International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 3, September 2026, pp. 1238∼1253
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i3.pp1238-1253

Classification of encryption attacks and strategies for mitigation

Anas Maaif , Khalid Zine-Dine
Department of Computer Science, Faculty of Sciences, Mohammed V University, Rabat, Morocco

Article Info

Article history:

Received Apr 28, 2025
Revised Mar 17, 2026
Accepted May 21, 2026

Keywords:

Classification of attacks
Cryptography
Cybersecurity
Defense strategies
Mitigation

ABSTRACT

Cryptography is essential for securing digital communications, yet it remains vulnerable to various malicious attacks. These attacks can be classified based on the type of cryptography they target symmetric or asymmetric. This paper presents a comprehensive classification of encryption attacks, examining the specific vulnerabilities associated with each cryptographic approach. By analyzing these attack vectors, the study shows the importance of understanding weaknesses in cryptographic systems. Furthermore, it proposes several mitigation strategies to strengthen defenses and enhance protection of sensitive information in the digital domain.

Corresponding Author:

Anas Maaifi
Department of Computer Science, Faculty of Sciences, Mohammed V University
Rabat, Morocco
Email: anas_maaifi@um5.ac.ma

1. INTRODUCTION

In today's digital landscape, where vast amounts of information are exchanged continuously, cryptography serves as the cornerstone of security, ensuring the confidentiality, integrity and authenticity of digital communications. However, cryptographic systems, while vital, are not immune to the ever-evolving threats posed by malicious actors. This paper explores the critical area of cryptographic security, emphasizing its foundational role in protecting digital interactions.

The study aims to provide a nuanced understanding of the various attacks that target cryptographic protocols, classifying them to better understand the threat landscape and facilitate the development of tailored defense strategies. Central to this classification is the distinction between symmetric and asymmetric cryptography, each presenting unique vulnerabilities that attackers can exploit.

The most common attack against symmetric cryptography is brute-force, while man-in-the-middle attacks are prevalent in asymmetric cryptography. Usually, attackers primarily focus on breaking or bypassing cryptographic keys, whereas defenders emphasize key length, algorithmic resilience and additional layers of protection tailored to specific attacks. Existing research, however, often addresses either attack descriptions or mitigation strategies in isolation, with few studies providing a structured classification that links each attack to its practical limitations and countermeasures. This approach limits comprehensive understanding, making it difficult for researchers to systematically assess security or design integrated defense strategies. A unified framework that categorizes attacks, aligns them with cryptographic paradigms, and highlights mitigation mechanisms remains largely absent in the literature.

By examining attacks against both types of cryptographic schemes, this work seeks to unravel the specific challenges inherent in each paradigm. Through a comprehensive classification, the paper highlights the

diverse attack vectors that threaten cryptographic security, paving the way for the strategies of robust security measures.

This paper is organized as follows: Section 2 provides a background on symmetric and asymmetric cryptography, laying the foundation for Section 3, where we classify the various attacks on each cryptographic type. In addition to understanding cryptographic attack methodologies, it is equally important to study the mechanisms used to defend against them. Effective defense strategies play a critical role in preserving data confidentiality, integrity and authenticity. As attack techniques continue to evolve, cryptographic systems must adopt adaptive and layered security measures. In this paper, section 4 therefore emphasizes defensive approaches and strategies as a fundamental component of secure cryptographic design, enhancing the resilience of cryptographic systems and ensuring better protection of sensitive information.

To address these challenges, this paper provides a structured analysis of cryptographic attacks and defenses. Specifically, it provides a taxonomy of attacks on symmetric and asymmetric cryptography, maps each attack to its associated vulnerabilities, risks, and practical implications, proposes mitigation strategies tailored to specific attack types, and highlights emerging directions, including AI and ML driven approaches to enhancing cryptographic security.

2. CRYPTOGRAPHY OVERVIEW

In this section, we focus on two primary types of symmetric and asymmetric cryptography, each of which plays a critical role in safeguarding information. We will examine their core objectives, functionality, and areas of application, shedding light on how they contribute to modern security practices.

2.1. Symmetric cryptography

Symmetric cryptography, also known as secret-key cryptography, involves the use of a single key for both encryption and decryption processes. It is characterized by its efficiency in encrypting large amounts of data due to the relatively simple and fast algorithms employed. Common examples of symmetric encryption algorithms include the advanced encryption standard (AES) and the data encryption standard (DES) (see [1]-[6]). The Figure 1 shows a comprehensive structure of symmetric cryptography.

Objectives of symmetric cryptography:

- Confidentiality: ensuring that information is only accessible to authorized parties who possess the secret key.
- Integrity: protecting data from unauthorized alterations.
- Authentication: verifying that the sender and receiver of the message are legitimate.
- Efficiency: enabling fast and resource-effective encryption and decryption of data.

Use cases:

- Data encryption: symmetric encryption is widely used for bulk data encryption, such as securing files, databases, and communication streams in real-time systems.
- Secure communication: symmetric keys are often employed in securing communication channels, such as in virtual private networks (VPNs) and secure sockets layer (SSL)/transport layer security (TLS) protocols.
- Disk and file encryption: full disk encryption tools like BitLocker and file encryption software like VeraCrypt rely on symmetric encryption to safeguard stored data.

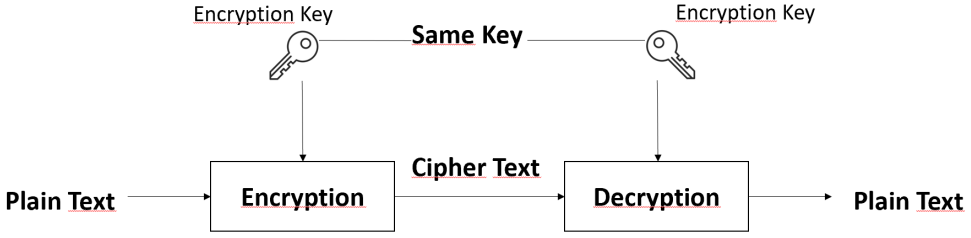

Figure 1. Symmetric cryptography protocol

Despite its advantages, symmetric cryptography presents challenges, particularly in the secure exchange of keys between parties. This vulnerability is addressed by asymmetric cryptography, which we explore in the following section.

2.2. Asymmetric cryptography

Asymmetric cryptography, also referred to as public-key cryptography, uses two distinct keys: a public key for encryption and a private key for decryption. The two keys are mathematically related, but it is computationally infeasible to derive the private key from the public key. Popular asymmetric algorithms include the Rivest-Shamir-Adleman (RSA), Diffie-Hellman, and elliptic curve cryptography (ECC) (see [7]-[11]). The Figure 2 shows a comprehensive structure of asymmetric cryptography.

