DOUBLE TRAP DOOR CHAMELEON HASH FUNCTION BASED SECURITY FOR FOG IoT NETWORK

Rhupini.S.K[1], Preneetha.J[2], Natraj.N.A[3]

[1]Student (B.E), ECE, Prince Shri Venkateshwara Padmavathy Engineering college, Tamil Nadu, India
[2] Student (B.E), ECE, Prince Shri Venkateshwara Padmavathy Engineering college, Tamil Nadu, India
[3]Asst. Professor, ECE, Prince Shri Venkateshwara Padmavathy Engineering college, Tamil Nadu, India

ABSTRACT

A new industrial revolution driven by digital data, computation, and automation is arriving. Human activities, industrial processes, and research lead to data collection and processing on an unprecedented scale, spurring new products, services, and applications, as well as new business processes and scientific methodologies. The cloud which cannot be constantly connected to the sensors, extends to an edge based paradigm-fog. However security is the major issue in this edge based service. Solution to this is by using expensive cryptographic methods like role based and access based control, which almost degrades the speed of the system. This paper comes up with a solution to this critical situation with a novel cryptographic method. Double trap door hash function is used in conjunction with the paillier homomorphic encryption to increase the security and decrease the delay. This generates two hash functions which gives a double layer security in preserving the credentials being transferred. By comparing the two hash values, the MDMS can validate the integrity and authenticity of the data sent by the concentrator. The working of the cryptosystem is tested in NS2.35 with an open environment , which experiences high amount of forgery attacks and security breaches.

Keyword: - *Fog, Internet of Things, Security, Double trap door Chameleon hash, Homomorphic encryption*

1. INTRODUCTION

The term IoT refers to a vision of a future internet where any object can communicate with other devices using Internet communication protocols. The IoT paradigm has been defined as a technology to connect objects that surround us providing a reliable communication and making available the services provided by them. The IoT paves way for the concept of 'anywhere' and 'anytime'. Cloud computing is the backbone of the activities of IoT. The cloud has revolutionized the delivery of online services such as systems monitoring, surveillance, healthcare or smart cities-related services, through the internet. The term "cloud computing" refers to a variety of Internet-based computing services. The difference between cloud-based and traditional software is that when the cloud is accessed, desktop, laptop or mobile device isn't the thing doing the actual computing. The computing happens in a large data centre outside organization, and can simply see the results of it on the screen. When there is a huge amount of users using the cloud, the response time increases and the latency reduces thereby reducing the reliability and efficiency of the cloud. The main drawbacks of cloud are

- Constant internet connectivity
- Latency
- Response time
- Traffic

Many IoT services are highly distributed and rely on a complex network of many low tier devices, such as sensors, for which a constant connection to cloud service providers is impractical. To mitigate these issues, cloud computing is extended to fog, by moving a substantial amount of computation, communication and short period storage to the edge of the network, that is, to the so-called fog nodes. This allows reducing data volume and traffic to the cloud servers, decrease latency and response delay and improve the ubiquity of the system, by which the users may access the services through different entry points (i.e., the cloud but also fog nodes).

1.1 Fog computing and its Capabilities

The fog is an extension of cloud. It forms an intermediate layer between the cluster or cloud and the sensor nodes. The devices cannot always be connected to the cloud. To handle the traffic in the large number of users handled by the cloud, the fog was designed to be an edge device. Thus, the Fog emerges as an extension of the Cloud paradigm escalating from the core of the network towards its edge; it is comprised of a heavily virtualized platform able to perform storage, processing, and networking activities between the Cloud servers and the end devices. The Fog comes to support novel applications and services that are not completely fit for the Cloud, granting them ubiquity, high resilience, low latency, decentralized management, and cooperation. The architecture of fog is shown in the figure 1.

