Internet of Things Authentication Based on Chaos-Lightweight Bcrypt

التخويل في انترنيت الاشياء بالاعتماد على طريقة التشفير المحدثة والمخففة (**chaos-Blowfish**)

Jolan Rokan Naif
Informatics Institute For Postgraduate Studies

Prof.Dr.Ghassan H. Abdul-majeed
University of Baghdad-Baghdad, Iraq

Assist. Prof. Dr. Alaa K. Farhan
University of Technology

الملخص

التخويل في انترنت الاشياء او في المدن الذكية او البيت الذكي هي تحديد أشياء مثل (المستخدمين ، العقد ، الأجهزة ، والوصول إلى البيانات) بطريقة مخولة باستخدام معلومات خاصة تعتمد على أساليب وتقنيات التشفير لتوفير عملية آمنة قوية عبر إنترنت الأشياء. يجب أن تكون كل الاجهزة والمتحسسات مخولة لربط العقد مع باقي الاجهزة لنقل البيانات عبر الشبكة.

في هذا العمل تم اقتراح تقنية تخويل جديدة بين العقد وأجهزة الحوسبة والمتحسسات بالاعتماد على تقنية تشفير مقترحة وهي طريقة التشفير بال Bcrypt الخفيفة مع نظام الفوضى رباعي الأبعاد (4- D chaos system).

تحتوي الطريقة المقترحة على ثلاث مراحل وهي (SHA1-128bit، HMAC 256-bit ، modified chaos-Blowfish with 128-bit) حيث تم تعديل هذه المراحل الثلاث لتكون متوافقة مع أجهزة IOT بسرعة عالية. للحصول على حماية قوية للتجزئة ، النظام الفوضوي ذو الأبعاد الأربعة المهجن من نظام لورينز مع المعادلة الفوضوية اللوجيستية) يستخدم مع المراحل الثلاث أعلاه. تُظهر نتائج اختبار الخوارزمية المقترحة انها اجتازت اختبارات NIST مع دورات CPU في نطاقات lightweight.

Abstract

Abstract—Authentication on IOT is to identify things such as (users, nodes, devices, and data accessing) by authorized manner using private information based on cryptography methods and techniques to provide a robust secure operation over the IoT. Each devises and sensors need to be authorized to connect nodes and other devices for transferring data through the network.

This paper proposed new authentication technique between nodes, computing devices, and sensors based on proposed lightweight Bcrypt encryption with 4-D chaos system.

The proposed lightweight Bcrypt authentication contain three stages (SHA1-128bit, HMAC 256 -bit, and modified chaos-Blowfish with 128-bit). These three stages modified to be compatible to IOT devices with high speed. For getting a strong hashing security, the four dimension chaotic system (hybrid from Lorenz system with logistic chaotic maps) used to combine with above three stages. Testing results show proposed algorithm passing the NIST tests with CPU cycles in lightweight ranges

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

In the last years, the Internet of Things (IoT)becomes as a important portion of our works and daily life by using different IoT sensors and devices aggregation data based on wireless-technology and can manipulated and transferred within the infrastructure of Internet. But these sensors and devices must interoperate within many security and privacy issues, such as secure-communication, authentications and authorizations, and confidentiality of the information. [1]

Sensors-network services collect data from low end widely IoT-devices types over long-distances. After the data sensor collection, the data will be transmitted to the IoT servers, which manipulates it for useful information extraction. These collected data may contain a private or sensitive information, it should be protected from the unauthorized use to ensured integrity and confidentiality. So, we should produce a high speed sensors and devices protection. [2]

Authentication has been used to address special security issue for different online-services. It may use to protect against different malicious attacks as message faking, false-information broadcast, acquiring controls over the networks, and many other different attacks [3].The problem is that, at large number of devices and sensors is protecting(authenticating) of all devices that connected to networks.

Bcrypt algorithm password hashes led naturally to a new password scheme which we call Bcrypt referring to the Blowfish encryption algorithm. Bcrypt uses a 128-bit salt and encrypts a 192-256 bit values. It takes advantage of the expensive key setup in blowfish. [4]

Blowfish is a symmetric block cipher designed with variable length keys provides an efficient encryption rate. However, Blowfish cipher is weak keys vulnerable attacks; therefore, keys must select carefully (from 32 to 448bits to achieve a security high level). The private key must be large enough to make brute force keys search is not workable. The Table I shows the symmetric ciphers list are used to meets the requirements of security (authentication fields). [3]

To achieve high speed authentication and protection with reduce the cost of computational protection algorithms, Jeyamala [5] propose integrated the S-box with Henon map into the selected protection(Blowfish and Data Encryption Standard (DES)) algorithm.

