Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/25/20240893

A comparative analysis of AES and RSA algorithms and their integrated application

Zehao Tuo

Beijing Foreign Language Academy, Beijing, 100089, China

201010130203@stu.swmu.edu.cn

Abstract. Cryptography holds a significant place in the realm of information technology, safeguarding sensitive information from unauthorized access. This discourse centers on two prevalent cryptographic algorithms: AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman). While AES falls under the category of symmetric cryptography, RSA is a part of asymmetric cryptography. The paper's objective lies in furnishing readers with an insightful exploration of both algorithms, delving into their respective strengths and weaknesses. A comprehensive examination reveals the intricate details of their operation, highlighting the security aspects and potential vulnerabilities. Understanding the fundamental distinctions and the operational mechanics of AES and RSA contributes to an enhanced perception of their application in diverse cryptographic contexts. Beyond this, the paper delves into the synergistic utilization of AES and RSA, presenting an innovative approach to reinforcing information security. This integration leverages the unique advantages of both algorithms, mitigating their individual limitations and bolstering the security framework. In essence, a balanced perspective on AES and RSA, along with their collaborative application, furnishes a robust foundation for informed cryptographic choices, fortifying the safeguarding of sensitive data against emerging security threats.

Keyword: AES, RSA, Hybrid Algorithm.

1. Introduction

In the realm of information security, the CIA triad (Confidentiality, Integrity, and Availability) represents three pivotal facets to heed [1]. Standing as a foundational element of information security, cryptography predominantly centers on data integrity and confidentiality. It utilizes mathematical algorithms to morph data into an unintelligible and scrambled form during storage and transmission, barring unauthorized access to the safeguarded data. Pinpointing the exact inception of cryptography proves challenging, with its comprehensive development emerging prominently during the Renaissance. Initially, cryptography was a basic practice of substituting letters with others in a distinctive sequence. The significant leap in cryptography occurred in 1977 with the U.S. government's release of the Data Encryption Standard (DES) [2], signaling a transformative progression in the field. The year 2001 saw the unveiling of the Advanced Encryption Standard by the American National Institute of Standards and Technology. Despite both DES and AES falling under the umbrella of symmetric cryptography, this article delves deeply into AES. Conversely, the RSA algorithm, emerging in 1978, is an exemplar of asymmetric cryptography. Named after its innovators

DOI: 10.54254/2753-8818/25/20240893

(Rivest, Shamir, and Adleman), RSA distinguishes itself by employing separate keys for encryption and decryption, with only the encryption key available publicly [3]. This contrasts with the symmetric cryptography approach where a singular secret key is utilized for both encryption and decryption processes.

2. Relevant theories

2.1. Euler's Totient Function

The Euler's totient function, also called totient function, is denoted as $\varphi(n)$. It counts the number of positive integers less than or equal to n that are relatively prime to n, or the number of totatives of n.

The equation of Euler's totient function is defined as follows:

$$\varphi(n) = \prod_{(p|n)} \left(1 - \frac{1}{p}\right) \tag{1}$$

For example, $\varphi(80)$ can be obtained through the following equation:

$$\varphi(80) = 80 * \left(1 - \frac{1}{2}\right)^4 * \left(1 - \frac{1}{5}\right) \tag{2}$$

2.2. Euler's theorem and Fermat little theorem

When positive integers a is relatively prime to n, there is the following formula:

$$a^{\varphi(n)} \equiv 1(mod\ n) \tag{3}$$

Fermat little theorem is a special condition derived from Euler's theorem (when n is a prime number):

$$a^{p-1} \equiv 1(mod\ p) \tag{4}$$

In this formula, p-1 = $\varphi(p)$. This is because p is a prime number, the only prime factor of n is n itself. Then, $\varphi(p)$ is given by this formula: $\varphi(p)$=p $*$ $(1 - \frac{1}{p})$=p-1

2.3. Chinese Remainder theorem

Chinese Remainder theorem is defined as:

When m_1, m_2, ..., m_n are pairwise coprime positive integers, and a_1, a_2, ..., a_n are integers, Then the following equation must have solution:

$$\begin{cases} x \equiv a_1(mod\ m_1) \\ x \equiv a_2(mod\ m_2) \\ \quad \dots \dots \\ x \equiv a_n(mod\ m_n) \end{cases} \tag{5}$$

