential algorithm for computing algorithms over an arbitrary finite field $\mathrm{GF}(p^n)$ (p any prime, $n \in \mathbf{Z}^{>0}$) is due to Adleman and Demarrais [AD93].

The case $\mathbf{G} = \mathrm{Cl}(\mathbf{Q}(\sqrt{-D}))$: Let $\mathbf{K}$ be the field generated by the square root of $-D$ where $D \in \mathbf{Z}^{>0}$ is squarefree. Then $\mathbf{K}$ has a generating polynomial $f(x) = x^2+D$, hence $(\mathbf{K}:\mathbf{Q}) = 2$. $\mathbf{K}$ is called an *imaginary quadratic field*. The discriminant of $\mathbf{K}$ is $\Delta = \frac{4}{\sigma^2}D$, where

$$\sigma = \begin{cases} 1 \text{ if } D \equiv 2,\ 3 \pmod 4, \\ 2 \text{ if } D \equiv 1 \pmod 4. \end{cases}$$

The DLP in the class group $\mathrm{Cl}(\mathbf{K})$ of $\mathbf{K}$ can be stated as follows. If $[\mathbf{g}], [\mathbf{z}] \in \mathrm{Cl}(\mathbf{K})$ are two ideal classes ($\mathbf{g}$, $\mathbf{z}$ ideals in $\mathbf{K}$) such that $[\mathbf{z}] = [\mathbf{g}]^c$, or equivalently, $\mathbf{z} \sim \mathbf{g}^c$ for some $c \in \{1, \dots, h\}$, find c. Here $h = |\mathrm{Cl}(\mathbf{K})|$ denotes the class number of $\mathbf{K}$. Note that this case differs from all the previous ones in that the order h of the group is not known a priori.

The index calculus method can be used to find the invariants of $\mathrm{Cl}(\mathbf{K})$ (Hafner & McCurley [HM89], McCurley [Mc89], Cohen, Diaz y Diaz & Olivier [CDO92]) as well as solve the DLP in $\mathrm{Cl}(\mathbf{K})$ (Buchmann & Williams [BW90b], McCurley [Mc89], Buchmann & Düllmann [BD91]). In fact, the precomputation stage is the same for both problems. The method is described in terms of binary quadratic forms in [HM89] and [Mc89]. For our purposes, we use the language of ideals in the field as in [BW90b] and [BD91].

Before we present the index calculus method in $\mathrm{Cl}(\mathbf{K})$, let us state an observation. Let p be a rational prime such that the Legendre symbol $\left(\frac{\Delta}{p}\right) = 1$. Then the decomposition of p in $\mathbf{K}$ is $(p) = \mathbf{p}\mathbf{p}'$, where $\mathbf{p}'$ is the conjugate ideal of $\mathbf{p}$. In particular, $(p)\mathbf{p}' = \mathbf{p}^{-1}$ and hence $\mathbf{p}' \sim \mathbf{p}^{-1}$. Under certain Extended Riemann Hypotheses (ERH), it can be shown that $\mathrm{Cl}(\mathbf{K})$ is generated by the ideal classes $[\mathfrak{p}_1], \ldots, [\mathfrak{p}_B]$, where $\mathfrak{p}_i$ is either of the prime ideals dividing p_i and p_i is the i-th rational prime such that $\left(\frac{\Delta}{p_i}\right) = 1$ and $p_i < C(\log|\Delta|)^2$ for some constant $C > 0$ (a value of C is given by Bach [Ba90]). The factor base is $\mathbf{B} = \{[\mathfrak{p}_1], \ldots, [\mathfrak{p}_B]\}$.

The algorithm differs fom the previous methods in that we compute relations of the form $\prod_{i=1}^{B}[\mathfrak{p}_i]^{s_i} = [(1)]$, i.e. the right-hand side is the identity in Cl(K) rather than a power of the class $[\mathfrak{g}]$. More formally, consider the group homomorphism $\Phi\colon \mathbb{Z}^B \to \mathrm{Cl}(K)$ given by $\Phi(s_1, \ldots, s_B) = \prod_{i=1}^{B}[\mathfrak{p}_i]^{s_i}$. Any element in the kernel $\mathbf{H}$ of Φ gives rise to a factor relation. Furthermore, $\mathbf{H}$ is a sublattice of $\mathbb{Z}^B$, and since Φ is surjective, it follows that the factor group $\mathbb{Z}^B/\mathbf{H}$ is isomorphic to Cl(K) under Φ.

It can be shown that the box $\mathbf{X} = \{(x_1, \ldots, x_B) \in \mathbb{Z}^B \mid 0 \le x_i \le |\Delta| \text{ for } 1 \le i \le B\}$ contains a basis of $\mathbf{H}$. To find a relation over the factor base $\mathbf{B}$, or equivalently, a vector in $\mathbf{H}$, generate a random vector $(x_1, \ldots, x_B) \in \mathbf{X}$. Then compute a reduced ideal $\mathfrak{a}'$ in the class of $\mathfrak{a} = \prod_{i=1}^{B}\mathfrak{p}_i^{x_i}$. $\mathfrak{a}'$ can be computed in time $O(\log|\Delta|)$; for details see Williams [Wi85a].[2]

Next, we compute the norm $N(\mathfrak{a}')$ and attempt to factor it over the primes $p_1, \ldots, p_B$. If we are successful, then $[\mathfrak{a}'] = [\mathfrak{a}]$ factors completely over $\mathbf{B}$, or more exactly, we have found a representation $\mathfrak{a}' = \prod_{i=1}^{B}\mathfrak{p}_i^{y_i}\mathfrak{p}_i'^{z_i}$, where $\mathfrak{p}_i\mathfrak{p}_i' = (p_i)$ and $y_i z_i = 0$ for $1 \le i \le B$. It follows that $(1) \sim \mathfrak{a}(\mathfrak{a}')^{-1} \sim \prod_{i=1}^{B}\mathfrak{p}_i^{x_i+z_i-y_i}$, hence we have found a factor relation and $(x_1+z_1-y_1, \ldots, x_B+z_B-y_B) \in \mathbf{H}$.

[2] It can be shown (see, for example, [BW88a, Theorem 2.7]) that there are at most two reduced ideals in each ideal class. This is not true for real quadratic fields, where there is usually a very large number of reduced ideals in each class. Hence this method fails for real quadratic fields, unless it is modified to identify a particular reduced ideal, see Section 12.1

Suppose we have found a basis $\underline{e}_1, \ldots \underline{e}_B$ of $\mathbf{H}$. Then the class number h of $\mathbf{K}$ is the determinant $\det(\underline{e}_1^T, \ldots, \underline{e}_B^T)$. Furthermore, any system $\underline{d}_1, \ldots, \underline{d}_k \in \mathbf{H}$ containing h linearly independent vectors generates a sublattice $\mathbf{J}$ of $\mathbf{H}$ of finite index $(\mathbf{H}:\mathbf{J})$, and the determinant h' of the lattice, which can be found using Hermite reduction, is a multiple of h, namely $h' = h(\mathbf{H}:\mathbf{J})$. Hence the class number can be computed as follows. We first find an approximation h^* of h such that $h \leq h^* < 2h$, using the analytic class number formula. This can be done in time $O(\log |\Delta|)$ (for details, see [Mc89]). Next, we successively generate factor relations as described above, and add the corresponding vector of exponents to $\mathbf{J}$ (initially, $\mathbf{J} = \emptyset$). After each relation, we compute the determinant h' of all the vectors in $\mathbf{J}$ (initially, $h' = 0$). If $(\mathbf{H}:\mathbf{J})$ is finite and $h' \leq h$, then $h' = h$ and we are finished, otherwise generate the next relation. The running time of this algorithm is $L(|\Delta|)^{\sqrt{2}+o(1)}$.

McCurley shows how the second stage of the index calculus method can be used to solve any instance of the DLP in $\mathrm{Cl}(\mathbf{K})$ in time $L(|\Delta|)^{1+o(1)}$ [Mc89]. Buchmann and Düllmann [BD91] illustrate how knowledge of the structure of the class group can reduce the running time as follows. At the end of the class number computation, $\mathbf{J}$ contains a basis $\underline{e}_1, \ldots, \underline{e}_B$ of $\mathbf{H}$. Computing the Smith normal form $S = \mathrm{diag}(N_1, \ldots, N_l, 1, \ldots, 1)$ of the basis yields an identity of the form $S = U^{-1}(\underline{e}_1^T, \ldots, \underline{e}_B^T)V$ where S, U, V are nonsingular matrices, i.e. $S, U, V \in \mathrm{GL}_B(\mathbf{Z})$, and $N_i > 1$ for $1 \leq i \leq l$. Let $U = [u_{ij}]_{i,j=1,\ldots,B}$ and $U^{-1} = [u'_{ij}]_{i,j=1,\ldots,B}$. If $\gamma_i = \prod_{j=1}^{B} \mathfrak{p}_j^{u_{ij}}$ $(1 \leq i \leq l)$, then $\mathrm{Cl}(\mathbf{K}) = \langle[\gamma_1]\rangle \times \cdots \times \langle[\gamma_l]\rangle$ and $\mathrm{ord}([\gamma_i]) = N_i$ for $1 \leq i \leq l$. Hence the Smith normal form computation of a basis of $\mathbf{H}$ yields the invariants of $\mathrm{Cl}(\mathbf{K})$ in time $L(|\Delta|)^{\sqrt{2}+o(1)}$. Furthermore, since $\mathfrak{p}_i = \prod_{j=1}^{l} \gamma_j^{u'_{ij}}$ $(1 \leq i \leq B)$, a representation of any element in $\mathrm{Cl}(\mathbf{K})$ as a product of powers of the classes of $\mathfrak{p}_1, \ldots, \mathfrak{p}_B$ yields a representation in terms of the classes of $\gamma_1, \ldots, \gamma_l$. It should be pointed out that the Cohen-Lenstra heuristics [CL84b] imply that $l = 1$, i.e. $\mathrm{Cl}(\mathbf{K})$ is cyclic, in over 97.75 % of all cases.

To solve $\mathfrak{z}^c \sim \mathfrak{g}$ for c ($\mathfrak{g}$, $\mathfrak{z}$ ideals in $\mathbb{K}$), generate a random $(x_1, \ldots, x_B) \in \mathbb{X}$. Compute the ideal $\mathfrak{a} = \mathfrak{g}\prod_{i=1}^{B}\mathfrak{p}_i^{x_i}$ and find a reduced ideal $\mathfrak{a}'$ in $[\mathfrak{a}]$. Attempt to factor $\mathfrak{a}'$ over the factor base $\mathbb{B}$. In case of success, we obtain $\mathfrak{a}' = \prod_{i=1}^{B} \mathfrak{p}_i^{y_i}\mathfrak{p}_i'^{z_i}$, $y_i z_i = 0$ for $1 \le i \le B$. Hence $\mathfrak{g} \sim \mathfrak{a}'\prod_{i=1}^{B}\mathfrak{p}_i^{-x_i} \sim \prod_{i=1}^{B}\mathfrak{p}_i^{y_i - z_i - x_i}$, whence we obtain a relation $\mathfrak{g} \sim \prod_{i=1}^{l}\gamma_i^{j_i}$. Similarly, we find $\mathfrak{z} \sim \prod_{i=1}^{l}\gamma_i^{k_i}$. Then $(1) \sim \mathfrak{g}^c\mathfrak{z}^{-1} \sim \prod_{i=1}^{l}\gamma_i^{j_i c - k_i}$, hence we can find c by solving the system of simultaneous congruences $cj_i \equiv k_i \pmod{N_i}$ $(1 \le i \le l)$ for c, using a generalized Chinese Remainder Theorem. Buchmann and Düllmann show that given the structure of $Cl(\mathbb{K})$, the solution c of $\mathfrak{g}^c \sim \mathfrak{z}$ can be computed in time $L(|\Delta|)^{\frac{1}{2}+o(1)}$.

It should finally be noted that even though the complexity of the index calculus method is noticably better than exponential, all the above algorithms are still completely impractical if the underlying structure is large. For example, with current technology, it would be impossible to compute in reasonable time an arbitrary discrete logarithm in a finite field GF(q) where q has approximately 200 digits. Hence, cryptographic schemes based on the DLP appear to be quite secure.

8.2.3 Other Schemes Using Discrete Logarithms

From the observations in the previous section, it follows that the original Diffie-Hellman key exchange protocol in GF(p)* can be broken in subexponential time. The same is true for all its extensions to arbitrary finite fields and for the variation using the class group of an imaginary quadratic field [BW88a]. The scheme using matrix rings over GF(p) proposed by Odoni et al [OVS84] can be shown to be no more secure than the original Diffie-Hellman scheme, since discrete logarithms in its underlying group can be computed by calculating logarithms in extension fields of GF(p) (see [Od84]).

Any system using the group of residues relatively prime to an integer n which is the product of two primes ([Sh85], [Mc88]) can be shown to be at least as difficult to break as it is to factor n. Hence this variation has the advantage of remaining secure if either factoring or the DLP in Galois fields GF(p) remain intractable, and may thus provide additional security over the original Diffie-Hellman scheme.

Finally, the best known algorithms for solving the DLP in the group of points on an elliptic curve over a finite field GF(q) as well as the curves of higher genus discussed in [Ko90] and used in the scheme [Ko88] are only those algorithms that work in arbitrary finite groups presented in Section 8.2.1, with one exception. If the elliptic curve is *supersingular*, i.e. the order of the corresponding group G is $q + 1 - t$ where t is a multiple of the characteristic of GF(q), then Menezes, Okamoto and Vanstone have shown that there is a probabilistic polynomial-time algorithm (in log q) for reducing the DLP in G to the DLP in an extension field GF(q^k) of GF(q) [MOV91]. Hence the DLP in these groups is probabilistically subexponential.

There are a number of other cryptographic systems which use exponentiation in a finite field GF(q), and whose security is consequently based on the DLP in GF(q). We give a brief overview of the most well-known ones.

The Pohlig-Hellman scheme [PH78]: This private-key cryptosystem was already mentioned in Section 2.1.4 and is very similar to RSA. The secret key is a pair of integers $e, d \in \{1, \dots, p\text{-}1\}$ such that $ed \equiv 1 \pmod{p\text{-}1}$. A message M is encrypted as $C \equiv M^e \pmod{p}$, and a cryptogram is deciphered as $M \equiv C^d \pmod{p}$. Any fast DLP algorithm in GF(p)* will break this system.

The Massey-Omura cryptosystem (Massey [Ma83], Wah & Wang [WW84]): Here, any finite field GF(q) can be used. Every user i secretly chooses an integer $e_i \in \{1, \dots, q\text{-}1\}$ relatively prime to q-1, and computes $d_i \in \{1, \dots, q\text{-}1\}$ such that $e_i d_i \equiv 1 \pmod{q\text{-}1}$. If Alice wishes to send a message M to Bob, she computes $x \equiv M^{e_\mathrm{A}} \pmod{q\text{-}1}$ and transmits x to Bob. Bob computes $y \equiv x^{e_\mathrm{B}} \equiv M^{e_\mathrm{A} e_\mathrm{B}} \pmod{q\text{-}1}$ and sends y back to Alice. Alice now forms $z \equiv y^{d_A} \equiv M^{e_B}$ (mod q-1) and transmits z to Bob, who finally computes $z^{d_B} \equiv M$ (mod q-1). Here, as usual, all remainders are taken to be between 1 and q-1. Note that this scheme requires no key exchange, but 1.5 rounds of communication per message. Furthermore, caution is advised in its use, since it is extremely vulnerable to an impersonation attack. A cryptanalyst C intercepting $x \equiv M^{e_A}$ (mod q-1) could send $y' \equiv M^{e_A e_C}$ (mod q-1) back to Alice. Alice, thinking that C is Bob, returns $z' \equiv M^{e_C}$ (mod q-1) to C, who could then read $M \equiv z^{d_C}$ (mod q-1). Hence all messages should be authenticated in this system.

The ElGamal cryptosystem [El85b]: This public-key system also makes use of arithmetic in an arbitrary finite field GF(q). All users publicly agree on q and an element $g \in$ GF(q). User i's secret and public keys are a random integer $a_i \in \{1, \dots, q\text{-}2\}$ and the element $g^{a_i} \in$ GF(q), respectively. In order for Alice to send a message $M \in$ GF(q) to Bob, she chooses a random $k \in \mathbb{Z}^{>0}$, looks up Bob's public key g^{a_B}, and sends the pair $(g^k, Mg^{a_B k})$ to Bob (i.e. the message M is "masked" by g^{a_B} and the "clue" to unmask M is g^k, but the clue is only useful to Bob, since only he knows a_B). To retrieve M, Bob computes $x = (g^k)^{a_B}$ and $Mg^{a_B k}x^{-1} = M$. Note that this system requires twice the bandwidth of all the previous schemes.

The ElGamal signature scheme [El85b]: This scheme uses GF(p)* as the underlying group. The prime p and a primitive root g (mod p) are agreed upon ahead of time by all parties. Then each user i secretly generates a random integer $a_i \in \{1, \dots, p\text{-}1\}$ and publishes $x_i \equiv g^{a_i}$ (mod p), $0 < x_i < p$. To sign a message $M \in \{1, \dots, p\text{-}1\}$, Alice secretly chooses a random integer $k \in \{1, \dots, p\text{-}2\}$ such that gcd(k, p-1) = 1, and computes $r \equiv g^k$ (mod p), $0 < r < p$. Then she finds $s \in \{1, \dots, p\text{-}2\}$ such that $M \equiv a_A r + ks$ (mod p-1). Then $g^M \equiv g^{a_A r + ks} \equiv x_A{}^r r^s$ (mod p). The signature S of M is the pair $S = (r, s)$, and S can easily be verified by computing both $x_A{}^r r^s$ (mod p) and g^M (mod p) and comparing the two. To forge a signature, a cryptanalyst must find $r \equiv g^k$ (mod p) and $s \equiv (M - a_A r)k^{-1}$ (mod p-1) without knowledge of k and a_A. This signature scheme has recently been proposed for a US national standard by the National Institute of Standards and Technology (NIST).

Schnorr's identification scheme [Sc90]: This scheme, intended to be employed on smart cards, can be used to prove one's identity, and is based in the DLP in a subgroup of GF(p)*. We first require the establishment of a trusted key authentication centre (KAC) for the registration of public keys. The KAC initially chooses primes p and q such that $q \mid p$-1, an element $g \in$ GF(p)* of order q, and its own private and public keys for a signature scheme of its choice. p, q, and the KAC's public key are made public. Any user wanting to register with the KAC chooses a secret integer $s \in \{1, \ldots, q\}$, computes $v \equiv g^{-s} \pmod{p}$, and submits v along with some identification to the KAC. The KAC verifies the users identification and generates an identification string I and a signature S of the pair (I, v).

If Peggy (the *prover*) wishes to prove her identity to Vic (the *verifier*), she first chooses a random integer $r \in \{1, \ldots, q\}$ and computes $x \equiv g^r \pmod{p}$ (this need only be done once and x can be reused for subsequent proofs of identity). Peggy now sends her identification string I, her public key v, the KAC's signature S of (I, v), and x to Vic. Vic verifies the validity of Peggy's public key v by checking the signature S. He then chooses a random integer $c \in \{1, \ldots, 2^t\}$ (the *challenge*) and sends c to Peggy (here, $t \in \mathbb{Z}^{>0}$ is a security parameter which should be chosen appropriately to make the scheme sufficiently secure and at the same time efficient). Peggy returns the value $y \equiv r+sc \pmod{q}$ to Vic. Finally, Vic computes $z \equiv g^y v^c \pmod{p}$. If $z = x$, he believes that Peggy is indeed who she claims she is. An improved version of this scheme was recently given by Brickell and McCurley [BM92].

8.3 Key Exchange Without a Group Structure

An important issue concerning the design of cryptographic schemes is the problem of whether the internal structure of the underlying mathematical set may give rise to cryptanalytic attacks. For example, it may be possible for an adversary to exploit the structure of the key space in order to drastically reduce the computational effort of an exhaustive key search or mount an attack which would not be successful if a less structured set were used as a key space. The Diffie-Hellman protocol is based on the arithmetic in a very structured set, namely a group. Hence the question arises of whether the full structure of a group is really essential for conducting this kind of key exchange. A close look at the scheme reveals that we really only require a set $\mathbb{G}$ with a multiplicative operation such that the following holds:

i) $\mathbb{G}$ is closed under the operation.

ii) From their respective computations, $(g^b)^a$ and $(g^a)^b$, the two parties need to be able to agree on a common key.

In the case where $\mathbb{G}$ is a group, i) is satisfied by definition and ii) holds because $(g^b)^a = (g^a)^b$ by associativity. Our question can now be rephrased as follows: is there a set satisfying conditions i) and ii), which can be used as the basis of a Diffie-Hellman-like key exchange protocol that is both secure and efficient?

The answer to this question is yes. The first and so far the only example for such a set was given by Buchmann and Williams [BW90a]. In abandoning the group structure, their scheme not only introduces a cryptographically (and quite unexpected) idea, but also employs a mathematically interesting concept which is due to Shanks, namely the *infrastructure* of a real quadratic field [Sh72]. The Buchmann-Williams scheme uses as key space the set of reduced principal ideals of such a field. There is a price to pay for giving up the group structure. The algorithms of the new scheme are more complicated and computationally more involved than the Diffie-Hellman protocol, and in fact, the overall complexity increases from linear to quadratic. In addition, the scheme requires more bandwidth and an additional round of communication, although in the second round, each party transmits at most one bit, and in almost all cases, only one of the partners needs to send the bit.

The basic ideas of the scheme are merely sketched in [BW90a]. Very few mathematical details are given and no computational aspects are discussed. More details of the protocol can be found in Scheidler, Buchmann & Williams [SBW93]. In the following five chapters, we will review the idea of the scheme, give all the necessary algorithms, provide a detailed approximation calculus, and discuss our implementation of the protocol. Chapter 9 presents the concept of infrastructure of a real quadratic field and how it can be used for key exchange. In Chapter 10, we introduce the main algorithms underlying the protocol and discuss implementation issues. Chapter 11 addresses the problem of establishing a unique common key and gives the details of the protocol. Part II of the thesis concludes with a brief analysis the scheme's security in Chapter 12 and a discussion of our computer implementation as well as some numerical examples in Chapter 13.

9 The Infrastructure of a Real Quadratic Field

9.1 Reduced Principal Ideals and Distances

For the basics about real quadratic fields, we refer the reader to Cohn [Co62] or Hua [Hu82]. For the material about ideals and their reduction, see Williams [Wi85a] and Williams & Wunderlich [WW87].

Let be a positive sqarefree rational integer. By adjoining the square root $\sqrt{D}$ to the rationals $\mathbb{Q}$, we obtain a *real quadratic field* $\mathbb{K} = \mathbb{Q}(\sqrt{D})$. A generating polynomial for $\mathbb{K}$ is $f(x) = x^2 - D$, hence $(\mathbb{K}:\mathbb{Q}) = 2$. As in Section 8.2.2, we define

$$\sigma = \begin{cases} 1 \text{ if } D \equiv 2, 3 \pmod 4 \\ 2 \text{ if } D \equiv 1 \pmod 4 \end{cases}$$

and we set $\omega = \dfrac{\sigma-1+\sqrt{D}}{\sigma} \in \mathbb{K}$. Then the maximal order of $\mathbb{K}$ is $\mathbb{O} = \mathbb{Z}[\omega]$. Since $\sqrt{D}$ and $-\sqrt{D}$ are the roots of $f(x) = x^2 - D$, $\mathbb{K}$ has two conjugate mappings, given by $\sigma_1(\sqrt{D}) = \sqrt{D}$ (the identity) and $\sigma_2(\sqrt{D}) = -\sqrt{D}$. Furthermore, since $\sqrt{D}, -\sqrt{D} \in \mathbb{R}$, we see that $\mathbb{K}$ is a totally real field, i.e. $s = 2$ and $t = 0$. For any $\alpha = x + y\sqrt{D} \in \mathbb{K}$ $(x, y \in \mathbb{Q})$, we will denote its algebraic conjugate by $\alpha' = \sigma_2(\alpha) = x - y\sqrt{D}$. Then we have $\mathrm{Tr}(\alpha) = \alpha+\alpha' = 2x$ and $\mathrm{N}(\alpha) = \alpha\alpha' = x^2 - y^2D$.