Alabaichi [6] proposed a new encryption algorithm depend on Blowfish Algorithm (BA) by adapts functions (F- functions) as Cylindrical Coordinate System (CCS). The F-functions are knowns as “Cylindrical Coordinate System with Dynamic Permutation Box” (CCSDPB).[6]

Ariel Roy et al [7] present a new BA modification that capitalized on the strength’s algorithm but support 128bits input blocks based on dynamical selection-encryption algorithm and cipher function execution reduction during random selected rounds.

1

Theda Flare G and et al[8] proposed a modification to the BA that deals with 128bit block-size and 128bit keys to achieves the encryption standard minimum requirements. The modification to save memory used two S-boxes only in each round but kept the original structure for easy migration. To prevent symmetry a derivation was added. The performance of algorithm was evaluated based on avalanche and time.

TABLE I: The symmetric ciphers list are used to meets the requirements of security (authentication fields).[3]

Cryptographic ciphers	Security Requirements support	Attack Mitigation
Blowfish	Authentication/Availability	Differential related-key attacks/Brute-force attack
PBAS	Authentication/Availability/ Confidentiality	DoS attack, Impersonation attack
Camellia	Authentication/Availability /Privacy Preservation	Impersonation attack/DoS/Sybil attacks
CAST	Authentication/Availability/ Confidentiality	Sybil/Impersonation attack/routing attacks

The Primary target of this article is to give a proposed for IoT authentication based on the modified lightweight Blowfish and hybrid 4-D chaotic maps in order to achieving a good speed protection authentication for all devices connected to the internal network.

Taxonomy and Comparison of Authentication Protocols for the IoT

In this section, we explain in details, IoT authentication protocol applied in or developed for the IoT devices. the IoT authentication protocols processes are:[9]

(1) network model definition (e.g., M2 M, Io S, IoV, and IoE).

(2) authentication model definition (e.g., mutual authentications, un-traceability, anonymity, and perfect forward secrecy).

(3) attack model (e.g., replay attack, privileged-insider attack, offline password guessing attack, and sensor node capture attack).

(4) countermeasures selection (e.g., cryptographic methods, access polynomials, biometrics, Smartcard, and Chaotic-Maps).

(5) suggestion of the protocol main phases (e.g., initials setup; registration processes).

(6) Security analyses using formal security verification

(7) Performance evaluation (e.g., storage cost, computation time, complexity, communication-overhead, and error-rates).

II. THE PROPOSED SYSTEM

There are three stages must cleared in this proposed IoT authentication system:

A. THE PROPOSED MODIFIED LIGHTWEIGHT BLOWFISH

The first stage of the proposed system is Blowfish algorithm (BA). We proposed a modification to BA to reduce the complexity computation, time of execution, and to save memory. In [8], they proposed a modification to the BA by enhancement the F-Function using P-array and logical shifting and operation. We developed BA based on the [8]. The modification adapted the BA was combines with 4D chaotic system (hybrid the Logistic map and Lorenz map) used to generate four chaos keys(KS1, KS2, KS3, and KS4). These chaos keys were used to increase the randomness of the encrypted results and give more strengths to the BA to avoid more attacks. Some of conditions in [8] still like the key expansion converted the 128 bit key length into several subkey arrays, and the total iterations number to generate all required subkeys decreased to 266. Also, the modified Blowfish reduced the size from the previous 4168 bytes to 2128 bytes. The all keys may be generated and stored before any operations. In addition, the P array consists of 20 (32-bit subkeys) as (P1, P2…P20). BA uses 4 S-Boxes consists of 256 individuals include 32-bits each (S1 - 0…255, S2 - 0…255) as well the modified BA decreases the S-boxes to two(for satisfying lightweight), below the steps of the modified Light weight:

1. initialize the P-array, S-boxes, main-key, chaos keys (KS1,KS2, KS3, and KS4) and other parameters and keys.
2. apply the XORed between P1 with first input (the first 32 bits), P2 with input (the second 32-bits) continuously until all key bits up to P20. Repeat for all P-array against the key bits. XORed the results with KS3.
3. Encrypt all-zero strings using the subkeys from steps (1 and 2).
4. substituted P1 and P2 with results of step 3 after XORed with Chaos key (KS4+KS3). Apply the same function for other P up to P20.
5. encrypt the step 3 output using the revised subkeys.
6. replaced P3 and P4 with result of step 5 after XORed with Chaos key (KS2+KS1).

Also, all P array entries will be replaced during cycles rounds in the BA encryption operations. The two S-boxes also changing during the encryption processing. The general block diagram of the proposed modified (with 8 iterations) is as illustrated in Fig.1.

