ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

Comparative Analysis of Cryptographic Algorithms and the Impact of Quantum Computing on Cybersecurity

Mahmidul Hasan[1]; Arifur Rahaman[2*]; Yeakob Ali[3]; Sabrina Tasnim[4]

[1,2,4]Department of Computer Science and Engineering, Sonargaon University, Bangladesh
[2,3]Department of Computer Science and Engineering,
Bangladesh Army International University of Science & Technology (BAIUST), Cumilla, Bangladesh

Corresponding Author: Arifur Rahaman[2*]

Publication Date: 2025/10/08

Abstract: In today's world, information is an asset, and preserving that asset has become a challenge. And there is no alternative to cryptography for preserving this information. Cryptography is used to ensure data security, so that third parties cannot steal, change or modify any data. It converts information from plain text to cipher text with the help of various algorithms, which are not easily readable or make any sense to normal human beings. The cipher text can be decrypted if its algorithm can be reverse-engineered to solve the mathematical equation. Currently, due to the development of our technology, our computer processors are getting faster, as a result, ordinary computers are also getting more powerful. Quantum computers are completing their work in a very fast time. Currently, Google has invented a chip "Willow". With this chip, they have solved a complex calculation or calculation (computing problem) in five minutes which would take conventional computer millions of years, and if artificial intelligence is added to it, then our information decoding and encoding algorithm will solve the equation in a few moments. As we know there are three branches of cryptography Symmetric, Asymmetric and Hash cryptography. Symmetric cryptography is currently very weak. The algorithms of the remaining two cryptographies will be very quickly compromised by quantum computers. This paper also highlights the key exchange, flexibility and security challenges of various cryptography, which determine the effectiveness of cryptographic systems, and the future state of cryptography will be discussed in comparison with quantum computers.

Keywords: *Cryptography, Symmetric, Asymmetric, Hash Algorithm, RSA, Diffie-Hellman, Blowfish, MD5, SHA-512, Cyber Security, Information Security, Data Protection.*

How to Cite: Mahmidul Hasan; Arifur Rahaman; Yeakob Ali; Sabrina Tasnim (2025) Comparative Analysis of Cryptographic Algorithms and the Impact of Quantum Computing on Cybersecurity. *International Journal of Innovative Science and Research Technology*, 10(10), 224-232. https://doi.org/10.38124/ijisrt/25oct096

I. INTRODUCTION

With the advancement of technology, cyber security has become an important part of our lives. Encryption and decryption methods are used to protect data on the Internet, which is the main topic of cryptography. Cryptography is a technique that transforms data, and it ensures security concepts such as data integrity, authorization, authentication, confidentiality and non-repudiation [1].

Cryptography is basically divided into three main branches: symmetric, asymmetric, and hash algorithms.

- *Symmetric Cryptography:*

In symmetric algorithms, encoding and decoding are done using the same key. It is relatively simple and fast but weak in terms of security. DES and AES are examples of such algorithms.

- *Asymmetric Cryptography:*

Asymmetric algorithms use two keys—public and also private keys. It is relatively uncomplicated and safe. Converting it from public key to private key is almost impossible, which makes it highly secure. RSA and Diffie-Hellman are popular examples of asymmetric cryptography.

- *Hash Cryptography:*

A hash algorithm is an irreversible method, where data is transformed in a way that cannot be decoded again. This is important for security, but threats can be created through wordlist and hash cracking. Among MD5, SHA-1, and SHA-512, SHA-512 is relatively secure, but MD5 is now much older and easier to decode.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

Below we provide a brief comparative analysis of different branches of cryptography and their algorithms, which will strengthen our understanding of the subject.

II. VARIOUS ALGORITHMS OF CRYPTOGRAPHY ALGORITHMS AND THEIR APPLICATIONS

A. DES (Data Encryption Standard):

DES is an ancient cryptographic algorithm invented by IBM in the 1970. It was originally intended to provide a secure encryption standard for commercial use, not a government project.

➢ *Applications of DES in Cryptography:*

DES was originally used to encrypt digital data. Its applications include:

- File Encryption: To protect important files from unauthorized users.
- Database encryption: To protect databases containing sensitive information.
- Communication Encryption: To protect data sent over a network.