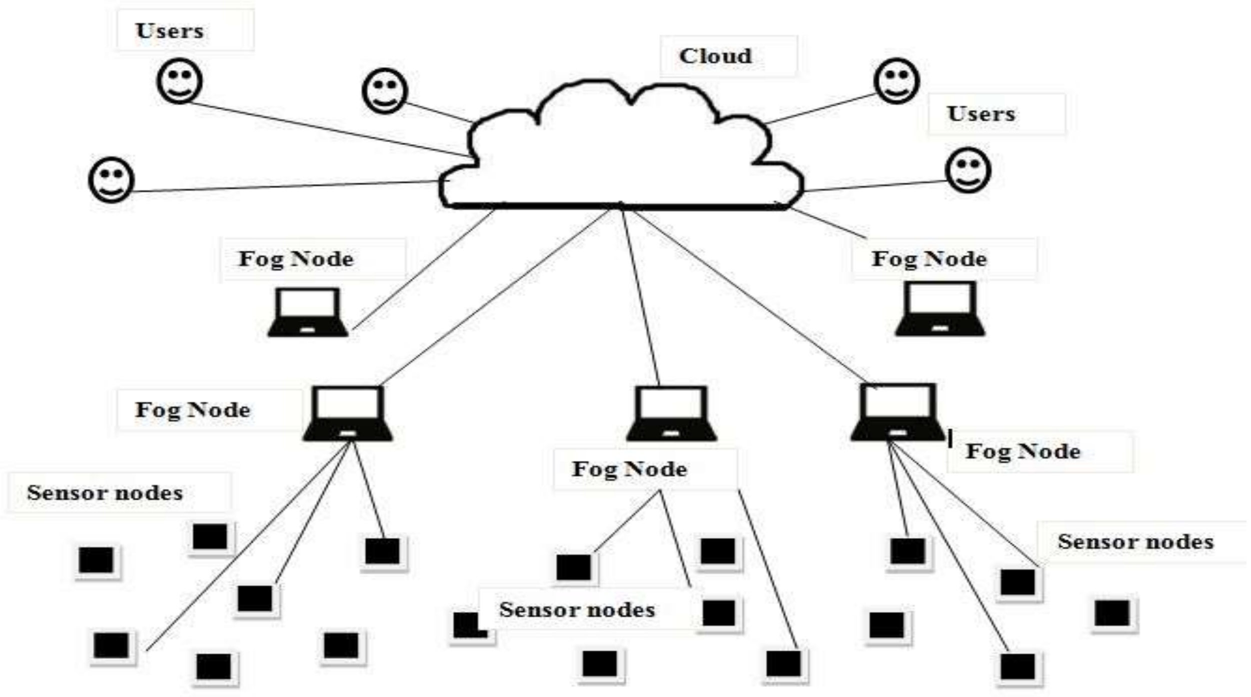


Fig -1 Fog Architecture

Fog computing-based IoT applications consist of a cloud service provider, one or several fog nodes at the edge of the network connected to the cloud and a potentially complex network of IoT de- vices below them. Fog nodes are in charge of routing requests to IoT devices and collecting and aggregating data from such devices; these data are then transmitted to the cloud, which provisions resources (computing and storage) to store and analyze the collected data and enforces high-level services based on a predefined system logic [1].

The sensor nodes are IoT devices capable of sensing a single magnitude (e.g., temperature, humidity, movement, light, etc.) and are deployed in specific locations, which are known by the nodes. Being lightweight, they only answer to sensing re- quests received by their corresponding fog node and they are not suited for highly computationally and/or energy demanding operations. As well as fog nodes, sensor nodes may be deployed in insecure areas and they are expected to be connected through untrusted network, which may result in being the target of different kinds of attacks and even becoming fully con- trolled by capable adversaries.

1.2 Challenges in Fog Computing

Fog computing-based IoT applications consist of a cloud service provider, one or several fog nodes at the edge of the network connected to the cloud and a potentially complex network of IoT devices below them. Among the several challenges inherent to fog computing for IoT, privacy and security arise as fundamental issues. More specifically, authentication, access control, rogue node and intrusion detection, and revocation are identified as main security challenges and they require a careful design due to the distributed nature of the fog computing architecture and the lack of computational and energy resources of the IoT devices. The main disadvantages are of fog computing are:

- Security and Privacy.
- Vulnerability to Attack.
- Limited control and Flexibility.
- Vendor Lock-In

Among all the major concern is the security and privacy. The encryption technique is proposed to increase the security and preserve the privacy of the data.