The sequence of the operation s started from the inputs block, the input block with divided into two blocks of size 64bits named (IL0 and IR0). The IL 64-bit block is XORed with the P-array (P1, P11) two 32-bit entries and get IL1. The P1 and P2 with results of step 3 after XORed with Chaos key(KS4+KS3). Apply the same function for other P up to P20. The IL1 (two 32bit data) passed to the modified F- function. The modified F-function have two S-boxes with 32-bit inputs, and the encryption equation (1). The modified F-function results will be XORed with IR0(32-bit data from input) and KS3, and store results in IR0.The IL1 and IR1 will swaps for preparing to

new cycle. In After finished the second cycle, the P3 and P4 with result of step 5 after XORed with Chaos key(KS2+KS1). We repeat the cycles up to the eighth round or more. In this proposed system, the number of cycle iteration depend on the KS1 (last significant number of KS1 will determine the number of iteration rounds but must more than 3 iterative rounds). After the n rounds, apply last swapping operation to exchange ILn and IRn reversing. The REn is XORed to (P9, P19) and LEn is XORed to (P10, P20). Finally, the ciphertext resulted from combine the LEn and REn. Inverse these steps will result a decrypted text.

The hybrid chaotic system used in this proposed system composed from the Lorenz chaos map (equation 3)[10] and Logistic chaos map (equation 4)[11] as shown in Fig.2. The Logistic chaos output will used in two manner (as KS1, and as initial to the Lorenz chaotic system). The last significant of each number resulted from the Logistic chaotic system will added to the initial periods of the Lorenz initial conditions

$$f(k_{n+1}) = r \cdot k_n(1 - k_n) \quad (1)$$

$$x_{n+1} = a \cdot (y - x_n) \quad (2)$$

$$y_{n+1} = c \cdot x - x \cdot z - y \quad (3)$$

$$z_{n+1} = x \cdot y - b \cdot z \quad (4)$$

Where a, b, c, r is the chaos parameters. x0,y0,z0, and k0 is the initial conditions for chaos map.

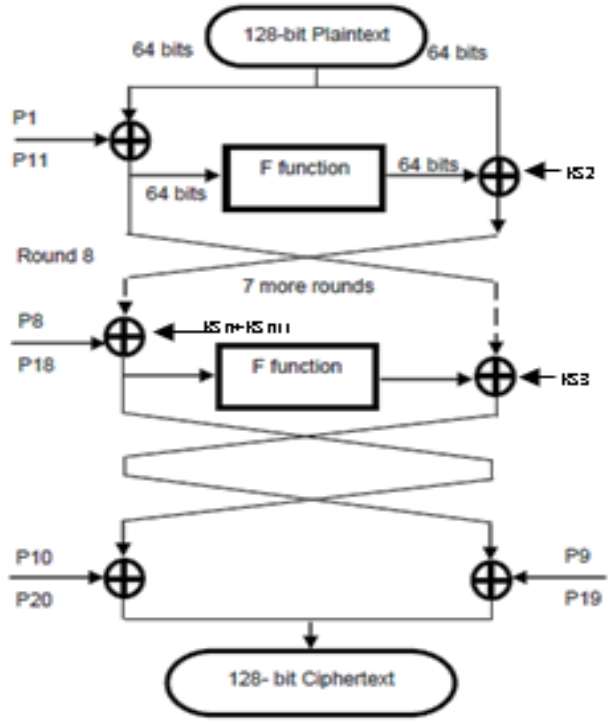

Fig.1. The Proposed Modification to Lightweight Blowfish

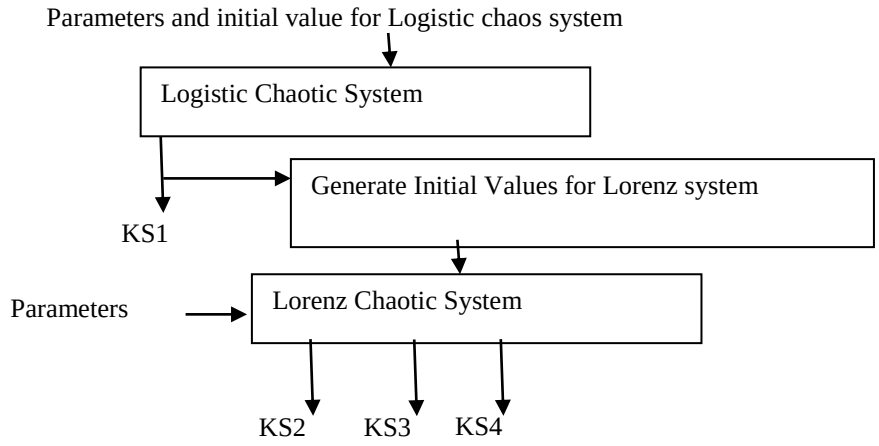

Fig.2. The hybrid chaotic system used in the proposed system.