➢ *Method of Converting Plain Text to Cipher Text:*

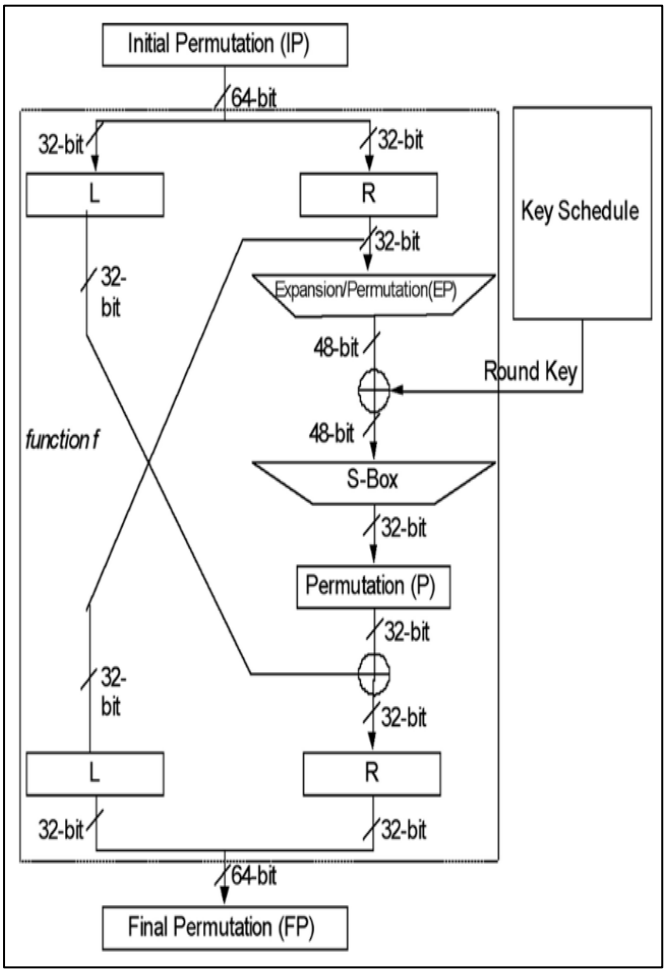

Fig 1 DES Process [9]

DES uses the same key for encryption and decryption. It works as a type of block cipher, meaning that it divides data into fixed-size blocks and encrypts each block separately.

- Initial Permutation: The bits of the plain text are rearranged into a specific pattern.
- Round function: This process is repeated 16 times. In each round, the data is split and processed through a complex series of substitution and permutation processes.
- Final Permutation: After 16 rounds, the final permutation is applied and the cipher text is generated.

➢ *DES Decryption Process:*

The decryption process is the opposite of the encryption process, as they convert ciphertext to plaintext using the same key.

➢ *Limitations of DES:*

- Key size: DES's 56-bit key size is too small for today's computing power. What is easily detected by brute force attack.
- Linear and Differential Cryptanalysis: These attacks further reduce the security of DES.

Due to the limitations of DES, it is not safe to use these days. Its successor, AES (Advanced Encryption Standard), is much more secure due to its larger key size and more complex mathematical structure. [2]

B. Advanced Encryption Standard:

AES is a very strong and most widely used symmetric-key encryption algorithm. It is much more secure and efficient than its predecessor, DES (Data Encryption Standard).

➢ *Applications of AES in Cryptography:*

AES is widely used. It is used:

- File Encryption: To protect important files from unauthorized users.
- Database encryption: To protect databases containing sensitive information.
- Communication Encryption: To protect data sent over a network.
- VPN (Virtual Private Networks): To protect data sent over the Internet.
- HTTPS: To secure web traffic.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

