The Evolution of Cryptography and a Contextual Analysis of the Major Modern Schemes

Anirudh Ketha

Received October 21, 2023
Accepted March 24, 2024
Electronic access April 15, 2024

Cryptography, the art and science of securing information, has a rich historical history, dating back to ancient civilizations. Over the centuries, it has evolved and matured to address the growing challenges and complexities posed by advancing technologies and the increasing sophistication of adversaries. The primary goal of cryptography is to enable secure communication between parties, ensuring that sensitive data remain confidential and protected from unauthorized access or manipulation. In today's rapidly advancing digital age, where virtually every aspect of our lives is interconnected and reliant on technology, the importance of cryptography cannot be overstated. With the proliferation of the Internet, cloud computing, and the widespread use of smartphones and other connected devices, the volume of data that is transmitted and stored electronically has increased exponentially. However, this digital transformation has also brought about new vulnerabilities, making our data susceptible to various threats, including data breaches, cyber-attacks, and malicious hacking attempts. As a result, cryptographic experts and security experts are constantly pushing the boundaries of cryptographic techniques to develop more robust and efficient schemes that can withstand the relentless onslaught of modern threats. Thus, as a scientific community, we must understand older cryptographic schemes to address the inherent limitations and vulnerabilities of our modern security in order to achieve better encryption schemes in the future. Cryptography experts continuously push the boundaries, developing robust techniques to counter modern threats. This paper delves into the historical context, examining the limitations of older schemes and their roots. It then navigates through the evolution of cryptography, shedding light on four major modern schemes: OTP, RSA, AES, and ECC. The contextual analysis delves into the strengths and applications of each major scheme. OTP, which excels in confidentiality and integrity, finds its niche in scenarios such as superencryption and diplomatic communications. RSA emerges as a secure choice for key exchange and authentication, contributing to secure email and online transactions. AES stands out for efficiency in bulk data encryption, becoming the global standard. ECC's unique strength lies in its computational efficiency, making it ideal for resource-constrained environments.

Research Question

How have modern cryptographic schemes evolved to address the limitations of older schemes? In which specific contexts do major security schemes like OTP, RSA, AES, and ECC excel? To research the evolution of modern cryptographic schemes and their specific contexts, we will analyze historical developments, compare the strengths and weaknesses of the major schemes, and synthesize this information to draw meaningful conclusions. The research will employ metrics such as confidentiality, integrity, computational efficiency, key distribution, resistance to attacks (including brute force and frequency analysis), and adaptability to diverse environments.

Historical Overview of Cryptography

The origins of cryptography can be traced back to ancient civilizations, where the need for secure communication led to the development of rudimentary encryption techniques [1]. The Egyptians, known for their advanced knowledge of writing and communication, employed hieroglyphic substitution ciphers as one of the earliest recorded forms of cryptographic schemes . By replacing individual hieroglyphs with others, they created secret messages that could only be understood by those with the knowledge of the cipher's key. The Romans, notably with the work of Julius Caesar during the Republic, made significant contributions to cryptography with the introduction of the Caesar cipher [2]. This simple substitution cipher involved shifting the letters of the alphabet by a fixed number of positions, known as the "key," to encrypt and decrypt messages. Caesar used this method to protect sensitive military communiques, thereby establishing the concept of cryptographic security for military and diplomatic purposes. As civilizations evolved, so did cryptographic techniques. During the Renaissance period, with the widespread adoption of printing presses, new challenges emerged in maintaining the confidentiality of information. To overcome this, polyalphabetic ciphers, such as the Vigenère cipher, were introduced to enhance security [3]. The Vigenère cipher utilized multiple cipher alphabets based on a keyword, making it significantly more resistant to brute force and other cryptographic

attacks. This innovation allowed for more robust encryption and laid the groundwork for more complex cryptographic schemes. The 19th century witnessed a surge in telegraph systems, necessitating the development of stronger and more efficient encryption methods. With the emergence of electronic communication, the demand for secure transmission of sensitive information grew exponentially. Cryptographers and mathematicians began exploring new encryption techniques, including the use of mathematical algorithms and computational approaches to enhance the strength and efficiency of cryptographic security. In the early 20th century, the development of electromechanical devices like the Enigma machine during World War II marked a significant milestone in the evolution of cryptography. The Enigma machine, used by the German military, utilized rotors and complex wiring to perform substitution and transposition on letters, creating a vast number of possible cipher combinations[4]. Despite its initial appearance of being unbreakable, the code-breaking efforts at Bletchley Park, led by pioneers like Alan Turing, eventually deciphered the Enigma-encoded messages, providing critical intelligence to the Allies. These historical advancements laid the groundwork for the continuous evolution of cryptographic techniques into the modern era. With the advent of computers and digital communication, cryptographic schemes have become increasingly sophisticated, incorporating principles from mathematics, computer science, and information theory to provide stronger security and protect sensitive data in today's interconnected world. The journey of cryptography from ancient civilizations to the digital age reflects its enduring significance as a fundamental pillar of modern cybersecurity.

