**Comparative Analysis of Traditional and Light weighted Algorithms**

Rimpi Rani

Assistant Professir(Computer Science)

Punjabi University TPD Malwa College Rampura Phul

Ph. DResearch Scholar Desh Bhagat University Mandi Gobindgarh (Pb)

Email rimpi.rani@pbi.ac.in

Prof (Dr.) R.K. Bathla

Senior Professor

Desh Bhagat University Mandi Gobindgarh (Pb)

**Abstract** - Cryptography algorithms are an essential tool for protecting sensitive data and communications. Cryptography algorithms can be compared based on several criteria. In this paper some traditional AES, RSA, SHA and light weighted SPECK, SIMON, CELFI: cryptography algorithms are compared based on key size, block size, rounds and security. This paper presents a comparative analysis of the popular traditional and new light weighted algorithms. Namely AES, RSA, SHA, SPECK, SIMON, CELFI. It's worth noting that the choice of algorithm depends on the specific use case and the level of security required. In general, traditional algorithms may be more appropriate for applications that require high levels of security, while lightweight algorithms may be more suitable for resource-constrained environments.

**Keywords** – Cryptography, Traditional, Light weighted, Algorithm.

**1) Introduction** - Cryptography algorithms are widely used in various applications to secure data. However, choosing the right encryption algorithm is crucial to ensure the security and performance of the system. Some traditional cryptography algorithms include AES, RSA, and SHA, while some new lightweight cryptography algorithms include SPECK, SIMON, and CLEFIA. The analysis is conducted based on various parameters, including security, performance, key size, and adoption. Cryptography algorithms have been around for a long time and have been widely used for securing data. However, with the emergence of new technologies such as IoT, cloud computing, and mobile devices, there is a growing need for more lightweight cryptography algorithms that are designed to run efficiently on resource-constrained devices. Here are some key differences between traditional cryptography algorithms and new lightweight algorithms:

Key size: Traditional cryptography algorithms typically use larger key sizes to provide stronger security. However, this also makes them more resource-intensive and may not be suitable for low-power devices. New lightweight cryptography algorithms, on the other hand, typically use smaller key sizes to reduce computational overhead.

Block size: Traditional cryptography algorithms often have larger block sizes, which can be inefficient for resource-constrained devices. New lightweight cryptography algorithms typically use smaller block sizes, which allows them to operate more efficiently on low-power devices.

Round function: Traditional cryptography algorithms typically use complex round functions that involve multiple operations, such as substitution and permutation. New lightweight cryptography algorithms often use simpler round functions to reduce computational overhead.

Security: Traditional cryptography algorithms have undergone extensive cryptanalysis and have been proven to be secure against a wide range of attacks. New lightweight cryptography algorithms are still relatively new and have not undergone as much cryptanalysis. This means that their security may not be as well-established.

**2) RelatedWork**

**2.1) Advanced Encryption Standard (AES) -** AES is a symmetric encryption algorithm that is widely used in both software and hardware applications. It is considered to be highly secure, efficient and flexible. It uses a symmetric key algorithm that is resistant to attacks, such as brute force attacks, and is widely used in both software and hardware applications. AES is highly recommended for any application that requires strong encryption and is used by governments, financial institutions, and other organizations to protect sensitive data. AES is very efficient and can encrypt and decrypt data quickly. It is also flexible and can be used with different key sizes depending on the level of security required. AES is widely considered to be highly secure and efficient, with low memory usage. AES encryption and decryption operations are very fast, especially when hardware acceleration is available. For example, on an Intel Core i5-6600K CPU, AES-128 encryption can achieve a speed of over 1 GB/s using the AES-NI instruction set.

Strengths:

AES is very fast and efficient, making it suitable for use in high-speed applications.

It has no known vulnerabilities and has withstood many years of analysis by cryptographers.

It is widely used and has strong community support, which makes it easier to integrate into existing systems.

Weaknesses:

Although AES is very secure, there is always the possibility of attacks such as side-channel attacks, which exploit weaknesses in the physical implementation of the algorithm.

