Post-Quantum Signatures

Johannes Buchmann Carlos Coronado Martin Döring

Daniela Engelbert Christoph Ludwig

Raphael Overbeck Arthur Schmidt Ulrich Vollmer

Ralf-Philipp Weinmann

October 29, 2004

Abstract

Digital signatures have become a key technology for making the Internet and other IT infrastructures secure. But in 1994 Peter Shor showed that quantum computers can break all digital signature schemes that are used today and in 2001 Chuang and his coworkers implemented Shor's algorithm for the first time on a 7-qubit NMR quantum computer. This paper studies the question: What kind of digital signature algorithms are still secure in the age of quantum computers?

1 Introduction

Digital signatures have become a key technology for making the Internet and other IT infrastructures secure. Digital signatures provide long term authenticity, integrity, and support for non-repudiation of data. Digital signatures are widely used in identification and authentication protocols for example for software downloads. Therefore, secure of digital signature algorithms are crucial for maintaining IT security.

But in 1994 Shor [67] showed that quantum computers can break all digital signature that are used today and in 2001 Chuang et al. [73] implemented Shor's algorithm on a 7-qubit quantum computer. Physicists predict that within the next 15 to 20 years there will be quantum computers that are sufficiently large to implement Shor's ideas for breaking digital signature schemes used in practice.

Naturally the following questions arise: What kind of digital signature schemes do we use when quantum computers exist? What do we know about their security and their efficiency? What is their standardization status? This is what we discuss in this paper.

It turns out that we are far from being able to replace existing digital signature schemes by new ones that are secure against quantum computer attacks. A lot of research and development is still necessary. We have to develop security models for digital signature schemes in the age of quantum computers. We have to identify algorithmic problems that are intractable for quantum computers and that can be used as the security basis for digital signature schemes. We have to design, implement and standardize post-quantum signature schemes and to investigate their security and efficiency.

The paper is organized as follows. In Section 2 we explain the practical relevance of digital signatures for IT-Security today. In Section 3 we discuss the current status of quantum attacks on digital signature schemes. In Section 4 we give an overview over possible candidates for computational problems that are intractable for quantum computers and that can be used as the security basis for digital signature schemes and in Section 2 we describe the signature algorithms that are believed to resist quantum computer attacks. Finally, in Section 7 we identify open research problems.

We would like to thank Dan Bernstein for inventing the notion "post-quantum cryptography" and Detlef Hühnlein, Ulrike Meyer, and Tobias Straub for their suggestions and input.

2 Digital signatures are crucial for secure IT systems

In this section we explain the practical relevance of digital signatures for IT-Security today.

2.1 Legislation

In recent years, most countries worldwide have been adapting legislation and regulations that recognize the legality of a digital signature. Such countries are Argentina, Australia, Austria, Belgium, Bermuda, Brazil, Bulgaria, Canada, Chile, Colombia, Costa Rica, Croatia, Czech Republic, Denmark, Dominican Republic, Ecuador, Estonia, Finland, France, Germany, Greece, Hong Kong, Hungary, India, Ireland, Israel, Italy, Japan, Luxembourg, Malaysia, Malta, Mexico, Netherlands, New Zealand, Nicaragua, Norway, Panama, Peru, Philippines, Poland, Portugal, Puerto Rico, Rumania, Russian Federation, Singapore, Slovak Republic, Slovenia, South Africa, South Korea, Spain, Sweden, Switzerland, Taiwan, Thailand, Trinidad/Tobago Republic, Tunesia, United Kingdom, USA, Uruguay, Venezuela, Vietnam. An overview over digital signature laws worldwide can be found in [24]. In those countries, handwritten signatures that are required by law may be replaced by digital signatures. An example: the US E-Sign law and the EU Directive for Digital Signatures allow insurance companies to forgo archiving the paper records on the condition that the original documents are electronically signed, thereby ensuring document authenticity.

2.2 Technology

IT-technology is ready for the use of electronic signatures. There are many standardized protocols that support digital signatures, for example S/MIME for digitally signing emails and a W3C draft for digitally signing HTML and XML documents. Standard software such as MS Internet Explorer, Word, Outlook, Power Point, Excel, Netscape Messenger, and Adobe Acrobat can digitally sign documents and handle digitally signed documents.

2.3 Applications

The following examples for applications of digital signatures are taken from [9], [6], [40], and [10].

1st American Mortgage lender uses digital signatures to properly present and sign mortgage applications online.

The Federal Aviation Administration (FAA) of the United States uses electronic signatures on a variety of regulatory documents including new pilot applications and renewals.

The German Health Professional card, which will be introduced in 2006, lets medical doctors digitally sign patient medical records.

The "Sistema de Pagos Electronicos Interbancarios" uses electronic signatures for transactions between Mexican banks.

The Trusted Computing Group (TCG) [71], an industry standards body, comprised of computer and device manufacturers, software vendors, and others such as Microsoft, Intel, IBM, HP and AMD, has specified the Trusted Platform Module (TPM) for enhancing the security of desktop computers. The TPM is a crypto-processor that provides digital signatures and other cryptographic functionality.

3 Quantum computers will break all digital signatures used today

The first and still most popular digital signature algorithm is RSA. RSA [63]. The security of RSA is based on the intractability of the integer factorization problem. There are a few other digital signature schemes that are used in practice, for example, the Digital Signature Algorithm DSA and the Elliptic Curve Digital Signature Algorithm ECDSA. The security of those schemes is based on the discrete logarithm problem in the multiplicative group of a prime field or in the group of points of an elliptic curve over a finite field. All digital signature algorithms used in practice can be found in the IEEE standard P1363 [41].

In 1994 Peter Shor [67], at AT&T's Bell Labs in New Jersey, discovered a remarkable quantum algorithm. It solves both the factoring problem and the discrete log problem in finite fields and on elliptic curves in polynomial time. So Shor's algorithm breaks all digital signature schemes in use today. Its invention sparked a tremendous interest in quantum computers, even outside the physics community. The core question is: can quantum computers be built in practice?

We give a brief history of quantum computers (see [36]).

In 1981 in his talk enitled "Simulating Physics With Computers" the famous physicist Richard Feynman made the first proposal for using quantum phenomena to perform computations. In 1985 David Deutsch, at the University of Oxford, described the first universal quantum computer. In 1993 Dan Simon [68], at Université de Montreal, invented an oracle problem for which quantum computers would be exponentially faster than conventional computers. This algorithm introduced the main ideas which were then developed in Peter Shor's factoring algorithm in 1994. In 1997 David Cory, A.F. Fahmy and Timothy Havel, and at the same time Neil Gershenfeld and Isaac Chuang at MIT published the first papers on quantum computers based on bulk spin resonance, or thermal ensembles. In 1998 the first working 2-qubit NMR computer was demonstrated at the University of California, Berkeley. In 1999 the first working 3-qubit NMR computer was demonstrated at IBM's Almaden Research Center. In 2000 the first working 5-qubit NMR computer and in 2001 the first working 7-qubit NMR computer were built at IBM's Almaden Research Center by Chuang and co-workers [73]. The 7-qubit computer factored the number 15 using Shor's algorithm. Although no bigger quantum computer has been built so far, there is remarkable progress in quantum computer technology.