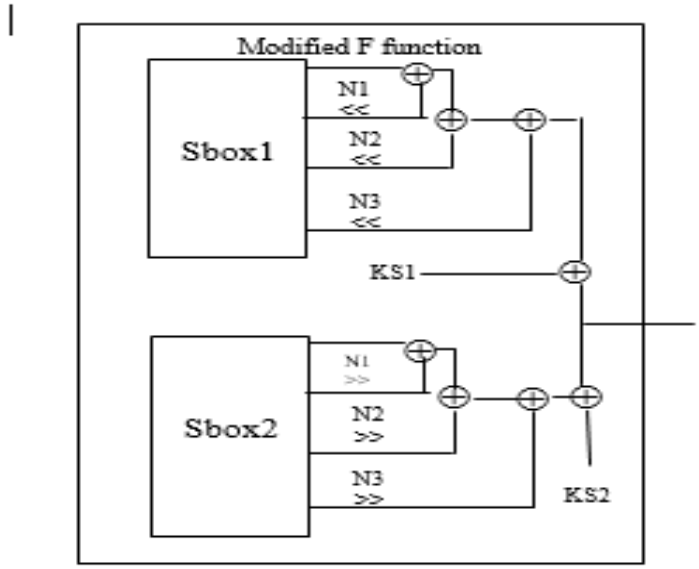

Fig.3. The proposed Modified F function

B. *THE PROPOSED MODIFIED F-FUNCTION*

The one of the complex operations in BA is the F-function. It affect in most BA security. In general case, the F-function accepts a stream data of 64-bitsize, this block of data will splits to 8-bits (as a, b,… up to the last 8 bits). Each 32-bits are utilized with S-box(there are two S-boxes proposed to used in this modified F-function). The S-Boxes outputs utilized by equations (1 and 2) and then concatenated to obtain the 64-bit output as shown :

F (IL0(first 32bit)) = (((S1(a) ⊕ S1(b) << N1 mod 2^{32}) □⊕ S1(c) << N2) + S1(d<< N3) mod 2^{32}) ⊕ KS1 …(5)

F (IL0(second 32bit))= (((S2(e) ⊕S2(f) << N4 mod 2^{32}) ⊕ S2(g) >> N5) ⊕ S2(a<< N6) mod 2^{32}) ⊕ KS2 ... (6)

Fig.3 illustrated the block diagram of the proposed modified F-function in the proposed modified blowfish

As shown in Fig.3, there are simple rotation during runtime to the S-boxes (by N1,N2 and N3 with max 5 rotations process) are derived from the last significant number of (KS1, KS2, and KS3)at runtime.

C.*THE PROPOSED IOT AUTHENTICATION SYSTEM*

The proposed authentication IoT system contains two stages, the first stage is collecting the data from the sensors. We used 40 sensors with four types of sensors for testing the proposed system. Each. The 4 sensors data collect and aggregates using Raspberry Pi type B. The second stage is to apply the combine of the lightweight Bcrypt hashing techniques like modified chaos-BA. We proposed to used combine of (proposed modified chaos-BA, SHA-1 128bits, MAC, and HMAC) to gotten the final hash (256 bits) for each group of sensors (10 group of 4 sensors types). The lightweight satisfied by reducing the BA F-function to two S-boxes and decreased the iterations rounds (dynamic iteration numbers).

The dynamic iteration numbers varying depending on the chaos key will getting more security to BA and protect against the key -round guess. The input sensors encrypted by using the modified chaos-BA, the encrypted data send to SHA-1(to generate the salt that concatenated with the Blowfish output), MAC, and at last HMAC 256 bits. Below the flowchart of the proposed system. The Fig.4 illustrated the proposed IoT authentication System using proposed Lightweight Bcrypt hashing (Sensors side). It is shows that the steps of the IoT system operations, all parameters and initial values will determines between the two sides (sender sensors-Raspberry side, and IoT server side). While the Fig.5 shows the Proposed IoT authentication System using proposed Lightweight Bcrypt hashing (IoT server side). In the IoT server side, the final hash will be generated by apply the proposed lightweight Bcrypt on the each received packet payload data, and comparing the results with final hash stored in the received packet. The packet accepted or refuse(dropped) depending on the comparing results.

