Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 8, No. 4, December 2020, pp.2333-2345

A new intelligent hybrid encryption algorithm for IoT data based on modified PRESENT-Speck and novel 5D chaotic system

Haider K. Hoomod[1], Jolan Rokan Naif[2], Israa S. Ahmed[2]
[1] Computer Science Department, College of Education, Mustanisiryah University
[2] Informatic Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatic

ABSTRACT

Modern application based on IoT sensors/devices are growth in several fields. In several cases, the sensing data needs to be secure in transmission to control / administrator side. In this paper, the proposed secure Internet of Things data sensing and proposed algorithms will be explained, based on the main overarching novel 5-D Hyper chaotic system and new encryption mechanisms (contains hybrid encryption and two modified encryption algorithms) controlled by Fuzzy rules. The encryption mechanism combined by using the structure of PRESENT and SPECK algorithm with novel 5-D chaotic system. Also, for encryption will use the modified mechanisms of Round steps in PRESENT algorithm by SPEECK which were adopted on an IoT sensing data transferring.

This proposed system provides a high level of security for any sensitive information that may be generated from sensors that may be installed in an important location to protect buildings and offices from theft by making certain modifications to the algorithms necessary to maintain the safety and security of the information, etc., which must be protected from Attacks. This system is designed to be effective in providing security features for data contents that include confidentiality, authentication and non-repudiation, and is compatible with all types of remote sensing data and sensors to send the final notification to the final administrator view.

The proposed system is designed to provide users with high flexibility and ease in managing change operations, speeding up encryption operations and intruding the contents of message packets (types and forms of different sensor data) at the point of origin and decrypting and checking packet integrity messages upon receipt. These features make users of this system more confident with each other. The proposed encryption mechanism and novel chaotic system passed different testes. The generated chaos key space at least 2^{2560}probable different combinations of the secret keys to break the system used brute force attack.

Keywords: Hybrid Encryption, IoT security, chaotic system, PRESENT, PRESENT-Speck

Corresponding Author:
Haider Kadhim Hoomod,
Computer Science Department,
College of Education, Mustansiriyah University, Baghdad, Iraq
Email:drhjnew@gmail.com

1. Introduction

The growing use of the Internet of Things has made the amount of information transmitted through the Internet very large, and with the increase of this information and increased sensitivity. The fear of the use of this information by unauthorized people has emerged [1]. IoT displays challenges in security relating that are determined by the IERC 2010 Strategic Research and Innovation Roadmap (SRIR). While some detailing is helpful, there are additional parts that needing to address by the research-society. Although there are numerous specific challenges of privacy, trust, and security in the IoT, they share many incidental non-functional requirements. [2]

The Internet of Things (IoT) offers real opportunities for wearable devices, smart homes, software, and information sharing via the Internet. Considering that the information that may be shared is private, maintaining the security of this information is a fundamental requirement in the Internet of Things. Addressing such obstacles is one of the continuing challenges that the Internet of Things has to face.

Over the last few years, various lightweight cryptography algorithms alternatives have been proposed and developed such as stream ciphers, block ciphers, hash functions, message authentication codes that aim to use in devices that can't provide most presenting codes and do not have enough resources (memory space, power

consumption and implementation time, connectivity hardware and software). Such as Radio frequency identifications (RFID) tags, sensors in wireless sensors networks (WSN) and IoT application, and smart cards, etc., rather than famous and large encryption algorithms such as AES and DES cryptography [3].
One of the symmetric-key cryptography classes is the ARX-based ciphers algorithms that meager function contain three operations (additions, bit-rotations, and Xor). ARX-based ciphers like: the block encryption (SIMON/SPECK, LEA, Chaskey, and others), the stream encryption like (Salsa 20, chacha, and SHA-3). In evaluation ARX cipher security, the cryptanalysis (like differential cryptanalysis) is taking the most significant attacks of an ARX cipher security [4].
Many researches in lightweight cryptosystem (block encryption) designed to be in useful utilize S-boxes to get nonlinearity operation; as eminent, the main features of (Simon/Speck) are independence on Sboxes. The important of S-box is that, when used in substitutions permutation network (SPN) as affect part for relatively security arguments, against different standard attacks. But for constrained platforms efficiency, S-box designs are not optimal. Bit permutations technique used in Lightweight block ciphers as part of a SPN. In some optimal manner, the bit permutations role is to dispersal bits around, and therefore allow SPN style security-arguments [5].
For the resources constrained devices security, the security mechanisms can be composed to be more lightly techniques. Lightweight block encryption can be main part of the basis of Lightweight security mechanism and methods, but there are a lot of Lightweight block ciphers researches are proposed recently such as PRESENT [3], SIMON/SPECK [6,7], SPARX [8], GIFT [9], and CHAM [10]. ARX-based blocks cipher can skillfully work but with some impact that can decreases the achievement of IoT devices/sensor data. The different ARX-based blocks cipher can deal with various key size or data block size or both. In SPECK key features, it can work with a set of seven different key block sizes (64, 72, 96, 128, 144, 192, and 256) and five data block size (32,48, 64, 96, and 128). So, the SPECK has different instances (ten cases) due to combinations of block and key sizes with a different rounds number.[4]
Kanso et al were focused on the use of the retail function based on a proposed messy map, to provide a high degree of security for messages as well as the key generated from this map, and the possibility of confusion and deployment can be used to resist the expected attacks on the contents of the hash. All these components were used to choose the appropriate way to maintain data integrity and authentication [11].
In recent years, very few light block blades have been shown with many design strategies. Skipjack's algorithm has a 64bit blocks length with an 80-bit key on an unbalanced Feistel network. New lightweight algorithms used idea of the new variants in using single S-box squared repeatedly [12]. A lightweight block cipher called PRESENT was used in many IoT applications. PRESENT design is very device-efficient, as it uses a few deployment layers without complex algebraic computation. The table (1) shown below illustrates the comparison between lightweight encryption algorithms based on block cipher cryptography.