Objectives of asymmetric cryptography:

- Confidentiality: protecting data by ensuring that only the holder of the private key can decrypt the information.
- Authentication: verifying the identity of the parties involved in the communication through digital signatures.
- Non-repudiation: ensuring that the sender cannot deny having sent the message, enabled by digital signature verification.
- Key exchange: enabling secure key distribution over insecure channels without the need for a shared secret.

Use cases:

- Digital signatures: asymmetric cryptography underpins digital signature schemes used for authenticating the sender's identity and ensuring message integrity.
- Key exchange protocols: protocols like SSL/TLS use asymmetric encryption to securely exchange symmetric keys, which are then used for faster data encryption during communication.
- Email encryption: pretty good privacy (PGP) and secure/multipurpose internet mail extensions (S/MIME) rely on asymmetric cryptography to secure email communications.
- Certificate authorities: public-key infrastructure (PKI) systems use asymmetric cryptography to issue digital certificates, enabling trust in secure communications.

While asymmetric cryptography overcomes the key distribution problem, it is computationally more intensive than symmetric cryptography, making it less efficient for encrypting large volumes of data. Both symmetric and asymmetric cryptographic systems serve critical roles in modern cybersecurity, with each offering unique strengths and addressing specific challenges. Symmetric cryptography excels in efficiency, making it ideal for encrypting large datasets, while asymmetric cryptography provides enhanced security features, particularly in key management and authentication. By understanding the objectives and applications of these two cryptographic types, we can better appreciate the multifaceted nature of cryptographic security and the importance of employing them in tandem to build robust, secure systems.

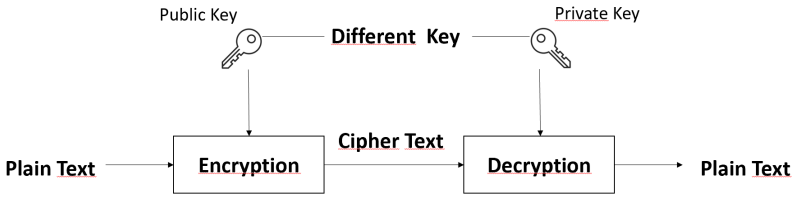

Figure 2. Asymmetric cryptography protocol

3. ATTACKS AGAINST CRYPTOGRAPHIC PROTOCOLS

Cryptographic protocols are fundamental to the security of modern digital systems, providing mechanisms for secure communication, data integrity, and authentication. However, these protocols are not invulnerable and have been the target of various attacks designed to exploit weaknesses in their underlying algorithms, implementation, or key management systems. The classification of these attacks can be broadly divided based on the type of cryptography used symmetric or asymmetric each of which is subject to distinct attack vec-

tors. Understanding these attack categories is essential for identifying vulnerabilities and developing more resilient cryptographic systems. In this section, we provide a comprehensive classification of attacks against both symmetric and asymmetric cryptographic protocols (see [12]-[19]).

3.1. Attacks against symetric cryptography protocols

3.1.1. Brute force attacks

A brute force attack is a method used by attackers to gain unauthorized access to a system by systematically attempting all possible combinations of keys or passwords until the correct one is found. This attack exploits the computational power available to test every potential key or password in a given key space, which is the total number of possible keys that can be generated based on the length and complexity of the password or encryption key.

Process: Figure 3 shows the process of a Bruteforce attack. The effectiveness of brute force attacks diminishes as key lengths increase, but they remain a persistent threat when passwords or encryption keys are inadequately complex.

Steps	Description
Key Space Calculation	The attacker first estimates the size of the key space, which is determined by the length of the key or password and the number of possible characters or symbols that can be used.
Exhaustive Search	The attacker uses software tools or scripts that systematically generate and test each possible key or password within the key space. The process begins with the simplest combinations (e.g., "aaaaaa" for passwords) and continues through more complex variations.
Verification	For each tested key or password, the system compares the input to the correct value. If the key matches, the system grants access; otherwise, the attacker continues testing further possibilities.
Success and Time to Break	The success of a brute force attack depends on several factors, including the length and complexity of the key, the processing power available to the attacker, and the speed of the system's response to each attempt. Longer, more complex passwords exponentially increase the time required to complete the attack, whereas shorter or simpler keys make the attack more feasible.

Figure 3. Bruteforce attack's process

Example of brute force attack: an attacker attempts to break a DES-encrypted message by systematically trying all 2^{56} possible keys until the correct key is found. Due to the relatively short key length of DES, such attacks have become feasible with modern computing power, rendering DES insecure against brute-force attacks.

3.1.2. Frequency analysis attacks

Frequency analysis attack is a cryptanalytic technique used to decipher information by exploiting the frequency of symbols in a ciphertext. This method is particularly effective against simple substitution ciphers, where each letter or symbol in the plaintext is consistently replaced by a corresponding symbol in the ciphertext. By analyzing the frequency of symbols in the ciphertext, attackers can deduce correspondences between the ciphertext symbols and the common letters in the plaintext, thereby breaking the cipher.

Process: Figure 4 shows the process of a frequency analysis attack. This type of attack is particularly effective against classical ciphers like the Caesar cipher and other simple substitution ciphers, where symbol frequencies in the ciphertext closely reflect those in the plaintext.

Example of frequency analysis attack: in a monoalphabetic substitution cipher, the attacker observes that the most frequent symbol in the ciphertext appears approximately 13% of the time. By correlating this frequency with the letter "E," which is the most common letter in English, the attacker can begin reconstructing the plaintext and eventually recover the full message.

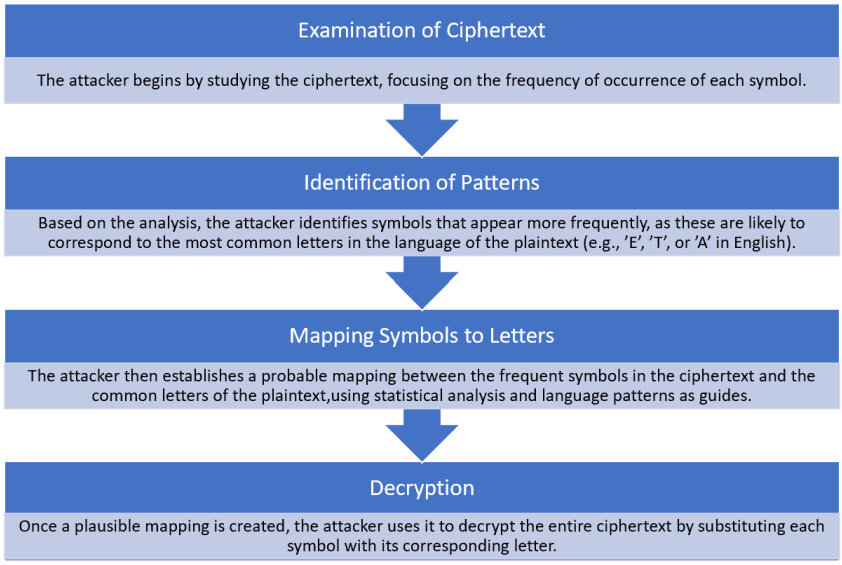

Figure 4. Frequency analysis attack process

3.1.3. Exhaustive search attacks

Exhaustive search attacks, one of the brute-force attacks, involve systematically attempting all possible combinations of characters to find a secret, such as a password or encryption key. This attack method is commonly used in scenarios like password and encryption key cracking, where every conceivable combination is tested until the correct one is identified. The effectiveness of an exhaustive search attack depends on the size of the solution space, which is the total number of possible combinations. Larger solution spaces make the attack more time-consuming and computationally expensive, as the number of combinations to be tested increases exponentially.