The discriminant of $\mathbb{K}$ is given by $\Delta = \dfrac{4}{\sigma^2}D$. From $s = 2$ and $t = 0$, it follows that the unit rank of $\mathbb{K}$ is $r = 1$, hence we have a unique fundamental unit $\eta > 1$. $R = \log \eta$ is the regulator of $\mathbb{K}$.

It can be shown that every ideal $\mathfrak{a}$ in $\mathbb{O}$ has a $\mathbb{Z}$-basis $\{a, b+c\omega\}$, i.e. $\mathfrak{a} = [a, b + c\omega]$, where $a, b, c \in \mathbb{Z}$. Here a, b, and c are unique, $0 \le b < a$, $c \mid b$, $c \mid a$, $ac \mid \mathrm{N}(b + c\omega)$, and $a = \mathrm{L}(\mathfrak{a})$, the least positive raional integer in $\mathfrak{a}$. $\mathfrak{a}$ is primitive if and only if $c = 1$. The unit ideal can be written as $\mathbb{O} = [1, \omega]$, i.e. $a = c = 1$, $b = 0$.

Recall that for any integral ideal a, an integer $\alpha \in$ a-{0} is called a minimum in a if there exists no $\beta \in$ a-{0} such that $|\beta| < |\alpha|$ and $|\beta'| < |\alpha'|$. If α is a minimum in a, then $-\alpha$ is a minimum in a as well, so henceforth we will require minima to be positive. Clearly, 1 is a minimum in O. There is an iterative procedure which enables us to generate the sequence of all the minima in O such that $1 = \mu_1 < \mu_2 < \mu_3 < \cdots$ (see [Wi85a, pp. 630f.]). The method essentially computes the continued fraction expansion of ω; details are given in the next section. Since η is a minimum in O and $\eta > 1$, it follows that $\mu_{l+1} = \eta$ for some $l \in \mathbb{Z}^{>0}$, and in fact $\mu_{j+ml} = \mu_j\eta^m$ for all $j \in \mathbb{Z}^{>0}$, $m \in \mathbb{Z}$ such that $j+ml \geq 1$. If D is chosen appropriately (see Section 12.2), then l might be as large as $O(\sqrt{D} \log \log D)$.

From Definition 1.2, we recall that an integral ideal a is said to be reduced if a is primitive and L(a) is a minimum in a. If a is reduced, then L(a) $< \sqrt{\Delta}$. The set $\mathfrak{R}$ of all reduced principal ideals in O consists of exactly those ideals which are generated by a minimum in O. Thus the iterative algorithm for generating the ordered sequence $(\mu_j)_{j>0}$ gives rise to a procedure for computing an ordered sequence $\mathfrak{r}_1 = (1), \mathfrak{r}_2, \mathfrak{r}_3, \ldots$ of reduced principal ideals. Since $\mu_{j+ml} = \mu_j\eta^m$, it follows that $\mathfrak{r}_{j+ml} = \mathfrak{r}_j$ for all $j \in \mathbb{Z}^{>0}$, $m \in \mathbb{Z}$ such that $j+ml \geq 1$. Hence the sequence $(\mathfrak{r}_j)_{j>0}$ is purely periodic with period length l. If we set $M = \{1 = \mu_1, \mu_2, \ldots , \mu_l\}$, i.e. M consists of all the minima $\mu \in$ O such that $1 \leq \mu < \eta$, then we can write $\mathfrak{R} = \{\mathfrak{r}_i = (\mu_j) \mid \mu_j \in M\} = \{(1) = \mathfrak{r}_1, \mathfrak{r}_2, \ldots , \mathfrak{r}_l\}$, so $\mathfrak{R}$ is finite and of cardinality l.

We wish to make $\mathfrak{R}$ the key space of a Diffie-Hellman key exchange protocol. The most obvious approach to setting up such a scheme is for all parties to publicly agree on a real quadratic field K and a minimum $\mu \in$ M. Then $\mathfrak{g} = (\mu)$ is a reduced ideal in K. Now each party generates a positive integer a and computes the ideal $\mathfrak{g}^a = (\mu^a)$ (a should be bounded by l, but since we generally do not know l, we can simply choose a suitable bound B for a). At this point the protocol fails, since $\mathfrak{g}^a$ need not be reduced.

We will see in the next section that it is possible to obtain a reduced principal ideal $\mathfrak{r}$ from $\mathfrak{g}^a$ in $O(\log D)$ arithmetic operations. This gives rise to an operation $*$ ("multiply &

reduce"), under which $\mathfrak{R}$ is closed. Hence we can attempt to save our previous approach as follows. Alice generates a random $a \in \{1, \ldots , B\}$, computes $\mathbf{g}^a$ and a reduced ideal $\mathbf{r}_A$ equivalent to $\mathbf{g}^a$, which she sends to Bob. Similarly, Bob generates a random $b \in \{1, \ldots , B\}$, and transmits an ideal $\mathbf{r}_B$ obtained from $\mathbf{g}^b$ to Alice. From $\mathbf{r}_B$ and a, Alice computes $\mathbf{r}_B{}^a$ and from this a reduced ideal $\mathbf{k}_A$. Similarly, Bob finds a reduced ideal $\mathbf{k}_B$ from b and $\mathbf{r}_A$. At this point, our scheme fails once again, since in general $\mathbf{k}_A \neq \mathbf{k}_B$, and in fact there is no connection or common feature between $\mathbf{k}_A$ and $\mathbf{k}_B$, other than that they are both reduced principal ideals.

To overcome this second obstacle, we need to provide a means for locating ideals in $\mathfrak{R}$ in such a way that the two parties can identify a unique reduced ideal at the end of their respective computations. In order to do this, we associate with each reduced ideal $\mathbf{r}_j = (\mu_j)$, μ_j a minimum in $\mathbf{O}$, a *distance* $\delta_j = \log \mu_j$.[1] Then δ_j is a strictly monotonically increasing function, i.e. $\delta_{j+1} > \delta_j$ ($j \in \mathbb{Z}^{>0}$). Furthermore, we can define the distance between a reduced ideal $\mathbf{r}_j$ and a real number x as $\delta(\mathbf{r}_j, x) = \delta_j - x$.[2] The unique reduced principal ideal *closest* to x is the $\mathbf{r} \in \mathfrak{R}$ such that $|\delta(\mathbf{r}, x)|$ is minimal.

We can now attempt a third approach to key exchange in $\mathfrak{R}$. Alice generates a and computes and sends to Bob the reduced principal ideal $\mathbf{a}$ closest to a, i.e. $|\delta(\mathbf{a}, a)|$ is minimal. Similarly, Bob provides Alice with the ideal $\mathbf{b}$ closest to his secret b. From $\mathbf{b}$ and a, Alice computes the ideal $\mathbf{k}$ closest to ba. Likewise, Bob uses b and $\mathbf{a}$ to find the ideal $\mathbf{k}$ closet to ab. Then both parties have generated the same key ideal.

This version of the protocol is successful in theory, but causes a problem in practice. Distances are irrational numbers and need to be rationally approximated. As we will see later on, the uncertainty in our approximation prevent us from knowing for $x \in \mathbb{R}$ the

[1] This definition agrees with the distance definition given in Stephens & Williams [SW89], but differs from the one in [Wi85a] and [WW87] by an additive term of $\log L(\mathbf{r}_n)$.

[2] This definition differs from the one in [BW90a] in its sign.

unique ideal $\mathfrak{r} \in \mathfrak{R}$ closest to x. However, we can identify two possible candidates for $\mathfrak{r}$. Let j be the unique positive integer such that $\delta_j \leq x < \delta_{j+1}$. If $\mathfrak{r}_j = (\mu_j)$ where $\delta_j = \log \mu_j$, then we call $\mathfrak{r}_j$ the ideal *closest to the left* of x and denote it by $\mathfrak{r}_-(x)$. Similarly, if $\mathfrak{r}_{j+1} = (\mu_{j+1})$ where $\delta_{j+1} = \log \mu_{j+1}$, then $\mathfrak{r}_{j+1} = \mathfrak{r}_+(x)$ is the ideal *closest to the right* of x. Clearly, the ideal closest to x is either $\mathfrak{r}_-(x)$ or $\mathfrak{r}_+(x)$. For any $x \in \mathbb{R}$, we will be able to generate an ideal which is either $\mathfrak{r}_-(x)$ or $\mathfrak{r}_+(x)$.

Now we are able to present our final and functioning version of a protocol for key exchange in $\mathfrak{R}$. D (and thus $\mathbb{K}$) is agreed upon publicly ahead of time. Alice secretly chooses a positive integer a and computes a reduced ideal $\mathfrak{a} \in \{\mathfrak{r}_-(a), \mathfrak{r}_+(a)\}$ and a rational approximation $\hat{\delta}(\mathfrak{a}, a)$ of its distance $\delta(\mathfrak{a}, a)$ from a. She sends both the ideal $\mathfrak{a}$ and its approximate distance $\hat{\delta}(\mathfrak{a}, a)$ from a to Bob. Similarly, Bob secretly chooses $b \in \mathbb{Z}^{>0}$ and determines a reduced ideal $\mathfrak{b} \in \{\mathfrak{r}_-(b), \mathfrak{r}_+(b)\}$ and an approximation $\hat{\delta}(\mathfrak{b}, b)$ of $\delta(\mathfrak{b}, b)$. He transmits both $\mathfrak{b}$ and $\hat{\delta}(\mathfrak{b}, b)$ to Alice. From $\mathfrak{b}$, $\hat{\delta}(\mathfrak{b}, b)$ and a, Alice computes a reduced ideal $\mathfrak{k}_A \in \{\mathfrak{r}_-(ab), \mathfrak{r}_+(ab)\}$. Likewise, Bob determines from $\mathfrak{a}$, $\hat{\delta}(\mathfrak{a}, a)$ and b a reduced ideal $\mathfrak{k}_B \in \{\mathfrak{r}_-(ab), \mathfrak{r}_+(ab)\}$. $\mathfrak{k}_A$ and $\mathfrak{k}_B$ need not be the same ideal; in fact, the two parties do not know whether they computed the same ideal. However, the exchange of at most two more bits of information will enable them to agree on a common key ideal $\mathfrak{k} \in \{\mathfrak{r}_-(ab), \mathfrak{r}_+(ab)\}$.

Two problems arise from this scheme:

1. Given a number a, how to find an ideal $\mathfrak{a} \in \{\mathfrak{r}_-(a), \mathfrak{r}_+(a)\}$. More generally, given a real number a, an ideal $\mathfrak{b} \in \{\mathfrak{r}_-(b), \mathfrak{r}_+(b)\}$, and $\hat{\delta}(\mathfrak{b}, b)$, how to find $\mathfrak{k}_A \in \{\mathfrak{r}_-(ab), \mathfrak{r}_+(ab)\}$
2. How do the communication partners detect whether or not $\mathfrak{k}_A = \mathfrak{k}_B$ and, in case $\mathfrak{k}_A \neq \mathfrak{k}_B$, how do they agree on a common key ideal $\mathfrak{k}$.

The arithmetic and algorithms to solve the first problem will be given in the next section and in Chapter 10. Chapter 11 presents a solution to the ambiguity problem of the key ideal.

9.2 Ideal Arithmetic

Let $\mathfrak{a} = [L(\mathfrak{a}), b + \omega]$ be any primitive ideal. If we set $Q = L(\mathfrak{a})\sigma$, $P = b\sigma + \sigma - 1$, then $\mathfrak{a}$ can be written as $\mathfrak{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$ where $P, Q \in \mathbb{Z}$. Then $\sigma \mid Q$ and $L(\mathfrak{a}) \mid N(b+\omega)$ implies $\sigma Q \mid D - P^2$. In this fashion, every primitive ideal $\mathfrak{a}$ can be associated with a pair (P, Q) of rational integer *coefficients*. For the unit ideal $\mathcal{O}$, we have $P = \sigma - 1$, $Q = \sigma$.

As mentioned earlier, there is a connection between ideal arithmetic and the theory of continued fractions. As in Section 1.3, denote by $\mathbf{P}$ the set of integral principal ideals. Let $\mathfrak{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] \in \mathbf{P}$ be primitive. If we set $Q_0 = Q$, $P_0 = \mathrm{P}$, $\phi_0 = \frac{P_0 + \sqrt{D}}{Q_0}$, and expand ϕ_0 into a continued fraction as described in Algorithm 9.1 below, then we obtain a sequence of primitive principal ideals $\mathfrak{a}_2, \mathfrak{a}_3, \ldots$ where $\mathfrak{a}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right]$ $(j \in \mathbb{Z}^{>0})$. We call $\mathfrak{a}_{j+1}$ the *right neighbour* and (in the case where $j \geq 2$) $\mathfrak{a}_{j-1}$ the *left neighbour* of $\mathfrak{a}_j$. If $\mathfrak{a}_1 = \mathfrak{a}$ is reduced, then $\mathfrak{a}_1 = \mathfrak{r}_k$ for some $k \in \mathbb{Z}^{>0}$, $\mathfrak{a}_j = \mathfrak{r}_{k+j-1}$ is reduced for all $j \geq 1$, and the sequence $(\mathfrak{a}_j)_{1 \leq j \leq l}$ will generate all the ideals in $\mathfrak{R}$. In this case, we can compute reduced principal ideals by starting at any $\mathfrak{r}_i$ $(i \geq k)$ and generating $\mathfrak{r}_{i+1}, \mathfrak{r}_{i+2}, \ldots$, or, applying the recursion "backwards" as in Algorithm 9.2, $\mathfrak{r}_{i-1}, \mathfrak{r}_{i-2}, \ldots, \mathfrak{r}_1$ (for the latter sequence, we require $i \geq k+1$). In the case where $\mathfrak{a}_1$ is not reduced, this method will yield a reduced ideal after $O(\log D)$ iterations. Hence, the continued fraction algorithm allows us to step through $\mathfrak{R}$ in either direction and to quickly find for any primitive principal ideal an equivalent reduced one. The algorithm is given in [WW87] and operates as follows. Let $d = \lfloor\sqrt{D}\rfloor$.

Algorithm 9.1 (Continued fraction algorithm, forward):

Input: Any primitive ideal $\mathfrak{a} = \left[\frac{Q}{\sigma}, \frac{P+\sqrt{D}}{\sigma}\right] \in \mathbf{P}$.

Output: A sequence of primitive ideals $\mathfrak{a}_1, \mathfrak{a}_2, \ldots$ in $\mathbf{P}$, where $\mathfrak{a}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1}+\sqrt{D}}{\sigma}\right]$ $(j \geq 1)$.

Algorithm: Set $P_0 = P$, $Q_0 = Q$,

$$q_{j-1} = \left\lfloor \frac{P_{j-1} + \sqrt{D}}{Q_{j-1}} \right\rfloor, \qquad P_j = q_{j-1}Q_{j-1} - P_{j-1}, \qquad Q_j = \frac{D - P_j^2}{Q_{j-1}} \qquad (j = 1, 2, \ldots).$$

Set $\mathfrak{a}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1}+\sqrt{D}}{\sigma}\right]$. Then $\mathfrak{a}_{j+1} = \psi'_j\, \mathfrak{a}_j$ where $\psi_j = \frac{\sqrt{D} - P_j}{Q_{j-1}}$ $(j \geq 1)$. □

Algorithm 9.2 (Continued fraction algorithm, backward):

Input: Any $\mathfrak{r}_i \in \mathfrak{R}$, $\mathfrak{r}_i = \left[\frac{Q_{i-1}}{\sigma}, \frac{P_{i-1}+\sqrt{D}}{\sigma}\right]$ $(i \geq 1)$.

Output: The sequence of ideals $\mathfrak{r}_{i-1}, \mathfrak{r}_{i-2}, \ldots, \mathfrak{r}_1 = \mathcal{O}$, where $\mathfrak{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1}+\sqrt{D}}{\sigma}\right]$ $(1 \leq j \leq i)$.

Algorithm: $Q_j = \frac{D - P_{j+1}^2}{Q_{j+1}}$, $q_j = \left\lfloor \frac{P_{j+1} + d}{Q_j} \right\rfloor$, $P_j = q_jQ_j - P_{j+1}$ $(j = i\text{-}2, \ldots, 1)$.

Set $\mathfrak{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-}+\sqrt{D}}{\sigma}\right]$. Then $\mathfrak{r}_j = \phi'_j \mathfrak{r}_{j+1}$ where $\phi_j = \frac{1}{\psi_j} = \frac{P_j + \sqrt{D}}{Q_j}$, $(1 \leq j \leq i)$. □

Theorem 9.1 a) Let $\mathfrak{a} = \left[\frac{Q}{\sigma}, \frac{P+\sqrt{D}}{\sigma}\right]$ be a primitive principal ideal and let $\mathfrak{a}_1, \mathfrak{a}_2, \ldots$ be the sequence computed by Algorithm 9.1. If $0 < Q_0 < \sqrt{D}$, then for all $j \geq 1$:

i) $\mathfrak{a}_j \in \mathfrak{R}$; if $k \geq 1$ is such that $\mathfrak{a}_1 = \mathfrak{r}_k$, then $\mathfrak{a}_j = \mathfrak{r}_{k+j-1}$.

ii) $q_j \geq 1, 0 < P_j < \sqrt{D}, \sigma \leq Q_j < 2\sqrt{D}$.

iii) $-1 < \frac{1}{\psi'_j} = \frac{P_j - \sqrt{D}}{Q_j} < 0$.

iv) $q_j = \left\lfloor \frac{P_{j+1} + d}{Q_j} \right\rfloor$, so the expressions for q_j in Algorithms 9.1 and 9.2 are equivalent.

b) Let $\mathfrak{r}_i = \left[\frac{Q_{i-1}}{\sigma}, \frac{P_{i-1} + \sqrt{D}}{\sigma}\right] \in \mathfrak{R}$, $(i \geq 2)$ and let $\mathfrak{r}_{i-1}, \mathfrak{r}_{i-2}, \ldots$ be the sequence computed by Algorithm 9.2. If $0 < Q_{i-2} < \sqrt{D}$, then for all $1 \leq j \leq i\text{-}1$:

i) $\mathfrak{r}_j \in \mathfrak{R}$, $\delta_j = \delta_{j+1} + \log |\phi'_j| = \delta_{j+1} - \log |\psi'_j|$.

ii) $q_{j-1} \geq 1, 0 < P_{j-1} < \sqrt{D}, \sigma \leq Q_{j-1} < 2\sqrt{D}$.

iii) $-1 < \phi'_{j-1} = \dfrac{P_{j-1} - \sqrt{D}}{Q_{j-1}} < 0.$

Proof: a) For i) - iii), see [Wi85a, p. 632]. iv) is proved in [WW87, Lemma 6.1].

b) It is easy to see that the recursion formulae of Algorithms 9.1 and 9.2 are the same. Hence if we use Algorithm 9.1, starting with $\mathfrak{r}_1 = \mathfrak{O}$, i.e. $Q_0 = \sigma$, $P_0 = \sigma - 1$, to compute a sequence $\mathfrak{r}_1, \mathfrak{r}_2, \ldots. \mathfrak{r}_i$, then Algorithm 9.2, starting at $\mathfrak{r}_i$, computes exactly the sequence $\mathfrak{r}_i, \ldots, \mathfrak{r}_1$. This implies ii), iii), and the fact that $\mathfrak{r}_j \in \Re$ for $1 \le j \le i-1$. The second part of i) follows from $\mathfrak{r}_j = \phi'_j\mathfrak{r}_{j+1}$ and $\phi_j = \dfrac{1}{\psi_j}$ for $1 \le j \le i-1$. ❑

The gain in distance in one step and in two consecutive steps of Algorithm 9.1 is bounded.

Theorem 9.2: Let $\mathfrak{a}_1$ be as in Theorem 9.1. Then for all $j \ge 1$:

a) $q_j < |\psi'_{j+1}| < q_j + 1.$

b) $\psi'_{j+1}\psi'_j > 2.$

c) $1 + \dfrac{1}{\sqrt{\Delta}} < |\psi'_j| < \sqrt{\Delta}.$

Proof: Let $j \ge 1$. Then $\psi_{j+1} + q_j = \dfrac{\sqrt{D}-P_{j+1}}{Q_j} + q_j = \dfrac{\sqrt{D}-P_{j+1}+q_jQ_j}{Q_j} = \dfrac{\sqrt{D}+P_j}{Q_j} = \phi_j = \dfrac{1}{\psi_j}.$

a) By Theorem 9.1 a) iii), $-1 < \dfrac{1}{\psi'_j} < 0$. Then $\psi'_{j+1} + q_j = \dfrac{1}{\psi'_j}$ implies $-1 < \psi'_{j+1} + q_j < 0$, hence $q_j < -\psi'_{j+1} = |\psi'_{j+1}| < q_j + 1$.

b) $\psi'_{j+1}\psi'_j = 1 - q_j\psi'_j = 1 + q_j|\psi'_j| > 1 + 1 = 2$ by Theorem 9.1 a) ii) & iii).

c) $|\psi'_j| = \dfrac{\sqrt{D}+P_j}{Q_{j-1}} < \dfrac{2\sqrt{D}}{\sigma} = \sqrt{\Delta}$ by Theorem 9.1 a) ii), and $|\psi'_j| = -\psi'_{j-1} = q_{j-1} - \dfrac{1}{\psi'_j}$ $= q_{j-1} + \dfrac{1}{\psi'_{j-1}} > 1 + \dfrac{1}{\sqrt{\Delta}}$ by Theorem 9.1 a) ii) and the previous result. ❑

By the Gauss-Kuz'min law (see for example Khinchin [Kh64, pp. 92f.]), for almost all continued fractions, a partial quotient q occurs with probability $\log_2\left(1 + \dfrac{1}{(q+1)^2 - 1}\right)$

Hence, we expect q_j to be small in most cases, for example $q_j = 1$ in 41.5%, $q_j \leq 10$ in 87.4% of all cases.

Let $\mathfrak{a}_1$ be a primitive principal ideal and let $\mathfrak{a}_m$ ($m \in \mathbb{Z}^{>0}$) be generated from $\mathfrak{a}_1$ using Algorithm 9.1. Then $\mathfrak{a}_m = \theta'_m \mathfrak{a}_1$ where $\theta_1 = 1$, $\theta_m = \prod_{k=1}^{m-1} \psi_k$. It follows that for any fixed $i \in \mathbb{Z}^{>0}$ and $j \geq i$, we have $\mathfrak{a}_j = \frac{\theta'_j}{\theta'_i} \mathfrak{a}_i$. If we set $\zeta_m = \frac{1}{\theta_m}$, i.e. $\zeta_1 = 1$, $\zeta_m = \prod_{k=1}^{m-1} \phi_k$ ($m \geq 1$), then for $1 \leq j \leq i$, we have $\mathfrak{a}_j = \frac{\zeta'_j}{\zeta'_i} \mathfrak{a}_i$.

In the special case where $\mathfrak{a}_1 = \mathfrak{r}_1 = (1)$, i.e. $\mathfrak{a}_m = \mathfrak{r}_m$, we have $(\theta'_m) = \mathfrak{a}_m = \mathfrak{r}_m = (\mu_m)$, and in fact $\mu_m = |\theta'_m|$. The following lemma summarizes some properties of and gives a simple recurrence relation for θ_m, ζ_m ($m \geq 1$).

