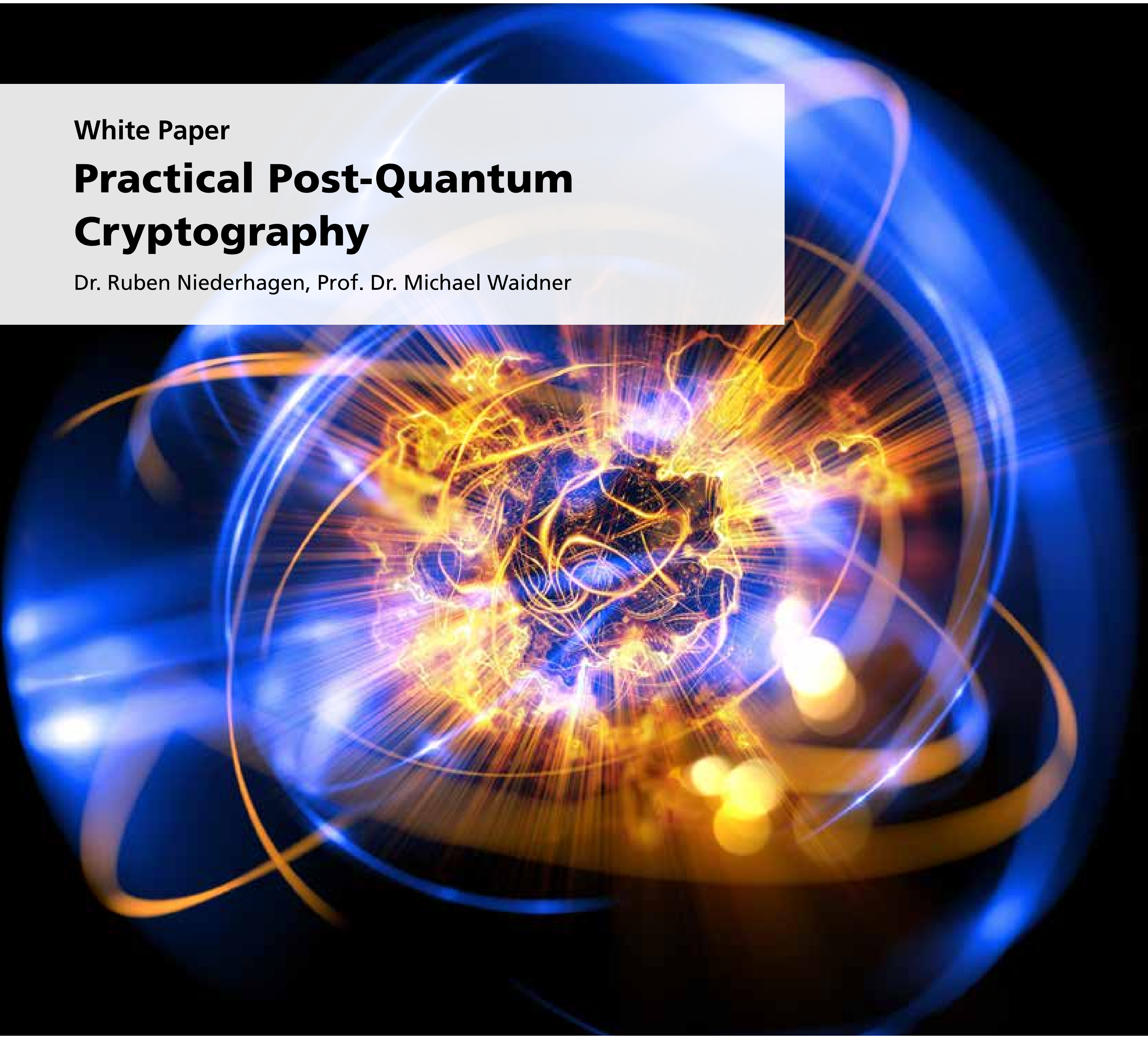
Fraunhofer
SIT

FRAUNHOFER INSTITUTE FOR SECURE INFORMATION TECHNOLOGY



White Paper

Practical Post-Quantum Cryptography

Dr. Ruben Niederhagen, Prof. Dr. Michael Waidner

Member of CRISP
Center for Research
in Security and Privacy

White Paper
Practical Post-Quantum Cryptography

August 18, 2017

Dr. Ruben Niederhagen

Department Cyber-Physical Systems Security (CSS)
ruben.niederhagen@sit.fraunhofer.de

Prof. Dr. Michael Waidner

Director Fraunhofer SIT *and*
Professor for Security in IT at TU Darmstadt
michael.waidner@sit.fraunhofer.de

Fraunhofer Institute for Secure Information Technology SIT
Rheinstraße 75, 64295 Darmstadt

Imprint

Contact:
Fraunhofer Institute for Secure Information Technology SIT
Rheinstraße 75, 64295 Darmstadt, Germany
Phone +49 6151 869-100
E-Mail info@sit.fraunhofer.de
URL www.sit.fraunhofer.de

Bibliographic information published by Die Deutsche Nationalbibliothek.
Die Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at www.dnb.de.
ISSN 2192-8169

Michael Waidner (Ed.)
SIT-TR-2017-02: Practical PostQuantum Cryptography
Ruben Niederhagen, Michael Waidner

Copyright cover picture: ©rybindmitriy / Fotolia

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. The quotation of those designations in whatever way does not imply the conclusion that the use of those designations is legal without the consent of the owner of the trademark.

Contents

Summary

Quantum computers are hanging over the security of our information like a sword of Damocles: We do not know when or even if quantum computers will become a reality — but once they arrive, they will break confidentiality, privacy, and authenticity of our modern communication. It will no longer be possible to trust digital certificates and signatures and it will no longer be possible to exchange secret keys for data encryption using current cryptographic primitives like RSA, ECC, DH, DSA, and so on. However, there is hope: The cryptographic community is working on *post-quantum cryptography* in order to provide alternatives using hard mathematical problems that cannot be broken by quantum computers. There is a zoo of alternative cryptographic primitives and protocols that are under investigation and standardization bodies like NIST and ETSI are starting processes to standardize post-quantum algorithms.

Yet, many challenges remain open, for example:

- Which schemes do we *trust*?
- Which *metrics* can we use in order to quantify the security of cryptographic schemes against quantum computers?
- What *parameters* do we choose in order to balance security and usability?
- How can we improve the *efficiency* of post-quantum schemes?
- How can we achieve *efficient implementations* of post-quantum schemes?
- How do we achieve *secure implementations* of post-quantum schemes?
- How can we *migrate* from current cryptography to post-quantum schemes?
- How can we *agilely* update products in the field?
- How do we make current systems *compatible* to post-quantum schemes?
- How do we *inform* industry, politics, and the public about quantum computing and post-quantum cryptography?
- What schemes and what parameters should we *standardize*?
- What is the impact of *legislation* and *regulation*?

Academic research in this area is mainly focusing on theoretical aspects of post-quantum cryptography while industry requires specific recommendations of cryptographic schemes, secure parameters, and implementations. This white paper gives an overview on the state-of-the-art in post-quantum cryptography in order to facilitate and motivate the conversation between academia, industry, and governments.

1 Introduction

Within the last three decades, digital communication has become a fundamental technology for modern society all around the globe. Its applications reach from *human-to-human* communication in telephony, mobile communication, email, and online chat via *human-to-machine* communication in Internet Commerce, online banking, telemedicine, and Industry 4.0 to *machine-to-machine* communication in aviation, automotive technology, and the Internet of things. These applications rely on the security of the communication, for example on its authenticity, privacy, and integrity. These security goals are achieved by the use of cryptography. The most important cryptographic primitives used today are:

- AES for symmetric encryption,
- RSA and ECC for public-key encryption,
- DSA and ECDSA for signatures,
- DH and ECDH for key exchange, and
- SHA-1, SHA-2, or SHA-3 for hashing.

These schemes are standardized by various entities, e.g., NIST, ISO, IETF, and BSI. They are considered secure against powerful attacks with conventional computing systems when secure parameters are used.

Cryptographic schemes rely on the assumption that certain mathematical or computational problems are hard to solve for an attacker. Many of the cryptographic primitives that we use today are based on the assumption that the *integer-factorization problem* and the *discrete-logarithm problem* are hard to solve. This assumption has proven reliable over the recent decades — in case traditional computing systems are used. However, in the mid-1980s, David Deutsch introduced the idea of using the laws of quantum physics for building a new type of computing systems [23], the *quantum computer*.