2. The block cipher present encryption algorithm

PRESENT algorithm was lightweight block ciphers with SP network-based cipher which deals with 31 rounds. There are two types of the PRESENT algorithm: PRESENT80 and PRESENT128 due to the key size (80 and 128 bits). PRESENT can avoid many attacks like (differential_cryptanalysis, linear_cryptanalysis, and brute force especially in PRESENT128. PRESENT also passed all NIST tests. [13, 14]
The key concept of PRESENT design is to allow lightweight and fast implementations was suggested by Bogdanov et al at CHES 2007. The block length in PRESENT is 64 bits, and the key length is 80 or 128 bits. It uses SP-network, the replacement layer consists in parallel of 16 4 x 4 S-boxes. The permutation layer that's applied in PRESENT is a regular bit-permutation, which helps to generate a simple security analysis and gets stellar performance in hardware/software operations, offering flexibility for various applications. Figure 1 shows the block diagram of PRESENT [14,15]. Through the 31 rounds comprises three steps:

- **Add Round key**: a simple round intermediate-state XOR bitwise.
- **Sub Column**: Parallel application of S-boxes in the same column to 4 bits. Table 2 provides the action of this S-box in hexadecimal notation.
- **P Layer:** The current bit permutation is given in the following table, Bit i of state is moved to bit position P(i) [16]. Table 3 illustrates the permutation layer.

Table 1. Lightweight block ciphers Comparison [17]

Ref.	Algorithm	Key Size (bits) Block Size (bits) Rounds			Structure	Performance				Merits	Attacks/Analysis
						Tech. (µM)	Power (µW)	Area (GE)	Throughput At 100Khz (Kbps)		
1	AES	128	128	10	SPN	0.13	2.48	2400	56.64	Supports larger key sizes, faster in both hardware and software.	Related key attack, Boomerang, Biclique cryptanalysis
2	PRESENT	80	64	32	SPN	0.18	1.54	1030	12.4	Ultra Lightweight cipher, Energy efficient.	Integral, Bottleneck attacks, truncated differential cryptanalysis, Side-channel attacks
		128				0.18	2.00	1339	12.12		
3	RECTANGLE	128	64	26	SPN	0.13	1.78	1787	246	Fast implementations using bit slice techniques	slide attack, related-key cryptanalysis, statistical Saturation Attack
4	HIGHT	128	64	32	FN	0.25	5.48	3048	188.20	Ultra-lightweight, provides high security, good for RFID tagging.	Impossible differential attack on 26th round, Biclique cryptanalysis
5	CLEFIA	128	128	18	FN	0.13	2.48	2488	39	Has fast encryption and decryption, lesser rounds, energy efficient	Key Recovery Attack on 10th round, Saturation Cryptanalysis
6	CAMELLIA	128	128	18, 24	SPN	-	1.54	6511	290.1	Resistance to brute force attack on keys, security levels comparable to AES.	Cache timing attacks , Impossible differential attack
7	TWINE	80, 128	64	36	FN	0.09	1.30	1866	178	Good for small hardware, efficient software performance	Meet-in-the-middle attacks, Saturation Attack
8	SIMON	128	128	64	SPN	0.13	1.32	1317	22.9	Supports several key sizes, performs well in Hardware	Differential fault attacks, Attacks on reduced versions
9	SPECK	128	128	32	SPN	0.13	1.40	1396	12.1	Performs better in software	Key Recovery, Boomerang attack

Table 2 . The substitution layer (Sbox)[13,14,15]

.

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*

Table 3. The permutation layer [13,14,16]

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

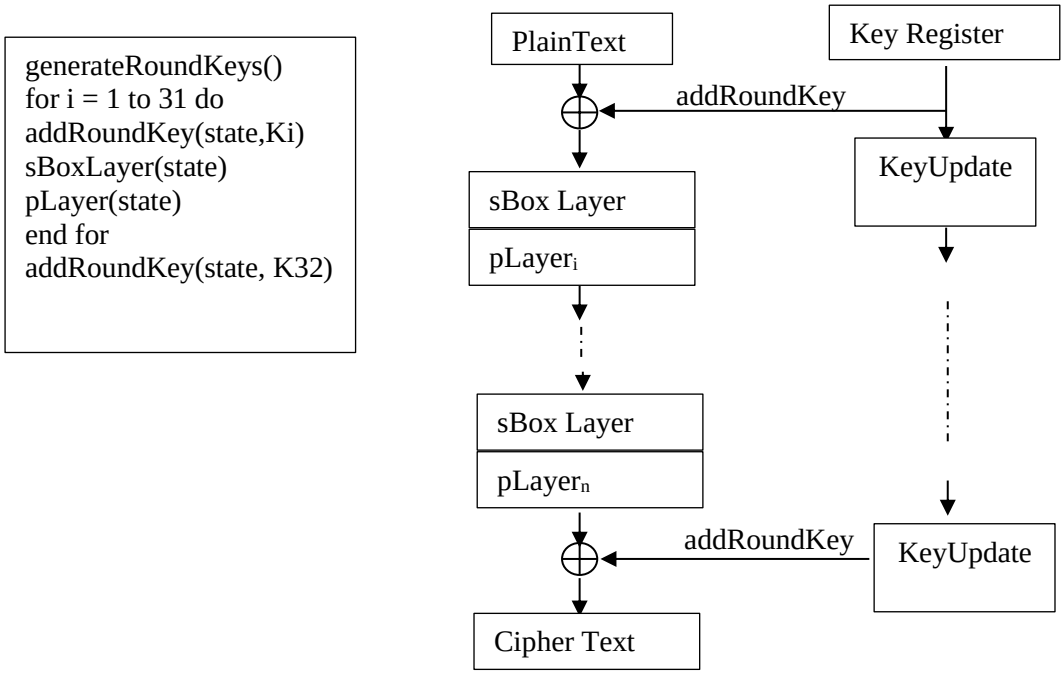

Figure 1. A block diagram of PRESENT algorithm. [14,15]