Lemma 9.3: For all $m \geq 1$:

a) $\theta_{m+2} = -q_m \theta_{m+1} + \theta_m$, $\qquad \zeta_{m+2} = q_m \zeta_{m+1} + \zeta_m$.

b) $|\theta'_{m+1}| > |\theta'_m| \geq 1$, $\qquad |\zeta'_{m+1}| < |\zeta'_m| \leq 1$.

c) $\operatorname{sgn}(\theta'_m) = (-1)^{m-1}$ $\qquad \operatorname{sgn}(\zeta'_m) = (-1)^{m-1}$.

Proof: We have $\theta_{m+2} = \psi_{m+1}\psi_m\theta_m$. From the proof of the previous theorem, $\psi_{m+1}\psi_m = -q_m\psi_m + 1$, whence follows the recurrence relation for θ_{m+2}. Similarly, we show $\phi_{m+1}\phi_m = q_m\phi_m + 1$, which yields the recurrence relation for ζ_{m+2}. Now $|\theta'_{m+1}| = |\psi'_m||\theta'_m|$ and by Theorem 9.1 a) iii), we have $1 > \frac{1}{|\psi'_m|}$, hence $|\theta'_{m+1}| > |\theta'_m|$. We show $\operatorname{sgn}(\theta'_m) = (-1)^{m-1}$ by induction on m. We have $\theta'_1 = 1 > 0$, $\theta'_2 = \psi'_1 = -\frac{P_1 + \sqrt{D}}{Q_0} < 0$ by Theorem 9.1 a) ii), and for $m \geq 0$, using the induction hypothesis for $m+1$ and m: $\theta'_{m+2} = -q_m(-1)^m|\theta'_{m+1}| + (-1)^{m-1}|\theta'_m| = (-1)^m(q_m|\theta'_{m+1}| + |\theta'_m|)$. The rest of the lemma follows from the identity $\zeta_m = \frac{1}{\theta_m}$. ❑

In order to find for any $x \in \mathbb{R}$ a reduced ideal $\mathfrak{r}_l \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$, we could use Algorithm 9.1, starting at $\mathfrak{r}_1 = (1)$, to generate a sequence of reduced ideals $\mathfrak{r}_2, \mathfrak{r}_3, \ldots$ with distances $\delta_2 = \log |\theta'_2|$, $\delta_3 = \log |\theta'_3|, \ldots$, until we obtain $\mathfrak{r}_l$ such that $\delta_l \leq x < \delta_{l+1}$. However, since $|\psi'_l| < 2\sqrt{D}$ for $l \geq 1$ by Theorem 9.2 c), each step advances us $O(\log D)$ in distance, hence this will require exponential computation time if x is polynomial in D. We need to move through $\mathfrak{R}$ at a much more rapid pace. To achieve this, we make use of Shanks' *infrastructure* idea ([Sh72]).

We impose an operation $*$ ("multiply & reduce") in $\mathfrak{R}$ as follows. If $\mathfrak{r}_i$, $\mathfrak{r}_j$ are reduced principal ideals with respective distances δ_i, δ_j, then $\mathfrak{r}_i * \mathfrak{r}_j$ is a reduced principal ideal $\mathfrak{r}_m$ such that $\delta_m \approx \delta_i + \delta_j$, i.e. $m \approx i + j$. Now if we want to find a reduced principal ideal $\mathfrak{r}_l$ such that $\delta_l \leq x < \delta_{l+1}$, where x is polynomial in D, we start with a reduced ideal $\mathfrak{r}_i$ with small $\delta_i = O(\log D)$. $\mathfrak{r}_i$ can be obtained using Algorithm 9.1 on $\mathfrak{r}_1 = (1)$. We then compute $\mathfrak{r}_j = \mathfrak{r}_i * \mathfrak{r}_i * \ldots * \mathfrak{r}_i$ where the number of terms is $n \approx \dfrac{x}{\delta_i}$. Then $\delta_j \approx n\delta_i \approx x$, and it can be shown that a few applications of either Algorithm 9.1 or Algorithm 9.2, starting at $\mathfrak{r}_j$, will yield $\mathfrak{r}_l$. If we adopt the fast exponentiation technique as described in Algorithm 2.1, we can compute $\mathfrak{r}_j$ using $O(\log n) = O(\log D)$ applications of $*$, hence this method is much faster than the single step method, provided the operation $*$ of two ideals can be done in time $O(\log D)$ and the computation of $\mathfrak{r}_l$ from $\mathfrak{r}_j$ requires at most $O(\log D)$ iterations of either one of Algorithms 9.1 or 9.2.

In order to define $*$ more formally, consider ideal multiplication as given in Section 1.3. Let $\mathfrak{r}_i, \mathfrak{r}_j \in \mathfrak{R}$. If we set $\mathfrak{c} = \mathfrak{r}_i\mathfrak{r}_j$, then $\mathfrak{c} = (\gamma)$ where $\log \gamma = \delta_i + \delta_j$, hence, $\mathfrak{c}$ would give us exactly our required distance. Unfortunately, $\mathfrak{c}$ need not be reduced. However, by using the reduction technique described in [WW87] (details will be given in Algorithm 9.4), we can compute a fixed reduced ideal $\mathfrak{r}_m$ which we define to be $\mathfrak{r}_i * \mathfrak{r}_j$ such that $\delta_m = \delta_i + \delta_j + \varepsilon$ where $|\varepsilon| = O(\log D)$, so $|\varepsilon|$ is usually very small relative to δ_i, δ_j. $\mathfrak{r}_m$ can be generated as follows. If we set $\mathfrak{a}_1 = \mathfrak{c}$ and apply the continued fraction algorithm as given in Algorithm 9.1 to the product ideal $\mathfrak{c} = \mathfrak{a}_1$ $O(\log D)$ times, then we obtain an ideal

$\mathfrak{a}_k$ which is reduced, i.e. $\mathfrak{a}_k = \mathfrak{r}_m$ for some $m \in \mathbb{Z}^{>0}$. Since ideal multiplication requires time $O(\log D)$, $\mathfrak{r}_i * \mathfrak{r}_j$ can in fact be computed in time $O(\log D)$.

The algorithm for ideal multiplication, which is basically Shanks' modification to Gauss' composition algorithm for quadratic forms, computes for two primitive ideals $\mathfrak{a}_1 = [a_1\ b_1+\omega]$, $\mathfrak{a}_2 = [a_2, b_2+\omega]$ the product ideal $\mathfrak{a}_3 = [a_3, b_3+\omega]$, where $a_3 = a_1a_2$ and $b_3 \equiv b_1 \pmod{a_1}$, $b_3 \equiv b_2 \pmod{a_2}$ ($\mathfrak{a}_3$ need not be primitive). The method is described in [Wi85a, p. 634] and in detail in [SW89, p. 624]. We will describe the latter version (the factor U is extracted to ensure that the product ideal $\mathfrak{c}$ is primitive).

Algorithm 9.3 (Ideal multiplication):

Input: $\mathfrak{r}_i = \left[\frac{Q_{i-1}}{\sigma}, \frac{P_{i-1} + \sqrt{D}}{\sigma}\right]$, $\mathfrak{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right] \in \mathfrak{R}\ (i, j \geq 1)$.

Output: $\mathfrak{c} \in \mathbb{P}$ primitive, $U \in \mathbb{Z}^{>0}$ such that $\mathfrak{r}_i\mathfrak{r}_j = U\mathfrak{c}$.

Algorithm:

1) Solve $\frac{Q_{i-1}}{\sigma}x_1 + \frac{Q_{j-1}}{\sigma}y_1 = Y = \gcd\left(\frac{Q_{i-1}}{\sigma}, \frac{Q_{j-1}}{\sigma}\right)$ for $x_1, y_1, Y \in \mathbb{Z}$.

2) Solve $\frac{P_{i-1}+P_{j-1}}{\sigma}x_2 + Yy_2 = U = \gcd\left(\frac{P_{i-1}+P_{i-1}}{\sigma}, Y\right)$ for $x_2, y_2, U \in \mathbb{Z}$.

3) Set $Q = \frac{Q_{i-1}Q_{j-1}}{\sigma U^2}$.

4) Set $X \equiv y_2x_1(P_{j-1}-P_{i-1}) + x_2\frac{D - P_{i-1}{}^2}{Q_{i-1}}\left(\bmod \frac{Q_{j-1}}{U}\right)$.

5) Set $P = P_{i-1} + \frac{XQ_{i-1}}{\sigma U} \pmod{Q}$. (If $U = Y$, then set $x_2 = 0, y_2 = 1$.)

6) Set $\mathfrak{c} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$. ❑

Theorem 9.4: If $\mathfrak{r}_i$, $\mathfrak{r}_j$ are such that the coeficients $Q_{i-1}, Q_{j-1}, P_{i-1}, P_{j-1}$ satisfy the bounds in Theorem 9.1 a) ii) and iii). Then Algorithm 9.3 performs $O(\log D)$ arithmetic operations on numbers requiring $O(\log D)$ bits of storage.

Proof: $Q_{i-1}, Q_{j-1}, P_{i-1}, P_{j-1} = O(\sqrt{D})$, hence all numbers throughout the algorithm are bounded by $O(D)$. Our algorithm performs a fixed number of arithmetic operations plus two applications of the Extended Euclidean Algorithm to solve the linear diophantine equations. The number of arithmetic operations performed by the Extended Euclidean Algorithm is logarithmic in its largest input number. ❑

The algorithm for reducing the product of two reduced ideals is simply the continued fraction algorithm as given in Algorithm 9.1, applied to the product ideal a number of times. The method is discussed in detail in [WW87].

Algorithm 9.4 (Ideal Reduction):

Input: $\mathfrak{a}_1 \in \mathbb{P}$ where $\mathfrak{a}_1 = \mathfrak{c} = \frac{1}{U}\mathfrak{r}_i\mathfrak{r}_j = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$ is computed by Algorithm 9.3.

Output: $\mathfrak{a}_k = \theta'_k\mathfrak{a}_1 \in \Re$, $\theta_k = (-1)^{k-1}\frac{G_{k-2}-B_{k-2}\sqrt{D}}{Q}$ such that $B_{k-2}, G_{k-2} \in \mathbb{Z}^{\geq 0}$ and $k \geq 2$.

Algorithm:

```
1) Set $Q'_0 = Q, P'_0 = P, B_{-2} = 1, B_{-1} = 0$.
2) *Repeat*, starting at $j = 1$:
       compute $q'_{j-1}, Q'_j, P'_j$ as in Algorithm 9.1
       set $B_{j-1} = q'_{j-1}B_{j-2} + B_{j-3}$
   *until* $\sigma \leq Q'_j \leq d$.
3) Compute one more quadruple $(q'_{j-1}, Q'_j, P'_j, B_{j-1})$ as in Step 2.
4) Set $k = j+1$, $\mathfrak{a}_k = \left[\frac{Q'_{k-1}}{\sigma}, \frac{P'_{k-1}+\sqrt{D}}{\sigma}\right]$, $\theta_k = (-1)^{k-1}\frac{G_{k-2} - B_{k-2}\sqrt{D}}{Q'_0}$,
   where $G_{k-2} = P'_{k-1}B_{k-2} + Q'_{k-1}B_{k-3}$. ❑
```

As soon as Q'_j is obtained such that $\sigma \leq Q'_j \leq d$, the ideal $\mathfrak{a}_k = \left[\frac{Q'_{k-1}}{\sigma}, \frac{P'_{k-1} + \sqrt{D}}{\sigma}\right]$ is reduced, so $\mathfrak{a}_k = \mathfrak{r}_m$ for some $m \in \mathbb{Z}^{>0}$. The extra iteration in Step 3 of the algorithm is to ensure that the bounds $0 < P'_{k-1} < \sqrt{D}$, $0 < Q'_{k-1} < 2\sqrt{D}$ of Theorem 9.1 are satisfied. (Note that we write P'_{j-1}, Q'_{j-1} instead of P_{j-1}, Q_{j-1} to indicate that these are the cofficients of an ideal $\mathfrak{a}_j$ which is not reduced for $j < k-1$. This notation is not to be confused with the notation $\alpha' \in \mathbb{K}$ which denotes the conjugate of α).

Lemma 9.5: Let $\mathfrak{c}$, B_i $(-2 \leq i \leq k\text{-}2)$, G_{k-2}, θ_k, $\mathfrak{a}_k$ be as in Algorithm 9.4. Then $0 \leq B_i < B_{i+1} < \frac{Q'_0}{\sqrt{D}}$ $(-2 \leq i \leq k\text{-}4)$, $B_{k-2} < (q'_{k-2} + 1)\frac{Q'_0}{\sqrt{D}}$, $G_{k-2} < 3\sqrt{D}B_{k-2}$.

Proof: $0 \leq B_i < B_{i+1}$ is clear from the recursion since all $q'_i \geq 1$ for $i \geq 1$ by Theorem 9.1 a) ii). From Theorem 4.2 in [SW89], we get $B_{k-3} < \frac{Q'_0}{\sqrt{D}}$ The bound for B_{k-2} follows from the recursion. The inequality for G_{k-2} can be obtained using the bounds on P'_{k-1}, Q'_{k-1} in Theorem 9.1 a) ii) (it is at this point that we need the extra iteration in Step 3 of Algorithm 9.4). ❑

Theorem 9.6: If $\mathfrak{c} = \frac{1}{U}\mathfrak{r}_i\mathfrak{r}_j$ where the coefficients of $\mathfrak{r}_i$ and $\mathfrak{r}_j$ satisfy the bounds of Theorem 9.4, then Algorithm 9.4 performs $O(\log D)$ arithmetic operations on numbers of $O(\log D)$ bits.

Proof: From Algorithm 9.3, we have $Q'_0 = O(D)$, $P'_0 = O(D)$ where $\mathfrak{c} = \mathfrak{a}_1 = \left[\frac{Q'_0}{\sigma}, \frac{P'_0 + \sqrt{D}}{\sigma}\right]$. If $\mathfrak{a}_k = \left[\frac{Q'_{k-1}}{\sigma}, \frac{P'_{k-1} + \sqrt{D}}{\sigma}\right]$, then it follows from Theorem 4.1 and Corollary 4.1.1 in [SW89] that $|P'_i| < \sqrt{D} + Q'_0$, $|Q'_i| \leq Q'_0$ for $0 \leq i \leq k\text{-}2$. Theorem 9.1 a) ii) yields $P'_{k-1}, Q'_{k-1} = O(\sqrt{D})$. From Lemma 9.5, we obtain $B_i = O(\sqrt{D})$ $(-2 \leq i \leq k\text{-}3)$ and $B_{k-2} < (q'_{k-2} + 1)B_{k-3} = O(D^{3/2})$, since $q'_{k-2} \leq \frac{P'_{k-2} + \sqrt{D}}{Q'_{k-2}} = O(D)$, and $G = O(D^2)$. (Indeed, by the Gauss-Kuz'min law, q'_{k-2} will be small most of the time, so generally, we have $|P'_{k-2}| = O(\sqrt{D})$, $B_{k-2} = O(\sqrt{D})$, $G_{k-2} = O(D)$). Hence all numbers in the algorithm are by a fixed power of D. By [WW87, Corollary 4.2.1], the maximum number of iterations is $O\left(\log \frac{|Q'_0|}{\sqrt{D}}\right) = O(\log D)$. ❑

Theorem 9.7: Let $\mathfrak{c} = \frac{1}{U}\mathfrak{r}_i\mathfrak{r}_j$ where $\mathfrak{r}_i$, $\mathfrak{r}_j$ are as in Theorem 9.4 and let $\mathfrak{a}_k$ be the reduced ideal computed from $\mathfrak{c} = \mathfrak{a}_1$ using Algorithm 9.4. If $\mathfrak{a}_k = \mathfrak{r}_m$, then $\delta_m = \delta_i + \delta_j + \varepsilon$ where $|\varepsilon| = \log D + O(1)$.

Proof: We have From Algorithm 9.3: $Q'_0 = \frac{Q_{i-1}Q_{j-1}}{\sigma U^2}$, so $\frac{|\theta'_k|}{U} = \frac{G_{k-2} + B_{k-2}\sqrt{D}}{Q_{i-1}Q_{j-1}}\sigma U$. Since Step 3 of Algorithm 9.4 ensures that $k \geq 2$, we have $B_{k-2} \geq 1$, so $\frac{|\theta'_k|}{U} > \frac{G_{k-2} + B_{k-2}\sqrt{D}}{4D} \geq \frac{1}{4\sqrt{D}}$, and from Lemma 9.5: $\frac{|\theta'_k|}{U} \leq |\theta'_k| < \frac{3B_{k-2}\sqrt{D}+B_{k-2}\sqrt{D}}{Q'_0} = \frac{4B_{k-2}\sqrt{D}}{Q'_0} < 4(q'_{k-2} + 1) = O(D)$. Set $\varepsilon = \log\frac{|\theta'_k|}{U}$, then $|\varepsilon| = \log D + O(1)$. If $\mathfrak{r}_i = (\mu_i)$, $\mathfrak{r}_j = (\mu_j)$, and if $\mathfrak{a}_k = \mathfrak{r}_m = (\mu_m)$, then $\mathfrak{r}_m = \theta'_k\mathfrak{c} = \frac{\theta'_k}{U}\mathfrak{r}_i\mathfrak{r}_j.$, so $\mu_m = \frac{|\theta'_k|}{U}\mu_i\mu_j$, hence $\delta_m = \log \mu_m = \log\left(\frac{|\theta'_k|\mu_i\mu_j}{U}\right) = \varepsilon + \delta_i + \delta_j$. ❑

Let $x, y \in \mathbb{R}$. Our next goal is to compute efficiently from ideals $\mathfrak{r}_i \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$ and $\mathfrak{r}_j \in \{\mathfrak{r}_-(y), \mathfrak{r}_+(y)\}$ an ideal $\mathfrak{r}_n \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$. To achieve this, we first compute $\mathfrak{c} = \frac{1}{U}\mathfrak{r}_i\mathfrak{r}_j$ and a reduced ideal $\mathfrak{r}_m$ such that $\delta_m = \delta_i + \delta_j + \varepsilon$, $|\varepsilon| = \log D + O(1)$, in time $O(\log D)$, using Algorithms 9.3 and 9.4. Then $\delta(\mathfrak{r}_m, x+y) = \delta_m - x - y = \delta(\mathfrak{r}_i, x) + \delta(\mathfrak{r}_j, y) + \varepsilon$, and $\mathfrak{r}_m$ need not yet be our correct ideal $\mathfrak{r}_n$. However, $\mathfrak{r}_n$ can be computed from $\mathfrak{r}_k$ using one of Algorithms 9.1 or 9.2. It remains to be shown that this requires no more than $O(\log D)$ iterations of either of these algorithms. We prove this result and discuss the details of the algorithm in the next section.

10 The Main Algorithms

10.1 Preliminaries for the Implementation

Since the evaluation of logarithms is computationally expensive and hence to be avoided, we introduce *exponential* distances. If $\mathfrak{r}_j = (\mu_j)$ is any reduced ideal with distance δ_j, then define its exponential distance as simply $e^{\delta_j} = \mu_j$. Similarly, if $x \in \mathbb{R}^{>0}$, we define

$$\lambda(\mathfrak{r}_j, x) = e^{\delta(\mathfrak{r}_j, x)} = \mu_j e^{-x}.$$

As indicated before, distances are generally irrational numbers, which need to be rationally approximated in our algorithms. More specifically, we approximate a distance $\lambda(\mathfrak{r}, x) \in \mathbb{R}$ by $\hat{\lambda}(\mathfrak{r}, x) \in \mathbb{Q}$ with a fixed *precision* of p bits, i.e. we write

$$\hat{\lambda}(\mathfrak{r}, x) = \frac{M(\mathfrak{r}, x)}{2^p},$$

where $M(\mathfrak{r}, x) \in \mathbb{Z}^{>0}$. We define the *relative error*

$$\rho(\mathfrak{r}, x) = \frac{\hat{\lambda}(\mathfrak{r}, x)}{\lambda(\mathfrak{r}, x)}.$$

We denote by $\hat{\mathfrak{r}}(x)$ the ideal actually computed by our algorithm, so we always have $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$. For abbreviation, let $\lambda(x) = \lambda(\hat{\mathfrak{r}}(x), x)$, $\hat{\lambda}(x) = \hat{\lambda}(\hat{\mathfrak{r}}(x), x) = \frac{M(x)}{2^p}$, $\rho(x) = \frac{\hat{\lambda}(x)}{\lambda(x)}$.

The following lemma is an immediate consequence of Theorem 9.2 a) and c).

Lemma 10.1: Let $x \in \mathbb{R}$ and let $j > 0$ be such that $\mathfrak{r}_-(x) = \mathfrak{r}_j$, $\mathfrak{r}_+(x) = \mathfrak{r}_{j+1}$.

a) $\frac{1}{q_{j-1} + 1} < \lambda(\mathfrak{r}_-(x), x) \leq 1 < \lambda(\mathfrak{r}_+(x), x) < q_{j-1} + 1.$

b) $\lambda(\mathfrak{r}_-(x), x) > \frac{1}{\sqrt{\Delta}}, \quad \lambda(\mathfrak{r}_+(x), x) < \sqrt{\Delta}.$

Proof: We have $\mathfrak{r}_+(x) = \psi'_j \mathfrak{r}_-(x)$. The inequlities $\lambda(\mathfrak{r}_-(x), x) \leq 1 < \lambda(\mathfrak{r}_+(x), x)$ are clear. If we show $\lambda(\mathfrak{r}_+(x), x) < |\psi'_j|$ and $\lambda(\mathfrak{r}_-(x), x) > \frac{1}{|\psi'_j|}$, then the Lemma follows immediately from Theorem 9.2 a) and c). From Theorem 9.1 b) i) $\mu_{j+1} = |\psi'_j|\mu_j$. Now $\lambda(\mathfrak{r}_+(x), x) = \mu_{j+1}e^{-x} = |\psi'_j|\mu_j e^{-x} = |\psi'_j|\lambda(\mathfrak{r}_-(x), x) < |\psi'_j|$ and $\lambda(\mathfrak{r}_-(x), x) > \dfrac{\lambda(\mathfrak{r}_+(x), x)}{|\psi'_j|} > \dfrac{1}{|\psi'_j|}$. □

For our implementation, we need to define a number of constants and state their properties. Let $B \in \mathbb{Z}^{\geq 2}$ be an upper bound on the secretly chosen exponents a, b such that B is polynomial bounded in D. Set

$$d^* = \lceil 2^p\sqrt{D}\,\rceil, \qquad \chi = 1 + \frac{1}{2^{p-1}}, \qquad g = 1 + \frac{1}{47d},$$

$$\gamma = \lceil g^{-1}2^p \rceil, \qquad K = \left(\frac{\chi^2}{1 - \gamma^{-1}}\right)^2, \qquad A = g^{\frac{1}{16B^2}}.$$

Also recall that $d = \lfloor D \rfloor$ and $\Delta = \dfrac{4}{\sigma^2}D$ is the discriminant of $\mathbb{K}$. For our computation, we require

$$2^p \geq 3072dB^2,$$

i.e. our precision is polynomial in D. For example, if a, b are bounded by $d = \lfloor\sqrt{D}\rfloor$, we must carry $\Omega(D^{\frac{3}{2}})$ bits of precision; a bound of $\lfloor\sqrt[4]{D}\rfloor$ on a and b requires a precision of $\Omega(D)$ bits, etc. Furthermore, γ will be a lower bound for all our approximate distances $\mathrm{M}(\mathfrak{a}, x)$ throughout our protocol. Then the following inequalities hold.