A classical computer performs operations on bits that can be in one of the two states "1" or "0" (also called "true" or "false", "high" or "low", "on" or "off"). A quantum computer performs operations on *qubits* that can be in the state "1" or "0" or in infinitely many *superposition* states of "1" and "0". At the end of a quantum computation, the qubits are *measured*. This forces qubits that are in superposition to snap into one of the two states "1" or "0" with a certain probability depending on their superposition state (see Figure 1.1).

The infinitely large space of superposition states during computation and the *entanglement* of qubits allows quantum computers to solve certain classes of

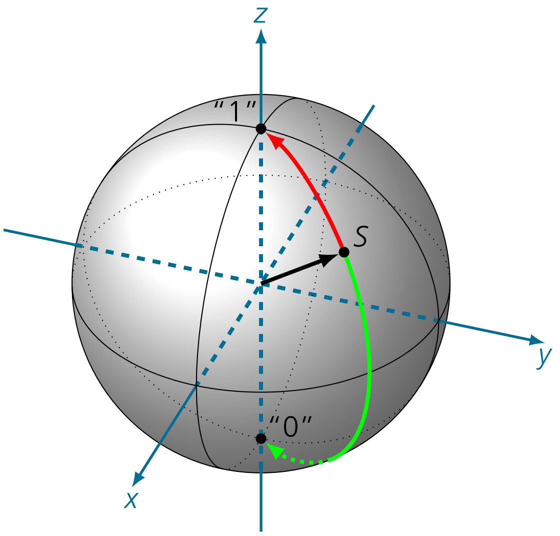


Figure 1.1: Representation of a qubit using a *Bloch sphere*. The state *S* of the qubit can be any point on the sphere. When the qubit is measured in respect to the *z*-axis, the state collapses to either "1" (red arrow) or to "0" (green arrow). The probability of the result depends on the latitude of the state.

problems much faster than classical computers. In particular, hard problems used in today's cryptography suddenly become feasible when a quantum computer is used. There are two fundamental algorithms of quantum computing that have an impact on the strength of cryptographic schemes:

Grover's algorithm for quantum computers gives a square-root speedup on search problems [33]. This improves brute-force algorithms that check every possible key. The square-root factor halves the exponent of the time complexity. This means, that for example a brute-force attack on AES-128 with a cost of at most 2^{128} AES-operations on a classical computing system can be finished with about 2^{64} AES-operations on a quantum computer [32]. The impact of Grover's algorithm can practically be averted by doubling security parameters. Doubling the key length of AES from 128-bit (AES-128) to 256-bit (AES-256) gives a cost of at least 2^{128} operations on a quantum computer and therefore is considered secure. Grover's algorithm also has an impact on the security of hash functions. It improves exhaustive search for preimages by a square-root factor [4] and exhaustive search for collisions by at most a cube-root factor [13, 7]. For example, computing a preimage for SHA-256 (SHA-2 with 256-bit has values) has a cost of at least $\sqrt{2^{256}} = 2^{128}$ operations on a quantum computer; computing a collision for SHA-256 has a cost of at least $\sqrt[3]{2^{256}} \approx 2^{85.3}$ operations.

Similar to AES and hash functions, the other cryptographic primitives listed above can be protected against Grover's algorithm by increasing their security parameters. Nevertheless, apart from AES and hash functions, they all are based on the hardness of either the integer factorization problem or the discrete-logarithm problem in a finite group. These mathematical problems are believed to be computationally hard; no efficient algorithm for solving these problems on classical computing systems is publicly known today. However, the second algorithm for quantum computers has an even more critical impact on the hardness these problems and thus on all cryptographic systems based on these computational problems.

Shor's algorithm solves integer factorization and discrete logarithms in polynomial time on a quantum computer [54, 55]. More generally, Shor's algorithm efficiently solves the hidden-subgroup problem for finite Abelian groups [39]. This directly breaks cryptographic primitives that are based on integer factorization, e.g., RSA, and the discrete-logarithm problem, e.g., Diffie-Hellman and ECC. The impact of Shor's algorithm cannot practically be mitigated by increasing the security parameters of the affected primitives, because the computational complexity of a quantum-computer attack using Shor's algorithm is similar to the computational complexity of using RSA and ECC. Choosing the security parameters large enough to defend against attacks by quantum computers makes using RSA and ECC itself infeasible.

Therefore, the only cryptographic primitives listed above that are able to withstand attacks by quantum computers are AES and hash functions with sufficiently large security parameters. All commonly used asymmetric primitives are going to be broken by quantum computers using Shor's algorithm. However, we are not unprepared: There are alternatives for the threatened primitives.

One way to protect data against quantum computers is to use quantum technology itself in a constructive way. There are schemes for quantum key distribution (QKD) that rely on quantum physics as opposed to mathematics. However, this does not make QKD schemes naturally secure against hacking [41]; they require defenses against physical attacks. An inherent problem of QKD is its requirement of a *pre-shared secret* for mutual authentication. Providing pairwise pre-shared secrets for a large number of communication endpoints in advance is infeasible for many applications. Furthermore, these schemes typically require point-to-point fiber-optic connections or line-of-sight in order to transmit photons from sender to receiver. Therefore, they are not compatible with the existing communication infrastructure and they do not scale to a communication network of the size of the Internet. They are also not useful for mobile communication, e.g., mobile phones, car-to-x communication, and wireless sensor networks.

A more practical solution therefore is to use different cryptographic schemes that rely on hard problems that cannot be solved efficiently on quantum computers. Cryptographic algorithms that are assumed to be secure against attacks by quantum computers are called *post-quantum cryptography*. Post-quantum cryptography is designed to replace the existing cryptographic primitives and is compatible with existing computing devices and communication systems. Currently, the cryptographic community is discussing several families of primitives for post-quantum cryptography. Each family relies on a different mathematical problem that is believed to be hard to solve even when the attacker has access to a quantum computer. The cryptographic community is investigating which of the proposed approaches is the most efficient and provides the best protection for data and information. However, only few of the proposed alternative schemes are already sufficiently mature for standardization and deployment.

Currently it is not yet foreseeable when sufficiently large and efficient quantum computers will be operational. It is not even guaranteed that all technical problems in the construction of quantum computers can be solved. However, physicists are quite optimistic that the technical problems eventually will be solved. Depending on the data that needs to be protected, it might not be sufficient to wait with the deployment of post-quantum cryptography until quantum computers actually are operational. Internet packets or encrypted data can be recorded or stolen today and stored until technology has advanced and is able to break the encryption. Therefore, data with high security requirements must be securely transmitted and stored as soon as possible. Furthermore, past has shown how slowly new cryptographic primitives are deployed and how slowly insecure primitives vanish. New standards of post-quantum secure cryptographic schemes are required as soon as possible. In 2016, NIST started standardization efforts with the intention to standardize quantum-secure algorithms (i.e., cryptographic algorithms that withstand attacks by quantum computers) within the next ten years [18].

Already today, the long-term threat by quantum computers must be considered for industrial appliances and products that will be deployed for a long time. Automotive technologies that are currently under development will reach the market within the next two to five years and will be in use for fifteen years and longer. Therefore, technology development and product life span may cover twenty to thirty years. Industrial appliances of Industry 4.0 have a similar life span. The longer the development and deployment of long-term, post-quantum secure technology takes, the higher is the risk that products and appliances will be vulnerable in the far future.

Industry is beginning to show an interest in using and commercialising post-quantum cryptography. For example, Google experimentally deployed the post-quantum key-exchange scheme NewHope for some connections between the Chrome browser and Google servers [12]. Intel Labs is doing research on integrating post-quantum cryptography into their products and their production process, e.g., for secure communication with chip-production facilities [14]. The post-quantum public-key schemes NTRUEncrypt and NTRUSign were commercially developed in the 1990s and are now licensed by the company Security Innovation[1]. Some companies are offering products that are advertised as post-quantum secure, e.g., PQ Solutions Limited[2] and InfoSec Global[3]. Other companies are offering software libraries, solutions, and consulting for post-quantum cryptography, e.g., evolutionQ Inc.[4], ISARA Corporation[5], and CryptoExperts[6]. This shows that post-quantum cryptography has left the academic realm and reached practical application for securing critical information.

[1] www.securityinnovation.com

[2] www.post-quantum.com

[3] www.infosecglobal.com

[4] www.evolutionq.com

[5] www.isara.com

[6] www.cryptoexperts.com

2 Challenges

There is a number of challenges that need to be solved in order to enable post-quantum cryptography for practical application:

Trust. Traditionally, trust in cryptographic schemes has grown over time: The longer no crucial attacks have been found against a scheme, the more a scheme is trusted. Some post-quantum schemes have been around for a while, are well trusted, and considered sufficiently mature for deployment. Other schemes are fairly young. If we want to deploy recently developed post-quantum schemes within the next five to ten years, we do not have time to "age" their trust. Instead, newly invented cryptographic schemes require thorough *security analysis* and *security proofs* that inspire trust without a long waiting period.