The solution x can be obtained by the equation (N is the product of all m_i:

$$x \equiv \sum_{i=1}^{n} a_i \times \frac{N}{m_i} \times \left[\left(\frac{N}{mi}\right)^{-1}\right] \tag{6}$$

3. RSA and AES cryptography

3.1. Introduction to AES

The Advanced Encryption Standard, initially termed the Rijndael algorithm [4], is the brainchild of Belgian cryptographers Joan Daemen and Vincent Rijmen. Presented alongside twenty other algorithms to the National Institute of Standards and Technology (NIST), the Rijndael algorithm was

DOI: 10.54254/2753-8818/25/20240893

ultimately published as the AES algorithm by NIST on November 26, 2001. Serving as a robust alternative to the Data Encryption Standard (DES), AES operates as a block cipher encryption algorithm, utilizing a symmetric key for both the encryption and decryption of data [5]. AES processes a plain-text of 128 bits, while accommodating encryption and decryption keys of 128 bits, 192 bits, and 256 bits. An additional two rounds are necessitated for every incremental 64 bits in the key. Within this framework, both text and key undergo a conversion into a 4*4 matrix, with each byte represented as a hexadecimal number. This discussion centers primarily on the utilization of a 128 bits key in AES. Figure 1 outlines the fundamental encryption process for a 128 bits key within AES. For decryption, this process is executed in reverse. The procedure unfolds across three primary sections: Key Expansion. Nine Rounds Circulation:

a. Byte Substitution.
b. Row Shifting.
c. Column Mixing.
d. Round Key Addition.

Final Round:

a. Byte Substitution.
b. Row Shifting.
d. Round Key Addition.

Each section plays a crucial role, ensuring the integrity and security of the encryption and decryption processes within the AES algorithm..

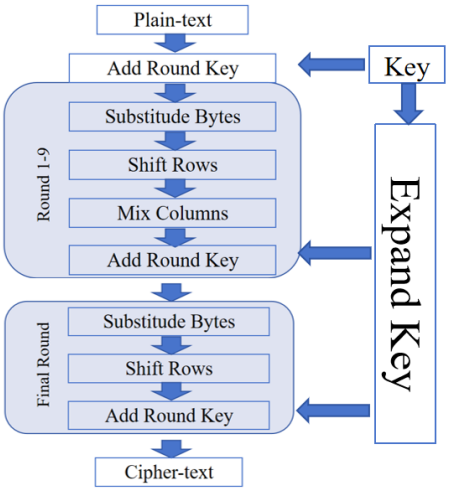

Figure 1. Process of encryption in AES (Photo/Picture credit: Original).

3.1.1. Specific process of AES. Key expansion in nine rounds and the final round, "Add Round Key" requires the original key to be expanded to generate ten round keys. As shown in Table 1.

Table 1. Original key and blank round key.

C1	C5	C9	C13					
C2	C6	C10	C14					
C3	C7	C11	C15					
C4	C8	C12	C16					
W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	.....

As shown in Table1., the 128 bits key is segmented into 16 sections, each section is denoted by hexadecimal number. Each column is denoted as W_i, and the columns of ten round keys are generated

DOI: 10.54254/2753-8818/25/20240893

by the initial four column. When i is not a multiple of four, $W_i=W_{i-4} \oplus T(W_{i-1})$($\oplus$ is a type of logical operation, called xor, such that if the values of a and b are not the same, the result is 1; If the values of a and b are the same, the result is 0). When i is a multiple of four, $W_i=W_{i-4} \oplus T(W_{i-1})$. Function T consist of three parts:

Rotation: Shifting each section by one byte, showing in Figure 2

Substitution: Using the S-BOX to substitude the result of Rotation.

Xor-ing: Performing a xor between the result from substitution and round constant [6].

Figure 2. Rotation process (Photo/Picture credit: Original).

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 3. AES S-BOX (Photo/Picture credit: Original).

Process within nine rounds circulation and final round.

Substitude Bytes.

In this process, each byte in the matrix is substituded using the s-box.