3. The block cipher speck algorithm

Speck was proposed publicly in June 2013 by a group of researchers in the US National Security Agency's Research Directorate. Speck should be flexible enough to perform well on the full spectrum of constrained platforms, and this motivated us to choose the simplest components possible. But the simplicity of the designs had an added benefit: we ended up with algorithms that have exceptional performance on high-end platforms as well. As far as, Speck has the highest throughput on 64-bit processing of any block cipher implemented in software [18].

The desire for flexibility through simplicity motivated us to limit the operations used within Simon and Speck to the following short list [5, 20]:

- modular addition and subtraction, + and −,
- bitwise XOR, ⊕,
- bitwise AND, &,
- left circular shift, S^{j}, by j bits, and
- right circular shift, S^{-j}, by j bits.

Speck gets its nonlinearity from the modular addition operation, which slightly favors software performance over hardware. Simon's nonlinear function is a bitwise AND operation, which tends to favor hardware over software. But modular addition can be computed efficiently in hardware, and similarly, bit-wise AND is easy and natural in software. The round functions for Simon 2*n* and Speck 2*n* each take as input an *n*-bit *round key k*, together with two *n*-bit *intermediate ciphertext* words. For Simon, the round function is the 2-stage Feistel map [5].

A further limitation we saw was that existing lightweight block ciphers tended to have a fixed block size, and one, or at most two, key sizes. We wanted to provide the additional flexibility to tailor a block and key size to the application at hand. To that end, Simon and Speck each have multiple instantiations, supporting block sizes of 32, 48, 64, 96, and 128 bits, and with up to three key sizes to go along with each block size. Each family provides ten algorithms in [18,20]. Table 3 lists the different block and key sizes, in bits.

Table 3. Simon/Speck parameters [18]

block size	key sizes
32	64
48	72, 96
64	96, 128
96	996, 144
128	128, 192, 256

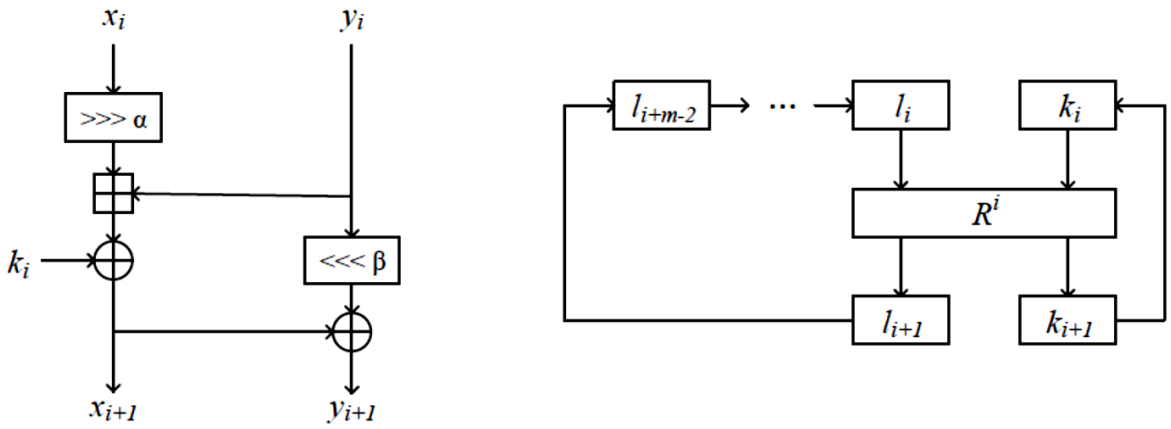

Figure. 3. The SPECK rounds function and the schedule keys. Ri is the SPECK round function with i acting as the round key [20]

4. The proposed IoT Security

The aim of the proposed system is to design a security mechanism based on different security algorithms in order to safe sensors data the administers computer through the network. The security mechanism is suggested to make the sensing data more secure through the network to connect the object sensors. The proposed system consists of four main layers (Data Collection layer, encryption layer, and authentication and integrity layer).

On the data collection layer, the sensors are distributed as clusters and each group contains several different sensors. Therefore, the sensors will be deployed in specific locations to read the situation or the surrounding environment by many parameters that will help to determine the decision by these parameters and in the data collect/aggregates layer, each distribution of data collection devices on each set of sensors. So that, each set of sensors is controlled by the device like microcontroller like Raspberry. The IoT devices work with continuous numerical data that is generated continuously and at specified time intervals. This data must be secured using an advanced security mechanism. Each part of the IoT device must be secured with the appropriate security system. The second layer is the security mechanism. The Proposed Security Mechanism (PSM) consists of three stages: 5-D chaos keys generation, sensing data encryption, and authentication(integrity). This layer will be used three proposed algorithms to provide the security and hashing authentication to the data that is collected by sensing layer.