1.3 Attribute Based Encryption (ABE)

Attribute based encryption is the next generation way of handling authorization. It comes under identity-based encryption. ABE simplifies access management, reduces risk due to unauthorized access and centralizes auditing and access policy. It establishes the missing link and eliminate human from the loop. ABE recognizes these attributes as the missing link and highlights its presence in access control decision. Thus, it provides more transparency while reasoning about access control. However, the computational cost of ABE is the main barrier that hinders its practical deployment. In particular, the decryption cost of the ABE ciphertext is proportional to the number of attributes involved in the access policies; and this cost may hamper the latency of fog services when decryption operations are performed on IoT devices with limited resources. The encryption technique used is also complex.

2. OUR PROPOSAL

Encryption needs to be authenticated to prevent an attacker from altering the information. In order to mitigate the security issues in fog based network devices the double trap door chameleon hash function is used in conjunction with the homomorphic encryption. This gives additional security with double authentication layers. The trap door hash and the homomorphic encryption has made it difficult to decrypt the messages in short duration of time. Even if the hackers have information to decrypt the messages they would have found those keys after the time has elapsed. This has made the proposal efficient than the existing crypto techniques that are used nowadays in securing the data from IoT networks. The meter data management system embeds a blind copy of its trapdoor key distribute it to all the smart meters using a polynomial-based key distribution scheme. The entire operation is divided into the below phases:

- Node setup
- Data collection and assembling
- Verification of hash value
- Blinding the key

The basic idea is that the owner of the trapdoor key i.e. the smart meter must generate the same chameleon hash value as the concentrator by computing a commitment using its own energy readings so that the MDMS can use it to verify the hash value of the concentrator. In this case, the commitment that is sent to the MDMS need not be signed which makes the scheme very efficient.

2.1 Trap Door Hash Function

The Chameleon-hash functions, also called trapdoor-hash functions, are hash functions that feature a trapdoor that allows a person to find arbitrary collisions. However, chameleon-hash functions are collision resistant as long as the corresponding trapdoor (or secret key) is not known. More precisely, a party who is privy of the trapdoor is able to find arbitrary collisions in the domain of the function. In essence, the concentrator aggregates the energy readings from the smart meters and calculates a chameleon hash value using the public keys that are associated to the two trapdoor keys of the chameleon hash function. To verify the chameleon hash value, a smart meter equipped with one of the trapdoor keys needs to calculate a commitment value using its own energy readings such that the resultant chameleon hash value is equivalent to the previous hash value sent by the concentrator.

First the cyclic group's domain parameters are set. Let SG be a subgroup generated by P and define a cryptographic secure keyed-hash function f : Q× SG → Q. Choose two random elements which acts as two trapdoor keys k, $x \in R$ Q and compute $K = kP$, $Y = xP$. The public hash key is PHK = (K, Y), and the private trapdoor key is TpK = (k, x). For the given hash family, we define the hash key PHK and the proposed chameleon hash function as

$$H_K : Q \times Q \rightarrow SG \tag{1}$$

$$H_K (m, r) = f(m,K) \cdot (K + Y) + rP \tag{2}$$

For a group of smart meters sm_1, sm_2,... sm_m where n is the number of meters in the group, each smart meter periodically sends an energy report m_{it} to the concentrator for aggregation where m_{it} denotes the readings at time t from smi. Upon receiving the individual reports from the smart meters, the concentrator aggregates the individual reports according to equation 3

$$m_{agg}(t) = \sum mi\ (t) \tag{3}$$

The message ((m1(t),m2(t), ……. m2(t)),r(t),SIGN(Hhk(t))) is signed by the concentrator's private key using any unforgivable signature scheme to prove authenticity and non-repudiation.

2.2 Generation of Chameleon Hash

The following steps show how the chameleon hash function is being generated from the smart meters value.The public key and the private key are generated and used in the generation of cryptographic value for encryption of the data.

- Chameleon hash public keys are generated as X=xG, Y=yG
- Energy readings are taken from smart meter: $mi_{(t)}$
- Random value: r(t)

The generation of the hash function is done using the SHA2 and HMAC algorithm and to provide security for the data being sent to the smart meters. Scalar multiplication is done, a = r(t). SG. The output of the encrypted value is shown in equation 4.

$$(H_{HK(t}(m_{agg}(t), r(t))= h2\ (h1(m_{agg}(t)\|X),Y)(X+Y)+r(t)SG(\mathrm{mod}\ n) \tag{4}$$

The above equation is generated in many parts. First the meter readings are collected and aggregated. A random value r(t) is choosed from [1,.....n] and its multiplied with the subgroup SG as 'a'. Then $h1(m_{agg}(t)\|X)$ is computed using the secure hash algorithm and stored as 'b'. A new hash is generated with b and y using hash based message authentication code and stored in function c. The product of both the keys with c is summed up to be stored in function d. The addition of a and a will the required encrypted output.