Lemma 10.2: a) $\gamma > 1$. b) $K > 1$.

c) $\chi + 2^{-p} < 1 + \dfrac{1}{2^{p-2}} < A < g$. d) $\chi^2\left(1 + \dfrac{g}{2^p}\right) < \sqrt{K} < A^3$.

e) $g^7 < 1 + \dfrac{1}{\sqrt{\Delta}}$ f) $\dfrac{(1 + 2^{-p})g^6}{1 - g^3 2^{-p}} < 1 + \dfrac{1}{\sqrt{\Delta}}$.

Proof: a) Clearly $g < 2$. $\gamma \geq g^{-1}2^p > 2^{p-1} \geq 1$.

b) Clear from $\chi > 1$ and $\dfrac{1}{1 - \gamma^{-1}} = \dfrac{\gamma}{\gamma - 1} > 1$ since $\gamma > 1$ by a).

c) $\chi + 2^{-p} = 1 + \dfrac{1}{2^{p-1}} + \dfrac{1}{2^p} = 1 + \dfrac{3}{2^p} < 1 + \dfrac{1}{2^{p-2}}$. To prove the inequality $1 + \dfrac{1}{2^{p-2}} < A$, note that $\left(1 + \dfrac{1}{2^{p-2}}\right)^{16B^2} = \left(1 + \dfrac{1}{2^{p-2}}\right)^{2^{p-2}\frac{16B^2}{2^{p-2}}} < \exp\left(\dfrac{16B^2}{2^{p-2}}\right)$. Since $\log\left(1 + \dfrac{1}{x}\right) > \dfrac{1}{1+x}$

for $x > 1$, it follows that $\log(g) \geq \frac{1}{47d+1} \geq \frac{1}{48d} \geq \frac{16B^2}{2^{p-2}}$. So $A^{16B^2} = g \geq \exp\left(\frac{16B^2}{2^{p-2}}\right)$ $> \left(1 + \frac{1}{2^{p-2}}\right)^{16B^2}$.

d) We have $\gamma < \frac{2^p}{g} + 1$, so $\gamma - 1 < \frac{2^p}{g}$ and $1 + \frac{g}{2^p} < 1 + \frac{1}{\gamma - 1} = \frac{1}{1 - \gamma^{-1}}$, hence $\chi^2\left(1 + \frac{g}{2^p}\right) < \sqrt{K}$. To prove $K < A^6$, we observe that $1 - \gamma^{-1} \geq 1 - \frac{g}{2^p}$ and $g < 2$, so it follows that $K \leq \frac{\left(1 + \frac{1}{2^{p-1}}\right)^4}{\left(1 - \frac{g}{2^p}\right)^2} < \frac{\left(1 + \frac{1}{2^{p-1}}\right)^4}{\left(1 - \frac{1}{2^{p-1}}\right)^2} < \frac{1}{\left(1 - \frac{1}{2^{p-1}}\right)^6}$ using $1 + x < \frac{1}{1-x}$ if $|x| < 1$.

Now $0 < 1 = 2^{p-1} - 2^{p-2} - 1$, so division by 2^{2p-3} we obtain $0 < \frac{1}{2^{p-2}} - \frac{1}{2^{p-1}} - \frac{1}{2^{2p-3}}$ or $1 < 1 + \frac{1}{2^{p-2}} - \frac{1}{2^{p-1}} - \frac{1}{2^{p-1}2^{p-2}} = \left(1 - \frac{1}{2^{p-1}}\right)\left(1 + \frac{1}{2^{p-2}}\right)$. Therefore $\frac{1}{1 - \frac{1}{2^{p-1}}} < 1 + \frac{1}{2^{p-2}}$ and $K \leq \left(1 + \frac{1}{2^{p-2}}\right)^6 < A^6$ by c).

e) $g^7 = \left(1 + \frac{1}{47d}\right)^7 = 1 + \frac{7}{47d} + \sum_{i=2}^{7} \binom{7}{i} \frac{1}{(47d)^i}$. Now $\binom{7}{i} \leq \binom{7}{3} = 35$ and $\frac{1}{(47d)^i}$ $\leq \frac{1}{(47d)^2} \leq \frac{1}{47^2 d}$ for $2 \leq i \leq 7$. So $g^7 < 1 + \frac{7}{47d} + \frac{6 \cdot 35}{47^2 d} \leq 1 + \frac{7 \cdot 47 + 6 \cdot 35}{47^2 d} = 1 + \frac{539}{2209d}$. Now $4d \geq 2(d+1) > 2\sqrt{D} \geq \sqrt{\Delta}$, so $g^7 < 1 + \frac{4 \cdot 539}{2209\sqrt{\Delta}} < 1 + \frac{1}{\sqrt{\Delta}}$.

f) We first observe that $g = 1 + \frac{1}{47d} \leq 1 + \frac{1}{47} = \frac{48}{47}$, hence $g^4 \leq \left(\frac{48}{47}\right)^4 < 2$. It follows that $2^p(g-1) - g^4 > 2^p(g-1) - 2 = \frac{2^p}{47d} - 2 \geq \frac{3072dB^2}{47d} - 2 \geq \frac{3072}{48} - 2 = 62 > 1$, hence $g2^p - g^4 >$ $1 + 2^p$, or $g(1 - g^3 2^{-p}) > 2^{-p} + 1$. Therefore $g > \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}$ and $\frac{(1 + 2^{-p})g^6}{1 - g^3 2^{-p}} < g^7 < 1 + \frac{1}{\sqrt{\Delta}}$ by e). ❑

10.2 The Algorithms

Our first algorithm in this section takes two input ideals $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$ and $\hat{\mathfrak{r}}(y) \in \{\mathfrak{r}_-(y), \mathfrak{r}_+(y)\}$ $(x, y \in \mathbb{R})$ and computes $\hat{\mathfrak{r}}(x+y) \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$. The second algorithm computes from a positive integer m and $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$ an ideal $\hat{\mathfrak{r}}(mx) \in \{\mathfrak{r}_-(mx), \mathfrak{r}_+(mx)\}$.

Algorithm 10.1: *Input:* $\hat{\mathfrak{r}}(x), \hat{\mathfrak{r}}(y) \in \mathfrak{R}$, $\mathrm{M}(x), \mathrm{M}(y)$ $(x, y \in \mathbb{R})$ such that

i) $\mathrm{M}(x), \mathrm{M}(y) \geq \gamma$,

ii) $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$, $\hat{\mathfrak{r}}(y) \in \{\mathfrak{r}_-(y), \mathfrak{r}_+(y)\}$,

iii) $g^{-1} \leq \rho(x)\rho(y) \leq g$.

Output: $\hat{\mathfrak{r}}(x+y) \in \mathfrak{R}$ such that $\hat{\mathfrak{r}}(x+y) \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$, $\mathrm{M}(x+y) \geq \gamma$.

Algorithm:

1. Compute $U \in \mathbb{Z}^{>0}$, $\mathfrak{c} = \left[\frac{Q'_0}{\sigma}, \frac{P'_0 + \sqrt{D}}{\sigma}\right] \in \mathbb{P}$, $\mathfrak{c} = \frac{1}{U}\hat{\mathfrak{r}}(x)\hat{\mathfrak{r}}(y)$ using Algorithm 9.3.

2. Set $\mathfrak{a}_1 = \mathfrak{c}$ and compute $\mathfrak{a}_k = \left[\frac{Q'_{k-1}}{\sigma}, \frac{P'_{k-1} + \sqrt{D}}{\sigma}\right] \in \mathfrak{R}$, $\theta_k = (-1)^{k-1}\frac{G_{k-2} - B_{k-2}\sqrt{D}}{Q'_0}$

   such that $\mathfrak{a}_k = \theta'_k\mathfrak{c}$ and $G_{k-2}, B_{k-2} \geq 0$, using Algorithm 9.4.

   {*Then* $\mathfrak{a}_k = \frac{\theta'_k}{U}\hat{\mathfrak{r}}(x)\hat{\mathfrak{r}}(y)$ *and* $\mathfrak{a}_k = \mathfrak{r}_m = \left[\frac{Q_{m-1}}{\sigma}, \frac{P_{m-1} + \sqrt{D}}{\sigma}\right]$ *for some* $m \in \mathbb{Z}^{>0}$.}

3. Set $T = \left\lceil 2^{2p}\frac{G_{k-2}2^p + B_{k-2}d^*}{Q} \right\rceil$, $\hat{\theta} = \frac{T}{2^{3p}}$, $L = g\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y)$.

4. a) *If* $1 \leq L \leq g^3$, *then* {*Case 1*}

   set $\hat{\mathfrak{r}}(x+y) = \mathfrak{r}_m$, $\mathrm{M}(x+y) = \left\lceil \frac{\hat{\theta}}{U}2^p\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{L2^p}{g} \right\rceil$.

   b) *If* $L < 1$, *then* {*Case 2*}

   compute q_{m-1}, P_m, Q_m using Algorithm 9.1

   set $T_m = 2^p$, $T_{m+1} = \left\lceil \frac{P_m2^p + d^*}{Q_{m-1}} \right\rceil$

   repeat, starting at $j = m+2$:

   compute $q_{j-2}, P_{j-1}\, Q_{j-1}$, using Algorithm 9.1, and $T_j = q_{j-2}T_{j-1} + T_{j-2}$

   until $T_j \geq \frac{2^p}{L} > T_{j-1}$

   set $n = j$, $\hat{\mathfrak{r}}(x+y) = \mathfrak{r}_n$, $\mathrm{M}(x+y) = \left\lceil \frac{T_n\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{T_nL}{g} \right\rceil$.

```
c) If $L > g^3$, then {Case 3}
    set $T_m = 2^p - 1$, $T_{m-1} = \left\lceil \frac{(P_{m-1}2^p + d^*)2^{p-1}}{Q_{m-2}(2^{p-1}+1)} \right\rceil = \left\lceil \frac{P_{m-1}2^p + d^*}{\chi Q_{m-2}} \right\rceil$
    repeat, starting at $j = m-2$:
        compute $Q_j$, $q_j$, $P_j$, using Algorithm 9.2 and $T_j = q_jT_{j+1} + T_{j+2}$
    until $T_{j-1} > L2^p \geq T_j$
    set $n = j$, $\hat{\mathfrak{r}}(x+y) = \mathfrak{r}_n$, $\mathrm{M}(x+y) = \left\lceil \frac{\hat{\theta}2^{2p}}{UT_n}\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{L2^{2p}}{gT_n} \right\rceil$. □
```

Theorem 10.3: Algorithm 10.1 computes $\hat{\mathfrak{r}}(x+y)$ and $\mathrm{M}(x+y)$ such that $\hat{\mathfrak{r}}(x+y) \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$ and $\mathrm{M}(x+y) \geq \gamma$.

Proof: Once we have computed the product ideal $\mathfrak{a}_1 = \mathfrak{c}$ in Step 1 and from $\mathfrak{a}_1$ a reduced ideal $\mathfrak{a}_k = \mathfrak{r}_m$, we need to step through $\Re$ in the appropriate direction, starting at $\mathfrak{r}_m$, to move close to $x+y$ and obtain $\hat{\mathfrak{r}}(x+y) = \mathfrak{r}_n \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$. To see that $\mathfrak{r}_n$ is indeed the correct ideal, it suffices to show that $\lambda(\mathfrak{r}_{n-1}, x+y) < 1 < \lambda(\mathfrak{r}_{n+1}, x+y)$.

Since $\lambda(\mathfrak{a}_k, x+y) = \frac{|\theta'_k|}{U}\lambda(x)\lambda(y)$, we see that $\hat{\theta}$ and L are approximations for $|\theta'_k|$ and $\lambda(\mathfrak{a}_k, x+y)$, respectively. For simplicity, let $\theta = |\theta'_k|$, $G = G_{k-2}$, $B = B_{k-2}$, $Q = Q'_0$. We first prove

$$\theta \leq \hat{\theta} < \chi\theta. \tag{1}$$

Proof of (1): $$2^{3p}\theta = \frac{2^{3p}G + 2^{3p}B\sqrt{D}}{Q} \leq 2^{2p}\frac{G2^p + Bd^*}{Q} \leq T < 2^{2p}\frac{G2^p + Bd^*}{Q} + 1$$

$$< 2^{2p}\frac{G2^p + B(2^p\sqrt{D} + 1)}{Q} + 1 = 2^{3p}\theta + 2^{2p}\frac{B}{Q} + 1.$$

Now $2^{3p}\chi = 2^{3p} + 2^{2p+1}$ and $2^{2p} \geq 3072^2d^2B^4 \geq \left(\frac{3072}{2}\right)^2(2d)^2 \geq \left(\frac{3072}{2}\right)^2 D$, since $D = (\sqrt{D})^2 < (d+1)^2 \leq (2d)^2$. Furthermore, Step 3 of Algorithm 9.4 guarantees that $k \geq 2$, so $B \geq B_0 \geq 1$ and $G \geq P_{-k-1}B \geq 1$. Therefore $2^{2p+1}\frac{B}{Q} \geq \frac{2^{2p+1}}{Q} > \frac{2^{2p+1}}{4D} \geq \frac{1}{2}\left(\frac{3072}{2}\right)^2 > 1$

and trivially $2^{2p+1}\frac{B\sqrt{D}}{Q} > 2^{2p}\frac{B}{Q}$. It follows that $T < 2^{3p}\theta + 2^{2p}\frac{B}{Q} + 1 < 2^{3p}\theta + 2^{2p+1}\frac{B\sqrt{D}}{Q} + 2^{2p+1}\frac{G}{Q} = 2^{3p}\theta + 2^{2p+1}\theta = 2^{3p}\chi\theta$, so (1) holds.

We also have

$$\frac{L}{g^3} < \lambda(\mathbf{a}_k, x+y) \le L. \qquad (2)$$

Proof of (2): $\lambda(\mathbf{a}_k, x+y) = \frac{\theta}{U}\lambda(x)\lambda(y) = \frac{\theta\hat{\lambda}(x)\hat{\lambda}(y)}{U\rho(x)\rho(y)} = \frac{\theta L}{\hat{\theta}g\rho(x)\rho(y)}$, so since $\chi < g$ by Lemma 10.2 c): $\frac{L}{g^3} < \frac{L}{g^2\chi} < \frac{L}{g\chi\rho(x)\rho(y)} < \lambda(\mathbf{a}_k, x+y) \le \frac{L}{g\rho(x)\rho(y)} \le L$.

The bounds on $\lambda(\mathbf{a}_k, x+y)$ show that our three cases $1 \le L \le g^3, L < 1, L > g^3$ correspond to $\lambda(\mathbf{r}_m, x+y) \approx 1, \lambda(\mathbf{r}_m , x+y) < 1, \lambda(\mathbf{r}_m, x+y) > 1$, respectively. For each case, we need to show $\mathrm{M}(\mathbf{r}_n, x+y) \ge \gamma$ and $\lambda(\mathbf{r}_{n-1}, x+y) < 1 < \lambda(\mathbf{r}_{n+1}, x+y)$.

Case 1: Here we have $\frac{1}{g^3} < \lambda(\mathbf{r}_m, x+y) \le g^3$, so by Theorem 9.2 c) and Lemma 10.2 e):

$\lambda(\mathbf{r}_{m-1}\ x+y) = \lambda(\mathbf{r}_m, x+y)|\psi'_{m-1}|^{-1} < g^3\left(1 + \frac{1}{\sqrt{\Delta}}\right)^{-1} < g^{-4} < 1$ and

$\lambda(\mathbf{r}_{m+1}, x+y) = |\psi'_m|\lambda(\mathbf{r}_m, x+y) > \left(1 + \frac{1}{\sqrt{\Delta}}\right)g^{-3} > g^4 > 1.$

Hence $\mathbf{r}_m \in \{\mathbf{r}_-(x+y), \mathbf{r}_+(x+y)\}$. Furthermore, $\mathrm{M}(x+y) \ge \left\lceil \frac{2^p}{g} \right\rceil = \gamma$.

Case 2: Here $\lambda(\mathbf{r}_m, x+y) < 1$, so we need to compute right neighbours of $\mathbf{r}_m$ in order to increase distance and move closer to $x+y$. Note that if $\mathbf{r}_j = \frac{\theta'_j}{\theta'_m}\mathbf{r}_m$, then $\frac{T_j}{2^p}$ is an approximation for $\left|\frac{\theta'_j}{\theta'_m}\right| = \prod_{i=m}^{j-1}|\psi'_i|$ $(j \ge m)$ and this expression increases as j increases. Hence, if $T_j \approx \frac{2^p}{L}$, then $1 \approx \frac{T_j}{2^p}L \approx \left|\frac{\theta'_j}{\theta'_m}\right|\lambda(\mathbf{r}_m, x+y) = \lambda(\mathbf{r}_j, x+y)$. We have

$$\left|\frac{\theta'_j}{\theta'_m}\right| \le \frac{T_j}{2^p} < \chi\left|\frac{\theta'_j}{\theta'_m}\right| \quad (j \ge m). \qquad (3)$$

Proof of (3) by induction on j. The case $j = m$ is $2^p = T_m < \chi 2^p$. For $j = m+1$, we have

$$2^p\left|\frac{\theta'_{m+1}}{\theta'_m}\right| \le 2^p|\psi'_m| < \frac{P_m 2^p + d^*}{Q_{m-1}} \le T_{m+1} < \frac{P_m 2^p + d^*}{Q_{m-1}} + 1 < \frac{P_m 2^p + \sqrt{D}2^{p+1}}{Q_{m-1}} + 1$$

$$= 2^p|\psi'_m| + \frac{1}{Q_{m-1}} + 1 < 2^p|\psi'_m| + 2 = \chi 2^p|\psi'_m| = \chi 2^p\left|\frac{\theta'_{m+1}}{\theta'_m}\right|.$$

From Lemma 9.3 a) and c), it follows that $\left|\frac{\theta'_{j+2}}{\theta'_m}\right| = q_j\left|\frac{\theta'_{j+1}}{\theta'_m}\right| + \left|\frac{\theta'_j}{\theta'_m}\right|$, hence, since all $q_j \geq 1$ for $j \geq m$, (3) follows trivially from the linear recursion for T_j.

Now $\lambda(\mathbf{r}_j, x+y) = \left|\frac{\theta'_j}{\theta'_m}\right|\lambda(\mathbf{r}_m, x+y) = \frac{|\theta'_j|2^p}{|\theta'_m|T_j}\frac{\theta}{\hat{\theta}}\frac{T_jL}{2^p}\frac{1}{g\rho(x)\rho(y)}$ for all $j \geq m$, so from (3), (1), the end condition of the *repeat* loop in Step 4 b), and condition iii) of Algorithm 10.1:

$\lambda(\mathbf{r}_{n-1}, x+y) = \frac{|\theta'_{n-1}|2^p}{|\theta'_m|T_{n-1}}\frac{\theta}{\hat{\theta}}\frac{T_{n-1}}{L2^p}\frac{1}{g\rho(x)\rho(y)} < 1\cdot1\cdot1\cdot\frac{1}{g}g = 1$ and as in case 1:

$$\lambda(\mathbf{r}_{n+1}, x+y) > \left(1 + \frac{1}{\sqrt{\Delta}}\right)\frac{|\theta'_n|2^p}{|\theta'_m|T_n}\frac{\theta}{\hat{\theta}}\frac{T_nL}{2^p}\frac{1}{g\rho(x)\rho(y)} > g^7\frac{1}{\chi}\frac{1}{\chi}\cdot1\cdot\frac{1}{g^2} = \frac{g^5}{\chi^2} > g^3 > 1,$$

hence $\mathbf{r}_n \in \{\mathbf{r}_-(x+y), \mathbf{r}_+(x+y)\}$. Furthermore $\mathrm{M}(x+y) \geq \left\lceil \frac{2^p}{g} \right\rceil = \gamma$.

Case 3: In this final case $\lambda(\mathbf{r}_m, x+y) > 1$, so we need to compute left neighbours of $\mathbf{r}_m$ in order to decrease distance and move closer to $x+y$. Here, if $\mathbf{r}_j = \frac{\zeta'_m}{\zeta'_j}\mathbf{r}_m$, then $\frac{T_j}{2^p}$ is an approximation for $\left|\frac{\zeta'_j}{\zeta'_m}\right| = \left|\frac{\theta'_m}{\theta'_j}\right| = \prod_{i=j}^{m-1}|\psi'_i|$ $(j \leq m)$, and this expression increases as j decreases. Hence, if $T_n \approx L2^p$, then $1 \approx \frac{2^pL}{T_n} \approx \left|\frac{\zeta'_m}{\zeta'_n}\right|L \approx \left|\frac{\zeta'_m}{\zeta'_n}\right|\lambda(\mathbf{r}_m, x+y) = \lambda(\mathbf{r}_n, x+y)$.

We show by induction on j $(j = m, m-1, \ldots)$:

$$\frac{1}{\chi}\left|\frac{\zeta'_j}{\zeta'_m}\right| < \frac{T_j}{2^p} < \left|\frac{\zeta'_j}{\zeta'_m}\right| \quad (j \leq m). \qquad (4)$$

Proof of (4): We have $\chi 2^p = 2^p + 2$, so $\frac{2^p}{\chi} = 2^p - \frac{2}{\chi}$.

Therefore, since $\chi < 2$: $\frac{2^p}{\chi} < 2^p\text{-}1 = T_m < 2^p$, whence follows the case $j = m$. For $j = m\text{-}1$:

$$\begin{aligned}\frac{2^p}{\chi}\left|\frac{\zeta'_{m-1}}{\zeta'_m}\right| &= \frac{2^p}{\chi}|\psi'_{m-1}| \leq \frac{d^*\text{-}P_{m-1}2^p}{\chi Q_{m-2}} \leq T_{m-1} < \frac{P_{m-1}2^p+d^*}{\chi Q_{m-2}} + 1\\ &< \frac{P_{m-1}2^p+\sqrt{D}2^{p+1}}{\chi Q_{m-2}} + 1 = \frac{2^p}{\chi}|\psi'_{m-1}| + \frac{1}{\chi Q_{m-2}} + 1\\ &< 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\left(2|\psi'_{m-1}| - \frac{1}{Q_{m-2}}\right)\end{aligned}$$

$$
\begin{aligned}
&= 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\frac{2P_{m-1} + 2\sqrt{D} - 1}{Q_{m-2}} \\
&< 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\frac{2 + Q_{m-2} - 1}{Q_{m-2}} \\
&= 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\left(1 + \frac{1}{Q_{m-2}}\right) < 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\left(1 + \frac{1}{2\sqrt{D}}\right) \\
&< 2^p|\psi'_{m-1}| + 1 - \frac{1}{\chi}\left(1 + \frac{1}{2^{p-1}}\right) = 2^p|\psi'_{m-1}| = 2^p\left|\frac{\zeta'_{m-1}}{\zeta'_m}\right|,
\end{aligned}
$$

using $\sqrt{D} < 2^{p-2}$ for the last inequality. Now from Lemma 9.3 a) and c), we obtain $|\zeta'_{j+2}| = -q_j|\zeta'_{j+1}| + |\zeta'_j|$, hence $\left|\frac{\zeta'_j}{\zeta'_m}\right| = q_j\left|\frac{\zeta'_{j+1}}{\zeta'_m}\right| + \left|\frac{\zeta'_{j+2}}{\zeta'_m}\right|$ $(j \leq m\text{-}2)$, so T_j and $\left|\frac{\zeta'_j}{\zeta'_m}\right|$ satisfy the same recursion. Hence the above inequalities follow since all $q_j \geq 1$ for $j \leq m\text{-}2$. Now, for all $j \leq m$: $\lambda(\mathbb{r}_j, x{+}y) = \left|\frac{\zeta'_m}{\zeta'_j}\right|\lambda(\mathbb{r}_m, x{+}y) = \frac{|\zeta'_m|T_j}{|\zeta'_j|2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_j}\frac{1}{g\rho(x)\rho(y)}$, so as in the previous case:

$$
\lambda(\mathbb{r}_{n-1}, x{+}y) = \frac{|\zeta'_m|T_{n-1}}{|\zeta'_{n-1}|2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_{n-1}}\frac{1}{g\rho(x)\rho(y)} < 1\cdot1\cdot1\cdot\frac{1}{g}g = 1 \text{ and}
$$

$$
\lambda(\mathbb{r}_{n+1}, x{+}y) > \left(1 + \frac{1}{\sqrt{\Delta}}\right)\frac{|\zeta'_m|T_n}{|\zeta'_n|2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_n}\frac{1}{g\rho(x)\rho(y)} > g^7\frac{1}{\chi}\frac{1}{\chi}\cdot1\cdot\frac{1}{g^2} = \frac{g^5}{\chi^2} > g^3 > 1,
$$

hence $\hat{\mathbb{r}}(x + y) \in \{\mathbb{r}_-(x + y), \mathbb{r}_+(x + y)\}$, and again, $\mathrm{M}(x + y) \geq \left\lceil \frac{2^p}{g} \right\rceil = \gamma$. □

Theorem 10.4: If $\hat{\mathbb{r}}(x)$, $\hat{\mathbb{r}}(y) \in \Re$ are such that the bounds of Theorem 9.4 and the conditions of Algorithm 10.1 hold, then Algorithm 4 performs $O(\log D)$ arithmetic operations on inputs requiring $p{+}2\log D{+}O(1)$ bits of storage at worst and $p{+}\frac{1}{2}\log D{+}O(1)$ in almost all cases. In particular, $\mathrm{M}(x{+}y) = O(2^p\sqrt{D})$ in the worst case and $\mathrm{M}(x{+}y) = O(2^p)$ almost always.

