Chapter 2
General Information about Cryptography

If the security assessments are not trusted, especially in cases of very high data security requirements, physical methods must be used for key generation and distribution, data encryption, and integrity and authenticity procedures. This ensures the confidentiality, integrity, and authenticity of the data against computationally powerful adversaries and mathematical attack methods that are still unknown (unpublished) today. This chapter provides an introduction to these topics, briefly compares mathematical methods, and discusses the generation and distribution of cryptographic keys, as well as QKD (Quantum Key Distribution), RKD (Radio signal Key Distribution), and MKD (Memeory Key Distribution).

2.1 Mathematical Versus Physical Methods in Cryptography

Physical cryptography methods can be used to achieve provably absolute data security in order to ensure the confidentiality, integrity, and authenticity of data, including the necessary key management. However, this statement does not assess the security of the underlying physical laws [Bern18],[1] i.e., the laws of quantum mechanics, the randomness of non-deterministic random number generators, etc. are not questioned.

If, for reasons of security assessment, mathematical cryptographic methods are dispensed with and physical methods are used, mathematics does not have to be dispensed with altogether. This raises the question of which mathematics may still be used. This mainly concerns post-processing in QKD and RKD, which has not yet been sufficiently investigated in terms of security for the individual methods currently in use.

[1] https://doi.org/10.48550/arXiv.1803.04520.

E. Piller and H. Schölnast, *Data Encryption at the Intersection of Mathematics and Physics*, SpringerBriefs in Information Security and Cryptography,
https://doi.org/10.1007/978-3-032-24764-3_2

2.2 Mathematical Methods of Cryptography

Cryptography can be used to achieve three main security objectives: confidentiality, integrity, and authenticity. Confidentiality refers to the encryption of data, while integrity refers to the ability to verify whether the data transmitted during telecommunications or stored during data storage has been altered. Authenticity allows the origin or authorship of the data to be determined.

When using mathematical cryptographic methods, symmetric methods such as AES (Advanced Encryption Standard, ISO/IEC 18033–3) [WP–AES],[2] ChaCha20 (by Bernstein) [WP–CCP][3] etc. are usually used for data encryption because they are much faster than asymmetric methods. In addition, many methods, such as AES-256 (AES with a 256-bit key), are also classified as quantum computer-secure. An assessment of the threat posed by future optical computers is still pending. Integrity and authenticity are usually solved with an electronic/digital signature, which comes from asymmetric cryptography. Today, ECDSA (Elliptic Curve Digital Signature Algorithm, ISO/IEC 14883–3) [WP–ECD][4] is mostly used for this purpose. If quantum computer security is desired, post-quantum cryptography methods must be used. For signatures, these are Crystals Dilithium, Falcon, or Sphincs+, and for asymmetric encryption and key exchange according to Diffie-Hellman (PKCS #3: Diffie-Hellman Key Agreement Standard), Crystals Kyber (ISO/IEC 27001:2022) [WP–Kyb],[5] because these methods have all been selected as quantum computer-secure by the NIST (US National Institute of Standards and Technology) [NIST25].[6] This means that no physical methods are required to achieve quantum computer security, as long as the non-provable security assessments are trusted.

2.3 Physical Cryptographic Methods

If these security assessments are not trusted, especially in cases of very high data security requirements, physical methods must be used for key generation and distribution, data encryption, and integrity and authenticity procedures. This ensures the confidentiality, integrity, and authenticity of the data against computationally powerful adversaries and mathematical attack methods that are still unknown (unpublished) today. These methods are discussed in more detail in the following Sects. 1.4 and 1.5.

All five physical cryptography methods/technologies discussed in the book—QKD in several technological variants, RKD, and MKD—only allow the generation and distribution of symmetric keys. Therefore, only symmetric cryptography

[2] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[3] https://en.wikipedia.org/wiki/ChaCha20-Poly1305.

[4] https://en.wikipedia.org/wiki/Elliptic_Curve_DSA.

[5] https://en.wikipedia.org/wiki/Kyber.

[6] https://csrc.nist.gov/pqc-standardization.

methods can be used with QKD, RKD, and MKD. Only symmetric methods can therefore be used to achieve the security objectives of confidentiality, integrity, and authenticity. These cryptographic methods are dependent on the physical methods QKD, RKD, and MKD because they only enable symmetric methods and because the key rates (measured in bits/second) enabled by the various methods and implementations of QKD, RKD, and MKD influence the selection of cryptographic methods.

