Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

Internet of Things Security using New Chaotic System and Lightweight AES

Jolan Rokan Naif [1]
informatics institute for postgradguate studies
iraqi commission for computers & informatics
Baghdad, Iraq
lnewjolan@gmail.com

Ghassan H. Abdul-majeed[2]
Ministry of Higher Education & Science Research
Baghdad, Iraq
ghassan@uob.edu.iq

Alaa K. Farhan[3]
dept. of computer sciences
Unversity of Technology of
Baghdad, Iraq
dralaa_cs@yahoo.com

Recived : 11\2\2019 Revised : 11 \3 \ 2019 Accepted : 17\3\2019

Abstract:
The Internet of Things (IoT)services and application were increasing during the last years in several life fields causes need to provide a secure identifier for protecting sensing data passing between IoT sensors/devices and embedded-subsystem connected by networks. This paper was proposed an algorithm for helping in IoT communication security can used in different IOT entities used in unofficial industrial machine to machine (M2M) communications, smart energy-grids, home or buildings and other computing devices.
This paper proposed a secure system using new proposed 4D chaotic system combined with the modified lightweight Advanced Encryption Standard (AES). The proposed 4-dimension (4D) chaos system Lyapunov was tested and pass for many initial periods and get a super chaos system (4 positive Lyapunov). Generated chaos keys (used JORN) were used in the lightweight AES and the Secure Hash Algorithm version 3 (SHA3-256). The Lightweight AES was design in case to reduce CPU computation cycles and complexity of AES. Results show that computation time for proposed system decreased (has 145% speedup more). The output of modified lightweight AES encryption System has the good statistical tests near to original AES that can avoid many attacks.
.

Keywords : IoT, IoT security, AES, chaos and AES.

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

1-INTRODUCTION

In the daily life, the Internet of Things (IoT) take an important work field due to the different uses of IoT sensors/devices data collecting through networks or Internet. The sensors and devices needed to achieve the security in transferring the sensing data. They need to the security issues, like communication (encryption, authentications and authorizations, secure protocols, secure routing, and other data and networking security) [11].

While the low-power consumption communication scheme with high secure named the Secure Low-Power Communications (SeLPC) algorithm was proposed decreasing the encryption-cycles of AES for reducing end devices data encryption-power. In the SeLPC, the enhance security levels using the encryption-key and D- Box update procedure. Also, the AES encryption processing simplification can reduce the power consumption. The SeLPC tests results show it was resisting attacks like known key, replays, and eavesdropping attacks, with minimized the encryption-power up to 26.2% comparing with the traditional- AES make it applicable with IoT environments. [8]

Lightweight security techniques are an aspiring and wish for fields which use to inspect the cryptographic primitive's implementation and algorithms for protect resources and devices outputs.

Lightweight security cryptography contains more one proposals algorithm like PRESENT(a lightweight block cipher with a block size of 64 bits and a key size of 80 or 128 bits), CLEFIA (Its name is derived from the French word clef, meaning "key"), KATAN(A Family of Small and Efficient Hardware-Oriented Block Ciphers), HEIGHT(high security and light weight), SIMON-SPECK, Fantomas, KLEIN (family of block ciphers)and many other algorithms. Lightweight security goals to gain sufficient security-levels with an optimum resource use. Block encryption algorithms (have permutation) are brief pointed in Table 1 [4].

Table 1. Comparison of lightweight block ciphers used in bit permutation.[4]

Name of the block cipher	Input block size	Key size	No. of rounds	Algorithm Design Pattern
HIGHT	64	128	32	GFN
Pickolo	64	80/128	25/31	GFN
PRESENT	64	80/128	31	SPN
DESLX	64	184	16	Feistel
Midori	64/128	128	16/20	SPN
mCrypton	64	64/96/128	12	SPN
AES	128	128/192/256	10/12/14	SPN
Clefia	128	128/192/256	18/22/26	GFN

The Primary target of this article is to give a proposed for IoT encryption based on the modified lightweight Advance Encryption Standard AES and new proposed 4-Dimension (4-D) chaotic system in order to achieving a stratify sensing data encryption in less time for all devices connected to the internal sensors network.

