Lynn Batten
Jennifer Seberry (Eds.)

LNCS 2384

Information Security and Privacy

7th Australasian Conference, ACISP 2002
Melbourne, Australia, July 2002
Proceedings

Springer

Lynn Batten Jennifer Seberry (Eds.

Information Security and Privacy

7th Australasian Conference, ACISP 2002
Melbourne, Australia, July 3-5, 2002
Proceedings

Springer

Volume Editors

Lynn Batten
Deakin University, Rusden Campus
Burwood Road, Melbourne, Victoria, Australia
E-mail: lmbatten@deakin.edu.au

Jennifer Seberry
University of Wollongong, Department of Computer Science
Northfields Avenue, Wollongong, NSW, Australia
E-mail: jennifer.seberry@uow.edu.au

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Information security and privacy : 7th Australasian conference ; proceedings / ACISP 2002, Melbourne, Australia, July 3 - 5, 2002. Lynn Batten ; Jennifer Seberry (ed.). - Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2384)
ISBN 3-540-43861-0

CR Subject Classification (1998): E.3, K.6.5, D.4.6, C.2, E.4, F.2.1, K.4.1

ISSN 0302-9743
ISBN 3-540-43861-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna e. K.
Printed on acid-free paper SPIN 10870504 06/3142 5 4 3 2 1 0

Preface

The Seventh Australasian Conference in Information Security and Privacy (ACISP) was held in Melbourne, 3–5 July, 2002. The conference was sponsored by Deakin University and iCORE, Alberta, Canada and the *Australian Computer Society*.

The aims of the *annual* ACISP conferences have been to bring together people working in different areas of computer, communication, and information security from universities, industry, and government institutions. The conferences give the participants the opportunity to discuss the latest developments in the rapidly growing area of information security and privacy.

The reviewing process took six weeks and we heartily thank all the members of the program committee and the external referees for the many hours of valuable time given to the conference.

The program committee accepted 36 papers from the 94 submitted. From those papers accepted 10 papers were from Australia, 5 each from Korea and USA, 4 each from Singapore and Germany, 2 from Japan, and 1 each from The Netherlands, UK, Spain, Bulgaria, and India. The authors of every paper, whether accepted or not, made a valued contribution to the conference.

In addition to the contributed papers, we were delighted to have presentations from the Victorian Privacy Commissioner, Paul Chadwick, and eminent researchers Professor Hugh Williams, Calgary, Canada, Professor Bimal Roy, ISI, Kolkota, India (whose invited talk was formally referred and accepted by the program committee), and Dr Hank Wolfe from Otago, New Zealand.

In addition we would like to thank Beom Sik Song, Willy Susilo, and especially Ken Finlayson for the vast work they put into getting this volume together in the time available.

July 2002

Lynn Batten
Jennifer Seberry

ACISP 2002

July 3-5, 2002, Melbourne, Australia

General Chair

Lynn Batten, Deakin University, Australia

Program Co-chairs

Lynn Batten, Deakin University, Australia
Jennifer Seberry, University of Wollongong, Australia

Program Committee

Colin Boyd	Queensland University of Technology, Australia
Mike Burmester	Florida State University, USA
Ed Dawson	Queensland University of Technology, Australia
Cunsheng Ding	University of Science & Technology, Hong Kong
Paul England	Microsoft, USA
Dieter Gollman	Microsoft, United Kingdom
Thomas Hardjono	VeriSign, USA
Kathy Horadam	RMIT, Australia
Kwangjo Kim	ICU, South Korea
Lars Knudsen	Technical University of Denmark, Denmark
Keith Martin	Royal Holloway, United Kingdom
Atsuko Miyaji	JAIST, Japan
Sangjae Moon	Kyungpook National University, South Korea
Yi Mu	Macquarie University, Australia
Eiji Okamoto	Toho University, Japan
Josef Pieprzyk	Macquarie University, Australia
Greg Rose	QUALCOMM, Australia
Rei Safavi-Naini	University of Wollongong, Australia
Qing Sihan	Academy of Science, China
John Snare	Adacel, Australia
Vijay Varadharajan	Macquarie University, Australia
Hugh Williams	University of Calgary, Canada
Yuliang Zheng	University of North Carolina, USA