Limitations of Older Cryptographic Schemes

While ancient and classical cryptographic schemes were groundbreaking in their time, they exhibited inherent weaknesses that necessitated innovation. One of the notable vulnerabilities in classical cryptography was the reliance on letter frequency patterns. Frequency analysis, a powerful cryptanalytic technique, exploited the tendency of certain letters to appear more frequently in a language than others[5]. By analyzing the frequency distribution of letters in the encrypted message, attackers could deduce the probable substitution patterns used in the cipher. This insight enabled them to unravel the encryption and decipher the original message. For instance, in English, the letter "e" is the most commonly used letter, followed by "t", "a", and "o". An attacker conducting frequency analysis would observe the pattern of characters appearing most frequently in the encrypted text and correlate them with the known frequency distribution of English letters. By identifying high-frequency characters in the ciphertext, the attacker could make educated guesses about the corresponding letters in the plaintext, thereby unraveling the encryption process. This method was both the primary reason that the Vigenere cipher became popular, but also why it slowly became obsolete. Unlike the Ceasar Cipher, the Vigenère cipher initially posed a challenge to cryptanalysts due to its resistance to simple letter frequency analysis. However, as techniques evolved, frequency analysis became more sophisticated, enabling cryptanalysts to break the Vigenère cipher. By analyzing patterns in the frequency of letters in the ciphertext, cryptanalysts could deduce information about the keyword and plaintext, ultimately compromising the security of the Vigenère cipher. Furthermore, the reliance on single keys for both encryption and decryption in classical ciphers posed additional security risks. If an attacker managed to intercept or deduce the key used by the sender to encrypt the message, they could use the same key to decrypt all subsequent communications, compromising the confidentiality of the entire communication channel. This lack of key diversity and the limited key space made classical ciphers more susceptible to cryptanalysis, and it became evident that stronger cryptographic techniques were necessary to secure sensitive information effectively. The annals of history are replete with instances where the exploitation of cryptographic systems has led to significant, tangible impacts on global events, societies, and the very course of history itself. One of the most notable examples of cryptographic vulnerabilities having a tangible impact was the breaking of the Enigma code during World War II. The Enigma machine, a sophisticated piece of cryptographic hardware used by Nazi Germany, was considered to provide unbreakable encryption. The security of the Enigma machine was primarily based on the complexity of its settings, which could be configured in a multitude of ways, theoretically ensuring the confidentiality of German military communications. The Enigma machine's security, while formidable, was not infallible. A significant vulnerability lay in its operational use, particularly the practice of key repetition and predictable message formatting by its operators. This provided cryptanalysts with a foothold for cryptanalysis. Additionally, the capture of key Enigma codebooks and hardware by the Allies provided crucial insights into the machine's encryption processes. The Allied efforts to crack the Enigma code, led by a team of cryptanalysts at Bletchley Park, including Alan Turing, resulted in the successful decryption of vast amounts of military communications. This breakthrough allowed the Allies to anticipate and counter German military strategies effectively, significantly contributing to the Allied victory. The interception of German U-boat positions, for instance, was instrumental in turning the tide of the Battle of the Atlantic. Another illustrative case study of cryptographic exploitation is Operation Bernhard, a secret Nazi plan during World War II aimed at destabilizing the British economy through the mass counterfeiting of British banknotes. The operation exploited the vulnerabilities in the security features of the British banknotes at the time, which were not as sophisticated or as difficult to replicate as they are today. The lack of advanced cryptographic techniques in the authentication of banknotes made it possible for the counterfeit notes to

be produced with a high degree of accuracy. While Operation Bernhard did not achieve its ultimate goal of destabilizing the British economy, it led to a significant influx of counterfeit currency, which posed a serious threat to the economic stability of Great Britain. The operation forced the British government to undertake a comprehensive redesign of its currency to protect against further counterfeiting efforts. The realization of these vulnerabilities fueled the drive for innovation in cryptographic methods, leading to the development of more advanced encryption techniques such as the Data Encryption Standard (DES) and later, the Advanced Encryption Standard (AES)[6]. These modern encryption algorithms addressed the limitations of classical ciphers by introducing stronger key management practices, more complex mathematical operations, and increased key sizes, making them highly resilient against frequency analysis and other cryptanalytic attacks. The evolution of cryptographic schemes continues to this day, with ongoing research and development efforts aimed at staying ahead of malicious attackers and ensuring secure communication and data protection in the ever-changing digital landscape

Evolution of Modern Cryptographic Schemes

In today's digital age, cyber threats have become increasingly sophisticated and pervasive. Hackers, cybercriminals, and state-sponsored actors continuously develop new attack methods to exploit weaknesses in cryptographic algorithms. The ability to break cryptographic codes can result in data breaches, financial losses, and privacy violations. Older methods of security, unfortunately, can no longer handle modern cyber threats. As such, cryptography has evolved to stay ahead of these malicious actors, enhancing its security measures and adopting techniques that can sufficiently withstand advanced attacks.

Frequency Analysis

One of the major vulnerabilities in classical cryptography was the reliance on letter frequency patterns. Attackers could exploit this vulnerability by using frequency analysis, a powerful cryptanalytic technique. Frequency analysis relied on the observation that certain letters in a language appeared more frequently than others. By analyzing the frequency distribution of letters in an encrypted message, attackers could deduce the probable substitution patterns used in the cipher, ultimately unraveling the encryption and deciphering the original message. To mitigate this vulnerability, modern cryptographic techniques have been developed, such as advanced encryption algorithms like AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman). These algorithms rely on complex mathematical operations that make frequency analysis ineffective. They ensure that the encrypted messages have a uniform distribution of symbols, thereby rendering frequency analysis less viable. For example, AES employs an SPN structure, dividing the data into blocks and applying multiple rounds of substitution and permutation operations. Through these multiple rounds, AES introduces confusion and diffusion in the encryption process. Confusion ensures a complex relationship between the ciphertext and the key, making it difficult to discern patterns. Diffusion ensures that a change in one bit of the plaintext affects many bits in the ciphertext. On the other hand, RSA is based on asymmetric encryption, using a pair of public and private keys. The public key is used for encryption, and the private key is used for decryption. The security of RSA relies on the difficulty of factoring the product of two large prime numbers. The encryption and decryption processes involve complex mathematical calculations, making it computationally infeasible for attackers to reverse-engineer the private key from the public key. By maintaining a balanced distribution of symbols, these algorithms prevent any single character or symbol from revealing information about the original plaintext. Conversely, OTP authentication addresses the vulnerability of letter frequency patterns by simply introducing dynamic passwords that change with each login. Unlike static passwords, which can be easily cracked through frequency analysis, OTPs provide stronger protection against attacks. Each time a user logs in, they receive a new one-time password, making it more difficult for attackers to decipher the encryption patterns used[7].

Symmetric Encryption

Symmetric encryption, an ancient technique dating back to the earliest civilizations, involves the use of a single secret key for both encryption and decryption. Early examples include substitution ciphers like the Caesar cipher. These classical ciphers relied on a single key for both encryption and decryption. This lack of key diversity and limited key space posed additional security risks. If an attacker managed to intercept or deduce the key used by the sender to encrypt the message, they could use the same key to decrypt all subsequent communications, compromising the confidentiality of the entire communication channel. Over time, this created a need for more robust security in the digital age and prompted significant advancements in symmetric encryption algorithms. To overcome these issues, modern cryptographic techniques introduced the concept of key diversity and a larger key space. For instance, symmetric cryptography uses a single key for both encryption and decryption, but this key is securely distributed and frequently changed, enhancing the security of the communication. The development of the Data Encryption Standard (DES) in the 1970s marked a significant milestone in symmetric encryption. DES adopted a Feistel network structure, dividing the data into blocks and applying multiple rounds of encryption using a subkey derived from the main secret key[8]. However, as computing power increased, DES became susceptible to brute-force attacks due to

