

The Impact of Quantum Computing on Present Cryptography

Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang
Department of Informatics, University of Oslo, Norway

Abstract—The aim of this paper is to elucidate the implications of quantum computing in present cryptography and to introduce the reader to basic post-quantum algorithms. In particular the reader can delve into the following subjects: present cryptographic schemes (symmetric and asymmetric), differences between quantum and classical computing, challenges in quantum computing, quantum algorithms (Shor's and Grover's), public key encryption schemes affected, symmetric schemes affected, the impact on hash functions, and post quantum cryptography. Specifically, the section of Post-Quantum Cryptography deals with different quantum key distribution methods and mathematical-based solutions, such as the BB84 protocol, lattice-based cryptography, multivariate-based cryptography, hash-based signatures and code-based cryptography.

Keywords—Quantum computers; post-quantum cryptography; Shor's algorithm; Grover's algorithm; asymmetric cryptography; symmetric cryptography

I. INTRODUCTION

There is no doubt that advancements in technology and particularly electronic communications have become one of the main technological pillars of the modern age. The need for confidentiality, integrity, authenticity, and non-repudiation in data transmission and data storage makes the science of cryptography one of the most important disciplines in information technology. Cryptography, etymologically derived from the Greek words hidden and writing, is the process of securing data in transit or stored by third party adversaries. There are two kinds of cryptosystems; symmetric and asymmetric.

Quantum computing theory firstly introduced as a concept in 1982 by Richard Feynman, has been researched extensively and is considered the destructor of the present modern asymmetric cryptography. In addition, it is a fact that symmetric cryptography can also be affected by specific quantum algorithms; however, its security can be increased with the use of larger key spaces. Furthermore, algorithms that can break the present asymmetric cryptoschemes whose security is based on the difficulty of factorizing large prime numbers and the discrete logarithm problem have been introduced. It appears that even elliptic curve cryptography which is considered presently the most secure and efficient scheme is weak against quantum computers. Consequently, a need for cryptographic algorithms robust to quantum computations arose.

The rest of the paper deals initially with the analysis of symmetric cryptography, asymmetric cryptography and hash functions. Specifically, an emphasis is given on algorithms that take advantage of the difficulty to factorize large prime numbers, as well as the discrete logarithm problem. We move on by giving an introduction to quantum mechanics and the challenge of building a true quantum computer. Furthermore,

we introduce two important quantum algorithms that can have a huge impact in asymmetric cryptography and less in symmetric, namely, Shor's algorithm and Grover's algorithm respectively. Finally, post-quantum cryptography is presented. Particularly, an emphasis is given on the analysis of quantum key distribution and some mathematical based solutions such as lattice-based cryptography, multivariate-based cryptography, hash-based signatures, and code-based cryptography.

II. PRESENT CRYPTOGRAPHY

In this chapter we explain briefly the role of symmetric algorithms, asymmetric algorithms and hash functions in modern cryptography. We analyze the difficulty of factorizing large numbers, as well as the discrete logarithm problem which is the basis of strong asymmetric ciphers.

A. Symmetric Cryptography

In symmetric cryptography, the sender and the receiver use the same secret key and the same cryptographic algorithm to encrypt and decrypt data. For example, Alice can encrypt a plaintext message using her shared secret key and Bob can decrypt the message using the same cryptographic algorithm Alice used and the same shared secret key. The key needs to be kept secret, meaning that only Alice and Bob should know it; therefore, an efficient way for exchanging secret keys over public networks is demanded. Asymmetric cryptography was introduced to solve the problem of key distribution in symmetric cryptography. Popular symmetric algorithms include the advanced encryption standard (AES) and the data encryption standard (3DES).

B. Asymmetric Cryptography

Asymmetric cryptography or public key cryptography (PKC) is a form of encryption where the keys come in pairs. Each party should have its own private and public key. For instance, if Bob wants to encrypt a message, Alice would send her public key to Bob and then Bob can encrypt the message with Alice's public key. Next, Bob would transmit the encrypted message to Alice who is able to decrypt the message with her private key. Thus, we encrypt the message with a public key and only the person who owns the private key can decrypt the message.

Asymmetric cryptography additionally is used for digital signatures. For example, Alice can sign a document digitally with her private key and Bob can verify the signature with Alice's known public key. The security of PKC rests on computational problems such as the difficulty of factorizing large prime numbers and the discrete logarithm problem. Such

kind of algorithms are called one-way functions because they are easy to compute in one direction but the inversion is difficult [1].

1) *Factorization Problem - RSA Cryptosystem*: One of the most important public-key schemes is RSA invented by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1977. RSA exploits the difficulty of factorizing bi-prime numbers. According to Paar and Pelzl [2], RSA and in general asymmetric algorithms are not meant to replace symmetric algorithms because they are computationally costly. RSA is mainly used for secure key exchange between end nodes and often used together with symmetric algorithms such as AES, where the symmetric algorithm does the actual data encryption and decryption. Kirsch [3] stated that RSA is theoretically vulnerable if a fast factorizing algorithm is introduced or huge increase in computation power can exist. The latter can be achieved with the use of quantum mechanics on computers, known as quantum-computers.

2) *Discrete Logarithm Problem (DLP)*: Asymmetric cryptographic systems such as Diffie-Hellman (DH) and Elliptic Curve Cryptography (ECC) are based on DLP. The difficulty of breaking these cryptosystems is based on the difficulty in determining the integer r such that $g^r = x \pmod p$. The integer r is called the discrete logarithm problem of x to the base g , and we can write it as $r = \log_g x \pmod p$. The discrete logarithm problem is a very hard problem to compute if the parameters are large enough.

Diffie-Hellman is an asymmetric cipher that uses the aforementioned property to transmit keys securely over a public network. Recently, keys larger or equal to 2048 bits are recommended for secure key exchange. In addition, another family of public key algorithms known as Elliptic Curve Cryptography is extensively used. ECC provides the same level of security as RSA and DLP systems with shorter key operands which makes it convenient to be used by systems of low computational resources. ECC uses a pair (x, y) that fits into the equation $y^2 = x^3 + ax + b \pmod p$ together with an imaginary point Θ (theta) at infinity, where $a, b \in \mathbb{Z}_p$ and $4a^3 + 27b^2 \neq 0 \pmod p$ [2]. ECC needs a cyclic Group G and the primitive elements we use, or pair elements, to be of order G . ECC is considered the most secure and efficient asymmetric cryptosystem, but this tends to change with the introduction of quantum computers as it is explained in the next sections.