In [3] proposed modified of Advanced Encryption Standard (AES) with a lightweight issue. A novel equation for constructing was used to generate the one-dimensional Substitution-Box in Modified Advanced Encryption Standard (MAES) transformation phase. MAES efficiency rate for packet transmission term is around 18.35% that indicates MAES energy consumes is less than AES and applicable for IoT Resources.

In [10] explores the AES duration-time and energy consumptions using various key size in resource constrained IoT edges (hardware and software). Results illustrated that the hardware is buffer size sensitive and only consumes lower-power when the large buffer sizes, and faster default CPU-cycles rates with consume fewer resources. While the security increased key size but increased resource consumption.

In [7] proposed a modified AES algorithm with secret key-bio-chaos (generate using biometric (Fingerprint) and combine Lorenz-Lu). Modified AES algorithm uses same operations of original AES except for Mix-Columns operation and compensation for it in two XOR and shift-cycle operations as well as has two keys-bio-chaos which mentioned earlier in the process of Generate Key-bio-chaos, each key has size 4*4 byte that changing random values completely in each encryption or decryption for email messages [7].

[1] focuses on AES algorithm security to enhance the AES level security. The main modification was proposed to enhance the traditional AES algorithm is XORing an additional byte with Sbox values, and random additional key was also used to increase AES security, but these modifications caused an increasing in the time Security and Strict-Avalanche Criterion.

In [2] proposes a method to enhance the overcomes of the fixed Sbox to improves the AES performance to be applicable in large image encryption. The other modification is replacing the MixColumn stage by chaotic-mapping with XOR operation to reduce the high computations in MixColumn transform. The results show that the proposed method was a very low-correlation encrypted adjacent pixels data coefficients with high speed security.

The [9] explain the investigates and explores the behavior of the AES algorithm by replacing two of its original modules, namely the S-Box and the Key Schedule, with two other chaos-based system. In design of the proposed system, three chaos modules are used (Lorenz system, Chen system, and 1-D multi-scroll system). While the initials generated by Pseudo

Random Number Generators (PRNG) for the three chaotic systems.

While in [5] the key security was focused on. Here, the modified AES has been tested and simulated by using different chaotic variations (1-D logistic map, cross map, and combination of both systems) . For the evaluation purpose, the CPU time has been taken as the parameter. The testing results of the modified AES algorithm illustrated that it takes more time and CPU time comparing with traditional AES algorithms but stall sensitive to the key used.

II. *TAXONOMY LIGHTWEIGHT CRYPTOGRAPHY FOR THE IOT*

The IoT becomes widespread in many computing fields of life, like embedded computing device, health care ,..etc. The main challenge is how to reduce power consuming. The IoT devices widespread needs more security algorithms in their communication/connection that leads to the lightweight cryptography approaches.

Lightweight cryptography techniques have become crucial area for researchers in IoT, mobile device, and embedded computing devices security. Hardware lightweight cryptography techniques deals with to increase the performance parameters like device area size, and amount of powers consumption. While the software based lightweight cryptography, techniques deals with decreasing the CPU -memory usage, device computing, computing complexity, and amount of energy -powers consumption. For getting a Lightweight encryption algorithm must build with International Organization for Standardization (ISO) lightweight cryptography standards ISO/IEC (29192-2P:2012) [4].

Many encryption/hashing algorithms as AES, 3-DES, CLEFIA, PRESENTs and SHA have tested their security and proven are well but not to used extensively in different sensors- devices due to consume device resources such as memory, CPU cycles, and power due the high-complexity. Hence, these lead to a lightweight security approaches [4].

III The proposed system

The proposed IoT security system contain two main functions: encryption (using chaos-modified lightweight AES), and Authentication using Hash techniques (chaos-SHA3-256bit). There are three stages must clear in this proposed IoT sensing data encryption system:

A.THE PROPOSED LIGHTWEIGHT MODIFIED AES

The main goal of this work is to design a lightweight encryption algorithm that can be used to protect Internet of Things (IoT) sensor data and that balances the demands for performance and speed. To achieve a lightweight algorithm, many different modifications were made to the Advanced Encryption Standard (AES) algorithm along with some reductions. The proposed Modified Lightweight AES algorithm uses the same operations as the original AES except for the MixColumns operation (which it compensates for in multi-XOR stages), shift-cycle operations and SHA3-128. The modified algorithm also has four chaos-keys (as shown in Figure 1).