its small key size of only 56 bits. To address this limitation, the Advanced Encryption Standard (AES) was introduced in 2001 as a replacement for DES. AES employs a substitution-permutation network (SPN) structure and supports key sizes of 128, 192, or 256 bits, significantly enhancing its security compared to DES. The SPN structure involves multiple rounds of substitution and permutation operations, providing a high level of confusion and diffusion in the encryption process. AES has since become the widely accepted standard for symmetric encryption, widely used in securing data, communication, and cryptographic applications worldwide. However, it is not without its drawbacks. Weak keys or IVs are a common vulnerability in AES encryption. Weak keys can make encryption vulnerable to attacks, while weak IVs can lead to predictable ciphertexts. Additionally, the main risk to AES is side-channel attacks. In these attacks, attackers try to pick up information leaking from a system to discover how the encryption algorithms work. This can only happen in non-secure systems. Because AES is meant to have easy accessibility, it is very susceptible to human error. With the rise of computational power and potential threats, symmetric encryption algorithms have further evolved to adapt to modern security demands. Block cipher modes of operation, such as Cipher Block Chaining (CBC), Electronic Codebook (ECB), and Counter (CTR), offer additional layers of security and data integrity. For instance, CBC mode XORs each plaintext block with the previous ciphertext block, introducing a dependency that ensures each block is encrypted differently. On the other hand, CTR mode transforms the block cipher into a stream cipher, enabling parallel encryption and decryption while preserving confidentiality and authenticity [9].

Asymmetric Encryption

Public key cryptography, also known as asymmetric cryptography, revolutionized the field of cryptography when it emerged in the 1970s in the work of Diffie-Hellman [10].The Diffie-Hellman key exchange is a foundational cryptographic protocol that has revolutionized secure communication over untrusted networks. Its brilliance lies in its ability to establish a shared secret key between two parties without ever transmitting the key itself. Instead, both parties independently generate their public and private keys based on agreed-upon parameters. These parameters consist of a large prime number p and a primitive root modulo p, denoted as g. The beauty of this process is that even if an eavesdropper intercepts the public keys exchanged during the communication setup, they cannot easily derive the shared secret key without knowledge of the private keys. The security of Diffie-Hellman relies on the presumed computational difficulty of solving the discrete logarithm problem. This mathematical challenge involves determining the private key (a or b) from the public key (A or B) and the agreed-upon parameters (p and g). This ability represented a massive shift in the use of cryptographic technology throughout various sectors. For example, in the IT security sector, asymmetric encryption plays a crucial role in SSL/TLS protocols, ensuring secure key exchange between users' browsers and websites. Another innovative application of asymmetric encryption is in securing email communication. By using public and private keys, asymmetric encryption allows users to send encrypted messages without the need to share a secret key, overcoming key distribution challenges effectively. Furthermore, asymmetric encryption has significantly impacted the finance industry, particularly in online banking and digital transactions. By leveraging public-key cryptography, banks and financial institutions can securely exchange sensitive information over the internet, such as account details and transaction data, without the risk of interception or tampering. For sufficiently large prime numbers and well-chosen parameters, this problem is computationally infeasible to solve efficiently. As such, Diffie-Hellman forms the basis for secure key exchange in numerous cryptographic applications, ensuring the confidentiality and integrity of data exchanged over the internet and other communication networks [11]. However, as there is no authentication involved, it is vulnerable to man-in-the-middle attack, and, as it is computationally intensive, it is expensive in terms of resources and CPU performance time. Because of these limitations of the basic method, the asymmetric methology was forced to evolve to become more efficient and safe. The defining feature of public key cryptography is the use of a key pair - a public key and a private key. The public key is freely available and used for encryption, while the private key is kept secret and used for decryption. This novel approach enabled secure communication without the need for a shared secret key between the sender and recipient. The breakthrough in public key cryptography came with the introduction of the RSA algorithm in 1977 by Rivest, Shamir, and Adleman (E. Milanov, "The RSA algorithm - University of Washington," Jun, 2009). RSA is based on the mathematical properties of large prime numbers and involves complex computations that make it computationally infeasible to reverse engineer the private key from the public key. The security of RSA relies on the difficulty of factoring the product of two large prime numbers, making it a highly secure and widely adopted encryption algorithm. One of the most significant advantages of public key cryptography is its ability to address the key distribution problem that was a major challenge in symmetric encryption. In symmetric encryption, a single secret key is shared between the sender and recipient, requiring a secure channel for exchanging this key. However, securely distributing the secret key over an insecure channel posed considerable difficulties. Public key cryptography provides an elegant solution to this problem by allowing the public distribution of encryption keys without compromising the security of the private decryption keys. This breakthrough has revolutionized secure communication and data protection, laying the foundation for modern cryptographic applications in various

domains. The advancement of public key cryptography has led to the development of other widely-used algorithms, such as the Elliptic Curve Cryptography (ECC). ECC offers the same level of security as RSA but with significantly shorter key lengths, making it more suitable for resource-constrained environments. Moreover, public key cryptography forms the backbone of digital signatures, enabling authentication and non-repudiation in electronic transactions and document verification. The continuous evolution of public key cryptography has been instrumental in bolstering data security, enabling secure e-commerce, secure communication, and the establishment of trust in the digital landscape.

One-Time Pad (OTP)

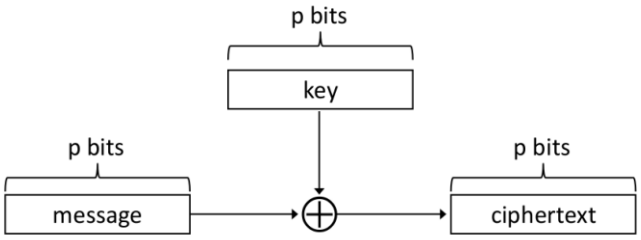

Fig. 1 A basic overview of the One-Time Pad. Image recreated from [7]

Origins

The One-Time Pad (OTP) is a symmetric encryption technique known for its perfect security. It involves the use of a random key that is as long as the plaintext, ensuring that the key is used only once. The key and the plaintext are combined using bitwise XOR (exclusive OR) to produce the ciphertext. Since the key is truly random and used only once, OTP provides perfect secrecy, meaning that the ciphertext does not reveal any information about the plaintext without the knowledge of the key[7]. The concept of the OTP dates back to the Vernam Cipher, proposed by Gilbert Vernam in 1917. OTP is a theoretically unbreakable cipher that provides perfect secrecy. It combines a secret key and a plaintext using exclusive or. The result is a cipher that doesn't contain any information about the plaintext. Its theoretical security relies on the key being truly random, perfectly secret, and used only once. This concept intrigued cryptographers and theorists due to its potential for achieving perfect secrecy, which means that an adversary with unlimited computational power cannot derive any information about the plaintext from the ciphertext, even if they have intercepted multiple ciphertexts.