External reviewers

Joonsang Baek, Monash University, Australia
Asha Baliga, RMIT University, Australia
Niklas Borselius, Royal Holloway, United Kingdom
Serdar Boztas, RMIT University, Australia
Laurence Bull, Monash University, Australia
Bernard Colbert, Telstra Research Laboratories, Australia
Robert Coulter, Deakin University, Australia
Ken Finlayson, Wollongong University, Australia
Goichiro Hanaoka, University of Tokyo, Japan
Marie Henderson, RMIT University, Australia
Matt Henricksen, QUT, Australia
Yvonne Hithchenson, QUT, Australia
Hartono Kurino, Wollongong University, Australia
Hiroaki Kikuchi, Japan
Jun Kogre, Fujitsu, Japan
Tanja Lange, Ruhr University, Germany
Bill Millan, QUT, Australia
Kenji Ohkuma, Toshiba, Japan
Marcus Peinado, Microsoft, USA
Ian Piper, Wollongong University, Australia
Chengxi Qu, University of New England, Australia
Matt Robshaw, Royal Holloway, United Kingdom
Nickolas Sheppard, Wollongong University, Australia
Igor Shparlinski, Macquarie University, Australia
Leonie Simpson, QUT, Australia
Masakazu Soshi, JAIST, Japan
Ron Steinfeld, Monash University, Australia
Karolyn Sprinks, Wollongong University, Australia
Willy Susilo, Wollongong University, Australia
Mitsuru Tada, Chiba University, Japan
Kapali Viswanthan, QUT, Australia
Yejing Wang, Wollongong University, Australia
Huaxiong Wang, Macquarie University, Australia
Tianbing Xia, Wollongong University, Australia
Masato Yamamichi, Japan
Jin Yuan, Hong Kong University of Science and Technology
Fangguo Zhang, ICU, Korea
Xianmo Zhang, Macquarie University, Australia

Table of Contents

Cryptanalysis I

Elliptic Curves

AES

Security Management

Authentication

Invited Talk

Oblivious Transfer

Cryptanalysis II

Dealing with Adversaries

A New Distributed Primality Test for Shared RSA Keys Using Quadratic Fields

Ingrid Biehl and Tsuyoshi Takagi

Technische Universität Darmstadt, Fachbereich Informatik,
Alexanderstr. 10, D-64283, Darmstadt, Germany
ttakagi@cdc.informatik.tu-darmstadt.de

Abstract. In the generation method for RSA-moduli proposed by Boneh and Franklin in [BF97] the partial signing servers generate random shares p_i, q_i and compute as candidate for an RSA-modulus $n = pq$ where $p = (\sum p_i)$ and $q = (\sum q_i)$. Then they perform a time-consuming distributed primality test which *simultaneously* checks the primality both of p and q by computing $g^{(p-1)(q-1)} = 1 \bmod n$. The primality test proposed in [BF97] cannot be generalized to products of more than two primes. A more complicated one for products of three primes was presented in [BH98].

In this paper we propose a new distributed primality test, which can *independently* prove the primality of p or q for the public modulus $n = pq$ and can be easily generalized to products of arbitrarily many factors, i.e., the Multi-Prime RSA of PKCS #1 v2.0 Amendment 1.0 [PKCS]. The proposed scheme can be applied *in parallel* for each factor p and q. We use properties of the group $Cl(-8n^2)$, which is the class group of the quadratic field with discriminant $-8n^2$.

As it is the case with the Boneh-Franklin protocol our protocol is $\lfloor \frac{k-1}{2} \rfloor$-private, i.e. less than $\lfloor \frac{k-1}{2} \rfloor$ colluding servers cannot learn any information about the primes of the generated modulus. The security of the proposed scheme is based on the intractability of the discrete logarithm problem in $Cl(-8n^2)$ and on the intractability of a new number theoretic problem which seems to be intractable too.