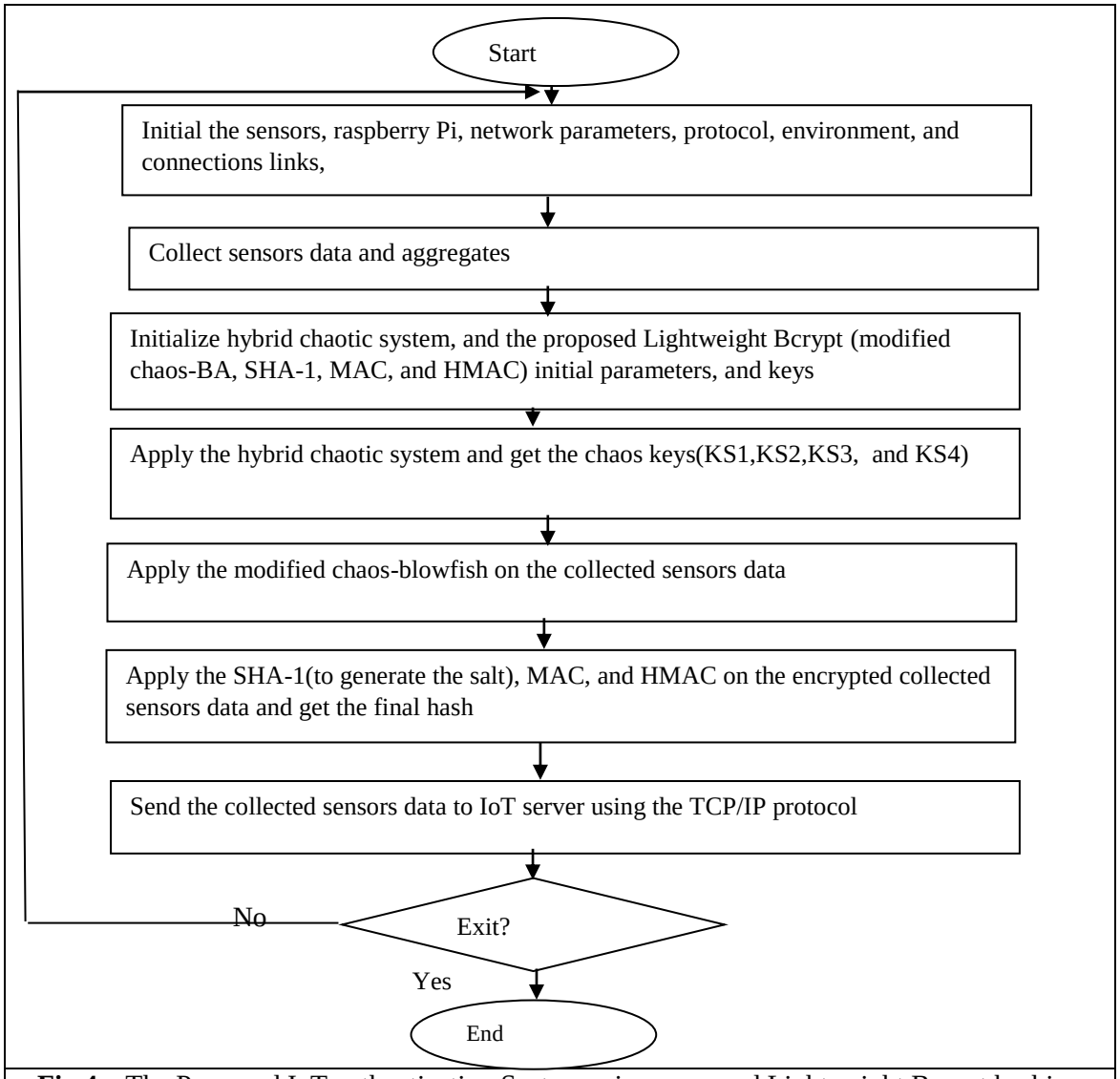

Fig.4. The Proposed IoT authentication System using proposed Lightweight Bcrypt hashing (Sensors side).