Metrics. The security metric for cryptographic schemes in respect to classical attacks is relatively well understood: the security parameters must be chosen such that the best known attack has a cost well above a certain computational threshold. However, neither the cost of the best known attack nor the threshold are fixed. Better attacks that reduce the computational cost and faster computing systems that raise the threshold are being developed. Consequently, the security parameters need to be updated from time to time. Post-quantum cryptography must follow a similar approach in respect to classical attacks. For some schemes, there is still a lot of development in the efficiency of classical attacks: the best known attacks improve rapidly. However, post-quantum schemes also require security metrics for attacks using quantum computers. The efficiency of quantum computations is not yet well understood. There is no large quantum computer available for practical analysis of quantum attacks. Therefore, all estimations on the security of post-quantum schemes against quantum computers are purely theoretical. This implies two problems: If the power of quantum computers is *underestimated*, security parameters might be too weak and schemes will be broken once quantum computers arrive. If the power of quantum computers is *overestimated*, security parameters are chosen too strong which reduces the usability and efficiency of the schemes and hinders their wide-spread deployment. Therefore, finding a precise metric for the security of cryptographic schemes against quantum computers is crucial.

Parameters. Given a precise security metric, we need to specify security parameters for the post-quantum schemes. Currently, many publications are focusing on security parameters that are secure against classical attacks, because this facilitates comparison with classical cryptographic schemes like RSA and ECC and because the security metrics against classical attacks are well understood. However, the main benefit of post-quantum schemes is their resistance against quantum-computer attacks. Therefore, we require post-quantum secure parameters for post-quantum schemes.

Efficient Schemes. Different post-quantum schemes have different resource requirements. However, currently not many schemes have competitive efficiency compared to classical cryptographic schemes. Therefore, we need improvements for the post-quantum schemes in order to reduce their resource requirements. There is ongoing research on how to reduce key-sizes and computational cost of post-quantum schemes. However, attempts to reduce resource requirements for example by introducing some redundant structure often resulted in a loss of security. The price for long-term secure post-quantum cryptography likely is higher cost in computation, storage, and communication demand.

Efficient Implementations. Despite ongoing efforts to reduce the resource requirements of post-quantum schemes, highly optimized implementations of post-quantum schemes will be required. In particular embedded and passively powered devices require efficient hardware implementations in order to reduce power demand and computation time. Nevertheless, some low-cost or legacy devices will not provide a sufficient amount of resources for post-quantum cryptography and investments in more powerful hardware will be required.

Secure Implementations. Implementing post-quantum algorithms in a secure manner poses new challenges. Side-channel cryptanalysis, fault injection attacks, and physical cryptanalysis have become powerful threats to classical cryptographic implementations. Similar attacks need to be considered for the implementation of post-quantum schemes. Post-quantum schemes might expose further attack vectors that need to be anticipated and secured. This research requires experts for both post-quantum cryptography and hardware security.

Migration. Existing systems need to be migrated to post-quantum security. Therefore, software implementations require secure update mechanisms. Fixed hardware implementations e.g., smart cards or security tokens will need to be replaced. During the transition time, while the security of post-quantum schemes has not yet been fully verified, a *hybrid* approach using both classical and post-quantum cryptography will be required: By using both classical and post-quantum schemes together, one can achieve high security even in case the post-quantum scheme might turn out to be insecure. Google has been using this approach in their experimental deployment of the post-quantum scheme NewHope [12].

Agility. Not all cryptographic primitives need to be replaced by post-quantum primitives right away. For example, authentication does not yet need to be post-quantum secure before quantum computers are built — an attacker cannot retro-actively break authentication that was performed in the past. However, ephemeral key exchange and symmetric encryption must be secure against quantum computers long time before attacks are using quantum computers — otherwise, an attacker is able to break into previously recorded communication. Certificates and digital signatures that expire in the far future must be secure against attacks using quantum computers. If signatures are not post-quantum secure yet, they must be renewed with post-quantum schemes before quantum computers are available. Therefore, whenever a secure post-quantum scheme is not yet required or available, applications must use agile protocols and update mechanisms that allow to upgrade to post-quantum primitives once they become available and before quantum computers are available.

Compatibility. Some post-quantum schemes have additional requirements for their execution environment that are different from classical schemes. For example, the signature scheme XMSS is *stateful*, i.e., an internal state needs to be stored between the computation of consecutive signatures. This state must not be lost and also must not return to an earlier state. This breaks interoperability with backup strategies that are designed to preserve older copies of data but do not guarantee that the most recent changes can be recovered. Using stateful signature schemes requires an adaptation of the data backup procedure; in the worst case a loss of the private signature key is preferable over returning to an old state because re-using the signature state might enable an attacker to forge valid signatures using only publicly available information. In general, the interoperability of post-quantum schemes with existing security infrastructures needs to be investigated.

Education. There are many misconceptions about the power of quantum computers in the public. Often quantum computers are falsely believed to provide instantaneous solutions to arbitrary computational problems. This is not the case; the actual power of quantum computers is limited to very specific algorithms and they provide improvements only to a limited number of applications. Coincidentally, cryptanalysis is one of these applications and the impact on the field of secure communication is severe. In order to draw the correct conclusions for development, management, regulations, and funding, the public in general and in particular managers, engineers, and politicians need to be informed about the impact of quantum computing and the solutions provided by cryptography.

Standardization. In 2016, the National Institute for Standards and Technology (NIST) in the US started a standardization process for post-quantum cryptography [46]. Also the European Telecommunications Standards Institute (ETSI) is working on the standardization of “quantum-safe” cryptography [28]. The standardization process depends on the input of academia and industry in order to achieve secure and usable standards.

Legislation and Regulation. There are national and international laws and regulations on qualified digital signatures (e.g., SigG in Germany), protection of private information (e.g., EU data protection rules), and the security of network and information systems (e.g., NIS Directive). These laws do not specify which specific technical procedures, cryptographic schemes, or parameters must be implemented but dictate that the goal of data protection must be achieved. Therefore, entities that process private data or offer qualified signatures are required by law to protect against state-of-the-art attacks. This will eventually also apply to attacks using quantum computers and appropriate protection mechanisms against quantum computers will become mandatory.

3 Families of Post-Quantum Schemes

The cryptographic community is discussing five different families of post-quantum cryptography, namely:

- code-based cryptography,
- lattice-based cryptography,
- hash-based cryptography,
- multivariate cryptography, and
- supersingular elliptic-curve isogeny cryptography.

Each of these families is based on different mathematical problems that are hard to solve both with traditional computers as well as quantum computers. They differ in efficiency, e.g., in the size of public and private keys, sizes of cipher texts and key-exchange messages, and computational cost, their maturity, and the amount of trust in their strength.

Efficiency of post-quantum schemes is important because it determines how well the schemes can be used on current and future devices, in particular on devices with few resources or limited network bandwidth like embedded and handheld devices. In general post-quantum schemes require more resources compared to traditional cryptography, in particular ECC. Therefore, security against quantum-computer attacks comes at a cost.

Some post-quantum schemes have been known and investigated for many years. For example code-based [43] and hash-based [44] schemes were introduced at the end of the 1970s. Therefore, code-based and hash-based cryptography is well understood and trusted. Multivariate cryptography developed over the 1980s [42] and its underlying mathematical problem is well understood as well. However, constructing an efficient public-key cryptosystem based on multivariate cryptography is challenging and only few multivariate public-key schemes are considered secure. Lattice-based schemes were introduced in the end of the 1990s [1]. Recently there have been advances in classical cryptanalysis of hash-based schemes. Therefore, the understanding and trust in lattice-based schemes is growing rapidly. Cryptography based on elliptic-curve isogenies was proposed in 2006 [52] and refined using supersingular curves in 2011 [38]. This approach has some distinct features that are interesting for the implementation of efficient key-exchange protocols. However, it is the most juvenile family of post-quantum cryptography and not yet deeply understood and not considered ready for practical application.