Keywords: Distributed RSA, primality test, parallel computation, quadratic fields.

1 Introduction

In recent literature the usage of distributed digital signature schemes is discussed as cost-friendly alternative for high security trust center applications. This allows to get rid of expensive measures to serve for the organizational security of a single signing server as it is the common practice in today's realizations. Even for the process of the generation of the secret keys, methods are known which allow a distributed computation among so-called partial signing servers, which guarantee the correctness of the result while preventing single parties from learning something about the secret keys.

L. Batten and J. Seberry (Eds.): ACISP 2002, LNCS 2384, pp. 1–16, 2002.

Here we consider the case of the generation of RSA-like moduli which is part of the distributed generation process of RSA keys. The distributed RSA-modulus generation by Boneh-Franklin in [BF97] consists of two steps. In the first step each server $i(i = 1, 2, ..., k)$ generates shares p_i, q_i of numbers p, q where $n = pq$ is a candidate for an RSA-modulus. The shares have to be kept secret while all servers generate the common public modulus $n = pq = (\sum_i p_i)(\sum_i q_i)$ by means of the so-called BGW protocol [BGW88], which is proved to leak no information about the shares p_i, q_i and about p, q apart from the value n as far as less than $\lfloor \frac{k-1}{2} \rfloor$ parties collude.

To come to an RSA-modulus, p and q have to be primes. If one of them is composite the whole process has to be restarted again. Thus the expected number of repetitions is approximately ℓ^2, if p and q are ℓ-bit numbers. To check the primality in the second step a distributed primality test has to be engaged, which checks simultaneously the primality of p and q. Thus, the costs for the primality check of each candidate pair and the expected number of repetitions to find a correct RSA-modulus are the reason for the considerable running time of this approach.

In more details the test has the following form: At first trial division is made to eliminate candidate pairs, which contain small divisors. Then the candidate pair is checked by means of the Fermat test, i.e. an integer $g \in \mathbb{Z}/n\mathbb{Z}$ is randomly chosen and the servers work together to check whether $g^{n+1} \equiv \prod_i g^{p_i+q_i} \bmod n$. To do so $g^{p_i+q_i} \bmod n$ is locally computed by server S_i and is sent to the other servers. The secret shares $p_i, q_i (i = 1, 2, ..., k)$ are not revealed. Notice that $g^{n+1} \equiv \prod_i g^{p_i+q_i} \bmod n$ is equivalent to $g^{(p-1)(q-1)} \equiv 1 \bmod n$ since $(p-1)(q-1) = n+1-\sum p_i - \sum q_i$. Since there are integers which pass this Fermat test with high probability even if they are composite, in a last step Boneh and Franklin engage a Fermat test in a more complicated group to cope with these cases. The whole test is a probabilistic test and it has to be iterated to guarantee with high probability that $n = pq$ is a product of two primes. If the primality test fails, then the whole procedure starting with the choice of distributed p and q has to be repeated.

After generating shared primes p_i, q_i, a public exponent e and secret shares d_i of a secret exponent $d = (\sum_i d_i)$ with $ed \equiv 1 \bmod (p-1)(q-1)$ are distributively computed. Catalo et al. proposed an efficient protocol to compute a sharing of d [CGH00]. Then the partial signing servers easily can sign messages m by individually publishing $s_i = m^{d_i} \bmod n$. Verification is done as usual by checking whether $(\prod_i s_i)^e \equiv m \bmod n$. Miyazaki et al. proposed a protocol to achieve a k-out-of-n threshold signature for $k < n$ [MSY01], which is based on the Simmons' protocol-failure of RSA cryptosystem. A similar construction was used for the distributed RSA, but it requires a trusted dealer and strong primes for its security proof [Sho99]. Dåmgard and Koprowski dropped the conditions, namely that the modulus must be a product of safe primes and that a trusted dealer generates the keys [DK01]. Recently, Fouque and Stern proposed a distributed RSA key generation for any type of RSA modulus [FS01]. We can combine these results to our proposed distributed primality test and construct a distributed RSA cryptosystem.