➢ *Plain Text to Cipher Text Conversion:*

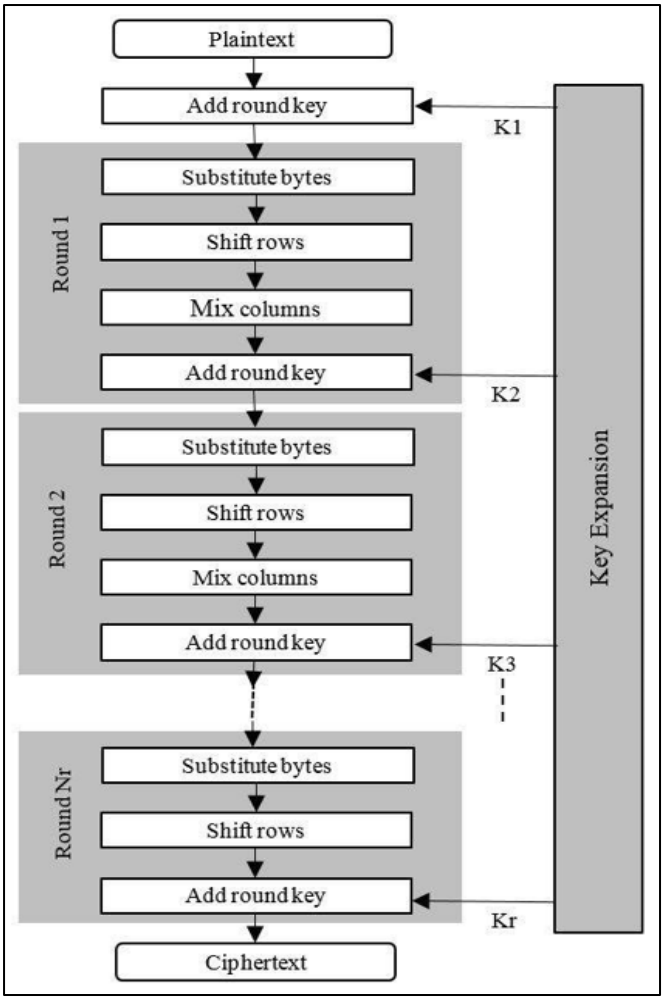

Fig 2 AES Process [10]

AES is one kind of block cipher, meaning it divides data into blocks of fixed size and encrypts each block individually. It is an iterative process, meaning it processes the data in multiple rounds. In each round, the data is transformed through a variety of arithmetic operations.

➢ *These Operations Include:*

- SubBytes: This is an operation where each byte in the block uses a lookup table and is replaced by another byte here.
- ShiftRows: In this operation, each row in the block is shifted to the left by a specified amount.
- MixColumns: Each column of the block is modified with an arithmetic function.
- AddRoundKey: The current block is XORed with a round key. These operations are repeated multiple times. The key used in each round is different.
- Advantages of AES Strong Security: AES is highly resistant to various types of cryptanalysis attacks. Efficiency: AES is relatively fast and efficient.
- Vulnerability: AES supports different key sizes, allowing users to select key sizes according to their own security needs.
- Widely used: AES is the most widely used symmetric key decryption and encryption algorithm worldwide.

➢ *Disadvantages and Limitations of AES:*

Although AES is a very powerful and widely used encryption algorithm, it also has some limitations.

- Computational complexity: High computational cost: Each round of AES involves various complex mathematical operations. As a result, it consumes a lot of computational resources, especially when using long keys.
- Hardware Dependency: Due to the high computational demand of AES, it usually relies on powerful processors or specialized hardware.
- Key management / key exchange: As a symmetric key encryption, AES requires key exchange in a secure manner.
- Key length: Although AES supports different key lengths, using longer keys increases the computational cost.
- Potential for future attacks: With the development of quantum computing, many cryptographic algorithms which is currently considered secure, including AES, may be broken in the future.
- Limitations of Use: Large Data Sets: AES may be less efficient for large data sets, especially when the data is constantly being updated.
- Embedded systems: The computational cost of AES can be excessive for embedded systems with limited resources.

Although AES is a very powerful and widely used encryption algorithm, it has some limitations. In the future, the development of quantum-resistant cryptography may help overcome the limitations of AES [3].

C. Rivest-Shamir-Adleman:

RSA is an asymmetric cryptography algorithm. RSA is the first algorithm implemented with a public key cryptography system that is widely used in the world today for secure data transmission.

➢ *Applications of RSA in Cryptography*

RSA has many applications, including:

- E-Commerce: To protect data in online transactions.
- Digital Signature: To verify the authenticity of the document.
- Virtual Private Network (VPN): To transmit information securely over a network.
- Software Security: To secure software licenses and activations.

➢ *The Process of Converting Plain Text to Cipher Text:*

RSA does its encryption and decryption by generating the public and private keys. The public key is shared with everyone and used to encrypt data. The private key is available only to the key holder and is a unique key used to decrypt encrypted data, which has no match to the public key. Generally, a public key can be created from a private key, but a private key cannot be created from a public key. So, the information is safe.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

➢ *RSA Encryption Mechanism, Decryption:*

- *Mechanism and Mathematical Basis:*

It is an asymmetric key encryption algorithm, which is based on private key and public key. The RSA encryption and decryption process relies on the mathematical properties of prime numbers. It is mainly implemented using modular arithmetic and Euclidean algorithms.

- *Mathematical Basis RSA Encryption is Based on Three Main Mathematical Processes:*

✓ *Modular Arithmetic:* Numbers for modulus n are arithmetically rotated.
✓ *Prime Factorization:* If we want to calculate the value of n, we will calculate it in such a way that p and q are the product of two prime numbers, n = p*q
✓ *Euclidean Algorithm:* GCD (Greatest Common Divisor) is calculated to generate public and private keys.