III. QUANTUM COMPUTING VS CLASSICAL COMPUTING

In 1982, Richard Feynman came up with the idea of *quantum computer*, a computer that uses the effects of quantum mechanics to its advantage. Quantum mechanics is related to microscopic physical phenomena and their strange behavior. In a traditional computer the fundamental blocks are called bits and can be observed only in two states; 0 and 1. Quantum computers instead use quantum bits also usually referred as *qubits* [4]. In a sense, qubits are particles that can exist not only in the 0 and 1 state but in both simultaneously, known as superposition. A particle collapses into one of these states when it is inspected. Quantum computers take advantage of this property mentioned to solve complex problems. An operation on a qubit in superposition acts on both values at the same time. Another physical phenomenon used in quantum computing is quantum

entanglement. When two qubits are entangled their quantum state can no longer be described independently of each other, but as a single object with four different states. In addition, if one of the two qubits state change the entangled qubit will change too regardless of the distance between them. This leads to true parallel processing power [5]. The combination of the aforementioned phenomena result in exponential increase in the number of values that can be processed in one operation, when the number of entanglement qubits increase. Therefore, a n -qubit quantum computer can process 2^n operations in parallel.

Two kinds of quantum computers exists; universal and non-universal. The main difference between the two is that universal quantum computers are developed to perform any given task, whereas non-universal quantum computers are developed for a given purpose (e.g., optimization of machine learning algorithms). Examples are, D-Wave's 2000+ qubits non-universal quantum computer [6] and IBM's 17 qubits universal quantum computer with proper error correction. IBM's quantum computer is currently the state of the art of universal quantum computers [7]. Both D-Wave and IBM have quantum computers accessible online for research purposes. Additionally, in October 2017, Intel in collaboration with QuTech announced their 17-qubits universal quantum computer [7].

Bone and Castro [8] stated that a quantum computer is completely different in design than a classical computer that uses the traditional transistors and diodes. Researchers have experimented with many different designs such as quantum dots which are basically electrons being in a superposition state, and computing liquids. Besides, they remarked that quantum computers can show their superiority over the classical computers only when used with algorithms that exploit the power of quantum parallelism. For example, a quantum computer would not be any faster than a traditional computer in multiplication.

A. Challenges in Quantum Computing

There are many challenges in quantum computing that many researchers are working on.

- Quantum algorithms are mainly probabilistic. This means that in one operation a quantum computer returns many solutions where only one is the correct. This trial and error for measuring and verifying the correct answer weakens the advantage of quantum computing speed [3].
- Qubits are susceptible to errors. They can be affected by heat, noise in the environment, as well as stray electromagnetic couplings. Classical computers are susceptible to bit-flips (a zero can become one and vice versa). Qubits suffer from bit-flips as well as phase errors. Direct inspection for errors should be avoided as it will cause the value to collapse, leaving its superposition state.
- Another challenge is the difficulty of coherence. Qubits can retain their quantum state for a short period of time. Researchers at the University of New South Wales in Australia have created two different types

of qubits (Phosphorous atom and an Artificial atom) and by putting them into a tiny silicon (*silicon 28*) they were able to eliminate the magnetic noise that makes them prone to errors. Additionally, they stated that the Phosphorous atom has 99.99% accuracy which accounts for 1 error every 10,000 quantum operations [9]. Their qubits can remain in superposition for a total of 35 seconds which is considered a world record [10]. Moreover, to achieve long coherence qubits need not only to be isolated from the external world but to be kept in temperatures reaching the absolute zero. However, this isolation makes it difficult to control them without contributing additional noise [3].

IBM in 2017, introduced the definition of *Quantum Volume*. Quantum volume is a metric to measure how powerful a quantum computer is based on how many qubits it has, how good is the error correction on these qubits, and the number of operations that can be done in parallel. Increase in the number of qubit does not improve a quantum computer if the error rate is high. However, improving the error rate would result in a more powerful quantum computer [11].

IV. CRYPTOSYSTEMS VULNERABLE TO QUANTUM ALGORITHMS

This section discusses the impact of quantum algorithms on present cryptography and gives an introduction to Shor's algorithm and Grover's algorithm. Note that Shor's algorithm explained in the following subsection makes the algorithms that rely on the difficulty of factorizing or computing discrete logarithms vulnerable.

Cryptography plays an important role in every electronic communication system today. For example the security of emails, passwords, financial transactions, or even electronic voting systems require the same security objectives such as confidentiality and integrity [12]. Cryptography makes sure that only parties that have exchanged keys can read the encrypted message (also called authentic parties). Quantum computers threaten the main goal of every secure and authentic communication because they are able to do computations that classical (conventional) computers cannot. Consequently, quantum computers can break the cryptographic keys quickly by calculating or searching exhaustively all secret keys, allowing an eavesdropper to intercept the communication channel between authentic parties (sender/receiver). This task is considered to be computational infeasible by a conventional computer [13].

According to NIST, quantum computers will bring the end of the current public key encryption schemes [14]. Table I adapted from NIST shows the impact of quantum computing on present cryptographic schemes.

A. Shor's Algorithm in Asymmetric Cryptography

In 1994, the mathematician Peter Shor in his paper "Algorithms for Quantum Computation: Discrete Logarithms and Factoring" [15], proved that factorizing large integers would change fundamentally with a quantum computer.

Shor's algorithm can make modern asymmetric cryptography collapse since it is based on large prime integer factorization or the discrete logarithm problem. To understand

how Shor's algorithm factorizes large prime numbers we use the following example. We want to find the prime factors of number 15. To do so, we need a 4-qubit register. We can visualize a 4-qubit register as a normal 4-bit register of a traditional computer. Number 15 in binary is 1111, so a 4-qubit register is enough to accommodate (calculate) the prime factorization of this number. According to Bone and Castro [8], a calculation performed on the register can be thought as computations done in parallel for every possible value that the register can take (0-15). This is also the only step needed to be performed on a quantum computer.

The algorithm does the following:

- $n = 15$, is the number we want to factorize
- $x =$ random number such as $1 < x < n - 1$
- x is raised to the power contained in the register (every possible state) and then divided by n
The remainder from this operation is stored in a second 4-qubit register. The second register now contains the superposition results. Let's assume that $x = 2$ which is larger than 1 and smaller than 14.
- If we raise x to the powers of the 4-qubit register which is a maximum of 15 and divide by 15, the remainders are shown in Table II.