The first stage of the proposed modified Lightweight AES algorithm (MLAES) is to use the generated 4-Dimension(4D) chaos keys that generated from the proposed new 4D chaotic system (named JORN) using equation (1). The chaos keys are (k1, k2,k3 and k4) used in encryption processing, shifting cycle number calculation, and AES round iteration number counting.

$$\left.\begin{aligned} xt[i+1] &= xt[i] - a*(1-xt[i]-zt[i])*dt \\ yt[i+1] &= yt[i] + (-yt[i]-xt[i]*zt[i]+r*yt[i]^2)*dt \\ zt[i+1] &= zt[i] + (xt[i]*yt[i]-b*zt[i])*dt \\ kt[i+1] &= kt[i]+(c * yt[i] * (xt[i]*zt[i] - kt[i]))*dt \end{aligned}\right\} \quad \ldots. (1)$$

Where a, b, c, r is the chaos parameters. While xt,yt,zt, kt and is the initial conditions for chaos map.

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

In this paper was proposed some modifications to the AES to reduce the complexity of computations, number of iterations, and execution time, and to save memory. One modification to the AES is to combine it with a 4D JORN chaos keys (K1, K2, K3, and K4). These chaos keys are used to increase the randomness of the encrypted results and increase the strength of the AES to avoid more attacks. The key expansion converted the 128-bit key length into several subkey arrays and the total number of iterations required to generate all required subkeys. In the MLAES, each key block has a size (4x4 bytes) that changes values completely in each encryption/decryption iteration.

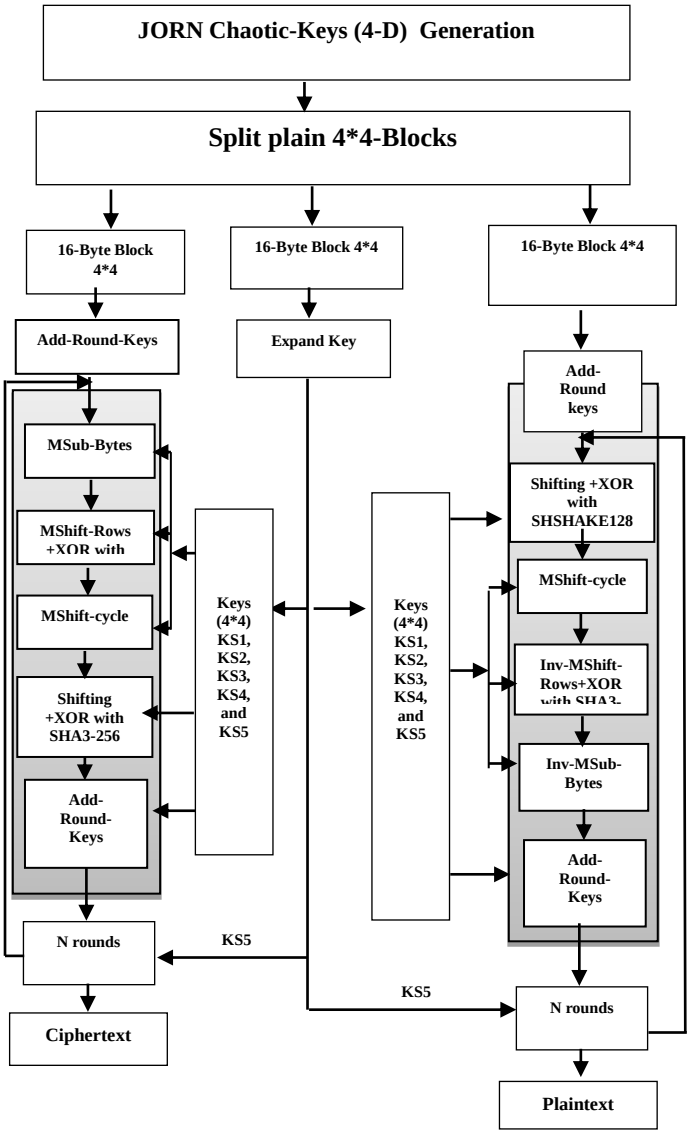


Figure 1: Encryption and Decryption Process of Modified Lightweight AES

The first proposed modification to the AES is enhancement of the Sub-bytes (s-box) by using the chaos keys block, Plain data block, logical function, with shifting operations. MLAES uses two s-boxes, with each s-box array consisting of 256 individuals with 64 bits each (S1 - 0…255, S2 - 0…255).