As already stated above, it is always the case that, in order to ensure a consistently high level of security in terms of confidentiality, integrity, and authenticity, all mathematical cryptographic methods whose security cannot be strictly proven mathematically should be avoided. However, in the case of the physical methods QKD and RKD, mathematical methods must also be used for post-processing in addition to the physical method. These may include, for example, cascade error correction, low-density parity checks (LDPC), or secure sketch for error correction, and universal hash functions with a Toeplitz matrix and the use of a fast Fourier transform (FFT) for privacy amplification.

2.4 Generation of Cryptographic Keys

All mathematical methods for generating cryptographic keys used today are based on complex mathematical problems, such as the discrete logarithm in multiplicative groups or in elliptic curves. The security of post-quantum cryptography (PQC) is based on mathematical problems that are difficult to solve and, from today's perspective, cannot be solved efficiently even with quantum computers. These include, in particular, lattice problems, code-based problems, multivariate equation systems, and hash-based methods. In the case of lattice problems (lattice-based cryptography), these are primarily the shortest vector problem (SVP) and learning with errors (LWE). Code-based problems are based on the difficulty of decrypting linear error-correcting codes. The security of "multivariate quadratic equation systems" is based on the difficulty of solving systems of nonlinear equations over finite fields. Hash-based methods use the one-way properties of cryptographic hash functions.

The security of these methods is based on the assumption that, above certain key lengths, there are no efficient algorithms that can solve these mathematical problems within a reasonable amount of time. Assuming that Moore's Law for the increase in computer processing speed continues to apply, this increase requires that the keys used must be extended by approximately 13 additional bits every 10 years in order to maintain the same level of security. However, with the advent of powerful quantum computers or optical computers, the situation is changing dramatically and requires post-quantum cryptography. However, as this assumption has not yet been proven, these established and mathematically based methods are associated with a degree of uncertainty that is difficult to assess. It is possible that there are as yet unpublished efficient algorithms that can be used to break these mathematical methods.

2.4.1 QKD

Quantum key distribution (QKD; see Chap. 3) takes a fundamentally different approach. Unlike conventional methods, QKD is based on fundamental principles of quantum mechanics. These principles arise from physical laws, such as the fact that measurements on quantum states inevitably change them, or the fundamental impossibility of copying unknown quantum states. Of the QKD technologies discussed in this book, only the "QKD with entangled photons" variant itself provides the entire randomness of the key during key generation, and does so independently on both sides. (Both key exchange partners have identical roles.) In the other two QKD technologies, additional non-deterministic random number generators provide the randomness, and the two key exchange partners have different roles. One of the two is the sole generator of the randomness of the raw key, while the role of the other partner is limited to pure selection, without the possibility of influencing the randomness of the raw key itself. In practical use with independent communication partners, this is unfavorable in terms of security, primarily because a random number generator is used on only one side.

2.4.2 RKD

Another physical method is Radio Signal Key Distribution (RKD, see Chap. 4), where security is also not based on mathematical assumptions, but on fundamental principles of the generation and propagation of electromagnetic waves, in particular ultra-short waves (VHF). In contrast to QKD, which operates with light quanta, RKD uses radio signals above 30 MHz. RKD takes advantage of the unique physical properties of wireless radio channels. RKD is based on two fundamental properties of high-frequency transmission: the inherent unpredictability (randomness) of channel properties and the reciprocity of radio transmission between two communication partners. With RKD, two communication devices transmit radio signals in both directions at approximately the same time and continuously measure the received signal properties. Typically, the signal strength (RSSI–Received Signal Strength Indicator), the phase angle, or the transit time of the signals are recorded. The key principle lies in the reciprocity of the radio channel: since both devices use the same physical transmission path, they measure almost identical channel characteristics. These common measurements form the basis for generating identical cryptographic keys on both sides without having to exchange this information via a separate channel. As with QKD with entangled photons, RKD does not require an additional non-deterministic random number generator for key generation; instead, randomness arises independently on both sides from physics. However, like QKD, RKD also requires mathematical post-processing and mutual data exchange, which makes it much more difficult to assess the security of the overall solution. In contrast to QKD, however, RKD

is very cost-effective. QKD and, above all, RKD has a relatively low key rate (bits generated per second).