What we observe in the results is a repeating sequence of 4 numbers (1,2,4,8). We can confidently say then that $f = 4$ which is the sequence when $x = 2$ and $n = 15$. The value f can be used to calculate a possible factor with the following equation:

$$\text{Possible factor: } P = x^{f/2} - 1$$

In case we get a result which is not a prime number we repeat the calculation with different f values.

Shor's algorithm can be used additionally for computing discrete logarithm problems. Vazirani [16] explored in detail the methodology of Shor's algorithm and showed that by starting from a random superposition state of two integers, and by performing a series of Fourier transformations, a new superposition can be set-up to give us with high probability two integers that satisfy an equation. By using this equation we can calculate the value r which is the unknown "exponent" in the DLP.

B. Grover's algorithm in Symmetric Cryptography

Lov Grover created an algorithm that uses quantum computers to search unsorted databases [17]. The algorithm can find a specific entry in an unsorted database of N entries in \sqrt{N} searches. In comparison, a conventional computer would need $N/2$ searches to find the same entry. Bone and Castro [8] remarked the impact of a possible application of Grover's algorithm to crack Data Encryption Standard (DES), which relies its security on a 56-bit key. The authors remarked that the algorithm needs only 185 searches to find the key.

Currently, to prevent password cracking we increase the number of key bits (larger key space); as a result, the number of searches needed to crack a password increases exponentially. Buchmann et al. [18] stated that Grover's algorithm have some applications to symmetric cryptosystems but it is not as fast as Shor's algorithm.

TABLE I. IMPACT ANALYSIS OF QUANTUM COMPUTING ON ENCRYPTION SCHEMES (ADAPTED FROM [14])

Cryptographic Algorithm	Type	Purpose	Impact From Quantum Computer
AES-256	Symmetric key	Encryption	Secure
SHA-256, SHA-3	–	Hash functions	Secure
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

TABLE II. 4-QUBIT REGISTERS WITH REMAINDERS

Register 1:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Register 2:	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8

C. Asymmetric Encryption Schemes Affected

All public key algorithms used today are based on two mathematical problems, the aforementioned factorization of large numbers (e.g., RSA) and the calculation of discrete logarithms (e.g., DSA signatures and ElGamal encryption). Both have similar mathematical structure and can be broken with Shor’s algorithm rapidly. Recent algorithms based on elliptic curves (such as ECDSA) use a modification of the discrete logarithm problem that makes them equally weak against quantum computers. Kirsch and Chow [3] mentioned that a modified Shor’s algorithm can be used to decrypt data encrypted with ECC. In addition, they emphasized that the relatively small key space of ECC compared to RSA makes it easier to be broken by quantum computers. Furthermore, Proos and Zalka [19] explained that 160-bit elliptic curves could be broken by a 1000-qubit quantum computer, while factorizing 1024-bit RSA would require a 2000-qubit quantum computer. The number of qubits needed to break a cryptosystem is relative to the algorithm proposed. In addition, they show in some detail how to use Shor’s algorithm to break ECC over GF(p).

On the other hand, Grover’s algorithm is a threat only to some symmetric cryptographic schemes. NIST [14] points out that if the key sizes are sufficient, symmetric cryptographic schemes (specifically the Advanced Encryption Standard-AES) are resistant to quantum computers. Another aspect to be taken into consideration is the robustness of algorithms against quantum computing attacks also known as quantum cryptanalysis.

In Table III, a comparison of classical and quantum security levels for the most used cryptographic schemes is presented.

D. Symmetric Encryption Schemes Affected

For symmetric cryptography quantum computing is considered a minor threat. The only known threat is Grover’s algorithm that offers a square root speed-up over classical brute force algorithms. For example, for a n-bit cipher the quantum computer operates on $(\sqrt{2^n} = 2^{n/2})$. In practice, this means that a symmetric cipher with a key length of 128-bit (e.g., AES-128) would provide a security level of 64-bit. We recall here that security level of 80-bit is considered secure. The Advanced Encryption Standard (AES) is considered to be one of the cryptographic primitives that is resilient in quantum computations, but only when is used with key sizes of 192 or 256 bits. Another indicator of the security of AES in the post-quantum era is that NSA (The National Security Agency)

allows AES cipher to secure (protect) classified information for security levels, SECRET and TOP SECRET, but only with key sizes of 192 and 256 bits [20].

TABLE III. COMPARISON OF CLASSICAL AND QUANTUM SECURITY LEVELS FOR THE MOST USED CRYPTOGRAPHIC SCHEMES

Crypto Scheme	Key Size	Effective Key Strength/Security Level (in bits)	
		Classical Computing	Quantum Computing
RSA-1024	1024	80	0
RSA-2048	2048	112	0
ECC-256	256	128	0
ECC-384	384	256	0
AES-128	128	128	64
AES-256	256	256	128

E. Hash Functions

The family of hash functions suffer from a similar problem as symmetric ciphers since their security depends on a fixed output length. Grover’s algorithm can be utilized to find a collision in a hash function in square root steps of its original length (it is like searching an unsorted database). In addition, it has been proved that it is possible to combine Grover’s algorithm with the birthday paradox. Brassard et al. [21] described a quantum birthday attack. By creating a table of size $\sqrt[3]{N}$ and utilizing Grover’s algorithm to find a collision an attack is said to work effectively. This means that to provide a $b - bit$ security level against Grover’s quantum algorithm a hash function must provide at least a $3b - bit$ output. As a result, many of the present hash algorithms are disqualified for use in the quantum era. However, both SHA-2 and SHA-3 with longer outputs, remain quantum resistant.

V. POST-QUANTUM CRYPTOGRAPHY

The goal of post-quantum cryptography (also known as quantum-resistant cryptography) is to develop cryptographic systems that are secure against both quantum and conventional computers and can interoperate with existing communication protocols and networks [14]. Many post-quantum public key candidates are actively investigated the last years. In 2016, NIST announced a call for proposals of algorithms that are believed to be quantum resilient with a deadline in November 2017. In January 2018, NIST published the results of the first round. In total 82 algorithms were proposed from which 59 are encryption or key exchange schemes and 23 are signature schemes. After 3 to 5 years of analysis NIST will report the findings and prepare a draft of standards [22]. Furthermore, the National Security Agency (NSA) has already announced

plans to migrate their cryptographic standards to post-quantum cryptography [23].