Process: the Figure 5 shows the process of an exhaustive search attack. Example of exhaustive search attack: an attacker targets an AES-128 encrypted file and attempts every possible key in the 2^{128} key space. Although theoretically possible, the enormous size of the key space makes such an attack computationally infeasible, demonstrating how increased key length mitigates exhaustive search attacks.

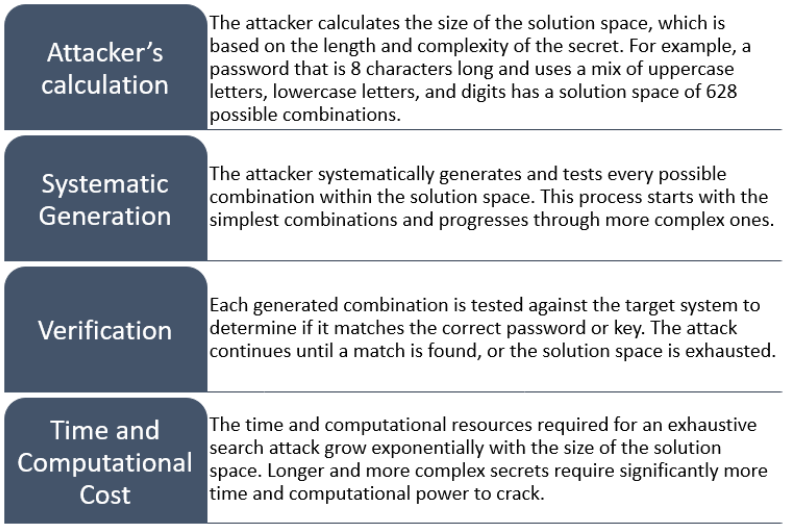

Figure 5. Exhaustive search attack's process

3.1.4. Differential attacks, or differential cryptanalysis

Differential cryptanalysis is an attack method targeting weaknesses in symmetric-key block ciphers by examining how differences in plaintext inputs affect the resulting ciphertexts. This technique exploits the patterns and relationships between differences in plaintexts and their corresponding ciphertexts to gain insights into the encryption algorithm's internal structure. The ultimate goal is to recover portions of the secret key or to enhance the efficiency of brute-force attacks. The success of differential cryptanalysis is highly dependent on the design and structure of the block cipher being targeted.

Process: the Figure 6 shows the process of an differential attack. Example of differential cryptanalysis attack: an attacker encrypts pairs of plaintexts with specific differences and observes the resulting ciphertext differences. By analyzing how these differences propagate through multiple rounds of a block cipher, such as DES, the attacker can deduce information about subkeys used in each round, significantly reducing the key search space.

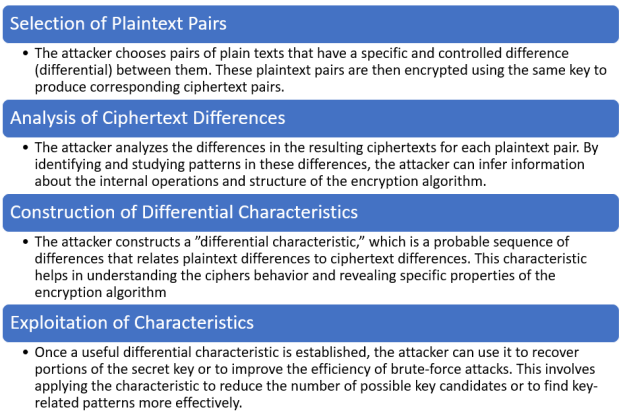

Figure 6. Differential attack's process

3.1.5. Oracle attacks

Oracle attacks involve exploiting a system that provides information about cryptographic operations to gain insights into protected secret information. These attacks leverage the system's responses to queries or operations to extract sensitive data, such as plaintext or cryptographic keys. Different types of oracle attacks exploit various aspects of the cryptographic system's behavior, including how it handles padding, timing, errors, and physical side-channel information.

a. Padding Oracle attacks
   - Scenario: in symmetric encryption schemes that use padding, attackers submit ciphertexts to an oracle and observe whether the decrypted plaintext's padding is valid.
   - Exploitation: by analyzing the oracle's responses, attackers can infer information about the plaintext and gradually deduce the entire message, leveraging the patterns in padding validity.

The Figure 7 shows the process of a padding Oracle attack.

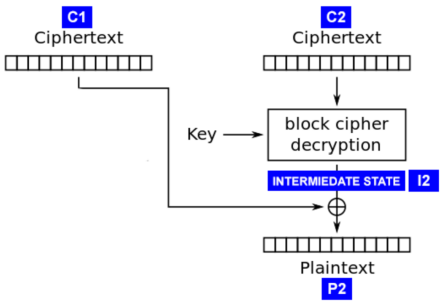

Figure 7. Padding Oracle attack's process

b. Timing Oracle attacks

- Scenario: these attacks exploit variations in the time taken for cryptographic operations to complete.
- Exploitation: attackers measure the time it takes for the system to process cryptographic operations to infer details about the encrypted data or cryptographic key. Variations in processing time can reveal information about the internal state of the system or specific bits of the key.

The Figure 8 shows the process of a padding Oracle attack.