2.4.3 MKD

In MKD (Memory Key Distribution; see Chap. 5), the communication partners use a non-deterministic random number generator to generate the raw material in the form of very long bit sequences, which are later used as a shared key. This raw material is then stored on suitable, highly secure, transportable storage media—which are inexpensive and available on the global market with high security certification—and physically transported to the future communication partner. When storing data, all users with the same read rights receive the storage media containing the keys. In MKD, randomness comes from additional non-deterministic random number generators that are available on both sides. This means that in MKD, randomness is determined independently by both sides, as in RKD and QKD with entangled photons. Even a faulty or poor-quality random number generator—possibly caused by an attacker or the manufacturer—thus poses no problem, because after an XOR operation, the better of the two generators always determines the minimum quality of the key bits. There are no work steps that would be equivalent to post-processing or other mathematical procedures that would make it difficult to assess the security of the overall solution. Therefore, MKD enables completely math-free key generation and distribution. Because MKD also allows for very long keys (suitable storage media are now available up to 16 TB), data encryption with a one-time pad is also possible, which uses only an XOR operation and is provably absolutely secure.

2.5 Distribution of Cryptographic Keys

2.5.1 Within a Key Exchange Pair

In the case of QKD and RKD, the distribution of the generated key material within the pair that performs the joint key generation is an integral part of key generation and cannot be separated from it. Generating the key and ensuring that both parties receive the same key goes hand in hand with these two types of procedures.

With MKD, generation and distribution are separate steps, because here a key exchange partner generates the key, makes a copy of it, and then physically transports this copy to the other party. To further increase security, the other party can also generate its own key and send a copy to the first party. Each of them then combines the self-generated and received keys into a common key using an XOR operation.

2.5.2 Distribution to Multiple Communication Partners

All of the methods discussed in this book involve symmetric encryption, which means that two people who want to communicate with each other must first be provided with identical key material. This can be done through direct key exchange between these two people, but, as will be shown in the following chapters, there are physical distance limits, particularly with QKD and RKD, beyond which direct key exchange is impossible. If the distance between the two communication partners is too great, direct key exchange is not possible.

Trusted Nodes

In such cases, previously generated keys must be distributed via a specially created infrastructure, the essential components of which are trusted nodes. Trusted nodes are nodes in a network through which the key material can be exchanged. New key material is constantly being generated between neighboring trusted nodes, but the endpoints (i.e., the points that actually want to communicate with each other) also generate the key material in pairs with one or more trusted nodes. Some of the key material generated in this network is then used to secure the transport of other key materials through this trusted node network. In this way, the key material can also be transported across physical distance boundaries.

However, this method comes at a price: on the one hand, a significant portion of the key material generated is not used for the intended communication, but rather to secure key distribution. Many consider the need to trust trusted nodes to be a major disadvantage. This is because the trusted nodes' memory stores all the key material used for communication, making these devices desirable targets for attack and requiring them to be specially protected. The necessary maintenance also poses a major challenge here.

MKD

5As shown in the chapter on Multi-Key Distribution (MKD), there are no physical distance limits for MKD, which is why there is no need for trusted node networks with this method/technology.

2.6 Security Objectives

2.6.1 Security Objective: Confidentiality

Confidentiality means that protected information can only be read by authorized persons. In other words, the protection goal of confidentiality is considered to have been achieved for encrypted messages if it is impossible to read the plain text without the key.

There is a single encryption method for which the achievement of this protection goal can be proven mathematically. This is the one-time pad (see Chap. 6.2), which is based on the bitwise XOR operation between the plaintext and a key of equal length. An essential additional condition is that each key may only be used for a single message (prohibition of multiple use). A further additional condition is that the key must be truly random, i.e., it must originate from a non-deterministic source (for details, see Chap. 6.2 and https://cryptography.study/phys/XOR).

For all other known encryption methods, it can be proven that there must always be an algorithm that makes it possible to read the plaintext without a key. The simplest algorithm that always works (except for the one-time pad) is to systematically try all possible keys, i.e., a brute force attack.

It is therefore clear that the trust goal of confidentiality cannot be achieved in the strict formulation just used by any other known method other than the one-time pad. In practical application, however, a weakened definition of this protection goal is sufficient in many cases:

Practically achievable confidentiality means that protected information can only be read by unauthorized persons with unrealistically high effort and/or with an extremely low probability of success.

This definition means that the ability of an encryption method to achieve practically achievable confidentiality depends on the algorithmic complexity of the most efficient key-breaking algorithm. In other words, the algorithm that allows an attacker to read the plaintext with the least effort or the greatest probability of success without having to know the key determines the extent of practically achievable confidentiality.