Contribution of This Paper

In this paper we give a new distributed primality test which can *independently* prove the primality of p or q for the public modulus $n = pq$. The new distributed primality test is based on the ideal arithmetic of non-maximal quadratic orders of quadratic fields. We use the map between two different class groups of non-maximal orders, namely $Cl(-8n^2)$ and $Cl(-8q^2)$. The kernel of the map $\varphi_p : Cl(-8n^2) \to Cl(-8q^2)$ is a cyclic group with order $p-(-2/p)$, where $(\cdot/p)$ is the Jacobi symbol modulo p. We give an algorithm, which distributively generates an ideal $\mathfrak{p}$ in the kernel of the map φ_p. Then we can check the primality of p by checking whether $\mathfrak{p}^{p\pm 1} \stackrel{?}{=} 1 \in Cl(-8n^2)$. Analogously we check the primality of q. Thus the proposed scheme can be applied *in parallel* for each factor p and q.

The security of the proposed distributed primality test depends on the discrete logarithm problem in $Cl(-8n^2)$ and a number theoretic problem, which can be characterized as follows and which seems to be intractable: find p or q given pairs of ideals $(\mathfrak{a}_1, \mathfrak{a}_2)$ and $(\mathfrak{b}_1, \mathfrak{b}_2)$ in $\mathcal{O}_{-8n^2}$, where $\widetilde{\varphi_p}(\mathfrak{a}_1) = \mathfrak{a}_1\mathcal{O}_{-8q^2}$, $\widetilde{\varphi_p}(\mathfrak{a}_2) = \mathfrak{a}_2\mathcal{O}_{-8q^2}$ are equivalent with exactly one reduction step in $\mathcal{O}_{-8q^2}$ and $\widetilde{\varphi_q}(\mathfrak{b}_1) = \mathfrak{b}_1\mathcal{O}_{-8p^2}$, $\widetilde{\varphi_q}(\mathfrak{b}_2) = \mathfrak{b}_2\mathcal{O}_{-8p^2}$ are equivalent with exactly one reduction step in $\mathcal{O}_{-8p^2}$.

To check a factor it is sufficient to do this test once. Thus the new test is a good candidate for a more efficient practical test compared to the well-known tests published so far. As another advantage the proposed distributed primality test can be easily generalized to products of different form, for example to products $n = p_1p_2p_3$ for primes p_1, p_2, p_3. We can apply it to the Multi-Prime RSA of PKCS # 1 v2.0 Amendment 1.0 [PKCS]. This extension appears more natural and efficient than the method proposed by Boneh and Horwitz in [BH98].

As in the Boneh-Franklin method we assume the servers to be honest but curious, i.e. they follow honestly the protocol but may try to deduce information about the factors of the candidate RSA-modulus by means of the exchanged information. Moreover we suppose that there is a secure communication channel between each pair of parties. Although our method can be generalized we concentrate for reason of simplicity on an k-out-of-k scheme, i.e. all k servers are needed to generate and test a candidate RSA-modulus. As it is the case with the Boneh-Franklin protocol our protocol is $\lfloor\frac{k-1}{2}\rfloor$-private, i.e. less than $\lfloor\frac{k-1}{2}\rfloor$ colluding servers cannot learn any information about the primes of the generated modulus.

In Section 2 we sketch the method of Boneh and Franklin. In Section 3 we present a new distributed multiplication method DistMult which is a variant of the BGW method and allows to distributively compute the product of arbitrarily many shared integers. This protocol will serve as a subroutine in our primality test. In Section 4 we will introduce as necessary basics ideals of quadratic orders and prove the mentioned properties of the maps φ_p. Then we present in Section 5 the new distributed primality test and analyze it.