- *RSA Key Generation (Key Generation):*

✓ Let us choose two large prime numbers p and q. For example, p=3 and q = 11.
✓ Calculate the modulus n. $n = p \times q = 3 \times 11 = 33$
✓ Calculate the totient function ($\phi(n)$). $\phi(n) = (p-1) \times (q-1) = (3-1) \times (11-1) = 2 \times 10 = 20$
✓ Let's choose a public exponent e, which is co-prime with $\phi(n)$. Usually, e = 7 is chosen because it is a small and effective number. The condition is $1 < e < \phi(n)$ and GCD $(e, \phi(n)) = 1$.
✓ Calculate the private key d, which satisfies the following condition: $d \times e \equiv 1 \pmod{\phi(n)}$. This gives the value of d using: $d = e^{-1} \pmod{\phi(n)}$

For example, if e = 7, d = 3.

- *RSA Encryption Process:*

✓ Convert plaintext (M) to numbers. For example, M = 31.
✓ Compute the encrypted text (C). $C = M^e \pmod{n}$ For example: $C = 31^7 \pmod{33} = 4$

- *RSA Decryption Process:*

✓ Decrypts encrypted text (C). $M = C^d \pmod{n}$ For example: $M = 4^3 \pmod{33} = 31$

- *Example (Briefly):*

✓ p = 3, q = 11, and n = 33.
✓ $\phi(n) = 20$, e = 7, d = 3.
✓ Plaintext (M) = 31.
✓ Encrypted text (C) =4.
✓ Decrypted text (M) = 31.

➢ *Features and Strengths of RSA:*

It relies on public and private keys. Extremely secure due to use of large prime numbers. Due to the modular mathematical complexity, it was safe until the advent of quantum computing.

➢ *Future Challenges:*

The RSA algorithm may be vulnerable to quantum computing attacks, where large prime numbers can be quickly factorized through Shor's algorithm. That is why post-quantum cryptography is being researched.

➢ *Benefits of RSA:*

- Asymmetric: Secure communication can be established using public and private keys.
- Widely used: E-commerce, digital signature etc. are used in many fields.
- Mathematically robust: based on factorization problems of large prime numbers, which are currently too difficult for computers.

➢ *Limitations of RSA:*

- Slow Speed: RSA is much slower than AES.
- Key Size: The key size of RSA is usually large, which increases the use of computational resources.
- Quantum Computing: When the development of quantum computers increases dramatically, the security of RSA will be threatened. [6]

D. Diffie-Hellman:

It is a key exchange cryptographic method. It was one of the earliest protocols of public key cryptography to ensure secure communication.

➢ *Application of Diffie-Hellman (DH):*

The DH protocol is currently used in many cases, such as:

- SSH (Secure Shell): Used for secure remote login and file transfer.
- VPN (Virtual Private Network): Used for secure communication on public networks.
- TLS (Transport Layer Security): Used to encrypt web traffic, such as HTTPS.
- WPA2/WPA3: Used to ensure the security of Wi-Fi networks.
- IoT (Internet of Things): For transferring encrypted data between smart devices.

➢ *How does Diffie-Hellman Work?*

- Simple Parameter Selection: Suppose Sarah and Ador choose a large prime number p and a prime root g of p. Example: p=23, g=5
- Private Key Selection: Sarah selects a random integer a. Example: a=6 and Ador selects a random integer b. Example: b=15.
- Public Key Computation and exchange: Sarah computes her public key:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

$A = g^a \pmod p = 5^6 \pmod{23} = 8$

And Sarah sends it to Ador.

Ador calculates its public key:

$B = g^b \pmod p = 5^{15} \pmod{23} = 19$

And sends it to Sarah.

- Shared Secret Key Calculation: Sarah calculates her shared secret key:

$K = B^a \pmod p = 19^6 \pmod{23} = 2$

Ador calculates its shared secret key:

$K = A^b \pmod p = 8^{15} \pmod{23} = 2$

The shared secret key for both Sara and Ador is K = 2.

➢ *Diffie-Hellman Characteristic:*

- Security: Its security rests on the discrete logarithm problem.
- Efficiency: The process is computationally simple and fast.
- Versatile use: It is used in VPN, TLS, WPA3, and many other protocols.

➢ *Diffie-Hellman Constraint:*

- Man in the Middle Attack: If a third party modifies the public key sent between Sarah and Ador, any attacker can establish a separate shared key with both.
- Quantum Computing: Quantum computing, due to its fast computation capabilities, could make it easier to solve discrete logarithm problems, which in practice can be a threat to Diffie-Hellman security.
- Uncertainty: Usually used only for key exchange; Data encryption is not done directly through it.