With standardized encryption methods that correspond to the current state of the art, the effort an attacker must expend to obtain the plaintext with a predetermined probability which depends on the length of the key used and, in many cases, also on the type of computer used (classical computer or quantum computer). The most efficient known key-breaking algorithm is used as a basis in each case. The recommended key lengths then correspond to a time expenditure for the attacker that, when using reasonably realistic hardware, is equivalent to a multiple of the age of the universe, or the key lengths correspond to a probability of success that is equivalent to the probability that in an unmanipulated lottery over dozens or hundreds of rounds, exactly the same numbers will always be drawn at random in each draw.

However, there is a problem: practically achievable confidentiality depends on the most efficient algorithm possible. But standardizations necessarily only take known algorithms into account. What if there is a very efficient key-breaking algorithm for an encryption method that is still unknown but will become known tomorrow? Or worse, what if the very party from whom you most urgently want to keep your secrets has discovered a very efficient key-breaking algorithm and can use it without making it public?

It would therefore be very helpful if it could be proven for at least one encryption method that, for fundamental reasons, there cannot be any algorithm for this method whose efficiency exceeds a certain threshold. (In technical terms, this is referred to as algorithmic complexity rather than efficiency.) Experts have been trying to provide

such proof for decades, but have not yet been successful. The one-time pad is an exception to this.

There is a widespread belief among experts that such limits do indeed exist for many encryption methods, which is good news for the trust that can be placed in standardized encryption methods. But an opinion is not proof, and based on current knowledge, it cannot be completely ruled out that any standardized encryption method could soon become ineffective because someone has found an efficient way to break it (with the exception of the one-time pad).

The only method that is completely exempted from all these considerations is the one-time pad,[7] for which the achievement of this protection goal can be proven mathematically and is therefore an important data encryption method for MKD.

2.6.2 Security Objective: Integrity

Integrity means guaranteeing that the sender receives exactly the data that the sender sent. Integrity therefore means that it is impossible to manipulate the data during transmission or while it is stored. This goal is achieved by adding checksums and authenticating the sender to the recipient.

The physical cryptography methods currently available only allow symmetric cryptography, but not asymmetric cryptography. Therefore, in addition to data encryption, MACs (message authentication codes for integrity assurance, ISO/IEC 9797–2) [WP–MAC][8] are also possible, which allow integrity and authenticity checks between two communication partners.

Integrity can be easily implemented with a high level of security using a symmetric method. For example, CCM-MAC (Counter with CBC-MAC, specified in RFC 3610), CBC-MAC (Cipher Block Chaining with MAC, ISO/IEC 10116) and GCM-MAC (Galois Counter Mode with MAC, ISO/IEC 13157–3) use the XOR function (see Chap. 6) for MAC calculation. In conjunction with the one-time pad for encryption, these MACs allow for provably absolute integrity security.

2.6.3 Security Goal: Authenticity

Authenticity means the verifiable certainty that a received message actually originates from the claimed sender and has not been generated or falsified by an unauthorized third party.

Asymmetric cryptography can be used to generate digitally signed certificates that allow a person, device, etc. to be assigned to a public key, which in turn can

[7] See Sect. 6.2One-Time Pad.

[8] https://en.wikipedia.org/wiki/Message_authentication_code.

be used to verify any digital signature. Without asymmetric cryptography—as is the case with physical methods—this elegant solution is not possible.

Therefore, physical methods and thus protection against computationally very strong opponents and mathematical attack methods that are still unknown (unpublished) today cannot be used even for blockchains and cryptocurrencies such as Bitcoin, whereby Bitcoin, for example, has not even taken the step towards post-quantum cryptography and thus protection against powerful quantum computers with mathematical methods.

All key exchange procedures discussed in this book generate keys that are intended for symmetric encryption methods. When these keys are used, a symmetric shared secret is usually employed. In conjunction with methods such as the Wegman-Carter method [Weg81],[9] it is possible to ensure not only the integrity of the transmitted data, but also the authenticity of the sending communication partner. However, the shared secret must be exchanged in pairs between all possible communication partners so that each communication partner can reliably identify their counterpart. With n communication partners, this means $n * (n - 1)/2$ shared secrets (e.g., $n = 100$, resulting in 4950 shared secrets). This number can be significantly reduced by forming groups in which several communication partners are assigned to groups. However, this means that it is no longer possible to assign a communication partner directly, but only to assign them to a group. Another option is star-shaped topologies with central communication nodes. Each party communicates directly only with this central server, which forwards all received messages to all participants. This means that with n communication partners, only n shared secrets need to be exchanged. However, each member of this group must then trust that the central node actually verifies the authenticity of all other group participants.

