Universidade do Minho
Escola de Engenharia

Manuel Tiago Carvalho Silva

The future of contemporary cryptography in Quantum environments. New technologies and algorithms (QKD) applied to industry

Dissertação de Mestrado
Mestrado em Engenharia e Gestão de Sistemas de Informação

Trabalho realizado sob a supervisão de:
Prof. Doutor Henrique Santos

February 2024

Table of contents

1 Context and Motivation

The exponential development of computers and technology has increased the amount of information exchanged, raising security concerns. Data encryption has been one of the most widely used methods for hundreds of years, and its use has allowed information to be exchanged more securely (Lin, 2014; Gupta, Gupta, & Singhal, 2014).

Cryptography methods have enabled society to evolve ways of exchanging sensitive information, financial transactions, communications, and personal data protection (P., Shari, & Pfleeger, 2015). Despite the constant evolution of personal computers, the arrival of quantum computing remains a possibility. The threat of standardized and widely accepted cryptography being broken is becoming more real. It is critical to comprehend the implications of these machines for traditional cryptography (Lakshmi & Murali, 2018).

Quantum key distribution (QKD) has emerged as a promising alternative, based on the principles of quantum mechanics, to guarantee secure communications (Tsai, Yang, Lin, Chang, & Chang, 2021). Thus, research into the future of contemporary cryptography in quantum environments, with a particular focus on new QKD technologies and algorithms, will promote a more secure and reliable approach to communication and data protection in a world that is increasingly dependent on technology (Alleaume, 2005).

2 Goals and expected results

In this dissertation, based on the principles of quantum mechanics and contemporary cryptography, particularly quantum cryptography algorithms and modern cryptography algorithms, the aim is to suggest a new way of transmitting cryptographic keys, making them more secure and combining the benefits of both worlds.

Thus, the expected results of this dissertation include:

- The integration of quantum algorithms into contemporary cryptography, the investigation and evaluation of different existing QKD protocols, including their advantages and limitations, and the proposal of solutions to circumvent the limitations of current cryptography.

- It is hoped that this dissertation will contribute to the study of secure and efficient communication channels for the exchange of cryptographic keys, thus promoting more excellent reliability and privacy in digital communications, meeting the growing demands for cyber security in an increasingly interconnected world. In addition, this research could also open the door to future advances in the field of cryptography, enabling new applications in critical sectors such as finance, health, and military communications.

3 State of Art

3.1 General principles of contemporary encryption

Classical encryption is essential in evolving methods that allow human beings to hide sensitive information. The constant evolution of personal machines has made it imperative to create more robust methods and mechanisms to increase the security of information exchange. Firstly, it is necessary to understand what encryption is.

Encryption is a method that is used to securely exchange information on an unsecured internet network or data storage site. Before the concept of a public key was created, for two individuals to communicate, a secret key exchange had to be established as a priory between the communication agents (Boneh, Sahai, & Waters, 2011).

In addition to the definition of encryption that has been presented, some associated concepts need to be clear and presented. NIST presented these concepts in the Computer Security Handbook [NIST95] in which, within the definition of computer security, three main objectives need to be guaranteed (Nieles, Dempsey, & Pillitteri, 1995; Stalling, 1989).

In Figure 3.1, adapted from (Irwin, 2023), we see a concept called the CIA triad note. According to (Irwin, 2023), it is an important concept used in information security management and ISO 27001. The following components, confidentiality, integrity, and availability, will be analyzed. In a more general initial analysis, each element of the triad is interconnected with the others. When ensuring the protection of one component, the ramifications that may arise in the others should be considered.

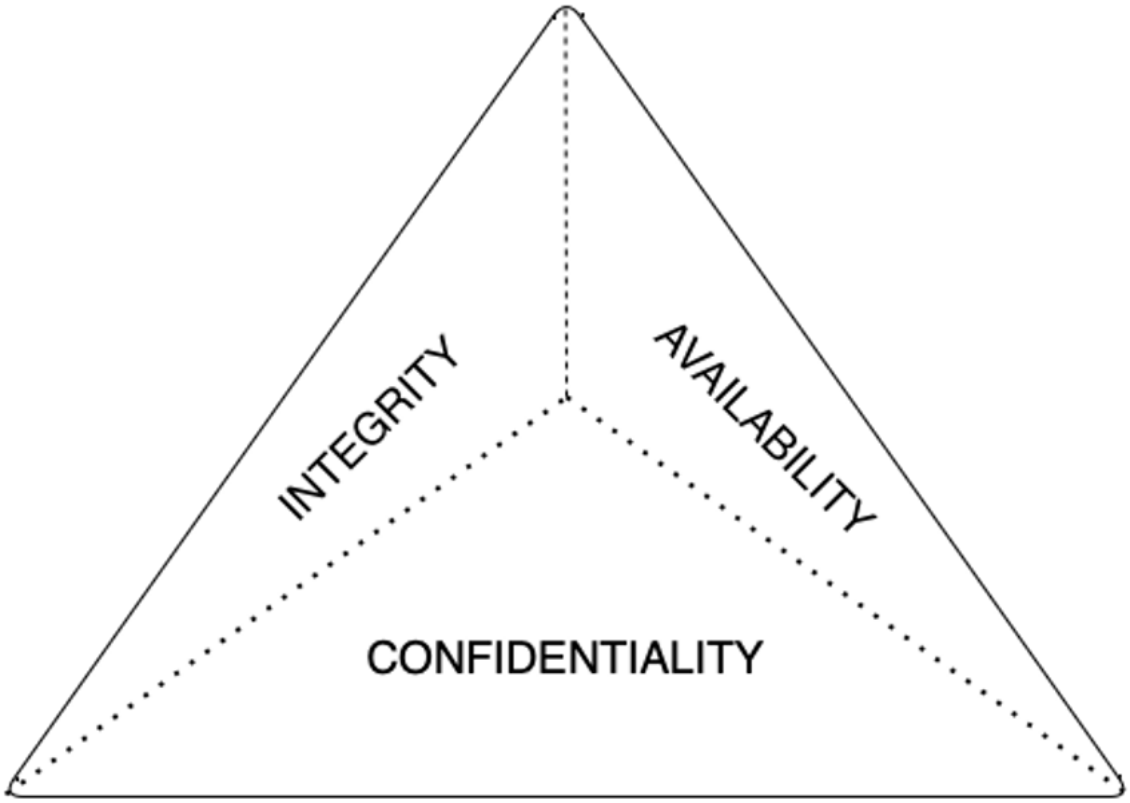

Figure 3.1: CIA-TRIAD

Confidentiality

Starting with the first concept/objective, confidentiality is one of the most easily understood concepts. Confidentiality means that only authorized people or systems can access protected data (P. et al., 2015). It's the easiest concept to understand since we consider what we should and shouldn't hide from the world. Although straightforward for (Stalling, 1989), it is the same; confidentiality also covers two other related concepts: Data confidentiality and privacy. Data confidentiality means that private or confidential information is not made available or accessible to unauthorized individuals; privacy means that individuals can control and decide what information can be seen and stored and by whom and to whom it can be made available (Stalling, 1989).

Although it's a concept that's easily understood and quickly linked to data, we can relate it to various aspects of the world, from an invention to the search for a known criminal. As it is important to guarantee the confidentiality of data, some aspects must be avoided. Otherwise, this property will not be guaranteed:

- An unauthorized individual accesses a data set;
- A program or process accesses a data set;

- A person is accessing data they are authorized to see but also accessing unauthorized data. So, it ends up being a more specific version of the first point;
- An unauthorized person accesses a certain amount of data, albeit approximately. For example, knowing the range of values in which a salary can be set;
- An unauthorized person learns about the development of new technology or even about business deals between companies (P. et al., 2015).

To avoid losing confidentiality, one of the strategies that can be used is using a VPN (virtual private network). On the other hand, using a secure data transfer mechanism can help safeguard confidentiality (Arogundade, 2023).

Integrity

Integrity guarantees protection against the modification or even destruction of information. This concept includes the guarantee of non-repudiation and authenticity of the information (Stalling, 1989). This guarantee must be given through the ability to detect data manipulation by third parties who are not authorized. Data manipulation can include inserting, deleting, or replacing data (Menezes, Oorschot, & Vanstone, 1996).

The correct and harmonized operation of the software and hardware guarantees the integrity of the data. With the evolution of processing capacity, the emergence of large-scale systems, and the ability to process more than one type of data using the same equipment, it has become essential to guarantee this integrity (Mayfield, Roskos, Welke, & Boone, 1991).

Therefore, some objectives must be taken into account when guaranteeing integrity:

- Preventing unauthorized users from making changes - This goes along with what the author (Stalling, 1989) says, since preventing unauthorized users from altering data or even system resources thus allows data integrity;
- Maintaining external and internal integrity - This relates to data and systems. The authenticity of data and its consistency with the real world are interdependent. With distributed systems, complexity has increased, and there is a need to ensure that data is integrated and consistent regardless of the machine.

- Preventing authorized users from altering data - There are risk reduction procedures instead of full system checks to prevent these events. As well as being an ethical issue, it is impossible to provide complete integrity in this respect. Therefore, Existing procedures make it possible to minimize the risk of this breach of integrity (Mayfield et al., 1991).

For (Mayfield et al., 1991), integrity is context-dependent and can have numerous meanings depending on the context and purpose to which it will be applied. Despite this interpretation, it should have authorized actions, resource separation and protection, error detection, and correction.

Encryption mechanisms play a crucial role in ensuring mechanisms provide an accurate and intact data state. This mechanism should prevent or detect if the message has been corrupted. Just like in confidentiality, the precision of control over who or what can access, to what resources, and in what ways must be respected (P. et al., 2015).

Availability

For (Hoffman & Zahadat, 2018), availability means that information can only be accessed by the appropriate parties. While this information may be important, there is no point in guaranteeing its security if it cannot be guaranteed.

As with the concepts presented above, such as integrity, availability can mean different things in different contexts. According to (P. et al., 2015), we can define certain criteria to define availability:

- There is a timely response to our request;
- Resources are allocated fairly to ensure that requesters are not favored over others;
- Competition is controlled to ensure that deadlocks are managed and access is supported as required;
- Competition is controlled to ensure that deadlocks are managed and access is supported as required;
- The system must be designed to cope with hardware and software failures to avoid abrupt data loss and prevent crashes. If an outage has to occur, it must be warned to avoid unwanted losses;
- The system can be used as intended. Failure to do so can be considered a failure of system availability.

According to (Arogundade, 2023), to ensure the availability of systems and, consequently, their networks and data, regular hardware and software updates should be performed, along with having backup plans and strategies for failure in the event of attacks. These mechanisms can reduce the impact and enable continuous information availability, ensuring its definition is met.

Authenticity

Using the CIA triad allows for a more precise definition of security objectives. However, for some authors, it is not sufficient, and for this purpose, two additional concepts are presented, which allow for a more comprehensive framework.

According to (Stalling, 1989), authenticity is the property of being genuine and verifiable. Through cryptographic methods such as hash functions, symmetric keys, and asymmetric keys, it is possible to verify the author's authenticity of a particular message (Agarwal et al., 2019).

Non-Repudiation

The goal of ensuring non-repudiation is to provide the ability to generate, collect, maintain, make available, and verify evidence related to a specific event. Verifying the evidence allows for the resolution of potential disputes regarding the occurrence or non-occurrence of a particular event or action.

Two types of evidence aid in verification. The main difference between them is the encryption technique used:

- Secure envelopes generated by an evidence-generating authority using symmetric cryptographic techniques;
- Digital signatures generated by an evidence-generating authority but using asymmetric cryptographic techniques (Tsohou, Kokolakis, Lambrinoudakis, & Gritzalis, 2010).

3.2 Hash functions

One of the best-known cryptographic functions is the hash function. A hash function is a mathematical computation that takes a certain amount of data at random as input and produces an output with a fixed size. These inputs are usually called messages, while the outputs are called message digests.

All hash functions have a property that allows them to guarantee that it is impossible to determine the input from the output. Thus, any attacker has no means of going back to the beginning of the process by making hash functions one-way functions. It thus becomes more difficult, not impossible, to determine the input from the output.

Hash functions are a wide range of functions, some of which are more powerful than others. These include cryptographic hash functions. These have another property in that two messages are unlikely to produce the same message digest. If this happens, it's called a collision. In the most modern hash functions, it is extremely difficult to create a collision, which doesn't even make methods for creating them known.

These functions are normally used to guarantee data integrity. Since they generate unique messages, it is possible to create a basis for comparison. If the summary that has been calculated differs from the base, it indicates that the file has been altered. This method is not very common and is very reliable, but it cannot be emphasised enough that it only conveys that the message has been altered and not what has been altered.

These hash functions don't use secret keys to work. To detect the changes mentioned above, they use MDCs (modification detector code). To ensure the authenticity of the data at the source, as well as data integrity, MACs (message authentication codes) are used, in which a secret key is used (Edwards, 2003; Menezes et al., 1996).

The advantages of hash functions include

- Deterministic, since an input message must always produce the same hash value
- Fast to compute
- Guarantees data integrity
- Difficult to analyse since a small change produces a completely different result
- No collisions

Limitations include:

- They are irreversible
- Although they are extremely rare and difficult to occur, there can be collisions
- They are not designed to guarantee confidentiality (Nakov, 2018)

3.3 Cipher algorithms

Symmetric keys

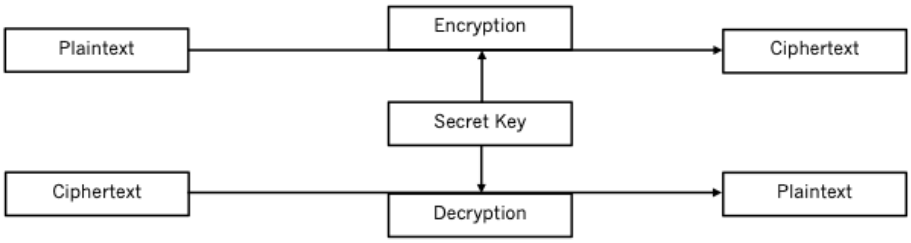

Figure 3.2: Symmetric key Process

Within classical cryptography, there are two categories of encryption, symmetric and asymmetric. The type of key used to encrypt and decrypt is the difference between the two (N. Kumar, Thakur, & Kalia, 2011). Both the sender and receiver have to agree on a secret key, which is shared. Given a message called plaintext, and using the key, the encryption algorithm produces an unreadable data set the same size as the original text. Decryption, on the other hand, is the reverse of encryption but uses the same key (N. Kumar et al., 2011). Several examples of algorithms use symmetric keys, such as DES, 3DES, AES, Blowfish, and so on. Although these are algorithms and ways of encrypting information used by society, they have advantages and limitations. The advantages of these algorithms include:

- They are designed to support large amounts of data. There are now hardware implementations that can encrypt hundreds of megabytes per second, while software implementations can only handle small amounts of data
- The keys used in symmetric key algorithms are small
- These algorithms can be used as primitives for the construction of cryptographic mechanisms, such as random number generation, digital signature schemes, the construction of more robust ciphers through transformations.

The disadvantages include:

- As the key used is the same for encryption and decryption, it must be shared by both the sender and the receiver
- In a network where the number of keys that need to be shared is large, a third party is needed to manage these keys
- The key must be constantly changed when a new communication is made
- The digital signature mechanisms that arise through this cipher mechanism require either large keys or a key management mechanism (a third element) in the communication (Yadav, Lal, & Arora, 2010)

Asymmetric keys

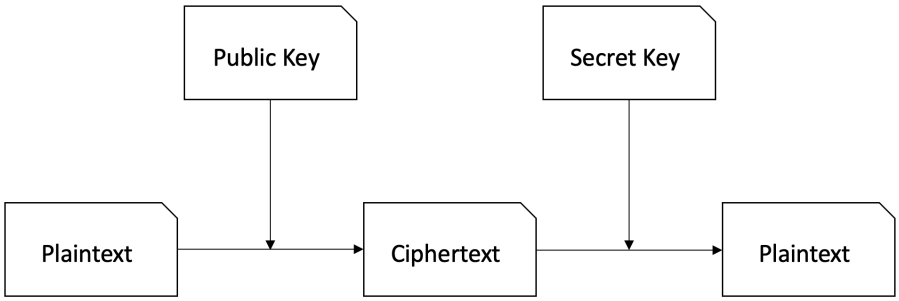

Figure 3.3: Asymmetric key Process

In contrast to a symmetric key, this mechanism eliminates the need for a "courier" to deliver the key in a secure channel before the intended message is sent. Encryption keys are public in asymmetric keys, while decryption keys are private. Therefore, only the person with the decryption key can view the content of the message. It's important to note that the public key must be designed so that the private key is not easily provable (Milanov, 2009). Several examples of algorithms use asymmetric keys, such as RSA, Diffie-Helman, DSA, and so on. Like symmetric cipher algorithms, asymmetric cipher algorithms are no exception and have advantages and limitations.

Advantages include:

- Only the symmetric key must be kept secret, and the authenticity of the public key must be guaranteed

- Only a credible third party that is functional and can operate in offline mode is needed to manage the keys on the network
- Depending on the world of use, the private and public keys must remain unchanged for longer periods
- Digital signature mechanisms are more efficient since the verification function of the public key is much smaller compared to the one used with the symmetric key implementation
- In a network with a larger number of hosts, the number of keys required is smaller than in the case of symmetric keys

The disadvantages include:

- Asymmetric key methods are considerably smaller than symmetric key methods
- The key size is considerably larger for both encryption and digital signature mechanisms
- Public key schemes are not demonstrably secure. Their security is based on numbers, i.e. theoretical problems
- Their historical legacy is not as great as that of symmetric cryptography (Yadav et al., 2010)

To summarise, we note that, according to (Joseph, Krishna, & Arun, 2015), a larger input block leads to slower execution of the encryption algorithm. Similarly, asymmetric encryption algorithms run slower than symmetric encryption algorithms, which can be a limitation when time is a priority. The size of the key is one of the points to consider when choosing the method that will be used to hide information. The keys used in symmetric key algorithms are smaller than those used in asymmetric key algorithms for the same level of security. Symmetric key algorithms produce outputs equal to or smaller than the original, while asymmetric key algorithms always make outputs larger. Regarding security, the evolution of symmetric key algorithms has been more notable, with AES being one of the most secure. As far as asymmetric key algorithms are concerned, although security is high, algorithms such as RSA need large keys (1024 bits) to guarantee a high level of protection; Diffie-Helman, on the other hand, is subject to man-in-the-middle attacks due to the exchange of the key generated through the use of an insecure channel. Although they are used, brute force or cryptanalysis attacks commonly destroy symmetric key algorithms. Asymmetric key algorithms are the target of oracle attacks and brute force attacks. The consumption and cost of generating the key are lower for symmetric keys than for asymmetric keys (Yadav et al., 2010; Joseph et

al., 2015; Bisht & Singh, 2007). The following table summarises the main differences between symmetric and asymmetric key algorithms.

Table 3.1: Asymmetric vs Symmetric key

PARAMETER	Algorithms	
	Asymmetric	Symmetric
Time consumption	High	Low
Key Length	High	Low
Power Consumption	High	Low
Security	High	High
Cost	Expensive	Expensive
Implementation	Simple	Complex
Flexibility	No	Yes
Resultant size	Greater than original text	Less or equal to original text
Vulnerabilities	Oracle attacks and collisions	Brute-force and Cryptanalysis
Possible keys	2^{128} or 2^{512}	2^{56} or 2^{128} or 2^{256}

3.3.1 Key Generation

Key generation is a basic subject in the realm of information security that will be covered in this part. Keys are essential for protecting sensitive data and guaranteeing the confidentiality, integrity, and validity of information in the context of cryptography and cybersecurity. Since keys are the building blocks of both encryption and decryption, their proper generation is necessary to build reliable and secure systems. We will examine the fundamental ideas behind key generation in this context, as well as the methods used to guarantee their robustness and the usefulness of this approach in protecting data from online attacks.

PRNG

The technique of creating random numbers relies on modular arithmetic to function. Every process like this begins with a seed. The input value directly influences the generated number sequence. The PRNG produces the same sequence of integers when the same input is supplied. With the help of the deterministic and recursive generation process, sequences can be created with just software. This means that

the seed's unpredictability is restored by entropy and irreproducibility, which is more likely to occur. High key generation speed is possible with this approach. It can benefit from quicker communications due to its affordability and ease of installation. Inadequate building practices may result in weaknesses in important high-key generation speed, which is possible with this approach. Because generation is periodic, poor construction can result in vulnerabilities in essential reconstruction or even system collapse. The key needs to be higher than 2^{128} to resist different types of attacks. It is possible to construct the PRNG with 8,16,32,64,128 bits of random numbers (Fathi-Vajargah, Asghari, & Vajargah, 2016; Márton, Suciu, Săcărea, & Creț, 2012; Patnala et al., 2020).

Prime Factorization

One of the techniques used in key generation, especially in asymmetric key mechanisms, is the factorization of prime numbers. The resistance of the key is mainly based on the resistance of the key to factorizing the two prime numbers that give rise to it (Rutkowski & Houghten, 2020).

ECC

An elliptic curve (ECC)is defined mathematically by $y^2 = x^3 + ax + b$, where a and b are parameters of the curve. The choice of these parameters is crucial for the security of the system. As an equation, choosing an initial point called the generator point (G) is necessary. In simple terms, the Elliptic Curve Cryptography (ECC) encryption method is based on scalar multiplication, meaning a point is added to itself several times according to a secret number, which is the private key.

Compared to RSA, which uses prime number factorization, its main advantage is that the key needed to guarantee the same security is significantly smaller. A 256 bit key generated using ECC has a 3072 bit key as its equivalent in RSA.

To obtain a public key in ECC, parameters must be input. These parameters will be introduced into the equation of the elliptic curve. Given the equation, the parameters a and b are integers. The q will be a sufficiently large integer prime number in 2^m. The generating point (G) is a point on the curve whose order is greater than N, where N is the minimum number of times that adding the point to itself returns it to the starting point. The public key must be chosen with smaller sums (n) than the minimum number of times. To calculate the public key, a product must be made between the generating point and the number chosen previously (n).

Hash

Focusing on the study of the most commonly used hashing algorithms, SHA-256 and MD5 stand out. SHA-256 is a cryptographic hash algorithm used in digital certificates to ensure data integrity. This algorithm takes a randomly sized message as input, which must be smaller than 2^{64}, and produces a 256-bit output that can be used in symmetric keys algorithms (Ambedkar, Bharti, & Husain, 2022).

On the other hand, MD5 (Message digest) is an algorithm that takes an arbitrary-sized input message and produces a 128-bit output, half of what SHA-256 generates.

To generate the hash key using MD5, four 32-bit variables need to be initialized. The original message is padded so that its length is congruent to 448 mod 512. To achieve this, padding bits are added, including a "1," followed by zeros, and then the original length of the message is added. The modified message is divided into 512-bit blocks, and each 32-bit block (16 blocks) is processed individually. Each block undergoes mathematical operations, which generate the hash when concatenated at the end.

To obtain a hash generated through the SHA-256 algorithm, eight 32-bit variables (H0 to H7) are initialized with defined values (448 mod 512). Similar to MD5, it is necessary to add padding bits and append the original length of the message. The message is divided into 512-bit blocks (64 bits each). Each block is further divided into 16 words of 32 bits. Forty-eight additional words are generated through non-linear functions for the calculation. Then, the variables are updated, and finally, all values are combined, resulting in the final 256-bit hash (Rachmawati, Tarigan, & Ginting, 2018).

Comparison

Generating cryptographic keys is crucial for the security of systems, and different mechanisms have different efficiencies. The Pseudo-Random Number Generator (PRNG) stands out for its fast production of pseudo-random numbers, and the algorithm's and hardware's efficiency influences its speed.

Elliptic Curve Cryptography (ECC) offers an efficient alternative, employing scalar multiplication on elliptic curves. Compared to traditional methods such as Prime Factorisation (used in RSA), ECC provides faster generation times and is notable for guaranteeing the same security with significantly smaller keys. ECC with a 256-bit key is highlighted as an efficient and secure option compared to PRNG, Prime Factorisation, and hashing (SHA-256 and MD5). The choice of secure key size depends on the algorithm, with 256 bits being recommended for ECC, while PRNG should guarantee keys greater than 2^{128}.

Prime Factorisation, although secure, can demand more computing resources, potentially making its key generation more time-consuming, especially as the key size increases. Meanwhile, hashing algorithms

such as SHA-256 and MD5 vary in hash generation time, depending on the specific implementation and the message size. SHA-256 is favored for its more robust security, even if it requires slightly more extensive processing than MD5.

3.3.2 Key management

Secure key generation is one of the most crucial aspects of ensuring the security of exchanged data and preventing unauthorized access. In the management of keys through symmetric key mechanisms, as described by (Menezes et al., 1996), the involvement of a trusted third-party entity is necessary. Each entity communicates with this third-party entity (TTP), which generates a session key. If two entities wish to share, the TTP generates a key, encrypts it, and sends it to each party. This method has advantages such as ease of adding or removing entities from the network, with each entity only needing to store the long-term symmetric key. However, its disadvantages include the necessity for communications to go through the TTP, and the TTP must store multiple long-term keys. If the TTP is compromised, all communications are at risk.

In the case of public key mechanisms, each entity possesses a private and public key. The public key and the entity's identity are stored in a central repository called the shared file. If an entity wants to communicate with another, it uses the recipient's public key stored in the public file to encrypt and send the message. The advantages of this mechanism include the absence of a third-party entity (TTP), flexibility in the location of the public file, and the need to store only a limited number of public keys for secure communication. It assumes that a passive attacker conducts any potential attack.

Although the secrecy of the key is important, it is of greater importance in asymmetric key mechanisms since the file containing the public keys can be altered. To get around these problems, there are mechanisms to check whether the message has been changed and to guarantee its authenticity and integrity. These are widely used when public-key mechanisms are utilized since public keys are more prone to attack.

Digital Signatures

A digital signature is similar to a pen signature on paper. The difference is that digital signatures use a public key. As with pen and paper, this signature assigns an identity to the person signing a document and records a commitment to it. As with a conventional signature, it is impossible to forge a digital signature (Patel et al., 2019).

A digital signature is associated with two processes: signing and encryption and decryption and validation. Firstly, a hash is used on a small portion to guarantee the integrity of the message. Next, this code (message digest) is encrypted with the sender's private key. This step ensures non-repudiation since if the original message is to be obtained, it is decrypted with the receiver's public key. Next, the signature message, the message, and the sender's public key are sent in a Packet. This packet is encrypted with the receiver's public key.

On the other side, this pack is decrypted with the receiver's private key. The message received is the input for the hash function to compute the message digest that guarantees the integrity of the message on the receiver's side. The signature message is then decrypted with the sender's public key. Finally, the message digest is obtained after decrypting the signature and computing the message digest from the original message. This guarantees integrity, authenticity, and non-repudiation (Kaur & Kaur, 2012).

Digital Certificates

A digital certificate allows users to authenticate themselves. Instead of asking everyone in an application to show themselves, there is authentication by a third party that uses digital certificates to enable this authentication. These certificates serve two purposes: to establish the identity of the person holding the certificate and to distribute the certificate's public key.

As introduced above, these certificates are managed by duly accredited third parties called certificate authorities (CA). Depending on the application requirements or the problem in question, these authorities can be commercial or even local. The CA is responsible for authenticating users before issuing a certificate. These certificates, in turn, have a digital signature. When a user presents a certificate, it is validated by the digital signature. If it is validated, it is authentic and intact. Although validated, it means that the CA has authenticated the user in question and that trust in the CA is guaranteed. Within a digital certificate, there are various pieces of information, including:

- The user's name
- The public key
- The date of issue
- The expiry date
- The name of the certification body

- The digital signature of the certification body (IBM, 2023)

There are, however, validations and revocations of certificates. The certificate's validity can be requested each time it is used from the CA or a validity period can be applied. Associated with certificate validity is certificate revocation. Certificate revocation should be used when there are suspicions that the certificate has been compromised. If the certificate is validated online, the CA can check its validity. If it is offline, there are Certificate Revocation Lists (CRLs). These lists contain certificates that have been revoked before their expiry date. This can happen if the key shown on the certificate has been compromised or the user no longer has the authority to use the same key (Kaur & Kaur, 2012).

We can conclude that the use of certificates has advantages and disadvantages. The benefits include privacy, ease of use, low cost, and flexibility. The limitations are security since they can be the target of attacks, performance and time lost in authentication, integration, since it is complicated to do so, and their management, which can become costly (Shacklett, 2021).

Diffie-Hellman

The Diffie-Hellman key exchange algorithm was the first public-key cryptographic algorithm invented in 1976. Its enduring security, allowing it to remain one of the most widely used algorithms, stems from its difficulty in calculating discrete logarithms in a finite field. In contrast, the computation of exponentials is exceedingly straightforward.

This algorithm is employed for a symmetric key distribution, enabling two parties to generate a shared secret key. However, even though they can achieve key generation through this process, it is not feasible to encrypt or decrypt using the key they both derived.

From a mathematical perspective, generating the secret key is straightforward. The parties must agree on a large prime number (n) and another number, the primitive root modulo, the initially agreed-upon modulus(g). These two chosen numbers do not necessarily have to be kept secret, and their agreement can be achieved through an insecure channel or even a channel with other users without any impact.

- The first participant chooses a large random number (X) and sends it to the other $X = g^x mod^1 n.$
- The second participant chooses another large random number (Y) and sends it to the other $Y = g^y mod n.$

After both numbers have been chosen and exchanged, the first and second parties will compute the numbers.

[1] mod = module

- The first participant performs their operation: $k = Y^x mod n$. Similarly, the second participant computes $k' = X^y mod n$.

The variables k and k' equal $g^x y mod n$. Anyone eavesdropping on the channel cannot compute this value since they only know n, g, X, and Y. Unless they calculate a discrete logarithm and recover x or y, intruders cannot solve the problem. Therefore, we can conclude that k is the secret key generated by both participants.

Despite being a more secure way of key generation, the choice of g and n can significantly impact the system's security. Two aspects should be considered: n is a large number, and $(n-1)/2$ is a prime number. It is worth emphasizing that security relies on the size of the chosen prime number (Menezes et al., 1996).

One of the advantages of this key generation mechanism is that secret keys are only created when needed. There is no need to store them in a location where, over time, they may be subject to more vulnerabilities. For parameter exchange, no pre-existing infrastructure is required other than the agreement between the two parties.

Despite being a secure method, it has weaknesses. For example, it provides no information about the parties' identity, opening the door to potential attacks. It has a significant computational requirement, making it vulnerable to a clogging attack, where an entity may request many keys, requiring more effort than necessary. Additionally, it cannot prevent a replay attack and is susceptible to a man-in-the-middle attack (Li, 2010). The following figure, which represents the Diffie-Hellman process, was adapted from (Shor, 1999).

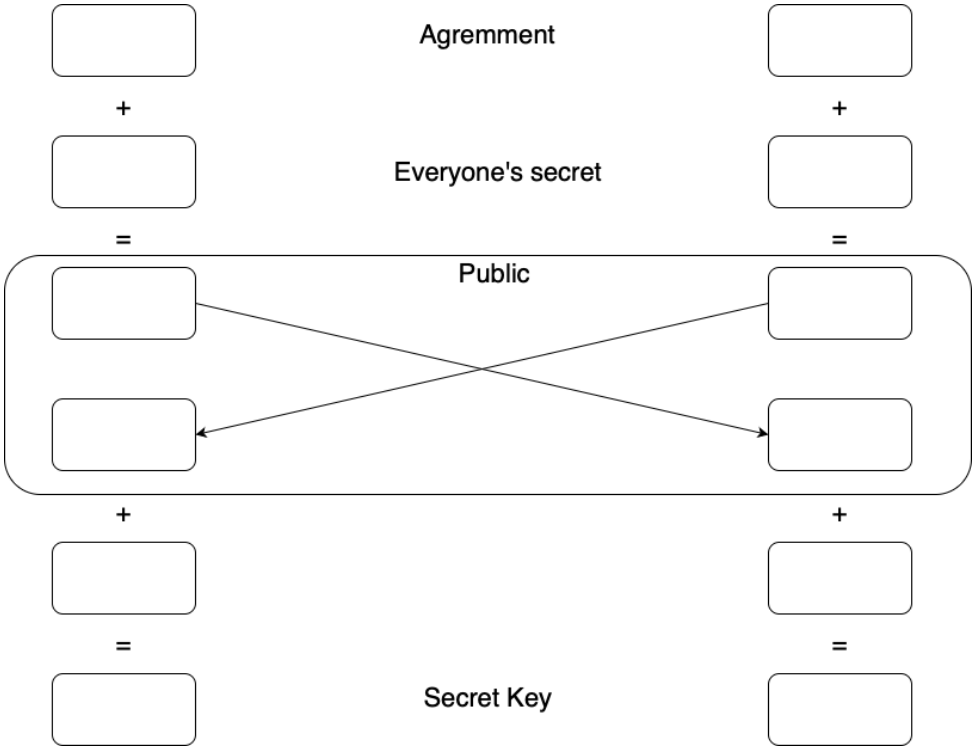

Figure 3.4: Diffie-Hellman Process

3.4 Quantum cryptography

The key ideas in quantum mechanics and their significance for creating safe cryptographic protocols are discussed here. A quick overview of PQC algorithms and the risks of quantum cryptography to conventional cryptography are examined.

Basic concepts of quantum mechanics

The bit is the fundamental unit of information for classical information processing. In quantum information processing, the corresponding unit is the qubit, a vector represented in two-dimensional space. Its representation requires the aid of physics, with quantum states represented by a "ket" $|\psi\rangle$. The two fundamental principles are superposition and entanglement (Portmann & Renner, 2022).

In simple terms, we can write the state of a qubit as $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. The states $|0\rangle$ and $|1\rangle$ form the basis for the vector space. On the other hand, α and β are complex or imaginary numbers, called probability amplitudes, that satisfy a mathematical equation $|\alpha|^2 + |\beta|^2 = 1$. If none of the complex numbers is 0, then we can say that $|\psi\rangle$ is in superposition, i.e., either the states are simultaneously $|0\rangle$ and $|1\rangle$.

Quantum entanglement is a physical phenomenon that occurs when quantum particles behave in a way that makes it impossible to describe their quantum state individually. This phenomenon has made it possible to discover that, when measurements are made in this situation, the parties that want to exchange information can produce stronger correlations than any other random classical correlation (Broadbent, Schaffner, abroadbe, & uottawaca B Christian Schaffner cschaffner, 2015).

Quantum threats to conventional encryption

Despite being disruptive, there are still challenges facing quantum cryptography. Quantum cryptography algorithms are predominantly probabilistic, meaning several solutions are returned, but only one is correct. This range of options brings with it the need to check which answer is correct, which takes away one of the strengths of quantum computing: its computing speed. As in classical computing, there can be bit-flips (a 0 can become a one or vice-versa), and inspection is not advisable as it can lead to the qubit being in a state of superposition (Mavroeidis, Vishi, Zych, & Jøsang, 2018).

The advent of quantum computing and cryptography has made it necessary to study the viability of cypher algorithms in the face of the computing power of these machines. Classical computers cannot

compete with the computing power that quantum machines can achieve, and tasks that previously seemed impossible promptly have become a reality. As a result, cryptographic systems and the keys they use can quickly be broken through intensive research, allowing intruders to enter the communication channel.

Quantum cryptography will, according to NIST, usher in the end of asymmetric cypher algorithms in about 20 years. However, the construction of large-scale quantum computers has not yet been announced (Chen et al., 2016). The following table 3.2, adapted from NIST, shows the impact of quantum cryptography on current encryption systems.

Algorithm	Type	Impact
AES	Symmetric Encryption	Keys > 256 bits
RSA	Asymmetric Encryption	Broken
HASH	—	Larger output
ECC	Key Generation	Broken
Diffie-Hellman	Key Generation and Key Exchange	Broken
Prime-Factorisation	Key Generation	Broken
Pseudo Random Number Generation	Key Generation	Depends

Table 3.2: Impact of a Quantum Computer

One of the algorithms that allow asymmetric cryptography and key generation to be broken is Shor's algorithm. It was developed in 1994 by Peter Shor, and its primary function is to discover the factors of the algorithm in polynomial time using a quantum computer, i.e., to find the prime factors, which are used, for example, in the factorization of prime numbers, for a key generation or in RSA, as presented above.

As mentioned earlier, the difficulty and resistance of RSA and prime factorization are mainly due to the computational difficulty of factorizing the numbers. To do this, Shor's algorithm uses a technique of periodically measuring objects. Shor's algorithm works so that two different prime numbers are chosen and multiplied together (n); it is then analyzed to determine whether the result (n) is an even number, a prime number, or the power of a prime number. Next, a random number (x) is chosen, and the maximum common divisor between that number and the result of the product between the prime numbers is calculated. If the value obtained differs from 1, then the number (x) is a factor of n. Next, a value close to one of the previously chosen numbers is chosen, which is a power of 2 (x^2). This number must be greater than n^2 but less than $2n^2$. Finally, mathematical operations are performed until the two factors are obtained (Bhatia & Ramkumar, 2020).

As it is the best known of the asymmetric key algorithms, mentioned several times throughout this part, it is essential to compare the impact on the keys generated by factorization and which are used in asymmetric key mechanisms. Currently, only small entries of qubits have been tested. In 2001, only seven qubits were needed to achieve the operation's success. Despite the operation's success, it is essential to note that it opens up the possibility of larger quantum computers carrying out factorization of larger numbers, such as those used by RSA today. To break a 2048-bit key, which is common in RSA, you need 4096 qubits. To break 160 bits of an ECC key, it is necessary 1000 qubits (Mavroeidis et al., 2018), and to break a 256 bits ECC key, which is equivalent to a 2048 RSA key, is essential 1300 to 1600 bits (Kirsch, 2015).

On the other hand, Grover's algorithm is well known for reducing the strength of symmetric cypher algorithms. This algorithm begins by preparing an initial state that depends on the number of terms. A Hadamard transformation is then applied to create a superposition of states. The target element is marked using an oracle and its amplitude is amplified using a diffusion operation. The last two steps are repeated in the amount of the square root of the number of terms. Measurements are made on the qubits to obtain the target state. The results are a quadratic acceleration compared to classic algorithms to get answers to search problems in unordered lists (Mandviwalla, Ohshiro, & Ji, 2018).

Unlike Shor's algorithm, Grover's algorithm only reduces the strength of the key. In other words, a 128-bit key, which is not recommended nowadays, would have 64-bit security. AES, recognized by the NSA, is resistant to attack by quantum computers, but the key size must be 192 or 256 bits (Mavroeidis et al., 2018). The strength of the key becoming smaller does not jeopardize the security of the information since to obtain the same level of security, the size of the key is doubled (Quantum, 2023).

Hash functions, which can be used as symmetric keys, are a particular case since they are not based on modular arithmetic and are one-way functions. Computers and quantum algorithms are good at discovering the structure behind classical algorithms and their mathematical problems, which are invalid with hash functions. Therefore, given their characteristics, hash functions are safe from attacks by quantum computers (Quantum, 2023).

In short, quantum cryptography algorithms have come to challenge the key generation and encryption systems we use today. Asymmetric key algorithms and the methods used to generate keys, such as prime number factorization or elliptic curves, have their days numbered with the development of these machines. Private vital mechanisms, such as AES, have seen their key discovery time halved, but simply doubling the size of the key restores its security. Hash functions, conversely, are immune to these attacks (Hassija et al., 2020).

Quantum Key Distribution

Nowadays, as we have already discussed in this work, protecting information is vital for the normal functioning of society. Information security is ensured by the importance of the symmetric key secret. The issue that emerges is how to transmit the key safely. These problems open the door for intruders to copy the key. On the other hand, asymmetric key mechanisms solve the transmission problem but are based on the difficulty of factorization problems for computers. Thus, with advances in hardware and algorithms, particularly in quantum computers, the use of QKD is an essential method for generating and distributing keys without both parties needing to have met beforehand (Diamanti, Lo, Qi, & Yuan, 2016).

One of the best-known and most widely used key generation and key distribution algorithms is BB84, created in 1984 by Bennett and Brassard. This algorithm is based on the principles of quantum mechanics, which have already been explained. Alongside using quantum mechanics principles, the information is encoded in the polarisation of individual photons. Since the measurement is probabilistic, an eavesdropper trying to probe the photons used to generate the key can be detected.

The process begins with one of the parties (Alice) sending a sequence of pulses ideally with a random polarization. The polarization of the photons depends on the base chosen for the measurement. Usually, 0 is determined to measure horizontally polarised photons, and 1 to measure vertically polarised photons. A different basis can be used to measure diagonally polarised photons. The recipient measures the polarisation received without the base used being known in advance by the sender. After transmission, the recipient reveals the basis it has used and the sender the bases that match. Both discard the bits where the basis was different, resulting in a key that is less than the original, and normally half the size, but matches both. The following figure 3.5 shows a practical example of the use of BB84 (Bouwmeester & Ekert, 2000).

Alice's Bits	0	1	1	0	1	0	0	1
Alice's Bases	+	+	X	+	X	X	X	+
Alice's polarisation	↑	→	↖	↑	↖	↗	↗	→
Bob's bases	+	X	X	X	+	X	+	+
Bob's polarisation	↑	↗	↖	↗	→	↗	→	→
Private key	0		1			0		1

Bases	0	1
+	↑	→
X	↗	↖

Figure 3.5: BB84

Although the BB84 algorithm has been explained in greater detail, it is not the only one that exists for this purpose. The Heisenberg's uncertainty principle, which deals with the limitation of the simultaneous precision of the measurement of two variables of a particle, i.e., the more we know about the position of a particle, the more difficult it becomes to measure the number of movements, so basically, it's impossible to measure the quantum state without disturbing the original state, or on the principle of entanglement, in which it is impossible to measure individual particles (Chemistry, 2023).

There are several algorithms besides BB84, including the following, separated by the principles mentioned above:

- Non-Entanglement Based - Based on this principle, some algorithms stand out. B92, which is a simplification of BBM92 as it only uses two polarisation states; SSP is a mathematical security proof that only proves resistance to certain attacks and COW, which uses particles in a state of entanglement
- Entanglement Based - BBM92, which is based on the exchange of quantum bits; E91, which uses parts of particles to distribute quantum keys; DPS, which uses quantum transformations (Nurhadi & Syambas, 2018).

The security of quantum key distribution is a relevant aspect and the mechanisms that protocols use to guarantee it are based on various factors, including the quantum principles presented above, superpo-

sition and entanglement, error detection and restarting the key generation process when an intrusion is detected, the quantum non-cloning theorem, security based on quantum physics and resistance to attacks by quantum computers (Woodward, 2012; Rabelo, n.d.).

One of the aspects covered in this chapter is the distribution and generation of quantum keys. The focus of QKDs is generating a key using quantum mechanics principles, as explained above. Although they are pretty secure, no plausible information was found comparing quantum key generation with classical key generation since they are based on quantum principles and mathematical factors.

Due to QKD being based on quantum concepts like the impossibility of quantum cloning, it offers strong theoretical security. One important characteristic is that it can actively detect eavesdropping because measuring a quantum state disturbs the system and alerts the people involved. Since the keys are created at random, it is impossible to listen in without being noticed. Furthermore, QKD ensures security even in the face of advancements in hardware and algorithms due to its resistance to quantum algorithms. QKD does have some restrictions, though. Because the quantum signal weakens over large distances, its efficacy may diminish with increasing distance. Due to the specialist hardware and infrastructure needed for implementation, there is a high cost and technical complexity. Unlike classical encryption systems, QKD can have a limited key transmission rate. Furthermore, after removing bits where the bases were different, the key generation process yields keys that are less than the original, often half, because of the probabilistic nature of quantum measurement. In conclusion, QKD distribution algorithms are good ways of generating and distributing keys. Distributing them using these methods could be a new way of transmitting keys generated by classic methods (A. Kumar & Garhwal, 2021).

Post-Quantum Cryptography - PQC

The security of these post-quantum cryptography algorithms is mainly based on very difficult problems, even for quantum computers. There are several family types of algorithms, such as algorithms based on the difficulty of decoding errors, the difficulty of certain mathematical problems represented in lattices and the security properties of hash functions (FISO, n.d.). The standardisation of these algorithms by NIST is one of the important steps in developing these mechanisms. These will enable the use of public key mechanisms and digital signatures (M. Kumar & Pattnaik, 2020; A. Kumar & Garhwal, 2021).

4 Application examples

Over the next several pages, we'll look at real-world examples to show how cryptography has evolved into a vital data integrity and safety component. From financial transactions to secure communications, from the blockchain to private data storage. We'll look closely at real-world instances and how security procedures are applied in various settings.

Email

Email is one of the ways human beings communicate. From companies to individuals, it is widely used for messaging, as well as for sending attachments. With so much appeal, it's natural that it's attractive to carry out attacks on this means of communication. To provide greater robustness and security for the messages exchanged, classic cryptographic methods are used. Despite this, there are various threats such as intruders who can read the messages, identity theft, modifications to the content of the messages, false messages, unprotected backups, and repudiation, among others. Email uses various protocols to fulfil its function, including SMTP, POP, IMAP and SSL/TLS.

One of the best-known is SSL/TLS. It is used to provide security for data generated at the application layer. This security is also provided through the use of PGP, which makes it possible to guarantee the confidentiality and authenticity of emails. It is an open-source project and is not managed by any organisation. It works as follows: when a message is sent, it sees a digital signature, compresses the information, encrypts it and makes an envelope in a 64-bit message. Although it is widely used, it has limitations and one of them is related to using private keys. The use of these keys can lead to a loss of information if the key is lost since it is the same key used to encrypt and decrypt (Chhabra, Professor, & Professor, 2015).

Blockchain

Blockchain technology is a way of producing a data structure with inherent security qualities. The principles of cryptography are attached to this technology, allowing for greater confidence in transactions. Data is structured in blocks and each block contains either one transaction or a set of transactions. Each block is linked to the previous one in a cryptographic chain, in such a way that it is virtually impossible to tamper with it (IBM, n.d.).

To ensure greater security in blockchain technology, a variant of ECC (ECDSA) is used that is based on DSA, which is a form of asymmetric key that is used to digitally sign messages. The implementation of this cryptographic form allows for a smaller key size and faster computation compared to prime number

factorisation. Another widely used form is hash functions, as they guarantee integrity, consistency and improved security (Dasgupta, Shrein, Kishor, & Gupta, 2019).

Bibliographical references

Agarwal, P., Mittal, S., Tiwari, A., Gupta, I., Singh, A. K., & Sharma, B. (2019, 5). Authenticating cryptography over network in data. *2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019*, 632-636.

Alleaume, R. (2005). Quantum key distribution and cryptography. Retrieved from https://www.osti.gov/etdeweb/biblio/20912875

Ambedkar, B. R., Bharti, P. K., & Husain, A. (2022). Enhancing the performance of hash function using autonomous initial value proposed secure hash algorithm 256. *Proceedings - 2022 IEEE 11th International Conference on Communication Systems and Network Technologies, CSNT 2022*, 560-565. Retrieved from https://ieeexplore.ieee.org/document/9787561

Arogundade, O. R. (2023). Network security concepts, dangers, and defense best practical. , *14*, 2023. Retrieved from www.iiste.org

Bhatia, V., & Ramkumar, K. R. (2020, 10). An efficient quantum computing technique for cracking rsa using shor's algorithm. *2020 IEEE 5th International Conference on Computing Communication and Automation, ICCCA 2020*, 89-94. Retrieved from https://ieeexplore.ieee.org/abstract/document/9250806/figuresfigures

Bisht, N., & Singh, S. (2007). A comparative study of some symmetric and asymmetric key cryptography algorithms. *International Journal of Innovative Research in Science, Engineering and Technology (An ISO*, *3297*. Retrieved from http://www.ijirset.com/upload/2015/march/43$_{AC}OMPARATIVE.pdf$

Boneh, D., Sahai, A., & Waters, B. (2011). Functional encryption: Definitions and challenges. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6597 LNCS*, 253-273. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-19571-6$_16$

Bouwmeester, D., & Ekert, A. (2000). The physics of quantum information.. Retrieved from https://mpl.mpg.de/fileadmin/user$_upload/Chekhova_Research_Group/Lecture_412.pdf$

Broadbent, A., Schaffner, C., abroadbe, B. A. B., & uottawaca B Christian Schaffner cschaffner. (2015, 12). Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography 2015 78:1*, *78*, 351-382. Retrieved from https://link.springer.com/article/10.1007/s10623-015-0157-4

Chemistry. (2023). *Heisenberg's uncertainty principle - chemistry libretexts.* Retrieved from

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). Nistir 8105 report on post-quantum cryptography. Retrieved from http://dx.doi.org/10.6028/NIST.IR.8105

Chhabra, G. S., Professor, G. S. C. A., & Professor, D. S. B. A. (2015). Review of e-mail system, security protocols and email forensics. Retrieved from https://www.researchgate.net/publication/286053691

Dasgupta, D., Shrein, J. M., Kishor, , & Gupta, D. (2019, 1). A survey of blockchain from security perspective. *Journal of Banking and Financial Technology 2018 3:1*, *3*, 1-17. Retrieved from https://link.springer.com/article/10.1007/s42786-018-00002-6

Diamanti, E., Lo, H. K., Qi, B., & Yuan, Z. (2016, 11). Practical challenges in quantum key distribution. *npj Quantum Information 2016 2:1*, *2*, 1-12. Retrieved from https://www.nature.com/articles/npjqi201625

Edwards, J. (2003). *In the most modern hash functions it is extremely difficult to create a collision, which doesn't even make methods for creating them known.* Retrieved from http://target0.be/madchat/crypto/papers/paper879.pdf

Fathi-Vajargah, B., Asghari, R., & Vajargah, B. F. (2016). A novel pseudo-random number generator for cryptographic applications. *Article in Indian Journal of Science and Technology*. Retrieved from https://www.researchgate.net/publication/297651143

FISO. (n.d.). *Bsi - post-quantum cryptography.* Retrieved from https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Post-Quanten-Kryptografie/post-quanten-kryptografie$_node.html$

Gupta, R., Gupta, S., & Singhal, A. (2014, 4). Importance and techniques of information hiding : A review. *International Journal of Computer Trends and Technology*, *9*, 260-265. Retrieved from http://arxiv.org/abs/1404.3063 http://dx.doi.org/10.14445/22312803/IJCTT-V9P149

Hassija, V., Chamola, V., Saxena, V., Chanana, V., Parashari, P., Mumtaz, S., & Guizani, M. (2020, 12). Present landscape of quantum computing. *IET Quantum Communication*, *1*, 42-48. Retrieved from https://onlinelibrary.wiley.com/doi/full/10.1049/iet-qtc.2020.0027 https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-qtc.2020.0027 https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-

qtc.2020.0027

Hoffman, L., & Zahadat, N. (2018). Securing democracy: A comparative look at modern and future us voting systems through the lens of the cia triad. *Journal of Information Assurance and Security*, *13*, 118-124. Retrieved from www.mirlabs.net/jias/index.html

IBM. (n.d.). *What is blockchain security? | ibm.* Retrieved from https://www.ibm.com/topics/blockchain-security

IBM. (2023, 10). *Digital certificates - ibm documentation.* Retrieved from https://www.ibm.com/docs/en/integration-bus/10.0?topic=overview-digital-certificates

Irwin, L. (2023, 2). *What is the cia triad and why is it important?* Retrieved from https://www.itgovernance.co.uk/blog/what-is-the-cia-triad-and-why-is-it-important

Joseph, D. P., Krishna, M., & Arun, K. (2015). Cognitive analytics and comparison of symmetric and asymmetric cryptography algorithms. , *6*. Retrieved from www.ijarcs.info

Kaur, R., & Kaur, A. (2012). Digital signature. *Proceedings: Turing 100 - International Conference on Computing Sciences, ICCS 2012*, 295-301. Retrieved from https://ieeexplore.ieee.org/abstract/document/6391693?casa$_{t}oken$ = $fhX7wylxNjsAAAAA:KeI16z_g2EuqNx_X6u6gE5LT6XZ2uCljsHoK05IlZWuzlJb98xGnhX$ M

Kirsch, Z. (2015). Quantum computing: The risk to existing encryption methods zach kirsch quantum computing: The risk to existing encryption methods. Retrieved from https://www.cs.tufts.edu/comp/116/archive/fall2015/zkirsch.pdf

Kumar, A., & Garhwal, S. (2021, 4). State-of-the-art survey of quantum cryptography. *Archives of Computational Methods in Engineering 2021 28:5*, *28*, 3831-3868. Retrieved from https://link.springer.com/article/10.1007/s11831-021-09561-2

Kumar, M., & Pattnaik, P. (2020, 9). Post quantum cryptography(pqc)-an overview: (invited paper). *2020 IEEE High Performance Extreme Computing Conference, HPEC 2020*. Retrieved from https://ieeexplore.ieee.org/abstract/document/9286147?casa$_{t}oken$ = $IL5LCpDOuqYAAAAA:zgqi9GecYaUE1yLOw58SkNeaP5WhKrgPZA22bflYy4XEWiHG$

Kumar, N., Thakur, J., & Kalia, A. (2011). Performance analysis of symmetric key cryptography algorithms: Des, aes and blowfish.

Lakshmi, P. S., & Murali, G. (2018, 6). Comparison of classical and quantum cryptography using qkd simulator. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, 3543-3547. Retrieved from https://ieeexplore.ieee.org/document/8390120

Li, N. (2010). Research on diffie-hellman key exchange protocol. *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, *4*. Retrieved from https://ieeexplore.ieee.org/abstract/document/5485276?casa$_token$ = $EGqfoSVGZb0AAAAA:KjJWT5IViTvHQkUzYHR5GuvCB6sBqNPQjWcs6og0f2w4ob48$

Lin, J. (2014). The new period encryption technology in the application the study of network security. *Applied Mechanics and Materials*, *685*, 599-602. Retrieved from https://www.scientific.net/AMM.685.599

Mandviwalla, A., Ohshiro, K., & Ji, B. (2018, 7). Implementing grover's algorithm on the ibm quantum computers. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2531-2537. Retrieved from https://ieeexplore.ieee.org/abstract/document/8622457?casa$_token$ = $ssrsTFKyS3kAAAAA:EYx0ZxqYSYJBsnd-AiTgl2nzuR_ThglkheO3HgSo6Ahd_mSf0eSnRencGDEl6_mpjOp1qVPwBA$

Mavroeidis, V., Vishi, K., Zych, M. D., & Jøsang, A. (2018). The impact of quantum computing on present cryptography. *IJACSA) International Journal of Advanced Computer Science and Applications*, *9*. Retrieved from www.ijacsa.thesai.org

Mayfield, T., Roskos, J. E., Welke, S. R., & Boone, J. M. (1991). Integrity in automated information systems.

Menezes, A. J., Oorschot, P. C. V., & Vanstone, S. A. (1996). *Applied cryptography*. CRC Press. Retrieved from https://mrajacse.files.wordpress.com/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf

Milanov, E. (2009). The rsa algorithm. Retrieved from https://pdfdirectory.com/pdf/0702-the-rsa-algorithm.pdf

Márton, K., Suciu, A., Săcărea, C., & Creț, O. (2012). Generation and testing of random numbers for cryptographic applications *. *PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A, OF THE ROMANIAN ACADEMY*, *13*, 368-377. Retrieved from https://acad.ro/sectii2002/proceedings/doc2012-4/11-Suciu.pdf

Nakov, S. (2018, 11). *Crypto hashes and collisions - practical cryptography for developers.* Retrieved from https://cryptobook.nakov.com/cryptographic-hash-functions/crypto-hashes-and-collisions

Nieles, M., Dempsey, K., & Pillitteri, V. Y. (1995, 6). An introduction to computer security: the nist handbook. Retrieved from https://www.nist.gov/publications/introduction-computer-security-nist-handbook

Nurhadi, A. I., & Syambas, N. R. (2018, 11). Quantum key distribution (qkd) protocols: A survey. *Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018*. Retrieved from https://ieeexplore.ieee.org/abstract/document/8527822?casa$_token = CyvOdBMpIAQAAAAA : 6ptVnc4M9XPKhUnuNE1K4LNSKPqc29lB0 - iXBWtJJQN2faEGP3kWhI8bwkHQxuBjFpspq5EByEA$

P., P. C., Shari, L., & Pfleeger, J. M. (2015). *Securtiy in computing* (5th ed.; B. Goodwin, Ed.). Prentice Hall. Retrieved from

Patel, Unnati, Patel, Ashaben, Suthar, Falguni, & Patel. (2019, 10). *(pdf) the study of digital signature authentication process.* Retrieved from https://www.researchgate.net/publication/336603564$_THE_STUDY_OF_DIGITAL_SIGNATU

Patnala, T. R., Jayanthi, D., Majji, S., Valleti, M., Kothapalli, S., & Karanam, S. R. (2020, 3). A modernistic way for key generation for highly secure data transfer in asic design flow. *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 892-897.

Portmann, C., & Renner, R. (2022, 6). Security in quantum cryptography. *Reviews of Modern Physics*, *94*, 025008. Retrieved from https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.94.025008

Quantum. (2023). *Securing the future: Understanding hash-based cryptography's role in quantum resistance.* Retrieved from https://www.btq.com/blog/hash-based-cryptographys-role-in-quantum-resistance

Rabelo, R. (n.d.). Aula 2: Informação e computação quântica introdução aos principais conceitos. Retrieved from https://sites.ifi.unicamp.br/mbonanca/files/2017/07/aula-2.pdf

Rachmawati, D., Tarigan, J. T., & Ginting, A. B. (2018, 3). A comparative study of message digest 5(md5) and sha256 algorithm. *Journal of Physics: Conference Series*, *978*, 012116. Retrieved from https://iopscience.iop.org/article/10.1088/1742-

6596/978/1/012116 https://iopscience.iop.org/article/10.1088/1742-6596/978/1/012116/meta

Rutkowski, E., & Houghten, S. (2020, 7). Cryptanalysis of rsa: Integer prime factorization using genetic algorithms. *2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings*.

Shacklett, M. E. (2021). *What is a digital certificate?* Retrieved from https://www.techtarget.com/searchsecurity/definition/digital-certificate

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, *41*, 303-332. Retrieved from https://www.comparitech.com/blog/information-security/diffie-hellman-key-exchange/ doi: 10.1137/S0036144598347011

Stalling, W. (1989). *Cryptography and network security* (7th ed.). Pearson. Retrieved from

Tsai, C. W., Yang, C. W., Lin, J., Chang, Y. C., & Chang, R. S. (2021, 4). Quantum key distribution networks: Challenges and future research issues in security. *Applied Sciences 2021, Vol. 11, Page 3767*, *11*, 3767. Retrieved from https://www.mdpi.com/2076-3417/11/9/3767/htm https://www.mdpi.com/2076-3417/11/9/3767

Tsohou, A., Kokolakis, S., Lambrinoudakis, C., & Gritzalis, S. (2010). Information systems security management: A review and a classification of the iso standards. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, *26 LNICST*, 220-235. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-11631-5_21

Woodward, A. (2012). *Privacy through uncertainty: Quantum encryption - scientific american blog network.* Retrieved from https://blogs.scientificamerican.com/guest-blog/privacy-through-uncertainty-quantum-encryption/

Yadav, B. S. K., Lal, S., & Arora, S. C. (2010). Some problems in symmetric and asymmetric cryptography doctor of philosophy in mathematics. Retrieved from https://citeseerx.ist.psu.edu/document?repid=rep1type=pdfdoi=c9ccb7efe817c03f082c