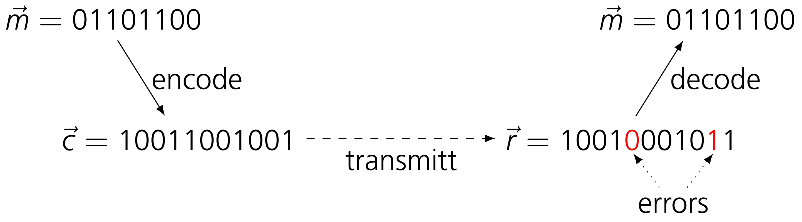


Figure 3.1: Example for error correction on an unreliable channel. The error-correcting code enables the receiver to correct a certain number of bit-errors during decoding.

3.1 Code-based cryptography

The basic idea of code-based public-key encryption is to use error-correcting codes in order to hide the contents of a message during transmission. Traditionally, error-correction codes are used to detect and correct bit errors when messages are transmitted over an unreliable channel. The code can be chosen to fulfill the requirements of the channel; in particular the number t of correctable bit errors can be determined.

First, the message $\vec{m}$ is converted into a code word $\vec{c}$ of the respective code (see Figure 3.1). This adds redundancy, i.e., the code word is longer than the message. Then $\vec{c}$ is transmitted over the channel. During transmission several bits of $\vec{c}$ might be flipped, the receiver does not receive $\vec{c}$ but $\vec{r} = \vec{c} \oplus \vec{e}$ where $\vec{e}$ is an error vector of some weight w (w bits in $\vec{e}$ are 1, the other bits are 0). Now, the receiver maps $\vec{r}$ to the closest code word $\vec{c'}$ in the code. If the number of errors in $\vec{r}$ is smaller than the number of errors that can be corrected, i.e., $w \leq t$, then $\vec{c'}$ is equal to the original code word $\vec{c}$ (otherwise decoding fails). Finally, the receiver applies the inverse of the encoding operation to $\vec{c'}$ and obtains the original message $\vec{m}$. However, decoding arbitrary (random) codes is computationally hard and can be infeasible depending on the code parameters. Nevertheless, there are specific codes for which efficient decoding algorithms are known. Therefore, in practice only such codes are used that have efficient decoding algorithms.

The main security assumption of code-based cryptography is the hardness of decoding a random linear code [49]. Even when taking quantum computers into account, only exponential-time algorithms are known. The first code-based public-key cryptosystem was proposed by McEliece in 1978 [43]. This scheme has not been fundamentally broken since, although the original parameters from 1978 are not considered secure anymore.

Instead of correcting errors of an unreliable channel, for the McEliece scheme we now assume to have a reliable channel and we deliberately add an error in order to protect the contents of a message against an eavesdropper. The public key of the receiver is a generator matrix G^{pub} of his code. The sender encrypts the message $\vec{m}$ by converting it into a code word and by adding a secret error vector $\vec{e}$ of weight t:

$$\vec{c} = \vec{m}G^{\text{pub}} \oplus \vec{e}.$$

The receiver decodes the "corrupted" code word $\vec{c}$ and obtains $\vec{m}$. Now, to make this cryptosystem secure, the attacker must not be able to distinguish the code from a random code. The public generator matrix G^{pub} must be "scrambled" in order to hide the secret structure of the code such that it does not give the attacker any information that allows him to use an efficient decoding algorithm in order to decode $\vec{c}$. The McEliece public-key cryptosystem describes how to compute G^{pub} in such a way while still allowing the owner of the private key to decode messages efficiently [43].

The main problem of the McEliece cryptosystem is the size of the keys. McEliece is using binary Goppa codes and requires key sizes of about 4MB in order to achieve post-quantum security. An alternative to McEliece is a variant due to Niederreiter [47]. Niederreiter gives some improvements to encryption and decryption cost and requires smaller public-key sizes. Furthermore, Niederreiter introduced a trick that can be used to further reduce the size of public keys in code-based schemes, e.g., to about 1MB for both McEliece and Niederreiter cryptosystems using Goppa codes [10].

Niederreiter uses a slightly different approach than McEliece in order to construct a public-key system. The basic idea is not to add a random error to the codeword before transmission but to encode the plain text *as error*, i.e., as a weight-t bit string. Instead of a generator matrix G^{pub}, a parity-check matrix H^{pub} is used as public key. The sender encodes the plain text as a bit string $\vec{e}$ with weight w and computes the syndrome

$$\vec{s} = H^{\text{pub}}\vec{e}^{\mathsf{T}}.$$

The receiver uses a syndrome-decoding algorithm to recover the original error vector $\vec{e}$. Again, the public parity check matrix H^{pub} must be "scrambled" such that the underlying secret structure is not revealed to an attacker [47].

This process is particularly suitable when sender and receiver want to share a random bit string, e.g., as key for symmetric encryption. In this case, the sender simply generates a random bit string with weight w and transmits it to the receiver as described. Then both the sender and the receiver hash the bit string in order to obtain a shared secret key for symmetric encryption.

The cryptographic community has strong confidence in the McEliece cryptosystem and in Niederreiter's cryptosystem using Goppa codes. The main problem of code-bases systems is the huge size of the public key. There have been several attempts to reduce key sizes by using different codes that have some "compressible" redundant structure in the public key (e.g., quasi-cyclic moderate parity check codes (QC-MDPC) [45]); however, in many cases, this structure has led to efficient classical (i.e. non-quantum) attacks on the cryptosystems.

Apart from public-key cryptosystems, there are also signature schemes [22], hash functions [5], and random-number generators [31] based on code-based cryptography.

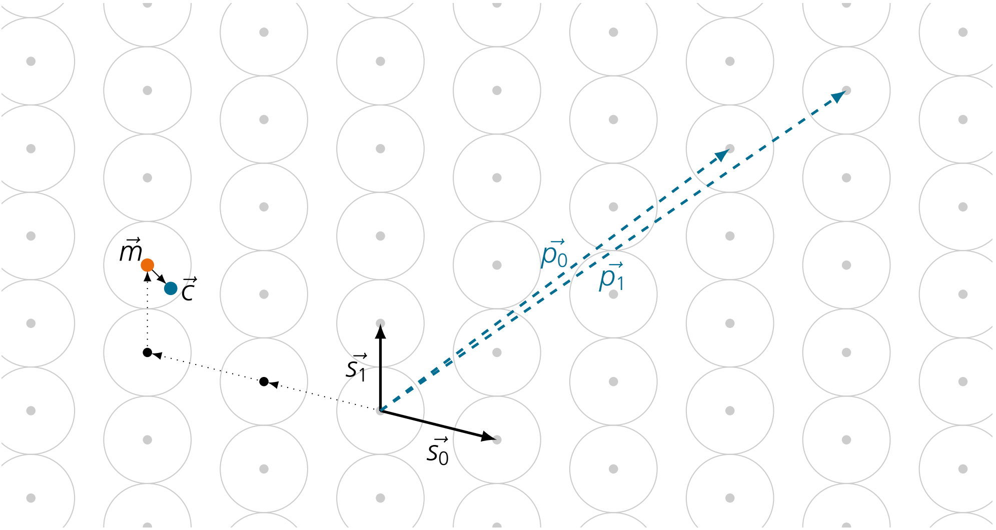


Figure 3.2: Example for lattice-based encryption in a two-dimensional lattice: The secret, well-formed base is $\{\vec{s}_0, \vec{s}_1\}$; the public, "scrambled" base is $\{\vec{p}_0, \vec{p}_1\}$. The sender uses $\{\vec{p}_0, \vec{p}_1\}$ to map the message to a lattice point $\vec{m}$ and adds an error vector to obtain the point $\vec{c}$. The point $\vec{c}$ is closer to $\vec{m}$ than to any other lattice point. Therefore, the receiver can use the well-formed secret base $\{\vec{s}_0, \vec{s}_1\}$ to easily recover $\vec{m}$ (dotted vectors); this is a hard computation for an attacker who only has the scrambled base $\{\vec{p}_0, \vec{p}_1\}$. For a secure scheme, the dimension of the lattice must be much higher than 2 as in this example.

3.2 Lattice-based cryptography

The underlying hard problem for lattice-based cryptography is the *shortest vector problem*: it is computationally hard to find the shortest vector in a high-dimensional lattice. The basic idea for constructing public-key encryption schemes using lattices is to use a well-formed high-dimensional lattice base s as secret private key and a scrambled version p of this base as public key (see Figure 3.2). For encryption, the sender of a message maps the message to a point $\vec{m}$ in the lattice using the public scrambled base. Then, the sender adds a random error to the lattice point such that the resulting point $\vec{c}$ is still closer to the original point $\vec{m}$ than to any other point in the lattice. This distorted point $\vec{c}$ is the cipher text which is sent to the receiver. Since the receiver is in possession of the secret, well-formed basis s of the lattice, he can recover the original lattice point $\vec{m}$ (the lattice point that is closest to the distorted cipher point) with low computational effort and obtain the original message.