The second modification makes shift-rows and shift-cycles deal with a dynamic number of shifts. The number of shifts will be extracted from the last significant numbers of K4 and K2, which change during each iteration.

The third modification is adding shift-XORed with a SHA3-256 process. In this process, the SHA3-256 will generated using KS5 and K1 (as initial values for the SHA3-256 bit). The 128-hash bit will be divided into two 64-bit blocks. Also, the 128-bit data block will be divided into two 64-bit blocks. Each data block will be shifted by a different number of shift cycles. The first data block is shifted by X1, where X1 is the last significant number of K1. The second data block will be shifted by X2, where X2 is the last significant number of K3. After applying the shifting operations, the data blocks are XORed with the SHA3-256 blocks. Also, Rows will be shifted using K4 and XORed with the new result hashed from the SHA3-256.

Finally, AES uses three key lengths: 128, 192, and 256, with 10, 12 and 14 iteration cycles, respectively, as well as the proposed MLAES designed to use a dynamic number of iterations (in the range of 4 to 9 iterations). The number of iterations is elicited from the last two significant numbers of KS5. These proposed modifications aim to make AES lightweight and reduce its operation time with acceptable encryption complexity and strength. The dynamic number of iterations depending on the chaos key (K4) will provide more security to MLAES and protect against key-round guesses and weak keys attacks.

B. The Proposed Modified Sub-Bytes(S-Boxes)

One of the complex operations in MLAES is the S-Box. It affects most MLAES security. In the general case, the S-Box accepts a stream of 128-bit data; this block of data is split into 16-bit blocks (as a, b, … up to the last 8 bits). Each 64 bits are utilized with S-box (there are two S-boxes proposed for use in this modified Sub-bytes). The S-Boxes shifted at each iteration using K1 to generate new S-box values.

C.The proposed IoT Security System

One of the main challenges in designing the IoT application is that the time of the processing and complexity. The proposed system was designed to reduce the processing time and decrease number of CPU cycles with acceptable security mechanism compatible with IoT devices and sensors. The proposed security operation for an IoT system contains two stages: data encryption using the proposed MLAES algorithm and data hashing using SHA3-256 bits. Figure 2 shows the block diagram of the proposed IoT system.

The first step in the IoT system is the collection of sensing data from the sensors/devices connected to the IoT system. In this work, 40 sensors were used in testing the proposed security system. These sensors were grouped into 10 groups. Each group contained four different types of sensors

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

controlled by Raspberry Pi type B units. The data from each group of sensors were collected and aggregated during slice times.

The second step is to apply the combination of the lightweight IoT security proposed encryption-hashing techniques (MLAES-128 bits and SHA3-256 bits) to the collected data before sending them to the IoT server. The proposed security is implemented and applied in each group controller (Raspberry Pi) to encrypt and hash sensor data.

Figure 2 illustrates the proposed IoT system flowchart (using the proposed MLAES-hash (Raspberry Pi/sensors side)). It shows that the steps of the IoT security system operations, all parameters and initial values will be determined between the two sides (sender Raspberry Pi side, and IoT server side). Meanwhile, Figure 3 shows the proposed IoT security system using the proposed MLAES-hash (IoT server side). On the IoT server side, the encrypted payload data are decrypted in received packets using the steps in Figure 1. Then, the decrypted data hash will be generated by applying the proposed SHA3-256 bit on the decrypted payload data, and it will be compared with the hash stored in the received packet. The packet will be accepted or refused (dropped) depending on the comparison results.