Shift Rows:

Each row of the matrix is shifted to the left. As showing in Figure 3, the first row shifted by zero byte; for each row that follows, one more byte is shifted.

Mix Column:

The resulting matrix of “shift rows” is mixed with a given matrix using a specified formula.

Add Round Key:

Performing a xor operation between the result from “mix column” and the round key. The first “Add Round key” that was done to plain-text in Figure 4 uses the original key to do xor operation.

DOI: 10.54254/2753-8818/25/20240893

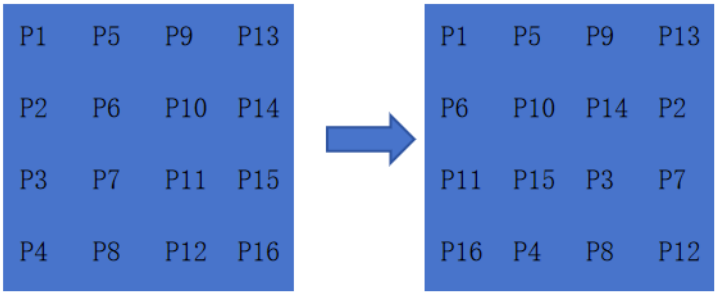

Figure 4. Process of shift rows (Photo/Picture credit: Original).

3.2. Introduction to RSA cryptography:

Using the Euler's theorem and building on Diffie and Hellman's idea, Rivest, Shamir, and Adleman designed RSA by 1978. It uses a public-key cryptosystem and digital signatures [7]. The term public-key cryptosystem refers to a cryptosystem with a public encryption key and a secret decryption key. Every sender can use the encryption key to transform the plain-text into the cipher-text and send it to the receiver. The mechanism of public key cryptosystem is denoted by the following formula:

$$D(E(M)) = M \tag{7}$$

It is extremely hard to compute the decryption key D from the encryption key E without additional information, so a public E does not undermine the confidentiality of D. It is a "trap-door one-way function", meaning that D can only be computed in one direction [8].

3.2.1. Mathematical mechanism of RSA. When Bob want to compute the private key and public key used in RSA. He first need to generate two large prime numbers p and q, let n=p*q, and find the φ(n). Because p and q are both prime numbers,φ(n) = (p - l) (q - 1). Then he find an integer b such that e is between 3 and n and relatively prime to φ(n) and a multiplicative inverse modulo d of e such that ed $\equiv$ $1(mod\ \varphi n)$. n and e are public, and d is kept secret.

When Alice wants to encrypt a plain text message, the plain text is converted into a integer, call it x. Then, x is segment such that each x_i is less than n. Alice can encipher x by the following formula:

$y_i \equiv {x_i}^e$(mod n). Alice then sends the cipher text y to Bob. To decipher the cipher text y, Bob can use the following formula:

$$x_i \equiv {y_i}^d (mod\ n) \tag{8}$$

The decryption and encryption process can be done in such way is largely because the process of generating the public and secret key. From the encryption and decryption process, ${y_i}^d \equiv {x_i}^{ed}$(mod n) can be obtained. Since ed $\equiv$ $1(mod\ \varphi(n))$, ${y_i}^d \equiv {x_i}^{ed}$(mod n) can be written as ${y_i}^d \equiv {x_i}^{k\,\varphi(n)+1}$(mod n). Since n is the product of two large number, x_i is always relatively prime to n except x is equal to p or q. According to Euler's theorem, ${x_i}^{k\,\varphi(n)+1} \equiv x_i$(mod n). A entire formula is obtained:

$${y_i}^d \equiv {x_i}^{ed} \equiv {x_i}^{k\,\varphi(n)+1} \equiv x_i (mod\ n) \tag{9}$$

3.2.2. Digital signature. RSA implements digital signature to ensure that senders can verify the message from the third party. If Bob, the person who has the secret key, wants to send a message to Alice, he can use the secret key d to sign the message m: s=$sig_{sk}(m) = m^d$(mod n)

Then, Bob sand the message m with the digital signature s to Alice. Alice can verify the signature using her public key e to get m': m'=s^e(mod n). then compare m' with m. If they are the same, then the message is from Bob since only Bob has the secret key.

DOI: 10.54254/2753-8818/25/20240893

3.2.3. Efficient method for decryption process. When using secret key d to decrypt the cipher text y_i, Chinese remainder theorem can be implemented to accelerate the process since owner of the secret key knows p and q. $x_i \equiv {y_i}^d (\text{mod n})$ can be rewrite as:

$$\begin{cases} x_i \equiv {y_i}^d (mod\ q) \\ x_i \equiv {y_i}^d (mod\ q) \end{cases} \tag{10}$$

For example, when Bob gets a cipher-text 178. The secret key d is 63, and p and q are 23 and 17 (In real situation, p and q are much larger, and that is why Chinese Remainder Theorem is used). The plain-text x_i satisfies the following equation: $x_i \equiv 178^{63} (\text{mod } 391)$

Applying Chinese Remainder Theorem, the equation can be rewrite as:

$$\begin{cases} x_i \equiv 178^{63} (\text{mod } 23) \\ x_i \equiv 178^{63} (\text{mod } 17) \end{cases} \tag{11}$$

After the simplification using basic property of congruence and Euler's Theorem, equation can be rewrite as:

$$\begin{cases} x_i \equiv 17^{19} (\text{mod } 23) \\ x_i \equiv 8^{15} (\text{mod } 17) \end{cases} \tag{12}$$

Then, using Chinese Remainder Theorem, x_i is obtained by the following formula:

$$x_i \equiv 17 \times 17 \times 17^{-1} (\text{mod } 23) + 8 \times 23 \times 23^{-1} (\text{mod } 17) \equiv 25 \tag{13}$$

3.3. Combined application of AES and RSA

Both AES and RSA have their advantages and disadvantages. With the help of the computer, AES can quickly encrypt large amounts of information. However, since AES is a symmetric key encryption, it is not secure as RSA. When p and q are large enough, RSA is almost impossible to attack, but the process of encryption and decryption can be time consuming. These limitations have made them hard to implement in some field. A combined application of AES and RSA may solve the problems to some extent.

3.3.1. A hybrid algorithm used in file encryption. File encryption requires an efficient, secure algorithm, so neither AES nor RSA can meet the requirement. A hybrid algorithm is proposed. Basically, the file is been encrypt for two times [9]. Firstly, AES is used to encrypt the file and generate first cipher-text C1. Then, the public key of RSA is used to encrypt the key of AES and C1 to get second cipher-text C2. During the decryption, the secret key of RSA is used to get C1 and the key of AES. Then, the key of AES is used to decrypt C1 to get the original file. Proposed in [10], this hybrid algorithm can secure the information even if an unauthorized third party knows the key of AES since only the secret key of RSA can decrypt C2. Also, this hybrid algorithm is more efficient than simply implement RSA. This is because AES divided text into many 128 bits sections, and each section is encrypted using RSA separately. A relative shorter section can enhance the efficiency of RSA.

3.3.2. A hybrid algorithm used in E-mail encryption. Symmetric encryption is widely used in E-mail encryption when sending sensitive and private information. However, the process of distributing the key can be risky. Also, since symmetric encryption does not have a digital signature, the identity of the information cannot be confirmed. A hybrid encryption using AES and RSA is proposed to solve such problem. As shown in Figure 5, two pairs of RSA key are generated, one pair is used to encrypted the AES key and plain-text, and another is used to sign the messages. This hybrid encryption implemented a digital signature to enable Bob and Alice to verify the identity of each other. Also, a leakage of RSA public key will not cause any risk since the secret key is not unknown for everyone except the designer of RSA key pair.

DOI: 10.54254/2753-8818/25/20240893

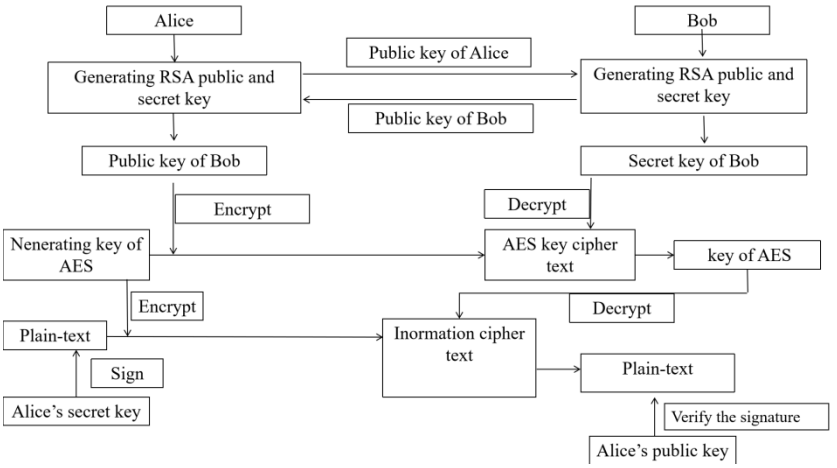

Figure 5. Process of Hybrid Encryption and Decryption (Photo/Picture credit: Original).

3.3.3. A Conclusion About hybrid algorithm using AES and RSA Algorithm. A hybrid encryption Using RSA and AES always uses AES to encrypt the data since AES is more efficient at encrypting large amount of data (it may also accelerate RSA algorithm since the text is divided into 128 bits sections). RSA is used to sign the message and to encrypt the key of AES. So, hybrid encryption avoids the process of distributing the key of AES and enables different parties to check the identity of each other.

4. Conclusion

The above discussion comprehensively explores the intricacies of encryption and decryption processes, emphasizing the utilization of formulas and theorems to enhance these operations. The text methodically explains the RSA digital signature mechanism and its crucial role in ensuring secure communication between entities. Additionally, it outlines the application of the Chinese Remainder theorem in decryption, augmenting the efficiency of the process.

Moreover, the discussion broadens to include the amalgamation of AES and RSA in a hybrid encryption algorithm. This dual application optimizes the encryption and decryption processes by harnessing the strengths of both algorithms, guaranteeing not only efficiency but also heightened security. Clear examples of hybrid encryption use cases in both file and email encryption provide a pragmatic understanding of its real-world application. In conclusion, this exploration offers profound insight into the realm of encryption, detailing processes, algorithms, and their diverse applications. It lays a robust foundation for comprehending the practical implementations and benefits of hybrid encryption mechanisms, especially the combined use of AES and RSA. The detailed exposition of processes and methodologies highlights the significance of employing stalwart encryption methods to safeguard communication and data, reinforcing the imperative for continued advancements in this essential technological field.

References

[1] Sahin, M. E. (2023). Memristive chaotic system-based hybrid image encryption application with AES and RSA algorithms. Physica Scripta, 98(7), 075216.

[2] Hamza, A., & Kumar, B. (2020, December). A review paper on DES, AES, RSA encryption standards. In 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 333-338). IEEE.

[3] Fatima, S., Rehman, T., Fatima, M., Khan, S., & Ali, M. A. (2022). Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing. Engineering Proceedings, 20(1), 14.

DOI: 10.54254/2753-8818/25/20240893

[4] Clavijo, A. M., Chacón, J. A., & Montiel, G. A. C. HYBRID ENCRYPTION PROTOTYPE COMBINING AES AND RSA ENCRYPTION METHODS. Libro de memorias.
[5] Ping, H. (2022). Network information security data protection based on data encryption technology. Wireless Personal Communications, 126(3), 2719-2729.
[6] Zhang, Q. (2021, January). An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption. In 2021 2nd international conference on computing and data science (CDS) (pp. 616-622). IEEE.
[7] Taneja, P., & Kalta, S. A Comparative Analysis of Cryptographic Algorithms: AES & RSA and Hybrid Algorithm for Encryption and Decryption.
[8] Lu, Z., & Mohamed, H. (2021). A complex encryption system design implemented by AES. Journal of Information Security, 12(2), 177-187.
[9] Zou, L., Ni, M., Huang, Y., Shi, W., & Li, X. (2020). Hybrid encryption algorithm based on AES and RSA in file encryption. In Frontier Computing: Theory, Technologies and Applications (FC 2019) 8 (pp. 541-551). Springer Singapore.
[10] Fatima, S., Rehman, T., Fatima, M., Khan, S., & Ali, M. A. (2022). Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing. Eng. Proc. 2022, 20, 14.