In 1985, the U.S. Government began funding research on quantum computers when physicists brought it to their attention that a quantum computer could potentially cripple national security. Corporations such as IBM, Boeing, Hewlett-Packard, or Microsoft and science-based educational institutions such as MIT, Caltech, or Stanford joined the bandwagon and committed funds and full-time resources to studying quantum computers. And remarkably, in April 2004, the founders of the University of Waterloo (UW) at Ontario, Canada, donated $33.3 million to UW's Institute for Quantum Computing bringing their research funding total to $100 million. An overview over quantum computing projects can be found in [61].

There is a good chance that large quantum computers can be built within the next 20 years. This would be a nightmare for IT security if there are no fully developed, implemented, and standardized post-quantum signature schemes.

4 Problems intractable for quantum computers

A necessary condition for the existence of a post-quantum signature scheme is the existence of a computational problem that is intractable for quantum computers and can be used as the security basis for a signature scheme. But currently, no signature scheme is known that is provably hard to break for conventional computers. So there is no hope to find an appropriate computational problem that is provably intractable for quantum computers.

However, there are a few results from complexity theory and there are a few candidates for computational problems which we review in this section.

4.1 Complexity theory

Nielsen and Chuang [59] give heuristic arguments that quantum computers cannot efficiently solve NP-hard problems. On the other hand, it has been shown by Brassard [13] that the security of a deterministic signature scheme cannot be reduced to the intractability of an NP-hard problem. Crepeau [11] shows that quantum cryptography cannot be used to design signature schemes. However, it is possible to use quantum algorithms in conventional signature schemes. For example, Okamato et al. [72] suggest such a scheme. So complexity theory does not really give us a hint where to look for appropriate computational problems.

4.2 CVP and related problems

A serious candidate for quantum-hard computational problems are lattice problems.

Let L be a *lattice* in $\mathbb{Z}^n$, $n \in \mathbb{N}$, that is, L is a subgroup L of $\mathbb{Z}^n$. The lattice L can be written as

$$L = \mathbb{Z}\mathbf{b}_1 + \cdots + \mathbb{Z}\mathbf{b}_k = \{\sum_{j=1}^{k} x_j \mathbf{b}_j : x_j \in \mathbb{Z}\}. \tag{1}$$

where the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{Z}^n$ are linearly independent. The *dimension* of L is k. The dimension of L is uniquely determined. The sequence $B = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is called a *basis* of L. The set of all bases of L is

$$B\mathrm{GL}_k(\mathbb{Z}) = \{BT : T \in \mathrm{GL}_k(\mathbb{Z})\} \tag{2}$$

where $\mathrm{GL}_k(\mathbb{Z})$ is the set of all invertible matrices in $\mathbb{Z}^{(k,k)}$, the set of all k by k matrices with integer entries. By the *length* of a vector $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{R}^n$ we mean its *euclidean length*

$$||\mathbf{v}|| = \sqrt{\sum_{i=1}^{n} v_i^2}. \tag{3}$$

The ith *successive minimum* of L, $1 \le i \le k$ is the radius of the smallest sphere that contains i linearly independent lattice vectors. It is denoted by $\lambda_i(L)$. In particular, $\lambda_1(L)$ is the length of a shortest nonzero lattice vector.

Lattices were introduced by Minkowski [56] in the geometry of numbers, a method which allows the solution of number theoretic problems by geometric and analytic means. There are various hard lattice problems that are used in cryptography. We describe the most important computational lattice problems. In this description we only consider lattices of dimension n in $\mathbb{Z}^n$ for some n. A lattice is represented by a lattice basis.

The most important problem in our context is the following.

Problem 1 *γ-closest vector problem (γ-CVP). Given a lattice L in $\mathbb{Z}^n$ for $n \in \mathbb{N}$, $\mathbf{x} \in \mathbb{Z}^n$, and $\gamma > 0$. Find a lattice vector $\mathbf{v}$ such that $||\mathbf{x} - \mathbf{v}|| \le \gamma||\mathbf{x} - \mathbf{w}||$ for all $\mathbf{w} \in L$.*

For $\gamma = 1$ this problem is called the *closest vector problem (CVP)*

Closely related to γ-CVP is the following problem

Problem 2 *γ-shortest vector problem (γ-SVP). Given a lattice L in $\mathbb{Z}^n$ for $n \in \mathbb{N}$ and $\gamma > 0$. Find a nonzero lattice vector $\mathbf{v}$ such that $||\mathbf{v}|| \le \gamma||\mathbf{w}||$ for all nonzero $\mathbf{w} \in L$.*

For $\gamma = 1$ this problem is called the *shortest vector problem (SVP)*

Ajtai [1] shows that SVP is NP-hard for randomized reduction. There is no cryptosystem whose security can be reduced to the intractability of SVP. However, the security of the cryptosystems of Ajtai-Dwork [3] and Regev [62] can be reduced to SVP in a subclass of lattices in which the shortest nonzero vector is unique up to sign (uSVP). Micciancio [52] proves that γ-SVP is NP-hard for randomized reduction if $\gamma < \sqrt{2}$. Van Emde-Boas [26] shows that CVP is NP-hard. Also, Arora et al. [8] showx that γ-CVP is NP-hard for $\gamma = (\log n)^c$ for every $c > 0$. Goldwasser and Goldreich give a complexity theoretic argument that γ-CVP cannot be NP-hard for $\gamma = \Omega(\sqrt{n}/\log n)$. A more detailed discussion of the complexity of lattice problems can be found in [53].

To solve γ-CVP in practice, the problem is reduced to some γ'-SVP by modifying the lattice appropriately. The famous LLL algorithm [46] solves α^{n-1}-SVP in polynomial time for any $\alpha > 0$. On the other hand, the algorithm of Kannan [42] solves SVP in exponential time. There are several improvements of the LLL algorithm. The BKZ algorithm [66] allows to approximate the shortest vector in a lattice much better than the LLL algorithm using more time. Recently, Schnorr [65] has suggested a heuristic sampling reduction technique that is expected to be very efficient in practice. Ludwig [47] has shown how to make Schnorr's algorithm even more efficient using quantum computers.

Ludwig has made experiments with the LLL and BKZ algorithm. Table 1 shows timings for successful CVP solutions. The lattice basis is the Hermite Normal Form (HNF) of a randomly selected matrix $B \in \{-n, \ldots, n\}^{(n,n)}$. The distance of the target vector from the closest lattice vector is $\min\{\|\bar{b}_i^*\|/2 : i = 1, \ldots, n\}$ where $[\bar{b}_1^*, \ldots, \bar{b}_n^*]$ is the Gram-Schmidt orthogonalization of the LLL reduction $\bar{B} = LLL(B)$ of the original lattice basis. That choice is appropriate to study the cryptanalysis of the Micciancio cryptosystem [51]. The signature variant of that system is described in Section 5.1. The timings are given in seconds of CPU time on a SunBlade 100 (500 MHz UltraSparc IIE Processor, 1 GByte RAM). The spikes in the graph are due to the necessary switch to the more powerful but slower BKZ reduction with BKZ parameter $\beta = 20$.