- Choose r(t) and find a = r(t).SG
- Calculate b= $h1(m_{agg}(t)\|X)$
- Compute value c = h2(b,Y).
- Take sum of the product, d= Xc+Yc.
- Output H_{HK}(t)(magg(t),r(t)) = d+a.
- Send singed (mi(t),r(t),SIGN(HHK(t)) to meter management.
- SEND (H_{HK}(t), h1(magg(t)‖X), r(t)) to smart meters.

The output which is the encrypted data is calculated by the equation 5. It is the product of hash function, message and the random element generated.

$$\text{Encrypted data}=H_{hkt}\ (m_{agg}, r(t)) \tag{5}$$

2.4 Verification of hash

The generation of trap door key executed by the smart meters every time. On receiving topics of output from the concentrator where $t \rightarrow (1 \leq t \leq T)$, each smart meter selects any one of them to calculate r' value so that the generated chameleon hash value of its own energy readings m' is equivalent to the chameleon hash value stored by the MDMS at time t. When the MDMS receives the commitments r1, r2 from smart meter, it uses the pre-shared key E_{Ki} smart meter to decrypt the message to recover r1, r2. The MDMS then divides r1'by r2" to reconstruct the actual value as r'. Using r1and the derived r2'values, the MDMS reconstructs the true value of r' as r''. If the verification is unsuccessful, it means that either the concentrator or the reporting smart meter is compromised. To verify the status, the MDMS may request another smart meter in the group to send the commitments to validate the chameleon hash value at time.

3. PAILLIER HOMOMORPHIC CRYPTOSYSTEM

The homomorphic encryption is a form of encryption where the operations can be performed on the cipher texts and yet the encrypted results which when decrypted would be the same as if the operations were performed on the decrypted text in the first place. In one sentence, data can be modified and calculated without being disclosed. The Computer Security Division's (CSD) Cryptographic Technology Group (CTG) is also following the progress of emerging technologies, such as fully homomorphic encryption (FHE). For concreteness and without loss of generality, our DTCH scheme is based on the Paillier cryptosystem. The concrete description of Paillier cryptosystem is shown as follows:

- Key Generation

The generation of the public and private key using the homomorphic encryption is as follows. Firstly two large prime numbers p & q are taken. Their product is stored in N. The coprimes are found using eq.7. Then modulo multiplicative inverse is taken and the two keys (private and public) are generated.

- Large prime numbers (p,q)
- N= pq (6)
- λ= (p-1)(q-1) (7)
- L(u)= (u-1)/n (8)
- m= $(L(g^{\lambda} \bmod n^2))^{-1}$ (9)
- Public key (pk) and private key(sk)

- Encryption

The cipher text is generated using the random value, the data is encrypted using the formula and sent to the destination. Random value r is generated to compute the encrypted message.

- Ciphertext = $g^m.r^n \bmod n^2$ (10)

- Decryption of the data is done in the server side after the data is computed and sent to the user.

Message= $L(c^{\lambda} \bmod n^2)$ (11)

The homomorphic encryption is done along with the double trap door chameleon hash function where the system provides double layer security.

4. COMMUNICATION BETWEEN THE NODES

In all the sensor nodes when deployed, if node N_i wants to communicate with another nodeN_j, they must implement the following steps to authenticate each other. Subsequently, they must establish a shared session key for securing their communication. The following flow chart explains the communication between the nodes.