2 RSA-Modulus Generation by Boneh-Franklin

In this section we sketch the distributed generation of an RSA-modulus by k servers as it is proposed by Boneh-Franklin in [BF97]. Let $S_i (i = 1, 2, ..., k)$ be the servers which are connected to each other by means of a secure channel. The server S_i locally generates two random integers p_i, q_i, and keeps them secret. Then all of them work together to generate and publishes the modulus $n = pq$ with $p = \sum_i p_i, q = \sum_i q_i$ by means of the BGW protocol [BGW88], which does not reveal any information about the partial shares $p_i, q_i (i = 1, 2, ..., k)$ or about p or q. The BGW protocol (adapted from protocols proposed in [BGW88]) and thus the Boneh-Franklin protocol is at most $\lfloor \frac{k-1}{2} \rfloor$-private, i.e. any coalition of $\lfloor (k-1)/2 \rfloor$ servers cannot learn any information which cannot be directly derived by n and their own shares.

In the following we will present a similar protocol, which allows to compute (additive) shares of the product of two (additively) shared numbers p, q.

After generation of the composite modulus in the protocol by Boneh and Franklin trial division is applied with primes up to some not too large bound B. Then the distributed Fermat primality test is applied: for a random element g in $\mathbb{Z}/n\mathbb{Z}$ it is checked whether $g^{n+1} \stackrel{?}{\equiv} \prod_i v_i \bmod n$, where $v_i \equiv g^{p_i+q_i} \bmod n$ and v_i is locally computed and published by the server S_i. Since $n+1-\sum_i (p_i+q_i) = (p-1)(q-1)$ the test is equivalent to the test of $g^{(p-1)(q-1)} \equiv 1 \bmod n$. As this relation might hold although n is not of the correct form it has to be repeated several times for different g. Unfortunately there are composite numbers n which are not the product of two primes but always pass the distributed Fermat test (for all g) though. Accordingly to Boneh and Franklin the probability to get such an integer is very small.

Apart from these rare exceptions the probability for g to lead to an accepting test is at most $1/2$ if n is not of the correct form. Thus it is sufficient to repeat the test t times to guarantee with probability $1 - 1/2^t$ that the found number is of the correct form.

The security of the test is based on the intractability of the discrete logarithm problem: To compute the shares p_i, q_i one has to solve the discrete logarithm problem for v_i which is supposed to be an intractable problem. Therefore, this distributed primality test is computationally secure and no server S_i can learn information about the shares p_j, q_j $(j = 1, 2, ..., k, j \neq i)$.

3 Distributive Multiplication of Shared Integers

We present a new variant of the BGW protocol, which is used in Section 5.

Algorithm 1 (DistMult)
Input: A prime $r > pq$ and $r > k$. Each party S_i has shares p_i, q_i $(i = 1, \ldots, k)$ such that $p \equiv \sum_i p_i \bmod r, q \equiv \sum_i q_i \bmod r$.
Output: Each party has a share w_i $(i = 1, \ldots, k)$ such that $pq \equiv \sum_i w_i \bmod r$.

1. Let $i = 1, 2, ..., k$ and $l = \lfloor \frac{k-1}{2} \rfloor$. The party S_i generates random polynomials $f_i(x), g_i(x) \in \mathbb{Z}_r[x]$ of degree l with $f_i(0) \equiv p_i \bmod r, g_i(0) \equiv q_i \bmod r$, a random polynomial $h_i(x) \in \mathbb{Z}_r[x]$ of degree $2l$ and keeps these polynomials secret. Moreover it sets $w_i \equiv -h_i(0) \bmod r$ as share.
2. For $j = 1, 2, ..., k$ each party S_i computes $p_{i,j} \equiv f_i(j) \bmod r, q_{i,j} \equiv g_i(j) \bmod r, h_{i,j} \equiv h_i(j) \bmod r$ and sends $p_{i,j}, q_{i,j}, h_{i,j}$ to party j for all $j \neq i$.
3. Each party S_i computes

$$n_i \equiv \left(\sum_{j=1}^{k} p_{j,i}\right)\left(\sum_{j=1}^{k} q_{j,i}\right) + \left(\sum_{j=1}^{k} h_{j,i}\right) \bmod r. \tag{1}$$