Ludwig has also studied the impact of the random sampling algorithm on BKZ reduced bases. Table 2 shows by which factor the results of one random sampling iteration were shorter than the shortest vectors of the original BKZ reduced basis. All bases were generated by the method suggested by Ajtai in [2].

Those results show that BKZ-reduction is sucessfull for quite high dimensions and Schnorr random sampling is a real improvement.

4.3 Coding theory

In this section we introduce the decoding problem, another computational problem that resists quantum computer attacks so far.

Let $n \in \mathbb{N}$ and let $F = \{0, 1\}$ be the field of two elements. Consider the

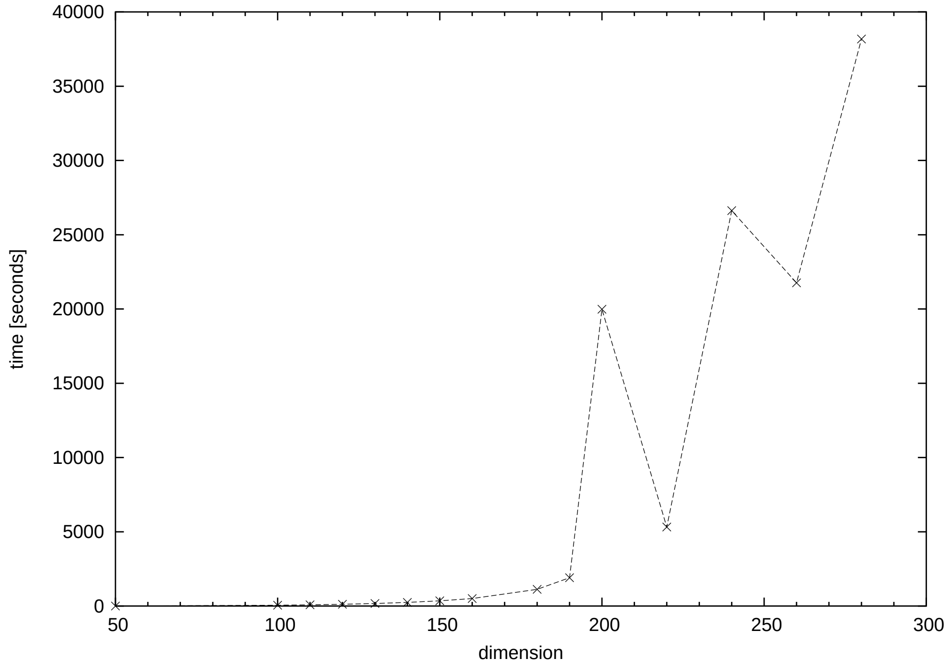

Table 1: Timings for solving CVP

dimension	RSR coefficient	BKZ block size		
		10	20	30
175	25	0.42	0.48	0.58
300	30	0.22	0.41	0.58
600	70	0.28	0.63	0.78

Table 2: Experiments for Schnorr's random sampling

F-vector space F^n. The *hamming weight* of $\mathbf{v} \in F^n$ is the number of nonzero entries in the vector $\mathbf{v}$. The *hamming distance* $\mathrm{dist}(\mathbf{v}, \mathbf{w})$ of $\mathbf{v}, \mathbf{w} \in F^n$ is the hamming weight of the difference $\mathbf{v} - \mathbf{w}$. Let $k \leq n$. An (n, k)-code over F is a k-dimensional subspace of F^n. The elements of such a code are called *code words*. For $d \in \mathbb{N}$ an (n, k, d)-code is an (n, k)-code for which d is the minimum hamming distance between two different code words.

Let $\mathcal{C}$ be an (n, k)-code for some $n, k \in \mathbb{N}$. A *generator matrix* for $\mathcal{C}$ is a matrix $C \in F^{(k,n)}$ whose rows are an F-basis of $\mathcal{C}$. We also say that C *generates* the code $\mathcal{C}$. The matrix C has rank k.

An important algorithmic problem is the following.

Problem 3 *Decoding problem. Given $n, k \in \mathbb{N}$, $k \leq n$, an (n, k)-code $\mathcal{C}$, and $\mathbf{y} \in F^n$. Find $\mathbf{x} \in \mathcal{C}$ such that* $\mathrm{dist}(\mathbf{x}, \mathbf{y})$ *is minimum.*

For $\mathbf{y} = 0$ the decoding problem is the *minimum weight problem* if $\mathbf{x} \neq 0$. Berlekamp, McEliece, and van Tilborg [12] show that the minimum weight problem is NP-complete.

Linear codes can be used for *error correction*. A message $\mathbf{m} \in F^k$ is encoded as

$$\mathbf{z} = \mathbf{m}C. \tag{4}$$

The encoded message $\mathbf{z}$ is transmitted. It is possible that during the transmission some bits of $\mathbf{z}$ are changed. The receiver receives the incorrect message $\mathbf{y}$. He solves the decoding problem, that is, he calculates $\mathbf{x} \in \mathcal{C}$ such that $\mathrm{dist}(\mathbf{x}, \mathbf{y})$ is minimum. If the error is not too big, that is, $\mathrm{dist}(\mathbf{z}, \mathbf{y}) < 1/2d$ where d is the minimum distance of any two distinct code words, then $\mathbf{x}$ is equal to the original message $\mathbf{z}$.

Linear codes are also used for encryption, for example in the McEliece cryptosystem [50] or in the Niederreiter cryptosystem [58]. To encrypt a message it is encoded and an error vector of fixed weight t is added. Decryption requires the solution of the decoding problem.

In order for error correction to be efficient, the decoding problem must be efficiently solvable. Also, coding theory based cryptosystems can only be secure if decoding is hard without the knowledge of a secret. This is both true for *binary Goppa codes*.

Decryption of a coding theory based cryptosystem means solving a decoding problem for which the weight of the error vector is known. If we have no special knowledge about the linear code such as a generating polynomial of a Goppa code, then generic methods for decoding can be used. In order to break cryptosystems based on linear codes the following problem must be solved.

Problem 4 *Crypto decoding problem. Given an* (n, k)*-code* $\mathcal{C}$*,* $n, k \in \mathbb{N}$*,* $n \geq k$*, the error weight* $t \in \mathbb{N}$ $t \leq n$*, and* $\mathbf{y} \in F^n$*. Find a vector of weight* t *in the coset* $\mathbf{y} + \mathcal{C}$*.*

Overbeck [14] has calculated Table 4 which shows the efficiency and security of the McEliece cryptosystem compared to the RSA cryptosystem. The column best attack refers to the time required by the general number field sieve attack on the RSA private key (which recovers the private key), and the attack from [15] on a McEliece ciphertext (which decrypts one cipher text block).

Note that the security comparison is made here for classical attackers. The picture changes drastically to the advantage of the McEliece system if we consider two systems to offer the same level of security if breaking them requires quantum computers with the same number of qubits.

