DOI: http://dx.doi.org/10.26483/ijarcs.v11i5.6657

Volume 11, No. 5, September-October 2020

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A NOVEL FRAMEWORK FOR PRIVACY-PRESERVING USING DATA AGGREGATION IN INTERNET OF THINGS (IOT)

Manas Ranjan Mohapatra
Research Scholar
Department of Computer Application
SSSUTMS, Sehore
M.P., India

Dr. Jitendra Sheetlani
Associate Professor
Department of Computer Application
SSSUTMS, Sehore
M.P., India

Dr. Rasmi Ranjan Patra
Assistant Professor
Dept. of CSA
CPGS, OUAT
Bhubaneswar, India

Abstract: Today with the development of Internet of Things (IoT) technology its security and privacy becomes very much essential because this technology provide the facility for accessing the devices or resources from anywhere and anytime. For providing the security and privacy to IoT various technologies has been developed but in this we propose data aggregation mechanism in which it uses one-time pad and symmetric key to make our data secure and preserve the privacy of it. This mechanism takes less computation time, improve processing speed and also require less cost in implementation. The implementation of proposed approach is done in widely used Python technology which produces better results. In last the proposed mechanism provides more security and preserves our data.

Keywords: **I**oT, Aggregator, Symmetric key, Privacy

I. INTRODUCTION

It seems that, billions and trillions [1, 2] of intelligent entities around us will be connected themselves and to their physical environment also [3] as per the prediction by the Internet of Things (IoT). IoT systems will provide a whole new set of advanced features based on gradual elegant data acquisition in a briefly colonized environment with intelligent things. Illustration of these may be health care system, modern house management system, smart city service or collaborative sensing applications [4, 5]. Gradually unseen, extensive collection of data and its processing and distribution in the middle of people's personal livings create significant privacy issues. Ignoring these problems may result unintended outcomes such as failure to accept but succeed in noble services, damaging reputation or high cost legal cases. Only 3 models of IoT involved projects which cause major difficulties as a result of unanswered privacy concerns are [6]:

1. Italy based retailer Benetton's general ban in the year 2003 [7, 8].

2. Repeal of the Dutch Smart Metering Bill (2009) [9].

3. Current opposition to the INDECT research project funded by EU FP7 [10, 11].

Privacy is a popular research interest in various technologies and application fields those mainly facilitating future IoT. This includes RFID, WSN, Web customization and mobile applications and programs. Even with significant additions from these communities, there is no full view of the growing IoT related privacy concerns because it is a developing concept that encompasses increasing technologies and shows an assortment of dynamic properties. Of these, we note the explosiveness in the number of smart objects and the advanced methods of system interaction and user response. Evidently, these modern IoT characteristics exaggerate privacy problems and pose unexpected threats that demand technical issues. These threats, whether acknowledged or novel, should be taken into account (i) the IoT's reference standards that clarify its particular entities and data flows, (ii) the current privacy legislation perspective, and (iii) the unique and evolving aspects of the IoT. Unawareness of emerging issues and their appropriate protections may jeopardize the performance of newly created services and their user privacy [11].

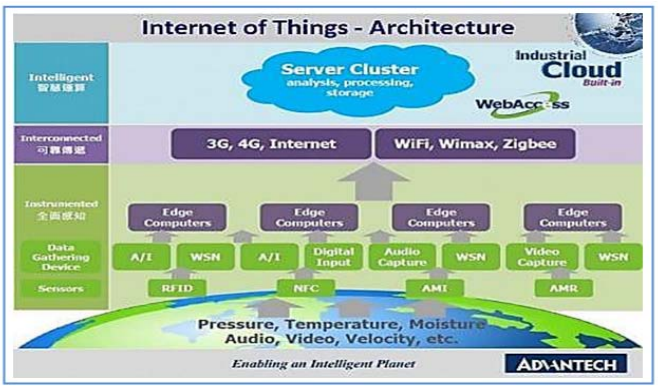


Figure 1. IoT Architecture

The organization of rest of the paper is as done as follows: Section 2 briefly discusses the work done by the various researchers for privacy preserving in IoT. Section 3 presents the proposed work and architecture developed for the implementation of privacy in IoT. In section 4 experimental results and its analysis is illustrated. And last section presents the overall conclusion of the work and their future aspects.

II. RELATED WORK

This section of the research paper discusses the earlier work by the various researchers and authors to implement privacy preserving in IoT which are explained below:

Feyza Yildirim Okay et al. introduced two numbers of SDA protocol (i) FCSG-DF (ii) FCSG-P with efficient as well as lightweight features with respect to SGs based on fog computing (FCSG), and these two provided solution for satisfying the above needs. This two SDA protocols features less computational overhead with much more improved response time. Moreover, hierarchical fog computing structure of "FCSG" preserves the privacy of precisely energy-consuming data through privacy-preserving data aggregation processes. Result of extensive performance evaluation and analysis shows that in comparison to cloud-assisted schemes which involves no data aggregation the proposed protocols are far superior considering data transmission and storage efficiency. Besides, it was verified in privacy analysis that these two protocols ensure each layer's data privacy in an effective way [12].

Mengyao Zheng et al. represents classification of cloud computing based privacy-preserving machine learning approaches that already exists. Moreover, it introduce a privacy-preserving inference scheme where IoT objects are run in a neural network which is lightweight in nature thereby obfuscate the data prior to its transmission, next in the cloud a deep neural network is run thereby classifying the previously obfuscated data. Performance evaluation is done with MNIST dataset that yields satisfactory results [13].

Prem Prakash Jayaraman et al. privacy-preserving problem is handled by suggesting new privacy-preserving methods for IoT data, introducing an IoT Architecture that holds privacy-preserving. Additionally by employing this concept its implementation is done as an efficient system of proof of concept for ensuring the data privacy. These techniques utilize a number of data stores of IoT cloud for privacy-preserving of collected IoT data. Both the architecture and implementation are on the basis of OpenIoT extension-an open source cloud solution for IoT. Experimental evaluations and its results are described too in terms of efficiency and performance [14].

Yuwen Pu et al. presented 2 different schemes for data aggregation in preserving customer's private information. First scheme slices the IoT device's data indiscriminately, keeping with it a single piece. Then symmetric encryption is done to send other pieces to all the group devices. After that, those sliced pieces received and that single piece added together. Then the immediate result after computation is reported to the aggregator. In addition, secure communication is ensured by the use of homomorphic and AES encryption. The second scheme also makes the use of that same slicing approach. To avoid the exposure of exchanged actual data of devices during mutual data merge, here noise data are introduced. Secure communication within devices and aggregator is achieved too using AES encryption. The experimental analysis signifies that, both these method guarantee integrity and confidentiality of IoT device's data, thus helps to defy external, internal and colluding attack, and likewise [15].

Rongxing Lu et al. proposed LPDA for fog computing enhanced IoT which is a lightweight approach for privacy preservation in data aggregation. It is formulated upon features of One-way hash chain, Chinese Remainder Theorem and Homomorphic Paillier encryption for solving aforementioned problem and additionally at the network edge for earlier filtering of injected false data. LPDA's in-depth security analysis signifies it as distinct privacy preserving technique being more secure and improved with privacy. Moreover, the result of extensive performance evaluations shows LPDA as truly light-weight in this type of IoT [16].

Inayat Ali et al. reviewed the latest PPDA techniques along with their comparative analysis. Here comparative analysis is carried out by taking into the factors which are privacy level, energy usage by sensor nodes, computational cost, sensor node life, communication overhead and resistance against malicious aggregator. Latest techniques are investigated here to give a detail analysis of the each and every step of these techniques. They have shown that their survey paper covers the latest and extensive study of PPDA techniques [17].

Chunqiang Hu et al. presented a data aggregation scheme for customer's private data preservation. This scheme slices the IoT device's data indiscriminately, keeping with it a single piece. Then symmetric encryption is done to send other pieces to all the group devices. After that, those sliced pieces received and that single piece added together. Then the immediate result after computation is reported to the aggregator. Analysis signifies that both data integrity and confidentiality can be achieved here thus it helps to defy external, internal and colluding attack, and likewise [18].

III. PROPOSED METHODOLOGY

Here we have described in brief, how the proposed architecture works for making the IoT devices secure and maintain their data more privacy or confidential. We have assumed that IoT device has certain storage and computational power. Here a group of IoT device is formed which belongs to same residential area and so on. These devices have a common secret key that is known to himself and aggregator. Our scheme design novel architecture to preserve the data transmitted from the IoT devices. Here, we draw the data flow diagram of the proposed architecture for privacy preserving of data aggregation. Initially, it takes the data as input from the various nodes or IoT devices, then we have applied the multilayer encryption technique to make our data secure. But when all the nodes will start transmission at the same time it increases the traffic over the network and also reduces the life of network, so to overcome this we need an aggregator which is connected to every node. Aggregator receives the signals coming from all the nodes and forms a single signal using aggregation algorithm and by adding aggregator key to signal send to the sink (receiver end).

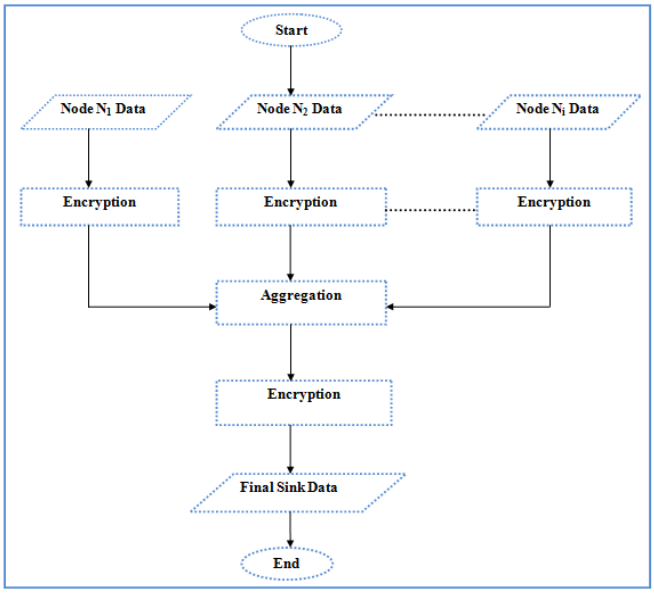


Figure 2. Data Flow diagram of Proposed Methodology

3.1. Architecture

As the Architecture Followed to make such system is like the following. The proposed system architecture is beneficial for:

1. Easy Encryption
2. Easy Implementation
3. Fast
4. Cost Efficient

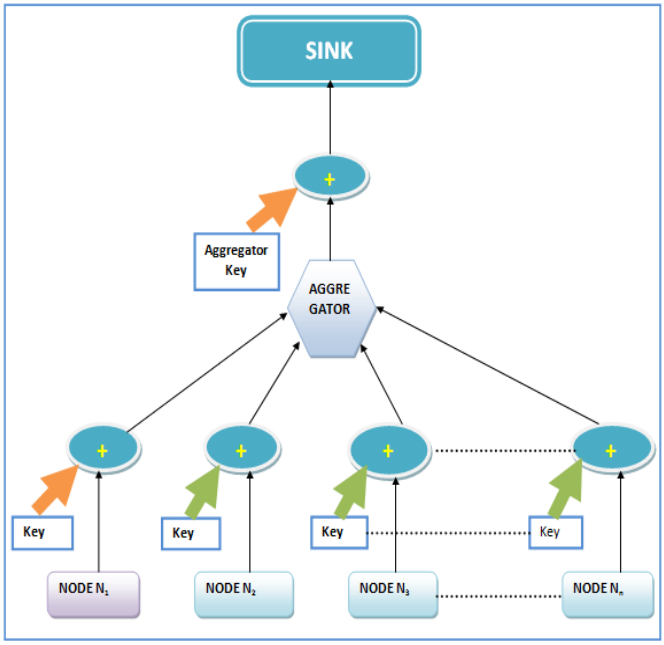


Figure 3. Proposed Architecture of Privacy Preserving for Data Aggregation

Description of Architecture

As we can see the architecture on the above page, let's try to brief all the elements.

1. Nodes
   As when we are working with IOT then we have many sensors for devices attached with them which passes the data to the main system. So, all those end devices are known as Nodes of the system.
2. Key
   As key is the unique code which we use to encrypt the Plain text when encryption takes place. Uniqueness of key decides security of cipher text.
3. Encryption
   The process to converting plain text into cipher or encrypted text is known as encryption. This technique converts a human readable message to a form which human can't simply read.
4. Aggregator
   When all nodes send data together to the main system, this traffic causes several problems like, increase traffic at main system, decrease life of all nodes, and traffic in the network. So to avoid such issues we need Aggregator which is connected with all nodes which combines all the signals from the nodes using data aggregation algorithm and then send to the main system.
5. Aggregator Key
   This is as same as the key, but it is he unique key of aggregator.
6. Sink
   Sink or main system is the brain of IOT devices, where all nodes need to send their data. Hence it is the processing unit of the system.

Proposed algorithm for the architecture

1. **AGG (EDN_i, AgKey)**
2. **EDNi (EKN_i, PTN_i)**
   - **2.1. Input ← PTN_i**
   - **2.2. UE (eku_i, ptu_i)**
     - **2.2.1. Input ← ptu_i**
     - **2.2.2. Use Calculation Matrix**
     - **2.2.3. Find dku_i**
     - **2.2.4. Find ctu_i using dku_i**
     - **2.2.5. Return ctu_i**
   - **2.3. Repeat step 2.2 for all $u_i \epsilon N_i$**
3. **Repeat step 2 for all N_i**
4. **Return EDN_i**
5. **Input ← EDN_i**
6. **Repeat step 2.2**
7. **Return AED_i**
8. **DAED (AED_i, DK_i)**
   - **8.1. UD ($aedu_i$, dku_i)**
   - **8.2. Input← $aedu_i$**
   - **8.3. Find $aptu_i$ using dku_i**
   - **8.4. Repeat step 8.1 for all $u_i \epsilon AED_i$**
   - **8.5. Return $aptu_i$**
9. **Return APT_i**

Here:

1. **AGG** is Aggregated Function performed by aggregator for aggregation of encrypted data (ED) of nodes N_i
2. N_i is set of nodes, where **i= 1, 2, 3……..n**
3. u_i is set of units in a particular node of N_i i.e. $u_i \epsilon N_i$
4. EDN_i is the set of Encrypted data from N_i number of nodes that contains set of cipher text resulted from unit encryption(ctu_i) i.e. $ctu_i \epsilon EDN_i$

5. **EKN_i** is the set of encrypted key for **N_i** nodes that contains set of encrypted key for unit encryption(**eku_i**) i.e. **$eku_i \epsilon EKN_i$**
6. **PTN_i** is the set of plain text input for **N_i** nodes that contains set of plain text resulted from unit encryption (**ptu_i**) i.e. **$ptu_i \epsilon PTN_i$**
7. **AED_i** is the aggregated encrypted data produced by aggregator from **N_i** nodes to the sink
8. **UE** is the Unit encryption
9. **AgKey** is the Aggregator key same as **EKN_i**
10. **DAED** is the decryption of aggregated encrypted data
11. **DK_i** is the set of dynamic key for decryption process that contains set of dynamic key for each unit decryption(**dku_i**) i.e. **$dku_i \epsilon DK_i$**
12. **UD** is the unit decryption
13. **$aedu_i$** is the set of encrypted unit data for **u_i** set of units, **i =1,2,3…..n**
14. **APT_i** is the set of resulted plain text that contains set of resulted plain text for unit decryption (**$aptu_i$**) i.e**. $aptu_i \epsilon APT_i$**

The proposed method follows a unique way of aggregation of the data. As all the nodes wants data must be sent to aggregator so that these can be aggregated and can be sent to the sink. But as the architecture is showing first all the data will be passed from encryption using each node's private key. Then after key and Cipher text generated from each node will be combined and send to aggregator, the aggregator then combines all the data using aggregation algorithm and then again passes the data to encryption method with its own unique key so that multilevel encryption will lead to increase in security. So this is all how the data flows inside the architecture. Later on we will discuss the whole process with example. All the encryption modules are the same. So first we need to be clear about the encryption module so that we'll be able to understand the whole working easily.

Stage 1: Encryption

Let's discuss the encryption technique used. As in above architecture we can see that we are using multilevel encryption so that security of data can be raised to peak. As every time the encryption follows following things.

1. Key
2. Data or Plain Text
3. Technique or Algorithm
4. Cipher Text

For better understanding the encryption module let's have a look at the unit module of Encryption.

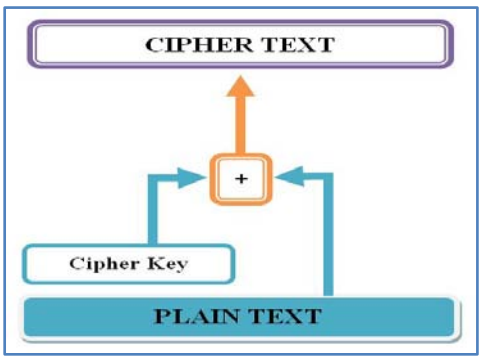


Figure 4. Encryption Process

Let's understand all the terminologies one by one:

1. Encryption: The process to converting plain text into cipher or encrypted text is known as encryption.
2. Plain Text: The data or message which needs to be converted or encrypted into cipher text is known as plain text.
3. Cipher text: The data which comes as result of encryption or we can say that the converted ex or message is known as Cipher text.
4. Key: The unique code used to encrypt the Plain the text is known as the key. The uniqueness of key decides the security of encryption.

Stage 2: Unit Encryption

As the scheme we are using follows a proper system or having its own architecture of encryption of Plain Text. Let's discuss the same.

Figure 5. Unit Encryption Process

Step 1

In this architecture first we need two things

1. Plain Text
2. Private Key

Let's take the example so that we'll be clear about the working.

Take the Plain text as "VALUES" And the Unique key As "KEY". As we first need to convert the key to the same length as of the Plain text. This can be simply achieved by repeating it till the length becomes same.

Updated key is like "KEYKEY". Now both the string is of 6 lengths.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1
3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2
4	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3
5	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4
6	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5
7	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6
8	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7
9	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8
10	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9
11	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10
12	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11
13	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12
14	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13
15	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14
16	16	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	17	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
18	18	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
19	19	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
20	20	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
21	21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
22	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
23	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
24	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
25	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
26	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Figure 6. Calculation Matrix

Step 2

Now our proposed method is completely based on a matrix as above in the Figure 6 (We also named this as calculation Matrix). Now by using that matrix we need to find the dynamic key. It's a unique key generated using Plain text and the static key so that security can be enhanced. So, we have our calculation matrix with us.

Step 3

Now we will convert the message and key into equivalent numbers as

Let 'A' is the '1', 'B' is the '2', and 'C' is the'3' and so on.

Now the plain text stands for V-> 22, A->1, L->12,

U->21, E->5, S->19

And the key stands for K->11, E->5, Y->25, K->11,

E- >5, Y->25

We have their equivalent Number with us.

Step 4

Now this time is to use the Calculate the Dynamic key using calculation matrix and formula for this is like

Dynamic key (x) = calculation matrix [key(x)] [Plain text(x)]

For every character of plain text and key we will trace character for dynamic key. By taking the key in X-axis and Plain text in Y-Axis

Taking this example forward as we have the

Numeric Plain text as [22, 1, 12, 21, 5, 19]

And the numeric Key as [11, 5, 25, 11, 5, 25]

The dynamic will be also of the same length because it is traced from the input string only.

Dynamic key (1) = calculation matrix [key (1)] [Plain text (1)]

= calculation Matrix [11] [22]

= 6

Dynamic key (2) = calculation matrix [key (2)] [Plain text (2)]

= calculation Matrix [5] [1]

= 5

Dynamic key (3) = calculation matrix [key (3)] [Plain text (3)]

= calculation Matrix [25] [12]

= 10

Dynamic key (4) = calculation matrix [key (4)] [Plain text (4)]

= calculation Matrix [11] [21]

= 5

Dynamic key (5) = calculation matrix [key (5)] [Plain text (5)]

= calculation Matrix [5] [5]

= 9

Dynamic key (6) = calculation matrix [key (6)] [Plain text (6)]

= calculation Matrix [25] [19]

= 17

We will first convert the dynamic key into equivalent characters.

As [F, E, J, E, I, Q] -> [6, 5, 10, 5, 9, 17]

Now we have following parameters with us

Input String = [V, A, L, U, E, S]

Converted Input string = [22, 1, 12, 21, 5, 19]

Key = [K, E, Y, K, E, Y]

Converted Key = [11, 5, 25, 11, 5, 25]

Dynamic Key = [F, E, J, E, I, Q]

Converted Dynamic Key = [6, 5, 10, 5, 9, 17]

Step 5

Now we need to encrypt the Plain text with dynamic key. To do so the formula is

Cipher Text (x) = (Plain Text (x) + Dynamic Key (x)) mod 27

Let's crack all,

Cipher Text (1) = (Plain Text (1) + Dynamic Key (1)) mod 27

= (22 + 6) mod 27

= 28 mod 27

= 1

Cipher Text (2) = (Plain Text (2) + Dynamic Key (2)) mod 27

= (1 + 5) mod 27

= 6 mod 27

= 6

Cipher Text (3) = (Plain Text (3) + Dynamic Key (3)) mod 27

= (12 + 10) mod 27

= 22 mod 27

= 22

Cipher Text (4) = (Plain Text (4) + Dynamic Key (4)) mod 27

= (21 + 5) mod 27

= 26 mod 27

= 26

Cipher Text (5) = (Plain Text (5) + Dynamic Key (5)) mod 27

= (5 + 9) mod 27

= 14 mod 27

= 1

Cipher Text (6) = (Plain Text (6) + Dynamic Key (6)) mod 27

= (19 + 17) mod 27

= 36 mod 27

= 9

Now we have

Encrypted data Numeric as = [1, 6, 22, 26, 14, 9]

Step 6

Converting the same into text we get = [A, F, V, Z, N, I]

Stage 3: Result of Encryption

The final Dynamic Key is ***F E J E I Q***

The final Encrypted Data or cipher text is ***A F V Z N I***

Stage 4: Decryption

Let's discuss the decryption technique used. As in above architecture we can see that we are using multilevel encryption so that security of data can be raised to peak. As every time the decryption follows following things:

1. Dynamic Key
2. Data or cipher Text
3. Technique or Algorithm
4. Plain Text

For better understanding the decryption module let's have a look at the unit module of Encryption.

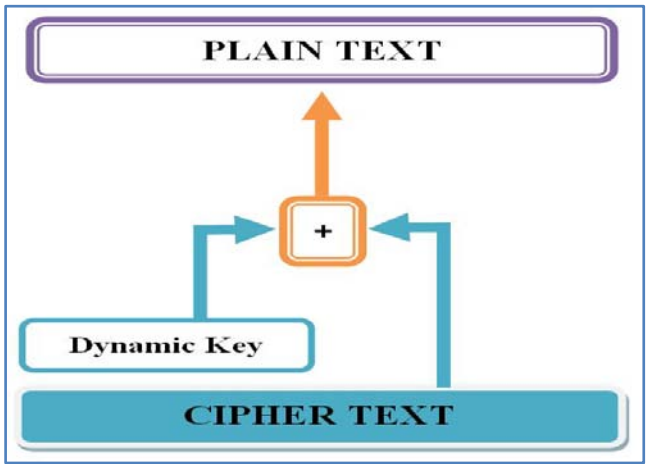


Figure 7. Decryption Process

Stage 5: Unit Decryption

As the scheme we are using follows a proper system of decryption so it also has an algorithm for the same.

As the result of encryption, we have the data as:
Dynamic Key = [F, E, J, E, I, Q]
Converted Dynamic Key = [6, 5, 10, 5, 9, 17]

Cipher text = [A, F, V, Z, N, I]
Converted Cipher text = [1, 6, 22, 26, 14, 9]

Now the decryption can be simply achieved by a formula as

Plain Text (x) = (Cipher text (x) – Dynamic key(x)) mod 27

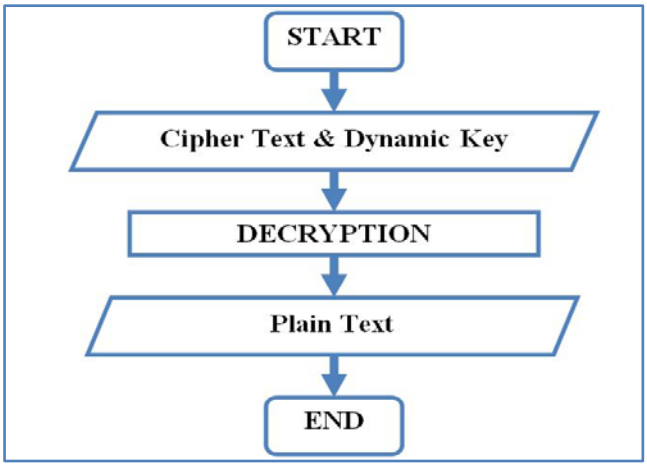


Figure 8. Reverse Unit Process

Let's solve for all values of x

Plain Text (1) = (Cipher text (1) – Dynamic key (1)) mod 27
= (1 – 6) mod 27
= (-5) mod 27
= 22
Plain Text (2) = (Cipher text (2) – Dynamic key (2)) mod 27
= (6 – 5) mod 27
= 1 mod 27
= 1
Plain Text (3) = (Cipher text (3) – Dynamic key (3)) mod 27
= (22 – 10) mod 27
= 12 mod 27
= 12
Plain Text (4) = (Cipher text (4) – Dynamic key (4)) mod 27
= (26 – 5) mod 27
= 21 mod 27
= 21
Plain Text (5) = (Cipher text (5) – Dynamic key (5)) mod 27
= (14 – 9) mod 27
= 5 mod 27
= 5
Plain Text (6) = (Cipher text (6) – Dynamic key (6)) mod 27
= (9 – 1) mod 27
= (-8) mod 27
= 19

As result, we got
Plain text as [22, 1, 12, 21, 5, 19]
Converted Output is like ***V A L U E S***

3.2. Benefits of Proposed Method

The following are the advantages of proposed system over previous techniques

1. The proposed encryption technique is a free length encryption and after a length of 676 it can be used as block encryption technique.
2. The proposed encryption technique follows symmetric encryption method, so encrypting the plain text is as same easy as decrypting the cipher Text.
3. If any attacker tries to attack the privacy through hit and trial then he will take ~5 x 10^{18} Years to decrypt the cipher text.
4. The proposed encryption technique is easy to implement so it takes less time for encryption and decryption.

5. All the previous methods based on one time encryption of data but as you can see the architecture of the system there are multilevel encryption so that higher security can be raised over data.
6. The proposed method of aggregation is more secure as compared to al the previous proposed methods.
7. The proposed encryption technique used a low-level computation, so that implementation of this is easy.
8. The proposed system implementation is cost effective as has low computational work so fewer resources will be required to implement.
9. The proposed encryption technique provides an easy to use and implement environment as well as having a tough security.
10. The proposed encryption techniques offer maximum number of possible cases of plain text as compared to all past data because of use of multi-level encryption and multi-layer of data for encryption.
11. The proposed encryption techniques can work with any length of key hence for long text the key can be repeated in a queue.
12. The proposed method is also suitable for the system having low budget, High security issues, Low computation power and having processing limits.
13. The system where we don't have limits over computation, Time, Cost, processing we can simply create recursive encryption by using the proposed architecture.

3.3. Comparison

Here we will compare the proposed encryption technique to all previous technique available till time

Table 1. Comparison of Proposed technique with existing techniques

Factors	RSA	DES	AES	Proposed Aggregation System
Time	Very High	High	Average	Average
Memory Consumption	Very High	High	Average	Low
Computation Need	High	High	Average	Low
Processing Need	High	High	Average	Low
Input Length	128 Bits	128 Bits	128 Bits	Any Size
Key Length	128/192/256	128/192/256	128/192/256	Any Size
Design Complexity	Difficult	Difficult	Difficult	Easy
Implementation	Difficult	Difficult	Difficult	Easy
Cost	High	High	Average	Very Low
Security	High	High	Average	Comparatively Highest
Cipher Type	Symmetric	Symmetric	Symmetric	Symmetric & Vigenere
Possible Combination	2^128	2^128	2^128	2^512 x 2^(256*4)
Time to crack all possible keys	1.02 x 10^18 Yrs	1.02 x 10^18 Yrs	1.02 x 10^18 Yrs	4.08 x 10^18 Ys

IV. EXPERIMENTAL RESULTS

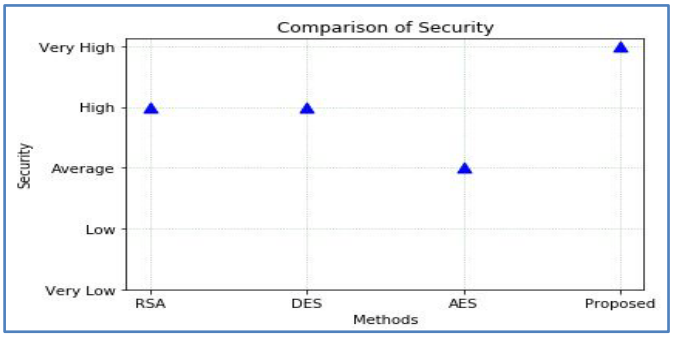
The result analysis of proposed method among various existing method such as RSA, DES and AES are done using metrics of measurement such as security, computational complexity, implementation complexity, design complexity, memory consumption and processing speed. Here, high, low, average, very high and very low operators is used for as a value of the proposed and exiting method which is depicted in table and through graph.

4.1. Security Analysis

In this section, we do the analysis of security parameter between proposed and exiting method which is shown in table 2 and comparative analysis is shown through graph 1. After analysis of security parameter, it is analyzed that our proposed method has very high security than the other existing security method.

Table 2. Security comparison of proposed and existing method

S. No.	Models	Security
1	RSA	High
2	DES	High
3	AES	Average
4	Proposed	Very High



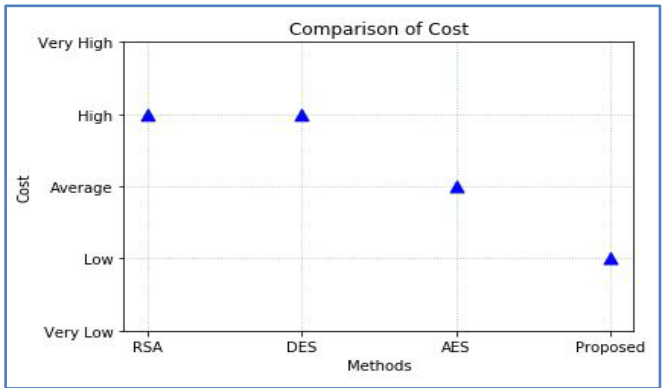
Graph 1. Graphical representation of Security analysis

4.2. Computational Cost Analysis

Here, we do the analysis of computation cost parameter between proposed and exiting method which is shown in table 3 and comparative analysis is shown through graph 2. Computational cost of modular exponentiation and multiplication operations is much higher than that of hash functions and addition operations; so, we will ignore the cost of hash operations and addition operations, and only focus on the computational cost incurred by encryption and decryption operations. After analysis of computational cost parameter, it is analysed that our proposed method has very low computational cost than the other existing security method.

Table 3. Computational cost comparison of proposed and existing method

S. No.	Models	Cost
1	RSA	High
2	DES	High
3	AES	Average
4	Proposed	Very Low



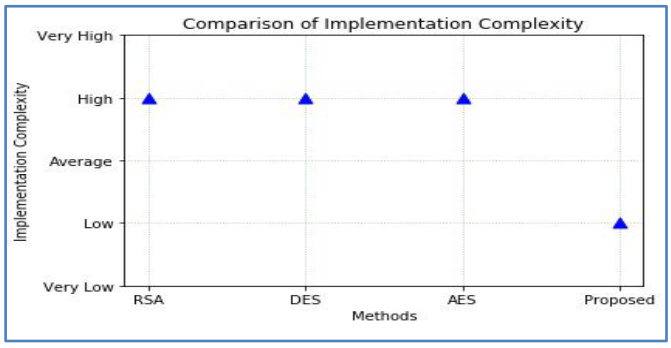
Graph 2. Graphical representation of computational cost analysis

4.3. Implementation Complexity Analysis

Here, we do the analysis of implementation complexity parameter between proposed and exiting method which is shown in table 4 and comparative analysis is shown through graph 3. After analysis of implementation complexity parameter, it is analysed that our proposed method has low implementation complexity than the other existing security method.

Table 4. Implementation complexity comparison of proposed and existing method

S. No.	Models	Implementation Complexity
1	RSA	High
2	DES	High
3	AES	High
4	Proposed	Low



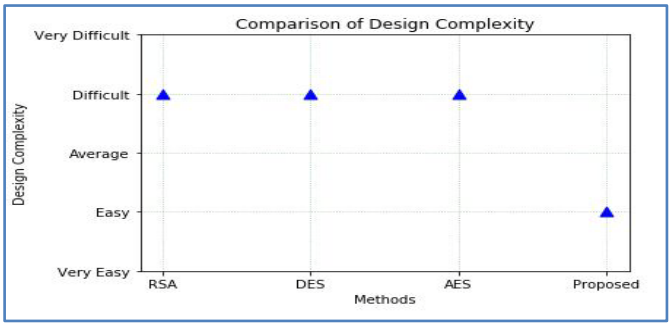
Graph 3. Graphical representation of Implementation complexity analysis

4.4. Design Complexity Analysis

Here, we do the analysis of design complexity parameter between proposed and exiting method which is shown in table 5 and comparative analysis is shown through graph 4. After analysis of design complexity parameter, it is analysed that our proposed method has low design complexity than the other existing security method.

Table 5. Design complexity comparison of proposed and existing method

S. No.	Models	Design Complexity
1	RSA	High
2	DES	High
3	AES	High
4	Proposed	Low



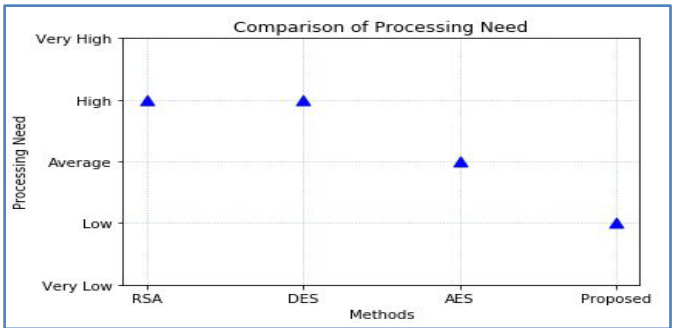
Graph 4. Graphical representation of Design complexity analysis

4.5. Processing need Analysis

Here, we do the analysis of processing need parameter between proposed and exiting method which is shown in table 6 and comparative analysis is shown through graph 5. After analysis of processing need parameter, it is analysed that our proposed method has low processing need than the other existing security method.

Table 6. Processing need comparison of proposed and existing method

S. No.	Models	Processing Need
1	RSA	High
2	DES	High
3	AES	Average
4	Proposed	Low



Graph 5. Graphical representation of processing need analysis

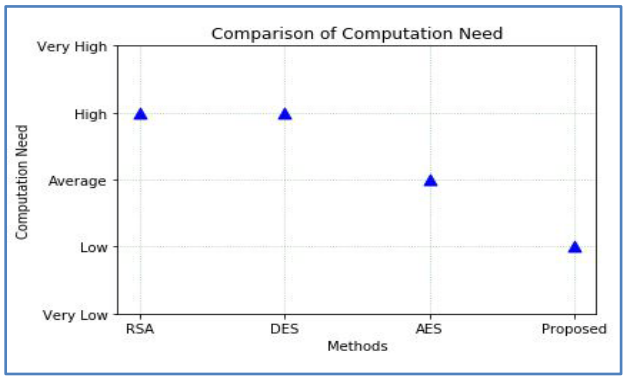
4.6. Computational Need Analysis

Here, we do the analysis of computational need parameter between proposed and exiting method which is shown in

table 7 and comparative analysis is shown through graph 6. After analysis of computational need parameter, it is analysed that our proposed method has low computational need than the other existing security method.

Table 7. Computational Speed comparison of proposed and existing method

S. No.	Models	Computation Need
1	RSA	High
2	DES	High
3	AES	Average
4	Proposed	Low



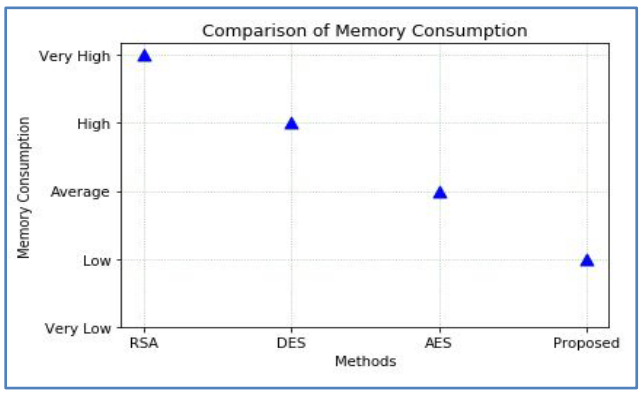
Graph 6. Graphical representation of computational need analysis

4.7. Memory Consumption Analysis

Here, we do the analysis of memory consumption parameter between proposed and exiting method which is shown in table 8 and comparative analysis is shown through graph 7. After analysis of memory consumption parameter, it is analysed that our proposed method has low memory consumption than the other existing security method.

Table 8. Memory consumption comparison of proposed and existing method

S. No.	Models	Memory Consumption
1	RSA	Very High
2	DES	High
3	AES	Average
4	Proposed	Low



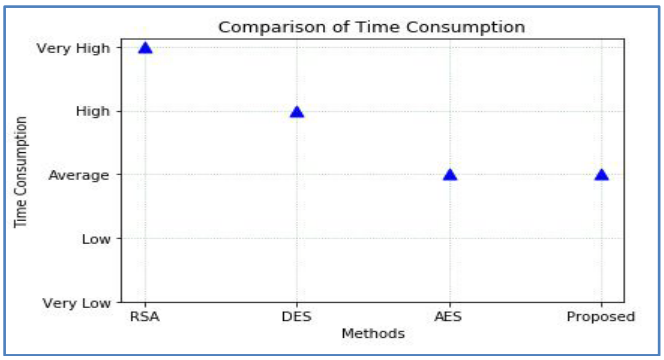
Graph 7. Graphical representation of memory consumption analysis

4.8. Time Consumption Analysis

Here, we do the analysis of time consumption parameter between proposed and exiting method which is shown in table 9 and comparative analysis is shown through graph 8. After analysis of time consumption parameter, it is analysed that our proposed method has average time consumption than the other existing security method.

Table 9. Time consumption comparison of proposed and existing method

S. No.	Models	Time Consumption
1	RSA	Very High
2	DES	High
3	AES	Average
4	Proposed	Average



Graph 8. Graphical representation of time consumption analysis

V. CONCLUSION

Our daily life is being reformed with Internet of Things (IoT) by connecting different gap between physical and digital world/internet. To empower real-time processing, seamless connection and ubiquitous sensing in IoT applications, privacy preserving data aggregation method is considered as a basic element, that utilize storage and computing resources to network edges. For providing security and privacy to IoT devices various security scheme has been developed but in this we design a novel security architecture which is fast and cost efficient than other security scheme. The evaluation of proposed architecture is done using the different parameters namely computation cost, implementation complexity, memory consumption, design complexity, processing need, computation need, computational cost etc and it is compared with the exiting security method RSA, DES and AES. After analysis of these security method it is analyzed that our proposed method requires less computational cost, consume less memory and the design and implementation complexity is also very less. Similarly, the proposed method also takes very less processing need and computational need which enhance IoT device efficiency. This proposed method uses multilayer encryption along with multilayer of data for improving IoT device's security. Confidentiality and integrity of usage data is ensured by using multilayer encryption which prevent the network from internal attack, external attack and collusion attack. The proposed method creates recursive encryption

which greatly enhances the security of data sharing through the network and IoT devices.

VI. FUTURE WORK

Effort can be made to discover a novel approach to decrease data exchange rate when confusing users' data among different IoT device. In future work, a comprehensive method is to be proposed under collusive attack model which will aim to accomplish full phase privacy and reliability.

In future we will also evaluate our proposed method in certain properties like fault tolerance, multi-functional and supporting multi-dimensional data. For now we also aim in expanding this approach and its deployment in different real-world IoT application.

VII. REFERENCES

[1] Evans D, "The Internet of Things - How the Next Evolution of the Internet Is Changing Everything", CISCO white paper 2011.

[2] David K, Jefferies N, "Wireless visions: A look to the future by the fellows of the wwrf", Vehicular Technology Magazine, IEEE dec 2012; 7(4):26–36, doi:10.1109/MVT.2012.2218433.

[3] Mattern F, Floerkemeier C, "From active data management to event-based systems and more", Springer-Verlag, 2010.

[4] Presser M, Krco Sa, "IOT-I: Internet of Things Initiative: Public Deliverables – D2.1: Initial report on IoT applications of strategic interest 2010".

[5] Atzori L, Iera A, Morabito G, "The Internet of Things: A survey. Computer Networks", 2010; 54(15):2787 – 2805, doi:10.1016/j.comnet.2010.05.010.

[6] J. H. Ziegeldorf, O. Garcia Morchon, and K. Wehrle, "Privacy in the Internet of Things: threats and challenges", Security and Communication Networks 7.12 (2014): 2728-2742.

[7] "Benetton to Tag 15 Million Items", RFID Journal. http://bit.ly/XXe4Wi, 2003.

[8] Albrecht K. Boycott Benetton, "No RFID tracking chips in clothing! Press Release", http://bit.ly/49yTca, Sep 2003.

[9] Cuijpers C, "No to mandatory smart metering does not equal privacy!", Tilburg Institute for Law, Technology, and Society: Webblog 2009.

[10] "The INDECT Consortium. INDECT project", http://www.indect-project.eu/, 2009.

[11] Munch V, "STOPP INDECT", http://www.stopp-indect.info, 2012.

[12] Feyza Yildirim Okay et al., "Fog computing-based privacy preserving data aggregation protocols", Trans Emerging Tel Tech. 2020; e3900. wileyonlinelibrary.com/journal/ett © 2020 John Wiley & Sons, Ltd. 1 of 23 https://doi.org/10.1002/ett.3900.

[13] Mengyao Zheng et al., "Challenges of Privacy Preserving Machine Learning in IoT", AIChallengeIoT'19, November 10–13, 2019, New York, NY, USA c 2019 Association for Computing Machinery. ACM ISBN 978-1-4503-7013-4/19/11. . . $15.00 https://doi.org/10.1145/3363347.3363357.

[14] Prem Prakash Jayaraman et al., "Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation", http://dx.doi.org/10.1016/j.future.2017.03.0010167-739X/© 2017 Published by Elsevier B.V.

[15] Yuwen Pu et al., "Two Secure Privacy-Preserving Data Aggregation Schemes for IoT", Hindawi Wireless Communications and Mobile Computing Volume 2019, Article ID 3985232, 11 pages https://doi.org/ 10.1155/2019/3985232.

[16] Rongxing Lu et al., "A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT", Article in IEEE Access · March 2017 DOI: 10.1109/ACCESS.2017.2677520.

[17] Inayat Ali et al., "Privacy-Presereving data aggregation in Resource-Constrained Sensor Nodes in Internet of Things: A Review", DOI: 10.1016/j.fcij.2017.11.004.

[18] Chunqiang Hu et al., "An Efficient Privacy-Preserving Data Aggregation Scheme for IoT", DOI: 10.1007/978-3-319-94268-1_14.