The cryptographic algorithms presented in this section do not rely on the hidden subgroup problem (HSP) such as factorizing integers or computing discrete logarithms, but different complex mathematical problems.

A. Quantum Key Distribution

Quantum Key Distribution (QKD) addresses the challenge of securely exchanging a cryptographic key between two parties over an insecure channel. QKD relies on the fundamental characteristics of quantum mechanics which are invulnerable to increasing computational power, and may be performed by using the quantum properties of light, lasers, fibre-optics as well as free space transmission technology. QKD was first introduced in 1984 when Charles Bennett and Gilles Brassard developed their BB84 protocol [24, 25]. Research has led to the development of many new QKD protocols exploiting mainly two different properties that are described right below.

Prepare-and-measure (P&M) protocols use the Heisenberg Uncertainty principle [26] stating that the measuring act of a quantum state changes that state in some way. This makes it difficult for an attacker to eavesdrop on a communication channel without leaving any trace. In case of eavesdropping the legitimate exchange parties are able to discard the corrupted information as well as to calculate the amount of information that has been intercepted [27]. This property was exploited in BB84.

Entanglement based (EB) protocols use pairs of entangled objects which are shared between two parties. As explained in III, entanglement is a quantum physical phenomenon which links two or more objects together in such a way that afterwards they have to be considered as one object. Additionally, measuring one of the objects would affect the other as well. In practice when an entangled pair of objects is shared between two legitimate exchange parties anyone intercepting either object would alter the overall system. This would reveal the presence of an attacker along with the amount of information that the attacker retrieved. This property was exploited in E91 [28] protocol.

Both of the above-mentioned approaches are additionally divided into three families; discrete variable coding, continuous variable coding and distributed phase reference coding. The main difference between these families is the type of detecting system used. Both discrete variable coding and distributed phase reference coding use photon counting and post-select the events in which a detection has effectively taken place [29]. Continuous variable coding uses homodyne detection [29] which is a comparison of modulation of a single frequency of an oscillating signal with a standard oscillation.

A concise list of QKD protocols for the aforementioned families is presented below.

Discrete variable coding protocols:

- BB84 [24, 25] - the first QKD protocol that uses four non-orthogonal polarized single photon states or low-intensity light pulses. A detailed description of this protocol is given below.

- BBM [30] - is an entanglement based version of BB84.
- E91 [28] - is based on the *gedanken experiment* [31] and the generalized Bell's theorem [32]. In addition, it can be considered an extension of Bennett and Brassard's (authors of BB84) original idea.
- SARG04 [33, 34] - is similar to BB84 but instead of using the state to code the bits, the bases are used. SARG04 is more robust than BB84 against the photon number splitting (PNS) attack.
- Six state protocol [35–37] - is a version of BB84 that uses a six-state polarization scheme on three orthogonal bases.
- Six state version of the SARG04 coding [38].
- Singapore protocol [39] - is a tomographic protocol that is more efficient than the Six state protocol.
- B92 protocol [40] - two non-orthogonal quantum states using low-intensity coherent light pulses.

Continuous variable coding protocols:

- Gaussian protocols
 - Continuous variable version of BB84 [41]
 - Continuous variable using coherent states [42]
 - Coherent state QKD protocol [43] - based on simultaneous quadrature measurements.
 - Coherent state QKD protocol [44] - based on the generation and transmission of random distributions of coherent or squeezed states.
- Discrete-modulation protocols
 - First continuous variable protocol based on coherent states instead of squeezed states [45].

Distributed phase reference coding protocols:

- Differential Phase Shift (DPS) Quantum Key Distribution (QKD) protocol [46, 47] - uses a single photon in superposition state of three basis kets, where the phase difference between two sequential pulses carries bit information.
- Coherent One Way (COW) protocol [48, 49] - the key is obtained by a time-of-arrival measurement on the data line (raw key). Additionally, an interferometer is built on a monitoring line, allowing to monitor the presence of an intruder. A prototype was presented in 2008 [50].

Discrete variable coding protocols are the most widely implemented, whereas the continuous variable and distributed phase reference coding protocols are mainly concerned with overcoming practical limitations of experiments.

1) *BB84 protocol*: BB84 is the first quantum cryptographic protocol (QKD scheme) which is still in use today. According to Mayers [51] BB84 is *provable secure*, explaining that a secure key sequence can be generated whenever the channel bit error rate is less than about 7% [52]. BB84 exploits the polarization of light for creating random sequence of qubits (key) that are transmitted through a quantum channel.

BB84 uses two different bases, base 1 is polarized 0° (horizontal) or 90° (vertical) with 0° equal to 0 and 90° equal to 1. Base 2 is polarized 45° or 135° with 45° equal to 1 and 135° equal to 0. Alice begins by sending a photon in one of the two bases having a value of 0 or 1. Both the base and the value should be chosen randomly. Next, Bob selects the base 1 or 2 and measures a value without knowing which base Alice has used. The key exchange process continues until they have generated enough bits. Furthermore, Bob tells Alice the sequence of the bases he used but not the values he measured and Alice informs Bob whether the chosen bases were right or wrong. If the base is right, Alice and Bob have equal bits, whereas if it is wrong the bits are discarded. In addition, any bits that did not make it to the destination are discarded by Alice. Now Alice can use the key that they just exchanged to encode the message and send it to Bob. BB84 is illustrated visually in Fig. 1.

Worthy to mentioning is that this method of communication was broken by Lydersen et al. in 2010 [53]. Their experiment proved that although BB84 is *provable secure* the actual hardware implemented is not. The authors managed to inspect the secret key without the receiver noticing it by blinding the APD-based detector (avalanche photodiode).

Yuan et al. [54] proposed improvements to mitigate blinding attacks, such as monitoring the photocurrent for anomalously high values. Lydersen et al. [55] after taking into consideration the improvements of Yuan et al. [54] succeeded again to reveal the secret key without leaving any traces.

2) *Photon Number Splitting Attack*: The crucial issue in quantum key distribution is its security. In addition to noise in the quantum channel, the equipment is impractical to produce and detect single photons. Therefore, in practice, laser pulses are used. Producing multiple photons opens up a new attack known as Photon Number Splitting (PNS) attack. In PNS attack, an attacker (Eve) deterministically splits a photon off of the signal and stores it in a quantum memory which does not modify the polarisation of the photons. The remaining photons are allowed to pass and are transmitted to the receiver (Bob). Next, Bob measures the photons and the sender (Alice) has to reveal the encoding bases. Eve will then be able to measure all captured photons on a correct bases. Consequently, Eve will obtain information about the secret key from all signals containing more than one photon without being noticed [57].