Proof: By Theorem 9.4, computing $\mathbb{c}$ takes $O(\log D)$ arithmetic operations on numbers bounded by $O(D)$. By Theorem 9.6, the same is true for the computation of $\mathbb{r}_m$. By Theorem 9.1, in obtaining $\hat{\mathbb{r}}(x{+}y)$ from $\mathbb{r}_m$, all coefficients computed by the neighbouring algorithm are bounded by $O(\sqrt{D})$. So we only need to prove that $\mathbb{r}_n = \hat{\mathbb{r}}(x{+}y)$ can be obtained from $\mathbf{r}_m$ in $O(\log D)$ iterations (i.e. $|n-m| = O(\log D)$) and that the maximum value of T_j is bounded by $O(2^pD^2)$ at worst and $O(2^p\sqrt{D})$ in almost all cases. From the proof of Theorem 9.7, we have $\frac{1}{4\sqrt{D}} < \frac{\theta}{U} < 4(q+1)$ where q is as in Theorem 9.7. By Lemma 10.1 a): $\lambda(x) < q'+1$, $\lambda(y) < q''+1$ for some partial quotients q', q'' generated by the continued fraction algorithm. This, together with condition iii) of Algorithm 10.1 and the inequalities $L \ge g\frac{\theta\gamma^2}{U2^{2p}} > g\frac{1}{4\sqrt{D}}\frac{1}{g^2}$ and $L < g\chi\frac{\theta}{U}\rho(x)\rho(y)\lambda(x)\lambda(y)$ yields the following bounds on L:

$$\frac{1}{4g\sqrt{D}} < L < 4\chi g^2(q+1)(q'+1)(q''+1). \qquad (5)$$

Distinguish between the same three cases as in Algorithm 10.1.

Case 1: $\mathrm{M}(x+y) = \left\lceil \frac{L2^p}{g} \right\rceil \le \lceil 2^p g^2 \rceil$. There is nothing else to prove.

Case 2: From Theorem 9.2 b), (5), and the terminating condition of the *repeat* loop in Step 4 b) of Algorithm 10.1: $4g\sqrt{D} > \frac{1}{L} > \frac{T_{n-1}}{2^p} \ge \left|\frac{\theta'_{n-1}}{\theta'_m}\right| = \prod_{i=m}^{n-2}|\psi'_i| > 2^{\frac{n-m-1}{2}}$ or $2^{n-m} < 32g^2D$. So $n - m = O(\log D)$.

Now $T_{n-1} < \frac{2^p}{L} < 2^{p+2}g\sqrt{D}$ and$T_n < (q_{n-2} + 1)T_{n-1} < (q_{n-2} + 1)g2^{p+2}\sqrt{D}$. From Theorem 9.2, $q_{n-2} < \sqrt{\Delta}$, and q_{n-2} is almost always small by the Gauss-Kuz'min law, so by Theorem 9.2 a) and c) $T_{n-1} = O(2^pD)$ at worst and $T_{n-1} = O(2^p\sqrt{D})$ almost always.

Finally $\mathrm{M}(x+y) \le \left\lceil (q_{n-2}+1)\frac{T_{n-1}L}{g} \right\rceil \le \left\lceil (q_{n-2} + 1)\frac{2^p}{g} \right\rceil$.

Case 3: Here $4\chi g^2(q+1)(q'+1)(q''+1) > L \ge \frac{T_n}{2^p} = \left|\frac{\zeta'_n}{\zeta'_m}\right| = \left|\frac{\theta'_m}{\theta'_n}\right| = \prod_{i=n}^{m-1}|\psi'_i| > 2^{\frac{m-n}{2}}$, so $2^{m-n} < \left(4\chi g^2(q+1)(q'+1)(q''+1)\right)^2 = O(D^4)$ in the worst case and $O(1)$ in almost all cases. Hence $m-n = O(\log D)$ and $m-n = O(1)$ almost always.

Now T_{n-1} is the largest value computed in the recursion. $T_{n-1} < (q_{n+1}+1)T_n \le (q_{n+1}+1)L2^p \le 4\chi g^2(q+1)(q'+1)(q''+1)(q_{n+1}+1)2^p$, so again $T_{n-1} = O(2^pD^2)$ at worst and $T_{n-1} = O(2^p)$ in almost all cases.

Finally $\frac{1}{T_n} < \frac{q_{n+1} + 1}{T_{n-1}} < \frac{q_{n+1} + 1}{L2^p}$, so $\mathrm{M}(x+y) \le \left\lceil (q_{n+1} + 1)\frac{2^p}{g} \right\rceil$. □

```
Algorithm 10.2: Input: $\hat{\mathfrak{r}}(x) \in \mathfrak{R}$ for $x \in \mathbb{R}$, $\mathrm{M}(x)$, $m \in \mathbb{Z}^{>0}$.
Output: $\hat{\mathfrak{r}}(mx)$, $\mathrm{M}(mx)$.
Algorithm: 1. Obtain the binary decomposition $m = \sum_{i=0}^{r} b_i\, 2^{r-i}$ of $m$, $b_i \in \{0,1\}$, $b_0 = 1$.
           2. Set $\hat{\mathfrak{r}}(z_0) = \hat{\mathfrak{r}}(x)$.
           3. For $i = 1$ to $r$ do
              a) Compute $\hat{\mathfrak{r}}(2z_{i-1})$, $\mathrm{M}_2(2z_{i-1})$ using Algorithm 10.1.
                 Set $\hat{\mathfrak{r}}(z_i) = \hat{\mathfrak{r}}(2z_{i-1})$, $\mathrm{M}(z_i) = \mathrm{M}(2z_{i-1})$.
              b) If $b_i = 1$ then compute $\hat{\mathfrak{r}}(z_i + x)$, $\mathrm{M}(z_i + x)$ using Algorithm 10.1.
                 Set $\hat{\mathfrak{r}}(z_i) = \hat{\mathfrak{r}}(z_i + x)$, $\mathrm{M}(z_i) = \mathrm{M}(z_i + x)$.
           4. Set $\hat{\mathfrak{r}}(mx) = \hat{\mathfrak{r}}(z_r)$, $\mathrm{M}(mx) = \mathrm{M}(z_r)$. □
```

This algorithm is the analogue of the exponentiation technique given in Algorithm 2.1. From Algorithm 10.1, it is clear that if $g^{-1} \le \rho(z_{i-1})^2 \le g$ after Step 3 a) and $g^{-1} \le \rho(2z_{i-1})\rho(x) \le g$ after Step 3 b) in each iteration of the algorithm, we have $\mathrm{M}(z_i) \ge \gamma$ and $\hat{\mathfrak{r}}(z_i) \in \{\mathfrak{r}_-(z_i), \mathfrak{r}_+(z_i)\}$ $(1 \le i \le r)$.

Theorem 10.5: Let $m \in \mathbb{Z}^{>0}$, $x \in \mathbb{R}$, and let $\hat{\mathfrak{r}}(x)$ satisfy the bounds of Theorem 9.4. Furthermore, assume that conditions i) - iii) of Algorithm 10.1 are satisfied for each application of Algorithm 10.1 in Step 3 of Algorithm 10.2. If m is polynomially bounded by D, then Algorithm 10.2 performs $O((\log D)^2)$ arithmetic operations on inputs of size $O(\log D)$.

Proof: By Theorem 10.4, since $p = O(\log D)$, all numbers have input size $O(\log D)$. Step 1 of Algorithm 10.2 takes $O(r) = O(\log m)$ operations and this is $O(\log D)$ if m is polynomially bounded by D. Steps 2 and 4 take $O(1)$ operations. For each iteration, Steps 3 a) and 3 b) each perform $O(\log D)$ operations. So the number of operations needed for Step 3 is $O(r \log D) = O((\log D)^2)$. □

Now that we have presented all the required algorithms for our protocol, two more problems remain to be solved:

1. Both communication partners need to start with an initial ideal such that conditions i) - iii) of Algorithm 10.1 are satisfied throughout the protocol, i.e. for each iteration of Algorithm 10.1.
2. Algorithm 10.2 computes one of two possible ideals. The two partners need not obtain the same ideal from Algorithm 10.2 and must be able to agree on a common unique key.

These two problems will be solved in Section 10.3 and Chapter 11, respectively.

10.3 Error Analysis

Lemma 10.6: Let x, y, $\hat{\mathbf{r}}(x)$, $\hat{\mathbf{r}}(y)$, $\mathrm{M}(x)$, $\mathrm{M}(y)$ be as in Algorithm 10.1. Then

$$\rho(x)\rho(y) \le \rho(x+y) < \sqrt{K}\,(x)\rho(y).$$

Proof: Assume $\mathrm{M}(x), \mathrm{M}(y) \ge \gamma$, $\hat{\mathbf{r}}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, $\hat{\mathbf{r}}(y) \in \{\mathbf{r}_-(y), \mathbf{r}_+(y)\}$ and $g^{-1} \le \rho(x)\rho(y) \le g$.

Proof: *Case 1*: $\rho(x+y) = \dfrac{\hat{\lambda}(\mathbf{r}_m, x+y)}{\lambda(\mathbf{r}_m, x+y)} \ge \dfrac{\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y)}{\frac{\theta}{U}\lambda(x)\lambda(y)} = \dfrac{\hat{\theta}}{\theta}\rho(x)\rho(y) \ge \rho(x)\rho(y).$

$$\rho(x+y) < \frac{\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y) + 2^{-p}}{\frac{\theta}{U}\lambda(x)\lambda(y)} < \rho(x)\rho(y)\left(\frac{\hat{\theta}}{\theta} + \frac{1}{2^p\frac{\theta}{U}\hat{\lambda}(x)\hat{\lambda}(y)}\right) = \rho(x)\rho(y)\frac{\hat{\theta}}{\theta}\left(1 + \frac{g}{L2^p}\right)$$

$$\le \rho(x)\rho(y)\frac{\hat{\theta}}{\theta}\left(1 + \frac{g}{2^p}\right) < \rho(x)\rho(y)\chi\left(1 + \frac{g}{2^p}\right) < \rho(x)\rho(y)\chi^2\left(1 + \frac{g}{2^p}\right)$$

$< \sqrt{K}\rho(x)\rho(y)$ by (1) in Theorem 10.3 and Lemma 10.2 d).

$$\textit{Case 2}\text{: } \rho(x+y) = \frac{\hat{\lambda}(\mathbf{r}_n, x+y)}{\lambda(\mathbf{r}_n, x+y)} \ge \frac{\frac{T_n\hat{\theta}}{U2^p}\hat{\lambda}(x)\hat{\lambda}(y)}{\left|\frac{\theta'_n}{\theta'_m}\right|\frac{\theta}{U}\lambda(x)\lambda(y)} = \rho(x)\rho(y)\frac{\hat{\theta}T_n}{\theta\,2^p}\left|\frac{\theta'_n}{\theta'_m}\right| \ge \rho(x)\rho(y).$$

$$\rho(x+y) < \frac{\frac{T_n\hat{\theta}}{U2^p}\hat{\lambda}(x)\hat{\lambda}(y) + 2^{-p}}{\left|\frac{\theta'_n}{\theta'_m}\right|\frac{\theta}{U}\lambda(x)\lambda(y)} = \rho(x)\rho(y)\left(\frac{T_n}{2^p}\left|\frac{\theta'_m}{\theta'_n}\right|\frac{\hat{\theta}}{\theta} + \frac{1}{2^p\left|\frac{\theta'_n}{\theta'_m}\right|\frac{\theta}{U}\hat{\lambda}(x)\hat{\lambda}(y)}\right)$$

$$< \rho(x)\rho(y)\frac{\hat{\theta}}{\theta}\left(\chi + \frac{g}{L2^p}\left|\frac{\theta'_m}{\theta'_n}\right|\right) < \rho(x)\rho(y)\chi\left(\chi + \frac{g}{L2^p}\frac{\chi 2^p}{T_n}\right)$$

$\le \rho(x)\rho(y)\chi^2\left(1 + \frac{g}{2^p}\right) < \sqrt{K}\rho(x)\rho(y)$ by (1) and (3) in Theorem 10.3, the ending condition in the *repeat* loop in Step 4 b) of Algorithm 10.1, and Lemma 10.2 d).

$$\textit{Case 3}\text{: } \rho(x+y) = \frac{\hat{\lambda}(\mathbf{r}_n, x+y)}{\lambda(\mathbf{r}_n, x+y)} \ge \frac{\frac{2^p}{T_n}\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y)}{\left|\frac{\zeta'_m}{\zeta'_n}\right|\frac{\theta}{U}\lambda(x)\lambda(y)} = \frac{\hat{\theta}}{\theta}\frac{2^p}{T_n}\left|\frac{\zeta'_n}{\zeta'_m}\right|\rho(x)\rho(y) > \rho(x)\rho(y).$$

$$\rho(x+y) < \frac{\frac{2^p}{T_n}\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y) + 2^{-p}}{\left|\frac{\zeta'_m}{\zeta'_n}\right|\frac{\theta}{U}\lambda(x)\lambda(y)} < \rho(x)\rho(y)\left(\frac{2^p}{T_n}\left|\frac{\zeta'_n}{\zeta'_m}\right|\frac{\hat{\theta}}{\theta} + \frac{1}{2^p\left|\frac{\zeta'_m}{\zeta'_n}\right|\frac{\theta}{U}\hat{\lambda}(x)\hat{\lambda}(y)}\right)$$

$$< \rho(x)\rho(y)\frac{\hat{\theta}}{\theta}\left(\chi + \frac{g}{L2^p}\left|\frac{\zeta'_n}{\zeta'_m}\right|\right) < \rho(x)\rho(y)\chi\left(\chi + \frac{g\chi T_n}{L2^{2p}}\right) \le \rho(x)\rho(y)\chi^2\left(1 + \frac{g}{2^p}\right)$$

$< \sqrt{K}\rho(x)\rho(y)$ by (1), (4), Step 4 c) of Algorithm 10.1, and Lemma 10.2 d). ❑

Theorem 10.7: Let $m \in \mathbb{Z}$, $1 \le m \le B$, $x \in \mathbb{R}$, $\hat{\mathbf{r}}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, $\mathrm{M}(x) \ge \gamma$ and let $\hat{\mathbf{r}}(x) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$ be such that Q and P satisfy the boundary conditions in Theorem 9.1 a) ii) and iii). If

$$g^{-\frac{1}{2B-1}} \le \rho(x) \le \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}},$$

then $\mathrm{M}(z_i) \geq \gamma$ and $\hat{\mathrm{r}}(z_i) \in \{\mathrm{r}_-(z_i), \mathrm{r}_+(z_i)\}$ throughout Algorithm 10.2 $(1 \leq i \leq r)$. In particular, $\mathrm{M}(mx) \geq \gamma$, $\hat{\mathrm{r}}(mx) \in \{\mathrm{r}_-(mx), \mathrm{r}_+(mx)\}$, and

$$\min\{1, \rho(x)^{2m-1}\} \leq \rho(mx) \leq \max\{1, \rho(x)^{2m-1}\}K^{m-1}.$$

Proof: Let $\mathrm{U}_0 = \max\{1, \rho(x)\} \geq 1$, $\mathrm{L}_0 = \min\{1, \rho(x)\} \leq 1$, and $2^r \leq m < 2^{r+1}$. Define

$$\mathrm{U}_{i+1} = K\mathrm{U}_0\mathrm{U}_i^2, \qquad \mathrm{L}_{i+1} = \mathrm{L}_0\mathrm{L}_i^2 \qquad (0 \leq i \leq r\text{-}1). \qquad (6)$$

Then $\mathrm{U}_{i+1} \geq \mathrm{U}_i \geq 1$ since $K > 1$, and $\mathrm{L}_{i+1} \leq \mathrm{L}_i \leq 1$. We prove by induction on i:

$$\mathrm{U}_i = \mathrm{U}_0^{2^{i+1}-1}K^{2^i-1}, \qquad \mathrm{L}_i = \mathrm{L}_0^{2^{i+1}-1}, \qquad (0 \leq i \leq r). \qquad (7)$$

Proof of (7): The case $i = 0$ is clear. Assume that $i \geq 1$ and show (7) for i+1 using (6) for i.
$\mathrm{U}_{i+1} = K\mathrm{U}_0\mathrm{U}_i^2 = K\mathrm{U}_0(\mathrm{U}_0^{2^{i+1}-1}K^{2^i-1})^2 = \mathrm{U}_0^{1+2(2^{i+1}-1)}K^{1+2(2^i-1)} = \mathrm{U}_0^{2^{i+2}-1}K^{2^{i+1}-1}$.
$\mathrm{L}_{i+1} = \mathrm{L}_0\mathrm{L}_i^2 = \mathrm{L}_0(\mathrm{L}^{2^{i+1}-1})^2 = \mathrm{L}_0^{1+2(2^{i+1}-1)} = \mathrm{L}_0^{2^{i+2}-1}$.

Setting $i = r$, we get $\mathrm{U}_r = \mathrm{U}_0^{2^{r+1}-1}K^{2^r-1} \leq \mathrm{U}_0^{2m-1}K^{m-1} \leq \mathrm{U}_0^{2B-1}K^{B-1}$. If $\mathrm{U}_0 = 1$ then $\mathrm{U}_r \leq K^{B-1} < A^{6(B-1)} = g^{\frac{3(B-1)}{8B^2}} < g$ by Lemma 10.2 d). If $\mathrm{U}_0 = \rho(x)$, then $\mathrm{U}_r \leq \rho(x)^{2B-1}K^{B-1} \leq \frac{g}{K^{B-1}}K^{B-1} = g$. So in either case $\mathrm{U}_i \leq g$ for $0 \leq i \leq r$.

Similarly, $\mathrm{L}_r = \mathrm{L}_0^{2^{r+1}-1} \geq \mathrm{L}_0^{2m-1} \geq \mathrm{L}_0^{2B-1}$. If $\mathrm{L}_0 = 1$, then $\mathrm{L}_r \geq 1 > g^{-1}$, and if $\mathrm{L}_0 = \rho(x)$, then $\mathrm{L}_r \geq \rho(x)^{2B-1} \geq g^{-1}$, so in either case $\mathrm{L}_i \geq g^{-1}$ for $0 \leq i \leq r$. We have

$$L_i \leq \rho(z_i) \leq U_i \qquad (0 \leq i \leq r). \qquad (8)$$

Proof of (8) by induction on i: The case $i = 0$ is $\mathrm{L}_0 \leq \rho\ (x) \leq \mathrm{U}_0$ and $\rho(z_0) = \rho(x)$. Assume now that $i > 0$. From the induction hypothesis for i, (6) and Theorem 10.6, we obtain:
case $b_{i+1} = 0$: $\rho(z_{i+1}) = \rho(2z_i) \geq \rho(z_i)^2 \geq \mathrm{L}_i^2 \geq \mathrm{L}_{i+1}$ since $\mathrm{L}_0 \leq 1$.

$$\rho(z_{i+1}) < \sqrt{K}\,\rho(z_i)^2 \leq \sqrt{K}\,\mathrm{U}_i^2 \leq \mathrm{U}_{i+1} \text{ since } K \geq 1, \mathrm{U}_0 \geq 1.$$

case $b_{i+1} = 1$: $\rho(z_{i+1}) = \rho(2z_i{+}x) \geq \rho(z_i)^2\,\rho(z_0) \geq \mathrm{L}_i^2L_0 = \mathrm{L}_{i+1}$.

$$\rho(z_{i+1}) < \sqrt{K}\,\rho(2z_i)\rho(x) < K\rho(z_i)^2\,\rho(z_0) \leq K\mathrm{U}_i^2\mathrm{U}_0 = \mathrm{U}_{i+1}.$$

From (8), it follows that $\mathrm{L}_0^{2m-1} \leq L_r \leq \rho(z_r) = \rho(mx) \leq \mathrm{U}_r \leq \mathrm{U}_0^{2m-1}K^{m-1}$.

Next we prove that

$$L_i \leq \rho(z_{i-1})^2 \leq U_i \text{ and (if } b_i = 1)L_i \leq \rho(2z_{i-1})\rho(x) \leq U_i \qquad (1 \leq i \leq r). \qquad (9)$$

Then it follows that $g^{-1} \leq \rho(z_{i-1})^2 \leq g$ after Step 3 a) and $g^{-1} \leq \rho(2z_{i-1})\rho(x) \leq g$ after Step 3 b) in each iteration of Algorithm 10.2, hence $\mathrm{M}(z_i) \geq \gamma$ and $\hat{\mathrm{r}}(z_i) \in \{\mathrm{r}_+(z_i), \mathrm{r}_-(z_i)\}$ for $1 \leq i \leq r$, and from the r-th iteration $\hat{\mathrm{r}}(mx) \in \{\mathrm{r}_-(mx), \mathrm{r}_+(mx)\}$ and $\mathrm{M}(mx) \geq \gamma$.

Proof of (9): Again, we use induction on i. For simplicity, we let $\rho(z_{-1}) = 1$.

The case $i = 0$ yields $L_0 \leq 1 \leq U_0$ and $L_0 \leq \rho(x) \leq U_0$. Now assume that our claim holds for $i \geq 1$ and prove it for $i+1$. Then from (6), (8), and Theorem 10.6:

$L_{i+1} = L_i^2 L_0 \leq L_i^2 \leq \rho(z_i)^2 \leq U_i^2 \leq K U_i^2 U_0 = U_{i+1}$, and in the case where $b_{i+1} = 1$:

$L_{i+1} = L_i^2 L_0 \leq \rho(z_i)^2 \rho(x) \leq \rho(2z_i)\rho(x) \leq \sqrt{K}\rho(z_i)^2\rho(x) \leq \sqrt{K} U_i{}^2 U_0 \leq U_{i+1}$. □

Theorem 10.8: Let $a, b \in \mathbb{Z}$, $1 \leq a, b \leq B$, $c \in \mathbb{R}$ and let $\hat{\mathrm{r}}(c) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$ be such that $\hat{\mathrm{r}}(c) \in \{\mathrm{r}_-(c), \mathrm{r}_+(c)\}$, $\mathrm{M}(c) \geq \gamma$, $A^{-1} \leq \rho(c) \leq A$, and Q and P satisfy the boundary conditions in Theorem 9.1 a) ii) and iii). Then $\mathrm{M}(abc) \geq \gamma$, $\hat{\mathrm{r}}(abc) \in \{\mathrm{r}_-(abc), \mathrm{r}_+(abc)\}$, and $g^{-1} \leq \rho(abc) \leq g$. Here $\hat{\mathrm{r}}(abc)$ is obtained by applying Algorithm 10.2 to $\hat{\mathrm{r}}(c)$ and b to compute $\hat{\mathrm{r}}(bc)$, then applying Algorithm 10.2 to $\hat{\mathrm{r}}(bc)$ and a.