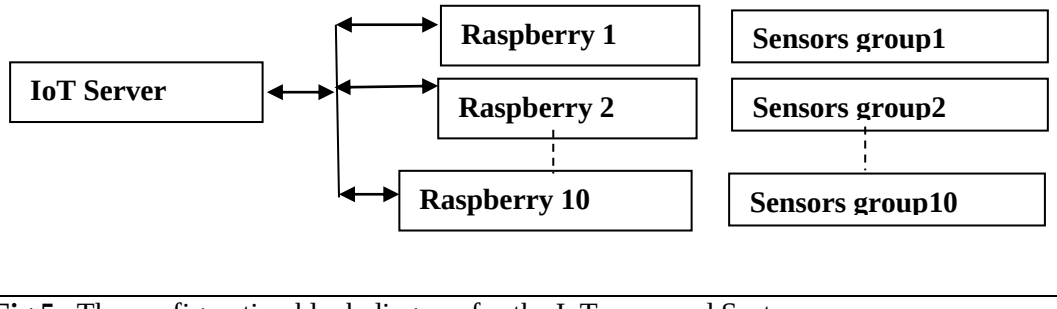

Fig.5. The configuration block diagram for the IoT proposed System

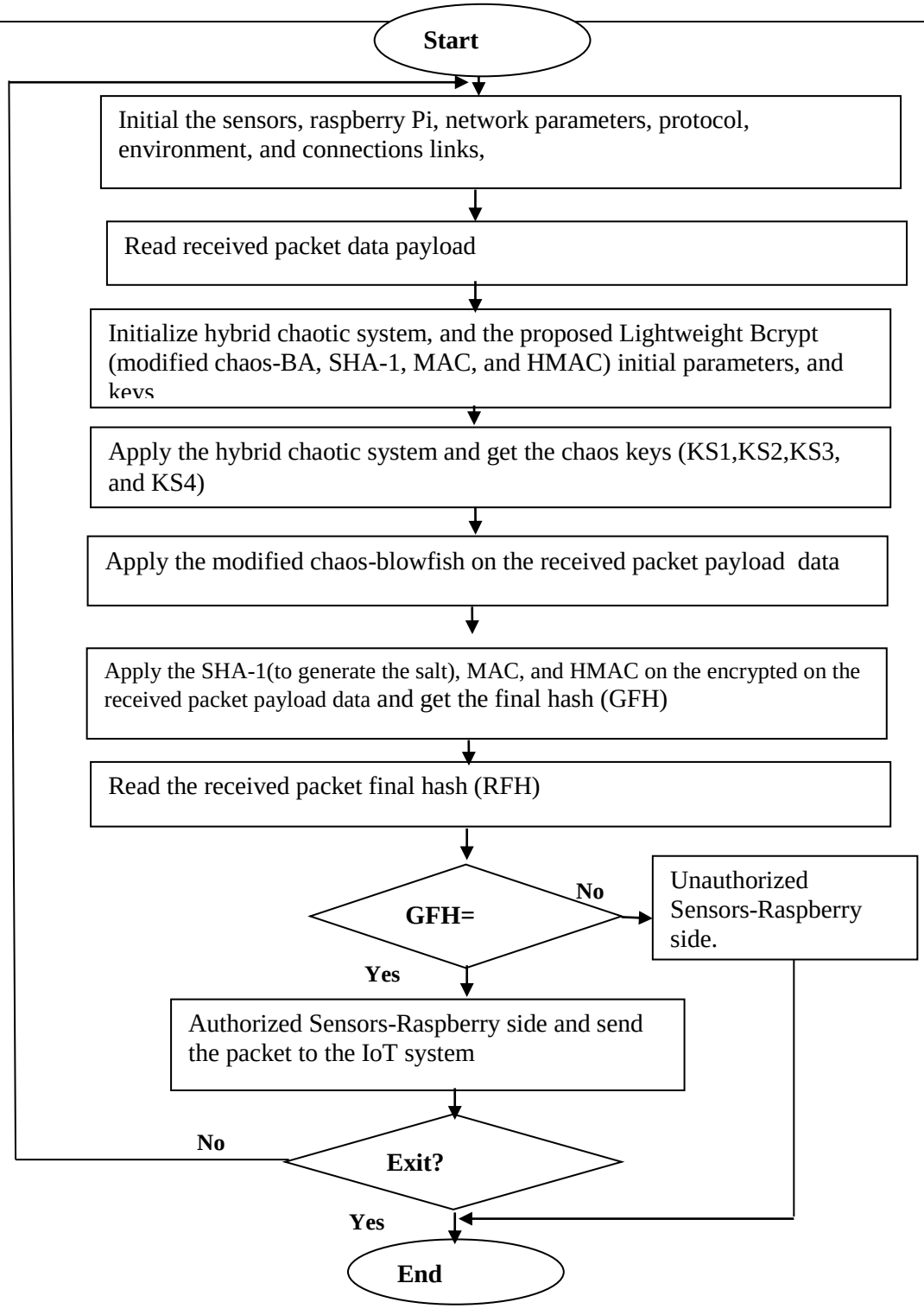

Fig.6 The proposed IoT authentication system using proposed Lightweight Bcrypt hashing (IoT server side)

In the IoT server (as shown in Fig.5), the checking operation for authentication of the sender raspberry through the network. Same all operations in Raspberry -sensors sides for generation final hash will applied in IoT server side, but the generated final hashed will compared with the received final hashed (from the sensors-raspberry packet) for each packet received by IoT server. Fig.6 shown the proposed IoT authentication system using proposed Lightweight Bcrypt hashing (IoT server).

III. RESULTS AND ANALYSIS

The proposed system tested using 40 sensors (10 groups and each group contain 4 sensors types: thermal, pressure, magnetic, and camera sensors). Each group connected and management by raspberry Pi B type. The raspberry device will collect and aggregate the sensors data, then applied the proposed Lightweight Bcrypt system to create the final hash. Finally, the Raspberry will send the sensors data with final hash to the IoT server. In IoT server side, the same operation will be done to calculate the final hash and comparing with the sending hash. We measured the time for the proposed Lightweight Bcrypt and the proposed modified BA. Table II shows the average encryption time for original and modified chaos-Blowfish with different iteration rounds (8,6,and 4) for the modified chaos-BA. The 4-rounds modified chaos-BA time is faster the other (is 43.0008 msec for encrypt 10 kB), while original BA take 51.2235 msec to encrypt the same file size.

Table III shows the Bcrypt hash time (for original and proposed Lightweight). It stalls the proposed lightweight Bcrypt faster than the Bcrypt without modification and chaos.