Different solutions have been proposed for mitigating PNS attacks. The most promising solution developed by Lo et al. [58] uses decoy states to detect PNS attacks. This is achieved by sending randomly laser pulses with a lower average photon number. Thereafter, Eve cannot distinguish between decoyed signals and non-decoyed signals. This method works for both single and multi-photon pulses [59].

B. Mathematically-based Solutions

There are many alternative mathematical problems to those used in RSA, DH and ECDSA that have already been implemented as public key cryptographic schemes, and for which the Hidden Subgroup Problem (HSP) [60] does not apply; therefore, they appear to be quantum resistant.

The most researched mathematical-based implementations are the following:

- Lattice-based cryptography [61]
- Multivariate-based cryptography [62]
- Hash-based signatures [63]
- Code-based cryptography [64]

The existing alternatives and new schemes emerging from these areas of mathematics do not all necessarily satisfy the characteristics of an ideal scheme. In the following subsections we are going to give an overview of these cryptographic schemes.

1) *Lattice-based Cryptography*: This is a form of public-key cryptography that avoids the weaknesses of RSA. Rather than multiplying primes, lattice-based encryption schemes involve multiplying matrices. Furthermore, lattice-based cryptographic constructions are based on the presumed hardness of lattice problems, the most basic of which is the shortest vector problem (SVP) [61]. Here, we are given as input a lattice represented by an arbitrary basis and our goal is to output the shortest non-zero vector in it.

The Ajtai-Dwork (AD) [65], Goldreich-Goldwasser-Halevi (GGH) [66] and NTRU [67] encryption schemes that are explained below are lattice-based cryptosystems.

In 1997, Ajtai and Dwork[65] found the first connection between the worst and the average case complexity of the Shortest Vector Problem (SVP). They claimed that their cryptosystem is *provably secure*, but in 1998, Nguyen and Ster [68] refuted it. Furthermore, the AD public key is big and it causes message expansion making it an unrealistic public key candidate in post-quantum era.

The Goldreich-Goldwasser-Halevi (GGH) was published in 1997. GGH makes use of the Closest Vector Problem (CVP) which is known to be NP-hard. Despite the fact that GGH is more efficient than Ajtai-Dwork (AD), in 1999, Nguyen[69] proved that GGH has a major flaw; partial information on plaintexts can be recovered by solving CVP instances.

NTRU was published in 1996 by Hoffstein et al. [67]. It is used for both encryption (*NTRUEncrypt*) and digital signature (*NTRUSign*) schemes. NTRU relies on the difficulty of factorizing certain polynomials making it resistant against Shor's algorithm. To provide 128-bit post-quantum security level NTRU demands 12881-bit keys [70]. As of today there is not any known attack for NTRU.

In 2013, Damien Stehle and Ron Steinfeld developed a *provably secure* version of NTRU (SS-NTRU) [71].

In May 2016, Bernstein et al. [72] released a new version of NTRU called "NTRU Prime". NTRU Prime countermeasures the weaknesses of several lattice based cryptosystems, including NTRU, by using different more secure ring structures.

In conclusion, among all the lattice-based candidates mentioned above NTRU is the most efficient and secure algorithm making it a promising candidate for the post-quantum era.

2) *Multivariate-based Cryptography*: The security of this public key scheme relies on the difficulty of solving systems of multivariate polynomials over finite fields. Research has shown that development of an encryption algorithm based on

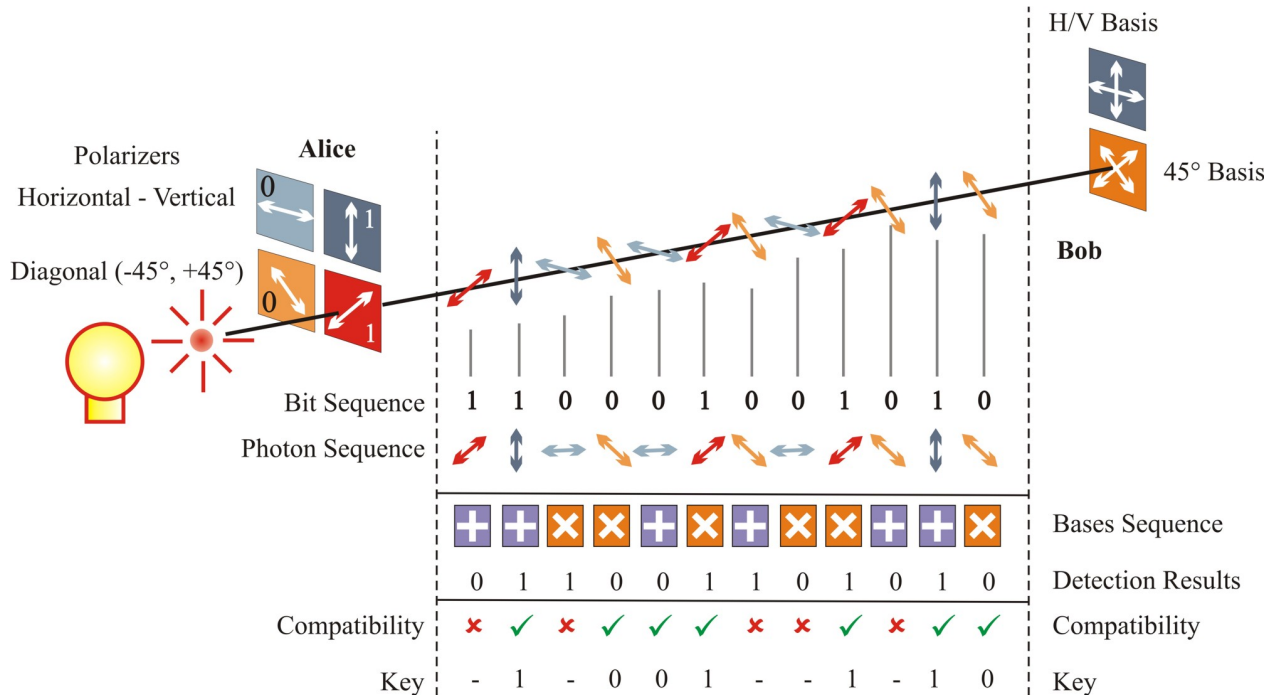


Fig. 1. Key exchange in the BB84 protocol implemented with polarization of photons (adapted from [56]).