The basic assumption for the security of this scheme is that an attacker who is not in possession of the well-formed base but only of the public scrambled base needs to spend an infeasible amount of computation in order to decipher the message. Finding a closest lattice point using the scrambled base (closest vector problem, CVP) and recomputing the well-formed base from the scrambled base (shortest vector problem, SVP) is believed to be computationally hard even for quantum computers. Other lattice-based schemes are based on the more general "learning with errors" (LWE) problem, which is closely related to coding theory and has security reductions to variants of SVP.

NTRUEncrypt is a commercial public-key encryption scheme based on lattices. The scheme has been patented by the company NTRU Cryptosystems which was acquired by Security Innovation in 2009. Security Innovation released the NTRU-Encrypt patents into public domain in March 2017. The initial parameters have been proven insecure; current presumably secure parameters require public key sizes of about 1.5 kB to 2.0 kB (for 256-bit classical security). The cipher text has the same length as the public key. Recent improvements to NTRUEncrypt are based on the Ring-LWE problem [56].

Closely related to NTRUEncrypt is the signature scheme NTRUSign. The original version of NTRUSign was broken, but there exist improved versions that prevent known attacks. Further lattice-based signature schemes are, e.g., BLISS [27], GLP [34], and TESLA [2]. However, the security of lattice-based schemes against quantum-computer attacks is not yet well-understood. Therefore, often there are no specific parameter recommendations for these signature schemes for post-quantum security. These schemes are quite juvenile and their security is under investigation (e.g., [15]).

Besides public-key encryption and signature schemes, there are key-exchange protocols that make use of the LWE problem. A prominent example is the protocol NewHope [3] that has been experimentally adopted by Google [12]. Unlike the classical Diffie-Hellman (DH) protocol, NewHope is not symmetric and needs two rounds for key agreement; it is rather based on public-key encryption, using a new key for each key exchange. Similar to the DH protocol, NewHope does not include authentication which needs to be achieved by other means. The rationale behind this design decision is to achieve long-term security of sensitive data for low cost. Breaking today's long-term public keys in the future, e.g., by using a quantum computer, does not break the privacy of the communication if a secure ephemeral key exchange protocol is used. By switching to post-quantum ephemeral key exchange now, an attacker in the future does not learn encryption keys even if he breaks long-term authentication keys. Therefore, combining a post-quantum ephemeral key exchange with a classical authentication scheme provides a cost-efficient, long-term secure authenticated key exchange for the interim period until all cryptographic primitives have been transitioned to post-quantum secure schemes.

3.3 Hash-based cryptography

The approach of hash-based cryptography is conceptually different from code-based and lattice-based cryptography. Hash functions are one-way functions that map bit-strings of an arbitrary length to relatively short, fixed-length bit-strings called *hash values*. There are three properties that are required for a cryptographic hash function:

1. *Preimage resistance*: It must be hard to compute a preimage of a hash value, i.e., a bit string that once hashed results in a given hash value.
2. *Second preimage resistance*: Given a bit string, it must be hard to find a different bit string that has the same hash value.
3. *Collision resistance*: It must be hard to find two arbitrary bit strings that have the same hash value.

Grover's algorithm gives at most the usual square-root speedup on brute-force preimage computations [4]. The best known classical algorithms for computing hash collisions are based on the birthday paradox and give a square-root speedup over brute-force search [48]. Grover's algorithm improves upon this speed-up and gives at most a cube-root speedup on brute-force collision search [13] — but the impact of this improvement is under dispute [7]. Hash functions are not affected by Shor's polynomial-time quantum algorithm. Therefore, hash functions are good candidates for the construction of post-quantum schemes. However, since by definition it is not computationally feasible to compute the inverse of a hash function, it is not known how to construct public-key encryption schemes using hash functions. Nevertheless, it is possible to construct signature schemes using only hash functions as building blocks.

As a basic example for the functionality of hash-based signatures consider the following scenario using a hash function h: Alice wants to sign a single-bit message. She creates a private signature key by randomly choosing two bit strings r_0 and r_1. She computes her public key as $\{s_0 = h(r_0), s_1 = h(r_1)\}$ and publishes $\{s_0, s_1\}$. Bob receives the public key and verifies that $\{s_0, s_1\}$ belongs to Alice. Eventually, when Alice wants to sign a one-bit message $m \in \{0, 1\}$, she publishes r_m together with the message. For example, let 1 encode "true" and 0 encode "false". For signing the message "true", Alice publishes r_1. Bob can easily verify the signature by computing $h(r_1)$ and comparing it to the public key element s_1. The signature must be from Alice since only she knew the preimage r_1 of s_1 and since it is computationally infeasible for an attacker to compute a preimage from s_1. However, this example describes a one time signature scheme: Alice can no longer use this private key since publishing the other value from her private key would reveal all private information to the public, Bob could no longer distinguish whether Alice or somebody else signed subsequent messages. Another obvious drawback of this basic scheme is the extremely limited length of the messages.

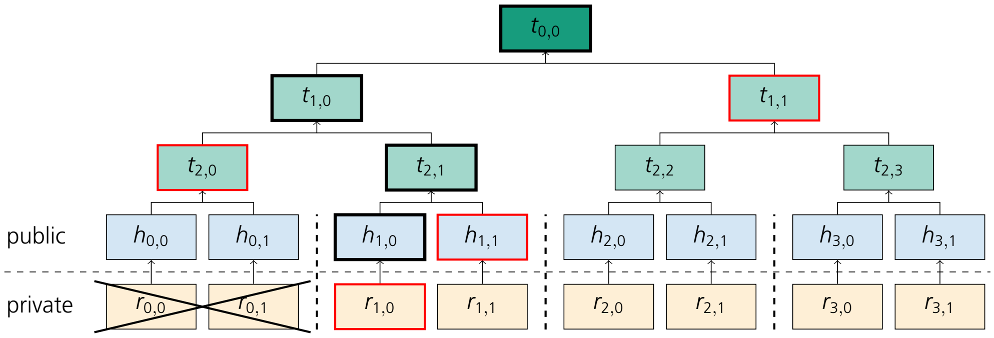


Figure 3.3: Example for a many-time signature scheme for single-bit messages. The public key of Alice is the root $t_{0,0}$ of the tree. The first component of the private key $(r_{0,0}, r_{0,1})$ has already been used and must not be used again. For signing the next message "false", Alice publishes the private key component $r_{1,0}$ of her second private key pair and the *verification path* $\{h_{1,1}, t_{2,0}, t_{1,1}\}$. Bob can verify the message "false" by first hashing $r_{1,0}$ in order to obtain $h_{1,0}$. Now, he can follow the verification path by computing $t_{2,1} = \text{hash}(h_{1,0}, h_{1,1})$ and $t_{1,0} = \text{hash}(t_{2,0}, t_{2,1})$ until he reaches the public key $t_{0,0} = \text{hash}(t_{1,0}, t_{1,1})$. Since only Alice knows all secret r-values, only she can have initially computed her public key $t_{0,0}$. For signature schemes that allow to sign longer messages than just single-bit messages, the private key values can be replaced by hash chains.

More elaborate hash-based signature schemes allow to perform more than one signature by using tree structures (see Figure 3.3) and an arbitrary message length by using hash chains. The total amount of possible signatures is typically limited and specified by a parameter for the scheme; in general, the signature size increases if more signatures for the same public key are required.

Hash-based signature schemes are considered very mature and have very reliable security estimates. The hash-based signature scheme XMSS [16] is currently in the standardization process by the IETF for use in Internet protocols [17]. A potential disadvantage of XMSS is its statefulness: In order to avoid re-usage of private key material, a state needs to be maintained that marks what parts of the private key already have been used and which parts are still available. For some applications, this might introduce additional cost. Also backup strategies might cause problems for state-based schemes because it is not secure to fall back to an old key state in case of data loss.

There are hash-based signature schemes that avoid maintaining a key state. A recent example is SPHINCS [9]. The price of stateless schemes compared to stateful schemes are larger signature sizes. The size of the public key for both stateless and stateful schemes is relatively small, typically about 64 bytes. The signature size of stateful schemes is in the range of 2-3 kB, the signature size of stateless schemes is about 40 kB.

$$
\begin{aligned}
x_0x_3 + x_2x_3 + x_0 + 1 &= 0\\
x_0x_1 + x_2x_3 + x_2 + 1 &= 0\\
x_0x_1 + x_0x_3 + x_0 + x_1 + 1 &= 0\\
x_1x_2 + x_2x_3 + x_3 &= 0
\end{aligned}
$$

Figure 3.4: Example for a multivariate polynomial system of four equations in four variables $x_0, \ldots, x_3$ (i.e., multivariate) of maximum degree two (i.e., quadratic) over $\mathbb{F}_2$. This particular quadratic system is small and therefore easy to solve. A solution of this system is $x_0 = 1$, $x_1 = 0$, $x_2 = 1$, $x_3 = 0$.