System	Size	Work factor (binary operations)		
	public key in bytes	encryption /block size	decryption /block size	best attack
McEliece $[1024, 524, 101]$	67,072	2^9	$2^{13.25}$	2^{65}
RSA 362-bit Modulus	46	2^{17}	2^{17}	2^{68}
McEliece $[2048, 1025, 187]$	262,400	2^{10}	$2^{14.5}$	2^{107}
RSA 1024-bit Modulus	256	2^{20}	2^{20}	2^{110}
RSA 2048-bit Modulus	512	2^{22}	2^{22}	2^{145}
McEliece $[4096, 2056, 341]$	1,052,672	2^{11}	$2^{15.5}$	2^{187}
RSA 4096-bit Modulus	1024	2^{24}	2^{24}	2^{194}

Table 4: The security of RSA versus McEliece

4.4 Combinatoric group theory

Combinatoric group theory studies presentations of non-commutative groups. In this section we explain basic problems of this theory. Our exposition—like most current proposals for cryptographic schemes in this setting—focuses on braid groups.

The n-th Braid group B_n is the group of isotopy classes of diffeomorphisms of the two-dimensional disk with n points removed that keep the boundary of the disk fixed. Group operation is composition. The group is infinite.

The group B_n can be presented on generators $\sigma_1, \ldots, \sigma_{n-1}$ with relations

$$\begin{aligned} \sigma_i\sigma_j &= \sigma_j\sigma_i && \text{for } |i-j| \geq 2 \\ \sigma_i\sigma_j\sigma_i &= \sigma_j\sigma_i\sigma_j && \text{for } |i-j| = 1 \end{aligned}$$

It enjoys a nice geometric interpretation in which each n-braid is represented by a collection of n intertwined strands whose end-points are affixed to two bars.

The word problem in B_n (which asks to decide equality between two words in elements from a given generating set) is solved efficiently by using the normal form introduced in [30], or subsequent variations. In normal form each n-braid is represented by a vector from $\mathbb{Z} \times (S_n)^\star$.

Composition and inversion of elements of B_n with l components are done in time $O(ln)$. See [16]. Efficient implementation of these operations at small parameter sizes ($n \leq 250$, $l \leq 40$) are reported in the same work.

The Conjugacy Search Problem (CSP) and its variations are the starting point for the construction of one-way functions.

Problem 5 (CSP) *Given two conjugated braids $p, p' \in B_n$, find $s \in B_n$ such that $p = sp's^{-1}$.*

This problem may be modified in two ways

(a) by demanding that the conjugating element come from a certain subgroup of B_n. The resulting problem is called the Generalized Conjugacy Search Problem (GCSP).

(b) by extending the input to multiple pairs of braids that are all conjugated by the same element. The resulting problem is called the Multiple Conjugacy Search Problem (MCSP).

Alternatively, one may use the weaker Braid Diffie-Hellman Problem (BDHP). Let L and R be two commuting sub-groups of B_n.

Problem 6 (BDHP) *Given a, $b_1 = xax^{-1}$ and $b_2 = yay^{-1}$ with $a \in B_n$, $x \in L$, and $y \in R$, find the element $xy\,a\,x^{-1}y^{-1}$.*

There are several different venues for attacking the CSP.

Summit sets. The idea of the summit set method is to define a distinguished sub-set of all conjugates of a given group element which can be efficiently computed. It dates back to Garside [30], and was later refined by El-Rifai and Morton in [25], and Gebhardt [31].

Let a be a group element. The Ultra Summit Set of a defined by Gebhardt is the maximal sub-set of the set of all conjugates of a of minimal length on which the so-called cycling operator operates bijectively. The time needed to obtain an element of the Ultra Summit Set given a is quadratic in n, and linear in the length of a. It is expected that the size of the Ultra Summit Set is linear in the length of a and that it can be computed likewise in linear time. Gebhardt [31] reports about solving the CSP in B_{100} with braids of length 1000 using the Ultra Summit Set method in less than a minute computing time.

Table 5 (which is excerpted from [31]) shows how the time needed to compute the Ultra Summit Set of a random element of B_n scales with increasing braid index n and length r. In order to solve the CSP for a pair of braids it suffices to compute the Ultra Summit Sets of the two elements and record the conjugating elements occuring in the procedure. Time is given in ms on a 2.4 GHz Pentium 4 PC.

r \ n	10	20	50	100
10	4.2	4.7	12	36
100	100	100	130	210
1000	16,000	19,000	21,000	23,000

Table 5: Average time needed to compute Ultra Summit Sets

Linear representation. There are representations of B_n in the general linear groups $Gl(n(n-1)/2, \mathbb{Z}[t^{\pm1}, q^{\pm1}])$ and $Gl(n, \mathbb{Z}[t_1^{\pm1}, \ldots, t_n^{\pm1}])$ of dimension

n and $n(n-1)/2$ with coefficients coming from the ring of two-variable (or, respectively, n-variable) finite Laurent series. Using one such representation it has been shown in [17] that the Braid Diffie-Hellman Problem can be solved within time bounds polynomial in the braid index n and the length of the input braids.

Note that any faithful efficiently computable representation of B_n in a matrix group yields an efficient means of solving the Decisional Conjugacy Problem (DCP).

Problem 7 (DCP) *Given $a, b \in B_n$, decide whether there exists $x \in B_n$ such that $b = xax^{-1}$.*

The solution is achieved by comparing the characteristic polynomials of the matrices representing a and b.

Length based attacks. Individual instances of the CSP that are the result of the key generation procedures of the crypto-systems proposed in [7] and [43] can efficiently be solved by conjugating the longer of the two braids in the given pair by random braids in such a way that the complexity of the result of the conjugation is minimal. See in particular [39].

4.5 Multi-Variate Quadratic Systems

The last class of possibly quantum-hard computational problems to be considered here concerns the solution of multi-variate quadratic systems over finite fields.

Let $K = \mathbb{F}_q$ be the finite field with q elements.

Problem 8 (MQ) *Let $n \in \mathbb{N}$, $m_i \in K$ and $g_i \in K[X_1, \ldots, K]$ have degree 2. Find $x_1, \ldots, x_n \in K$ such that*

$$g_i(x_1, \ldots, x_n) = m_i, \qquad \textit{for all } 1 \leq i \leq n. \tag{5}$$

The general MQ-Problem is known to be NP-complete, see [29].

The standard method for solving multi-variate polynomial systems involves computing the Gröbner basis of the system. Run-time bounds for known Gröbner basis algorithms depend exponentially on the size of the input.

In order to introduce a trap-door for the owner of the secret key in a PKCS that allows her to solve (5) efficiently, the polynomials g_i need to be derived from an effectively solvable system. The transformation is then hidden and serves as secret key.

We describe one such derivation. Let L/K be a finite field extension of degree n, and let σ denote the corresponding Frobenius map. Let $r \in \mathbb{N}$ with $r < n$ and f be a quadratic polynomial in $L[T_0, \ldots, T_r]$. Let $m \in L$, and consider the equation

$$f(\sigma^0(x), \sigma^1(x), \ldots, \sigma^r(x)) = m \tag{6}$$

which can be efficiently solved provided a solution exists. The asymptotically fastest algorithm for solving (6) requires execution of $(d^2 + d \cdot \#L)(\log d)^{O(1)}$ operations in L where d is the degree of f in x, see [74].