PSM contains a Fuzzy control to select the one of the proposed encryption algorithms in each block encryption depending on the chaos keys to achieve better security results. PSM is provided through a combination of security algorithms that work together and are interconnected in their work. The use of PLSM came from the need to handle many types of the attacks that faces IoT environment. PLSM combine more than one security stage, these stages are:

- Proposed new novel 5-D Chaos key generation
- Fuzzy Encryption Control (using K1, K2, and K3)
- Hash function (SHA3 -256)
- Proposed Encryption Algorithms (Hybrid Present-Speck Algorithm (HPSA), Modified Present Algorithm (MPA), and Modified Speck Algorithm (MSA)).

Chaos Keys Generation was used to generate random numbers by using a 5-D novel chaotic system (named as Haider chaotic system) with different initials and parameters values to produce 5-D chaos keys values. Chaos keys used in all PSM algorithms: in generating their encryption/hash keys, and in some encryption functions.
Fuzzy control stage was proposed to controlling the selection one encryption algorithm (from the three proposed encryption algorithms) for encrypt the sensing data block. The selection is depending on the 27 fuzzy rules based on chaos keys number of (K1, K2, and K3). Each K1, K2, and K3 generated sequences number will divided in to set of two digits sequences (with positive range from 00-99) to be helpful in decision of Fuzzy rules.
In authentication process, SHA3-256 algorithm will be used to produce a faster hash for collecting sensing data. So, we proposed to use the hash function with seed and keys from the chaos keys generated. This architecture was relied upon as the basis for building an algorithm for hashing production and a two-dimensional from chaotic system will be used to produce alternative primary values for values that will be replaced with the initial values of SHA3-256 algorithm.

Figure 4 shows the structure and the relationship between the proposed system and all the improved algorithms proposed, PLSM and other layer components.

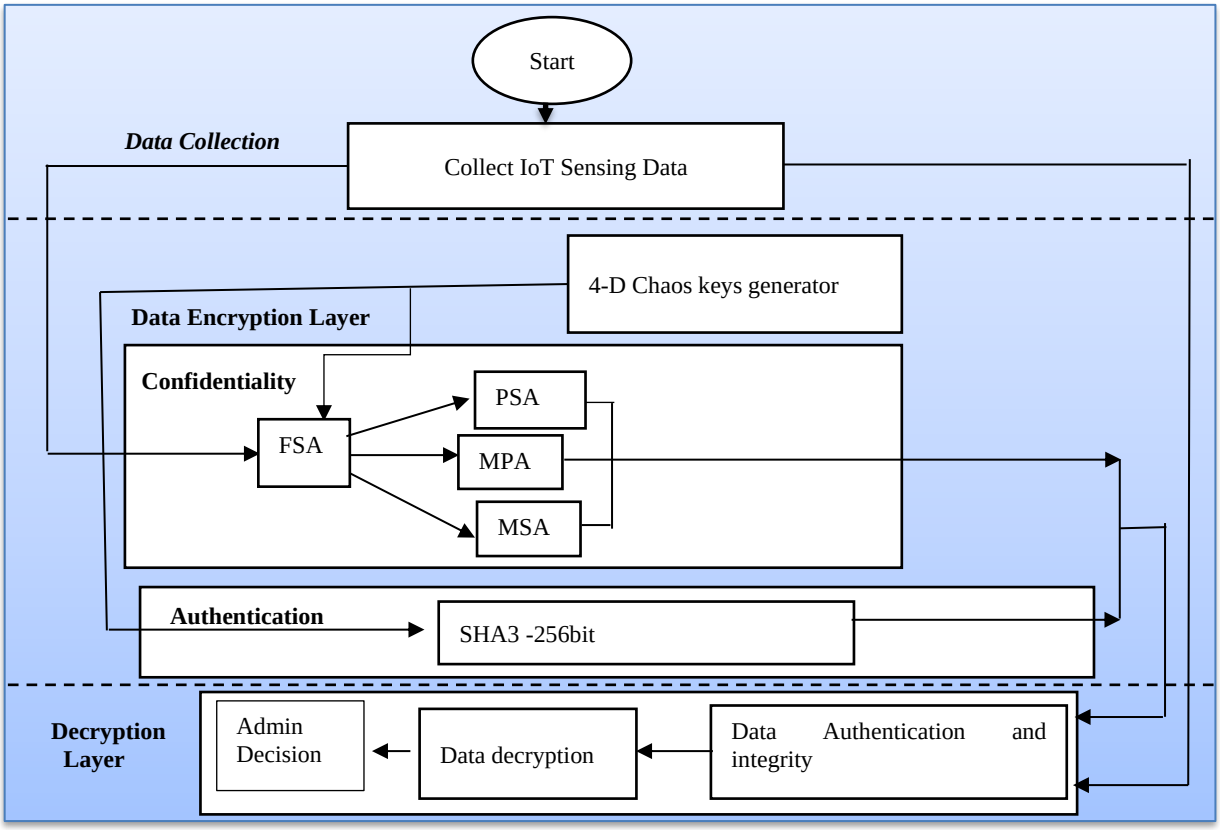

Figure 4. Block diagram of the proposed system structure

4.1. The 5-D chaos keys generation

The proposed novel 5-D chaotic system of differential dynamic chaotic equations is being examined (tested) and implemented on the basis of chaos-theory, as well as to investigate chaotic properties (such as randomness, dynamics, and sensitivity to the principle(initials) and equation-parameters) in generating a set of numerical output sequence. The proposed novel 5-D chaotic system equations are:

$$\begin{aligned} Xn+1 &= (r*(Pn-Yn)-u*(Zn-Kn))/(Yn*Xn-u) \\ Yn+1 &= (s*(Yn-Xn)+(Xn - s*Kn)-Pn*r)/(1+Yn) \\ Z\,n+1 &= (u*(Yn-Zn)-Xn/(s-Yn)+u*(Zn-r))/(1+Zn) \\ Kn+1 &= (Xn*u- r - b*Zn - u*Pn)/u \\ Pn+1 &= b*(S*Kn - u*Xn- Yn**u)/Kn \end{aligned} \quad (1)$$

