

Message Based Key Distribution Technique for Establishing a Secure Communication Channel in IoT Networks

G.V.Hindumathi

Computer Science & Engineering Department, JNTUK, Kakinada
E-mail: hindu.gundala@gmail.com

Dr. D. Lalitha Bhaskari

Computer Science and Systems Engineering, Andhra University, Visakhapatnam
E-mail: lalithabhaskari@yahoo.co.in

Received: 14 July 2019; Accepted: 24 July 2019; Published: 08 November 2019

Abstract—Internets of Things (IoT) are distinguished by different devices, which support the ability to provide innovative services in various applications. The main aspects of security which involves maintaining confidentiality and authentication of data, integrity within the IoT network, privacy and trust among IoT devices are important issues to be addressed. Conventional security policies cannot be used directly to IoT devices due to the limitation of memory and high power consumption factors. One of the security breaches in the intranet is lack of encryption due to the IoT devices infrastructure. The basic IoT devices are 8-bit, low-cost, limited memory and power consumption devices which limit the complex algorithm execution. The key distribution is another major challenge in IoT network.

This paper proposes a solution to transmitting messages by adopting Random Number generation and distribution of session key for every message without any difficulty. It gives better result to resist from the brute force attack in a network.

Index Terms—Key distribution, Random key generation, Seed value, Session Key, AES algorithm.

I. INTRODUCTION

Major security challenge in the IoT network is an increase of overall attacks, as compared to isolated (i.e. non-connected) systems. Upcoming years will be witnessing the mass production of IoT devices due to their wide applications and cost-effectiveness. Many institutions and organizations show interest to develop and adapt the IoT devices at a greater volume, however, the major constrain is needed to look after the security issues.

This is evidenced by a study done by Hewlett Packard (HP), where it was shown that more than 70% of devices connected to the cloud are susceptible to serious attacks due to lack of proper security measurements[1]. This may

be due to following factors.

- Lack of improper transport encryption: Since several IoT devices are simple “uni-taskers” and have limitations in memory size, power processing as well. The proper encryption of communication among IoT devices is a vital element for cloud adaptability. For example, with improper encryption, the function of 8-bit microcontroller, uses to control remotely located electronic gadgets are substandard to reach SSL (Secure Socket Layer).
- Insufficient authentication and authorization: Due to inadequate authentication, password requirements (secure, periodical resets etc.) and failure to re-authenticate at several stages, these IoT devices have weak authentication and compromised security in the cloud [2].
- Key compromising: Attackers identifies a key in a network through brute force attack then all transactions between sender and receiver leads to compromise.

In the entire network, the IoT devices connect physically and interact with many devices like virtual things and human resources. In order to develop the healthy interaction among the devices, effective security is needed. In case of compromised security, the entire network gets affected and the devices lack communication with each other.

Moreover, protecting the IoT is hard and intricate. The security should be implemented to prevent attacks on IoT devices which target physical threats, diverse communication channels, denial of service, identity fabrication and so on [3]. Eventually, the multiple heterogeneous IoT devices located at diverse environment can exchange information with each other which further complicates the design and deployment of efficient, interoperable and scalable security mechanisms [4].

Simple encryption technology is needed to implement in IoT cryptographic algorithm. The IoT network may have severe security issues, like man-in-the-middle attack constantly counterfeit attacks in the network layer. These attacks can take and send fake information to insecure IoT nodes in the network. Identity authentication and data confidentiality mechanism are mainly used to prevent the secure data access by unauthorized nodes [5].

Several studies elucidated about the measures to be taken in security mechanisms in addition to the earlier mentioned issues [6]. Heterogeneity has a great influence on the protocol and network security services that must be employed in the IoT. Constrained devices will interact with various heterogeneous devices (e.g. other constrained devices, full-fledged web servers) either directly or through gateways.