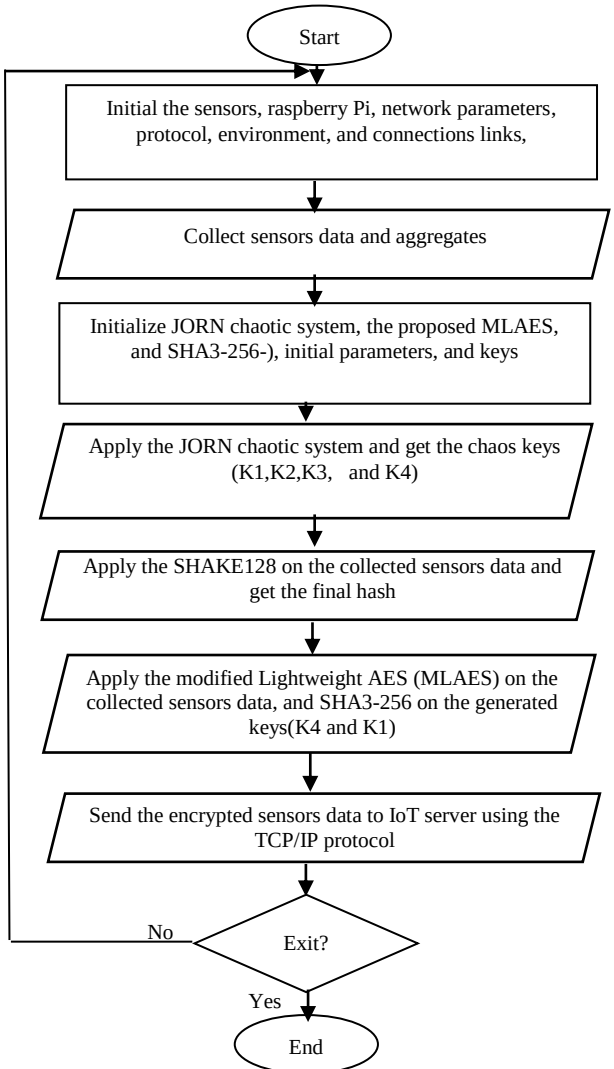


Figure 2 The Proposed IoT Security System Using Proposed Modified Lightweight AES-Hashing (Raspberry Side).

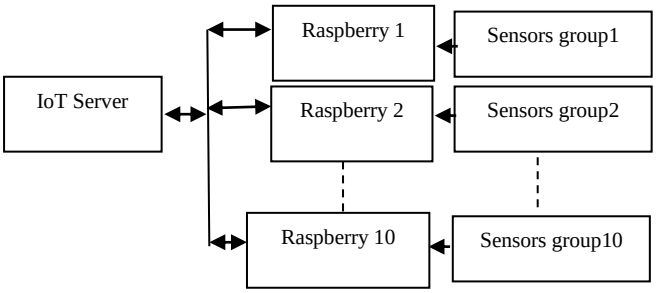
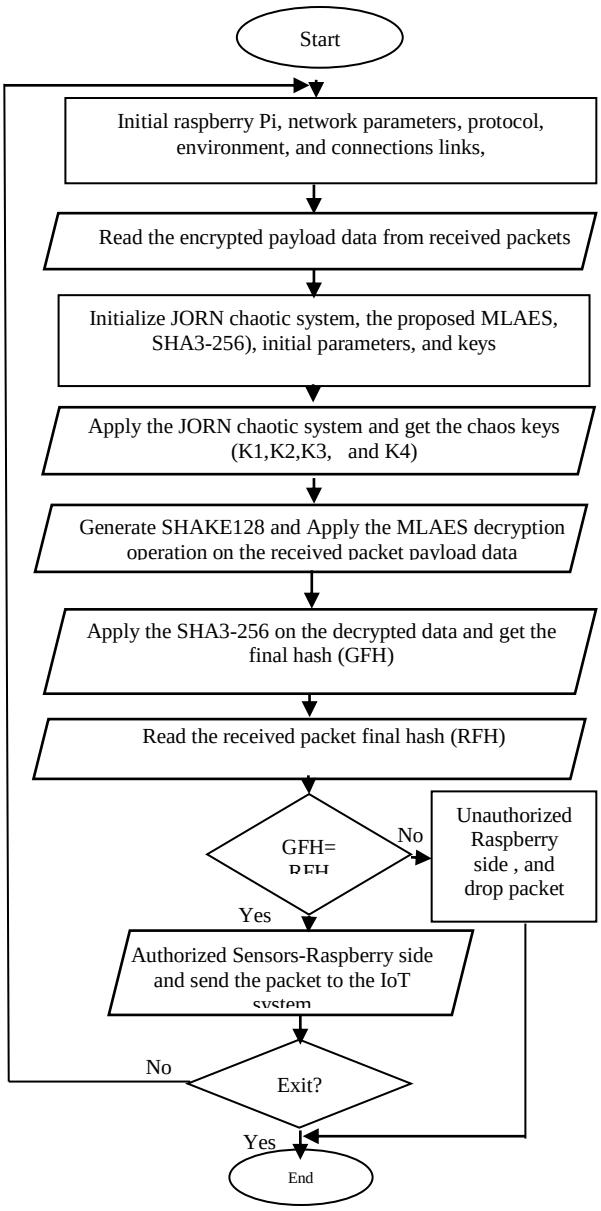