   Then party S_i sends n_i to the first party S_1.
   Notice that for $t(x) \equiv ((\sum f_j) * (\sum g_j) + (\sum h_j))(x) \bmod r$ follows: t has degree $2l$ and $n_i \equiv t(i) \bmod r$.
4. Since $k \geq 2l + 1$ the first party S_1 knows enough interpolation points to interpolate $t(x)$. Thus it computes $w_1 \equiv t(0) - h_1(0) \bmod r$ by

$$w_1 \equiv \left(\sum_{i=1}^{k} n_i \cdot \left(\prod_{j \neq i} \frac{j}{j-i}\right)\right) - h_1(0) \bmod r. \tag{2}$$

The protocol is correct since

$$\begin{aligned}
\sum_i w_i &\equiv (t(0) - h_1(0)) + \sum_{i=2}^{k}(-h_i(0)) \\
&\equiv \left(\sum_i f_i(0)\right)\left(\sum_i g_i(0)\right) + \left(\sum_i h_i(0)\right) - \left(\sum_i h_i(0)\right) \\
&\equiv (\sum_i p_i)(\sum_i q_i) \\
&\equiv pq \bmod r.
\end{aligned}$$

The protocol is l-private since less than $l+1$ colluders won't learn any information on any of the used polynomials (compare Shamirs secret sharing method) and each set of $k-1$ numbers appears with the same probability as set of $k-1$ output shares. Thus it is $\lfloor \frac{k-1}{2} \rfloor$-private and one can prove the following theorem:

Proposition 1. *Any coalition of at most* $\lfloor \frac{k-1}{2} \rfloor$ *parties can simulate the transcript of the* DistMult *protocol, thus the* DistMult *protocol is* $\lfloor \frac{k-1}{2} \rfloor$*-private.*

Please notice that the protocol can be modified to compute $pq \bmod a$ for some arbitrary integer a. To do so it has to be applied for each prime divisor r of a and the results have to be combined by the usual methods of Chinese remaindering and Hensel lifting. Even if the prime factorization of a is not known the protocol can be applied and either works or leads to a refinement of the factorization of a. In the latter case it has to be restarted with the newly improved

partial factorization. Thus this method terminates and causes at most $O(\log |a|)$ restarts.

Moreover, the protocol can repeatedly be applied to compute the product of arbitrarily many factors $m = g_1 \dots g_c$ supposed there is a prime $r > m$ and each factor g_j is shared among the parties by shares $g_j^{(i)}$ such that $\sum_i g_j^{(i)} \equiv g_j \bmod r$. Using a tree-like multiplication sequence one needs about $O(\log c)$ applications of the above protocol.

In the original BGW protocol the product value is publicly known after the protocol. Obviously this can be achieved by publishing all w_i in the DistMult protocol.

4 Mathematical Backgroud

4.1 Quadratic Order

In this section we will explain the arithmetic of ideals of quadratic orders which we will use in this paper. A more comprehensive treatment can be found in [Cox89].

A *discriminant* Δ is a non-square integer such that $\Delta \equiv 0, 1 \bmod 4$. It is called *fundamental* if $\Delta \equiv 1 \bmod 4$ and is square-free, or $\Delta/4 \equiv 2, 3 \bmod 4$ and is square-free. In this paper we use only negative discriminants. The *quadratic field* of discriminant Δ is $\mathbb{Q}(\sqrt{\Delta}) = \mathbb{Q} + \sqrt{\Delta}\mathbb{Q}$. The *quadratic order* of discriminant Δ is $\mathcal{O}_\Delta = \mathbb{Z} + \frac{\Delta+\sqrt{\Delta}}{2}\mathbb{Z}$. Every element $\alpha \in \mathcal{O}_\Delta$ can be represented as $\alpha = (x + y\sqrt{\Delta})/2$ for some $x, y \in \mathbb{Z}$. Every ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$ can be represented by

$$\mathfrak{a} = m\left(a\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2}\mathbb{Z}\right), \tag{3}$$