➢ *Diffie-Hellman: Future Developments:*

- Elliptic Curve Diffie-Hellman (ECDH): More secure and faster than standard Diffie-Hellman.
- Post-Quantum Cryptography: Efforts are underway to develop new algorithms to survive against quantum computing threats. [5]

E. Message-Digest Algorithm 5:

It converts a fixed length of plaintext into a fixed length of hashed ciphertext. It was designed by Ronald Rivest in 1991 and is an updated version of MD4.

➢ *Mathematical Basis of MD5:*

MD5 is a hash function that generates a specified 128-bit (16-byte) hash value from a specified input of any length. It is a deterministic function, meaning the same input always produces the same output. MD5 works mathematically in several steps:

- *Input Padding:*

MD5 uses padding to reduce the input length to multiple of 512-bits. The padding process is as like this:

✓ First a '1' is appended to the end of the input.
✓ The length is then filled to 48-bits by adding the required number of '0's.
✓ Finally, the original length of the input is added as 64-bits.

- *Partitioning the Input into Blocks:*

After padding the input is partitioned into 512-bit blocks. Each block is worked on separately.

- *Setting Initial Values:*

MD5 uses four 32-bit initial registers (A, B, C, D). These are initially set to mean:

✓ A = 0x67452301
✓ B = 0xEFCDAB89
✓ C = 0x98BADCFE
✓ D = 0x10325476

- *Block Processing:*

Every block is processed according to a specific sequence:

✓ *Function Selection (F, G, H, I):*

Different mathematical functions are used in each step. They are mainly based on XOR, AND, OR, NOT.

$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$

$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$

$H(B, C, D) = B \oplus C \oplus D$

$I(B, C, D) = C \oplus (B \vee \neg D)$

Every step uses inputs, function results, a fixed constant table value, and bit-rotation operations.

- *Updating Registers:*

A, B, C, D are updated with the new value at the end of each step.

- *Final Hash Value Generation:*

Finally, after processing each block, the values of A, B, C, D are combined to form a 128-bit value.

➢ *Applications of MD5 in Cryptography:*

MD5 has been used in several areas, such as:

- File integrity check: After downloading a file, its MD5 hash value is checked to see if the file has been modified.
- Password storage: Instead of storing passwords directly in the database, their MD5 hash values are stored.
- Data Signature: The MD5 hash value of a document can be used as a digital signature.
- Cache checking: Web servers use MD5 to verify the integrity of cached content.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

Converting in text to hash means MD5 is a one-way hash function. It given an input (plain text), a specific output (hash value) is obtained. MD5 works as a block cipher, where the plain text is divided into too many blocks of fixed size. Each block is processed through a series of operations and produces a hash value. These hash values are combined to form a final hash value.

➢ *A Limitation of MD5 is the Possibility of Collisions:*

Different plaintexts in code can produce the same hash value after hashing, which can then be encoded into two or more plaintexts of the same ciphertext, which is called a collision.

- *Security:*

MD5 is no longer impossible to crack, meaning finding two different inputs that produce the equal hash value will be very easy with a quantum computer. Moreover, this can be proven even with a very ordinary computer. Because of this, MD5 is not considered secure anymore [6][7].

F. Secure Hash Algorithm 512:

It is a hash function, which changes a given length of plain text into a hash value. It is an updated version of the SHA-2 and a larger version of SHA-256. It was first published in 2001 and is still widely used. Converting plain text to hash means SHA-512 is a one-way function. Given an input (plain text), a specific output (hash value) is obtained, but it is practically impossible to decode. SHA-512 is a complex mathematical function that processes input data in several steps. This process involves bit shifting, rotation, and various arithmetic operations. This process results in a 512-bit hash value.

➢ *Mathematical Basis of SHA-512:*

It is a hash function, which basically generates a 512-bit hash value from the plain text.

- *Input Padding: SHA-512 uses Padding to Divide the input Data into 1024-Bit Blocks.*

The adding process is as follows:

✓ A '1' is added to the input answer.
✓ The length is filled to 896-bits by adding the required '0'.
✓ The actual length of the input is added at the end as 128-bits.