Proof: We want to apply Theorem 10.7 first to $\hat{\mathrm{r}}(c)$ and b, then to $\hat{\mathrm{r}}(bc)$ and a. Hence we first need to show that $g^{-\frac{1}{2B-1}} \leq \rho(c) \leq \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}}$. To prove these inequalities, observe that by Lemma 10.2 d): $A^{2B-1}K^{B-1} < A^{2B-1}A^{6(B-1)} = A^{8B-7} = g^{\frac{8B-7}{16B^2}} < g$, so $A^{2B-1} < \frac{g}{K^{B-1}} < g$, hence $g^{-\frac{1}{2B-1}} < A^{-1} \leq \rho(c) \leq A < \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}}$. By Theorem 10.7, we have $\mathrm{M}(bc) \geq \gamma$, $\hat{\mathrm{r}}(bc) \in \{\mathrm{r}_-(bc), \mathrm{r}_+(bc)\}$, and $\min\{1, \rho(c)^{2b-1}\} \leq \rho(bc) \leq \max\{1, \rho(c)^{2b-1}\}K^{b-1}$.We know that the coefficients of $\hat{\mathrm{r}}(bc)$ satisfies the boundary conditions of Theorem 9.1, so only $g^{-\frac{1}{2B-1}} \leq \rho(bc) \leq \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}}$ remains to be proven.

Assume first that $\rho(c) \geq 1$, then from our above result $\rho(bc) \geq 1 > g^{-\frac{1}{2B-1}}$. From Lemma 10.2 d), we get $g = A^{16B^2} > A^{4B^2}K^{2B^2} > A^{(2B-1)^2}K^{2B(B-1)} = (A^{2B-1}K^{B-1})^{2B-1}K^{B-1}$, so

$\rho(bc) \le \rho(c)^{2b-1}K^{b-1} \le \rho(c)^{2B-1}K^{B-1} \le A^{2B-1}K^{B-1} < \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}}$. Now consider the case $\rho(c) < 1$. Then $\rho(bc) \ge \rho(c)^{2b-1} \ge \rho(c)^{2B-1} \ge A^{-(2B-1)} = g^{-\frac{2B-1}{16B^2}} > g^{-\frac{2B-1}{(4B-2)^2}} > g^{-\frac{1}{2B-1}}$ and from $g > A^{12B^2} > K^{2B^2} > K^{2B(B-1)} = K^{(B-1)(2B-1)}K^{B-1}$, we obtain $\rho(bc) \le K^{b-1} \le K^{B-1} < \left(\frac{g}{K^{B-1}}\right)^{\frac{1}{2B-1}}$. If follows once again from Theorem 10.7 that $\mathrm{M}(abc) \ge \gamma$, $\hat{\mathfrak{r}}(abc) \in \{\mathfrak{r}_-(abc), \mathfrak{r}_+(abc)\}$, and $\min\{1, \rho(bc)^{2a-1}\} \le \rho(abc) \le \max\{1, \rho(bc)^{2a-1}\}K^{a-1}$ and .

We finally need to show that $g^{-1} \le \rho(abc) \le g$. $\rho(bc) \ge 1$ implies $\rho(abc) \ge 1 > g^{-1}$, $\rho(abc) \le \rho(bc)^{2a-1}K^{a-1} < \rho(bc)^{2B-1}K^{B-1} \le \frac{g}{K^{B-1}}K^{B-1} = g$. In the case where $\rho(bc) < 1$, it follows that $\rho(abc) \ge \rho(bc)^{2a-1} \ge \rho(bc)^{2B-1} \ge g^{-1}$ and $\rho(abc) \le K^{a-1} \le K^{B-1} \le A^{6(B-1)} = g^{\frac{3(B-1)}{8B^2}} < g$. ❑

Lemma 10.9: Let $\mathfrak{r} = (\mu) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] \in \mathfrak{R}$ where $\mathfrak{r}$ is obtained from $\mathfrak{O}$ by applying Algorithm 9.1 to $\mathfrak{O}$ a few times (at least once). Set $c = \log|\mu|$, $\hat{\mathfrak{r}}(c) = \mathfrak{r}$, $\mathrm{M}(c) = 2^p$. Then $\hat{\mathfrak{r}}(c)$ and $\mathrm{M}(c)$ satisfy the conditions of Theorem 10.8.

Proof: P and Q satisfy the boundary conditions in Theorem 9.1 a) ii) and iii). Since $\lambda(\mathfrak{r}, c) = |\mu|e^{-c} = 1$, we have $\mathfrak{r} \in \{\mathfrak{r}_-(c), \mathfrak{r}_+(c)\}$. Furthermore $\mathrm{M}(c) = 2^p \ge \gamma$, so $\rho(c) = 1$. ❑

Now if both communication partners start on an initial ideal $\hat{\mathfrak{r}}(c)$ generated as described in Lemma 10.9, they can obtain their respective final ideals $\hat{\mathfrak{r}}(abc)$ such that $\mathrm{M}(abc) \ge \gamma$, $g^{-1} \le \rho(abc) \le g$, and the conditions of Algorithms 10.1 and 10.2 as well as those for Theorems 10.4 - 10.8 are satisfied throughout their entire computation. Given the above bounds on the relative error $\rho(abc)$, we show in the next chapter that the parties are able to agree on a common unique key.

11 The Protocol

11.1 Solving the Ambiguity Problem

Before resolving the ambiguity in the ideal computed in the protocol, we need a method for computing from $\hat{\mathfrak{r}}(abc)$ not only its neighbouring ideals as done in Algorithms 9.1 and 9.2, but also their approximate distances.

Algorithm 11.1 (Neighbouring):
Input: $\mathfrak{r}_j \in \mathfrak{R}$, $\mathrm{M}(\mathfrak{r}_j, x)$ $(x \in \mathbb{R}, j \geq 2)$.
Output: $\mathfrak{r}_{j+1}, \mathfrak{r}_{j-1} \in \mathfrak{R}$, $\mathrm{M}(\mathfrak{r}_{j+1}, x)$, $\mathrm{M}(\mathfrak{r}_{j-1}, x)$.
Algorithm: Compute $\mathfrak{r}_{j+1}$ using Algorithm 9.1 and $\mathfrak{r}_{j-1}$ using Algorithm 9.2. Compute rational approximations $\hat{\psi}_j$, $\hat{\phi}_{j-1}$ of $|\psi'_j|$ and $|\phi'_{j-1}|$, respectively, as follows. Define $t = s+p$ where

$$s = \begin{cases} 0 & \text{if } P_{j-1} < d, \\ \lfloor \log_2(2d+1) \rfloor & \text{if } P_{j-1} = d. \end{cases}$$

Set $\hat{\psi}_j = \dfrac{2^p P_j + d^*}{2^p Q_{j-1}}$, $\hat{\phi}_{j-1} = \dfrac{\tilde{d} - 2^t P_{j-1}}{2^t Q_{j-1}}$, where $\tilde{d} = \lceil 2^t \sqrt{D} \rceil$. Compute $\mathrm{M}(\mathfrak{r}_{j+1}, x) = \lceil \hat{\psi}_j \mathrm{M}(\mathfrak{r}_j, x) \rceil$, $\mathrm{M}(\mathfrak{r}_{j-1}, x) = \lceil \hat{\phi}_{j-1} \mathrm{M}(\mathfrak{r}_j, x) \rceil$. ❑

Lemma 11.1: Let $x \in \mathbb{R}$, $\mathfrak{r}_{j-1} \in \mathfrak{R}$, so, $\mathfrak{r}_j, \mathfrak{r}_{j+1} \in \mathfrak{R}$. Then

$$\rho(\mathfrak{r}_j, x) \leq \rho(\mathfrak{r}_{j\pm 1}, x) \leq \frac{1 + 2^{-p}}{1 - \mathrm{M}(\mathfrak{r}_{j\pm 1}, x)^{-1}} \rho(\mathfrak{r}_j, x).$$

Furthermore, if t is as in Algorithm 11.1, then $t \leq p + \log_2 d + 2$.

Proof: $\dfrac{\hat{\psi}_j}{|\psi'_j|} = \dfrac{P_j + 2^{-p} d^*}{P_j + \sqrt{D}} \geq 1$. Similarly $\dfrac{\hat{\phi}_{j-1}}{|\phi'_{j-1}|} = \dfrac{2^{-t}\tilde{d} - P_{j-1}}{\sqrt{D} - P_{j-1}} \geq 1$.

$\dfrac{\hat{\psi}_j}{|\psi'_j|} < \dfrac{P_j + \sqrt{D} + 2^{-p}}{P_j + \sqrt{D}} = 1 + \dfrac{1}{2^p(P_j + \sqrt{D})} < 1 + 2^{-p}$, since $P_j \geq 0$, $\sqrt{D} > 1$.

$\dfrac{\hat{\phi}_{j-1}}{|\phi'_{j-1}|} < \dfrac{2^{-t} + \sqrt{D} - P_{j-1}}{\sqrt{D} - P_{j-1}} = \dfrac{1}{2^t(\sqrt{D} - P_{j-1})} + 1$. If $P_{j-1} < d$, then $\sqrt{D} - P_{j-1} > 1$, so $\dfrac{\hat{\phi}_{j-1}}{|\phi'_{j-1}|}$ $< 1 + 2^{-t} = 1 + 2^{-p}$. If $P_{j-1} = d$, then $2d + 1 \leq 2^s$, so $\sqrt{D} - P_{j-1} = \dfrac{D - P_{j-1}{}^2}{\sqrt{D} + P_{j-1}} = \dfrac{D - d^2}{\sqrt{D} + d} \geq$ $\dfrac{1}{\sqrt{D} + d} > \dfrac{1}{2d + 1} \geq 2^{-s}$, using $\sqrt{D} < d+1$, and $\dfrac{\hat{\phi}_{j-1}}{|\phi'_{j-1}|} \leq 1 + \dfrac{1}{2^{t-s}} = 1 + 2^{-p}$.

$\rho(\mathbb{r}_{j+1}, x) = \dfrac{\hat{\lambda}(\mathbb{r}_{j+1}, x)}{\lambda(\mathbb{r}_{j+1}, x)} \geq \dfrac{\hat{\psi}_j\hat{\lambda}(\mathbb{r}_j, x)}{|\psi'_j|\lambda(\mathbb{r}_j, x)} \geq \rho(\mathbb{r}_j, x)$ and

$\rho(\mathbb{r}_{j+1}, x) < \dfrac{\hat{\psi}_j\hat{\lambda}(\mathbb{r}_j, x) + 2^{-p}}{|\psi'_j|\lambda(\mathbb{r}_j, x)} < (1 + 2^{-p})\rho(\mathbb{r}_j, x) + \dfrac{\rho(\mathbb{r}_{j+1}, x)}{2^p\hat{\lambda}(\mathbb{r}_{j+1}, x)}$, so

$\rho(\mathbb{r}_{j+1}, x)\left(1 - \dfrac{1}{\mathrm{M}(\mathbb{r}_{j+1}, x)}\right) < (1 + 2^{-p})\rho(\mathbb{r}_j, x)$ and

$\rho(\mathbb{r}_{j+1}, x) < \dfrac{1 + 2^{-p}}{1 - \mathrm{M}(\mathbb{r}_{j+1}, x)^{-1}}\rho(\mathbb{r}_j, x)$.

Repacing j+1 by j-1, ψ'_j by ϕ'_{j-1}, and $\hat{\psi}_j$ by $\hat{\phi}_{j-1}$, we can use exactly the same arithmetic to prove $\rho(\mathbb{r}_{j-1}, x) \geq \rho(\mathbb{r}_j, x)$ and $\rho(\mathbb{r}_{j-1}, x) < \dfrac{1 + 2^{-p}}{1 - \mathrm{M}(\mathbb{r}_{j-1}, x)^{-1}}\rho(\mathbb{r}_j, x)$.

If $P_{j-1} < d$, then $s = 0$, so $t = p$. If $P_{j-1} = d$, then $2^{s-1} \leq 2d+1$ implies $2^{s-2} \leq d$, since 2^{s-2} is even for $s \geq 3$ and at most 1 for $s \leq 2$, and $2d+1$ is odd and at least 3. It follows that $2^s \leq 4d$ and $t \leq \log_2(4d) + p$. ❑

Denote by $\mathbb{r}(x)$ the reduced ideal closest to x, i.e. $|\delta(\mathbb{r}(x), x)| < |\delta(\mathbb{r}, x)|$ for any reduced principal ideal $\mathbb{r} \neq \mathbb{r}(x)$. Let $\lambda_0(x) = \lambda(\mathbb{r}(x), x)$; analogously, we define $\mathrm{M}_0(x)$, $\hat{\lambda}_0(x)$, $\rho_0(x)$. Then $\mathbb{r}(x) \in \{\mathbb{r}_-(x), \mathbb{r}_+(x)\}$, so Algorithm 10.1 computes either $\mathbb{r}(x+y)$ or one of its neighbours; similarly, Algorithm 10.2 generates $\mathbb{r}(mx)$ or one of its neighbours.

Our protocol will be such that Alice and Bob are either both able to determine $\mathbb{r}(abc)$ or, if this is impossible, they will both obtain $\mathbb{r}_+(abc)$. The next two lemmas give the details of resolving the ambiguity problem.

Lemma 11.2: Let $x \in \mathbb{R}$ and $g^{-1} \leq \rho(x) \leq g$. If $g^{-1} \leq \hat{\lambda}(x) \leq g$, then $g^{-2} < \lambda_0(x) < g^2$.

Proof: For brevity omit the argument x. If $\delta = \delta(x) = \delta(\hat{\mathbf{r}}(x), x)$, then by definition $|\delta_0| \leq |\delta|$, which gives four cases, depending on the signs of δ_0 and δ:

1) $\lambda \geq \lambda_0 \geq 1$. 2) $\lambda \leq \frac{1}{\lambda_0} \leq 1$. 3) $\lambda \geq \frac{1}{\lambda_0} \geq 1$. 4) $\lambda \leq \lambda_0 \leq 1$.

Suppose first $\lambda_0 \leq g^{-2}$, so $\lambda_0 < 1$.

If $\lambda \leq 1$ then from case 4: $\hat{\lambda} = \rho\lambda \leq \rho\lambda_0 < gg^{-2} = g^{-1}$ which is a contradiction.

If $\lambda \geq 1$ then from case 3: $\hat{\lambda} \geq \rho\frac{1}{\lambda_0} > g^{-1}g^2 = g$ which is again a contradiction.

Suppose now that $\lambda_0 \geq g^2$, so $\lambda_0 > 1$.

If $\lambda \leq 1$ then from case 2: $\hat{\lambda} \leq \rho\frac{1}{\lambda_0} < gg^{-2} = g^{-1}$, again a contradiction.

If $\lambda \geq 1$ then from case 1: $\hat{\lambda} \geq \rho\lambda_0 > g^{-1}g^2 = g$, contradiction. ❑

Lemma 11.3: Let $a, b, c, \hat{\mathbf{r}}(c), \mathrm{M}(c)$ be as in Theorem 10.8.
If $\lambda_0(abc) \geq g^2$ or $\lambda_0(abc) \leq g^{-2}$ then $\hat{\mathbf{r}}(abc) = \mathbf{r}_+(abc)$.
If $g^{-2} < \lambda_0(abc) < g^2$ then $\mathbf{r}(abc)$ can be determined from $\hat{\mathbf{r}}(abc)$.

Proof: Again omit the argument abc for brevity. If $\lambda_0 \geq g^2$ or $\lambda_0 \leq g^{-2}$, then by Lemma 11.2: $\hat{\lambda} > g$ (we cannot have $\hat{\lambda} < g^{-1}$, since $\hat{\lambda} = \mathrm{M}2^{-p} \geq \gamma 2^{-p} \geq g^{-1}$). It follows that $\lambda = \frac{\hat{\lambda}}{\rho} > gg^{-1} = 1$, so $\hat{\mathbf{r}} = \mathbf{r}_+$.

Now let $g^{-2} < \lambda_0 < g^2$. From Theorem 10.8: $g^{-1} \leq \rho \leq g$ and $\hat{\mathbf{r}} \in \{\mathbf{r}_-, \mathbf{r}_+\}$, so $\mathbf{r} = \hat{\mathbf{r}}$ or $\mathbf{r}$ is one of the neighbours of $\hat{\mathbf{r}}$. Therefore by Lemma 11.1: $\rho \leq \rho_0 \leq \frac{1 + 2^{-p}}{1 - \mathrm{M}_0^{-1}}\rho$.

Now $\mathrm{M}_0 = \hat{\lambda}_0 2^p = \rho_0\lambda_0 2^p \geq \rho\lambda_0 2^p > g^{-1}g^{-2}2^p = g^{-3}2^p$, so $1 - \mathrm{M}_0^{-1} > 1 - g^3 2^{-p}$ and $\frac{1 + 2^{-p}}{1 - \mathrm{M}_0^{-1}} < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}$. Hence $g^{-3} < \rho\lambda_0 \leq \rho_0\lambda_0 \leq \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho\lambda_0 < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}} g^3$.

Since $\rho_0\lambda_0 = \hat{\lambda}_0$, one can determine an ideal $\mathbf{a}$ which is either $\hat{\mathbf{r}}$ or a neighbour of $\hat{\mathbf{r}}$ such that $g^{-3} \leq \hat{\lambda}(\mathbf{a}, abc) < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}} g^3$ and $\rho \leq \rho(\mathbf{a}, abc) < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho$. If $\mathbf{r} = \psi\mathbf{a}$ where $\psi \in \mathrm{K}^*$, then $|\psi| = \frac{\lambda_0}{\lambda(\mathbf{a}, abc)} = \frac{\lambda_0\rho(\mathbf{a}, abc)}{\hat{\lambda}(\mathbf{a}, abc)} > \frac{g^{-1}\rho}{\frac{1 + 2^{-p}}{1 - g^3 2^{-p}} g^3} \geq \frac{1 - 2^{-p}g^3}{(1 + 2^{-p})g^6} > \frac{1}{1 + \frac{1}{\sqrt{\Delta}}}$

and $|\psi| < gg^3\frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho \leq \frac{(1 + 2^{-p})g^6}{1 - 2^{-p}g^3} < 1 + \frac{1}{\sqrt{\Delta}}$, using Lemma 10.2 f). From Theorem 9.2 c), it follows that $\mathfrak{a} = \mathfrak{r}$ is the ideal closest to abc. ❑

Now assume that either of the communication partners computes a final ideal $\hat{\mathfrak{r}}(abc)$ with distance $\mathrm{M}(abc)$. He/She then determines the ideal's two neighbours and their respective distances. If among these three ideals there is one, say $\mathfrak{a}$, which satisfies $\frac{2^p}{g^3} < \mathrm{M}(\mathfrak{a}, abc) < \frac{(1 + 2^p)g^3}{1 + 2^{-p}g^3}$, then $\mathfrak{b} = \mathfrak{r}(abc)$ from the proof of the previous lemma. Otherwise, by the same lemma, we must have $\lambda_0(bac) \leq g^{-2}$ or $\lambda_0(bac) \geq g^2$ and hence $\hat{\mathfrak{r}}(abc) = \mathfrak{r}_+(abc)$. With this final observation, we are able to present the entire protocol.

11.2 The Final Protocol

1. Alice and Bob agree on D , an ideal $\mathfrak{r} \in \mathfrak{R}$ (obtained by applying Algorithm 9.1 to the ideal $\mathfrak{O} = \left[1, \frac{\sigma-1+\sqrt{D}}{\sigma}\right]$ one or more times), and a bound $B \in \mathbb{Z}^{\geq 2}$. They compute $p = \lceil \log_2(3072dB^2) \rceil$ and $\mathrm{M}(c) = 2^p$ according to Lemma 10.9 where $c = \log|\mu|$, $\mathfrak{r} = (\mu)$, i.e. $\mathfrak{r} = \hat{\mathfrak{r}}(c)$. D, $\mathfrak{r}$, and B can be made public.
2. Alice secretly chooses $a \in \{1, ..., B\}$ and from $\hat{\mathfrak{r}}(c)$ computes $\hat{\mathfrak{r}}(ac) = \left[\frac{Q_A}{\sigma}, \frac{P_A + \sqrt{D}}{\sigma}\right]$ and $\mathrm{M}(ac)$, using Algorithm 10.2. She sends the triple $(P_A, Q_A, \mathrm{M}(ac))$ to Bob.
3. Bob secretly chooses $b \in \{1, ..., B\}$ and from $\hat{\mathfrak{r}}(c)$ computes $\hat{\mathfrak{r}}(bc) = \left[\frac{Q_B}{\sigma}, \frac{P_B + \sqrt{D}}{\sigma}\right]$ and $\mathrm{M}(bc)$, using Algorithm 10.2. He sends the triple $(P_B, Q_B, \mathrm{M}(bc))$ to Alice.
4. From $\hat{\mathfrak{r}}(ac)$, $\mathrm{M}(ac)$, and b, Bob computes $\hat{\mathfrak{r}}(bac)$ and its two neighbours as well as their approximate distances (i.e. M values) using Algorithms 10.2 and 11.1. If he finds among these an ideal $\mathfrak{a}$ such that $\frac{2^p}{g^3} < \mathrm{M}(\mathfrak{a}, bac) < \frac{(1 + 2^p)g^3}{1 - 2^{-p}g^3}$, then $\mathfrak{a} = \mathfrak{r}(bac)$. In this case he sends '0' back to Alice. If he cannot find such an ideal, then he has computed $\mathfrak{r}_+(bac)$. In this case he sends '1' to Alice.

5. From $\hat{\mathfrak{r}}(bc)$, $M(bc)$, and a, Alice computes $\hat{\mathfrak{r}}(abc)$, $M(abc)$ using Algorithm 10.2. If she received '0' from Bob, then she computes the neighbours of $\hat{\mathfrak{r}}(abc)$ and their approximate distances and attempts to compute $\mathfrak{r}(abc)$. If successful, she sends '0' back to Bob. The common key is then $\mathfrak{r}(abc)$. Otherwise the ideal $\hat{\mathfrak{r}}(abc)$ she computed is $\mathfrak{r}_+(abc)$. In this case she sends '1' to Bob. If Alice received '1' from Bob, then he was unable to determine $\mathfrak{r}(bac)$ in which case the ideal $\hat{\mathfrak{r}}(abc)$ computed by Alice is $\mathfrak{r}_+(abc)$. This is then the key.
6. If Bob sent '1', then the ideal $\hat{\mathfrak{r}}(bac) = \mathfrak{r}_+(bac)$ is the key. If Bob sent and received '0', then the ideal $\mathfrak{a}$ he computed in Step 4 is the key. If Bob sent '0' and received '1', then Alice was unable to determine $\mathfrak{r}(abc)$. The key is then the ideal $\hat{\mathfrak{r}}(bac) = \mathfrak{r}_+(bac)$ initially computed by Bob.