Fixing a basis $(\omega_1, \ldots, \omega_n)$ of L as vector space over K and representing each element of L as K-linear combination of the ω_i we may consider (6) as a quadratic system $\mathbf{f}(x_1, \ldots, x_n) = \mathbf{m}$ over K given by polynomials $f_1, \ldots, f_n$. If s and t are now two affine linear transformations of K^n, then the system

$$\mathbf{g} = t \circ \mathbf{f} \circ s = t(\mathbf{m}) \tag{7}$$

can be efficiently solved for any vector $\mathbf{m} = (m_1, \ldots, m_n)$ for which (6) with $m = m_1\omega_1 + \cdots + m_n\omega_n$ is solvable.

A PKCS that uses the trap-door just described is called a Hidden-Field-Equations (HFE) system. In it, the vector $\mathbf{g} = (g_1, \ldots, g_n)$ is the public key, whereas the triple $(\mathbf{f}, s, t)$ serves as private key. The first HFE like systems were suggested in [49], [60] and [21].

The special form of system (7) can be used to facilitate its solution. Faugère and Joux showed in [28] and [27] that a Gröbner basis attack on the system (7) can be performed in $O(n^{10})$ operations if $q^{r-1}(q+1) \leq 512$ and $K = \mathbb{F}_2$. The reason for the efficency of this approach lies in the fact that the bound for the degree of the polynomials occuring in the Gröbner basis computation depends on the degree of the hidden function f (in x), and not on the number n of polynomials in the system.

Allan Steel[1] gives a table of timings for the solution of HFE systems with parameters n, the number of equations varying from 25 through 80, and fixed d, the degree of the hidden polynomial f. Timings are in seconds on an Athlon XP 2800+ with the exception of the last one where the computation was done on a 750MHz Sunfire v880. The software used was Magma 2.11-8.

[1] http://magma.maths.usyd.edu.au/users/allan/gb/

n	25	30	35	40	45	80
t	56.6	28.9	94.2	230.5	530.8	25.4h

Table 6: Timings for the solution of HFE systems

Instead of solving (7) directly, it is also possible to compute the secret data $(\mathbf{f}, s, t)$ from $\mathbf{g}$ by solving a large overdetermined system in the coefficients through relinearization, see [44] and [23]. It remains, however, an open question to exactly describe the run-time behavior of these algorithms.

5 Digital signatures

We describe signature schemes currently appear to be secure against quantum computer attacks.

5.1 Lattice based signatures

The basic idea of lattice based signature schemes is the following. The public key is a basis of a lattice L in $\mathbb{Z}^n$ for some $n \in \mathbb{N}$. The secret key is a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L with short vectors. Given some vector $\mathbf{z} \in \mathbb{Z}^n$, the secret key allows the computation of a lattice vector $\mathbf{v}$ that is close to $\mathbf{z}$. This can be done as follows. Write

$$\mathbf{z} = \sum_{i=1}^{n} x_i \mathbf{b}_i \tag{8}$$

and set

$$\mathbf{v} = \sum_{i=1}^{n} \lfloor x_i \rceil \mathbf{b}_i \tag{9}$$

where $\lfloor r \rceil$, $r \in \mathbb{R}$, is the nearest integer to r. Without the knowledge of the secret key the problem of computing such a lattice vector $\mathbf{v}$ is intractable. However, given the public information, anybody can verify that the lattice point $\mathbf{v}$ is close to $\mathbf{z}$. In this situation the signature scheme uses a hash function that maps a message m to a vector $\mathbf{z}$ in $\mathbb{Z}^n$. The signature of m is a lattice point that is close to $\mathbf{z}$. Only the signer can generate such a signature but everybody can verify it.

We explain two schemes that use the described principle.

The scheme of Micciancio [51] is a modification of the scheme by Goldreich, Goldwasser, and Halevi (GGH) [34]. It generates the secret and public lattice bases as follows. Fix the security parameter $n \in \mathbb{N}$. Select a random matrix in $\{-n, \ldots, n\}$. Apply lattice basis reduction to the basis that consists of the columns of that matrix. The result is the secret key B. The public key P is the Hermite normal form of B.

Ludwig [48] shows that the Micciancio cryptosystem can only be secure if $n \geq 780$. For $n = 780$ the key size in the Micciancio system is > 1MByte. The HNF computation takes several days. Ludwig also expects that the improvements in the Schnorr sampling reduction algorithm [64] imply a minimum dimension of 2000.

NTRUSign [38], [37] is a more efficient lattice based cryptosystem. One of the system parameters is a positive integer N. For example, in an unbroken instance of NTRUSign $N = 251$ is selected. The secret and public keys are polynomials with small integer coefficients of degree $< N$. They determine the NTRU-lattice L of dimension $2N$. In fact, from the secret key a basis of L with short vectors can be determined. Also, all computations in NTRU are efficient polynomial calculations. Therefore, NTRU has small keys and can extremely efficiently be implemented also in hardware. An example for key sizes is the following. If $N = 251$ and a few more parameters are selected, then the public key has bit length 2008 and both the signature and the secret key have bit length 251. NTRUSign can be found in the standard [18].

However, the security status of NTRU is unclear. Solving an approximate close vector problem is sufficient but not necessary for the security of NTRUSign. There have been many attacks on NTRU and NTRUSign which made modifications of those systems necessary (for example, see [32], [33], [55], [55], [54],). Therefore, it is unclear how secure NTRUSign really is. Nevertheless, currently NTRUSign is one of the most efficient candidate for a signature scheme secure against quantum attacks.

5.2 Code based signatures

Various coding theory based signature schemes have been suggested, for example, by Xinmei [75]. That system was attacked and modified by Harn and Wang [35] and finally broken by Alabbadi and Wicker [4]. A proposal by

Alabbadi and Wicker [5] was broken by Stern [69]

The only coding theory based signature scheme that is not broken so far is by Courtois, Finiasz, and Sendrier [20]. Their scheme is based on the Niederreiter cryptosystem [58]. We refer to that system as the CFS signature scheme. The CFS signature scheme uses a Goppa code. In the CFS signature scheme, the hash of the document to be signed augmented by a counter is hashed until the hash value is a decodable syndrome. The signer uses his secret key to determine the corresponding error-vector.

The inventors of the CFS signature scheme suggest the use of a $[2^{16} = 65536, 65392, 19]$-code. Then the size of the public key is 524 Kbytes. The expected number of hash values, that have to be inspected before a decodable syndorme is found, is $9! = 362880$. This requires approximately 2^{38} bit operations. The binary length of a signature is 145. The verification requires 2^{22} bit operations. The attack of Cantenaut and Chabaud [15] allows generating signatures without the knowledge of the public key using approximately 2^{83} bit operations.

5.3 Signatures in Braid Groups

Cryptosystems based on problems in combinatoric group theory were first suggested in [7]. A key-agreement and an encryption scheme using the braid group as the underlying group was proposed in [43]. A signature scheme followed in [45].

We explain two variants of the signature scheme proposed in [45]. Both rely on the assumption that for suitably chosen parameters the Decisional Conjugacy Problem is simple while the Conjugacy Search Problem is hard.

The signer Alice chooses a pair of conjugated braids p and $p' = sps^{-1}$. The pair (p, p') is her public key, s is the private one. It is assumed that the instance of the Conjugacy Search Problem given by the public key is hard.