Where: Pn, Kn, Zn, Yn, Xn are selected to be numerical values start with initials values. Mn, Zn, Yn, Xn belong to interval (-10, 0.95), s=(1.1, 65), r=(12,120), b=(9.0,10.5), and u=(0.001, 10.1)

The proposed novel 5D chaotic system (named Haider chaos system) was implemented and tested, and the Lyapunov exponents were computed for different initials and parameters. The proposed novel 5-D chaotic system has super chaotic Lyapunov values with positive values in five values with maximum Lyapunov values (0.14503543, 0.06404174, 0.06267101, 0.06537709, and 0.05376733) using parameters (s=1.1, r=12, b=9.0, and u=0.01).

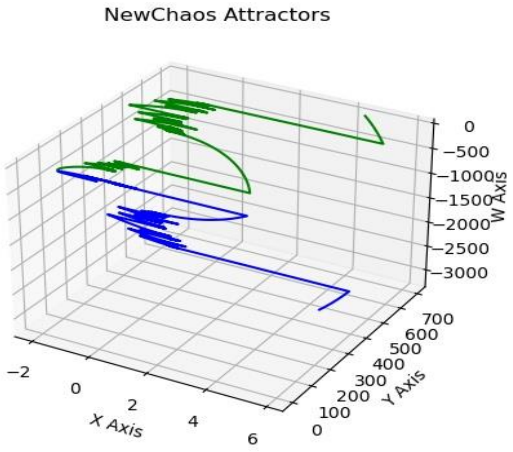

Figure 5. The map results of the novel 5-D system (Haider chaos system)

3.2. The proposed encryption algorithms

The proposed encryption algorithms content three encryption algorithms: proposed hybrid Speck-Present (HSPA), modified Present (MPA), and modified Speck (MSA) algorithms. First, the hybrid Speck-Present algorithm designed from combined the Speck algorithm (with 10 rounds for reduce the Speck encryption time) with the Present algorithm. Figure (5) illustrated the hybrid Speck-Present algorithm. A Speck algorithm was insert as a layer in the Present round layers to increase the complexity of the Present encryption results and I proposed algorithm is modified increase the Present algorithm to avoid many attacks like impossible differential attack.

The modification adapted the HSPA was combines with 5-D chaos keys (K1, K2, K3, K4, and K5) generated by (proposed novel 5-D chaotic system). They were distributed between Present algorithm and Speck algorithm as encryption keys. These chaos keys were used to rise the randomness to the output cipher text and gives best strengths to the HSPA.

The second proposed algorithm is the modified Present algorithm (MPA). The modification to the Present is by adding the multi-level XOR operation in the Present Rounds. The chaos keys (K1, K2, K3, and K5) are used in the XOR operation to increase the strength of the output encrypted text with randomness of the keys. The round keys will be generated from (K2 and K4). Figure (6) shown the proposed modified Present algorithm.

The third proposed encryption algorithm is the modified Speck algorithm (MSA). Figure (7) shows the proposed modification to the Speck encryption round. The chaos keys add with XOR operations added to the rounds in order to get more security and randomness to the output results.