Note that if Bob sends '1', Alice need not reply. Altogether:

Bob	*Alice*	*Key ideal*
sends '1'	no reply necessary	$\mathfrak{r}_+(abc)$
sends '0'	sends '1'	$\mathfrak{r}_+(abc)$
sends '0'	sends '0'	$\mathfrak{r}(abc)$

The actual key is the bit string given by the binary representation of the coefficients of the key ideal (or any substring thereof).

12. Security

12.1 The Discrete Logarithm Problem in $\mathfrak{R}$

A cryptanalyst can break our protocol if he is able to determine $\{\mathfrak{r}_-(abc), \mathfrak{r}_+(abc)\}$, given D, c, and the precision parameter p, but without knowing a or b. This can be achieved by solving the problem of finding for any $x \in \mathbb{R}^{>0}$ one of $\mathfrak{r}_-(x)$, $\mathfrak{r}_+(x)$. If $\mathfrak{r}_-(x) = \mathfrak{r}_j$ $(j \geq 1)$, then $\delta(\mathfrak{r}_-(x), x) = \delta_j - x$, so this problem is equivalent to the problem of finding for any $x \in \mathbb{R}^{>0}$ a positive integer j such that $\delta_j \leq x < \delta_{j+1}$. We can formulate the *discrete logarithm problem* in $\mathfrak{R}$ as follows: for any given $\mathfrak{r}_j \in \mathfrak{R}$, find its distance δ_j.

Any fast algorithm for solving the DLP in $\mathfrak{R}$ gives rise to an efficient algorithm for breaking our scheme. Suppose an eavesdropper can quickly solve any instance of the DLP. To break the protocol, he intercepts $\hat{\mathfrak{r}}(ac) = \mathfrak{r}_j$ and computes its distance δ_j. Then $\frac{\mathrm{M}(ac)}{2^p} = \hat{\lambda}(ac) \approx \lambda(ac) = \exp(\delta_j - ac)\eta^k$ for some $k \in \mathbb{Z}$ (recall that η is the fundamental unit of $\mathbb{K}$). Therefore, a is an integer close to $c^{-1}\left(\delta_j - \log \frac{\mathrm{M}(ac)}{2^p} + k\,R\right)$ where R is the regulator of $\mathbb{K}$. More specifically, $g^{-1} \leq \rho(ac) \leq g$ implies $\left|ac - \delta_j + \log \frac{\mathrm{M}(ac)}{2^p} - k\,R\right| \leq \log g$. Since R is usually larger than ac (see next section), k will tend to be quite small; thus an adversary can retrieve a in a few trials for k values. Similarly, he can find b and hence the key ideal.

Since $\delta_j = \log \mu_j$ for $\mathfrak{r}_j = (\mu_j) \in \mathfrak{R}$, the DLP in $\mathfrak{R}$ is equivalent to the problem of finding, for any reduced principal ideal $\mathfrak{r}_j$, a generator μ_j. Until recently, the fastest known algorithm for solving this problem was due to Buchmann and Williams [BW88b] and has a rigorous complexity bound of $O(\sqrt{R}\Delta^{o(1)})$, which can be as large as $O(\Delta^{\frac{1}{4}+\varepsilon})$ if D is chosen appropriately. An index calculus method is sketched in [BW90b] and partially discussed in more detail in [CDO92]. It is an extension of the DLP algorithm for imaginary quadratic fields. For its discussion, we will use the same notation as in Section 8.2.2 (here, B

denotes the cardinality of the factor base $\mathbb{B} = \{[\mathbf{p}_1], \dots, [\mathbf{p}_B]\}$, not the bound on the exponents in out protocol).

As pointed out before, there are many reduced ideals in the principal class, so we need to associate with each reduced principal ideal a fixed generator in our computation. Instead of using the sublattice $\mathbb{H}$ of $\mathbb{Z}^B$ of all $\underline{\mathbf{x}} = (x_1, \dots, x_B) \in \mathbb{Z}^B$ such that $\prod_{i=1}^{B} \mathbf{p}_i^{x_i} \sim (1)$, we define a new sublattice of $\mathbb{Z}^B \times \mathbb{R}$ as $\mathbb{H}' = \{(x_1, \dots, x_B, \log|\alpha|) \in \mathbb{Z}^B \times \mathbb{R} \mid (x_1, \dots, x_B) \in \mathbb{H}, \alpha \in \mathbb{K}^*, \prod_{i=1}^{B} \mathbf{p}_i^{x_i} = (\alpha)\}$. So we identify with each vector in $\mathbb{H}$ a generator of the corresponding principal ideal. The determinant of $\mathbb{H}'$ is hR, and in [CDO92], it is shown how the computation of a generating system and a basis of $\mathbb{H}'$ can yield the structure of the class group, the class number, and the regulator of $\mathbb{K}$. In fact, the same transformations as in the imaginary case are performed, except that they are now carried out on vectors of B+1 coordinates.

To find a generator of a reduced principal ideal $\mathbf{r}$, find a random vector $(x_1, \dots, x_B, x_{B+1})$ such that $1 \le x_i \le \Delta$ for $i \in \{1, \dots, B+1\}$) and compute a reduced ideal $\mathbf{r}'$ in the class of $\mathbf{a} = \mathbf{r}\prod_{i=1}^{B} \mathbf{p}_i^{x_i}$; say $(\alpha)\mathbf{r}' = \mathbf{a}$, $\alpha \in \mathbb{K}^*$. Here, the last coordinate x_{B+1} randomizes the choice of $\mathbf{r}'$. As before, we attempt to factor $\mathbf{r}'$ over $\mathbb{B}$, obtaining in case of success a representation $\mathbf{r}' = \prod_{i=1}^{B} \mathbf{p}_i^{y_i} \mathbf{p}_i'^{z_i} = \prod_{i=1}^{B} p_i^{z_i} \mathbf{p}_i^{y_i - z_i}$, where $y_i z_i = 0$ for $1 \le i \le B$. Then $\mathbf{r}' = \alpha^{-1}\mathbf{a} = \alpha^{-1}\mathbf{r}\prod_{i=1}^{B} \mathbf{p}_i^{x_i}$, so

$$\mathbf{r} = \alpha\mathbf{r}'\prod_{i=1}^{B} \mathbf{p}_i^{-x_i} = \alpha\prod_{i=1}^{B} p_i^{z_i} \mathbf{p}_i^{z_i - y_i - x_i},$$

hence if we set $v_i = z_i - y_i - x_i$ for $1 \le i \le B$, then $\prod_{i=1}^{B} \mathbf{p}_i^{v_i} \sim (1)$, so $\underline{\mathbf{v}} = (v_1, \dots, v_B) \in \mathbb{H}$. Let $\underline{\mathbf{e}}'_1, \dots, \underline{\mathbf{e}}'_{B+1}$ be a basis of $\mathbb{H}'$ and define $\underline{\mathbf{e}}_i \in \mathbb{H}$ to be the vector consisting of the first B coordinates of $\underline{\mathbf{e}}'_i$ $(1 \le i \le B)$. Then $\underline{\mathbf{e}}_1, \dots, \underline{\mathbf{e}}_B$ is a basis of $\mathbb{H}$, and we can express $\underline{\mathbf{v}}$ in terms of this basis; write $\underline{v} = \sum_{i=1}^{B} k_i \underline{e}_i$, $k_1, \dots , k_B \in \mathbb{Z}$. If $v_{B+1} = \sum_{i=1}^{B} k_i e_{i,B+1}$, then $\underline{v} = (v_1, \dots , v_B, v_{B+1}) \in \mathbf{H}'$. Now $e_{i,B+1}$ is the logarithm of a generator of the ideal $\prod_{i=1}^{B} \mathbf{p}_j^{e_{ij}}$, say $e_{i,B+1} = \log |\alpha_i|$ where $\prod_{j=1}^{B} \mathbf{p}_j^{e_{ij}} = (\alpha_i)$ $(1 \le i \le B)$. Thus

$$\exp(v_{B+1}) = \exp\left(\sum_{i=1}^{B} k_i e_{i,B+1}\right) = \prod_{i=1}^{B} \exp(k_i \log |\alpha_i|) = \prod_{i=1}^{B} |\alpha_i|^{k_i},$$

yielding the following identity of ideals:

$$(\exp(v_{B+1})) = \prod_{i=1}^{B} (\alpha_i)^{k_i} = \prod_{i=1}^{B} \prod_{j=1}^{B} \mathbf{p}_j^{e_{ij}k_i} = \prod_{j=1}^{B} \mathbf{p}_j^{v_j},$$

hence $\mathbf{r} = \alpha \prod_{i=1}^{B} p_i^{z_i} \mathbf{p}_i^{v_i} = \left(\alpha \prod_{i=1}^{B} p_i^{z_i} \exp(v_{B+1})\right)$ and we have found a generator of $\mathbf{r}$. The complexity of this method appears to be the same as that of the imaginary case, assuming ERH as usual.

The details for generating relations over the factor base and determining the structure of Cl(K) as well as an implementation of the first stage of the index calculus method are given in [CDO92]. As a numerical example, we mention that the computation for $\Delta = 10^{40}+1$ (a discriminant far too small to guarantee security in our scheme) took 8.3 hours on a SparcStation 2. For large values of D, this method is totally impractical.

It should be pointed out that a fast algorithm for solving the DLP can be used to find the regulator R of $\mathbb{K}$ quickly, see [BW90a] and [BW90b]. By a result of Schoof [Sc83a], we know that if it is possible to find R quickly, then D can be factored quickly. Thus the DLP in $\mathbb{K} = \mathbb{Q}(\sqrt{D})$ is at least as difficult as factoring D. Finally, no method for breaking our scheme is known other than solving the DLP and exhaustive search.

12.2 Safe Choice of D

To prevent an exhaustive key search attack, we need to ensure that the number l of reduced principal ideals in $\mathbb{K}$ is sufficiently large. Since $\eta = |\theta'_{l+1}| = \prod_{j=1}^{l} |\psi_i'| < \left(\sqrt{\Delta}\right)^l$ by Theorem 9.2 c), we have $R = \log \eta < \frac{l}{2} \log \Delta$, and therefore $l > \frac{2R}{\log \Delta}$. Hence we require a lower bound on R.

Consider the analytic class number formula for real quadratic fields $2hR = L(1, \chi)\sqrt{\Delta}$, where $L(1, \chi)$ is defined as follows. The *Kronecker symbol* χ of $\mathbb{K}$ is an extension of the Jacobi symbol $\left(\frac{\Delta}{\cdot}\right)$. For any odd prime p, set $\chi\,(p) = \left(\frac{\Delta}{p}\right)$. Furthermore, set

$$\chi(2) = \begin{cases} 0 \text{ if } \Delta \text{ is even,} \\ 1 \text{ if } \Delta \equiv 1 \pmod 8, \\ -1 \text{ if } \Delta \equiv 5 \pmod 8. \end{cases}$$

Recall for this definition that $\Delta \equiv 0, 1 \pmod 4$. For any $n \in \mathbb{Z}^{>0}$ with unique prime factorization $n = p_1{}^{e_1} \cdots p_r{}^{e_r}$, set $\chi(n) = \chi(p_1)^{e_1} \cdots \chi(p_r)^{e_r}$. Then for any $s \in \mathbb{C}$, $L(s, \chi) = \sum_{n \geq 1} \frac{\chi(n)}{n^s}$ is the Dirichlet L-series corresponding to the Kronecker symbol χ. By a result of Littlewood [Li82], we have $L(1, \chi) > \frac{C}{\log \log \Delta}$ (assuming ERH), where $C = \frac{\pi^2}{12e^{\gamma}(1+o(1))}$ and $\gamma = \lim_{n\to\infty} \left(\sum_{k=1}^{n} \frac{1}{k} - \log n \right) = 0.577...$ is Euler's constant. Hence, we expect that $l > \frac{C}{h} \frac{\sqrt{\Delta}}{\log \log \Delta}$, and we need to bound h from above.

No rigorous lower bound for h is known. Gauss conjectured that $h = 1$ for infinitely many real quadratic fields, and his conjecture remains open. However, the heuristics of Cohen and Lenstra [CL84a], [CL84b] suggest that the odd part of h is small with high probability. While a rigorous proof of this result unfortunately remains unknown, there is strong numerical evidence in support of the Cohen-Lenstra heuristics.

Let h^* be the odd part of the class number, i.e. $h = 2^m h^*$ where $m \geq 0$ and h^* is odd. We will attempt to bound h^*. Let $\mathrm{Cl}_0(\mathrm{K})$ be the subgroup of $\mathrm{Cl}(\mathrm{K})$ of order h^*. As in [CL84a, p. 29], we define

$$w(n) = \sum_{\substack{\mathrm{G} \text{ up to isomorphism} \\ |\mathrm{G}| = n}} \frac{1}{|\mathrm{Aut}(\mathrm{G})|},$$

where G is any finite group and Aut(G) denotes the group of automorphisms of G. The heuristics in [CL84b] imply that for any group G, the probability that $\mathrm{Cl}_0(\mathrm{K})$ is isomorphic to G is $\frac{1}{2c_1|\mathrm{G}||\mathrm{Aut}(\mathrm{G})|}$ for some constant $c_1 > 0$ (the explicit value of c_1 is given in [CL84b]). Then the probability that $h^* = n$ for some odd $n \in \mathbb{Z}^{>0}$ is

$$\Pr(h^*{=}n) = \sum_{\substack{\mathrm{G} \text{ up to isomorphism} \\ |\mathrm{G}| = n}} \frac{1}{2c_1 n|\mathrm{Aut}(\mathrm{G})|} = \frac{1}{2c_1}\frac{w(n)}{n}.$$

Using the identities of Theorem 3.2 in [CL84a, pp. 29f.] and a technique analogous to the one employed in Landau [La36], we can prove that $\sum_{\substack{n>x \\ n \text{ odd}}} w(n) = c_1 \log x + c_2 + O\left(\frac{\log x}{x}\right)$.

where $c_2 > 0$ is a constant that can be explicitly evaluated. Finally, partial summation yields

$$\sum_{\substack{n>x \\ n \text{ odd}}} \frac{w(n)}{n} = \frac{c_1}{x} + O\left(\frac{\log x}{x^2}\right);$$

hence, $\Pr(h^*{>}x) = \frac{1}{2x} + O\left(\frac{\log x}{x^2}\right)$, and it follows that h^* is small with high probability.

Now it is known (see for example [Co62]) that h is odd if $D = p$, $D = 2p$, or $D = p_1 p_2$ where p is any odd prime and p_1, p_2 are primes congruent to 3 (mod 4). More cases of values of D for which the even part of the class number can be bounded are given in Kaplan [Ka76]. Thus by selecting such a D value which is large, we expect that it would be most unlikely that $l < \frac{\sqrt{D}}{10^{20} \log D \log \log D}$, say. This renders the likelihood of success of a search technique to be very slight indeed.

13. Implementation

13.1 Optimization

Consider the bounds on the value of L in each of the three cases of Algorithm 10.1. From (5) in the proof of Theorem 10.4, we obtain $1 \leq L \leq g^3$ in case 1, $\frac{1}{4g\sqrt{D}} < L < 1$ in case 2, and $1 < L < 4\chi g^2(q+1)(q'+1)(q''+1)$ in case 3. Hence we expect case 1 to occur very rarely, since it not only leaves an extremely narrow range for possible values of L, but also corresponds to a very unlikely event, namely having found $\hat{P}(x+y)$ immediately after the reduction step. Case 3 is expected to occur slightly more often, and case 2 should occur almost always, since it permits a very large range of L values. Our computations verify this observation. In all our examples, case 1 was never encountered, case 3 occurred very rarely (at most once per application of Algorithm 10.2), and case 2 occurred almost all the time.

In addition, we expect the number of iterations of Algorithm 9.2 in case 3 to be small, since it was proven to be $O(1)$ almost always. Again, our examples confirm this; in fact, Algorithm 9.2 was never called more than twice, even for our largest discriminants which were approximately 200 digits.

Considering the fact that in addition to the $O(\log D)$ calls in Algorithm 10.1, Algorithm 9.1 is also applied $O(\log D)$ times for each ideal reduction, we see from the above remarks that Algorithm 9.1 is used very frequently throughout our protocol. Hence, we focused out optimization efforts on this part of our computation and used the following modified version of the continued fraction algorithm which is due to Tenner (see [WW87]) in our implementation. Let $\mathfrak{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$ be a primitive principal ideal. Set

$$P_0 = P, \qquad Q_0 = Q, \qquad Q_{-1} = \frac{D - P_0^2}{Q_0}, \qquad t_0 = \begin{cases} 0 \text{ if } Q > 0, \\ 1 \text{ if } Q < 0, \end{cases}$$

$$P_0 + d + t_0 = q_0 Q_0 + r_0, \text{ (i.e. } q_0 = \left\lfloor \frac{P_0 + d + t_0}{Q_0} \right\rfloor \text{ and } r_0 = P_0 + d + t_0 - q_0 Q_0\text{)},$$

and for $j \geq 0$:

$$P_{j+1} = d + t_j - r_j, \qquad Q_{j+1} = Q_{j-1} - q_j(P_{j+1} - P_j),$$

$$t_{j+1} = \begin{cases} 0 & \text{if } Q_{j+1} > 0, \\ 1 & \text{if } Q_{j+1} < 0, \end{cases} \qquad P_{j+1} + d + t_{j+1} = q_{j+1}Q_{j+1} + r_{j+1},$$

$$\text{(i.e. } q_{j+1} = \left\lfloor \frac{P_{j+1} + d + t_{j+1}}{Q_{j+1}} \right\rfloor \text{ and } r_{j+1} = P_{j+1} + d + t_{j+1} - q_{j+1}Q_{j+1}\text{).}$$

This modification, though not as intuitive as the familiar version of the continued fraction algorithm, represents a significant speed-up and can be used for both Algorithms 9.4 and 10.1. It is particularly useful if division with remainder is a single operation as was the case in our multiprecise arithmetic package, since q_{j+1} and r_{j+1} are computed in one step. It cuts down the number of divisions and multiplications by half (i.e. from 2 to 1 per step) and merely introduces one extra addition if the ideals are reduced.

13.2 Computational Results

We implemented our protocol in C language, using a multiprecise integer arithmetic package written by Stephens [St89]. At the time, the only machine available to us was a DEC MicroVAX. Tests show that a more modern environment (such as the hardware and software used for the implementation of our cryptosystem presented in Part I) yields computation times which are approximately 100 times faster than those achieved by the MicroVAX. This estimate was obtained by extensively running Tenner's continued fraction algorithm on both the MicroVAX and the DECStation 5000.

In all our examples, we encountered the simple case of the protocol where Bob and Alice both compute $\mathrm{r}_+(abc)$ and only Bob needs to send his bit 1. Again, we expect this, since the bounds given in Step 4 of the protocol leave an extremely narrow range for M(a, *abc*) and force M(a, *abc*) to be very close to 2^p.

We will give three numerical examples. The computations were done in two fields whose discriminants are Mersenne primes (i.e. $D = \Delta = 2^p - 1$ for some prime p). From Section 12.2, it follows that both fields have odd class number. The first example used a discriminant of 33 digits (by no means secure enough for practical purposes) and was run with an exponent of order $\sqrt{D}$; the other two computations were done in a field with a 183 digit discriminant and were performed on two exponents of order $\sqrt[4]{D}$ (a size which we consider sufficiently secure for such a large discriminant) and $\sqrt{D}$, respectively. Our exponents were simply random integers. The computation times are given in the following table.

Discriminant $\Delta = D$	2^{107}-1	2^{607}-1	2^{607}-1
# digits in D	33	183	183
Size of exponent	$\sqrt{D}$	$\sqrt[4]{D}$	$\sqrt{D}$
# digits in exponent	16	45	91
order of precision	$D^{3/2}$	D	$D^{3/2}$
approx. # digits of precision	50	183	275
CPU time, MicroVAX	3.3 minutes	41 minutes	97 minutes
CPU time, DECStation 5000 (estimated)	2 seconds	25 seconds	58 seconds

14 Conclusion and Open Problems

In Part I of this dissertation, we presented a public-key cryptosystem using arithmetic in cyclotomic fields of degree λ-1, where λ is a prime. The scheme's security is equivalent to the difficulty of factoring the modulus. The scheme employs a Euclidean division algorithm, which in turn is used for computing integer prime divisors of rational primes and computing residue symbols in the field. We gave details for the cases $\lambda = 2, 3$, and 5.

A number of problems remain unsolved. While the Euclidean division algorithm as well as the gcd method for finding prime divisors of rational primes can be used for the cases $\lambda = 7$ and $\lambda = 11$, the residue symbol computation becomes more complicated, due to the fact that there is more than one independent fundamental unit (two such units for $\lambda = 7$ and four for $\lambda = 11$). The complexity of the other prime divisor method (Algorithm 6.1) depends on the number l of reduced (principal) ideals in the field. If l is bounded, then the algorithm is linear in log p, where p is the rational prime whose divisor we wish to compute. As pointed out in Section 6.1, computations show that $l = 1$ if $\lambda \leq 7$. These computations have not been carried for the cases $11 \leq \lambda \leq 19$. We conjecture that in these cases, l is small; possibly $l = 1$ as well.

With respect to residue symbols, it is unknown how to compute $\left[\frac{\alpha}{\beta}\right]$ for $\alpha, \beta \in \mathcal{O}-\{0\}$ efficiently without making use of Euclidean division; nor do we know how to evaluate $\left[\frac{\alpha}{\mathfrak{a}}\right]$ for $\alpha \in \mathcal{O}-\{0\}$ and a non-zero integral ideal $\mathfrak{a}$ in $\mathcal{O}$ efficiently. In fact, it appears that the only known method is factoring the denominator and computing the residue symbol for each individual prime factor. The Euclidean division approach will fail for $\lambda \geq 23$; the question of whether or not the cases $\lambda = 13$, 17, and 19 yield rings of integers which are Euclidean remains open.

Finally, it may be possible to extend the approach by Williams [Wi80] and Loxton et al [LKBS92] to fields with $\lambda \geq 5$.

Part II of the thesis presented a key exchange protocol based on the infrastructure of a real quadratic field. The scheme is the first and so far the only version of a Diffie-Hellman protocol which does not require a group structure. It remains to be seen whether there are other sets which are suitable for Diffie-Hellman key exchange, but which are not groups.

As with all the previous Diffie-Hellman protocols, the only known way of breaking our scheme (other than exhaustive key search, which appears to be infeasible if our parameters are chosen with care) is to solve the discrete logarithm in the underlying structure. The DLP in the set $\mathfrak{R}$ of reduced principal ideals in a real quadratic field $\mathbb{K}$ is essentially the problem of finding for each ideal $\mathfrak{a} \in \mathfrak{R}$ a generator. It appears that an index calculus approach similar to the one used for determining the structure of the class group and the regulator of $\mathbb{K}$ can be used for solving the DLP. As usual, this method is subexponential in the size of the discriminant of $\mathbb{K}$. The DLP can be shown to be at least as difficult as the problem of factoring the discriminant of $\mathbb{K}$. Since the question of whether the DLP is equivalent in difficulty to breaking the scheme remains unanswered, it is unknown whether the security of the protocol is equivalent to the problem of factoring the field discriminant.

References

[Ad79] L. M. Adleman, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography", *Proc. 20th Annual IEEE Symposium on Foundations of Computer Science*, 1979, p. 55-60.

[AD93] L. M. Adleman and J. Demarrais, "A subexponential algorithm for discrete logarithms over all finite fields", to appear in *Math. Comp.*

[Ba90] E. Bach, "Explicit bounds for primality testing and related problems", *Math. Comp.* v. 55, no. 191, July 1990, pp. 355-380.

[BBl79] B. Blakley and G. R. Blakley, "Security of number-theoretic public-key cryptosystems against random attack", I, *Cryptologia*, vol. 2, no. 4, October 1978, pp. 305-321; II, *Cryptologia*, vol. 3, no. 1, January 1979, pp. 29-42; III, *Cryptologia*, vol. 3, no. 2, April 1979, pp. 105-118.