Alice signs a message m as follows: First m is hashed by way of a fixed public hash function H to an element $q = H(m) \in B_n$. The signature consists of $q' = sqs^{-1}$. A verifier first validates the public key by checking that $p \sim p'$ and then checks the signature itself by testing whether $p'q' \sim pq$.

To forge a signature for a given message m in a no-message attack is to solve the Matching Conjugate Search Problem (MCSP) which asks for given $p \sim p'$ and q in B_n to produce a braid q' such that $p'q' \sim pq$. This problem is at least as hard as the Conjugacy Search Problem.

The second version of the signature scheme introduces randomization. Random braids are drawn from the finite subset of all braids of given length, random conjugates operationally from a subset of the Super Summit Set of the given element starting with a random braid of given length as conjugator. The quality of this randomization process appears to be not fully understood.

Public and private keys are identical to those of the simple variant. In order to sign document m, Alice chooses randomly $b \in B_n$ and computes successively $\alpha = b^{-1}xb$, $y = H(m\|\alpha)$, $\beta = b^{-1}yb$, and $\gamma = b^{-1}aya^{-1}b$. The signature is (α, β, γ).

The signature is valid if $\alpha \sim x$, $\beta \sim \gamma y$, and $\alpha\gamma \sim x'y$.

For a further modified version of the randomized scheme, Ko *et.al.* claim security against adaptive-chosen message attacks under the assumption of the hardness of the Matching Triple Search Problem which is an obvious generalization of the MCSP.

Due to the progress in solving the CSP at currently proposed parameter sizes it has been proposed to substitute the braid groups in the above schemes with other non-commutative groups for which the word problem is likewise efficiently solvable.

5.4 Signatures from Multi-Variate Quadratic Systems

There exists a large number of varieties of the HFE signature scheme proposed in [60].

The general procedure of these signature schemes is as follows. A finite base field K and some $n \in \mathbb{N}$ is chosen. Messages concatenated with some random data are hashed to vectors in K^n. The signer then solves the multivariate system (5) using his trap-door decomposition of $\mathbf{g} = (g_1, \ldots, g_n)$. If this succeeds the solution serves as signature. If it does not, then the message is concatenated with a new random string, and the procedure repeated until it is met with success.

The variations of HFE concern the choice of the ground field, the form of the hidden polynomial f, and the form in which the public key is presented. In this public key some equations might be removed, others added, and some variables fixed and eliminated *a priori*.

A notable example is the signature scheme SFLASH [22] which was included in the final portfolio of the NESSIE project[2] of the European Union

[2]The NESSIE project provided recommendations regarding the use of cryptographic

for use in low security resource constrained environments.

SFLASH uses a version of the HCE system called $C^\star$ where $K = \mathbb{F}_{2^7}$, the univariate trap-door polynomial f is actually a monomial and 11 polynomials g_i are deleted from the public key. Randomization is unnecessary here since f as a monomial map is surjective.

A slightly modified version of SFLASH [57] is reported to produce signatures and allow verification in less than 1ms on standard PC hardware. Key generation takes approximately the same amount of time. Public key size is 112.3 KBytes and signatures are 469 bits long.

5.5 Merkle Signatures

In his thesis, Merkle invented a signature scheme whose security is based on the collision freeness of an arbitrary cryptographic hash function and an arbitrary one-time signature algorithm. The idea of the Merkle signature scheme is the following:

In the one-time algorithm that is used by Merkle's scheme a secret signature algorithm sign is applied to the message m. The result is the signature

$$\tau = \text{sign}(m).$$

The signature can be verified by applying the public verification function

$$\texttt{verify}(\tau, m) = \begin{cases} \texttt{true} \\ \texttt{false} \end{cases}$$

The secret key is sign. The public key is `verify`.

By themselves, one-time signature schemes cannot be used in an open network. Any user must be able to obtain any public verification function and he must be able to convince himself of the correctness of that function. For example if digital signatures are used to authenticate an Internet user, that user must use a new key pair for each authentication and there is no way for the authentication server to know all the public keys.

Merkle's scheme makes the use of one-time signatures in an open network easier.

The *key generation* works as follows. A signer fixes a maximum number $N = 2^n$ of signatures that he wants to produce and selects a collision free

---

schemes for signature, integrity and encryption in the member states of the EU.

hash function h. He generates N pairs (X_i, Y_i), $0 \leq i < N$ of secret and public keys. From the keys Y_i, $0 \leq i < N$ the signer constructs a public key for the whole system. This is done via a binary tree. The leaves of that tree are

$$W_{i,n} = Y_i, \quad 0 \leq i < N.$$

The nodes $W_{i,j}$ on level j, $0 \leq j < n$, $0 \leq i < N$ are defined recursively as

$$W_{i,j} = h(W_{2i-1,j+1} \circ W_{2i,j}), \quad 0 \leq j < n, 0 \leq i < N \tag{10}$$

where $\circ$ denotes concatenation of bitstrings. The public key is $(W_{0,0}, n)$. So it contains only the root of the tree and its depth instead of containing many thousands of public one-time keys. Also, there is a systematic method for generating the secret keys from a seed. So the Merkle tree can be computed as necessary.

Suppose that the signer has already signed $i - 1$ messages with $i \leq N$. To *sign* a new message m the signer computes

$$\tau = \mathrm{sign}_{X_i}(m).$$

The signer also constructs a sequence $P_0, \ldots, P_n$ of nodes that helps the verifier to convince himself of the correctness of Y_i by constructing the path in the tree from Y_k to the root $W_{0,0}$. The signer sets

$$P_0 = W_{0,0}. \tag{11}$$

For $j \in \{1, \ldots, n\}$ the signer calculates

$$k = \left\lfloor \frac{i}{2^j} \right\rfloor \tag{12}$$

and

$$P_j = \begin{cases} W_{k+1,j} & \text{if } k \text{ is even} \\ W_{k-1,j} & \text{if } k \text{ is odd.} \end{cases} \tag{13}$$

The signature is

$$s = (\tau, Y_i, P_1, \ldots, P_n). \tag{14}$$

The signature is longer than the one time signature. Here, the algorithm pays for saving public key space. However, for 2^n possible signatures the signature length is only n.

The *verification* consists of two steps. First, the verifier calculates $\texttt{verify}(\tau, m)$. If $\texttt{verify}(\tau, m) = \texttt{false}$, then the verifier rejects the signature. If $\texttt{verify}(\tau, m) = \texttt{true}$, then the validity of the key Y_i is reduced to the validity of the root of the tree. The verifier constructs the path from Y_i to the root $W_{0,0}$ in the Merkle graph. The verifier calculates the path $(Q_j)_{0 \leq n}$ in the Merkle graph as follows. He sets

$$Q_n = Y_i \tag{15}$$

For $j \in \{0, \ldots, n-1\}$ the signer calculates

$$k = \left\lfloor \frac{i}{2^{j+1}} \right\rfloor \tag{16}$$

and

$$Q_j = \begin{cases} h(Q_{j+1}, P_{j+1}) & \text{if } k \text{ is even} \\ h(P_{j+1}, Q_{j+1}) & \text{if } k \text{ is odd.} \end{cases} \tag{17}$$

If $Q_0 = W_{0,0}$, then the signature is accepted. Otherwise, it is rejected.