However, it is essential to employ efficient cryptographic algorithms which can provide a high throughput even in 8 bit and 16-bit entities, as well as they, need to implement simple security algorithms which provide an end-to-end secure communication channel. These protocols require credentials, thus optimal key management systems must be implemented to distribute these credentials and to help in establishing the necessary session keys between nodes [2].

II. RELATED WORKS

Majority of the IoT devices do not encrypt communications when connect to the cloud and local network [1]. Since IoT devices collect and transmit confidential data, encryption is essential while transmitting data over IoT network [7]. Of IoT devices, 8-bit microcontroller devices share majority proportion which is uni-taskers and have size and power processing constraints. Due to these constraint factors the simple IoT devices unable to support the required processing power for secure communications in intranet as well as internet networks [2].

Large computations and huge memory capacity are the general requirements for mainframe security which are the major challenges to implement them in IoT devices. Generally, in a real-time network pool to get quick response, IoT devices should run self-healing architecture which consumes large memory [8].

Management of energy in the sensor nodes is imperative since they are powered by battery systems. The IoT nodes in the network platform should get support from the MAC layer protocols for regulating duty cycle and network layer protocol for data aggregation designs and multipoint to point transmission. IoT devices have limited energy sources to perform communication between different nodes in the network. The fundamental challenge in the IoT communication is connecting the different energy nodes by optimizing their limited energy source [9]. Furthermore; the persistent problem to connect IoT devices in the network is security attacks. Since most IoT devices connected in wireless have more security issues due to limited physical

accessibility to sensors, actuators and openness of the systems in the network pool, transient and permanent random failures are very common which can be exploited by hackers. To have a long life, the IoT devices should recover from security attacks spontaneously. The IoT devices should be able to prevent unanticipated attacks by the hackers.

Current adapted low capacity IoT devices fail to recover from security attacks. The well-framed systems in the cloud need to detect the attack, diagnose the attack, and deploy countermeasures and repair, whereas this is a challenge for IoT devices due to their low storage capacity and processing constraint [10]. It is essential to maintain significant hardware support [11] for providing encryption, authentication, and digital signature attestation and secure keys. Even if new devices are security-aware, dealing with legacy devices will prove difficult [8]. Despite the availability of a large number of standard encryption technologies the major challenge is to design or develop encryption algorithms which are fast and with less energy consumption [12].

Now we are discussing about the different existing techniques for above mentioned problems.

A. Using Proper Secure channel

IoT devices should be able to negotiate the real parameters like algorithms, strength and protection mechanisms while opening a secure channel. Usually, certain configurations cannot be executed by these restrained devices as well as adapting the devices to the network may be the major disadvantage. Entire data essentially may not be needed to a strong protection during data transmission flow. In addition, analysis of essentiality of a number of security protocols implementation in smooth workflow environment connected to critical IoT devices is necessary. Indeed, it is essential to observe whether the present internet protocols can be employed in these devices. Lastly, a secure path sight is important on the incoming connections to IoT devices. If this secure channel is compromised with the intruders then entire secure data will be lost.

B. Using Data encryption standards

The physical layer network needs to be protected by a potent cryptographic algorithm to ensure the security of the entire network service. In addition, data encryption is highly essential for the wide application areas of a wireless sensor network. Since computing power and storage space of IoT nodes in the network are limited the energy consumption and adaptation of complex asymmetric encryption algorithms for computation makes them difficult to apply in wireless sensor networks. Generally, data encryption algorithms are classified into two categories symmetric encryption algorithm and public key encryption algorithms. However, in wireless communications, the symmetric encryption algorithm is intensively used due to its simple computation methods [9].

C. Using proper key distribution mechanisms:

Extensive studies need to be performed on the key management issues for the wireless network. Key management includes distribution, secret key generation, storage, updating, and destruction process [13].