- *Initial Hash Value:*

SHA-512 uses eight primary 64-bit registers (H0 - H7), which are set as follows:

✓ H0 = 0x6a09e667f3bcc908
✓ H1 = 0xbb67ae8584caa73b
✓ H2 = 0x3c6ef372fe94f82b
✓ H3 = 0xa54ff53a5f1d36f1
✓ H4 = 0x510e527fade682d1
✓ H5 = 0x9b05688c2b3e6c1f
✓ H6 = 0x1f83d9abfb41bd6b
✓ H7 = 0x5be0cd19137e2179

- *Processing of Message Blocks:*

SHA-512 processes 80 rounds for each message block. The following steps are followed in each round:

✓ *Message Schedule Preparation:*

The input block is converted into 80 sub-blocks from W[0] to W[79]. The first 16 blocks are taken from the input, the remaining blocks are generated by mathematical relations:

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$$

Where,

$$\sigma_0(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus (x \gg 7)$$

$$\sigma_1(x) = ROTR^19(x) \oplus ROTR^61(x) \oplus (x \gg 6)$$

✓ *Compression Function:*

The following mathematical equation is used to update the registers in each round:

$$T_1 = H + \Sigma_1(E) + CH(E, F, G) + K_t + W_t$$

$$T_2 = \Sigma_0(A) + MAJ(A, B, C)$$

$$H = G,\ G = F,\ F = E,\ E = D + T_1,\ D = C,\ C = B,\ B = A,\ A = T_1 + T_2$$

Where,

$$\Sigma_0(x) = ROTR^28(x) \oplus ROTR^34(x) \oplus ROTR^39(x)$$

$$\Sigma_1(x) = ROTR^14(x) \oplus ROTR^18(x) \oplus ROTR^41(x)$$

and,

$$CH(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$$

$$MAJ(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$

✓ *Update the Hash Value:*

At the end of each round, eight registers (A to H) are updated.

✓ *Final Output:*

After all block processes are completed, the final 512-bit hash is created by concatenating the values of H0 to H7.

➢ *Various Attacks and Techniques of SHA-512:*

- Collision Attack: Creating a collision with SHA-512 is extremely difficult, as it has 2^{256} possible outputs. Attempts to create collisions are made using modern quantum computing methods.
- Preimage Attack: Since SHA-512 is a strong hash function, this type of attack takes an extremely long time to execute.
- Man in the Man Attack (MITM): It can be reduced by using the HMAC method.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

➢ *Quantum Computing and the Security of SHA-512:*

- Quantum risk: The use of quantum computing could weaken the security of SHA-512. Shore's algorithm and Grover's algorithm can do partial attacks on SHA-512.
- Resistance to SHA-512: SHA-512 provides n-bit security. With quantum computers this comes down to n/2. For example, 512/2=256-bit security in SHA-512 is still very strong.
- Future Security: New algorithms are being developed under Post-Quantum Cryptography, which may replace SHA-512.

➢ *Advantages of SHA-512 Security:*

It is a highly secure hash algorithm. It is very difficult to crack.

- Efficiency: SHA-512 is relatively fast and efficient.
- Widely used: SHA-512 is widely used in various cryptographic applications.

➢ *Limitations of SHA-512:*

- Complexity: The algorithm of SHA-512 is very complex, difficult to understand and implement.

➢ *The Future of SHA-512:*

It is still considered a secure hash algorithm. However, with the development of quantum computing, the security of SHA-512 may be questioned. SHA-512 is a strong and secure hash function used in various cryptographic applications.

➢ *Applications of SHA-512 in Cryptography:*

SHA-512 has been used in various fields, such as:

- File integrity check: After downloading a file, its SHA-512 hash value is checked to see if the file has been modified.
- Digital signature: The SHA-512 hash value of a document can be used as a digital signature.
- Password storage: Instead of storing passwords directly in the database, their SHA-512 hash value is stored.
- Blockchain: Blockchain technology uses SHA-512 to connect blocks and add new blocks.

Other cryptographic applications: SHA-512 is also used in other cryptographic applications, such as MAC and HMAC[7][8].

III. FEATURES COMPARISON OF DIFFERENT ALGORITHMS

Cryptography is an integral part of the modern digital age, ensuring data privacy, integrity, and authentication. This study analyzes the characteristic differences between many cryptographic algorithms, such as DES, AES, RSA, Diffie-Hellman, MD5, and SHA-512, and their practical applications. Each algorithm has its own framework and limitations, which are used to meet different security needs.

Table 1 Comparison of DES, AES, RSA, Diffie-Hellman, MD5, and SHA-512

Feature	DES	AES	RSA	Diffie-Hellman	MD5	SHA-512
Block Size	64-bit	128-bit	Not applicable (asymmetric encryption)	Not applicable (Key exchange)	128-bit	512-bit
Key Size	56-bit	128,192,256-bit	1024,2048, 4096-bit	1024,2048, 4096-bit	Not defined	Not defined
Structure	Feistel Network	Substitution-Permutation Network	Asymmetric Cryptography	Asymmetric Cryptography	MD-family Algorithm	Merkle-Damgard Construction
Usage	Data Encryption	Data Encryption	Data Encryption, Key Exchange	Key Exchange	Hashing	Hashing
Flexibility	Low	Medium	Medium	Medium	Low	High
Known Attacks	Brute Force, Meet-in-the-middle	Timing Attacks	Factorization Attacks	MITM (Man-in-the-Middle)	Collision Attacks	Potential Quantum Attacks
Security	Weak	Strong	Strong	Strong	Weak	Very Strong
Current use	Outdated for security, rarely used (e.g., Triple DES_)	Widely used for web and data security	Digital signatures, Key Management	Key Exchange	Integrity checks in non-secure contexts	Secure applications requiring high protection
Efficiency	Low	High	Low	High	High	Medium
Complexity	Simple	Moderate	Complex	Complex	Simple	Complex