3.4 Multivariate cryptography

Multivariate cryptography is based on the hardness of the $\mathcal{MQ}$-problem. Solving multivariate quadratic systems of equations over finite fields is NP-hard: As opposed to linear systems, there is no efficient algorithm for solving random multivariate polynomial systems. Figure 3.4 shows an example for a small multivariate polynomial system. The hardness of solving a specific system depends on the size of the underlying finite field, the number of variables, and the degree of the system. Nevertheless, if the number of equations and variables is sufficiently large, even systems of quadratic equations over $\mathbb{F}_2$ (degree two, smallest finite field) are hard to solve.

Classically, multivariate polynomial systems can be solved using different algorithms. In a brute-force search, all possible values for the variables are tested until the correct solution is found. This requires an exponential amount of time depending on the number of variables [11]. An asymptotically more efficient approach is to solve the system numerically; there are several algorithms with different properties, e.g., the F_4/F_5 family [29, 30] and algorithms based on extended linearization (XL) [19]. When taking quantum computers into account, Grover's algorithm gives only the usual square-root speedup on exhaustive search [53].

For constructing an asymmetric public-key system, the public key itself is a set of multivariate quadratic polynomials and the private key often is the knowledge of a trapdoor that allows to efficiently solve the multivariate system. Usually the trapdoor is constructed by computing the public key $\mathcal{P}$ of m polynomials in n variables as the composition of two affine maps T and S and one quadratic map $\mathcal{Q}$ which is chosen such that it can be easily inverted [26], so $\mathcal{P} = T \circ \mathcal{Q} \circ S$. For signing a message z, Alice computes signature w by computing $z' = \text{hash}(z)$, $y = T^{-1}(z')$, $x = \mathcal{Q}^{-1}(y)$, and $w = S^{-1}(x)$. Bob can simply verify the signature using the public key $\mathcal{P}$ of Alice by checking that $\text{hash}(z) = \mathcal{P}(w)$. The public key $\mathcal{P}$ must be constructed in such a way that an attacker is not able to invert the system. However, the composition of matrices T, $\mathcal{Q}$, and S is not necessarily a hard instance of the $\mathcal{MQ}$-problem. Therefore, several compositions that were proposed have been broken. Nevertheless, there are constructions that are believed to be strong.

This construction allows to use under-defined systems for signature schemes, i.e., systems with more variables than equations. The equation system composed of z and $\mathcal{P}$ may have more than one solution. Any of these solutions is a valid signature. The confidence in multivariate signature schemes is quite high. For example, there is a consensus that the HFEv- signature scheme [50, 40] can be considered secure. The disadvantage of HFEv- is its relatively large public key [51]. Research on parameters that are post-quantum secure is still ongoing. Under the assumption that systems of 200–256 variables over $\mathbb{F}_2$ (or systems of similar entropy over larger fields) are required to withstand attacks by quantum computers, the size of the public key is between 500kB and 1MB. Other examples for promising multivariate signature schemes are Rainbow [25] and MQDSS [36].

There are attempts to reduce the key sizes by using systems in fewer variables but of higher degree and over much larger fields. However, the security of such systems is not well understood. A second approach is to use sparse systems in order to compress the public key. However, the sparsity usually leads to a loss in security; construction of secure sparse systems is an open research question [26, Sec. 6.1].

The situation looks different for public-key encryption schemes: Here the resulting equation system must have only one solution, otherwise the cipher text cannot be uniquely decrypted. In order to achieve this for arbitrary inputs, the system must be over-defined, i.e., the public key $\mathcal{P}$ must have more polynomials than variables. Many constructions for public-key encryption that have been proposed were broken quickly because the trapdoor could not effectively be hidden from an attacker. Currently, there are not many multivariate public-key encryption schemes that are considered secure. An example is the PMI-plus public-key encryption system [24]. PMI-plus is secure against known attack strategies but it is considered to be too premature for confidence in its security. Building a strong, efficient, and secure multivariate encryption scheme is an open challenge.

Constructions based on random multivariate systems can also be used for pseudo random-number generators, cryptographic hash functions, and symmetric encryption. For example, the symmetric block cipher QUAD [6] is using $m + n$ quadratic polynomials with n variables over $\mathbb{F}_2$. These polynomials are not secret. QUAD uses a state that is initialized with a secret n-bit key. In each iteration, the equations are evaluated at the state vector. The result of the first m polynomials is appended to the key stream, the state is updated with the result of the last n polynomials. Iteratively, this allows to compute a key stream of arbitrary length. This key stream is xored on the data stream. The disadvantage of QUAD is its low efficiency compared to AES.

3.5 Supersingular elliptic-curve isogeny cryptography

Classical elliptic-curve cryptography (ECC) works on points on specific elliptic curves: operations like addition and scalar multiplication are performed on points and also the exchanged data structures in cryptographic protocols are coordinates of points. However, instead of computing on points of an elliptic curve, one can also define operations between different elliptic curves. Operations that map a curve onto another curve have different properties. Maps with certain properties are called isogenies.

Using isogenies between elliptic curves for building cryptographic schemes is a relatively new approach compared to the schemes described in the previous sections. Public-key cryptosystems based on isogenies were introduced in 2006 by Rostovtsev and Stolbunov [52, 57]. A major drawback of the scheme was the long computation time required for encryption and decryption. Even worse, in 2010 Childs, Jao and Soukharev found a subexponential quantum computer attack on this scheme [20].

In 2011 Jao and De Feo extended the idea of using isogenies on ordinary elliptic curves to *supersingular* elliptic curves [38]. Due to the special structure of supersingular elliptic curves, the Childs/Jao/Soukharev attack does not work. Furthermore, the efficiency of encryption and decryption is greatly improved. However, due to the novelty of cryptographic schemes based on isogenies of supersingular elliptic curves, there is not yet great confidence in these schemes. Therefore, they are currently not consensually considered as candidates for post-quantum public-key encryption.

Nevertheless, due to their symmetric nature, schemes based on isogenies on supersingular elliptic curves have a very similar structure to classical DH and ECDH schemes. In particular, isogenies are the only post-quantum approach that enables a Diffie-Hellman like key exchange, the supersingular isogeny Diffie-Hellman (SIDH) key exchange. There are SIDH implementations with very competitive performance and small message sizes for the key exchange [21]. If indeed supersingular elliptic-curve isogenies prove to be secure against classical as well as quantum-computer attacks, they are very interesting candidates for post-quantum cryptography.

4 Comparison of Post-Quantum Schemes

The different families of post-quantum schemes vary heavily in their resource requirements. Figure 4.1 shows a comparison of key and message sizes of selected post-quantum schemes. Post-quantum schemes in general require larger public keys and larger signature/cipher text/message sizes than classical schemes. However, this cost is the price for schemes that are secure against attacks using quantum computers. RSA, ECC, and DH are not an option in scenarios that are taking quantum computers into account.

In many cases, the attempt to reduce the resource requirements by introducing additional structure into the schemes resulted in successful attacks. Therefore, for some schemes, a reduction of the public-key size or the data storage/transmission requirements might not be possible.

Signatures. Arguably the most trusted public-key signature schemes are hash-based schemes. They require small public keys of 64–1,056 bytes which is in the range of classical RSA and ECC signatures. However, the size of hash-based signatures is 2.5–41 kB which is much larger than the sizes of classical signatures. Multivariate-based schemes are still under investigation. They require public-key sizes of 500 kB to 1 MB which is much larger than classical schemes but they have very small signature sizes.

Public-key encryption. There is strong confidence in the McEliece and Niederreiter encryption schemes (using Goppa codes). The size of their public keys is about 1 MB and therefore very large compared to classical schemes. The size of the cipher text (used, e.g., for key encapsulation) is only around 190 bytes which is in the range of classical schemes. Lattice-based schemes are also quite mature but probably less trusted compared to code-based schemes. The NTRUEncrpyt scheme for example requires 1.5–2.0 kB for both keys and cipher text.