Basic symmetric key pre-distribution schemes are probabilistic key distribution, deterministic key distribution [14]

1. Probabilistic key distribution: The Random Key Protocol (RKP) includes two different stages, namely the secret key identification and formation of the path to send the key. Random Keys are gathered into a single group to generate the secret keys. Then keys are randomly selected by the node from this key pool and transmitted to different sensor nodes. While broadcasting, certainly some probability may occur to share the common key for two or more sensor nodes. The second stage is useful for protecting the common key without share to two or more sensor nodes in the network. Here, an IoT device sends the common key to its neighbors using previously built trust management process and secure channel. This procedure continues till the common key reaches to the actual destination node. The disadvantage of this method is each and every node in the network must trust its neighbors.

2. Deterministic key distribution: The deterministic key distribution again divided into two types, Offline key distribution and Server-based key distribution which are discussed below.

Offline key distribution: Since the offline key distribution process is simple and easy to share, generally it is used in wireless sensors networks. Based on this protocol the same network key is distributed to all sensor nodes in the network or common key may get shared between two nodes for secure communication. In absence of the third-party, nodes generate the same session key after some period of time or few transactions between those nodes. The advantage of this method is less power

consumption and need not require complex encryption algorithms. The disadvantage of this process is the physical sensor node attack. Whenever the node is compromised then entire secret information is revealed.

Server-based key distribution: In this context Key Distribution Centre (KDC) is used to distribute the session keys among sensor nodes in the network. All the nodes in the network trust only KDC rather than all their neighbors. The disadvantage of this method is an establishment secure channel between KDC and node that needs another shared key to distribute the session key.

Symmetric encryption may have the following issues:

- The complexity of key exchange protocol based on the symmetric cryptosystem for a wireless sensor network is very high.
- Prudent confidentiality problem key: In wireless sensor network IoT nodes are exposed to direct attacks. If the IoT node is compromised it may cause a serious security breach in the network.
- Weak digital signatures and message authentication: Conventional message authentication code used in symmetric encryption algorithm occupies lots of storage space and causes additional power consumption.

III. PROPOSED METHODOLOGY

In this study we developed key distribution schemes. So we tried to implement a novel method to distribute the key for security. As illustrated in Fig.1, our scheme is composed of three phases, namely "System setup," "Key agreement phase," and "Secure data transmission phase". Table 1, Gives the notations used in our proposed system.

The proposed system comprises of 32 bit data. For data larger than 32 bit, the data must be divided in chunks of 32 bit. A RKG function generates a 32 bit key.

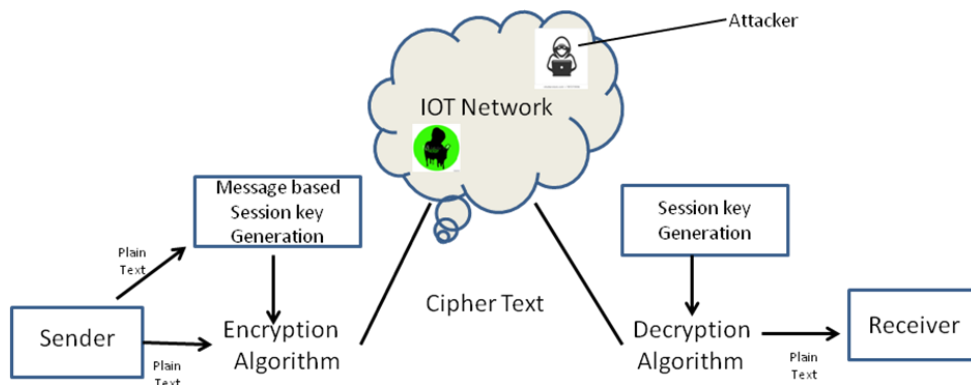


Fig.1. Proposed Architecture

A. System Setup

The hardware is basic IoT controller i.e., an ATmega328P. ATmega328P controllers supported IDE's,

those also an open- source platform for testing real-time embedded systems. In this work, an ATmega328P is an 8-bit microcontroller with 32KB flashes memory (about 2KB reserved for the boot loader), 2KB SRAM (Static

RAM) is used to implement this algorithm. The reason for choosing this 8-bit controller is because of its low power consumption which is required for all battery powered IoT devices. Here we are using same types of two ATmega328P controllers connected with Wi-Fi modules to implement the proposed system.