AES is a symmetric encryption algorithm, which means that the same key is used for both encryption and decryption. This can be a weakness if the key is compromised.

AES is a symmetric-key block cipher that encrypts data in 128-bit blocks with a key length of 128, 192, or 256 bits.

The algorithm for AES involves several steps:

First, the plaintext is divided into 128-bit blocks and XORed with a 128-bit initialization vector (IV).

Second, the XORed data is encrypted using a series of rounds that involve substitution and permutation operations using a set of precomputed round keys derived from the key.

Third, the resulting cipher text is decrypted in the reverse order of the encryption process to generate the original plaintext.Top of Form

**2.2) Rivest-Shamir-Adleman (RSA)-** RSA is a public-key encryption algorithm that is commonly used for secure communication and digital signatures. It is based on the mathematical properties of prime numbers and is considered to be highly secure. RSA is based on the difficulty of factoring large prime numbers and is widely considered to be highly secure. However, RSA is relatively slow and less efficient than symmetric encryption algorithms like AES. RSA is a public-key encryption algorithm that is widely used for secure communication and digital signatures. RSA encryption and decryption operations are relatively slow compared to symmetric key algorithms like AES, and the memory usage is higher due to the larger key size required for equivalent security. For example, on the same CPU, RSA-2048 encryption can achieve a speed of around 13.6 MB/s.

Strengths:

RSA is very secure and widely used, making it a good choice for many applications.

It has been extensively studied and has withstood many years of analysis by cryptographers.

It is widely supported and easy to integrate into existing systems.

Weaknesses:

RSA is relatively slow and resource-intensive, making it less suitable for high-speed applications.

RSA relies on the difficulty of factoring large prime numbers, which could potentially be broken by advances in mathematics or computing power.

It is vulnerable to certain types of attacks, such as timing attacks and side-channel attacks.

RSA (Rivest-Shamir-Adleman):

RSA is an asymmetric-key cryptosystem that uses a public key for encryption and a private key for decryption.

The algorithm for RSA involves several steps:

First, the user generates a pair of public and private keys by selecting two large prime numbers and computing the modulus and a set of parameters.

Second, the public key is shared with others, who can use it to encrypt messages and send them securely to the user.

Third, the user decrypts the messages using the private key, which can only be accessed by the user.

Fourth, the user can also sign messages using the private key, which can be verified by others using the public key.

**2.3 Secure Hash Algorithm (SHA)-** It is very fast and efficient, making it suitable for use in high-speed applications.

It has no known vulnerabilities and has withstood many years of analysis by cryptographers.

It is widely used and has strong community support, which makes it easier to integrate into existing systems.

Weaknesses:

SHA is a one-way function, which means that it is not possible to recover the original data from the hash value. This can be a weakness if the original data needs to be recovered.

SHA relies on the difficulty of finding collisions, which could potentially be broken by advances in mathematics or computing power.:

SHA is a family of cryptographic hash functions that generate a fixed-size digest of the input data.

The algorithm for SHA involves several steps:

First, the input data is padded and divided into fixed-size blocks.

Second, the message schedule is generated by applying a series of logical and arithmetic operations to the input data.

Third, the hash function computes a series of hash values by iterating over the message schedule and updating a set of intermediate variables.

Fourth, the final hash value is computed by concatenating and formatting the intermediate variables according to the specific SHA variant used.

**2.4 SPECK**- The Speck algorithm is a block cipher encryption algorithm designed by the United States National Security Agency (NSA). It is part of the NSA's portfolio of cryptographic algorithms known as the SIMON and SPECK family of ciphers, which are designed to provide efficient, lightweight encryption for use in constrained environments such as embedded systems and low-power devices.

Speck is a symmetric key cipher, which means that it uses the same secret key for both encryption and decryption. It operates on blocks of data, typically 64 or 128 bits in length, and applies a series of substitution and permutation operations to transform the input plaintext into cipher text.

The Speck algorithm has two main parameters: the block size and the key size. The block size determines the size of the blocks of data that the algorithm operates on, while the key size determines the length of the secret key used to encrypt and decrypt the data. The algorithm is known for its simplicity and efficiency, as well as its resistance to cryptanalysis attac

Speck is a family of lightweight block ciphers designed for use in constrained environments.