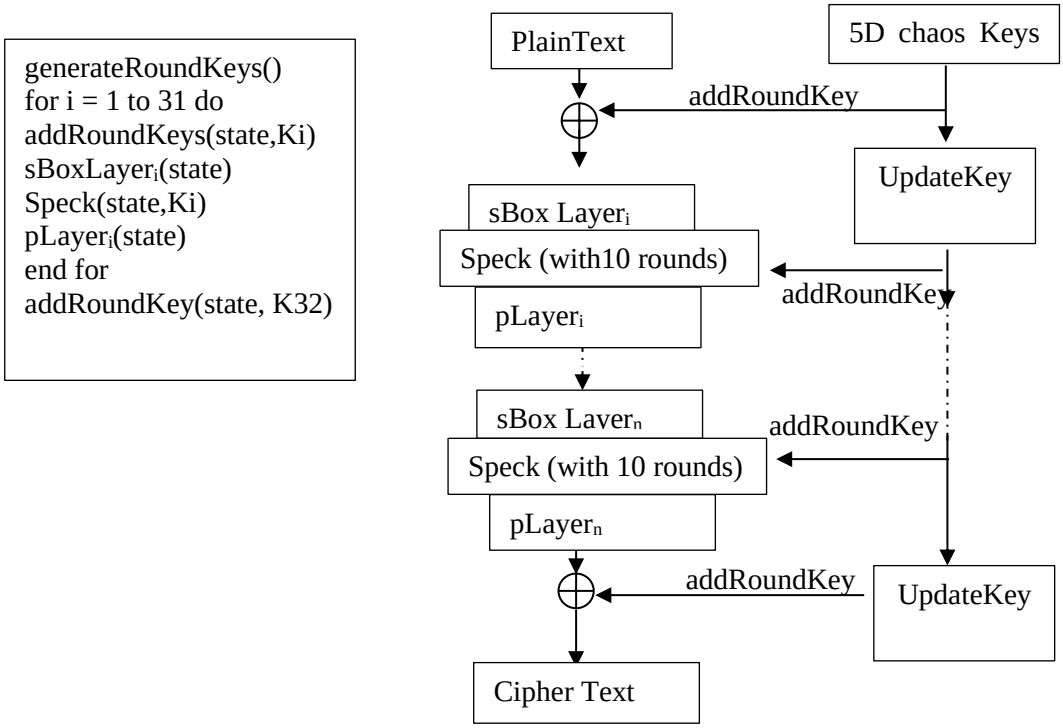

Figure 5. The block diagram of the proposed hybrid speck-present algorithm

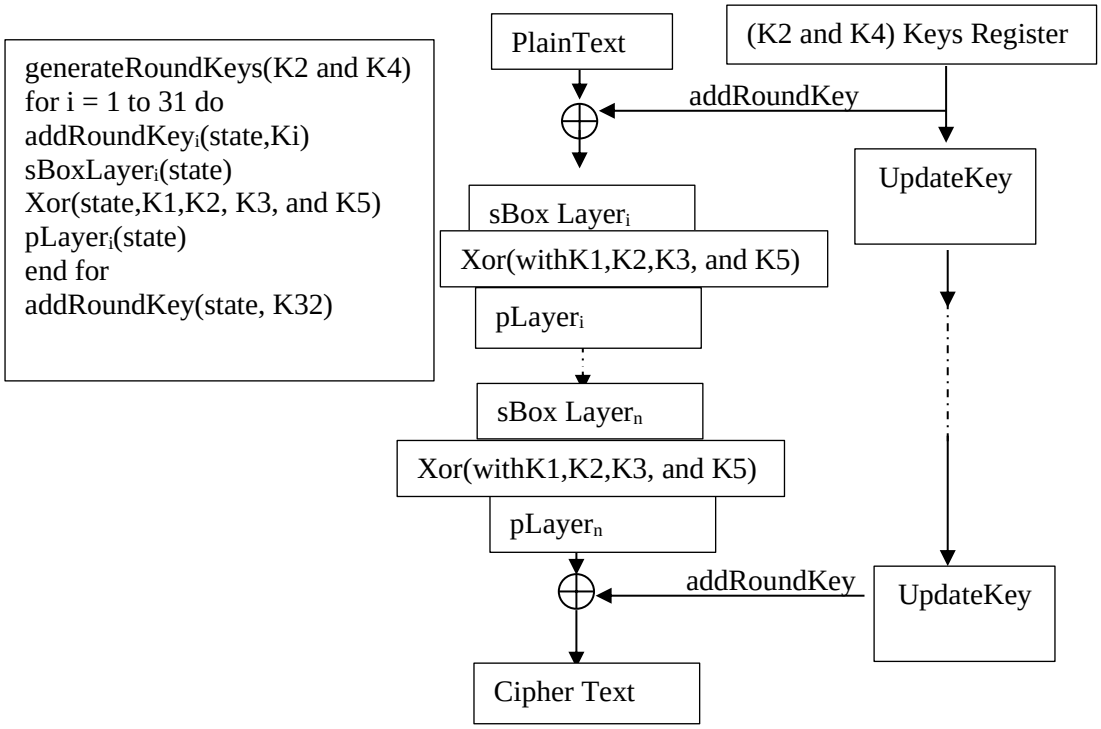

Figure 6. The block diagram of the proposed modified present algorithm

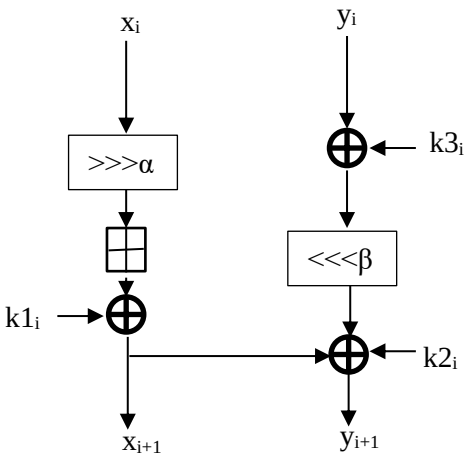

Figure 7. The round function of the proposed modified speck algorithm (MSA)

3.3. Fuzzy encryption control

The generated chaos keys (K1, K2, and K) were used in the selection the encryption algorithm for each block IoT sensing data. The Fuzzy compare and take decision between the two digits from each K1, K2, and K3 sequences during time encryption and apply the Fuzzy rules to take decision. Figure (8) illustrated Fuzzy membership function. The following rules are applied to calculate the decision ratio. Table (3) illustrates fuzzy rules.