Table 1. Notations

Notation	Definition
K_s	Session key
SV	Seed Value
SIOT	Sender's IOT device
RIOT	Receiver's IOT Device
RKG	Random Number Generator for 32 bit
	The Concatenation Operator
\oplus	The Exclusive Operator
AES	Advanced Encryption Standard
PT	24 bit Plain Text
CT	32 bit Cipher Text
X	LSB byte of Plain Text
Y	MSB byte of Plain Text
CPT	Concatenated Plain Text
a,c	16 bit Constants
X_n	Previous Key Value
n* Function	Function executes n times

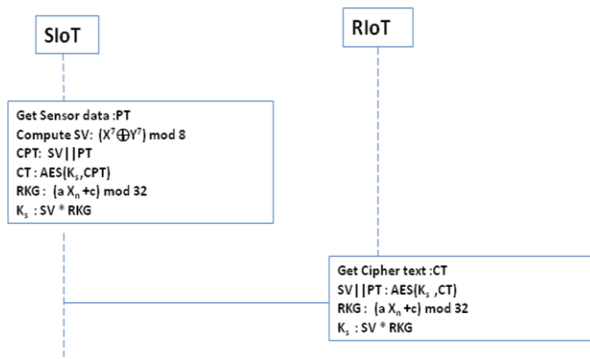


Fig. 2. Key Agreement phase and secure data transmission phase

B. Key Agreement Phase

The first Key needs to be sent through a secure transmission channel to the receiver by the corresponding sender. In this paper gives a solution to change this key

for every message to encrypting or decrypting. In this phase sender, sends the relevant information to calculate the key at receiver side through sender's data.

Step1: Fig.2 shows sender's (SIoT) and receiver's (RIoT) processes. The First step proposes the message based key generation at sender side. The current system proposes a session key for every message.

Step 2: Computes the Seed Value (SV)

$$SV = (X^m \oplus Y^n) \bmod 8 \tag{1}$$

Where X, Y are LSB, MSB Bytes of plaintext respectively and m,n is chosen as 7. The complexity of Seed value generation function may vary with powers of X and Y.

Step 3: Concatenates SV and PT to store the result in CPT. Sends the CPT in encrypted format to receiver through a transmission channel. After decryption of the cipher text receiver computes the Session key.

$$K_s = SV * RKG \tag{2}$$

Here RKG is Random Key Generation Function. This K_s is used for decryption of data coming in next round.

Secure data transmission Phase:

Step 1: In Fig.2, at sender side a complex AES encryption algorithm used for encrypting the concatenating data given by the step1. Here we used previous stored session key K_s for encrypting the data.

Step 2: Destination gets the cipher text through insecure channel, and decrypt the CT by using previously stored K_s . After decrypting the text by receiver, it gets seed value for input of RKG function to get the next session key.

C. Key generation and Encryption Process

Majority of the IoT devices do not have the secure data in intranet communications due to power and memory constraints. In the proposed approach, key- management plays a vital role to send the data securely. In this regard, mutation in the key for every message transfer gives a better solution to security in IoT Network.

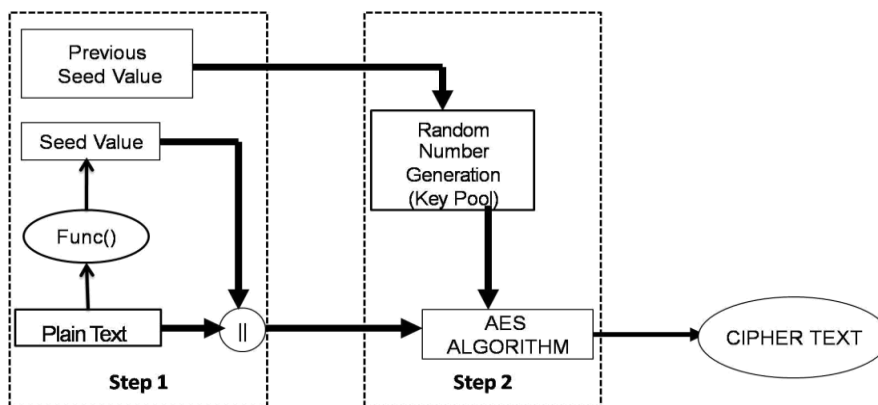


Fig.3. Key generation and Encryption processes

Fig.3 represents the schematic diagram of the Key Generation and Encryption process. This process has following two major steps.

The Algorithm 1 explains the whole encryption process with different stages. The IoT node gets the raw data from the different sensors connected to it. This raw data is called as plain text for IoT device to encrypt. Since the security measures are absent in the intranet network the plain text of IoT devices is mere compromised. In the proposed approach, we append the seed value to provide iterations for the random key generator to original plain text. It gives the information about the key for next encryption/decryption process. This seed value computes through a function that calculates based on the plain text content. IoT device gets the information from its sensor device, which is applied to the above function. Then it generates a secret seed value for calculating the next key. As well as Step1 gives the details about previous seed value. The IoT device stores the information of previous seed as an input for RKG.

Algorithm 1: Key generation and Encryption Process

Input: Plain Text, Previous seed value, RKG function and AES encryption algorithm

Output: Cipher Text

Begin

Step1: Compute the Seed Value using equation (1).

This seed value concatenate with plain text.

Step2: To get a secret key, execute RKG function with inputs of previous seed value and a ,c constant values with equation (2).Then implement AES algorithm with generated secret key and appended plain text.

End

The random key generator is playing a major role in this step. It gets the previous seed value from the step1

and stored it in an input variable. Applying above all inputs to that the RKG function that returns a key value with seed number of iterations. Implementation of Encryption algorithm with the Step1 results in a new secret key from RKG function. It produces a cipher text, which transfers to another IoT device that knows all the inputs.

D. Key Generation and Decryption Process

End to end communication can be possible with this approach in a secure format. Receivers of IoT devices should also be an 8-bit controller device to make this process easier to compute.

Fig.4 explains the schematic diagram of the Key Generation and Decryption process of this approach. This process can be performed in three steps for computing the plain text with the cipher text received from internet devices. The Algorithm2 gives the different steps of decryption process.

Algorithm 2: Key Generation and Decryption Process

Input: Cipher Text, Previous Seed value, RKG function

Output: Plain Text, Seed value for next secret key

Begin

Step1: Receives the cipher text value from sender and get the stored seed value (Previous).

Step2: To get the secret key for decryption, Execute RKG function with an input of previous seed value and a,c constant values Implement AES algorithm to generate plain text of given cipher text through secret key.

Step3: Output of a step2 is plain text and seed value. This seed value must be stored for next Iteration of Algorithm2.

End

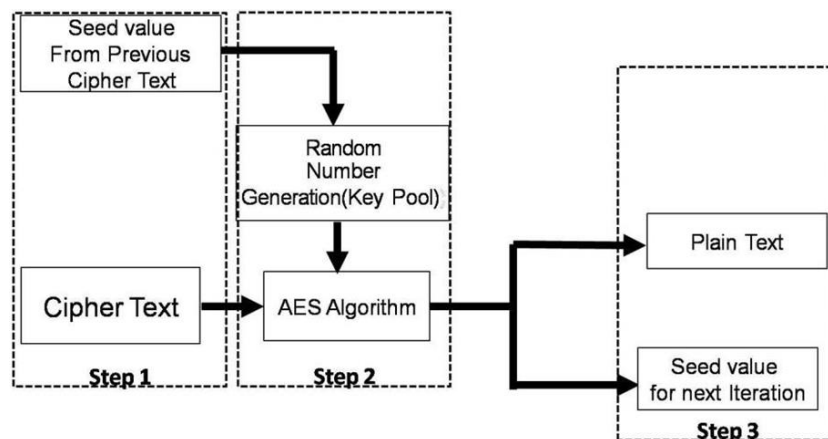


Fig.4. Key Generation and Decryption Processes

Receiver IoT device also stores the previous seed value received from the previous cipher text. This will helpful for a secret key generation for decrypting the

message. And Receiver receives cipher text from cloud/internet devices.

Previous seed value and other inputs (a,c) are applied

to RKG function. Using those mentioned inputs RKG generates a new key but seed value gives information about iterations of RKG function. Then this secret key will be applied for decryption algorithm using the cipher text.

IV. EXPERIMENTAL RESULTS

Observe with multiple plaintexts and compute the time complexity also for both encryption and decryption.

A. Random keys

Random Key generator (RKG) generates 32-bit key values randomly. New keys are independent compared to previous key values. Fig.5 describes the multiple random keys from the RKG. The RKG was used to generate a new secret key to encrypt and/or decrypt for a given plain text.

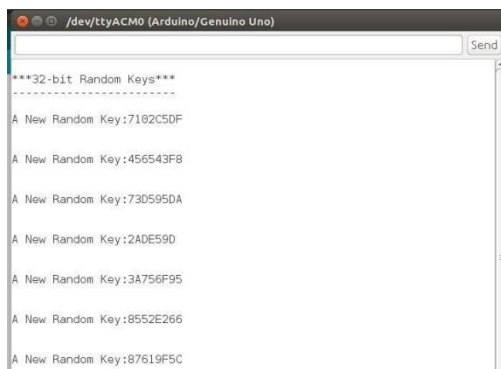


Fig.5. Multiple Random Keys

B. Side process

AES algorithm is used as an encryption process. Fig.6 gives the whole information about sender side.



Fig.6. Encryption Process

1. Encryption time: It provides encryption time complexity in milliseconds for conversion from plain text to cipher text.
2. PLAIN: It is again divided into two parts. The first part consists of seed value to generate the key using RKG. And second part gives the information about plain text given by the sensor node.
3. CIPHER: Gets the cipher text after execution of the Encryption algorithm. It is transmitted to receiver IoT device.

4. Initial vector is known by both senders as well as a receiver.

C. Receiver side process

Fig.7 describes the following process.

1. Decryption Time: It provides decryption time complexity in milliseconds for conversion from cipher text to plain text.
2. CIPHER: Receiver got the cipher text from the source.
3. PLAIN: Applying random secret key and cipher text as inputs to AES algorithm. Then it provides a seed value as a first part of the text and second part is original plain text.
4. Initial Vector for AES algorithm. The Initial vector is common for both sender as well as the receiver.

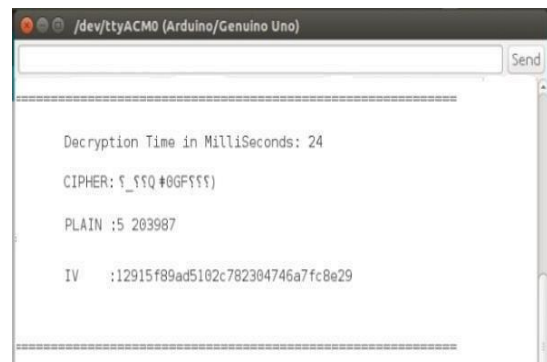


Fig.7. Decryption Process

V. SECURITY ANALYSIS

So many types of attacks are attacked during transmission channel. One of the attacks is cipher text only attack. Cryptanalysis knows the cipher text only sent through transmission channel. She/he (attacker) may identify the different patterns of cipher texts with different plain texts if sender uses same key for entire transmission data. It is verify difficult to identify the plain text directly without knowing the keys. Another attack is brute force attack.

Brute force attack: An attacker checks for every possibility to identify the key to decrypt the encrypted text.

Table 2. Exponential growth of a brute force attack

Password Length	Time to Brute Force attack
24 bit ASCII	12 hrs
32 bit ASCII	48 hrs
40 bit ASCII	7 months
48 bit ASCII	51 years
54 bit ASCII	682 years

Here, an attacker tries to identify every session key in

transmission using each possibility. Table 2 gives, generalized Time to crack the key by an attacker using Arduino.

Here, as an attacker we implemented Brute force algorithm to identify the key for equivalent plain text of known cipher text.

Fig.8 and Fig.9 explains the time to crack the different cipher texts.

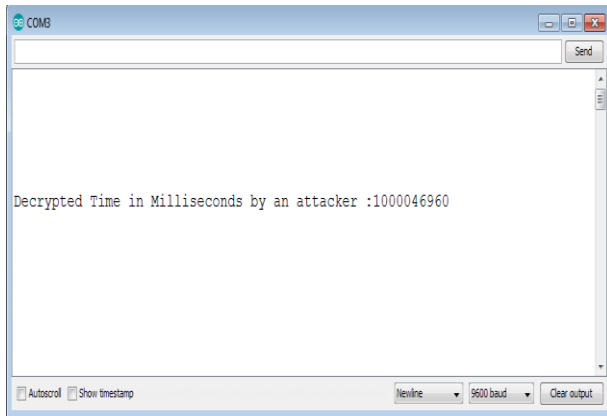


Fig.8. Time to crack the cipher text1

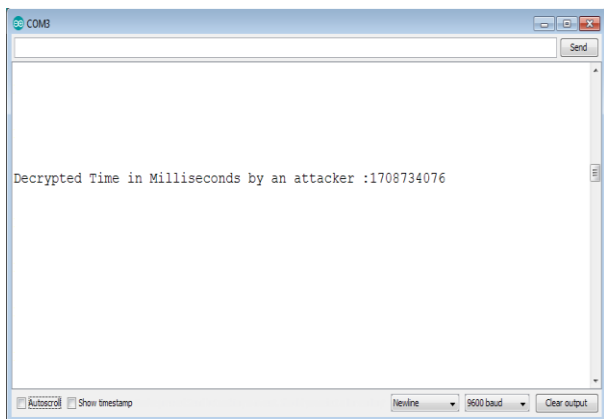


Fig.9. Time to crack the cipher text2

Here, we were trying to crack the multiple cipher texts and we got different time variations.

In this paper we are discussing two different scenarios

1. Cipher text only attack: An attacker can't be identified the plain text of relevant encrypted text because of unknown key.
2. Brute force attack: In our proposed system changes the secret key for every message. If, identifies the key by an attacker, but cannot decrypt every message transmitted by a sender with that same key. The attacker must compute the key for every message rather than once for entire transaction between source and destination.

VI. CONCLUSION AND FUTURE DIRECTION

The next generation computers are completely interconnected to the IoT devices. The overall security

measures are imperative for the development of technologies from the IoT context. In our present study, we proposed a novel approach to solve the issues of security breach by using a random key generation and to provide key distribution that establishes secure channel among IoT devices. Symmetric encryption algorithms are mostly used in IoT devices because these are lightweight algorithms. Hence only a single key is enough to compute the Encryption/Decryption process. In this approach, the generated key frequently changes (Session key) for every transmitted message. Hence an attacker can't access the key using brute force attack concept because the key size is 32 bit wide. Limitations for proposed work are use only 32 bit key, 32 bit data and 32 bit encryption techniques.

However further studies are needed to implement more strong security in IoT network. In this proposed method the main limitation is, we did not include a signature (Authentication) part for an IoT device participated in the network. This is a major challenge in security measures while the destined IoT device can't identify the authentication of sender IoT device. And also the key length increases for secure transmission. Hence our current study supports strong confidentiality for security issues.

REFERENCES

- [1] Newsletter, Lack of security in the Internet of Things devices. Study by HP (https://link.springer.com/chapter/10.1007/978-3-319-50758-3_3)
- [2] Weber, Rolf H., and Evelyne Studer. "Cybersecurity in the Internet of Things: Legal aspects." *Computer Law & Security Review* 32, no. 5 (2016): 715-728.
- [3] Babar, Sachin, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, and Ramjee Prasad. "Proposed security model and threat taxonomy for the Internet of Things (IoT)." In *International Conference on Network Security and Applications*, pp. 420-429. Springer, Berlin, Heidelberg, 2010.
- [4] Roman, Rodrigo, Jianying Zhou, and Javier Lopez. "On the features and challenges of security and privacy in distributed internet of things." *Computer Networks* 57, no. 10 (2013): 2266-2279.
- [5] Alaba, Fadele Ayotunde, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. "Internet of Things security: A survey." *Journal of Network and Computer Applications* 88 (2017): 10-28.
- [6] Roman, Rodrigo, Pablo Najera, and Javier Lopez. "Securing the internet of things." *Computer* 44, no. 9 (2011): 51-58.
- [7] Airehrou, David, Jairo Gutierrez, and Sayan Kumar Ray. "Secure routing for internet of things: A survey." *Journal of Network and Computer Applications* 66 (2016): 198-213.
- [8] Stankovic, John A. "Research directions for the internet of things." *IEEE Internet of Things Journal* 1, no. 1 (2014): 3-9.
- [9] Jing, Qi, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. "Security of the Internet of Things: perspectives and challenges." *Wireless Networks* 20, no. 8 (2014): 2481-2501.
- [10] Wood, Anthony D., Lei Fang, John A. Stankovic, and Tian He. "SIGF: a family of configurable, secure routing

protocols for wireless sensor networks." In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pp. 35-48. ACM, 2006.

- [11] Ravi, Srivaths, AnandRaghunathan, and SrimatChakradhar. "Tamper resistance mechanisms for secure embedded systems." In *VLSI Design, 2004. Proceedings. 17thInternational Conference on*, pp. 605-611. IEEE, 2004.
- [12] Bandyopadhyay, Debasis, and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization." *Wireless Personal Communications* 58, no. 1 (2011): 49-69.
- [13] Zhang, Zhi-Kai, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong- Kuan Chen, and ShihpyngShieh. "IoT security: ongoing challenges and research opportunities." In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7thInternational Conference on*, pp. 230-234. IEEE, 2014.
- [14] Nguyen, Kim Thuat, Maryline Laurent, and NouhaOualha. "Survey on secure communication protocols for the Internet of Things." *Ad Hoc Networks* 32 (2015): 17- 31.
- [15] Vazirani, Umesh V., and Vijay V. Vazirani. "Efficient and secure pseudo-random number generation." In *Foundations of Computer Science, 1984. 25th Annual Symposium on*, pp. 458-463. IEEE, 1984.
- [16] Abdullah, AkoMuhamad. "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data." (2017).

Authors' Profiles



G.V.Hindumathi is currently pursuing Ph.D. in Jawaharlal Nehru Technological University, Kakinada, India. She is specialized in Internet of Things and Network Security. Her research topic is on Security issues on Internet of Things. She works as Assistant Professor in Gayatri Vidya Parishad College of

Engineering(Autonomous).



Dr. D. Lalitha Bhaskari works as Professor in Andhra University, Visakhapatnam, and Andhra Pradesh. Her areas of expertise include: Deep Learning, Network Security, and Image Processing. And she got Young scientist award from by IEI.

How to cite this paper: G.V.Hindumathi, D. Lalitha Bhaskari, "Message Based Key Distribution Technique for Establishing a Secure Communication Channel in IoT Networks", *International Journal of Computer Network and Information Security(IJCNIS)*, Vol.11, No.11, pp.28-35, 2019. DOI: 10.5815/ijcnis.2019.11.04