[BBo79] G. R. Blakley and I. Borosh, "Rivest-Shamir-Adleman public-key cryptosystems do not always conceal messages, *Comps. & Maths. with Appl.* 5, 1979, pp. 169-178.

[BD91] J. A. Buchmann and S. Düllmann, "On the computation of discrete logarithms in class groups", *Advances in Cryptology – CRYPTO '90 Proceedings*, Springer-Verlag, Berlin 1991, pp. 134-139.

[BDW90] J. A. Buchmann, S. Düllmann, and H. C. Williams, "On the complexity of a new key exchange system", *Advances in Cryptology – EUROCRYPT '89 Proceedings*, Springer-Verlag, Berlin 1990, pp. 597-616.

[Be28] W. E. H. Berwick, "The arithmetic of quadratic number fields", *The Mathematical Gazette*, vol. XIV, no. 192, January 1928, pp. 1-11.

[Be70] E. R. Berlekamp, "Factoring polynomials over finite fields", *Math. Comp.*, vol. 24, no. 111, July 1970, pp. 713-735.

[Be82] S. Berkovitz, "Factoring via superencryption", *Cryptologia*, vol. 6, no. 3, July 1982, pp. 229-237.

[BFMV84] I. F. Blake, R. Fuji-Hara, R. C. Mullin, and S. A. Vanstone, "Computing logarithms in fields of characteristic two", *SIAM J. Algebraic Discrete Methods* 5, 1984, pp. 276-285.

[BLP93] J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance, "Factoring integers with the number field sieve", *The Development of the Number Field Sieve*, Springer-Verlag, to appear.

[BM92] E. F. Brickell and K. S. McCurley, "An interactive identification scheme based on discrete logarithms and factoring", *J. Cryptology* 5, 1992, pp. 29-39.

[Br88] G. Brassard, *Modern Cryptology*, Springer-Verlag, New York 1988.

[Bu87a] J. A. Buchmann, "On the computation of units and class numbers by a generalization of Lagrange's algorithm", *J. Number Theory*, vol. 26, no. 1, May 1987, pp. 8-30.

[Bu87b] J. A. Buchmann, "On the period length of the generalized Lagrange algorithm", *J. Number Theory*, vol. 26, no. 1, May 1987, pp. 31-37.

[BW87a] J. A. Buchmann and H. C. Williams, "On principal ideal testing in totally complex quartic fields and the determination of certain cyclotomic constants", *Math. Comp.* v. 48, no. 177, January 1987, pp. 55-66.

[BW87b] J. A. Buchmann and H. C. Williams, "On principal ideal testing in algebraic number fields, *J. Symb. Comp.* 4, 1987, pp. 11-19.

[BW88a] J. A. Buchmann and H. C. Williams, "A key-exchange system based on imaginary quadratic fields", *J. Cryptology*, vol. 1, no, 2, 1988, pp. 107-118.

[BW88b] J. A. Buchmann and H. C. Williams, "On the infrastructure of the principal ideal class of an algebraic number field of unit rank one", *Math. Comp.*, vol. 50, no. 182, April 1988, pp. 569-579.

[BW90a] J. A. Buchmann and H. C. Williams, "A key-exchange system based on real quadratic fields", *Advances in Cryptology – CRYPTO '89 Proceedings*, Springer-Verlag, Berlin, 1990, pp. 335-343.

[BW90b] J. A. Buchmann and H. C. Williams, "Quadratic Fields and Cryptography", *Number Theory and Cryptography*, Cambridge University Press, Cambridge (Mass.), 1990, pp. 9-25.

[CD91] C. C. Chuang and J. C. Dunham, "Matrix extensions of the RSA algorithm", *Advances in Cryptology – Crypto '90 Proceedings*, Springer-Verlag, Heidelberg (Germany), 1991, pp. 140-155.

[CDO92] H. Cohen, F. Diaz y Diaz, and M. Olivier, "Calculs des nombres de classes et de régulateurs de corps quadratiques en temps sous-exponentiel", to appear in *Séminaire de Théorie des Nombres de Paris*, 1992.

[CL84a] H. Cohen and H. W. Lenstra, Jr., Heuristics on class groups, *Number Theory* (Nordwijkerhout, 1983), Lecture Notes in Mathematics, vol. 1052, Springer-Verlag, New York, 1984, pp. 26-36.

[CL84b] H. Cohen and H. W. Lenstra, Jr., Heuristics on class groups of number fields, *Number Theory* (Nordwijkerhout, 1983), Lecture Notes in Mathematics, vol. 1068, Springer-Verlag, New York, 1984, pp. 33-62.

[Co62] H. Cohn, *Advanced Number Theory*, Dover Publications, New York, 1962.

[Co84] D. Coppersmith, "Fast evaluation of discrete logarithms in fields of characteristic two" *IEEE Trans. Inf. Theory*, vol. IT-30, 1984, pp. 587-594.

[COS86] D. Coppersmith, A. M. Odlyzko, R. Schroeppel, "Discrete logarithms in GF(*p*)", *Algorithmica* 1, 1986, pp. 1-15.

[DDDHL83] R. A. DeMillo, G. I. Davida, D. P. Dobkin, M. A. Harrison, and R. J. Lipton (eds.), *Applied Cryptology, Cryptographic Protocols, and Computer Security Models*, American Mathematical Society, Providence, Rhode Island, 1983.

[De83] D. E. R. Denning, *Cryptography and Data Security*, Addison Wesley, Reading (Mass.), 1983.

[Di66] Dickson, *History of the Theory of Numbers*, vol. 1, Chelsea, New York 1966.

[DH76] W. Diffie and M. Hellman, "New Directions in Cryptography", *IEEE Trans. Inf. Theory*, vol. IT-22, 6, November 1976, pp. 644-654.

[Ec83] A. Ecker, "Finite semigroups and the RSA cryptosystem", *Cryptography Proceedings, Burg Feuerstein 1982*, Springer-Verlag, Heidelberg (Germany) , 1983, pp. 353-370.

[Ei44a] G. Eisenstein, "Beweis des Reciprocitätssatzes für die cubischen Reste in der Theorie der aus dritten Wurzeln der Einheit zusammengesetzten compexen Zahlen", *J. Reine Angew. Math.*, vol. 27, no. IV, 1844, pp. 289-310.

[Ei44b] G. Eisenstein, "Nachtrag zum cubischen Reciprocitätssatzes für die aus dritten Wurzeln der Einheit zusammengesetzten compexen Zahlen. Criterien des cubischen Characters der Zahl 3 und ihrer Theiler.", *J: Reine Angew. Math.*, vol. 28, no. I, 1844, pp. 28-35.

[El85a] T. ElGamal, "A subexponential-time algorithm for computing discrete logarithms over GF(p^2)", *IEEE Trans. Inf. Theory*, vol. IT-31, 1985, pp. 473-481.

[El85b] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Inf. Theory*, vol. IT-31, 1985, pp. 469-472.

[Go85] J. A. Gordon, "Strong primes are easy to find", *Advances in Cryptology – Proceedings of EUROCRYPT 84*, Springer-Verlag, Heidelberg (Germany), 1985, pp. 216-223.

[Go93a] D. M. Gordon, "Discrete logarithms in GF(p) using the number field sieve", *preprint*.

[Go93b] D. M. Gordon, "Discrete logarithms in GF(p^n) using the number field sieve", *preprint*.

[Gr91] T. Granlund, *GNU MP – The GNU multiple precision Arithmetic Library, Edition 1.2*, Free Software Foundation, Inc., Cambridge (Mass.), 1991.

[GS88] J. Grollman and A. L. Sellman, "Complexity measures for public-key cryptosystems", *SIAM J. Comp.*, vol. 17, no. 2, April 1988, pp. 309-335.

[Ha86] J. Hasted, "On using RSA with low exponent in a public key network", *Advances in Cryptology – CRYPTO '85*, Springer-Verlag, Berlin, 1986, pp. 403-408.

[He78] T. Herlestam, "Critical remarks on some public-key cryptosystems", *BIT*, vol. 18, no. 4, 1978, pp. 493-496.

[HK89] L. Harn and T. Kiesler, "Improved Rabin's scheme with high efficiency", *Electronic Letters*, vol. 25, no. 11, May 1989, pp. 726-728.

[HM89] J. L. Hafner and K. S. McCurley, "A rigorous subexponential algorithm for computation of class groups", *J. Amer. Math. Soc.*, vol. 2, no. 4, October 1989, pp. 837-849.

[HR83] M. E. Hellman and J. M. Reyneri, "Fast computation of discrete logarithms in GF(*q*)", *Advances in Cryptology – Proceedings of CRYPTO '82*, Plenum Press, New York, 1983, pp. 3-13.

[Hu82] L. K. Hua, *Introduction to Number Theory*, Springer-Verlag, New York, 1982.

[Ja46] C. G. J. Jacobi, "Über die Kreistheilung und ihre Anwendung auf die Zahlentheorie", *J. Reine Angew. Math.* 30, 1846, pp. 166-182.

[Ja88] P. Jamnig, "Securing the RSA cryptosystem against cycling attacks", *Cryptologia*, vol. 12, no. 3, July 1988, pp. 159-164.

[Ka67] D. Kahn, *The Codebreakers: The Story of Secret Writing*, Macmillan Publishing Co., New York 1967.

[Ka76] P. Kaplan, "Sur le 2-groupe des classes d'idéaux des corps quadratiques", *J. Reine Angew. Math.* 283/284, 1976, pp. 313-363.

[Kh64] A. Y. Khinchin, *Continued Fractions*, The University of Chicago Press, Chicago, 1964.

[Kn73] D. E. Knuth, *The Art of Computer Programming*, vol. 3: *Sorting and Searching*, Addison-Wesley, Reading (Mass.), 1973.

[Kn81] D. E. Knuth, *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms*, Addison-Wesley, Reading (Mass.), 1981.

[Ko87a] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, New York, 1987.

[Ko87b] N. Koblitz, "Elliptic curve cryptosystems", *Math. Comp.* 48, 1987, pp. 203-209.

[Ko88] N. Koblitz, "Hyperelliptic cryptosystems", *J. Cryptology*, vol. 1, no. 3, 1988, pp. 139-150.

[Ko90] N. Koblitz, "A family of Jacobians suitable for discrete log cryptosystems", *Advances in Cryptology – CRYPTO '88 Proceedings*, Springer-Verlag, Berlin 1990, pp. 94-97.

[Kr22] M. Kraitchik, *Théorie des Nombres*, vol. 1, Gauthier-Villars, Paris 1922.

[Kr24] M. Kraitchik, *Recherches Sur la Théorie des Nombres*, Gauthier-Villars, Paris 1924.

[KR82] D. Kravitz and I. Reed, "Extension of RSA cryptostructure: a Galois approach", *Electronic Letters* 18, 1982.

[Ku75] E. E. Kummer, *Collected Papers*, v. 1, Springer, Berlin, 1975.

[La36] E. Landau, "On a Titchmarsh-Estermann sum", *J. Lond. Math. Soc.* 11, 1936, pp. 242-245.

[La69] E. Landau, *Vorlesungen über Zahlentheorie*, Chelsea, New York 1969.

[Le07] D. N. Lehmer, "A theorem in the theory of numbers", *Bull. Amer. Math. Soc.* 14, 1907, pp. 501-502.

[Le69] D. H. Lehmer", Computer technology applied to the theory of numbers", *Studies in Number Theory*, vol. 6, Math. Assoc. of America, 1969, pp. 117-151.

[Le75] H. W. Lenstra, Jr., "Euclid's algorithm in cyclotomic fields", *J. Lond. Math. Soc.* (2), 10, 1975, pp. 457-465.

[Le79] H. W. Lenstra, Jr., "Euclidean number fields I", *Math. Intelligencer* 2, 1979/80, pp. 6-15.

[Le87] H. W. Lenstra, Jr., "Factoring integers with elliptic curves", *Annals of Math.*, vol. 126, no. 3, November 1987, pp. 649-673.

[Le92] H. W. Lenstra, Jr., Private communication.

[Li82] J. E. Littlewood, "On the class number of the corpus $P(\sqrt{-k})$", *Proc. London Math. Soc.* 27, 1982, pp. 358-372.

[LKBS92] J. Loxton, D. S. P. Khoo, G. J. Bird, and J. Seberry, "A cubic RSA code equivalent to factorization", *J. of Cryptology*, v. 5, no. 2, 1992, pp. 139-150.

[LLMP90] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The number field sieve", to appear, Extended Abstract in: *Proc. 22nd Annual ACM Symp. on the Theory of Computing*, Baltimore (Maryland), May 14-16, 1990, pp. 564-572.

[LLMP93] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The factorization of the ninth Fermat number", to appear in *Math. Comp.*

[LM84] R. Lidl and W. B. Mueller, "Permutation polynomials in RSA cryptosystems", *Advances in Cryptology – Proceedings of CRYPTO '83*, Plenum Press, New York, 1984, pp. 293-301.

[Lo92] R. Lovorn, *Rigorous, subexponential algorithms for disccrete logarithms over finite fields*, Ph.D. Thesis, University of Georgia, May 1992.

[LP92] H. W. Lenstra, Jr. and C. Pomerance, "A rigorous time bound for factoring integers", *J. Amer. Math. Soc.* 5, 1992, pp. 483-516.

[Ma83] J. L. Massey, "Logarithms in finite cyclic groups – cryptographic issues", *Proc. 4th Benelux Symposium on Information Theory*, 1983, pp. 17-25.

[MB75] M. Morrison and J. Brillhart, "A method of factoring and the factorization of F_7", *Math. Comp.* 29, 1975, pp. 183-205.

[Mc88] K. S. McCurley, "A key distribution scheme based on factoring", *Journal of Cryptology* 1, 1988, pp. 95-205.

[Mc89] K. S. McCurley, "Cryptographic key distribution and computation in class groups", *Number Theory and Applications*, Kluwer, Dordrecht (The Netherlands), 1989, pp. 459-479.

[Mc90] K. S. McCurley, "The discrete logarithm problem", *Proc. Symposia in Applied Mathematics* 42, 1990, pp. 49-73.

[Me78] R. C. Merkle, "Secure communication over insecure channels", *Comm. ACM* 21, 1978, pp. 294-299.

[Me79] R. Merkle, *Secrecy, Authentication, and Public Key Systems*, Ph.D. dissertation, Electrical Engineering Department, Stanford University, 1979.

[Mi86] V. Miller, "Use of elliptic curves in cryptography", *Advances in Cryptology – Proceedings of Crypto '85*, Springer-Verlag, New York, 1986, pp. 417-426.

[MM76] J. M. Masley and H. L. Montgomery, "Cyclotomic fields with unique factorization", *J. Reine Angew. Math.* 286/287, 1976, pp. 248-256.

[MN81] W. B. Mueller and W. Noebauer, "Some remarks on public-key cryptosystems", *Studia Sci. Math. Hung.* 16, 1981, pp. 71-76.

[MOV91] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *Proc. 23rd Annual Symp. on the Theory of Computing*, 1991, pp. 80-89.

[NBS77] "Data Encryption Standard", FIPS Publication 46, National Bureau of Standards, Washington, D.C., January 15, 1977.

[Od84] A. M. Odlyzko, "Discrete logarithms and their cryptographic significance", *Advances in Cryptology – Proceedings of EUROCRYPT '84*, Springer Verlag, Heidelberg (Germany), 1984, pp. 224-314.

[OVS84] R. W. K. Odoni, V. Varadharajan, and P. W. Sanders, "Public-key distribution in matrix rings", *Electronic Letters* 20, 1984, pp. 386-387.

[PH78] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Trans. Inf. Theory*, vol. IT-24, no. 1, January 1978, 106-110.

[Po74] J. M. Pollard, "Theorems on factorization and primality testing", *Proc. Cambridge Philos. Soc.* 76, 1974, pp. 521-528.

[Po75] J. M. Pollard, "A Monte Carlo Method for factorization", *BIT* 15, 1975, pp. 331-334.

[Po78] J. M. Pollard, "Monte Carlo methods for index computation mod p", *Math. Comp.* 32, 1978, pp. 918-924.

[Po85] C. Pomerance, "The quadratic sieve factoring algorithm", *Advances in Cryptology – Proceedings of EUROCRYPT '84*, Springer Verlag, Heidelberg (Germany), 1985, pp. 169-182.

[Po87] C. Pomerance, "Fast, rigorous factorization and discrete logarithm algorithms", *Discrete Algorithms and Complexity: Proc. Japan-U.S. Joint Seminar*, June 4, 1986, Kyoto, Japan, Academic Press, Orlando (Florida), 1987, pp. 119-143.

[Ra79] M. O. Rabin, *Digitized Signatures and Public-Key Functions as Intractable as Factorization*, M.I.T. Lab for Computer Science, Tech. Rep. LCS/TR-212, 1979.

[Ri78] R. L. Rivest, "Remarks on a proposed cryptanalytic attack on the MIT public-key cryptosystem", *Cryptologia*, vol. 2, no. 1, January 1978, pp. 62-65.

[Ri79] R. L. Rivest, "Critical remarks on "Critical remarks on some public-key cryptosystems", *BIT*, vol. 19, no. 2, 1979, pp. 274-275.

[RSA78] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Comm. ACM*, vol. 21, no. 2, February 1978, pp. 120-126.

[Sa90] A. Salomaa, *Public-Key Cryptography*, Springer-Verlag, Berlin, 1990.

[SBW92] R. Scheidler, J. A. Buchamnn and H. C. Williams, "Implementation of a key exchange protocol using real quadratic fields", to appear in *Journal of Cryptology.*

[Sc83a] R. J. Schoof, "Quadratic Fields and Factorization", *Computational Methods in Number Theory* (H. W. Lenstra, Jr. and R. Tijdeman, eds.), Math. Centrum Tracts, no. 155, part II, Amsterdam, 1983, pp. 235-286.

[Sc83b] C. P. Schnorr, "Is the RSA scheme safe?", *Cryptography Proceedings*, Burg Feuerstein, 1982, Springer Verlag, Heidelberg (Germany), 1983, pp. 325-329.

[Sc90] C. P. Schnorr, "Efficient identification and signatures for smart cards", *Advances in Cryptology – CRYPTO '89 Proceedings*, Springer-Verlag, Berlin, 1990, pp. 239-252.

[Se87] M. Seysen, "A probabilistic factorization algorithm with quadratic forms of negative discriminant", *Math. Comp.* 48, 1987, pp. 757-780.

[Sh49] C. E. Shannon, "Communication theory of secrecy systems", *Bell Syst. Tech. J.* 28, October 1949, pp. 656-715.

[Sh72] D. Shanks, "The infrastructure of a real quadratic field and its applications", *Proc. 1972 Number Theory Conference*, Boulder (Colorado), 1972, pp. 217-224.

[Sh73] D. Shanks, "Five number-theoretic algorithms", *Proc. Second Manitoba Conference on Numerical Mathematics, October 5-7, 1992 , Congressus Numerantium* VII, Utilitas Mathematica, Winnipeg (Canada), 1973, pp. 51-70.

[Sh85] Z. Shmuely, *Composite Diffie-Hellman Public-Key Generating Systems are Hard to Break*, Technical Report # 356, Computer Science Department, Technion-Israel Institute of Technology, February 1985.

[Sh90] J. Shallit, "On the worst case of three algorithms for computing the Jacobi symbol", *J. Symb. Comp.* 10, 1990, pp. 593-610.

[SJ73] E. Smith and K. Jacobs, *Introductory Astronomy and Astrophysics*, W. B. Saunders Co., Philadelphia 1973.

[SL93] P. Smith, M. Lennon, "LUC: A new public key system", *preprint.*

[Sm65] H. J. S. Smith, *Report on the Theory of Numbers*, Chelsea, New York 1965.

[Sm73] R. Smadja, "Sur le groupe des classes des corps de nombres", *C. R. Acad. Sc. Paris*, vol. 276, Series A, June 25, 1973, pp. 1639-1641.

[Sm93] P. Smith, "LUC public-key encryption", *Dr. Dobb's Journal*, January 1993, pp. 44-48.

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

[SP79] D. R. Smith and J. T. Palmer, "Universal fixed messages and the Rivest-Shamir-Adleman cryptosystem, *Mathematika* 26, 1979, pp. 44-52.

[SS77] R. Solovay & V. Strassen, "A fast Monte Carlo test for primality", *SIAM J. of Computing* 6, 1977, pp. 84-85.

[St89] A. J. Stephens, *mp : A Multi-Precise Integer Package*, Department of Computer Science, University of Manitoba, June 9, 1989.

[ST79] I. N. Stewart and D. O. Tall, *Algebraic Number Theory*, Chapman and Hall, London, 1979.

[Su65] Suetonius, *The Lives of the Twelve Caesars*, Heritage Press, New York 1965.

[SW89] A. J. Stephens and H. C. Williams, "Some computational results on a problem concerning powerful numbers", *Math. Comp.*, vol. 50, no. 182, April 1989, pp. 619-632.

[SW92] R. Scheidler & H. C. Williams, *A Public-Key Cryptosystem Utilizing Cyclotomic Fields*, Technical Report 15/92, Department of Compter Science, University of Manitoba, November 1992.

[Us09] J. Uspensky, "Note sur les nombres entiers dépendant d'une racine cinquième de l'unité", *Math. Ann.* 66, 1909, pp. 109-112.

[Va85] V. Vaharadharajan, "Extension of RSA cryptosystems to matrix rings", *Cryptologia*, vol. 9, no. 2, April 1985, pp. 140-153.

[Wa82] L. C. Washington, *Introduction to Cyclotomic Fields*, Springer-Verlag, New York, 1982.

[Wi76] K. S. Williams, "Explicit forms of Kummer's complementary theorems to his law of quintic reciprocity", *J. Reine Angew. Math.* 288, 1976, pp. 207-210.

[Wi80] H. C. Williams, "A modification of the RSA public-key encryption procedure", *IEEE Trans. Inf. Theory*, vol. IT-26, no. 6, November 1980, pp. 726-729.

[Wi82] H. C. Williams, "A p+1 method of factoring", *Math. Comp.* vol. 39, no. 159, July 1982, pp. 225-234.

[Wi85a] H. C. Williams, "Continued fractions and number-theoretic computations", *Rocky Mountain J. Math.*, vol. 15, no. 2, Spring 1985, pp. 621-655.

[Wi85b] H. C. Williams, "Some public-key crypto-function as intractable as factorization", *Cryptologia*, vol. 9, no. 3, July 1985, pp. 223-237.

[Wi86] H. C. Williams, "An M^3 public-key encryption scheme", *Advances in Cryptology - CRYPTO '85 Proceedings*, Springer, Berlin 1986, pp. 358-368.

[Wi90] M. J. Wiener, "Cryptanalysis of short RSA secret exponents", *IEEE Trans. Inf. Theory*, vol. 36, no. 3, May 1990, pp. 553-558.

[WM68] A. E. Western and J. C. P. Miller, "Tables of indices and primitive roots", *Royal Society Mathematical Tables*, vol. 9, Cambridge University Press, 1968.

[WS79] H. C. Williams and B. Schmid, "Some remarks concerning the M.I.T. public-key cryptosystem", *BIT*, vol. 19, no. 4, 1979, pp. 525-538.

[WW84] P. K. S. Wah and M. Z. Wang, "Realization and application of the Massey-Omura lock", *Proc. International Zürich Seminar*, 1984, pp. 175-182.

[WW87] H. C. Williams and M. C. Wunderlich, "On the parallel generation of the residues for the continued fraction factoring algorithm", *Math. Comp.*, vol. 48, no. 177, January 1987, pp. 405-423.