The algorithm for Speck involves several steps:

First, the plaintext is divided into two halves and XORed with a round key.

Second, the halves are substituted and permuted using a set of fixed rotation constants and a key-dependent permutation function.

Third, the resulting halves are XORed and the process is repeated for a number of rounds.

Fourth, the resulting cipher text is generated by concatenating the final halves.

**2.5 Simon -** The SIMON algorithm is a family of block ciphers designed by the United States National Security Agency (NSA) as part of their portfolio of lightweight encryption algorithms. SIMON is similar to another NSA-designed algorithm, Speck, in that it is designed for use in resource-constrained environments such as embedded systems and low-power devices.

There are several variants of the SIMON algorithm, each with a different block size and key size. The most common variants are SIMON-32/64, SIMON-48/72, SIMON-48/96, SIMON-64/96, SIMON-64/128, SIMON-96/96, SIMON-96/144, SIMON-128/128, and SIMON-128/192.

Like Speck, SIMON is a symmetric key cipher, which means that it uses the same secret key for both encryption and decryption. It operates on blocks of data, typically 32, 48, 64, 96, or 128 bits in length, and applies a series of substitution and permutation operations to transform the input plaintext into cipher text. SIMON is known for its simplicity and efficiency, as well as its resistance to cryptanalysis attacks. It has been adopted by a number of organizations and is used in a variety of applications, including military communications, secure messaging, and secure storage.

As with any cryptographic algorithm, the security of SIMON depends on the strength of the key used and the implementation of the algorithm. The NSA has published specifications and reference implementations of SIMON, and it has undergone extensive analysis by the cryptographic community. It is a family of block ciphers designed for use in constrained environments.

The algorithm for Simon involves several steps:

First, the plaintext is divided into two halves and XORed with a round key.

Second, the halves are substituted and permuted using a set of fixed constants and a key-dependent permutation function.

Third, the resulting halves are XORed and the process is repeated for a number of rounds.

Fourth, the resulting cipher text is generated by concatenating the final halves.

**2.6 Clerfia** -Top of Form

Clefia is a symmetric key block cipher algorithm developed by Sony Corporation. It is designed to provide efficient and secure encryption for use in various applications, including storage devices, communication systems, and financial transactions.

Clefia operates on 128-bit blocks of data and supports three different key lengths: 128, 192, and 256 bits. It uses a substitution-permutation network (SPN) structure, similar to other block ciphers like AES, to transform the input plaintext into cipher text. However, Clefia also includes some unique features, such as a key whitening step and a key-dependent S-box generation method.

Clefia has undergone extensive analysis by the cryptographic community and is known for its security and efficiency. It has been selected as a standard algorithm in the ISO/IEC 29192-2:2012 international standard for lightweight cryptography.

While Clefia is not as widely used as some other block ciphers like AES, it has found applications in areas such as embedded systems and low-power devices, where its efficiency and security properties are particularly beneficial.

Clefia is a block cipher based on the Advanced Encryption Standard (AES) and designed to resist certain attacks on AES.

The algorithm for Clefia involves several steps:

First, the plaintext is divided into two halves and XORed with a round key.

Second, the halves are substituted and permuted using a set of fixed constants and a key-dependent permutation function.

Third, the resulting halves are XORed and the process is repeated for a number of rounds.

**4) Fourth, the resulting ciphertext is generated by concatenating the final halvesTop of Form**

**Findings and Results -** It's found (as shown in Table 1) that while these algorithms are widely used for secure communication and data protection, they are subject to ongoing research and potential vulnerabilities. It's important to use them correctly and keep them updated to minimize the risk of attacks .In summary, different cryptography algorithms have different strengths and weaknesses. AES is widely considered to be highly secure and efficient, while RSA and ECC are more commonly used for public-key encryption. The choice of algorithm depends on the specific requirements of the application, such as security level, speed, and efficiency. It is important to keep in mind that the strength of any cryptographic algorithm depends on its implementation and the key management practices used. Different cryptography algorithms have different strengths and weaknesses. AES is widely considered to be highly secure, efficient, and flexible, while RSA and ECC are more commonly used for public-key encryption. The choice of algorithm depends on the specific requirements of the application, such as security level, speed, and efficiency.