Operation

The security of the OTP is based on the fundamental principles of information theory and the fact that each key is used only once. To encrypt a message using OTP, both the sender and the receiver must have a copy of the same random key, which is kept secret and never reused for any other message. The sender then XORs each bit of the plaintext message with the corresponding bit of the key to create the ciphertext. The resulting ciphertext appears random and offers no clue about the original message without knowledge of the specific key. In a perfect case like this where a key is truly random and kept secret, and used only once for encrypting a particular plaintext message, the relationship between the plaintext and ciphertext is entirely unpredictable. As a result, without knowledge of the specific key, it is computationally infeasible to decipher the ciphertext to recover the original plaintext.

To construct an OTP message, first, fix an integer $l > 0$. The message space M, key space K, and ciphertext space C are all equal to $\{0,1\}^l$ which can be seen in Figure 1.

1. Gen: the key-generation algorithm chooses a key from $K = \{0,1\}^l$ according to the uniform distribution (i.e., each of the 2^l strings in the space is chosen as the key with probability exactly 2^{-l}).
2. Enc: given a key $k \in \{0,1\}^l$ and a message $m \in \{0,1\}^l$, the encryption algorithm outputs the ciphertext $c := k \oplus m$.
3. Dec: given a key $k \in \{0,1\}^l$ and a ciphertext $c \in \{0,1\}^l$, the decryption algorithm outputs the message $m := k \oplus c$[11].

Significance and Evolution

However, in practice, achieving the ideal conditions for OTP can be challenging. Generating truly random and unpredictable keys can be difficult, and securely distributing and managing the keys for each communication session is a logistical challenge. Additionally, the key must be as long as the plaintext, which means that it can become impractical for large volumes of data. Moreover, securely exchanging and managing large random keys for every communication session can be cumbersome, especially in real world applications with multiple users and frequent data exchange[11].

Despite its perfect security, the OTP has limited practicality in most real-world scenarios. Digital versions of one-time pad ciphers have been used by nations for critical diplomatic and military communication, but the problems of secure key distribution make them impractical for most applications. Only in areas where the utmost security is necessary do we see any real world application of OTP. Because the security of OTP lies in its longer key lengths and theoretical perfect randomness, it is basically impossible to make OTP practical for widespread real-life use. While it remains an important theoretical concept

in cryptography, its widespread use is often impractical in modern communication systems, where more efficient and practical encryption methods are employed. Nonetheless, the OTP serves as a significant milestone in the history of cryptography, illustrating the fundamental principles of perfect secrecy and inspiring further advancements in cryptographic techniques.

Data Encryption Standard (DES)

Origins

The origins of DES can be traced back to the 1970s when computers were becoming increasingly prevalent in various applications. With the growing reliance on electronic communication and the need to safeguard sensitive information, the demand for a standardized encryption algorithm without the limitations shown in OTP became evident. In response, IBM developed the Data Encryption Standard, which was later adopted by the U.S. government as an official encryption standard in 1977. DES was a pioneering effort in developing a standardized cryptographic algorithm for widespread use.

Operation

DES operates on fixed-size blocks of plaintext data, typically 64 bits in length, and uses a 56-bit secret key for encryption and decryption as shown in Figure 2. The encryption process involves multiple rounds of permutation, substitution, and mixing operations to transform the plaintext into ciphertext. At the core of DES lies the Feistel network, a structure that ensures its reversible nature, allowing encryption and decryption to be carried out using the same algorithm.

Initial Permutation (IP): The initial permutation step rearranges the bits of the plaintext according to a predefined permutation table, introducing a form of confusion to the data. This step ensures that each bit of the input has an impact on multiple bits of the output, enhancing the diffusion properties of the cipher.

Feistel Network: The Feistel network consists of multiple rounds (16 rounds in the case of DES). During each round, the 64-bit data block is split into two halves, the left and right halves. The right half undergoes a series of operations, including expansion using E-boxes, bitwise XOR with the subkey, substitution using S-boxes, and a permutation known as the P-box. The S-boxes are a set of fixed tables used for substitution. The E-box is responsible for expanding the 32-bit data block to 48 bits. During each encryption round, the 32-bit data block is expanded through permutation to create a 48-bit intermediate block. This step introduces diffusion, spreading the influence of each bit across a larger portion of the data, further enhancing the cryptographic strength of DES. The P-box is then used to rearrange the bits of the 32-bit data block after the S-box substitution step in each round. The P-box ensures that the output from the S-boxes is mixed in a specific manner, adding additional confusion and making the encryption process more complex. The permutation in the P-box is a fixed mapping defined by the DES algorithm.In DES, there are eight S-boxes, each taking 6 bits of data as input and producing 4 bits of output. These S-boxes introduce non-linearity into the encryption process, making it harder for attackers to detect patterns in the ciphertext. The specific values in the S-boxes are carefully chosen during the DES algorithm design to increase the security and resistance to attacks.

In DES, a total of 16 rounds of this function occur, comprised of the following steps:

1. The output of the IP is taken as an input for the Feistel function, let's call it X.
2. X is divided into two parts with 32 bits each – the LPT and RPT.
3. Subkeys (K_1, K_2, K_n, etc.) are derived from the original key, and one key is applied during each round.
4. The result (f) of LPT is carried out using RPT and $K_2 - f(RPT, K_1)$.
5. Mathematical function XOR is performed on $f(RPT, K_1)$ and LPT, resulting in L_1.
6. L_1 is swapped with RPT. So, the next function is performed on RPT, resulting in L_2.

Final Permutation (FP): The final permutation is the reverse of the initial permutation, rearranging the bits of the ciphertext to produce the final encrypted output [11].

Significance and Evolution

Upon its introduction, DES was considered highly secure and served as the cornerstone of encryption for many years. However, as computing power increased, the 56-bit key length of DES was found to be vulnerable to brute-force attacks. With advancements in cryptanalysis, it became evident that DES needed to be replaced by more robust encryption algorithms to withstand modern attacks.

Advanced Encryption Standard (AES)

Origin

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm that has become the gold standard for securing sensitive data in modern cryptographic applications. Developed in the late 1990s, AES was selected by the U.S. National Institute of Standards and Technology (NIST) to replace the aging Data Encryption Standard (DES) due to its robustness and

efficiency. AES operates on fixed-size blocks of data and uses a variable key size, providing a high level of security against various cryptographic attacks.

Operation

AES supports three key sizes: 128 bits, 192 bits, and 256 bits, which offer increasing levels of security. The length of the key affects the key schedule (i.e., the sub-key that is used in each round) as well as the number of rounds, but does not affect the high-level structure of each round. In contrast to DES, which uses a Feistel structure, AES is essentially a substitution-permutation network. The encryption process involves a series of rounds as can be seen at Figure 2, each comprising distinct operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. These operations contribute to AES's strength by introducing confusion and diffusion, making it highly resistant to attacks.

1. In the AddRoundKey step, a 128-bit sub-key is derived from the master key, and is interpreted as a 4-by-4 array of bytes. The state array is updated by XORing it with this sub-key. This key mixing step ensures that the encryption process is dependent on the specific round key being used.
2. In the SubBytes step, each byte of the data block is substituted using a fixed S-box. The S-box is a predefined lookup table that introduces non-linearity into the encryption process, preventing attackers from deducing patterns in the ciphertext. It is calculated using a bijection of $\{0,1\}^8$.
3. In the ShiftRows step, the bytes in each row of the state array are shifted to the left as follows: the first row of the array is untouched, the second row is shifted one place to the left, the third row is shifted two places to the left, and the fourth row is shifted three places to the left. All shifts are cyclic so that, e.g., in the second row the first byte becomes the fourth byte.
4. In the *MixColumns* step, an invertible transformation is applied to the four bytes in each column. This transformation has the property that if two inputs differ in $b > 0$ bytes, then applying the transformation yields two outputs differing in at least $5 - b$ bytes. This step further enhances AES's security by combining the bytes of each column using a matrix multiplication operation.

This process further obscures the relationships between the bytes, making it challenging for attackers to retrieve the original data without the correct key [11].

Significance and Evolution

The strength of AES lies in its well-designed algorithm and its ability to withstand various attacks, including brute force attacks, differential cryptanalysis, and linear cryptanalysis. Its variable key size and efficient execution on modern computing devices make it an ideal choice for a wide range of applications, from securing data in transit over the internet to protecting sensitive information on personal devices, making AES one of the forefront cryptographic schemes in the current day. While AES is generally regarded as a highly secure encryption standard, however, there are always ongoing discussions and improvements within the field of cryptography. For example, AES-192 and AES-128 are not considered quantum resistant due to their smaller key sizes. AES-192 has a strength of 96 bits against quantum attacks and AES-128 has 64 bits of strength against quantum attacks, making them both insecure.

RSA (Rivest-Shamir-Adleman)

Origin

Unlike AES, DES, and OTP, RSA, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, is an asymmetric encryption algorithm that revolutionized public-key cryptography in the late 1970s. RSA is based on the mathematical properties of large prime numbers and relies on the computational difficulty of factoring the product of two large primes. The RSA algorithm is a type of asymmetric encryption that uses two different but linked keys. It is by far the most widely used asymmetric cryptosystem in the world and is used for securing communication in various domains, from cell phone communication to online banking.

Operation

In RSA, the key pair consists of a public key (e, n) and a private key (d, n), both generated from large prime numbers. The public key is used for encryption, while the private key is used for decryption. The security of RSA is based on the infeasibility of factoring the modulus n, which is the product of two large primes. The encryption process in RSA involves converting the plaintext message into a numerical value and raising it to the power of the public exponent e, modulo n. The result is the ciphertext, which can only be decrypted using the private key. The decryption process in RSA involves raising the ciphertext to the power of the private exponent d, modulo n. This operation recovers the original plaintext message. To summarize, the RSA algorithm could be calculated using the following steps:

1. Choose two large prime numbers, p and q.
2. Calculate $n = p \times q$.
3. Calculate $z = (p-1)(q-1)$.
4. Choose a number e where $1 < e < z$.

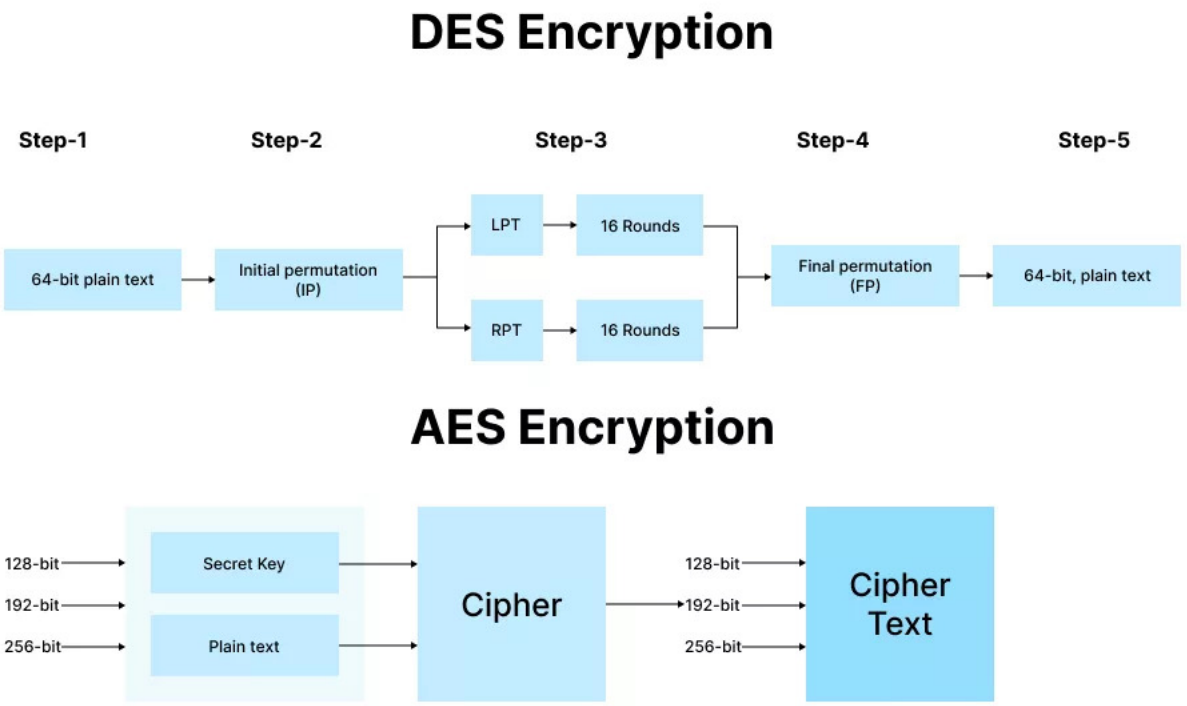

Fig. 2 A showcase of the differences between AES and DES. Image recreated from [12]

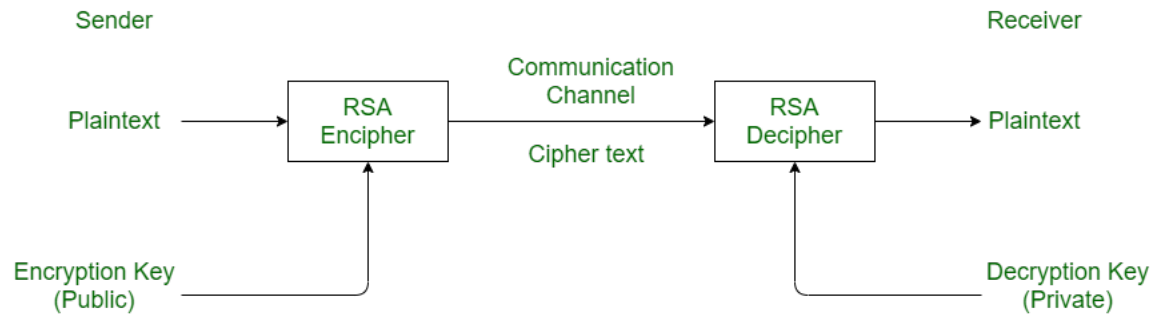

Fig. 3 An explanation of asymmetric RSA encryption. Image recreated from [13]

5. Calculate $d = e^{-1} \mod z$.

6. Bundle the private key pair as (n, d).

7. Bundle the public key pair as (n, e).

Significance and Evolution

RSA has been widely used for secure communication, digital signatures, and key exchange. However, as computing power has increased, the security of RSA has become vulnerable to attacks like the General Number Field Sieve (GNFS) algorithm, which can factor large integers efficiently. The GNFS algorithm employs sophisticated mathematical techniques and optimiza-

tions, allowing it to handle very large numbers effectively. It operates by creating an appropriate number field and then finds suitable elements within that field that lead to the factorization of the modulus. GNFS has been instrumental in breaking several RSA keys with large key sizes, highlighting the vulnerability of RSA to this advanced factoring algorithm. Additionally, RSA is perhaps the most vulnerable of the four major algorithms to the rise of quantum computing. Quantum computing can break RSA encryption by using Shor's algorithm to factor large numbers quickly. RSA encryption's security is based on the difficulty of factoring large products of prime numbers. Quantum computers can factor large numbers much faster than classical computers. This can break RSA encryption by finding the private key from the public key. To address this, researchers are exploring alternative approaches and larger key sizes to enhance RSA's security such as increasing key size. Additionally, hybrid cryptosystems combining RSA with symmetric encryption algorithms like AES have become popular, offering guaranteed security with the efficiency of symmetric encryption. Despite its challenges, RSA remains a fundamental and widely deployed cryptographic algorithm, playing a crucial role in securing the digital world and protecting sensitive information. As researchers continue to explore innovative techniques and stronger key sizes, RSA's legacy as a cornerstone of public-key cryptography endures in the ever-evolving landscape of information security.

Elliptic Curve Cryptography (ECC)

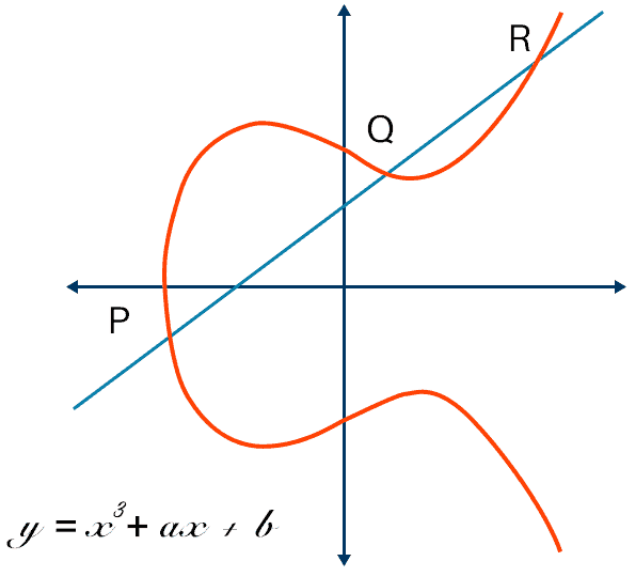

Fig. 4 A diagram of an elliptic curve [14]

Origin

Elliptic Curve Cryptography (ECC) is an asymmetric encryption algorithm that offers strong security with shorter key lengths compared to traditional public-key cryptography schemes like RSA. ECC is based on the mathematical properties of elliptic curves, which are a group of points that satisfy a specific mathematical equation as shown in Figure 4. These curves have unique properties that make them suitable for cryptographic operations.

Operation

The foundation of ECC lies in the concept of elliptic curves over finite fields. In simple terms, an elliptic curve can be represented by an equation of the form $y^2 = x^3 + ax + b \mod p$, where a and b are constants, and p is a prime number [15]. A finite field, also known as a Galois field, is a mathematical structure that consists of a finite set of elements along with two binary operations, usually addition and multiplication. The elements in a finite field are typically represented by integers modulo a prime number, which for this example would be p. The curve's points, together with an additional point at infinity, form an additive group, enabling various cryptographic operations.

In ECC, the key pair consists of a private key and a corresponding public key, both derived from points on an elliptic curve over the finite field. The security of ECC is based on the intractability of the elliptic curve discrete logarithm problem, which involves finding the private key from the public key. Because cryptographic operations for ECC are performed over a finite field, its key sizes are generally smaller than that of other schemes, making it particularly advantageous in resource-constrained environments, such as mobile devices and IoT devices, where computational power and memory are limited.

The key operations in ECC involve scalar multiplication and point addition on the elliptic curve. Scalar multiplication is the process of multiplying a scalar (the private key) with a point on the curve to obtain another point (the public key). Point addition is used for combining two points on the curve to obtain a third point, which is crucial for generating the shared secret in key exchange protocols like Elliptic Curve Diffie-Hellman (ECDH).

Significance and Evolution

ECC has gained widespread adoption in various cryptographic applications due to its strong security and efficiency. Its shorter key lengths result in faster computation, reducing the computational overhead and power consumption in cryptographic operations. As technology advances and the need for secure communication grows, ECC continues to be a prominent choice for securing data and ensuring the confidentiality and integrity of sensitive information.

Contextual Analysis of OTP, RSA, AES, and ECC

In the ever-evolving landscape of data security and digital communication, cryptographic schemes play a pivotal role in safeguarding sensitive information and ensuring secure transactions. Understanding the diverse contexts in which modern cryptographic schemes excel is of paramount importance for optimizing their application and mitigating potential risks. It is particularly important to understand the four major cryptographic schemes, OTP, RSA, AES, and ECC, delving into their respective strengths and effectiveness in specific scenarios. By discerning the contextual advantages of these cryptographic techniques, we can make informed decisions regarding their deployment, paving the way for guaranteed data protection and secure communication across various domains. OTP is particularly useful in scenarios where maintaining the confidentiality and integrity of user accounts is critical, such as superencryption, numbers stations, quantum key distribution, or diplomatic communications. By generating a new password for each login attempt, OTP mitigates the risk of password reuse or exposure, significantly reducing the chances of unauthorized access to sensitive information. The strength of OTP lies in its unpredictability. The passwords are generated based on a combination of factors, such as a user's unique identifier, a secret key, and a timestamp. This ensures that each OTP is different from the previous one. However, since generating truly random and unpredictable keys can be difficult, and securely distributing and managing the keys for each communication session is a logistical challenge, OTP is very difficult to use in situations where confidentiality and security are not the foremost issue and its inefficiency could be excused. Additionally, the key must be as long as the plaintext, which means that it can become impractical for large volumes of data. RSA emerges as the perfect choice for secure key exchange and authentication. In scenarios where maintaining the confidentiality and authenticity of transmitted information is paramount, RSA's asymmetric encryption ensures that messages can be safely exchanged between parties without the need for a shared secret key. Its digital signatures further add an essential layer of non-repudiation, providing indisputable evidence of message authenticity and source verification. As a result, RSA finds extensive utilization in a plethora of domains, ranging from secure email communication, online transactions, and secure web browsing to the protection of sensitive information during digital interactions. Understanding the exceptional capabilities of RSA in these contexts empowers organizations and individuals to instill trust and confidence in their communication channels, bolstering data security and mitigating potential threats. On the other hand, the exploration of AES, as a symmetric encryption algorithm, reveals its unparalleled efficiency and speed in bulk data encryption. The symmetric nature of AES allows for faster processing of data, making it particularly suitable for resource-intensive applications where swift and seamless encryption and decryption are of utmost importance. AES's robustness and reliability have led to its widespread adoption as the standard encryption algorithm in various cryptographic applications. From securing data in cloud storage and protecting sensitive information during file transfers to safeguarding confidential communications in modern networking systems, AES proves to be a workhorse in the realm of data protection and secure transmission. Understanding how AES excels in these contexts equips stakeholders with the knowledge needed to make informed decisions on its appropriate deployment, enabling them to harness its potential to its fullest extent. The unique strength of Elliptic Curve Cryptography (ECC) in asymmetric encryption with shorter key lengths. By leveraging the mathematical properties of elliptic curves, ECC offers the same level of security as traditional algorithms like RSA but with significantly reduced key sizes. This has led to the integration of ECC in Internet of Things (IoT) devices, where its ability to provide high-level security with relatively small key sizes is particularly advantageous. For example, in smart home systems, ECC is employed to secure the communication between various smart devices and the central control unit. This not only ensures the confidentiality and integrity of the data exchanged but also minimizes the computational load on devices that often have limited processing power and energy resources. ECC has also been pivotal in securing transactions and data exchange. Financial apps, for instance, leverage ECC to protect user transactions even on devices with limited battery life and processing capabilities. This is achieved through ECC's efficient algorithms, which require less computational power compared to traditional cryptographic methods, thereby extending the battery life of mobile devices while maintaining a high level of security. As the demand for secure communication and data protection grows in interconnected devices and applications, ECC becomes increasingly valuable in meeting these challenges effectively.

Conclusion

The history of cryptography has unfolded over millennia, reflecting the ongoing imperative to secure sensitive information. From the ancient Caesar cipher to the Renaissance Vigenère cipher and 19th-century telegraph encryption methods, each era witnessed incremental advancements in the field. However, these historical cryptographic techniques, while serving their respective purposes, shared common limitations that became increasingly pronounced as technology and communication systems advanced. One notable limitation of these older schemes was their vulnerability to relatively simple attacks. For instance, a common limitation of older cryptographic schemes was their susceptibility to brute force attacks. Brute force attacks, characterized by their exhaustive search of all possible keys, posed a significant limitation to the security of antiquated schemes. These old cryptographic methods, often characterized by short

key lengths and limited complexity, were susceptible to brute force attacks due to their constrained key spaces. For instance, attackers employing brute force tactics could systematically try all possible keys within a relatively short amount of time, exploiting linguistic patterns or character repetitions present in these ciphers. This vulnerability becomes even more pronounced as computational power increases, allowing adversaries to expedite the key search process. The weakness of these ancient cryptographic schemes could also be seen by their weakness to frequency analysis. Frequency analysis could be employed to break simpler ciphers, exploiting patterns in the language or the repetition of certain characters. This susceptibility posed a significant challenge to the confidentiality of encrypted messages and emphasized the need for more robust cryptographic methods. Furthermore, these early cryptographic techniques predominantly relied on shared secret keys for both encryption and decryption processes. While this approach was suitable for many historical contexts, it introduced inherent vulnerabilities. The necessity of securely distributing and maintaining these keys presented logistical challenges, as compromised keys could lead to the complete compromise of encrypted information. As societies progressed and communication networks expanded, the limitations of these older cryptographic methods became increasingly evident. The advent of more sophisticated adversaries and advanced computing capabilities highlighted the need for cryptographic techniques that could withstand more rigorous scrutiny and attacks. It could be said that modern cryptographic schemes represent the pinnacle of this evolution, tailored to address contemporary security challenges. These schemes offer unique attributes that make them indispensable in today's digital landscape. OTP remains unmatched in providing perfect security, particularly for scenarios prioritizing confidentiality and integrity. However, securely managing unpredictable keys can pose practical challenges. The crux of OTPs security lies in the generation of truly random and unpredictable keys, often referred to as "pads". These pads must be as long as the plaintext being encrypted, and each pad should be used only once to maintain the system's integrity. The challenge arises in generating these truly random keys, as any discernible patterns or repetitions in the keys can potentially be exploited by adversaries. Furthermore, securely distributing these lengthy and unique keys to both the sender and recipient in a timely and secure manner can be logistically cumbersome, especially in large-scale or long-term communication scenarios. Quantum Random Number Generators (QRNGs) and Quantum Key Distribution (QKD) show promising strides towards overcoming OTP's key generation and secure distribution challenges. However, the high costs, technological barriers, and operational complexities associated with these quantum technologies have limited their widespread adoption. QRNGs, while capable of generating truly random keys essential for OTP, are expensive and complex to integrate into existing systems. Similarly, QKD, known for its potential to securely distribute keys, is hampered by limitations in transmission distance and the need for specialized equipment, making it impractical for global communication networks. Furthermore, the logistical challenges of managing the vast amounts of random data required for OTP, ensuring keys are never reused, and the secure storage of keys, add layers of complexity that hinder the practical application of OTP encryption. Despite the theoretical allure of OTPs perfect secrecy, these practical impediments have kept it from becoming a viable option for everyday encrypted communications. The gap between the theoretical promise of OTP and its real-world applicability remains wide, with current technological and logistical solutions falling short of making OTP a practical reality for secure communication on a broad scale. In contrast to OTP, the Data Encryption Standard (DES) offered a practical and widely deployable encryption solution, by sacrificing only a small fraction of the security guaranteed. It ensured data security and confidentiality through its symmetric key encryption, making it suitable for a range of applications, from securing electronic communications to protecting sensitive data at rest. Nevertheless, DES did come with limitations, most notably its relatively short key length, which can render it vulnerable to modern computing capabilities. This vulnerability has prompted the evolution of DES into more robust encryption standards like the Advanced Encryption Standard (AES). As a symmetric encryption algorithm, AES shines in bulk data encryption, offering unparalleled efficiency and speed. It is well-suited for resource-intensive applications where swift encryption and decryption are vital. Its widespread adoption has established it as the global standard for data, communication, and cryptographic applications. While AES excels in numerous aspects, it does come with its set of limitations. AES's efficiency and speed, while advantageous for bulk data encryption, can also be a limitation in certain contexts. In situations where more complex cryptographic operations or key management protocols are required, AES's streamlined design may not align with the specific security needs, potentially necessitating the use of more specialized cryptographic schemes. On the other hand, RSA is an asymmetric encryption algorithm and excels in secure key exchange and authentication. It ensures data confidentiality and authenticity without relying on shared secrets. Its use of digital signatures further enhances data security, enabling trust and confidence in communication channels. However, like any cryptographic scheme, it is not without its limitations. One of the primary limitations of RSA is its computational overhead. RSA operations involve complex mathematical calculations, particularly for key generation and encryption, which can be resource-intensive. This computational cost can be a challenge in scenarios where efficiency and speed are critical, such as high-frequency financial transactions or resource-constrained devices like IoT sensors. As a way to almost offset the limitations of RSA, ECC provides security equivalent to traditional algorithms like RSA but with signif-

icantly smaller key sizes. This computational efficiency suits resource constrained environments like mobile devices and IoT. ECC's importance continues to grow in meeting the demands of secure communication and data protection. As we stand on the brink of unprecedented technological advancements, the field of cryptography faces a future full of challenges and opportunities for innovation. Quantum computing, with its ability to process information in fundamentally new ways, has the potential to break many of the cryptographic algorithms currently relied upon for secure communication. Algorithms such as RSA and ECC, which form the backbone of digital security, could be rendered obsolete by quantum algorithms capable of solving complex mathematical problems, like factoring large numbers or computing discrete logarithms, in a fraction of the time taken by classical computers. The potential for quantum computers to decrypt information previously considered secure necessitates a paradigm shift in cryptographic practices. However, just as challenges have spurred innovation in the past, the threat of quantum computing is catalyzing the development of post-quantum cryptography (PQC). This emerging field focuses on designing cryptographic algorithms that are secure against both classical and quantum computing threats, ensuring the longevity and integrity of digital security mechanisms. PQC represents the next frontier in the cryptographic arms race, embodying the field's proactive stance against evolving technological capabilities. Moreover, the future of cryptography is not solely defined by the quest to mitigate quantum threats. Emerging technologies such as blockchain and distributed ledger technology present new applications and challenges for cryptographic methods, pushing the boundaries of what is possible in terms of decentralized security and trust. Similarly, the growing interconnectedness of devices in the Internet of Things (IoT) ecosystem raises complex issues around cryptographic scalability, energy efficiency, and the seamless integration of security protocols into a myriad of devices. As cryptography navigates these future challenges, its evolution will undoubtedly reflect the lessons learned throughout its storied history. The field has continually adapted to technological advancements, shifting paradigms, and emerging threats, always with the aim of safeguarding information in an ever-changing world. From the ciphers of ancient times to the quantum-resistant algorithms of the future, cryptography's journey is one of perpetual innovation, driven by the dual imperatives of security and adaptability.

Compliance with Ethical Standards

Ethical Approval: This research has been conducted in accordance with the ethical standards and guidelines set forth by the International Journal of Information Security.

Informed Consent: No participants were involved in this study.

Conflicts of Interest: The author declares that they have no conflicts of interest related to this research.

Competing Interests

The author declares that they have no competing interests that could influence the research findings or the content presented in this manuscript.

Research Data Policy and Data Availability Statements

Data Availability: The datasets and materials used or analyzed during the current study are available from the corresponding author upon reasonable request.

Data Sharing: Researchers interested in accessing the data used in this study may contact the corresponding author to request access.

Data Privacy and Security: We affirm that all data used in this study have been handled in compliance with data privacy and security regulations. Personal information has been anonymized or de-identified to protect the privacy of individuals.

References

1 W. Kotas, *A Brief History of Cryptography*.

2 D. Luciano and G. Prichett, *The College Mathematics Journal*, **18**, 2–17,.

3 A.-A. Aliyu and A. Olaniyan, *International Journal of Computer Applications*, **135**, 46–50.

4 N. Smart and N. Smart, *Cryptography Made Simple*, 133–161 16.

5 G. Stoneburner, A. Goguen and A. Feringa, *Risk management guide for information technology systems - hhs.gov*, Risk Management Guide for Information Technology Systems.

6 A. Hamza and B. Kumar, 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART, p. 333–338.

7 F. Rubin and S. Singh, *Cryptologia*, **20**, 359–364.

8 S. Fatima, T. Rehman, M. Fatima, S. Khan and M. Ali, *Comparative analysis of AES and RSA algorithms for Data Security in cloud computing*.

9 R. Bhanot and R. Hans, *Review and Comparative Analysis of various encryption algorithms*.

10 W. Diffie and M. Hellman, *New Directions in Cryptography*, Association for Computing Machinery, New York, NY, USA, 1st edn, p. 365–390.

11 J. Katz and Y. Lindell, *Introduction to modern cryptography*, CRC Press/Taylor Francis.

12 J. Mehta and E. Milanov, *Jun*.

13 pppankaj, *Difference between RSA algorithm and DSA*.

14 Analytic, *Elliptic curve cryptography explained*.

15 V. Kapoor, V. Abraham and R. Singh, *Ubiquity*, **2008**, 1–8.