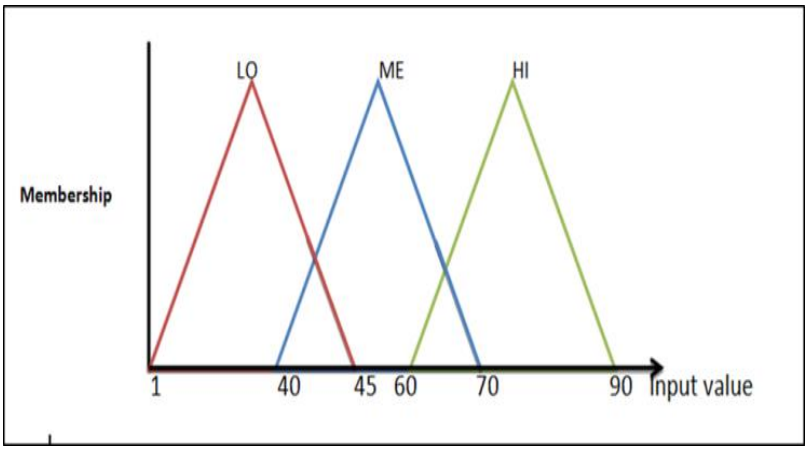

Figure 8. Fuzzy membership function

For Fuzzy rules set,
If K1= LO and K2 =LO and K3=LO then use HSPA
If K1= ME and K2 =LO and K3=LO then use HSPA
If K1= LO and K2 = ME and K3=LO then use MPA
If K1= ME and K2 = ME and K3=LO then use MPA
If K1= HI and K2 =LO and K3=LO then use MSA
If K1= HI and K2 =ME and K3=LO then use MSA
If K1= LO and K2 = HI and K3=LO then use MPA
If K1= ME and K2 = HI and K3=LO then use MSA
If K1= HI and K2 = HI and K3=LO then use HPSA
If K1= LO and K2 =LO and K3=ME then use MPA
If K1= ME and K2 =LO and K3=ME then use HSPA
If K1= LO and K2 = ME and K3=ME then use MSA
If K1= ME and K2 = ME and K3=ME then use MSA
If K1= HI and K2 =LO and K3=ME then use HSPA
If K1= HI and K2 =ME and K3=LO then use MPA

If K1= LO and K2 = HI and K3=ME then use MPA
If K1= ME and K2 = HI and K3=ME then use MSA
If K1= HI and K2 = HI and K3=ME then use MPA
If K1= LO and K2 =LO and K3=HI then use HSPA
If K1= ME and K2 =LO and K3=HI then use HSPA
If K1= LO and K2 = ME and K3=HI then use MSA
If K1= ME and K2 = ME and K3=HI then use MSA
If K1= HI and K2 =LO and K3=HI then use MPA
If K1= HI and K2 =ME and K3=HI then use MPA
If K1= LO and K2 = HI and K3=HI then use MSA
If K1= ME and K2 = HI and K3=HI then use MSA
If K1= HI and K2 = HI and K3=HI then use HPSA

5. Experimental results

IoT sensing data security is important trend fields. In the case of the implementation of the suggested system, we need different requirements, including these requirements hardware and software. Hardware requirements are (types of sensors, Raspberry pi devices, as well as a computer responsible and network. Software requirements are (Python programming language, Raspberry Linux operating system, as well as Windows 10 operating system for the computer responsible).

A Hybrid Speck-Present algorithm, modified Present and modified Speck algorithms are implement and tested to ensure that the IoT sensing data more secure and avoiding many attacks. The results of these proposed algorithms are tested using the NIST tests, humming distance, and Entropy.

Applied tests and benchmark on the proposes ciphering algorithms (NIST tests) shown in Table 4 shows the time execution for the proposed HSPA, MPS, and MSA comparing with their algorithm before modifications.

Table 4. The average time results (in sec)

Operation	HPSA time (sec)	MPA time (sec)	PA time (sec)	MSA time (sec)	SA time (sec)
Encrypt (128B)	0.00151	0.00110	0.00108	0.001	0.0010
Decrypt (128B)	0.00243	0.00200	0.00200	0.0011	0.0010
Encrypt (1 KB)	0.0551	0.0487	0.0480	0.0021	0.0020
Decrypt (1KB)	0.0562	0.0507	0.0499	0.0022	0.0020
Encrypt (10KB)	0.204	0.186	0.179	0.0345	0.0342
Decrypt (10KB)	0.226	0.198	0.188	0.0351	0.0347
Encrypt (100KB)	0.358	0.317	0.310	0.0555	0.0551
Decrypt (100KB)	0.366	0.321	0.312	0.0557	0.0552
Encrypt (1MB)	1.278	0.988	0.979	0.1230	0.1289
Decrypt (1MB)	1.299	1.01	0.997	0.1234	0.1290

From Table 4, the proposed algorithms have close execution-time in (encryption, decryption) comparing with the algorithms before modification, the inputs were generated random cases with various sizes (128 Byte ,… 1MB) with an input-block 128bits. The average time encryptions of the proposed system from (0.0015 sec to 1.299 sec). Table 5 illustrates the results of NIST tests of the proposed encryption algorithms to guarantee that the proposed HPSA, MPA, and MSA have the best security, and can avoid different attacks. The proposed encryption algorithms are passed in all NIST tests.

Table 5. NIST tests results of the proposed cipher algorithms

Test Name	**HPSA**	**MPA**	**MSA**
Frequency (Monobit) test	0.8797	0.8781	0.6467
Runs test	0.7561	0.6998	0.5341
Discrete Fourier transform	0.3034	0.1989	0.1443
Block frequency	0.8556	0.8320	0.6744
Longest runs test	0.1897	0.0673	0.0564
Cumulative sums test	0.8003	0.7988	0.6896
Serial test	0.9812	0.8870	0.7564
Matrix rank test	0.6645	0.5998	0.5009
Overlapping template test	0.9788	0.9089	0.9105
Linear complexity test	0.9754	0.9631	0.9547
Nonoverlapping template test	0.7987	0.6698	0.6734
Random excursions variant test	0.7856	0.6867	0.6549
Random excursions test	0.9803	0.8324	0.8276

Our proposals are resistant to many of the cryptanalytical attacks common to block chips and chaotic systems. The system proposed provides adequate safety and reliability, according to Table 5. Tables 6, 7, 8, and 9 illustrated the Correlation Coefficients Analysis**,** humming distance, entropy, MAE, NPCR, and UACI.