Multivariate equations is difficult [13]. Multivariate cryptosystems can be used both for encryption and digital signatures. Tao et al. [73] explained that there have been several attempts to build asymmetric public key encryption schemes based on multivariate polynomials; however, most of them are insecure because of the fact that certain quadratic forms associated with their central maps have low rank. The authors [73] proposed a new efficient multivariate scheme, namely Simple Matrix (ABC), based on matrix multiplication that overcomes the aforementioned weakness. In addition, multivariate cryptosystems can be used for digital signatures. The most promising signature schemes include Unbalanced Oil and Vinegar (multivariate quadratic equations), and Rainbow. UOV has a large ratio between the number of variables and equations (3:1) making the signatures three times longer than the hash values. In addition, the public key sizes are large. On the other hand, Rainbow is more efficient by using smaller ratios which result in smaller digital signatures and key sizes [12].

3) *Hash-based Signatures*: In this subsection, we introduce the Lamport signature scheme invented in 1979 by Leslie Lamport. Buchmann et al. [18] introduced concisely the scheme. A parameter b defines the desired security level of our system. For 128-bit b security level we need a secure hash function that takes arbitrary length input and produces 256-bit length output; thus, SHA-256 is considered an optimal solution that can be fitted with our message m .

Private key: A random number generator is used to produce 256 pairs of random numbers. Each number is 256 bits. In total our generated numbers are $2 \times 256 \times 256 = 16$ KB. Therefore, we can precisely say that the private key consists of $8b^2$ bits.

Public key: All generated numbers (private key) are hashed

independently creating 512 different hashes (256 pairs) of 256-bit length each. Therefore, we can precisely say that the public key consists of $8b^2$ bits.

The next step is to sign the message. We have a hashed message m and then for each bit (depending on its value 0 or 1) of the message digest we choose one number from each pair that comprise the private key. As a result, we have a sequence of 256 numbers (relative to the bit sequence of the hashed message m). The sequence of numbers is the digital signature published along with the plaintext message. It is worth noting that the private key should never be used again and the remaining 256 numbers from the pairs should be destroyed (*Lamport one-time signature*).

The verification process is straightforward. The recipient calculates the hash of the message and then, for each bit of the hashed message we choose the corresponding hash from the public key (512 in number). In addition, the recipient hashes each number of the sender's private key which should correspond to the same sequence of hashed values with the recipient's correctly chosen public key values. The security of this system derives by the decision of using the private key only once. Consequently, an adversary can only retrieve 50 percent of the private key which makes it impossible to forge a new valid signature.

Buchmann et al. [18] explained that in case we want to sign more than one messages, chaining can be introduced. The signer includes in the signed message a newly generated public key that is used to verify the next message received.

Witernitz described a one time signature (WOTS) which is more efficient than Lamport's. Specifically, the signature size and the keys are smaller [74]. However, OTSs are not suitable for large-scale use because they can be used only once.

Merkle introduced a new approach that combines Winternitz's OTS with binary trees (Merkle Signature Scheme). A binary tree is made of nodes. In our case each node represents the hash value of the concatenation of the child nodes. Each of the leaf nodes (lowest nodes in the tree hierarchy) contains a Winternitz's OTS which is used for signing. The first node in the hierarchy of the tree known as root node is the actual public key that can verify the OTSs contained in the leaf nodes [74].

In 2013, A. Hulsing improved the WOTS algorithm by making it more efficient without affecting its security level even when hash functions without collision resistance are used [75].

Currently two hash-based signature schemes are under evaluation for standardization. Specifically, the eXtended Merkle Signature Scheme (XMSS) [76] which is a stateful signature scheme, and Stateless Practical Hash-based Incredibly Nice Collision-resilient Signatures (SPHINCS) [77] which is as the name indicates a stateless signature scheme.

4) *Code-based Cryptography*: Code-based cryptography refers to cryptosystems that make use of error correcting codes. The algorithms are based on the difficulty of decoding linear codes and are considered robust to quantum attacks when the key sizes are increased by the factor of 4. Furthermore, Buchmann et al. [18] state that the best way to solve the decoding problem is to transform it to a Low-Weight-Code-World Problem (LWCWP) but solving a LWCWP in large dimensions is considered infeasible. It would be easier to comprehend the process of this scheme by using Buchmann's [18] concise explanation of McEliece's original code-based public-key encryption system. We define b as the security of our system and it is a power of 2. $n = 4b \lg b$, $d = \lg n$, and $t = 0.5n/d$.

For example, if $b = 128$ then $n = 512 \log_2(128)$ which is equal to 3584. $d = 12$ and $t = 149$. The receiver's public key in this system is $d \times n$ matrix K with coefficients F_2 . Messages to be encrypted should have exactly t bits set to 1 and for the encryption the message m is multiplied by K . The receiver generates a public key with a hidden Goppa code structure (error-correction code) that allows to decode the message with Patterson's algorithm, or even by faster algorithms. The code's generator matrix K is perturbed by two invertible matrices which are used to decrypt the ciphertext to obtain the message m .

As for any other class of cryptosystems, the practice of code-based cryptography is a trade-off between efficiency and security. McEliece's cryptosystem encryption and decryption process are fast with very low complexity, but it makes use of large public keys (100 kilobytes to several megabytes).

VI. CONCLUSION

In today's world, where information play a particularly important role, the transmission and the storage of data must be maximally secure. Quantum computers pose a significant risk to both conventional public key algorithms (such as RSA, ElGamal, ECC and DSA) and symmetric key algorithms (3DES, AES). Year by year it seems that we are getting closer to create a fully operational universal quantum computer that

can utilize strong quantum algorithms such as Shor's algorithm and Grover's algorithm. The consequence of this technological advancement is the absolute collapse of the present public key algorithms that are considered secure, such as RSA and Elliptic Curve Cryptosystems. The answer on that threat is the introduction of cryptographic schemes resistant to quantum computing, such as quantum key distribution methods like the BB84 protocol, and mathematical-based solutions like lattice-based cryptography, hash-based signatures, and code-based cryptography.

ACKNOWLEDGMENT

This research is supported by the Research Council of Norway under the Grant No.: IKTPLUSS 247648 and 248030/O70 for Oslo Analytics and SWAN projects, respectively. This research is also part of the SecurityLab of the University of Oslo.