First, in a direct comparison between DES and AES, it is clear that AES is currently the most secure and effective method for data encryption. DES is vulnerable to collisions and brute force attacks due to its 56-bit key length, whereas AES supports 128, 192, and also 256-bit keys. AES is

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

considered an improved version of DES due to its faster processing capabilities and security.

Second, comparing RSA and the Diffie-Hellman algorithm shows that RSA is a powerful algorithm widely used in asymmetric cryptography. It is mainly used for data encryption and digital signature. Although Diffie-Hellman is primarily used for key exchange, it is vulnerable to MITM attacks. Although the mathematical structure of RSA is more complex, its security is more effective than that of Diffie-Hellman.

Third, a comparison between MD5 and SHA-512 shows that the MD5 hash function is inadequate for modern security needs due to the risk of collisions. SHA-512, in contrast, is a very powerful hash function that provides a 512-bit output. It is capable of meeting high-security requirements, such as blockchain, digital signatures and other cryptographic applications.

An important aspect of this research is to identify the practical relevance of various algorithms and their limitations. Older algorithms like DES and MD5 are not sufficient for modern needs, but they have historical importance. On the other hand, AES, RSA, and SHA-512 are successful in meeting current security needs. While Diffie-Hellman is useful for fast and simple key exchange, it is only usable in certain situations.

In addition, the potential impact of quantum computing will play an important role in determining the future of cryptography technology. Quantum computing can have a negative impact on asymmetric algorithms such as RSA and Diffie-Hellman, as it is efficient at factoring large prime numbers. However, AES and SHA-512 are still considered secure. In this context Post-Quantum Cryptography research and implementation is becoming a necessity.

IV. THE FUTURE OF CRYPTOGRAPHY IN QUANTUM COMPUTING

As technology advances, quantum computing will play a formidable role in the future of cryptography. With this new technology, it will probably be easier to think about the arrangements for them. Advances in quantum computing technology can enable cryptographic code breaking as well as hack the security of public key cryptographic systems.

➢ *Some of the Key Principles used in Quantum Computing are:*

- Key exchange: Classical cryptography techniques can be easily broken by modern algorithms. With the advent of quantum computing, the security of key exchange mechanisms will be compromised and new security measures will need to be taken.
- Key distribution: Second-category protocols, such as BB84, may not be secure with quantum computing. The protocols that quantum computing uses can be hacked and new attacks will need to be created.
- Simulation and import: Quantum computing techniques may be rich for cryptographic exploration, but they may be limited to simulation and import steps.

To address these challenges, we may need to improve our cryptographic steps, especially with quantum computing. Therefore, our security technologies and processes need to continue to evolve and change in a safe manner to keep pace with advances in quantum computing.

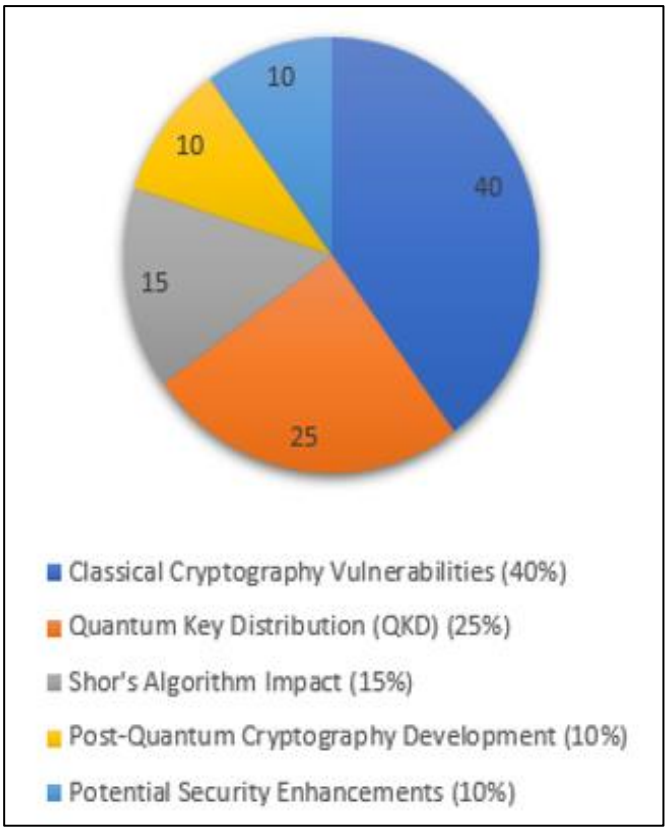

Fig 3 Impact of Quantum Computing on Cryptography

The pie chart above shows Quantum Computing and Potential Implications of Cryptography, broken down into different categories.