Table 6. Correlation coefficients of encrypted text

Text Size (byte)	**HPSA**	**MPA**	**MSA**
100	0.00723	0.00325	0.00224
200	0.00691	0.00415	0.00251
300	0.00645	0.00329	0.001341
400	0.0068	0.00426	0.00214
500	0.00635	0.00618	0.00211

Table 7. Entropy results of encrypted texts

Text Size (byte)	**HPSA**		**MPA**	**MSA**
100	7.2021		7.1641	7.3226
200	7.2374		7.3519	7.51124
300	7.1038		7.0573	7.15723
400	7.6259		7.2572	7.6571
500	7.54417		7.3413	7.75413

Table 8. Hamming distance results of encrypted texts

Text Size (byte)	**HPSA**	**MPA**	**MSA**
100	363	406	396
200	752	784	791
300	1245	1328	1367
400	1547	1569	1632
500	1624	1852	1847

Table 9. Plaintext sensitivity in terms of MAE, NPCR and UACI

Measure type	**HPSA**	**MPA**	**MSA**
MAE	36.994	29.257	31.827
NPCR	63.987	57.365	63.258
UAC1	20.765	16.476	19.254

Apparently, the HPSA, MPA, and MSA have a sensitivity to the plaintext; and indicates they more sensitivity to changing in output results.

6. Conclusions

In this paper, we have presented the lightweight intelligent IoT data sensing cipher system depends on robust novel chaotic system (for generating encryption keys) and hybrid or combination to the encryption algorithm (PRESENT-SPECK) with some modification, got the satisfactory NIST, Correlation Coefficients, Hamming distance, Entropy, MAE, NPCR and UACI. The proposed cipher algorithms build under the ARX-based structure with useful confusion and diffusion properties with Fuzzy controlling. The proposed cryptosystem analysis results prove the proposed cipher algorithms have strong security due to a highly large key-space (least 2^{2560}probable various combines sets of the secret keys) and the all NIST test are passes for both encryption algorithm and generated keys.

References

[1] P. Gokhale, O. Bhat and S. Bhat" Introduction to IOT", International Advanced Research Journal in Science, Engineering and Technology, Vol. 5, No. 1, 2018.

[2] A. Majed, H. K. Hoomod:" Secure Email of Things Based on Hyper Chaotic system", Al-Mustansiriyah University, Baghdad, Iraq, M.Sc. Thesis ,2020.

[3] A. Bogdanov, L.R. Knudsen, G. Leander, et al., "Present: an ultra-lightweight block cipher". *Proceedings of the international workshop on cryptographic hardware and embedded systems*, Vienna, 10–13 September 2007, pp.450–466. Berlin: Springer.

[4] L. Song, Z. Huang, Q. Yang, "Automatic Di_erential Analysis of ARX Block Ciphers with Application to SPECK and LEA", Australasian Conference on Information Security and Privacy, 9723. 379-394. 10.1007/978-3-319-40367-0_24

[5] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers," Simon and Speck: Block Ciphers for the Internet of Things", ACR Cryptol. ePrint Arch., Vol. 2015, p. 585, 2015.

[6] R. Beaulieu, S. Treatman-Clark, D. Shors, et al. The Simon and speck lightweight block ciphers. In: Proceedings of the 52nd ACM/EDAC/IEEE design automation conference (DAC), San Francisco, CA, 8–12 June 2015, pp.1–6. New York: IEEE.

[7] G. Yang, B. Zhu, V. Suder, et al., "The Simeck family of lightweight block ciphers", Proceedings *of the international workshop on cryptographic hardware and embedded systems*, Saint-Malo, pp.307–329, 2015.

[8] D. Dinu, L. Perrin, A. Udovenko, et al., " Sparx: a family of ARX-based lightweight block ciphers provably secure against linear and differential attacks", *Proceedings of NIST workshop on Lightweight Crypto CRYPT'16*, p.121. Gaithersburg, MD: NIST, 2016.

[9] S. Banik, S.K. Pandey, T. Peyrin, et al., "Gift: a small present", *Proceedings of the 19th international conference on cryptographic hardware and embedded systems*, Taipei, Taiwan, pp.321–345, 2017.

[10] B. Koo, D. Roh, H. Kim, et al., "Cham: a family of lightweight block ciphers for resource-constrained devices". *Proceedings of the international conference on information security and cryptology*, Seoul, South Korea, pp.3–25, 2017.

[11] A. G. Sawant, S. Kamthe, Y. Shaha, B. Morajkar and A. Sakpal., "Implementation of SIMON & SPECK Algorithm." Journal of emerging technologies and innovative research, Vol. 6, No.1, pp.292-296, 2019.

[12] G. Leander, C. Paar, A. Poschmann, K. Schramm, "New lightweight DES variants.", Fast Software Encryption, *Lecture Notes in Computer Science*, Vol. 4593, 2007.

[13] N. bagheri and R. Ebrahimpour and N. Ghaedi Bardeh,"New differential fault analysis on PRESENT", EURASIP Journal on Applied Signal Processing, p.145, 2013.

[14] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher", Cryptographic Hardware and Embedded Systems, *Lecture Notes in Computer Science*, vol 4727, 2007.

[15] J.Y. Cho, ed. by J Piepryzk, "Linear cryptanalysis of reduced-round PRESENT", CT-RSA (Springer Heidelberg, pp. 302–317, 2010

[16] W. Zhang, et al., "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms" *Science China Information Sciences*, Vol.58, No.12, pp.1-15, 2015.

[17] I. Bhardwaj, A. Kumar, and M. Bansal. "A review on lightweight cryptography algorithms for data security and authentication in IoTs", 2017 *4th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2017.

[18] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, R. Beaulieu, et al., "The SIMON and SPECK lightweight block ciphers." 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp.1-6, 2015.

[19] L. Song, Z. Huang, Q. Yang, "Automatic Differential Analysis of ARX Block Ciphers with Application to SPECK and LEA.", *Lecture Notes in Computer Science*, vol 9723, 2016.