Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

A Review of Lightweight Cryptography Algorithm for Healthcare using Multi-Level Encryption

Isha Sharma, Monika Saxena

Department of Computer Science Banasthali Vidyapith, Rajasthan, India

***Abstract*—**In this day and age, 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 device level security also plays important role to secure data. Now a day we need to secure device level, network level and application-level data together. In the proposed concept authors try to secure data on all three levels using lightweight cryptography algorithms and multi-level encryption methods.

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

***keywords*—** *Cryptography, Lightweight RSA, Lightweight PRESENT, Multilevel Encryption, Healthcare*

Introduction

As a lot of information is transmitted via 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 multi-level encryption is employed to increase the system's security over previous cryptosystems. The method of multi-level 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.

As more IoT devices connect to the internet and exchange data with one another, the importance of lightweight cryptography has grown. Sensitive data must be encrypted to be protected. Lightweight cryptography is the application of cryptographic algorithms and protocol to resource-constrained environment, like embedded systems, smart cards, and Internet of Things devices. These systems need cryptography designed specifically for them because they have limited memory, energy and computational capacity. PRESENT [4], a privacy enhanced reduced sized-block cypher was introduced by [3] as a very thin block cypher.

High security, low power consumption, and small code size were all considered when creating the cypher. Since its debut, present has become more and more well-known in the field of cryptography, and it is currently utilized in many different applications, including smart cards, RFID tags, and wireless sensor networks.

2. Security overview of lightweight cryptographic algorithm:

To secure data, cryptographic algorithms are employed. A method for encrypting data into cypher text for safe transmission is called cryptography. Symmetric and asymmetric cyphers are the two categories of cryptographic cyphers. The same key is used in symmetric key encryption to encrypt and decrypt data. This encryption technique is comparatively quick and incredibly safe. The primary drawback ofsymmetric key encryption is the sharing of the key between communicating parties. The encryption of the data is compromised if the key is obtained by an adversary. Data confidentially and integrity are guaranteed by symmetric key algorithm, but authentication is not. AES, DES, 3DES, BLOWFISH, and Other traditional symmetric key ciphers are examples. Authentication, confidentiality, and integrity are all provided by asymmetric encryption. To ensure confidentiality and integrity, the sender encrypt data with his public key, and receiver decrypt it with his private

key. The sender encrypt data with his private key to ensure authentication, and the recipient verifies it by decrypting it using the sender's public key [15]. One benefit of asymmetric cryptography is that it supports all security mechanisms, including key sharing. The only drawback is the size of the keys, which increase complexity and slow down encryption. Elliptic Curve cryptography (ECC), Deffie-Hellmen key exchange (DH), and Rivest, Shamir and Adleman's RSA are the most widely used algorithms.

3. Light Weight Algorithms

Low-power technologies pervade our daily lives, from home appliances to digital assistants to medical equipment. Given the low power condition under which these devices operate, as well as the fact that they frequently contain our 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 available power is more limited. Smaller key sizes, for example, are desirable in such a setting, through 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 providing 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 algorithm that can provide a reasonable level of safety in a variety of applications are required when operating in these contexts.

4. Lightweight RSA Algorithm

The assumption that underpins the RSA asymmetric key encryption system is difficult to find the factor of large integers. The public key is distributed to all members of the framework, while the private key in RSA is kept secret. The RSA algorithm uses three steps: key generation, message encryption and message decryption [14]. The following steps are displayed:

4.1 key generation

- choose two arbitrary large prime integers, pn and qn, that are typically of comparable size.
- Determine N= pn×qn.
- Determine φ(N) = (pn -1) × (qn-1).
- Select an integer p such that gcd (P, φ (N)) = 1; P <ex< φ (N).
- Find the exponent of decryption, d, such that d =1 mod φ(N).
- At this point, the public keys are d and N (d, N), and the private key are ex and N (ex, N).

4.2 Encryption of RSA

the sender follows these steps to encrypt the message "Me": the cypher text generated after encryption is denoted by "C", and its formula is Cm = Meex Mod N.

the sender's public key is used by user B to encrypt the message "Me" in the RSA encryption step.

4.3 Decryption of RSA

Using the private key "d", the reception should take the following action to obtain the message "Me" or plaintext from "Cm":

Me Cmd mod N.

4.4 Proposed RSA

A method that speeds up the creation and decryption of RSA keys is one of the ways the suggested scheme improves the existing RSA technique.

To make the RSA algorithm more complex, a new element called "s_n" was introduced. The presence of three prime numbers instead of two means that the time required to generates keys must be reduced, and the variable "N" analysis difficulty must increase.