Figure 3 The Configuration block diagram for the IoT proposed System

In the IoT server (administrator computer used to generate the response after data sensing verified and analysis as shown in Figure 3), the received payload packet will be decrypted using MLAES algorithm. The chaos generation will use the same initial values and parameters used on the sender side. After completing the decryption operation, the decrypted data will



be used to generate hash (using SHA3-256 bit) to check the authenticity of the sender Raspberry pi and received packet side. All of the same operations performed on the Raspberry pi side for generation of the final hash will applied on the IoT server side, but the generated final hash will be compared with the received final hash (stored in the Raspberry packet) for each packet received by the IoT server. Figure 4 shows the proposed IoT encryption system using proposed Lightweight AES-Hashing (IoT server).

Figure 4 The Proposed IoT Security System Using Proposed Modified Lightweight AES-Hashing (IoT Server Side).

IV. RESULTS AND ANALYSIS

To test the proposed system by implementing it using 40 sensors (10 groups each containing 4 sensor types: thermal, pressure, magnetic, and camera sensors). Each group was connected to and managed by a Raspberry Pi B type device. The Raspberry device collected and aggregated the sensor data, then applied the proposed MLAES-Hash to obtain the encrypted data and create the final hash. Finally, the Raspberry Pi sent the encrypted sensor data with the final hash to the IoT server through the network.

On the IoT server side, the encrypted payload data were first read from the received packet. The decryption operation was applied using the proposed MLAES algorithm. The decrypted data were used to extract the hash value. The same operation was done to calculate the final hash, which was compared with the stored hash in the packet in order to accept or refuse the packet.

The time was measured for the proposed MLAES-Hash operations. Table 2 shows the average encryption time for the original [7] and modified Lightweight AES algorithms with different numbers of iteration rounds (4, 6, and 8). For 4 rounds of iteration, the proposed MLAES algorithm is faster (140.1 msec to encrypt 10 kB), while the original AES algorithm takes 161.2 msec to encrypt the same file size. This difference was shown in all results (as shown in Figure 5). Table 3 shows the MLAES-hash time (for the original and proposed Lightweight AES algorithms). It indicates that the proposed lightweight MLAES-Hash is consistently faster than the AES-Hash without modification or chaos. Table 4 presents the NIST statistical test results for the proposed MLAES. It shows that the proposed MLAES passes all of the randomness tests for the different numbers of rounds of iteration.

Table 2 Time measurement for the Proposed MLAES

Text size(byte)	Original AES (msec)	Modified AES [7] (msec)	MLAES (msec) (8 rounds)	MLAES (msec) (6 rounds)	MLAES (msec) (4 rounds)
10	2.908	2.506	2.23	2.12	1.98
25	2.910	2.561	2.36	2.21	2.10
70	3.123	2.889	2.56	2.42	2.31
100	4.232	7.6525	3.12	2.98	2.42
1000	33.122	27.510	26.51	25.10	21.89
2000	86.987	80.126	78.31	76.19	61.23
10000	161.2235	195.67	150.76	145.07	140.21
500000	1153.333	1135.673	1110.54	1106.85	1092.11
1000000	2985.369	2741.567	2730.70	2575.10	2430.99