where $m \in \mathbb{Z}$, $a \in \mathbb{Z}_{>0}$, and $b \in \mathbb{Z}$ such that $b^2 \equiv \Delta \pmod{4a}$ [BW88]. When a is a prime integer, then we call $\mathfrak{a}$ a *prime ideal*. The *norm* of an ideal $\mathfrak{a}$ is defined by $N(\mathfrak{a}) = aq^2$. A *fractional ideal* $\mathfrak{a}$ of $\mathcal{O}_\Delta$ is a subset of $\mathbb{Q}(\sqrt{\Delta})$ of the form $\mathfrak{a} = q\left(a\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2}\mathbb{Z}\right)$, where $q = m/d \in \mathbb{Q}$ and a, b, and m satisfy the criteria in equation (3). Then (q, a, b) is called the *standard representation* of ideal $\mathfrak{a}$. If $q = 1$ holds for an ideal $\mathfrak{a}$, then the ideal $\mathfrak{a}$ is called *integral*. If $q = 1$ holds for an ideal $\mathfrak{a}$, then $\mathfrak{a}$ is said to be *primitive* and in that case we represent $\mathfrak{a}$ by (a, b). For two given ideals $\mathfrak{a}, \mathfrak{b}$, we can compute their product $\mathfrak{ab}$ which needs $O((\log(\max\{N(\mathfrak{a}), N(\mathfrak{b})\}))^2)$ bit operations (see, for example, [BW88]). A fractional ideal $\mathfrak{a}$ of $\mathcal{O}_\Delta$ is *invertible* if there exits another fractional ideal $\mathfrak{b}$ such that $\mathfrak{ab} = \mathcal{O}_\Delta$. The set of invertible ideals is $\mathcal{I}_\Delta$. For an element $\gamma \in \mathbb{Q}(\sqrt{\Delta})$, the ideal $\mathfrak{a}$ generated by γ is called a *principal* ideal. We denote it by $\mathfrak{a} = (\gamma)$ or $\mathfrak{a} = \gamma\mathcal{O}_\Delta$ and then γ is called the *generator* of the principal ideal $\mathfrak{a}$. The set of principal ideals is denoted by $\mathcal{P}_\Delta$.

Two fractional ideals $\mathfrak{a}$ and $\mathfrak{b}$ of $\mathcal{O}_\Delta$ are called *equivalent* (i.e. $\mathfrak{a} \sim \mathfrak{b}$) if there is $\alpha \in \mathbb{Q}(\sqrt{\Delta})$ such that $\mathfrak{a} = \alpha\mathfrak{b}$. The *class group* $Cl(\Delta)$ of the quadratic order

$\mathcal{O}_\Delta$ is defined as $Cl(\Delta) = \mathcal{I}_\Delta/\mathcal{P}_\Delta$ with respect to the equivalent relation $\sim$. If $\mathfrak{a}$ is a fractional ideal then we denote by $[\mathfrak{a}]$ the corresponding class. For a primitive ideal $\mathfrak{a}$ in $\mathcal{I}_\Delta$, we say that $\mathfrak{a} = (a, b)$ is *reduced* if $|b| \leq a \leq c = (b^2 - \Delta)/4a$ and additionally $b \geq 0$ if $a = c$ or $a = |b|$. There is only one reduced ideal in every equivalence class. For a primitive ideal $\mathfrak{a}$ we denote by $Red_\Delta(\mathfrak{a})$ the uniquely determined reduced ideal equivalent to $\mathfrak{a}$. Define the *reduction operator* $\rho_\Delta((a,b)) = (\frac{b^2-\Delta}{4a}, -b)$ for primitive ideals $\mathfrak{a} = (a, b)$. One can prove that for $\mathfrak{a}' = \rho_\Delta(\mathfrak{a})$ that $\mathfrak{a}^{-1} * \mathfrak{a}'$ is principal. Then one can compute $Red_\Delta(\mathfrak{a})$ by $O(\log^2 N(\mathfrak{a}))$ repeated applications of ρ_Δ (see [BB97]).