REFERENCES

- [1] M. Dušek, N. Lütkenhaus, and M. Hendrych, "Quantum cryptography," *Progress in Optics*, vol. 49, pp. 381–454, 2006.
- [2] C. Paar and J. Pelzl, "Introduction to Public-Key Cryptography," in *Understanding Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 149–171.
- [3] Z. Kirsch, "Quantum Computing: The Risk to Existing Encryption Methods," Ph.D. dissertation, Tufts University, Massachusetts, 2015, <http://www.cs.tufts.edu/comp/1116/archive/fall2015/zkirsch.pdf>.
- [4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. New York, NY, USA: Cambridge University Press, 2011.
- [5] R. Jozsa, "Entanglement and Quantum Computation," in *Geometric Issues in the Foundations of Science*, S. Huggett, L. Mason, K. Tod, S. Tsou, and N. Woodhouse, Eds. Oxford University Press, July 1997.
- [6] W. Tichy, "Is quantum computing for real?: An interview with catherine mcgeoch of d-wave systems," *Ubiquity*, vol. 2017, no. July, pp. 2:1–2:20, Jul. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3084688>
- [7] M. Soeken, T. Häner, and M. Roetteler, "Programming quantum computers using design automation," *arXiv preprint arXiv:1803.01022*, 2018.
- [8] S. Bone and M. Castro, "A Brief History of Quantum Computing," *Surveys and Presentations in Information Systems Engineering (SURPRISE)*, vol. 4, no. 3, pp. 20–45, 1997, http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/spb3/.
- [9] J. Muhonen and T. Dehollain, "Storing Quantum Information For 30 Seconds In a Nanoelectronic Device," *Nature Nanotechnology*, vol. 9, pp. 986–991, 2014.
- [10] D-Wave, "Quantum Computing: How D-Wave Systems Work," <http://www.dwavesys.com/our-company/meet-d-wave>.
- [11] L. S. Bishop, S. Bravyi, A. Cross, J. M. Gambetta, and J. Smolin, "Quantum volume," Technical report, 2017., Tech. Rep., 2017.
- [12] M. Campagna and C. Xing, "Quantum Safe Cryptography and Security: An Introduction, Benefits, Enablers and Challenges," ETSI, Tech. Rep. 8, 2015.
- [13] W. Buchanan and A. Woodward, "Will Quantum Computers be the End of Public Key Encryption?" *Journal of Cyber Security Technology*, vol. 1, no. 1, pp. 1–22, 2016.
- [14] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, "NIST: Report on Post-Quantum Cryptography," NIST, Tech. Rep., 2016.
- [15] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, ser. SFCS '94. Washington, DC, USA: IEEE Computer Society, 1994, pp. 124–134.