Table 3 Time Measurement for the Proposed Lightweight IoT Security System

Text size (Kbyte)	Original AES-Hash Time (msec) 10 rounds	MLAES-Hash time (msec) 10 rounds	MLAES-Hash time (msec) 8 rounds	MLAES-Hash time (msec) 6 rounds	MLAES-Hash time (msec) 4 rounds
0.1	6.430	3.93	3.43	3.22	3.10
1	35.786	31.67	29.23	25.88	24.23
10	88.567	78.56	74.41	71.22	68.33
100	166.998	152.21	150.10	145.13	139.78
500	1164.700	1131.49	1128.11	1119.10	1097.98
1000	2996.111	2741.11	2736.08	2694.34	2487.39

Table 4 Randomness Testes Results for the proposed MLAES.

NIST statistical tests Results Name	Original AES (10 rounds)	MLAES (10) rounds	MLAES (8) rounds	MLAES (6) rounds)	MLAES (4) rounds
Frequency (Monobit) test	2.20	1.201	0.710	0.700	0.690
Runs test	7.95	4.746	4.101	4.001	3.98
Discrete Fourier transform	0.39	0.282	0.267	0.245	0.19
Block frequency	0.901	0.610	0.601	0.600	0.610
Longest runs test	0.132	0.272	0.240	0.222	0.198
Cumulative sums test	0.921	0.718	0.641	0.610	0.589
Serial test	4.62	1.41	1.22	0.999	0.81
Matrix rank test	1.34	0.901	0.851	0.731	0.656
Overlapping template test	0.65	0.451	0.301	0.290	0.190
Linear complexity test	1.65	1.102	0.939	0.930	0.880
Nonoverlapping template test	0.997	0.721	0.712	0.705	0.640
Random excursions variant test	0.894	0.522	0.511	0.50	0.480
Random excursions test	0.988	0.962	0.954	0.952	0.918

V. Conclusions

After implementation of the proposed system, was observed that even with these modifications to the original AES algorithm, the algorithm's security remains robust. The proposed modified AES algorithm is intact but also faster and more lightweight, making it more desirable for embedding in

IoT devices and sensors because of its reduced power consumption. Also, the results in table 2 show that the proposed modified AES faster than the algorithm in [7].

The proposed MLAES algorithm passes all NIST statistical tests. Therefore, brute-force attacks fail against the proposed MLAES algorithm.

The proposed MLAES is designed with less complex functions and was tested to calculate the CPU cycles during different rounds. We recorded a range of 6890 to 12200 cycles (for 4 to 10 rounds) in the case of a 128-bit encryption/decryption algorithm. This places the proposed MLAES algorithm towards the low end of the range of CPU cycles required by lightweight encryption algorithms.

Finally, the MLAES algorithm has a security advantage in that the results of NIST testing show that it was not broken by differential/linear attacks.

VI. References

[1] Abdullah Al- Mamun, Shawon S. M. Rahman, Tanvir Ahmed Shaon and Md Alam Hossain,"SECURITY ANALYSIS OF AES AND ENHANCING ITS SECURITY BY MODIFYING S-BOX WITH AN ADDITIONAL BYTE", International Journal of Computer Networks & Communications (IJCNC) Vol.9, No.2,pages 69-88, March 2017.

[2] Ali Abdulgader, Mahamod Ismail,Nasharuddin Zainal,Tarik Idbeaa" Enhancement of AES algorithm based on chaotic maps and shift operation for image encryption", Journal of Theoretical and Applied Information Technology, Vol.71 No.1, January 2015.

[3] Arnab Rahman Chowdhury, Junayed Mahmudy, Abu Raihan Mostofa Kamaly, Md. Abdul Hamid "MAES: Modified Advanced Encryption Standard for Resource Constraint Environments" ,IEEE,2018.

[4] C.G.Thorata and V.S.Inamdarb, "Implementation of new hybrid lightweight cryptosystem", Elsevier, Applied Computing and Informatics, 4 May 2018, https://doi.org/10.1016/j.aci.2018.05.001.