Every non-fundamental discriminant Δ can be represented by $\Delta = \Delta_1 f^2$, where Δ_1 is the fundamental discriminant and f is a positive integer called the *conductor* (we write Δ_f instead of $\Delta_1 f^2$). Moreover, the order $\mathcal{O}_\Delta = \mathbb{Z} + f\mathcal{O}_{\Delta_1}$ is called *non-maximal order* with conductor f and $\mathcal{O}_{\Delta_1}$ is called the *maximal order*. For a quadratic order $\mathcal{O}_\Delta$ with conductor f, we say that a non-zero fractional ideal $\mathfrak{a}$ is *prime* to f if the denominator and the numerator of $N(\mathfrak{a})$ are relatively prime to f. For a fractional principal ideal $\gamma\mathcal{O}_\Delta$ ($\gamma \in \mathbb{Q}(\Delta)$), we define: $\gamma\mathcal{O}_\Delta$ is *prime to* f if the denominator and the numerator of $N(\gamma)$ are relative prime to f. We denote the subgroup of all fractional ideals prime to f by $\mathcal{I}_\Delta(f)$. The subset of $\mathcal{I}_\Delta(f)$ which is generated by the principal ideals $\gamma\mathcal{O}$ ($\gamma \in \mathbb{Q}(\Delta)$), whose norm is relative prime to f, is a subgroup of $\mathcal{I}_\Delta(f)$ and is denoted by $\mathcal{P}_\Delta(f)$. It is well known that any ideal class in $Cl(\Delta)$ contains an ideal prime to the conductor f. One can prove that $Cl(\Delta) \tilde{\to} \mathcal{I}_\Delta(f)/\mathcal{P}_\Delta(f)$.

4.2 The Idea of a New Primality Test

Let Δ_{fg} be the non-fundamental discriminant $\Delta_1(fg)^2$. The relationship of ideals in the order tower of $\mathcal{O}_{\Delta_{fg}} \subset \mathcal{O}_{\Delta_f} \subset \mathcal{O}_{\Delta_1}$ plays the main role in our proposed distributed primality test. There is a nice structure in the relationship between $Cl(\Delta_{fg})$ and $Cl(\Delta_f)$, namely:

Proposition 2. *Consider the map* $\varphi_g : Cl(\Delta_{fg}) \to Cl(\Delta_f)$ *with* $\varphi_g([\mathfrak{a}]) = [\mathfrak{a}\mathcal{O}_{\Delta_f}]$ *then*

$$\#Ker(\varphi_g) = g \prod_{p|g} \left(1 - \left(\frac{\Delta_f}{p}\right)\right) \frac{1}{p}.$$

Especially, if g *is prime and* $\left(\frac{\Delta_f}{g}\right) = -1$ *then* $\#Ker(\varphi_g) = g + 1$.

Proposition 3. *Define* $\varphi_g^{-1} : Cl(\Delta_f) \to Cl(\Delta_{fg})$ *by* $\varphi_g^{-1}([\mathfrak{A}]) = [\mathfrak{A} \cap \mathcal{O}_{fg}]$. *Then it follows for all* $\mathfrak{a} \in \mathcal{I}_{\Delta_{fg}}$ *and* $[\mathfrak{b}] = \varphi_g^{-1}(\varphi_g([\mathfrak{a}]))$ *that* $[\mathfrak{a}][\mathfrak{b}]^{-1} \in Ker(\varphi_g)$.

We want to use this for a new primality check for g: The idea is to compute a kernel element $[\mathfrak{p}]$ of φ_g and then to check whether $[\mathfrak{p}^{\#Ker(\varphi_g)}]$ is principal. Thus we have to explain a method to compute such a kernel element.

Since $Cl(\Delta) \tilde{\to} \mathcal{I}_\Delta(d)/\mathcal{P}_\Delta(d)$ for arbitrary discriminant Δ and integer d and this isomorphism is a holomorphism with respect to ideal multiplication, one