4.5 The Proposed RSA-based generation

the following actions must be taken by user "A" in order for the key to generate in a three-prime dependent manner:

- There are generated three large prime numbers: pn, qn and sn.
- Compute N =pn×qn×sn.
- Determine that φ(N) = (pn -1) (qn -1) (sn-1). In order for gcd (ex, φ (N)) = 1, ➢ selects e, 1 <ex< φ (N).
- Find d such that mod φ (N) = ex×d.
- Determine dp such that ex×dp = 1 mod (pn-1) and dq such that ex×dq = 1 mod (qn -1).
- Finds Qin such that

o qn ×Qin = 1 mod pn. if pn>qn
o pn ×Qin = 1 mod qn. if qn>pn

Public key Ku= (ex, N) and private key Kr= (Qin, dp, dq, pn, q).

4.6 The Proposed LWRSA-based encryption

user "B" encrypts the message "Me" by carrying out the subsequent actions:

- user "B" should get the public key of user "A" (ex, N)
- cypher text Cm = Meex Mod N.

4.7 Decryption for the proposed RSA

We apply the idea of RSA with CRT to the decryption process in order to retrieve the message from cipher-text Cm. the recipient needs to take the following actions:

- Ma = Cdp mod pn.
- Mb = Cdq mod qn
- h= (Qin×(Ma-Mb)) mod pn
- Me =Mb+(h×qn) = plaintext

Th proposed RSA algorithm is flowcharted in figure 1, which is provided below and is recommended in this paper.

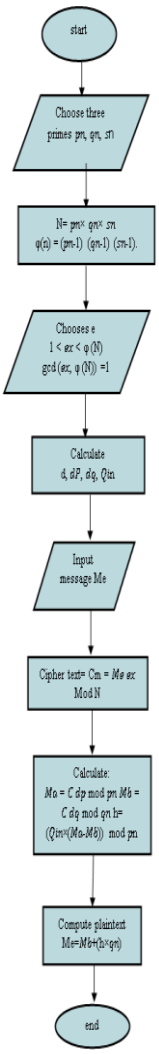

Fig1: Process of LW RSA Algorithm [14]

5 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 environment 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 to other researchers, the current algorithms key size is determined by the application's level of security. Because some researcher implemented it or expected that the 128-bit key wouldn't be helpful in practical application, 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 a development of lightweight block cipher [5].

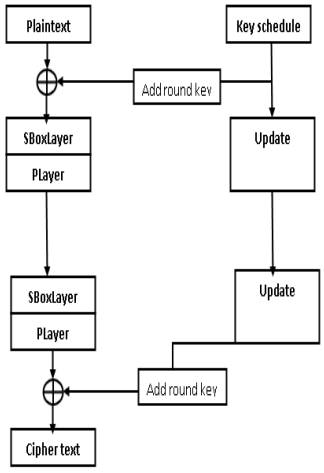

Fig 2: process of present algorithm [4]

The following is 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 follow:

1) addRou
ndKey: XOR 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 work with the 64-bit state.

5.1 Improvement of the Present S-box

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 anti-fixed 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 enhanced s-boxes.

Table1. The PRESENT'S –box [4]

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[4]

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[4]

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].

6. Related Work & Proposed Methodology

The most typical and well-known architecture consists of three layers. This IoT study was first implemented in its early stages. Application, network and perception layer are the three levels it indicates.

Application layer: this layer is where the user interacts. It is in charge of giving the client access to software resources. An illustration would be a smart home app where user could press a bottom to turn on a coffee maker, for instance. Application-specific resource is supplied to the customer by the application layer. It lists servals applications for the Internet of Things, including smart homes, smart cities, and smart healthcare [17].

Network layer: it is necessary to distribute and store the data that these devices collect. The network layer bears responsibilities for this. It connects these clever objects to other clever and intelligent objects. Data transfer is under its purview as well. Servers, network devices, and smart objects area all connected by the network layer. Sensor data distribution and analysis are also done with it [18].

Perception layer: The IoT architecture's physical layer is this perception layer. The primary components utilized in these are sensor and embedded systems. According to the requirement, these gather a IoT of data. Connectivity with the environment is also facilitated by edge devices, sensors, and actuators. It can identify other intelligent things or objects in the environment, or it can identify specific spatial parameters [16].

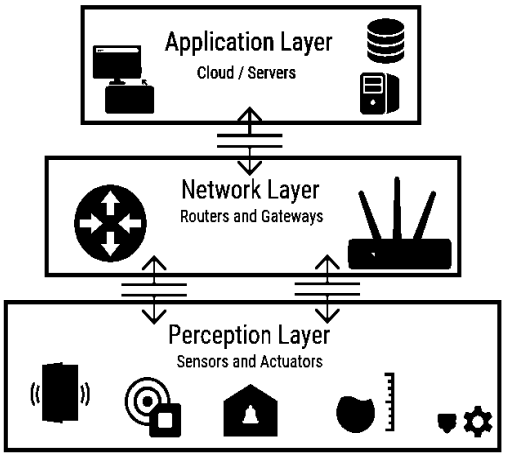

Fig 3: Multi-level Architecture [17]

We have collected healthcare datasetfromKaggle and applied multi-level encryption on given dataset by using lightweight RSA algorithm and present algorithm. Lightweight RSA is performed with key value 64 bit and 3 numbers of rounds and present algorithm is also performed with key value 44 bit and healthcare dataset is collected of 2 sized 3.51 MB and 27 MB which is considered as minimum and maximum size of data to calculate time and space complexity.

The LW RSA algorithm also includes a pseudo random number generation unit and a GCD computing unit for key generation.

7. Implementation&ComparativeResults

As per proposed methodology algorithms are applied on two different sizes of healthcare data. The following are comparative results of LW RSA and present algorithms in form of graphs and tables.

4.1Comparative Time complexity of LWRSA algorithm and present algorithm for small size healthcare data (3.51 MB)

Table 4: Time Complexity

Algorithm	Level	Time (In Sec)	
		Encryption	Decryption
LW RSA algorithm	L1	0.0290	0.0350
	L2	0.0240	0.0240
	L3	0.0160	0.0240
PRESENT algorithm	L1	0.484	0.604
	L2	0.502	0.616
	L3	0.486	0.600

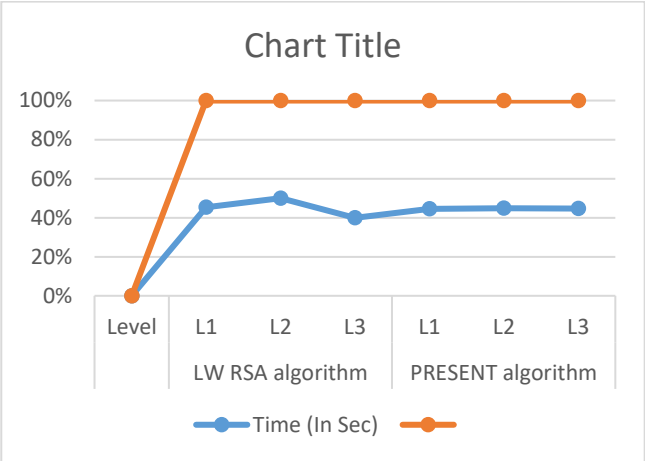

Fig4: Computational Time using Multi level Encryption

Description: we have collected healthcare dataset and applied multilevel of encryption on given dataset by using Lightweight RSA and present algorithm to calculate time complexity.

4.2 Comparative Time complexity of LWRSA algorithm and present algorithm forlargesize healthcare data (27 MB)

Table5: Time Complexity

Algorithm	Level	Time (In Sec)	
		Encryption	Decryption
LW RSA algorithm	L1	0.1195	0.320
	L2	0.160	0.290
	L3	0.200	0.230
PRESENT algorithm	L1	10.009	5.997
	L2	10.406	5.848

	L3	5.168	5.239

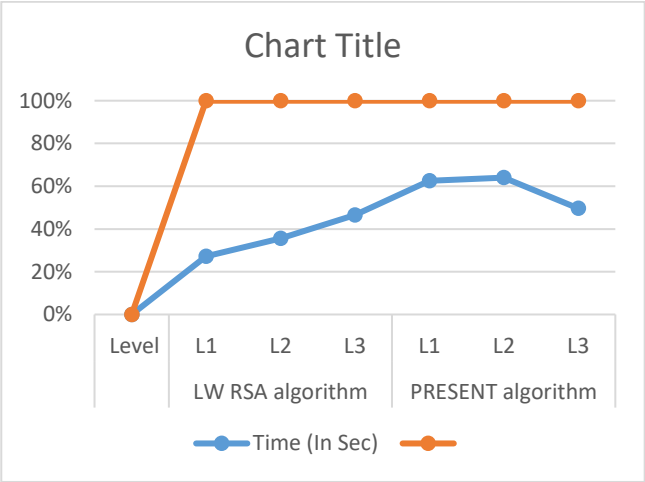

Fig5: Computational Time using Multi level Encryption

Description: we have collected healthcare dataset and applied multilevel of encryption on given dataset by using Lightweight RSA and present algorithm to calculate time complexity.

4.3 Comparative Analysis of LW RSA Algorithms

	Level	Data Size (IN MB)	Memory Used (Bytes)	Key Size (Bits)	Key Genration Time (Sec)	No of Rounds	Encryption Time(In Sec)	Decryption Time(In Sec)	Total Time(E+D)
LW RSA algorithm	L1	3.51	722	2048	1.9665	12	0.029	0.035	0.064
	L2	3.51	737	2048	1.9665	12	0.024	0.024	0.048
	L3	3.51	783	2048	1.9665	12	0.016	0.024	0.04
	L1	27	720	2048	2.1473	12	0.1195	0.032	0.1515
	L2	27	783	2048	2.1473	12	0.016	0.029	0.045
	L3	27	753	2048	2.1473	12	0.02	0.023	0.043
	L1	3.51	1545	4096	15.492	13	0.025	0.0819	0.1069
	L2	3.51	1536	4096	15.492	13	0.027	0.0692	0.0962
	L3	3.51	1494	4096	15.492	13	0.019	0.0646	0.0836
	L1	27	1525	4096	5.2873	13	0.0657	0.08	0.1457
	L2	27	1476	4096	5.2873	13	0.016	0.1292	0.1452
	L3	27	1446	4096	5.2873	13	0.0589	0.1159	0.1748

4.4 Comparative Analysis ofpresent Algorithms

	Level	Data Size (IN MB)	Memory Used (Bytes)	Key Size (Bits)	Key Genration Time (Sec)	No of Rounds	Encryption Time(In Sec)	Decryption Time(In Sec)	Total Time(E+D)
present algorithm	L1	3.51	4896612	2048	1.9748	10	0.484	0.604	1.088
	L2	3.51	4896612	2048	1.9748	10	0.502	0.616	1.118
	L3	3.51	4896612	2048	1.9748	10	0.486	0.6	1.086
	L1	27	38817612	2048	1.169	10	10.009	5.997	16.006
	L2	27	38817612	2048	1.169	10	10.406	5.848	16.254
	L3	27	38817612	2048	1.169	10	5.168	5.239	10.407
	L1	3.51	4896612	4096	1.333	10	0.835	1.269	2.104
	L2	3.51	4896612	4096	1.333	10	0.817	1.033	1.85
	L3	3.51	4896612	4096	1.333	10	1.124	1.379	2.503
	L1	27	38817612	4096	1.9665	10	8.058	5.07	13.128
	L2	27	38817612	4096	1.9665	10	9.168	5.735	14.903
	L3	27	38817612	4096	1.9665	10	7.038	5.292	12.33

We have collected 3 key size of data 570, 2048 and 4096. We have worked on medium size of data. When we apply time complexity on the key size of 2048 bits, the time get decreased and we apply memory complexity, memory size is increase. As we know IoT is work with low memory.

Conclusion

As per the results Lightweight RSA performed very well with the comparison of present algorithm. So, we can use lightweight RSA in IoT based applications like smart cities. Lightweight RSA gives better performance using multi-level encryptions. So, either we can use lightweight RSA in multi-level or hybrid algorithm for better device level security.

Compliance with Ethical Standards:

Funding:

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

Conflict of Interest:

There are no disclosed conflicts of interest for the writers. The manuscript's content has been reviewed by all co-authors, who agreed with its contents and have no financial interest to disclose. We attest that the submission is our original work and isn't being considered for publication by another outlet.

Ethical approval:

None of the writers of this article have ever conducted any research on humans and animals.

Declaration:

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

Availability of supporting data:

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

Author Contribution

Author Isha Sharna and Monika Saxena confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

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 7th international 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 Algorithm. *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 for Science and Engineering*, *46*, 4015-4037.
13. Kubba, Z. M. J., &Hoomod, H. K. (2020, November). Modified PRESENT Encryption algorithm based on 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.
15. Abd Zaid, M. M., & Hassan, S. (2018). Lightweight RSA Algorithm Using Three Prime Numbers. *Int. J. of Engineering & Technology*, *7*(4.36), 293-295.
16. Khattak, H. A., Shah, M. A., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems*, *100*, 144-164.
17. Swamy, S. N., Jadhav, D., & Kulkarni, N. (2017, February). Security threats in the application layer in IOT applications. In *2017 International conference on i-SMAC (iot in social, mobile, analytics and cloud) (i-SMAC)* (pp. 477-480). IEEE.
18. Peng, H., Liang, L., Shen, X., & Li, G. Y. (2018). Vehicular communications: A network layer perspective. *IEEE Transactions on Vehicular Technology*, *68*(2), 1064-1078.