It has been shown that the Merkle signature scheme is secure against chosen message attacks provided that the hash function used is collision resistant.

$\mathcal{N}$	Key Generation	Signing	Verification
	time in sec	time in ms	
10	2.45	22.8	0.88
11	4.95	25.2	0.94
12	9.31	27.4	1.00
13	18.61	29.7	1.07
14	37.32	31.9	1.13
15	74.40	35.3	1.20
16	149.40	38.0	1.27

Table 8: Merkle Signature Scheme

Coronado [19] has implemented the Merkle scheme. Table 8 on page 21 shows the efficiency of the Merkle key generation, the Merkle signature generation, and the Merkle signature verification. That table shows that the efficiency of the Merkle signature scheme is competitive. However, it has to be studied how the Merkle signature scheme with its limited number of possible signatures can be used in today application scenarios.

The timing was measured on a SunBlade-100 SunOS 5.8 Generic 450 MHz. and the hash function used is SHA-1.

6 Other suggestions

Okamoto et al. [72] suggest public-key systems that use quantum computers in the key generation process.

7 Conclusion

The threat of quantum computer attacks and the relevance of digital signature schemes for information technology security makes the development of post-quantum signature schemes necessary. Presently, complexity theory does not help to find intractable computational problems that can be used as the security basis for post-quantum signature schemes. However, there are several candidates for such computational problems. Also, there are several candidates for post-quantum signature schemes. The most efficient ones are NTRU, SFLASH and the Merkle scheme.

But many research problems have still to be solved. We identify a few.

Is it true that the computational problems from Section 4 remain intractable in the age of quantum computers? What is a good security notion for post-quantum signatures? How can quantum-hard instances be efficiently generated? Can we say more about the security of NTRU and SFLAH? Are there other efficient lattice based signature schemes? Is there an efficient coding theory based signature scheme? Can the Merkle- signature scheme be used in todays signature applications?

References

[1] M. Ajtai, *The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract)*, Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ACM Press, 1998, pp. 10–19.

[2] ———, *The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice*, in STOC'03 [70], pp. 396–406.

[3] M. Ajtai and C. Dwork, *A public-key cryptosystem with worstcase / average-case equivalence*, Proceedings of the 29th Annual Symposium on Theory of Computing (STOC), ACM Press, 1997, pp. 284–293.

[4] M. Alabbadi and S.B. Wicker, *Security of Xinmei digital signature scheme*, Electronics Letters **29(9)** (1992), 890–891.

[5] ———, *A digital signature scheme based on linear error-correcting block codes*, ASIACRYPT '94, vol. LNCS 917, Springer 1995, pp. 238–248.

[6] *Alphatrust*, http://www.alphatrust.com/projects/default.asp.

[7] I. Anshel, M. Anshel, and D. Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters **6** (1999), 1–5.

[8] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, Journal of Computer and System Sciences **54** (1997), no. 2, 317–331.

[9] ARX, http://www.arx.com/documents/products/Insurance_brochure.pdf.

[10] http://www.banxico.org.mx/dDisposiciones/Disposiciones2019/11-2004.html.

[11] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp, *Authentication of quantum messages*, 43rd Annual IEEE Symposium on Foundations of Computer Science FOCS, 2002, pp. 449– 458.

[12] E. Berlekamp, R. McEliece, and H. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Transactions on Information Theory **24(3)** (1978), 384–386.

[13] G. Brassard, *A note on the complexity of cryptography*, IEEE Transactions on Information Theory **25** (1979), 232–233.

[14] J. Buchmann, D. Engelbert, R. Overbeck, and A. Schmidt, *Coding theory based public key cryptography*, in preparation.

[15] A. Canteaut and N. Sendrier, *Cryptanalysis of the original McEliece cryptosystem*, Advances in Cryptology - ASIACRYPT '98 Proceedings, Springer-Verlag, 1998, pp. 187–199.

[16] Jae Choon Cha, Ki Hyoung Ko, Sang Jin Lee, Jae Woo Han, and Jung Hee Cheon, *An efficient implementation of braid groups*, Advances in Cryptology – ASIACRYPT 2001 (Colin Boyd, ed.), Lecture Notes in Computer Science, vol. 2248, Springer-Verlag, 2001, pp. 144–156.

[17] Jung Hee Cheon and Byungheup Jun, *A polynomial time algorithm for the braid diffie-hellman conjugacy problem*, http://eprint.iacr.org/2003/019/, 2003.

[18] Consortium for Efficient Embedded Security, *EESS#1: Implementation aspects of NTRUEncrypt and NTRUSign (version 2.0)*, June 2003, http://www.ceesstandards.org/.

[19] C. Coronado, *On the security and the efficiency of the merkle signature scheme*, in preparation.

[20] N. Courtois, M. Finiasz, and N.Sendrier, *How to achieve a McEliece-based digital signature scheme*, Advances in Cryptology - ASIACRYPT 2001, vol. 2248, Springer-Verlag, 2001, pp. 157–174.

[21] N. Courtois, L. Goubin, and J. Patarin, *C*-+ and HM- Variations around two schemes of T. Matsumoto and H. Imai*, ASIACRYPT (Kazuo Ohta and Dingyi Pei, eds.), Lecture Notes in Computer Science, vol. 1514, Springer, 1998, pp. pp. 35–49.

[22] ———, *SFLASH, a fast asymmetric signature scheme for low-cost smartcards. Primitive specification and supporting documentation*, submitted to the NESSIE project. see also http://www.minrank.org/sflash-b-v2.pdf, October 2001.

[23] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, EUROCRYPT (Bart Preneel, ed.), Lecture Notes in Computer Science, vol. 1807, Springer, 2000, pp. pp. 392–407.

[24] *Digital signature legislation*, http://rechten.uvt.nl/simone/ds-lawsu.htm.

[25] E. A. El-Rifai and H. R. Morton, *Algorithms for positive braids*, Quarterly Jounal of Mathematics Oxford Series (2) **45** (1994), no. 2, 479–497.

[26] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Tech. Report 81-04, University of Amsterdam, Department of Mathematics, Netherlands, 1981.

[27] J. C. Faugère, *Algebraic cryptanalysis of HFE using Gröbner bases*, Rapport de recherche de l'INRIA-Lorraine, Equipe: SPACES 4738, INRIA, February 2003, http://www.inria.fr/rrrt/rr-4738.html.

[28] J. C. Faugère and A. Joux, *Algebraic Cryptanalysis of Hidden Field Cryptosystems Using Gröbner Bases*, CRYPTO (Dan Boneh, ed.), Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. pp. 44–60.

[29] M. R. Garey and D. S. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*, ch. Appendix A7.2, W. H. Freeman and Company, 1979.

[30] F. A. Garside, *The braid group and other groups*, Quarterly Journal of Mathematics Oxford Series (2) **20** (1969), 235–254.

[31] V. Gebhardt, *A new approach to the conjuagacy problem in garside groups*, http://arxiv.org/abs/math.GT/0306199, 2003.

[32] C. Gentry, J. Jonsson, J. Stern, and Michael S., *Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001*, Advances in Cryptology – Asiacrypt 2001 (C. Boyd, ed.), LNCS, vol. 2248, Springer-Verlag, 2001, pp. 1–20.