Key-exchange. Key-exchange schemes can easily be constructed from public-key encryption schemes by generating and transmitting new keys for each session. Specific key-exchange schemes provide better performance in respect to key-generation time and bandwidth demand. The fairly recent lattice-based key-exchange scheme NewHope requires to send network packages of about 2 kB compared to only 32–64 bytes for classical ECDH. The supersingular-isogeny scheme SIDH requires only 564 bytes and therefore is much closer to classical schemes. However, this scheme is very young and not well trusted.

Scheme		**Public key size**	**Data size**
		(bytes)	(bytes)
Public-key signatures:			
• Hash based:			
– XMSS (stateful)	[17]	64	2,500 – 2,820
– SPHINCS (state free)	[9]	1,056	41,000
• Multivariate based:			
– HFEv-*	[51]	500,000 – 1,000,000	25 – 32
Public-key encryption:			
• Code based:			
- McEliece	[10]	958,482 – 1,046,739	187 – 194
• Lattice based:			
– NTRUEncrypt	[35, 37]	1,495 – 2,062	1,495 – 2,062
Key exchange:			
• Lattice based:			
– NewHope	[3]	—	1,824 – 2,048
• Supersingular isogenies:			
– SIDH	[21]	—	564
Classical schemes:			
• RSA:			
– RSA-2048		256	256
– RSA-4096		512	512
• ECC:			
– 256-bit		32	32
– 512-bit		64	64
• Key exchange:			
– DH		—	256 – 512
– ECDH		—	32 – 64

* Values using field $\mathbb{F}_2$ and parameter n (number of variables) between 200 and 256.

Figure 4.1: Sizes of public keys and transmitted data (signature, cipher text, or key-exchange message respectively) for several post-quantum schemes in comparison to some classical schemes.

5 Conclusion

Within the last three decades, the theoretical idea of building universal quantum computers has led to successful practical experiments and to solutions of fundamental technical problems for the construction of quantum computers. We still do not known when or even if large and reliable quantum computers can be built — but neither can we be sure that building practical quantum computers is impossible. Therefore, we must take the threat of quantum computers against information security sincere and we must prepare the transition to post-quantum cryptography ahead of time to be ready in case quantum computers become a reality.

Post-quantum cryptography is aiming to provide cryptographic primitives that are secure against attacks using quantum computers. It is using mathematical problems that are believed to be hard to solve by both classical and quantum computers. Several post-quantum schemes are well understood and are considered strong candidates for standardization and practical application.

Important tasks in order to widely deploy quantum-secure schemes are:

- *Standardization:* Standardization bodies, e.g., NIST, IETF, and ETSI, have started to standardize post-quantum cryptography. The standardization process requires feedback on the strength, efficiency, security, and practical applicability of post-quantum schemes.

- *Implementation:* Industry requires efficient and secure software implementations of (standardized) post-quantum schemes that provide compatible interfaces with current software and that are compatible with current hardware. Furthermore, new hardware devices, e.g., smart cards, security tokens, hardware security modules (HSMs), and cryptographic coprocessors need to be developed that implement post-quantum cryptography.

- *Testing:* Software and hardware implementations of post-quantum secure schemes must be tested: Theoretically secure cryptographic schemes could be broken due to faulty implementations. Implementations need to be checked for and secured against side-channel attacks.

- *Education:* Industry, politics, and the public need to be informed about the exact computational powers of quantum computers and about the existence of and need for post-quantum cryptography. Aside from cryptanalysis, quantum computers have many positive practical applications in physics, biology, chemistry, and so on, but there are many myths about quantum computers that need to be dissolved.

Bibliography

[1] M. Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *ACM Symposium on Theory of Computing — STOC '96*. ACM, 1996, pp. 99–108 (cit. on p. 9).

[2] E. Alkim, N. Bindel, J. Buchmann, Ö. Daelen, and P. Schwabe. *TESLA: Tightly-Secure Efficient Signatures from Standard Lattices*. IACR Cryptology ePrint Archive, Report 2015/755. 2015 (cit. on p. 13).

[3] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. "Post-quantum key exchange — a new hope". In: *Proceedings of the 25th USENIX Security Symposium*. USENIX Association, 2016 (cit. on pp. 13, 20).

[4] M. Amy, O. D. Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck. *Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3*. arXiv:1603.09383. 2016 (cit. on pp. 3, 14).

[5] D. Augot, M. Finiasz, and N. Sendrier. "A Family of Fast Syndrome Based Cryptographic Hash Functions". In: *Progress in Cryptology — Mycrypt 2005*. Ed. by E. Dawson and S. Vaudenay. Vol. 3715. LNCS. Springer, 2005, pp. 64–83 (cit. on p. 11).

[6] C. Berbain, H. Gilbert, and J. Patarin. "QUAD: A Practical Stream Cipher with Provable Security". In: *Advances in Cryptology — EUROCRYPT 2006*. Ed. by S. Vaudenay. Vol. 4004. LNCS. Springer, 2006, pp. 109–128 (cit. on p. 17).

[7] D. J. Bernstein. "Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?" In: *Workshop Record of SHARCS'09: Special-purpose Hardware for Attacking Cryptographic Systems*. 2009 (cit. on pp. 3, 14).

[8] D. J. Bernstein, J. Buchmann, and E. Dahmen, eds. *Post-Quantum Cryptography*. ISBN: 978-3-540-88701-0. Springer, 2009 (cit. on pp. 24, 26).

[9] 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". In: *Advances in Cryptology — EUROCRYPT 2015*. Ed. by M. Fischlin and E. Oswald. Vol. 9056. LNCS. Springer, 2015, pp. 368–397 (cit. on pp. 15, 20).

[10] D. J. Bernstein, T. Lange, and C. Peters. "Attacking and Defending the McEliece Cryptosystem". In: *Post-Quantum Cryptography — PQCrypto 2008*. Ed. by J. Buchmann and J. Ding. Vol. 5299. LNCS. Springer, 2008, pp. 31–46 (cit. on pp. 11, 20).

[11] C. Bouillaguet, C.-M. Cheng, T. Chou, R. Niederhagen, and B.-Y. Yang. “Fast Exhaustive Search for Quadratic Systems in $\mathbb{F}_2$ on FPGAs”. In: *Selected Areas in Cryptography — SAC 2013*. Ed. by T. Lange, K. Lauter, and P. Lisonek. Vol. 8282. LNCS. Springer, 2013, pp. 205–222 (cit. on p. 16).

[12] M. Braithwaite. *Experimenting with Post-Quantum Cryptography*. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html. 2016 (cit. on pp. 5, 7, 13).

[13] G. Brassard, P. Høyer, and A. Tapp. “Quantum cryptanalysis of hash and claw-free functions”. In: *Theoretical Informatics — LATIN’98*. Ed. by C. L. Lucchesi and A. V. Moura. Vol. 1380. LNCS. Springer, 1998, pp. 163–169 (cit. on pp. 3, 14).

[14] E. Brickell. *Intel Strategy for Post Quantum Crypto*. https://pqcrypto2016.jp/data/Brickell-Post_Quantum_Strategy-PQC_2016_final.pdf. Invited talk at Post-Quantum Cryptography — PQCrypto 2016. 2016 (cit. on p. 5).

[15] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom. “Flush, Gauss, and Reload – A Cache Attack on the BLISS Lattice-Based Signature Scheme”. In: *Cryptographic Hardware and Embedded Systems — CHES 2016*. Ed. by B. Gierlichs and A. Y. Poschmann. Vol. 9813. LNCS. Springer, 2016, pp. 323–345 (cit. on p. 13).

[16] J. Buchmann, E. Dahmen, and A. Hülsing. “XMSS — A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions”. In: *Post-Quantum Cryptography — PQCrypto 2011*. Ed. by B.-Y. Yang. Vol. 7071. LNCS. Springer, 2011, pp. 117–129 (cit. on p. 15).

[17] D. Butin, A. Hülsing, A. Mohaisen, and S.-L. Gazdag. *XMSS: Extended Hash-Based Signatures*. Internet-Draft draft-irtf-cfrg-xmss-hash-based-signatures-07. Work in Progress. Internet Engineering Task Force, 2016 (cit. on pp. 15, 20).

[18] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. *Report on Post-Quantum Cryptography*. DRAFT NISTIR 8105. NIST, 2016 (cit. on p. 5).

[19] C.-M. Cheng, T. Chou, R. Niederhagen, and B.-Y. Yang. “Solving Quadratic Equations with XL on Parallel Architectures”. In: *Cryptographic Hardware and Embedded Systems — CHES 2012*. Ed. by E. Prouff and P. Schaumont. Vol. 7428. LNCS. Springer, 2012, pp. 356–373 (cit. on p. 16).