[5] Chittaranjan Pradhan,and Ajay Kumar Bisoi" CHAOTIC VARIATIONS OF AES ALGORITHM", International Journal of Chaos, Control, Modelling and Simulation (IJCCMS) Vol.2, No.2,pages 19-25, June 2013

[6] Deepika khambra, Poonam Dabas," Secure Data Transmission using AES in IoT", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 6, Issue 6, pages 283-289, June 2017.

[7] Haider K Hoomod, A M Radi," New Secure E-mail System Based on Bio-Chaos Key Generation and Modified AES Algorithm", IOP Conf. Series: Journal of Physics: Conf. Series 1003 (2018).

[8] KUN-LIN TSAI, YI-LI HUANG, FANG-YIE LEU2, ILSUN YOU, YU-LING HUANG1, AND CHENG-HAN TSAI, "AES-128 Based Secure Low Power Communication for LoRaWAN IoT Environments", IEEE ACCESS, 2018.

[9] Naif B. Abdulwahed," CHAOS-BASED ADVANCED ENCRYPTION STANDARD", Thesis, King Abdullah University of Science and Technology,2013.

[10] Pedro Sanchez Munoz, Nam Tran, Brandon Craig, Behnam Dezfouli, and Yuhong Liu," Analyzing the Resource Utilization of AES Encryption on IoT Devices", Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2018, Hawaii, USA.

[11] Ximeng Liu, Yang Yang, Kim-Kwang Raymond Choo, and Huaqun Wang, "Security and Privacy Challenges for Internet-of-Things and Fog Computing," Hindawi Publishing Corporation, Wireless Communications and Mobile Computing, Volume 2018, Article ID 9373961

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

أمان إنترنت الأشياء باستخدام نظام فوضوي (Chaos)جديد مع AES المخففة

جولان روكان نايف[١]
معهد المعلوماتية للدراسات العليا
الهيئة العراقية للحاسبات والمعلوماتية
العراق- بغداد
newjolan@gmail.com

غسان حميد عبد المجيد[٢]
دائرة البحث والتطوير
وزارة التعليم العالي والبحث العلمي
العراق -بغداد
ghassan@uob.edu.iq
ghassan@rdd.edu.iq
ghmajeed@gmail.com

علاء كاظم فرحان[٣]
قسم علوم الحاسوب
الجامعة التكنولوجية
العراق -بغداد
dralaa_cs@yahoo.com

المستخلص:

تزايدت خدمات وتطبيقات إنترنت الأشياء (IoT) خلال السنوات الأخيرة في العديد من مجالات الحياة ، حيث تحتاج إلى توفير معرف آمن لحماية بيانات الاستشعار(sensors) التي تمر بين أجهزة / أجهزة IoT والنظام الفرعي المدمج المتصل بالشبكات. تم اقتراح في هذه الورقة خوارزمية للمساعدة في أمن الاتصالات(IoT) والتي يمكن استخدامها في مختلف كيانات (IoT) المستخدمة في الاتصالات الصناعية مثل آلة الى لآلة (M2M) ، وشبكات الطاقة الذكية ، والمنزل الذكي أو المباني الذكية وغيرها من أجهزة الحوسبة.
اقترحت هذه الورقة نظامًا آمنًا باستخدام نظام فوضوي جديد رباعي الابعاد (4- Dimension) مقترن مع معيار التشفير المتقدم المخفف (AES). تم اختبار نظام الفوضى المقترح ذي الأربعة أبعاد بواسطة المعيار (Lyapunov) وتمريره لعدة فترات ابتدائية والحصول على نظام فوضى فائق (Lyapunov إيجابي). وايضا استخدامت مفاتيح الفوضى المولدة (المستخدمة في JORN) في AES المخففة وخوارزمية التجزئة الآمنة (SHA3-256) تظهر النتائج أن وقت حساب النظام المقترح انخفض (تسارع بنسبة ١٤٥٪ وأكثر). إن إخراج نظام تشفير AES المخفف المعدل لديه اختبارات إحصائية جيدة بالمقارنة مع AES الأصلي والذي يمكنه تجنب العديد من الهجمات.