Bottom of Form

However, it's important to note that the performance of a cryptography algorithm depends on many factors, such as the specific implementation, key size, and hardware acceleration. It's crucial to select the right algorithm based on the specific requirements of the application . Cryptography algorithms are widely used to secure data in various applications including cloud computing, e-commerce, mobile devices, and many more. It's important that the choice of cryptography algorithm depends on the specific application and the level of security required.

Speck, Simon, and Clefia are all lightweight encryption algorithms designed for use in constrained environments such as embedded systems and low-power devices. They are all symmetric key block ciphers that operate on blocks of data and use a substitution-permutation network (SPN) structure to transform plaintext into cipher text.

However, they have some differences. Speck has a simpler design than Simon and uses fewer rounds, making it faster and more efficient in some cases. Simon, on the other hand, has a larger number of variants with different block and key sizes, providing greater flexibility for different applications. Clefia is unique in its use of key whitening and key-dependent S-box generation, which can improve its security.

Overall, each algorithm has its strengths and weaknesses, and the choice of which algorithm to use will depend on the specific requirements of the application, such as the level of security needed, the available resources, and the desired performance. It is important to carefully evaluate and test any cryptographic algorithm before using it in a real-world system to ensure its security and effectiveness. Comaprasion of all algorithms is shown in table 1.

Top of Form

|  Cryptography Algorithm | Key Size | Block Size | Round Function | Security |
| --- | --- | --- | --- | --- |
| AES | 128-256 bits | 128 bits | Complex (substitution, permutation) | Well-established |
| RSA | 2048-4096 bits | N/A | Complex (modular arithmetic) | Well-established |
| SHA | 256-512 bits | 512 bits | Complex (message expansion, compression) | Well-established |
| SPECK | 64-128 bits | 64-128 bits | Simple (addition, XOR, rotation) | Limited analysis |
| SIMON | 64-128 bits | 64-128 bits | Simple (addition, XOR, rotation) | Limited analysis |
| CLEFIA | 128-256 bits | 128 bits | Simple (addition, XOR, rotation) | Limited analysis |

Table 1. Comparasion of algorithms

**4. Conclusion** -As you can see, traditional cryptography algorithms tend to use larger key sizes and block sizes, as well as more complex round functions. This provides stronger security, but also requires more computational resources. New lightweight algorithms, on the other hand, use smaller key sizes and block sizes, as well as simpler round functions, to reduce computational overhead. However, because they are relatively new, their security has not been as extensively analyzed as traditional algorithms.

It's worth noting that the choice of algorithm depends on the specific use case and the level of security required. In general, traditional algorithms may be more appropriate for applications that require high levels of security, while lightweight algorithms may be more suitable for resource-constrained environments.

**References**

1] Taylor, Onate & Emmah, Victor. (2018). Comparative Analysis of Cryptographic Algorithms in Securing Data.

2] Alrubaiei, Mazoon & AlYarubi, Thuraiya & Kumar, Basant. (2020). Critical Analysis of Cryptographic Algorithms. 1-7. 10.1109/ISDFS49300.2020.9116213.

3] Kumar, Vinod. (2014). A Comparative Study of Cryptographic Algorithms A Comparative Study of Cryptographic Algorithms A Comparative Study of Cryptographic Algorithms A Comparative Study of Cryptographic Algorithms. International Journal of Computer Science and Network. 3. 2277-5420.

4] Alabi, Orobosade & Thompson, Aderonke & Alese, Boniface & Gabriel, Arome. (2020). Cloud Application Security using Hybrid Encryption. Communications on Applied Electronics. 7. 25-31. 10.5120/cae2020652866.

5] Alrubaiei, Mazoon & AlYarubi, Thuraiya & Kumar, Basant. (2020). Critical Analysis of Cryptographic Algorithms. 1-7. 10.1109/ISDFS49300.2020.9116213.

Top of Form

Top of Form