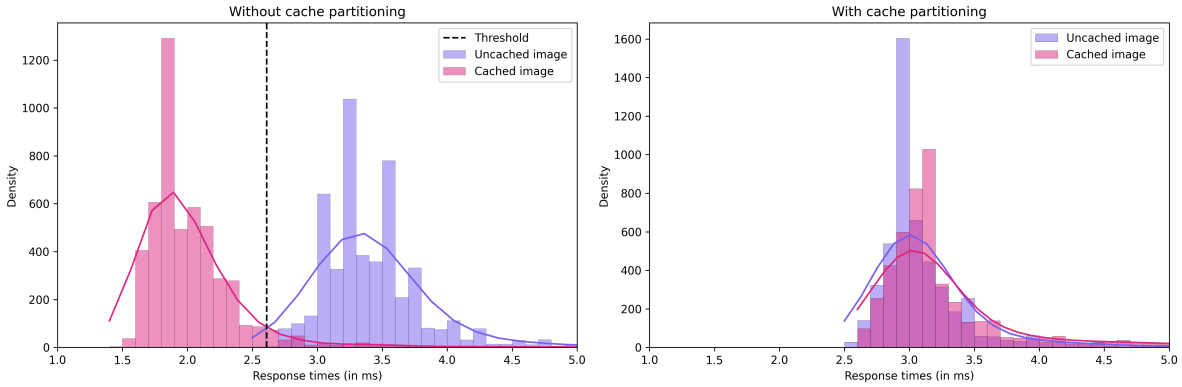

Figure 8. Timing Oracle attacks

c. Error-based Oracle attacks

- Scenario: involves exploiting error messages generated when cryptographic operations fail, such as during decryption or validation processes.
- Exploitation: attackers analyze the content and patterns of error messages to gain insights into the correctness of parts of the cryptographic input, potentially revealing information about the secret key or plaintext.

d. Side-channel Oracle attacks

- Scenario: These attacks exploit physical information leaked during the execution of cryptographic algorithms, such as variations in power consumption, electromagnetic radiation, or acoustic emissions.
- Exploitation: By observing and analyzing these side-channel leaks, attackers can infer details about the cryptographic key or the internal workings of the cryptographic algorithm.

The Figure 9 shows the side channel Oracle attack. Process: the Figure 10 shows the process of a side channel Oracle attack. Symmetric cryptography protocols are vulnerable to a wide range of attacks, including brute force, frequency analysis, exhaustive search, differential, and Oracle attacks. Additional attack types such as replay attacks, meet-in-the-middle attacks, side-channel attacks, known-plaintext attacks (KPA), and chosen-plaintext attacks (CPA) further demonstrate the diversity of potential threats.

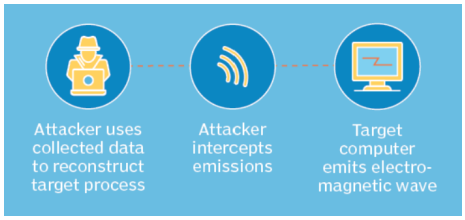

Figure 9. Side-channel Oracle attacks

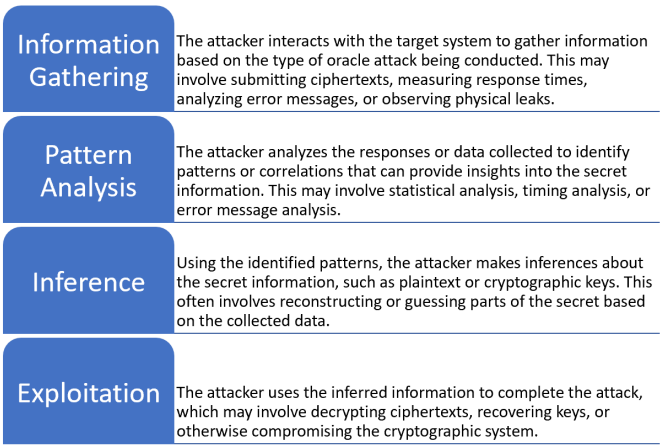

Figure 10. Side-channel Oracle attack's process

3.2. Attacks against asymetric cryptography protocols

3.2.1. Factorization attacks

Factorization attacks target the prime factors of public keys used in asymmetric cryptographic algorithms, such as RSA, with the objective of recovering the private key. This type of attack exploits the mathematical difficulty of factoring a large number into its prime components, which is the foundation of RSA's security. The process involves decomposing the public key typically a product of two large prime numbers into its component primes, thereby enabling the attacker to derive the private key.

Process: the Figure 11 shows the process of a factorization attack. Example of factorization attack (RSA): an attacker targets an RSA public key by attempting to factor the modulus n, which is the product of two large prime numbers. If the attacker successfully factors n into its prime components, they can compute the private key and decrypt encrypted messages or forge digital signatures. Historical advances in integer factorization algorithms have demonstrated that insufficient key sizes make RSA vulnerable to such attacks.

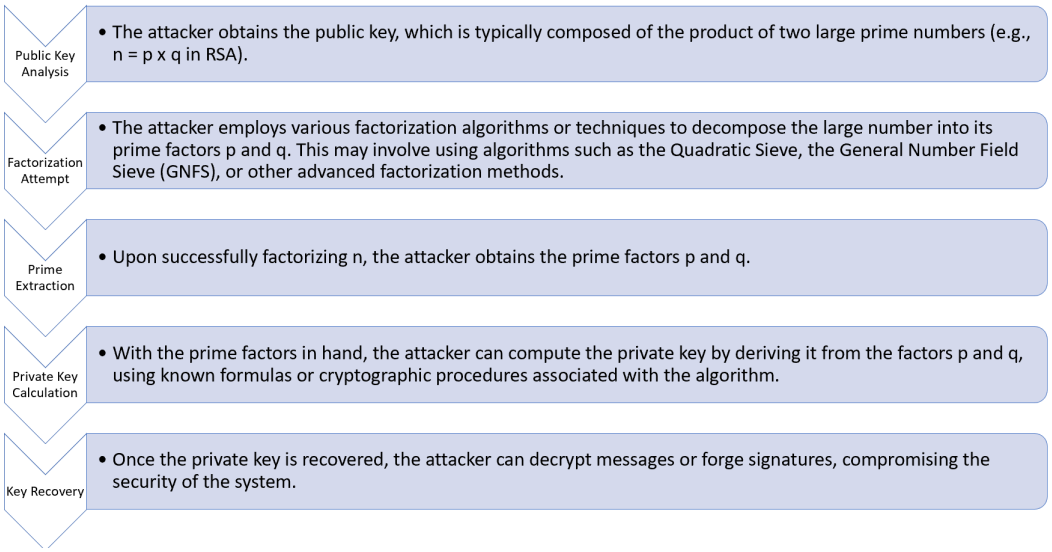

Figure 11. Factorization attack's process

3.2.2. Exhaustive search for private keys

Exhaustive Search for Private Keys, also known as brute-force search, is a cryptographic attack in which an attacker systematically tests all possible private key combinations to find the correct one. This method

is used in asymmetric key algorithms and relies on the impracticality of exhaustively searching the entire key space due to its vast size. The feasibility of this attack decreases as the size of the key space grows, making it increasingly difficult to discover the correct key.

Process: the Figure 12 shows the process of an exhaustive search attack. Example of exhaustive Search for private keys: an attacker attempts to recover an asymmetric private key by systematically testing all possible key values. Although theoretically possible, the extremely large key spaces used in modern asymmetric algorithms such as RSA and ECC make exhaustive search attacks computationally infeasible when appropriate key lengths are employed.

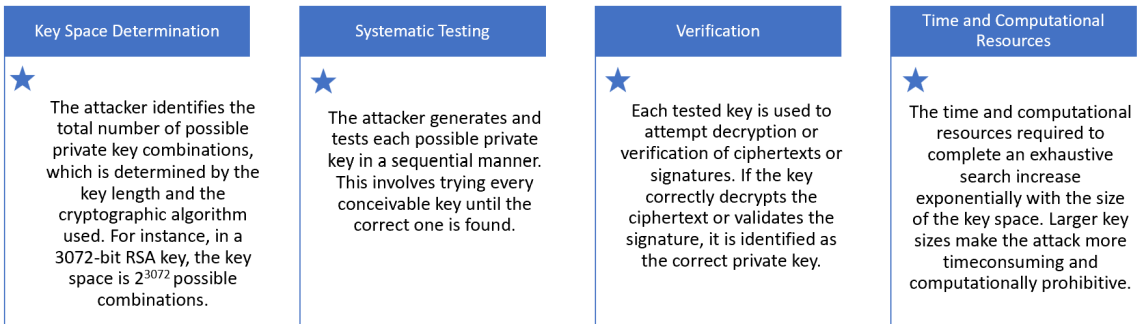

Figure 12. Exhaustive search's process

3.2.3. Collision attacks

Collision attacks target hash functions with the objective of finding two distinct inputs that produce the same output hash value, known as a collision. Hash functions are designed to map arbitrary data to fixed-size hash values. In a collision attack, the attacker aims to find two different inputs, M_1 and M_2, such that $H(M_1) = H(M_2)$, where H represents the hash function. The ability to find such collisions undermines the integrity and security guarantees provided by the hash function.

Process: the Figure 13 shows the process of a collision attack. Example of collision attack on hash functions Used in asymmetric cryptography: an attacker finds two distinct messages that produce the same hash value using a weak hash function. In digital signature schemes, this allows the attacker to substitute a legitimate signed message with a malicious one that has the same hash, thereby undermining message integrity and authentication in asymmetric cryptographic systems.

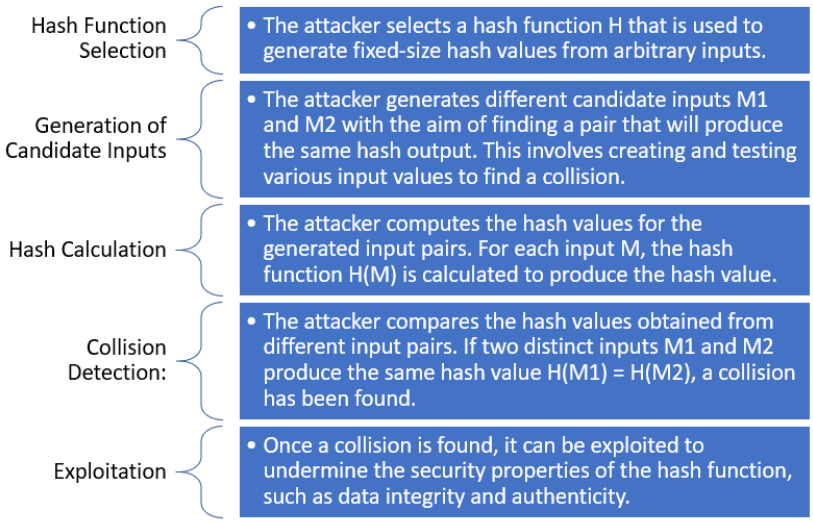

Figure 13. Collision attack's process

3.2.4. Timing analysis attacks

Timing analysis attacks are a type of side-channel attack that exploit variations in the time taken by a cryptographic system to perform specific operations. By measuring these time variations, attackers can infer sensitive details, such as cryptographic keys. These attacks fall under the category of side-channel attacks, which leverage unintended information leakage from the physical implementation of a cryptographic system. Timing variations can arise due to factors such as input data, system state, or internal cryptographic processes. The primary objective of timing analysis attacks is to analyze timing differences to reveal information about the cryptographic key or other sensitive details.

Process: the Figure 14 shows the process of a timing analysis attack. Example of RSA timing analysis attack: an attacker measures the time taken by a server to perform RSA private-key operations, such as decryption or signature generation. By analyzing variations in execution time correlated with key-dependent operations, the attacker can infer information about the private key. This type of attack exploits implementation weaknesses rather than flaws in the RSA algorithm itself.

Figure 14. Timing analysis attack's process

3.2.5. Side-channel attacks

Side-channel attacks exploit unintentional information leaks from the physical implementation of a cryptographic system, circumventing the mathematical robustness of the algorithms themselves. These attacks leverage observable side-channel information, such as power consumption, electromagnetic radiation, execution time, or even sound, to infer sensitive details such as cryptographic keys. By analyzing these unintentional leaks, attackers can gain insights into the internal workings of the cryptographic system [20].

Common types of side-channel attacks:

- Timing attacks: exploit variations in the time taken for cryptographic operations. By measuring the execution time of different operations, attackers can infer details about the internal state or key material.
- Power analysis attacks: monitor the power consumption of a cryptographic device during operations. Variations in power consumption can be correlated with specific computations, revealing information about the cryptographic key or intermediate values.
- Electromagnetic analysis attacks: observe electromagnetic radiation emitted by a cryptographic device during operations. The patterns in electromagnetic emissions can provide insights into the internal processes and cryptographic keys.
- Acoustic analysis attacks: capture and analyze the sound produced by a cryptographic device while it is operating. Variations in sound patterns can be exploited to deduce details about cryptographic computations.
- Cache attacks: exploit the behavior of cache memory during cryptographic operations. By analyzing cache access patterns and memory usage, attackers can infer information about the data being processed.

– Fault attacks: induce intentional faults or errors in a cryptographic device during operation. By analyzing how these faults affect the system's behavior, attackers can reveal internal states or cryptographic keys.

Process: the figure below (Figure 15) shows the process of a side channel attack. Example of side-channel attack on asymmetric cryptography devices: an attacker observes the power consumption or electromagnetic emissions of a hardware device performing RSA or ECC operations. By correlating these physical leakages with cryptographic computations, the attacker can recover sensitive information such as private keys, even when the underlying cryptographic algorithm is mathematically secure.

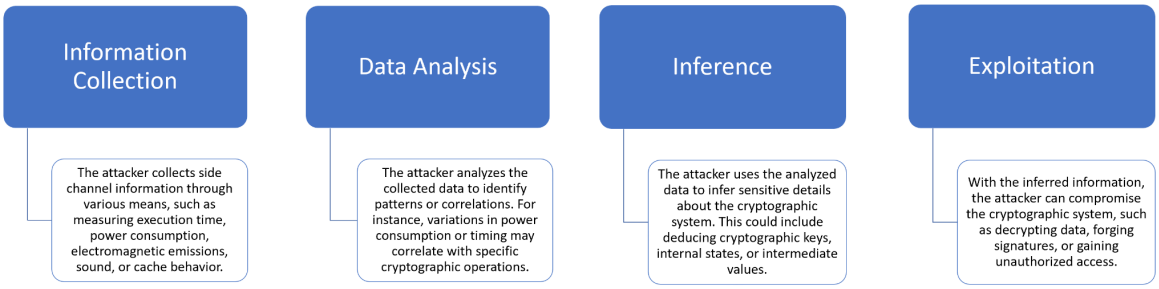

Figure 15. Side channel attack's process

3.2.6. Man-in-the-middle attacks

Man-in-the-middle (MitM) attacks are a type of cyberattack in which an unauthorized third party intercepts and potentially alters the communication between two legitimate parties. The attacker may impersonate one or both parties to manipulate or eavesdrop on the communication. MitM attacks can occur in various scenarios, including Wi-Fi networks, public networks, unsecured websites, or through compromised hardware devices. Common techniques used in MitM attacks include DNS spoofing [21], ARP spoofing [22], session hijacking [23], and SSL/TLS stripping [24]. Successful MitM attacks can lead to unauthorized access to sensitive information, injection of malicious content, or manipulation of communications.

The Figure 16) shows the man in the middle attack. The figure above summarizes the sequence of a Man-in-the-Middle attack, where an attacker secretly positions themselves between a client and a server. The attacker intercepts the client's requests, potentially capturing or altering sensitive data. These messages are then forwarded to the legitimate server to maintain the appearance of normal communication. The server's responses are likewise intercepted and relayed back to the client. This process demonstrates how MITM attacks compromise data confidentiality and integrity without detection.

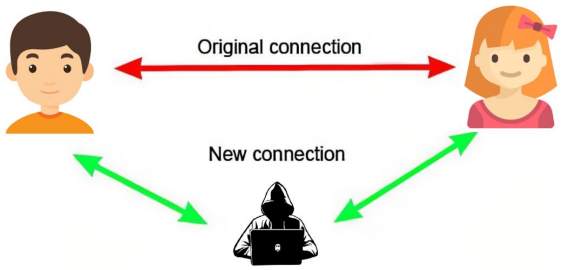

Figure 16. Man in the middle attacks

Process: the Figure 17 shows the process of a man in the middle attack. Example of man-in-the-middle attack on public key exchange: during an unsecured public key exchange, an attacker intercepts the communication between two parties and substitutes their own public key in place of the legitimate one. As a result, both parties unknowingly encrypt messages using the attacker's key, allowing the attacker to decrypt, modify, and re-encrypt messages while remaining undetected. This attack highlights the importance of authentication mechanisms such as digital certificates.

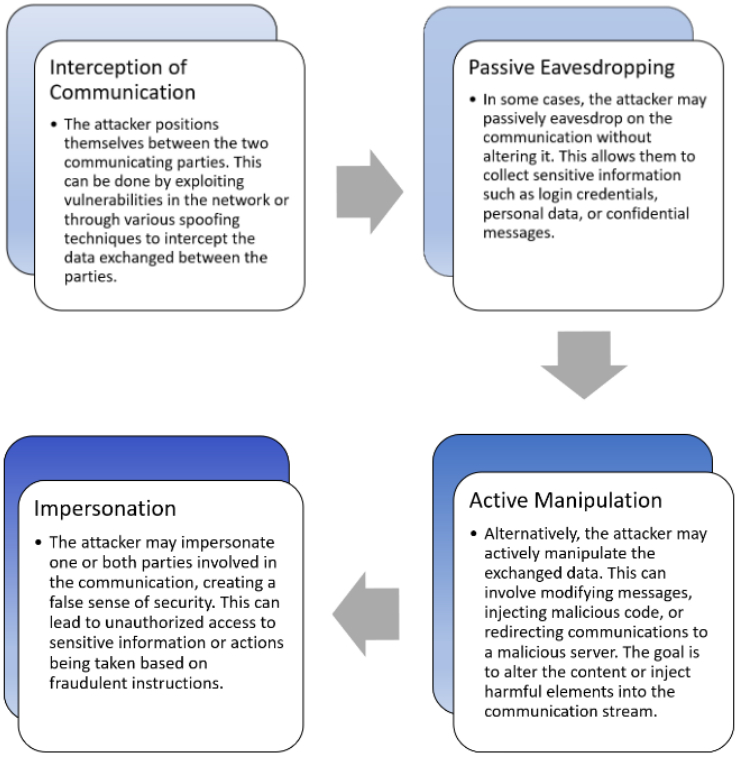

Figure 17. Man in the middle attack process

Asymmetric cryptography protocols face a variety of attacks, including factorization attacks, exhaustive search for private keys, collision attacks, timing analysis, side-channel atacks, and man-in-the-middle attacks. These attacks exploit both mathematical weaknesses and implementation flaws, highlighting that the security of public-key systems depends on careful design and key management. Additional attack types continue to emerge, demonstrating that the threats to asymmetric cryptography are diverse and evolving. It is important to note that the examples discussed here are not exhaustive, and many other attack methods exist.

4. DEFENSE STRATEGIES AGAINST CRYPTOGRAPHIC ATTACKS

This section presents a comprehensive overview of defense strategies against cryptographic attacks, structured into two complementary parts. The first part discusses practical and theoretical mitigation mechanisms designed to strengthen cryptographic systems against known threats. The second part provides a structured analytical mapping in the form: Attack → Weakness Exploited → Mitigation, highlighting the relationship between vulnerabilities and corresponding countermeasures. Together, these perspectives offer a systematic framework for enhancing cryptographic resilience.

4.1. Defense strategies

Following the examination of symmetric and asymmetric cryptographic attacks, this section focuses on the defense strategies designed to mitigate these threats and strengthen cryptographic systems. It discusses practical and theoretical countermeasures such as the use of stronger key lengths, secure key management practices, and the adoption of advanced and quantum-resistant cryptographic algorithms. Emphasis is placed on building resilient security architectures capable of adapting to evolving attack techniques and emerging computational threats:

a. Brute force

To combat brute force attacks, it is important to enforce account lockouts after multiple failed login attempts, apply strong password policies, and use two-factor authentication (2FA) to enhance security. These measures help protect sensitive data from compromise.

b. Frequency analysis attacks
To mitigate frequency analysis attacks, modern block and stream ciphers should be used alongside more complex encryption techniques that obscure statistical relationships between plaintext and ciphertext, making such attacks ineffective.
c. Exhaustive search attacks
Defense against exhaustive search attacks relies on using longer and more complex passwords or encryption keys, employing algorithms with large key sizes, and implementing protections such as rate limiting, account lockouts, and multi-factor authentication.
d. Differential attacks
To resist differential attacks, systems should employ strong and well-designed block ciphers and ensure regular updates and reviews of cryptographic protocols and algorithms.
e. Oracle attacks
Mitigating oracle attacks requires secure coding practices that avoid leaking sensitive information through error messages, designing cryptographic systems to handle errors safely, using well-established protocols resistant to such attacks, and regularly updating implementations to address emerging vulnerabilities (see [25]).
f. Factorization attacks
Protection against factorization attacks involves using large key sizes, adopting alternative algorithms such as elliptic curve cryptography (ECC), staying informed about advances in factorization techniques, and regularly updating cryptographic practices.
g. Exhaustive search for private keys
To counter exhaustive search for private keys, it is essential to use large key sizes, update key lengths regularly, adopt secure alternatives like ECC, and remain aware of developments in computing and cryptography.
h. Collision attacks
Mitigating collision attacks requires the use of collision-resistant hash functions such as SHA-256 or SHA-3, continuous updates to stronger hash functions, awareness of emerging vulnerabilities, and proper implementation of secure hashing techniques.
i. Timing analysis attacks
To defend against timing analysis attacks, cryptographic implementations should use constant-time algorithms, incorporate techniques such as dummy operations or fixed execution times, avoid timing leaks in code, and undergo regular security audits and testing.
j. Side-channel attacks
Protection against side-channel attacks includes ensuring constant-time cryptographic operations, introducing randomness to obscure leaked information, designing hardware to reduce physical leakage, implementing fault detection mechanisms, and conducting regular security audits.
k. Man-in-the-middle attacks
To prevent man-in-the-middle attacks, secure communication protocols such as HTTPS should be used, strong authentication mechanisms like 2FA should be implemented, public networks should be used cautiously with VPNs when necessary, and cryptographic protocols must be correctly implemented.

By promoting a culture of awareness and cooperation, while utilizing advanced encryption and secure communication protocols, we can more effectively safeguard our sensitive information from interception and tampering.

4.2. Structured analytical mapping

Attack on symmetric cryptography relies on the security of shared secret key and key length and is exploitable by several types of attacks. Understanding these attacks helps to strengthen protocol design and implementation. The chart (Figure 18) shows Attacks against symmetric cryptography protocols.

Attacks on symmetric systems mainly exploit weak keys, statistical patterns, or implementation flaws. Using strong algorithms, sufficient key sizes, and secure modes of operation greatly reduces risk. Proper implementation is as important as the algorithm itself for ensuring security. Asymmetric cryptography uses a public and private key pair to secure communications. Its security depends on mathematical hardness and correct protocol implementation. However, multiple attacks target both its theoretical and practical weaknesses. By knowing such attacks it can help reinforce the design and implementation of the protocol.

Attack	Weakness Exploited	Mitigation
Brute Force Attack	Small key size; insufficient key entropy	Use large key sizes (e.g., AES-256); enforce strong key generation; implement rate limiting
Exhaustive Search Attack	Predictable or limited key space	Increase key length; use cryptographically secure random number generators (CSPRNGs)
Frequency Analysis Attack	Statistical patterns in plaintext (common in weak or classical ciphers)	Use modern block ciphers (AES); apply proper modes of operation; avoid deterministic encryption
Differential Cryptanalysis	Structural weaknesses in cipher design; predictable input–output differences	Use well-studied algorithms (AES); apply sufficient rounds; avoid custom cipher designs
Oracle Attack (e.g., Padding Oracle)	Leakage of error information; improper validation; distinguishable responses	Use authenticated encryption (AES-GCM); implement constant error messages; apply constant-time processing

Figure 18. Attacks against symmetric cryptography protocols

The chart (Figure 19) shows Attacks against symmetric cryptography protocols. Strong key sizes, secure hash functions, and authenticated exchanges are essential defenses. Combining robust algorithms with secure implementation ensures long-term protection. Strong key sizes, secure hash functions, and authenticated exchanges are essential defenses. Combining robust algorithms with secure implementation ensures long-term protection.

Attack	Weakness Exploited	Mitigation
Factorization Attack (e.g., RSA)	Small key size; weak prime generation	Use large key sizes (≥2048-bit RSA); generate strong random primes; use ECC where appropriate
Exhaustive Search for Private Keys	Small key size; weak randomness in key generation	Use sufficiently large key sizes; secure key generation using CSPRNGs
Collision Attack (e.g., on hash functions)	Weak hash functions (e.g., MD5, SHA-1)	Use collision-resistant hashes (SHA-256, SHA-3); apply digital signature best practices
Timing Analysis Attack	Variable-time operations revealing private key information	Implement constant-time algorithms; use cryptographic libraries hardened against timing leaks
Side-Channel Attack	Leakage through power consumption, electromagnetic signals, or cache behavior	Use hardware protections; apply masking/blinding techniques; constant-time implementations

Figure 19. Attacks against asymmetric cryptography protocols

5. CONCLUSION

This paper has provided a comprehensive classification of attacks on cryptographic systems, highlighting the diverse and evolving nature of these threats. By examining various attack vectors and their implications for both symmetric and asymmetric cryptographic methods, the study underscores the critical importance of understanding these vulnerabilities. This knowledge is essential for crafting effective, tailored defense strategies to protect sensitive information. The recommended mitigation strategies are vital for reinforcing cryptographic systems against malicious threats. As cryptographic techniques and attack methods continue to evolve, ongoing research and adaptation are crucial. Staying updated on advancements in cryptography and emerging threats is

essential for maintaining strong protection of digital assets.

Future work will focus on integrating advanced cryptographic approaches such as homomorphic encryption and lattice-based cryptography, which offers strong resistance against quantum attacks. In addition, artificial intelligence and machine learning technologies will be leveraged to develop adaptive encryption methods, streamline key management, and detect anomalous patterns indicative of potential attacks. By combining these emerging techniques, cryptographic frameworks can achieve real-time threat responsiveness, automated defenses, and enhanced resilience against both classical and emerging threats. Embracing these innovations will not only strengthen defenses but also contribute to a more secure and future-proof digital landscape.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Maaifi Anas	✓	✓			✓	✓		✓	✓					
Khalid Zine-Dine		✓		✓	✓					✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] S. Chandra, S. Bhattacharyya, S. Paira, and S. S. Alam, "A study and analysis on symmetric cryptography," in *Proc. Int. Conf. Science Engineering and Management Research (ICSEMR)*, Nov. 2014, pp. 1–6, doi: 10.1109/ICSEMR.2014.7043664.