- [16] U. Vazirani, "On The Power of Quantum Computation," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1743, pp. 1759–1768, 1998.
- [17] L. Grover, "A Fast Quantum Mechanical Algorithm For Database Search," Bell Labs, New Jersey, Tech. Rep., 1996.
- [18] D. Bernstein, E. Dahmen, and Buch, *Introduction to Post-Quantum Cryptography*. Springer-Verlag Berlin Heidelberg, 2010.
- [19] J. Proos and C. Zalka, "Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves," *Quantum Info. Comput.*, vol. 3, no. 4, pp. 317–344, 2003.
- [20] National Security Agency, "National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information," NSA, Tech. Rep., 2003.
- [21] G. Brassard, P. Høyer, and A. Tapp, *Quantum Cryptanalysis of Hash and Claw-Free Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 163–169.
- [22] D. Moody, "The ship has sailed: The nist post-quantum crypto competition." [Online]. Available: <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/asiacrypt-2017-moody-pqc.pdf>
- [23] N. Kobitz and A. Menezes, "A riddle wrapped in an enigma," *IEEE Security Privacy*, vol. 14, no. 6, pp. 34–42, Nov 2016.
- [24] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution, and Coin-Tossing," in *Proc. 1984 IEEE International Conference on Computers, Systems, and Signal Processing*, no. 560, 1984, pp. 175–179.
- [25] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental Quantum Cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 3–28, 1992.
- [26] E. Panarella, "Heisenberg uncertainty principle," in *Annales de la Fondation Louis de Broglie*, vol. 12, no. 2, 1987, pp. 165–193.
- [27] H. Singh, D. Gupta, and A. Singh, "Quantum key distribution protocols: A review," *Journal of Computational Information Systems*, vol. 8, pp. 2839–2849, 2012.
- [28] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, p. 661, 1991.
- [29] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of modern physics*, vol. 81, no. 3, p. 1301, 2009.
- [30] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without bell's theorem," *Physical Review Letters*, vol. 68, no. 5, p. 557, 1992.
- [31] D. Bohm, *Quantum theory*. Courier Corporation, 1951.
- [32] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Physical review letters*, vol. 23, no. 15, p. 880, 1969.
- [33] V. Scarani, A. Acin, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," *Physical review letters*, vol. 92, no. 5, p. 057901, 2004.
- [34] A. Acin, N. Gisin, and V. Scarani, "Coherent-pulse implementations of quantum cryptography protocols resistant to photon-number-splitting attacks," *Physical Review A*, vol. 69, no. 1, p. 012309, 2004.
- [35] C. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pp. 175–179, 1984.
- [36] D. Bruß, "Optimal eavesdropping in quantum cryptography with six states," *Physical Review Letters*, vol. 81, no. 14, p. 3018, 1998.
- [37] H. Bechmann-Pasquinucci and N. Gisin, "Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography," *Physical Review A*, vol. 59, no. 6, p. 4238, 1999.
- [38] K. Tamaki and H.-K. Lo, "Unconditionally secure key distillation from multiphotons," *Physical Review A*, vol. 73, no. 1, p. 010302, 2006.
- [39] B.-G. Englert, D. Kaszlikowski, H. K. Ng, W. K. Chua, J. Řeháček, and J. Anders, "Efficient and robust quantum key distribution with minimal state tomography," *arXiv preprint quant-ph/0412075*, 2004.
- [40] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical review letters*, vol. 68, no. 21, p. 3121, 1992.
- [41] N. J. Cerf, M. Levy, and G. Van Assche, "Quantum distribution of gaussian keys using squeezed states," *Physical Review A*, vol. 63, no. 5, p. 052311, 2001.
- [42] F. Grosshans and P. Grangier, "Continuous variable quantum cryptography using coherent states," *Physical review letters*, vol. 88, no. 5, p. 057902, 2002.
- [43] C. Weedbrook, A. M. Lance, W. P. Bowen, T. Symul, T. C. Ralph, and P. K. Lam, "Quantum cryptography without switching," *Physical review letters*, vol. 93, no. 17, p. 170504, 2004.
- [44] J. Lodewyck, M. Bloch, R. García-Patrón, S. Fossier, E. Karpov, E. Diamanti, T. Debuisschert, N. J. Cerf, R. Tualle-Brouiri, S. W. McLaughlin *et al.*, "Quantum key distribution over 25 km with an all-fiber continuous-variable system," *Physical Review A*, vol. 76, no. 4, p. 042305, 2007.
- [45] C. Silberhorn, T. C. Ralph, N. Lütkenhaus, and G. Leuchs, "Continuous variable quantum cryptography: Beating the 3 db loss limit," *Physical review letters*, vol. 89, no. 16, p. 167901, 2002.
- [46] K. Inoue, E. Waks, and Y. Yamamoto, "Differential phase shift quantum key distribution," *Physical Review Letters*, vol. 89, no. 3, p. 037902, 2002.
- [47] —, "Differential-phase-shift quantum key distribution using coherent light," *Physical Review A*, vol. 68, no. 2, p. 022317, 2003.
- [48] N. Gisin, G. Ribordy, H. Zbinden, D. Stucki, N. Brunner, and V. Scarani, "Towards practical and fast quantum cryptography," *arXiv preprint quant-ph/0411022*, 2004.
- [49] D. Stucki, N. Brunner, N. Gisin, V. Scarani, and H. Zbinden, "Fast and simple one-way quantum key distribution," *Applied Physics Letters*, vol. 87, no. 19, p. 194108, 2005.
- [50] D. Stucki, C. Barreiro, S. Fasel, J.-D. Gautier, O. Gay, N. Gisin, R. Thew, Y. Thoma, P. Trinkler, F. Vannel *et al.*, "Continuous high speed coherent one-way quantum key distribution," *Optics express*, vol. 17, no. 16, pp. 13 326–13 334, 2009.
- [51] D. Mayers, "Unconditional Security in Quantum Cryptography," *Journal of the ACM*, vol. 48, no. 3, pp. 351–406, 2001.
- [52] C. Branciard, N. Gisin, B. Kraus, and V. Scarani, "Security of Two Quantum Cryptography Protocols Using The Same Four Qubit States," *Physical Review A*, vol. 72, no. 3, p. 032301, sep 2005.
- [53] L. Lydersen, C. Wiechers, D. E. C. Wittmann, J. Skaar, and V. Makarov, "Hacking Commercial Quantum Cryptography Systems by Tailored Bright Illumination," *Nature Photonics*, pp. 686–689., October 2010.
- [54] Z. Yuan, J. Dynes, and A. Shields, "Avoiding the Blinding Attack in QKD," *Nature Photonics*, vol. 4, pp. 800–801, December 2010.
- [55] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov, "Avoiding the Blinding Attack in QKD," *Nature Photonics*, vol. 4, pp. 801–801, December 2010.
- [56] V. Makarov, "Quantum Cryptography and Quantum Cryptanalysis," Ph.D. dissertation, Norwegian University of Science and Technology Faculty of Information Technology, NTNU, 2007, <http://www.vad1.com/publications/phd-thesis-makarov-200703.pdf>.
- [57] G. Brassard, N. Lütkenhaus, T. Mor, and B. C. Sanders, "Security aspects of practical quantum cryptography," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2000, pp. 289–299.
- [58] H.-K. Lo, X. Ma, and K. Chen, "Decoy state quantum key distribution," *Physical review letters*, vol. 94, no. 23, p. 230504, 2005.
- [59] M. Haitjema, "A survey of the prominent quantum key distribution protocols," 2007.
- [60] S. J. Lomonaco, J. Kauffman, and L. H., "Quantum Hidden Subgroup Problems: A Mathematical Perspective," *Quantum*, pp. 1–63., 2002.
- [61] D. Micciancio, "Lattice-Based Cryptography," in *Post-Quantum Cryptography*, 2009, no. 015848, pp. 147–192.
- [62] J. Ding and B.-Y. Yang, "Multivariate Public Key Cryptography," *Post-Quantum Cryptography*, pp. 193–241, 2009.
- [63] C. Dods, N. P. Smart, and M. Stam, "Hash Based Digital Signature Schemes," *Cryptography and Coding*, vol. 3796, pp. 96–115, 2005.
- [64] R. Overbeck and N. Sendrier, "Code-based Cryptography," in *Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 95–145.

- [65] M. Ajtai and C. Dwork, "A Public-Key Cryptosystem With Worst-Case/Average-Case Equivalence," *Proceedings of The 29th Annual ACM Symposium on Theory of Computing - STOC '97*, pp. 284–293., 1997.
- [66] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-Key Cryptosystems from Lattice Reduction Problems," *Advances in Cryptology - {CRYPTO} '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, vol. 1294, pp. 112–131, 1997.
- [67] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," *Algorithmic number theory*, pp. 267–288, 1998.
- [68] P. Nguyen and J. Stern, *Cryptanalysis of the Ajtai-Dwork Cryptosystem*. Springer Berlin Heidelberg, 1998, pp. 223–242.
- [69] P. Nguyen, "Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem," *Advances in Cryptology - CRYPTO*, vol. 1666, pp. 288–304, 1999.
- [70] P. S. Hirschhorn, J. Hoffstein, N. Howgrave-Graham, and W. Whyte, *Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 437–455.
- [71] D. Stehle and R. Steinfeld, "Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over Ideal Lattices," *Cryptology ePrint Archive*, Report 2013/004, 2013.
- [72] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU Prime," *IACR Cryptology ePrint Archive*, vol. 2016, p. 461, 2016.
- [73] C. Tao, A. Diene, S. Tang, and J. Ding, "Simple Matrix Scheme for Encryption," in *International Workshop on Post-Quantum Cryptography*. Springer, 2013, pp. 231–242.
- [74] R. C. Merkle, *A Certified Digital Signature*. New York, NY: Springer New York, 1990, pp. 218–238.
- [75] H. Andreas, *W-OTS+ –Shorter Signatures for Hash-Based Signature Schemes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–188.
- [76] J. Buchmann, E. Dahmen, and A. Hülsing, "XMSS-a Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions," *Post-Quantum Cryptography*, pp. 117–129, 2011.
- [77] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn, *SPHINCS: Practical Stateless Hash-Based Signatures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 368–397.