[20] A. Childs, D. Jao, and V. Soukharev. “Constructing elliptic curve isogenies in quantum subexponential time”. In: *Journal of Mathematical Cryptology* 8.1 (2014). arXiv:1012.4019, pp. 1–29 (cit. on p. 18).

[21] C. Costello, P. Longa, and M. Naehrig. “Efficient Algorithms for Supersingular Isogeny Diffie-Hellman”. In: *Advances in Cryptology — CRYPTO 2016*. Ed. by M. Robshaw and J. Katz. Vol. 9814. LNCS. Springer, 2016, pp. 572–601 (cit. on pp. 18, 20).

[22] N. T. Courtois, M. Finiasz, and N. Sendrier. "How to Achieve a McEliece-Based Digital Signature Scheme". In: *Advances in Cryptology — ASIACRYPT 2001*. Ed. by C. Boyd. Vol. 2248. LNCS. Springer, 2001, pp. 157–174 (cit. on p. 11).

[23] D. Deutsch. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 400.1818 (1985), pp. 97–117 (cit. on p. 2).

[24] J. Ding and J. E. Gower. "Inoculating multivariate schemes against differential attacks". In: *International Workshop on Public Key Cryptography*. Vol. 3958. LNCS. Springer, 2006, pp. 290–301 (cit. on p. 17).

[25] J. Ding and D. Schmidt. "Rainbow, a New Multivariable Polynomial Signature Scheme". In: *Applied Cryptography and Network Security — ACNS 2005*. Ed. by J. Ioannidis, A. Keromytis, and M. Yung. Vol. 3531. LNCS. Springer, 2005, pp. 164–175 (cit. on p. 17).

[26] J. Ding and B.-Y. Yang. "Multivariate Public Key Cryptography". In: *Post-Quantum Cryptography* [8]. Ed. by D. J. Bernstein, J. Buchmann, and E. Dahmen. Springer, 2009, pp. 193–241 (cit. on pp. 16, 17).

[27] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. "Lattice Signatures and Bimodal Gaussians". In: *Advances in Cryptology — CRYPTO 2013*. Ed. by R. Canetti and J. A. Garay. Vol. 8042. LNCS. Springer, 2013, pp. 40–56 (cit. on p. 13).

[28] European Telecommunications Standards Institute (ETSI). *Quantum-Safe Cryptography*. http://www.etsi.org/technologies-clusters/technologies/quantum-safe-cryptography. 2016 (cit. on p. 8).

[29] J.-C. Faugère. "A new efficient algorithm for computing Gröbner bases (F_4)". In: *Journal of Pure and Applied Algebra* 139.1–3 (1999), pp. 61–88 (cit. on p. 16).

[30] J.-C. Faugère. "A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)". In: *International Symposium on Symbolic and Algebraic Computation — ISSAC 2002*. ACM, 2002, pp. 75–83 (cit. on p. 16).

[31] J.-B. Fischer and J. Stern. "An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding". In: *Advances in Cryptology — EUROCRYPT 1996*. Ed. by U. Maurer. Vol. 1070. LNCS. Springer, 1996, pp. 245–255 (cit. on p. 11).

[32] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. "Applying Grover's Algorithm to AES: Quantum Resource Estimates". In: *Post-Quantum Cryptography — PQCrypto 2016*. Ed. by T. Takagi. Vol. 9606. LNCS. Springer, 2016, pp. 29–43 (cit. on p. 3).

[33] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Symposium on Theory of Computing — STOC '96*. ACM, 1996, pp. 212–219 (cit. on p. 3).

[34] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems". In: *Cryptographic Hardware and Embedded Systems — CHES 2012*. Ed. by E. Prouff and P. Schaumont. Vol. 7428. LNCS. Springer, 2012, pp. 530–547 (cit. on p. 13).

[35] J. Hoffstein, J. Pipher, and J. H. Silverman. "NTRU: A ring-based public key cryptosystem". In: *Algorithmic Number Theory: Third International Symposiun, ANTS-III*. Ed. by J. P. Buhler. Vol. 1423. LNCS. Springer, 1998, pp. 267–288 (cit. on p. 20).

[36] A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe. "From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures". In: *Advances in Cryptology — Asiacrypt 2016*. Ed. by J. H. Cheon and T. Takagi. Vol. 10032. LNCS. Springer, 2016, pp. 135–165 (cit. on p. 17).

[37] *IEEE P1363.1: Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*. IEEE Std 1363.1. 2008 (cit. on p. 20).

[38] D. Jao and L. De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography — PQCrypto 2011*. Ed. by B.-Y. Yang. Vol. 7071. LNCS. Springer, 2011, pp. 19–34 (cit. on pp. 9, 18).

[39] R. Jozsa. "Quantum factoring, discrete logarithms, and the hidden subgroup problem". In: *Computing in Science Engineering* 3.2 (2001), pp. 34–43 (cit. on p. 4).

[40] A. Kipnis, J. Patarin, and L. Goubin. "Unbalanced Oil and Vinegar Signature Schemes". In: *Advances in Cryptology — EUROCRYPT '99*. Ed. by J. Stern. Vol. 1592. LNCS. Springer, 1999, pp. 206–222 (cit. on p. 17).

[41] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov. "Hacking commercial quantum cryptography systems by tailored bright illumination". In: *Nature Photonics* 4.10 (2010), pp. 686–689 (cit. on p. 4).

[42] T. Matsumoto and H. Imai. "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption". In: *Advances in Cryptology — EUROCRYPT 1988*. Ed. by D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, et al. Vol. 330. LNCS. Springer, 1988, pp. 419–453 (cit. on p. 9).

[43] R. J. McEliece. "A Public-Key Cryptosystem Based On Algebraic Coding Theory". In: *Deep Space Network Progress Report* 44 (Jan. 1978), pp. 114–116 (cit. on pp. 9–11).

[44] R. C. Merkle. *Secrecy, authentication, and public key systems*. Ph.D. thesis, Electrical Engineering, Stanford. 1979 (cit. on p. 9).

[45] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto. "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes". In: *IEEE International Symposium on Information Theory — ISIT 2013*. 2013, pp. 2069–2073 (cit. on p. 11).

[46] National Institute of Standards and Technology (NIST). *Post-Quantum crypto Project*. http://csrc.nist.gov/groups/ST/post-quantum-crypto/. 2016 (cit. on p. 8).

[47] H. Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory". In: *Problems of Control and Information Theory* 15 (1986), pp. 19–34 (cit. on p. 11).

[48] P. C. van Oorschot and M. J. Wiener. "Parallel Collision Search with Cryptanalytic Applications". In: *Journal of Cryptology* 12.1 (1999), pp. 1–28 (cit. on p. 14).

[49] R. Overbeck and N. Sendrier. "Code-based cryptography". In: *Post-Quantum Cryptography* [8]. Ed. by D. J. Bernstein, J. Buchmann, and E. Dahmen. Springer, 2009, pp. 95–145 (cit. on p. 10).

[50] J. Patarin. "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms". In: *Advances in Cryptology — EUROCRYPT '96*. Ed. by U. Maurer. Vol. 1070. LNCS. Springer, 1996, pp. 33–48 (cit. on p. 17).

[51] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, and J. Ding. "Design Principles for HFEv- Based Multivariate Signature Schemes". In: *Advances in Cryptology — ASIACRYPT 2015*. Ed. by T. Iwata and J. H. Cheon. Vol. 9452. LNCS. Springer, 2015, pp. 311–334 (cit. on pp. 17, 20).

[52] A. Rostovtsev and A. Stolbunov. *Public-key Cryptosystem Based on Isogenies*. IACR Cryptology ePrint Archive, Report 2006/145. 2006 (cit. on pp. 9, 18).

[53] P. Schwabe and B. Westerbaan. "Solving Binary $\mathcal{MQ}$ with Grover's Algorithm". In: *Security, Privacy, and Applied Cryptography Engineering — SPACE 2016*. Ed. by C. Carlet, M. A. Hasan, and V. Saraswat. Vol. 10076. LNCS. Springer, 2016, pp. 303–322 (cit. on p. 16).

[54] P. W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Foundations of Computer Science*. IEEE, 1994, pp. 124–134 (cit. on p. 4).

[55] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Review* 41.2 (1999), pp. 303–332 (cit. on p. 4).

[56] D. Stehlé and R. Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *Advances in Cryptology — EUROCRYPT 2011*. Ed. by K. G. Paterson. Vol. 6632. LNCS. Springer, 2011, pp. 27–47 (cit. on p. 13).

[57] A. Stolbunov. "Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves". In: *Advances in Mathematics of Communications* 4.2 (2010), pp. 215–235 (cit. on p. 18).