➢ *Category Description:*

- Classical Cryptography Vulnerabilities (40%): Current cryptographic systems such as RSA or ECC could be infirm to quantum computers.
- Quantum Key Distribution (QKD) (25%): Quantum technology is opening up the possibility of secure key exchange.
- Shor's Algorithm Impact (15%): Quantum algorithms could break classical encryption systems.
- Post-Quantum Cryptography Development (10%): Quantum-resistant algorithms are being developed.
- Potential Security Enhancements (10%): New technologies show potential for security improvements.

This data shows us how quantum computing could revolutionize cryptographic systems and cybersecurity. [11]

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct096

V. CONCLUSION

In the comparative analysis of the algorithms, we find that each algorithm has its own characteristics, structure and field of application. DES, AES, RSA, Diffie-Hellman, MD5, and SHA-512—each of these algorithms has a different design and functionality. They fulfill different needs according to their strengths, weaknesses and application areas. DES was one of the early block cipher algorithms. It was once popular due to its simple design and limited key size. However, over time cryptanalysis and powerful computing power rendered DES unusable. AES, as the successor to DES, offers more security and flexibility. Due to its large key size and block size, it is currently the most reliable symmetric cryptography algorithm. It is used in various applications worldwide, such as Wi-Fi encryption, VPN, and secure transmission. RSA and Diffie-Hellman are two aspects of asymmetric key cryptography. However, RSA's mathematical structure and strong security still make it popular. The main weakness of Diffie-Hellman is the risk of Man-in-the-Middle Attack. MD5 was once a popular hash function. But it is no longer used for security purposes due to the risk of collision attacks. SHA-512, which is part of the SHA-2 group, is one of the most powerful hash algorithms available today. Its large output size and complex mathematical structure make it effective against collision attacks. In comparison, AES and SHA-512 are currently recognized as the most reliable and secure algorithms. RSA and Diffie-Hellman are still an essential part of cryptography today, especially in key exchange and data security. On the other hand, older algorithms like DES and MD5 have become obsolete as technology advances.

Each algorithm has specific strengths and limitations. It is important to choose them by understanding the appropriate application and context. Growing Cyber Threats and Quantum Computing.

REFERENCES

[1]. William, S. (1999). Cryptography and network security: principles and practice, Prentice-Hall, Inc
[2]. https://academickids.com/encyclopedia/index.php/Data_Encryption_Standard#google_vignette
[3]. https://www.geeksforgeeks.org/advanced-encryption-standard-aes/
[4]. The RSA Algorithm by Evgeny Milanov. Publish 3 June 2009
[5]. Diffie-Hellman:Key Exchange and Public Key Cryptosystems by Sivanagaswathi Kallam , Master of science Math and Computer Science Department Indiana State University TerreHaute,IN,USA (9/30/2015)
[6]. R. Rivest, MIT Laboratory for Computer Science and RSA Data Security, Inc. (April 1992)
[7]. A comparative study of Message Digest 5(MD5) and SHA256 algorithm D Rachmawati1* , J T Tarigan1* and A B C Ginting 1* 1Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Medan 20155, Indonesia
[8]. SHA-512/256 by 1. Shay Gueron (Department of Mathematics, University of Haifa, Israel), 2. Simon Johnson (Mobility Group, Intel Corporation, Israel Development Center, Haifa, Israel), 3 . Jesse Walker (Intel Architecture Group, Intel Corporation, USA 4 Security Research Lab, Intel Labs, Intel Corporation, USA)
[9]. Figure 1 - uploaded by Marcelo Lubaszewski, https://www.researchgate.net/figure/Block-diagram-of-DES-algorithm_fig1_220850878
[10]. Fig 6 - uploaded by Muhammad Faheem Mushtaq , https://www.researchgate.net/figure/Advanced-Encryption-Standard-AES-Algorithm_fig5_321587376
[11]. Security in quantum cryptography; Christopher Portmann∗ Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland. Renato Renner† Institute for Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland