Comparative analysis of Conventional & Lightweight Security Algorithms for Multi-Level Encryption in IoT

Isha Sharma*, Monika Saxena,

Department of Computer Science, Banasthali Vidyapith, Newai, Rajasthan, 304022, India
Department of Computer Science, Banasthali Vidyapith, Newai, Rajasthan, 304022, India

Abstract

Today, when most businesses and organizations keep their data on clouds, network security is crucial for both users and organizations. Thus, a solution must be found to ensure data security while being transmitted through networks. The most effective method in this case is to encrypt the data that is transmitted from the source to destination. The original data is then once more decrypted from the encrypted data at the destination. Some of the data, such as some government information or military communications, banking, transaction, care is more important. When the data is in the wrong hands, it can have serious negative effects.

This study offers a multilevel encryption as a solution to the aforementioned problem. Data is safer with multilayer encryption than with standard encryption, which makes use of multiple rounds of encryption using the same or various keys, resulting in a sophisticated or powerful algorithm.

Keywords-Cryptography, Encryption, Decryption, Private-Key, Public-Key, RSA, PRESENT, Multilevel Encryption

1. Introduction:

As a lot of information is transmitted via the network today, data security is crucial. The best way to make data protected over networks is to use an appropriate privacy transformation methodology. Different strategies are used to safeguard the sensitive data. Nowadays, the majority of data is protected using encryption and certificate technology. The majority of techniques rely on cryptography [9].

A novel idea called multilevel encryption is employed to increase the system's security over previous cryptosystems. The method of multilevel encryption entails encryption plain text once or more using the same or different keys. It increases the process's complexity and power over what it was before.

2. RSA

The RSA algorithm was created at MIT in 1977 by Ron Rivest, Adi Shamir, and Leonard Adelman. An asymmetric key is used in the RSA algorithm. The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the most common and widely accepted general-purpose public-key cryptography methods. In RSA (public-key cryptography), a public key & a private key are used for encryption & decryption. Anyone can use the public key to encrypt information as well as data, however, only the private key can decrypt messages & data encrypted with the public key.

The RSA algorithm has three basic steps:

1**. Key generation:**

RSA employs both a public & a private key, with the public key encrypting messages and the private key decrypting them. The following is the key generation procedure:

a. Choose any two large prime numbers, P and Q.
b. Using the given formula, compute N. N=P * Q

c. Determine phi (N): Φ () = (P-1) * (Q-1), Φ (n) is Euler's totient.

d. Determine the public key exponent e such that 1 < e <Φ(n) and e, Φ(n) are co-prime, i.e. GCD (e, Φ(n)) = 1

e. Determine the private key exponents d such that d * e= 1 * mod (Φ (n)), where is the multiplicative inverse of e^{-1} mod (Φ (n))

Public key: (n, e)

Private Key: (n, d)

2. **Encryption:**

Plaintext: M <n

Cipher text: $C = M^E$ (mod n)

3. **Decryption:**

Cipher text: C

Plaintext: $M = C^D$ (mod n)

RSA has become the most widely used algorithm because it enables the public key is used to encrypt messages, while the private key is used to decrypt them. It ensures data confidentiality, integrity, authenticity, and untrustworthiness. It is relevant to the emphasis that RSA is extremely vulnerable to attacks due to poor key generation. As a result, the public & private keys are generated with the assistance of two large random prime numbers. Because it is difficult to compute factors, the RSA algorithm becomes effective. Multiplying two primes is simple, but doing the opposite to form an exact factor is difficult and becomes more difficult as the values of p and q increase. The longer the modulus's key length, the longer it takes to factor [14]. The RSA factoring challenge was a challenge to computational number theory researchers to crack RSA keys used in cryptography. Although the RSA100 was affected by April 1, 1991, many numbers are still unfactored and are expected to remain so for some time. Short's algorithm, on the other hand, is a polynomial-time quantum computer algorithm with for integer factorization developed in 1994. However, due to the qubits required by quantum computers, only a factor of 21 can be broken. As a result, no quantum computer has been built to date that can recover RSA factors [6].

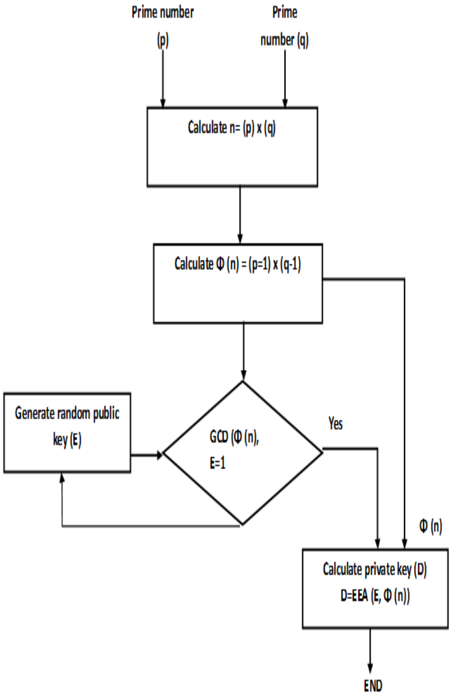

Prime number (p)
Prime number (q)
Calculate n= (p) x (q)
Calculate Φ (n) = (p=1) x (q-1)
Generate random public key (E)
GCD (Φ (n), E=1
Yes
Φ (n)
Calculate private key (D) D=EEA (E, Φ (n))
END

Fig 1: Process of RSA Algorithm [14]

3. Light Weight Algorithm Introduction

Low-power technologies pervade our daily lives, from home appliances to digital assistants to medical equipment. Given the low power conditions under which these devices operate, as well as the fact that they frequently contain valuable private information, a reasonable level of security is required [12]. Typical encryption techniques do not always perform effectively on these devices due to potential limitations in both the hardware (i.e. memory) and software (i.e. processing speed). Low power devices will struggle because they lack the processing power of smartphones and laptops, for example, if a video stream or a large stream of data must be protected quickly. Security suffers as tradeoffs are made in a low-power system because the available power is more limited. Smaller key sizes, for example, are desirable in such a setting, though performing so may reduce the system level of security. As a result, the goal of lightweight cryptography is to use as little memory, processing power, or other resources as possible while toll provides some level of security [10].

Memory size, processing performance, and latency may be software limits for lightweight devices. Lightweight hardware may have space, performance, and power consumption limits. Lightweight cryptographic algorithms that can provide a reasonable level of safety in a variety of applications are required when operating in these contexts.

Present Algorithm

The PRESENT algorithm is a block cipher algorithm with a symmetric key. It grew in 2007 in Orange Laboratories. It was designed as a (ultra-lightweight block cipher) suitable for lightweight cryptography in resource-constrained environments by the International Standards Organization in 2012[13].

The algorithm is used in devices that have restricted storage or consume little power (for example, Internet of Things devices). This algorithm’s implementation is based on data and keys. A single block of data entered into the encryption and decryption algorithm is 64 bits long. The executable key can be 80 or 128 bits long. Based on other researchers, the current algorithm key size is determined by the application’s level of security. Because some researchers implemented it or expected that the 128-bit key wouldn’t e helpful in practical applications, the 80-bit key was emphasized over the 128-bit key. With many lightweight models, the PRESNT algorithm introduced an advantage milestone in lightweight cryptography in 2007, resulting in the development of a lightweight block ciphers [5]

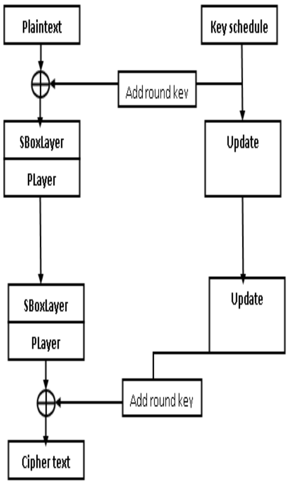


Fig 2: process of the present algorithm [9]

The following is the pseudo code to describe the PRESENT algorithm:

GenerateroundKeys ()

AddRoundKey (STATE, Ki) for i=1 to 31.

end for addRoundKey (STATE, K32), sBoxLayer (STATE)

pLayer (STATE)

The encryption procedure is as follows:

1) addRoundKey: XOR is the round key with the 64-bit input.
2) S-box: the S-box transformation is a non-linear 4-bit word substitution that operates independently on each of the state 4-bit words.
3) Player: A permutation transformation that works with the 64-bit state.

4.1 S box improvement ideas

4.2 Improvement of the PRESENT S-box

The PRESENT" s non-linear substitution layer employs a single 4-bit s-box. Such an s-box's implementation is typically far smaller than that of an 8-bit s-box. In hexadecimal notation, tab represents the function of this box. Nonlinearity, differential uniformity, immune correlation, avalanche effecting, or avoidance, or fixed or antifixed points are all desirable s-box achievements [4]. By using a genetic algorithm to design S-boxes, this work produces optimized s-boxes (s-box S1 and S-box S2) with diffusion rates, which resolves an anti-fixed-point difficulty in the PRESENT s-box. Tables 2 and 3 show the enhanced s-boxes.

Table1. The PRESENT'S –box

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(X)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table2. Improvement S-box S1

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(X)	C	6	1	4	9	0	A	D	3	E	F	8	B	7	5	2

Table3. Improvement S-box S2

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(X)	2	5	E	9	1	D	0	B	4	3	7	C	F	6	A	8

In the genetic algorithm, the enhanced methods in this paper use the concept of crossover and mutation. These crossover operators are used in pairs to exchange genetic data between people in larger groups [11].

Multilevel encryption: Diagram of 3 level architecture

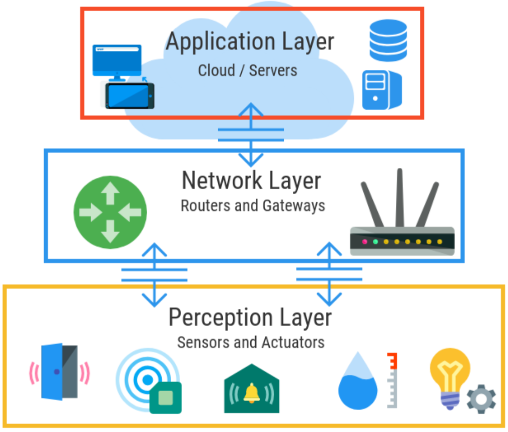


Fig1: Multi-Level Architecture [5]

4. Related work and implementation

The combination of the RSA and PRESENT techniques for symmetric & asymmetric encryption is not a new concept. For purchase researchers and experimenters, security is always a top priority. Some implementations have been done by researchers using a hybrid approach as well as a multilevel cryptography approach [7].

Security includes several phases that must be applied to the data, including RSA, DES, and a hybrid algorithm. String modulus, private, public, and integer keys were used in the previous step.

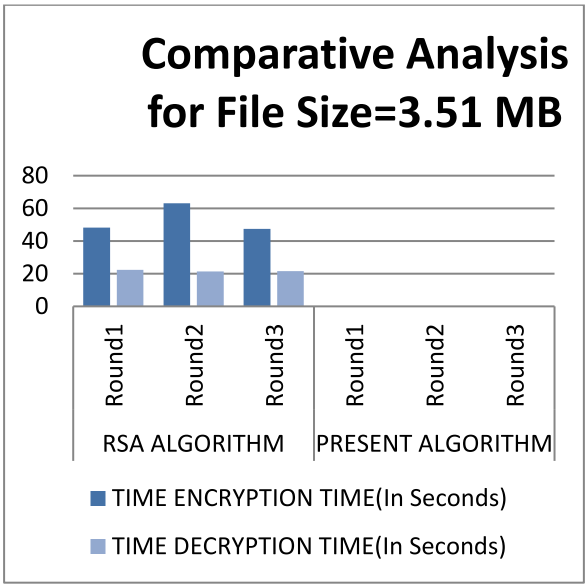
Another approach was to combine a symmetric and public key method for key exchange, where the latter was used for encryption. The RSA algorithm also includes a pseudo random number generation unit and a GCD computing unit for key generation.

3DS uses multilevel encryption, and AES provides cryptographic assurance of message integrity. The multiphase or multi-level encryption techniques encrypts plain text (messages) in single or multi-level (k times) encryption using the same of different keys. As a result, the overall system security is improved [8].

5. Results & Discussion

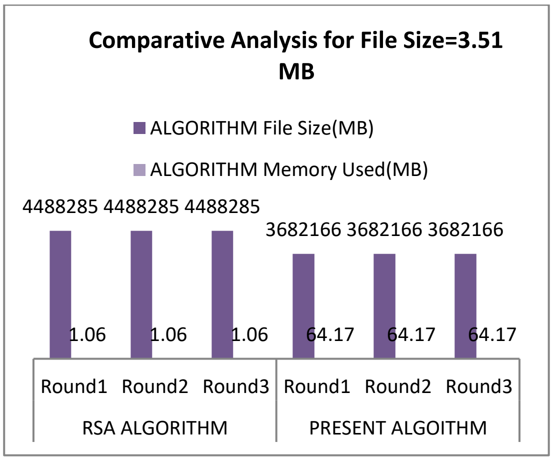
Graph results of RSA algorithm and present algorithm including time using small csv

ALGORITHM		**TIME**	
		ENCRYPTION TIME (In Seconds)	**DECRYPTION TIME (In Seconds)**
RSA ALGORITHM	Round1	48.14	22.27
	Round2	63.03	21.38
	Round3	47.4	21.49
PRESENT ALGORITHM	Round1	0.169	0.22
	Round2	0.195	0.215
	Round3	0.166	0.2



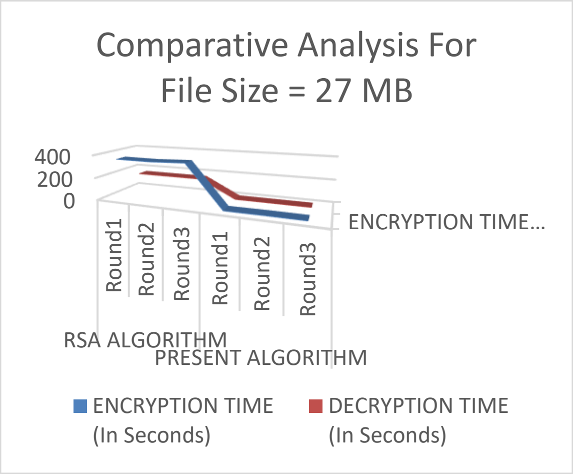
Graph results of RSA algorithm and the present algorithm including memory for small csv

ALGORITHM			
		File Size (MB)	Memory Used (MB)
RSA ALGORITHM	Round1	4488285	1.06
	Round2	4488285	1.06
	Round3	4488285	1.06
PRESENT ALGOITHM	Round1	3682166	64.17
	Round2	3682166	64.17
	Round3	3682166	64.17



Graph results of RSA algorithm and present algorithm including time using large csv

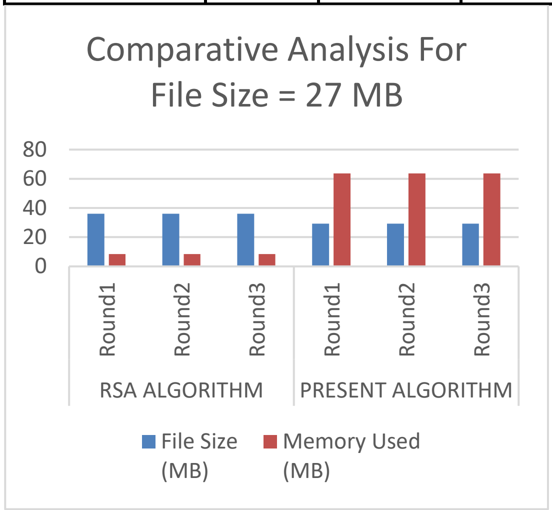
ALGORITHM		**TIME**	
		ENCRYPTION TIME (In Seconds)	**DECRYPTION TIME (In Seconds)**
RSA ALGORITHM	Round 1	359.45	160.92
	Round 2	357.6	160.74
	Round 3	368.41	160.66
PRESENT ALGORITHM	Round 1	1.761	1.917
	Round 2	3.772	1.693
	Round 3	2.243	1.792



Graph results of RSA algorithm and the present algorithm including memory for large csv

ALGORITHM			
		File Size (MB)	**Memory Used (MB)**
RSA ALGORITHM	Round1	36.165407	8.37
	Round2	36.165407	8.37

	Round3	36.165407	8.37
PRESENT ALGORITHM	Round1	29.195907	63.78
	Round2	29.195907	63.78
	Round3	29.195907	63.78



The above graph shows the three multilevel encryption using RSA and PRESENT algorithm.

6. Conclusion

The comparison of the PRESENT algorithm and RSA algorithm is that the size of the PRESENT algorithm is less. The size of RSA algorithm is more and the PRESENT algorithm time is more, so we have used the conventional algorithm. Multiple encryptions have the advantage of providing better security because the original data's confidentially can still be maintained even if a few of the component cyphers are broken or some of the secret keys are recognized.

7. Compliance with Ethical Standards:

Funding:

This study is not funded by any. There is no financial interest to report.

Conflict of Interest:

The authors have no conflict of interest to declare. All co-authors have seen and agree to the content of the manuscript and there is no financial interest to report. We certify that the submission is an original work and is not under review in any other publication.

Ethical approval:

This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions:

We certify that the submission is an original work and is not under review in any other publication.

Availability of supporting data:

We certify that the submission is an original work and is not under review in any other publication.

REFRENCES

1. Padhiar, S., & Mori, K. H. (2022). A Comparative Study on Symmetric and Asymmetric Key Encryption Techniques. In *Implementing Data Analytics and Architectures for Next Generation Wireless Communications* (pp. 132-144). IGI Global.
2. Fujisaki, E., & Okamoto, T. (2013). Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, *26*, 80-101.
3. Qadir, A. M., &Varol, N. (2019, June). A review paper on cryptography. In *2019, the 7thinternational symposium on digital forensics and security (ISDFS)* (pp. 1-6). IEEE.
4. Tang, Z., Cui, J., Zhong, H., & Yu, M. (2016). A random PRESENT encryption algorithm based on dynamic S-box. *International journal of security and its applications*, *10*(3), 383-392.
5. Lara-Nino, C. A., Morales-Sandoval, M., & Diaz-Perez, A. (2016, February). An evaluation of AES and present ciphers for lightweight cryptography on Smartphone. In *2016 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (pp. 87-93). IEEE.
6. Zhou, X., & Tang, X. (2011, August). Research and implementation of RSA algorithm for encryption and decryption. In *Proceedings of 2011 6th international forum on strategic technology* (Vol. 2, pp. 1118-1121). IEEE.
7. Gutub, A. A. A., & Khan, F. A. A. (2012, November). Hybrid crypto hardware utilizing symmetric key and public-key cryptosystems. In *2012, international conference on advanced computer science applications and technologies (ACSAT)* (pp. 116-121). IEEE.
8. Gupta, H., & Sharma, V. K. (2013). Multiphase encryption: A new concept in modern cryptography. *International Journal of Computer Theory and Engineering*, *5*(4), 638.
9. Isha, Saxena, M., & Jha, C. K. (2022). Multilayered Architecture for Secure Communication and Transmission for Internet of Things. In *Soft Computing for Security Applications: Proceedings of ICSCS 2022* (pp. 691-699). Singapore: Springer Nature Singapore.
10. Sharma, I., & Saxena, M. (2023). A Review of Lightweight Cryptographic Algorithms. *Available at SSRN 4366916*.
11. Katuk, N., & Chiadighikaobi, I. R. (2022). An Enhanced Block Pre-Processing of PRESENT Algorithm for Fingerprint Template Encryption in the Internet of Things Environment. *International Journal of Communication Networks and Information Security (IJCNIS)*, *13*(3).
12. Panahi, P., Bayılmış, C., Çavuşoğlu, U., & Kaçar, S. (2021). Performance evaluation of lightweight encryption algorithms for IoT-based applications. *Arabian Journal of Science and Engineering*, *46*, 4015-4037.
13. Kubba, Z. M. J., & Hoomod, H. K. (2020, November). Modified PRESENT Encryption algorithm based on the new 5D Chaotic system. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032023). IOP Publishing.
14. 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.