[2] A. B. Kidmose, "A study and analysis on symmetric cryptography," M.S. thesis, 2022.

[3] M. U. Bokhari and Q. M. Shallal, "A review on symmetric key encryption techniques in cryptography," *Int. J. Computer Applications*, 2016, doi: 10.5120/ijca2016911203.

[4] P. Singh and P. Shende, "Symmetric key cryptography: Current trends," *Int. J. Computer Science and Mobile Computing*, vol. 3, no. 12, Dec. 2014.

[5] B. Mandal, S. Chandra, S. S. Alam, and S. S. Patra, "A comparative and analytical study on symmetric key cryptography," in *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 2014, doi: 10.1109/ICECCE.2014.7086646.

[6] R. Deepu and V. Krishnan, "Study of symmetric key cryptography algorithms," *Int. J. Computer Techniques*, vol. 2, no. 2, pp. 45-50, Mar. 2015.

[7] Z. Meng and Y. Wang, "Asymmetric encryption algorithms: Primitives and applications," in *Proc. IEEE 2nd Int. Conf. Electronic Technology, Communication and Information (ICETCI)*, May 2022, doi: 10.1109/ICETCI55101.2022.9832032.

[8] S. Nithya and E. G. D. P. Raj, "Survey on asymmetric key cryptography algorithms," *J. Advanced Computing and Communication Technologies*, vol. 2, no. 1, Feb. 2014.

[9] K. Maduguma and T. Gundu, "Multi-key asymmetric cryptography: A model for preserving privacy in work-from-home environments," in *Proc. 23rd Eur. Conf. Cyber Warfare and Security (ECCWS)*, 2024.

[10] J. Sivakumar and H. Begum, "Integer factorization in RSA encryption: Challenge for cloud attackers," *Int. J. Computer Science Trends and Technology*, vol. 5, no. 2, pp. 1–5, Mar.–Apr. 2017.

[11] A. K. Yau, K. G. Paterson, and C. J. Mitchell, "Padding oracle attacks on CBC-mode encryption with secret and random IVs," in *International Workshop on Fast Software Encryption*, Jul. 2005, pp. 299-319, doi: 10.1007/11502760_20.

[12] A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," *Jurnal Pendidikan Teknologi Informasi*, Oct. 2018.
[13] S. Abdulkarim, S. K. Ahmad, and F. Binord, "A review of faults attack on symmetric key cipher and appropriate counter measures," *OIRT J. Information Technology*, vol. 2, no. 1, pp. 1–7, doi: 10.53944/ojit-2206.
[14] M. Ziatdinov, "Using frequency analysis and Grover's algorithm to implement known ciphertext attack on symmetric ciphers," *Lobachevskii J. Mathematics*, vol. 34, no. 4, pp. 313-315, Dec. 2013, doi: 10.1134/S1995080213040148.
[15] P. Radanliev, "Cyber-attacks on public key cryptography," *Preprints*, Sep. 2023, doi: 10.20944/preprints202309.1769.v1.
[16] N. Asaithambi, "A study on asymmetric key cryptography algorithms," *Int. J. Computer Science and Mobile Applications*, vol 3. no. 4, pp. 8-13, Apr. 2015.
[17] R. Rani and S. Venkateswarlu, "Security against timing analysis attack," *Int. J. Electrical and Computer Engineering*, vol. 5, no. 4, pp. 759-764, Aug. 2015, doi: 10.11591/ijece.v5i4.pp759-764.
[18] A. H. A. Ghafar and M. R. K. Ariffin, "Timing attack analysis on AAβ cryptosystem," *J. Computer and Communications*, Mar. 2014, doi: 10.4236/jcc.2014.24001.
[19] A. Pulkit, S. Naval, and V. Laxmi, "A survey of side-channel attacks in context of cache—taxonomies, analysis and mitigation," *Cryptography and Security*, Dec. 2023, doi: 10.48550/arXiv.2312.11094.
[20] M. A. Hussain, H. J. Z. A. Hussien, and A. Ibrahim, "DNS protection against spoofing and poisoning attacks," in *Proc. 3rd Int. Conf. Information Science and Control Engineering (ICISCE)*, Jul. 2016, pp. 1308-1312, doi: 10.1109/ICISCE.2016.279.
[21] Y. Liu, K. K. Dong, L. Dong, and B. Li, "Research of the ARP spoofing principle and a defensive algorithm," *WSEAS Transactions on Communications*, vol. 7, no. 5, pp. 516-520, 2008, doi: 10.5555/1456071.1456089.
[22] A. K. Baitha and S. Vinod, "Session hijacking and prevention technique," *Int. J. Engineering & Technology*, vol. 7, no. 2.6, pp. 193-198, 2018, doi: 10.14419/ijet.v7i2.6.10566.
[23] S. Puangpronpitag and N. Sriwiboon, "Simple and lightweight HTTPS enforcement to protect against SSL striping attack," in *Proc. Int. Conf. Computational Intelligence, Communication Systems and Networks (CICSyN)*, 2012, pp. 229-234, doi: 10.1109/CICSyN.2012.50.
[24] D. Salama, H. A. A. Elkader, and M. M. Hadhoud, "Evaluating the effects of symmetric cryptography algorithms on power consumption for different data types," *Int. J. Network Security*, vol. 11, no. 2, Jan. 2010.
[25] X. Chang, W. Li, A. Yan, P. W. M. Tsang, and T.-C. Poon, "Asymmetric cryptosystem based on optical scanning cryptography and elliptic curve algorithm," *Scientific Reports*, vol. 12, no. 1, May 2022, doi: 10.1038/s41598-022-11861-x.

BIOGRAPHIES OF AUTHORS

Anas Maaifi Received his first engineering degree from the national school of applied sciences, Network and Telecommunication, Morocco, in 2015. He is currently working as a cybersecurity engineer. His main research interests focus on Cybersecurity and Artificial Intelligence. He can be contacted at email: anas_maaifi@um5.ac.ma.

Khalid Zine-Dine He has over 24 years of combined academic and professional experience in the field of computer systems/networks and cybersecurity. Prior to academic tenure, he served for 4 years as a Network and System Security Project Manager in the banking sector. Currently, he serves as a Full Prof. His research interests cover a broad spectrum, including Syst./Network security, Smart grids and AI. He can be contacted at email: khalid.zinedine@fsr.um5.ac.ma.