This must be distinguished from key generation. In all the methods presented in the book, key generation takes place in pairs, i.e., between exactly two parties. To ensure authenticity during key generation in the QKD and RKD methods, the Wegman-Carter method (or another equivalent method) is also used. In the case of MKD, the key exchange involves the physical transfer of a data carrier, which enables the use of authentication methods from the physical world, such as high-security smart cards, etc., in conjunction with PIN (personal identification number) or biometric features (two-factor authentication).

2.7 Common Communication Roles (Alice, Bob, Eve, Mallory)

In cryptography, in the field of network protocols, and also in subdisciplines of physics, it is common to talk about communication partners with specific roles or characteristics. Over the past few decades, a few names for recurring roles

[9] https://doi.org/10.1016/0022-0000(81)90033-7.

have become established in international specialist literature and have now become "standard names" [WP–A&B].[10] Four of these names are also used in this book:

2.7.1 Alice and Bob

These are the names of the two legitimate communication partners. These two parties do not want other parties to gain possession of information that is the subject of legitimate communication. The names *Alice* and *Bob* are derived from the first two letters of the alphabet, but otherwise have no special meaning.

In many communication protocols, Alice and Bob are not exactly the same. If one of them actively initiates the communication (e.g., by dialing a phone number) and the other responds to this action (e.g., by picking up the phone or accepting the call), then the party that acts first or starts sending is called "Alice" (because A is the first letter of the alphabet). The other legitimate party is then called "Bob."

In the field of quantum key distribution in particular, Alice initiates communication in two of the three methods described. In the third method (QKD with entanglement), in RKD and MKD, the roles of the two communication partners are exactly the same. In this case, it does not matter which of the two has which name. However, the names are still "Alice" and "Bob."

2.7.2 Eve and Mallory

The English verb *"to eavesdrop*" means to *listen in on a conversation secretly*. The interception of other people's messages is therefore referred to as *eavesdropping* in technical jargon. Due to the phonetic similarity of the first syllable of these terms to the English first name "Eve," this name has become established as the name for the party attempting to eavesdrop on the communication between Alice and Bob. Eve expressly does not attempt to actively interfere with the communication.

A malicious attacker who intends to alter the content of the transmitted information, inject new information, or block information is given the name *Mallory* (from *malicious attacker*) in this international quasi-standard nomenclature.

In the case of QKD protocols (see Chap. 3), however, a key feature of these protocols is that Eve unintentionally alters information or prevents it from reaching the recipient by eavesdropping. But this is typical of attempts to eavesdrop on quantum information. These changes are not intentional on Eve's part. On the contrary, she would actually prefer it if this did not happen, because her eavesdropping activities are reliably detected precisely by these changes. Due to the lack of intent to manipulate, it is therefore not justified to use the name *Mallory* for someone who unintentionally leaves traces by eavesdropping on the quantum channel. Therefore,

[10] https://en.wikipedia.org/wiki/Alice_and_Bob.

in the world of QKD protocols, the name *Eve* is always used for a party whose goal is to passively eavesdrop.

References

[Bern18] Bernstein, D.J., Is the Security of Quantum Cryptography Guaranteed by the Laws of Physics? arXiv preprint, (2018). https://doi.org/10.48550/arXiv.1803.04520

[WP-AES] Wikipedia contributors, Advanced Encryption Standard Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[WP-CCP] Wikipedia contributors, ChaCha20-Poly1305. Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/ChaCha20-Poly1305

[WP-ECD] Wikipedia contributors, Elliptic Curve Digital Signature Algorithm. Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Elliptic_Curve_DSA

[WP-Kyb] Wikipedia contributors, Kyber. Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Kyber

[NIST25] National Institute of Standards and Technology (NIST); Computer Security Resource Center (CSRC), PQC Standardization Process (Post-Quantum Cryptography). Created 03 Jan 2017; updated 11 Dec 2025. https://csrc.nist.gov/pqc-standardization

[WP-MAC] Wikipedia contributors, Message authentication code. Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Message_authentication_code

[Weg81] M.N. Wegman, L. Carter, New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**, 265–279 (1981). https://doi.org/10.1016/0022-0000(81)90033-7

[WP-A&B] Wikipedia contributors, Alice and Bob. Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Alice_and_Bob