Table IV illustrates the NIST statistical tests for the proposed modified chaos-Blowfish. It is showing the proposes modification chaos-BA for different round passing the randomness tests. Table V shows the sample of proposed lightweight Bcrypt Hashing

TABLE II Time measurement for the proposed modified chaos-blowfish testing.

Text size(byte)	**Original Blowfish time(msec)**	**Modified chaos Blowfish time(msec) (8 rounds)**	**Modified chaos Blowfish time(msec) (6 rounds)**	**Modified chaos lightweight Blowfish time(msec) (4 rounds)**
10	1.0123	1.0144	1.0121	1.0101
25	3.0124	3.0144	3.0121	3.0101
70	4.0143	4.0158	4.0135	4.0120
100	5.1070	5.1011	5.0956	4.0920
1000	15.0944	15.0828	15.0310	11.8012
2000	20.3042	20.2267	20.1125	16.9315
10000	51.2235	51.2121	50.0854	43.0008
500000	123.5675	123.4679	121.0983	118.0219
1000000	205.9887	205.6545	185.8977	173.0034

TABLE III: Time Measurement for the Proposed Lightweight Bcrypt(LWBcrypt) Testing.

Text size(byte)	Bcrypt without modification time(msec)	LWBcrypt time(msec) (8 rounds)	LWBcrypt time (msec) (6 rounds)	LWBcrypt time(msec) (4 rounds)
10	**3.5321**	**3.5243**	**3.2387**	**2.7681**
25	**5.2542**	**5.2433**	**5.1097**	**4.0845**
70	**7.2981**	**7.21121**	**7.1005**	**6.4261**
100	**10.3970**	**10.3432**	**10.0087**	**9.3334**
1000	**23.4719**	**23.4672**	**23.1988**	**18.720**
2000	**29.3906**	**29.1897**	**28.1231**	**23.450**
10000	**67.4082**	**67.0001**	**66.5673**	**61.740**
500000	**156.2678**	**156.0899**	**151.332**	**139.981**
1000000	**276.8872**	**276.4567**	**270.098**	**256.321**

TABLE IV: Randomness tests Results for Modified chaos Blowfish (MCBA) using different rounds**.**

NIST statistical tests Results Name	Original Blowfish	MCBA (8 rounds)	MCBA (6 rounds)	MCBA (4 rounds)
Frequency (Monobit) test	0.7597	0.8832	0.8612	0.7094
Runs test	0.7234	0.6895	0.6563	0.6012
Discrete Fourier transform	0.1046	0.1037	0.1031	0.1001
Block frequency	0.8398	0.7952	0.7893	0.7209
Longest runs test	0.0332	0.0361	0.0351	0.0342
Cumulative sums test	0.6721	0.7555	0.7456	0.6108
Serial test	0.7562	0.8701	0.8530	0.8270
Matrix rank test	0.5214	0.5345	0.5441	0.5342
Overlapping template test	0.9123	0.9089	0.9105	0.9000
Linear complexity test	0.9583	0.9631	0.9547	0.9402
Nonoverlapping template test	0.6728	0.6698	0.6734	0.6666
Random excursions variant test	0.6945	0.6867	0.6549	0.6412
Random excursions test	0.7189	0.8324	0.8276	0.8109

IV. CONCLUSIONS

The proposed system was designed to improve the security of the sensor/device data transferrin IoT network by proposed authorization algorithm under the Lightweight issues. After implementation of the proposed system, we see that even still these modifications are carry out on the original security algorithms the collection of the modified security algorithms (modified chaos-BA, SHA-1, MAC, and HMAC) called the proposed Lightweight Bcrypt. The original algorithm security BA remains robust (for some cases) and the proposed modified (chaos-BA) algorithms also more intact and secure due the randomness of the chaos keys and reconfiguration of F-functions box, but faster than and lightweight to be more satisfy to embedded in to IoT devices and

sensors for power consuming.From the tests results, the proposed modified chaos-BA algorithm passed NIST statistical tests for different data size encryption. Therefore, fails on the proposed algorithm can avoid the Bruteforce Attack.

To get the min Lightweight requirement, the proposed Lightweight Bcrypt algorithm (like proposed modified chaos-BA algorithm) was designed with less complexity function (as show in run time computation), while the CPU cycles during different rounds averaging between 8904 to 11655 cycles for different (encryption rounds and data size) .Using SHA1, MAC, and HMAC help authentication operation in order to avoiding many attacks.

V. REFERENCES

[1] 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

[2] Taehwan Park, Hwajeong Seo, Sokjoon Lee, and Howon Kim, "Secure Data Encryption for Cloud-Based Human Care Services," Hindawi Publishing Corporation, Journal of Sensors, Volume 2018, Article ID 6492592

[3]Ahmer Khan Jadoon, Licheng Wang, Tong Li ,and Muhammad Azam Zia, "Lightweight Cryptographic Techniques for Automotive Cybersecurity," Hindawi Publishing Corporation, Wireless Communications and Mobile Computing, Volume 2018, Article ID 1640167

[4] Niels Provos and David Mazieres," A Future Adaptable Password Scheme", The OpenBSD Project

[5] Jeyamala Chandrasekaran and S. J. Thiruvengadam "Ensemble of Chaotic and Naive Approaches for Performance Enhancement in Video Encryption," Hindawi Publishing Corporation, The Scientific World Journal, Volume 2015, Article ID 458272.

[6] Ashwak Mahmood Alabaichi," A Dynamic 3D S-Box based on Cylindrical Coordinate System for Blowfish Algorithm", Indian Journal of Science and Technology", Vol 8(30), November 2015.

[7] Ariel Roy L. Reyes, Enrique D. Festijo and Ruji P. Medina," Securing One Time Password (OTP) for MultiFactor Out-of-Band Authentication through a 128-bit Blowfish Algorithm", International Journal of Communication Networks and Information Security (IJCNIS) Vol. 10, No. 1, April 2018.

[8] Theda Flare G. Quilala, Ariel M. Sison, Ruji P. Medina," Modified Blowfish Algorithm", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 12, No. 1, October 2018, pp. 38~45. V

[9] Mohamed Amine Ferrag,1,2 Leandros A. Maglaras,3 Helge Janicke,3 Jianmin Jiang,4 and Lei Shu5," Authentication Protocols for Internet of Things: A Comprehensive Survey", Hindawi Publishing Corporation, Security and Communication Networks, Volume 2017, Article ID 6562953.

[10] A. A. Elsadany , A. M. A. El-Sayed," On a complex Logistic Difference Equation", International Journal of Modern Mathematical Sciences, 2012, 4(1): 37-47

[11] Ihsan PEHL˙IVAN, Yılmaz UYARO˘GLU," A new chaotic attractor from general Lorenz system family and experimental implementation", Turk J Elec Eng & Comp Sci, Vol.18, No.2, 2010.

TABLE V: Example of The proposed Lightweight Bcrypt Hashing

Sensor	Sensor Data	BA	MCBA(8R)	MCBA(6R)	MCBA(4R)	SHA	HMAC	Final Hash
Camera Sensor	0100110101101011 01	0b150495318b286373758acbd59ff1bf	9edef2a5aac1f15b2880f2cda504e819	fc4ea75eac522c9725db754d8ed41e9d	a5c8d94e82727d3dd06018a3e4a36863	53706596a8244b659d11d7711f76eb3f4b318aa649064469af1459d9f2fbf99c	a1394d3dbd844d6dbcb93f74091186786011 3e4ca71f429d8bb11315fff1445a	$2y$10$yS1YMcF0vRZKPMScm/gKAOwX.RzNBRGvqK5s11ad23EJ4r6BxxPeM
Magnetic Sensor	390.7	48bf942b4405c4f4a758a5371810b646	13c150fa8d2800075d37cf32cf4f0ff5	882f0f29ffd64502f4ca157cd05ad5b6	63d2412048e28e06773eb051af52063a	1f4ba555f6014c33884a4d17af31605a60d4ff06be714b96aab01d244e215f81	6235a4c0b07c47f5a15be3412e41ab8566886aca14004a3387435c868665d0a8	$2y$10$2rS3byxu6kSZ9RQCAVvZBEcxEatF56NyQLnVeWW3nKSJCE4dIJ51
Pressure Sensor	35.67	23db7d54ae0f50c56fee3a38c8758ae3	3ff622f41c9577a79f95f9d468464194	f35fbfda4ed4ab85b3f1271b742fa076	99ca75aeee35ecca441b418846f4f103	fb6e5a83044943bca74470b526678fbf25c2d05db0924c4eb20d6d488d0169e1	7316f68134d044ea83a0508a04c9e8cfd453432f32c84bdd955628de0fad91bf	$2y$10$Rc6aG34kWjJ3ao4FG1jlFdV1C6LRTVzQMNcwLFVXA5ny3pw4SV45B
Thermal Sensor	1000011 0101010 1011110 0001110 1101101 1110010 1001011 1101011 0110111 0011100 1110100 1111100 011	b112051fca2c8d6e7fe72072a1038316	1ab46548b849a08edf497b6cfe2e6b62	71c1479ebc993e8248f717f9ca40bf63	18af9de5ac82f7bdc15eac725468a9f5	699a2b5cdeef4438a81819bd9538d3073b9b42c7870746d992a2277b08a85733	01d19d25586c41a582a7a37f893a6b8964e7561ed2dc4bfe84bbcda618bead12	$2y$10$olPac5GF2XHy1hD67Mxyb.hwqMstq4erY12ameaYN61erO.P7R12O

TABLE V: Example of The proposed Lightweight Bcrypt Hashing