[33] C. Gentry and M. Szydlo, *Cryptanalysis of the revised NTRU signature scheme*, Advances in Cryptology – Eurocrypt 2002 (L. Knudsen, ed.), Lecture Notes in Copmuter Science, vol. 2332, Springer-Verlag, 2002, pp. 299–320.

[34] O. Goldreich, S. Goldwasser, and S. Halevi, *Public-key cryptosystems from lattice reduction problems*, Advances in Cryptology – Crypto'97 (Burton S. Kaliski, Jr., ed.), LNCS, vol. 1294, Springer-Verlag, 1997, pp. 112–131.

[35] L. Harn and D.-C. Wang, *Cryptanalysis and modification of digital signature scheme based on error-correcting codes*, Electronics Letters **28(2)** (1992), 157–159.

[36] *History of quantum computing*, http://www.wordiq.com/definition/Timeline_of_quantum_computing.

[37] J. Hoffstein, J. Pipher, and J. Silverman, *The NTRU signature scheme: Theory and practise*, http://www.ntru.com/cryptolab/archive_center.htm.

[38] J. Hoffstein, J. Pipher, and J. H. Silverman, *NSS: An NTRU lattice-based signature scheme*, Advances in Cryptology – Eurocrypt 2001 (B. Pfitzmann, ed.), LNCS, vol. 2045, Springer-Verlag, 2001, pp. 211–228.

[39] D. Hofheinz and R. Steinwandt, *A practical attack on some braid group based cryptographic primitives*, Practice and Theory in Public Key Cryptography, PKC 2003 (Y.G. Desmedt, ed.), Lecture Notes in Computer Science, vol. 2567, Springer-Verlag, 2003, pp. 187–198.

[40] http://bmj.bmjjournals.com/cgi/content/full/329/7458/131.

[41] *IEEE P1363 Draft standard specifications for public key cryptography*, http://grouper.ieee.org/groups/1363/P1363/.

[42] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Research **12** (1987), 415–440.

[43] K.H.Ko, S.J.Lee, J.H.Chean, J.W.Han, J.S.Kang, and C.Park, *New public-key cryptosystem using braid groups*, Advances in Cryptology – CRYPTO 2000 (Mihir Bellare, ed.), Lecture Notes in Computer Science, vol. 1880, Springer-Verlag, 2000, pp. 166–183.

[44] A. Kipnis and A. Shamir, *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization*, CRYPTO (Michael J. Wiener, ed.), Lecture Notes in Computer Science, vol. 1666, Springer, 1999, pp. pp. 19–30.

[45] Ki Hyoung Ko, Doo Ho Choi, Mi Sung Cho, and Jang Won Lee, *New signature scheme using conjugacy problem*, http://eprint.iacr.org/2002/168/, 2002.

[46] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515–534.

[47] C. Ludwig, *A faster lattice reduction method using quantum search*, ISAAC 2003: 14th International Symposium on Algorithms and Computations (T. Ibaraki, N. Katoh, and H. Ono, eds.), LNCS, vol. 2906, Springer, 2003, pp. 199–208.

[48] C. Ludwig, *The security and efficiency of micciancio's cryptosystem*, Cryptology ePrint Archive, Report 2004/209, 2004, http://eprint.iacr.org/.

[49] T. Matsumoto and H. Imai, *Public Qudaratic Polynomial-Typles For Efficient Signature-Verification and Message-Encryption*, EUROCRYPT (C. G. Günther, ed.), Lecture Notes in Computer Science, vol. 330, Springer, 1988, pp. pp. 419–453.

[50] R.J. McEliece, *A public key cryptosystem based on algebraic coding theory*, DSN progress report **42-44** (1978), 114–116.

[51] D. Micciancio, *Improving lattice based cryptosystems using the Hermite normal form*, Cryptography and Lattices (Providence, RI, USA) (Joseph H. Silverman, ed.), LNCS, vol. 2146, Springer-Verlag, 2001, pp. 126–145.

[52] ______, *The shortest vector problem is NP-hard to approximate to within some constant*, SIAM Journal on Computing **30** (2001), no. 6, 2008–2035.

[53] D. Micciancio and S. Goldwasser, *Complexity of lattice problems*, Kluwer Academic Publishers, 2002.

[54] S. Min, G. Yamamoto, and K. Kim, *On the security of the NTRUSign signature scheme*, Proc. of the 2004 Symposium on Cryptography and Information Security, 2004.

[55] S. Min, G. Yamamoto, and K. Kim, *Weak property of malleability in ntrusign*, ACISP04, 2004.

[56] H. Minkowski, *Geometrie der Zahlen*, Teubner, Leipzig, 1896.

[57] N. Courtois and L. Goubin and J. Patarin, *SFLASHv3, a fast asymmetric signature scheme*, Cryptology ePrint Archive, Report 2003/211, 2003, http://eprint.iacr.org/2003/211.

[58] H. Niederreiter, *Knapsack-type cryptosystems and algebraic coding theory*, Problems of Control and Information Theory **15** (1986), no. 2, 159–166.

[59] M.A. Nielsen and I.L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[60] J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88*, CRYPTO (Don Coppersmith, ed.), Lecture Notes in Computer Science, vol. 963, Springer, 1995, pp. pp. 248–261.

[61] *Quantum computing projects*, http://www.vcpc.univie.ac.at/~ian/hotlist/qc/projects.shtml.

[62] Oded Regev, *New lattice based cryptographic constructions*, in STOC'03 [70], pp. 407–416.

[63] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, **21** (1978), 120–126.

[64] C. P. Schnorr, *Lattice reduction by random sampling and birthday methods*, STACS 2003: 20th Annual Symposium on Theoretical Aspects of Computer Science (H. Alt and M. Habib, eds.), LNCS, vol. 2607, Springer, 2003, pp. 146–156.

[65] ______, *Fast LLL-type lattice reduction*, 2004, http://www.mi.informatik.uni-frankfurt.de/research/papers/StSeg2.ps.

[66] C. P. Schnorr and M. Euchner, *Lattice basis reduction: Improved practical algorithms and solving subset sum problems*, Math. Programming **66** (1994), 181–199.

[67] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Computing **26** (1997), 1484–1509.

[68] D.R. Simon, *On the power of quantum computation*, SIAM J. Computing **26** (1997), no. 5, 1474–1483.

[69] J. Stern, *Can one design a signature scheme based on error-correcting codes*, ASIACRYPT '94, LNCS, vol. 917, 1995, pp. 424–426.

[70] *Proceedings of the thirty-fifth annual acm symposium on theory of computing*, ACM Press, 2003.

[71] https://www.trustedcomputinggroup.org/home.

[72] K. Tanaka und S.Uchiyama T.Okamoto, *Quantum public key cryptosystems*, Proc. Of CRYPTO 2000, LNCS **1880** (2000), 147–165, Springer-Verlag.

[73] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, *Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature **414** (2001), 883–887.

[74] Joachim von zur Gathen and Victor Shoup, *Computing frobenius maps and factoring polynomials*, Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, ACM Press, 1992, pp. 97–105.

[75] W. Xinmei, *Digital signature scheme based on error-correcting codes*, Electronics Letters **26(13)** (1990), 898–899.