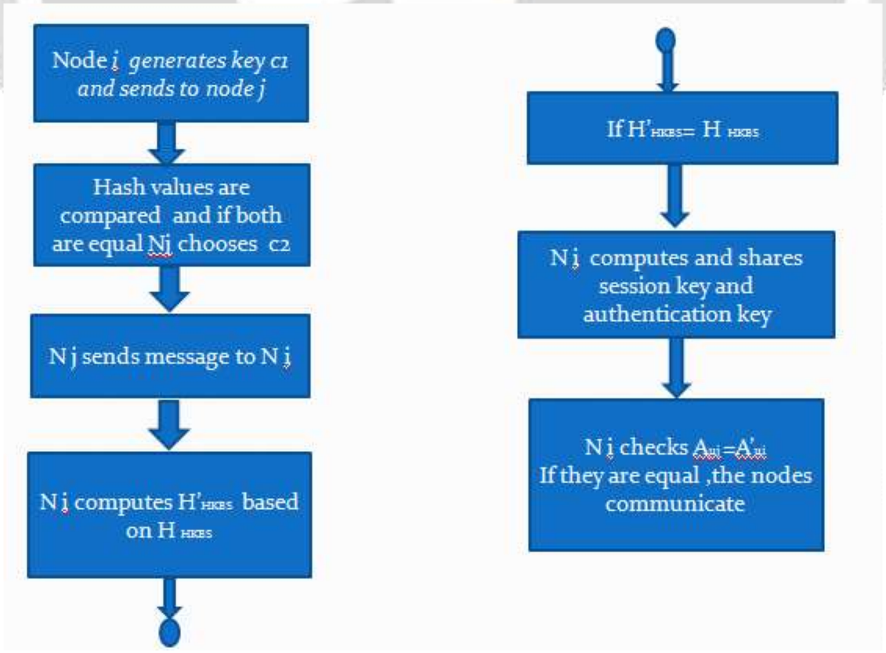

Node i generates key c1 and sends to node j
Hash values are compared and if both are equal Nj chooses c2
N j sends message to N i
N i computes H'HKBS based on H HKBS
If H'HKBS= H HKBS
N i computes and shares session key and authentication key
N i checks Aai =A'ai If they are equal ,the nodes communicate

Fig.2 Node to node communication

Even if the attacker obtains the user's information , then also the attacker cannot pass the authentication and key establishment phase, because he cannot compute the session key Au. Hence, our DTCH can resist forgery attacks. The inputs converted as ciphertexts are used directly for any computation done at the midway. Whereas, the existing methods have to decrypt the message and then perform the computation.

5. RESULTS AND ANALYSIS

The testing of this method is done using the NS2.35 (network simulator). The corresponding simulation outputs depict the deployment of nodes and their data transfer in untrusted systems where the network is highly vulnerable to attacks. Figure 6.1 shows the creation of the nodes as per the fog architecture.

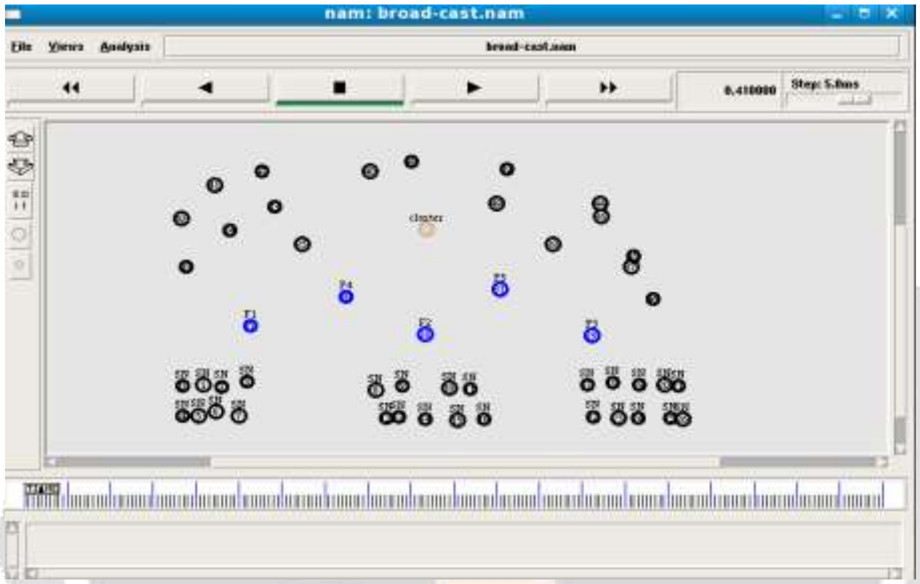
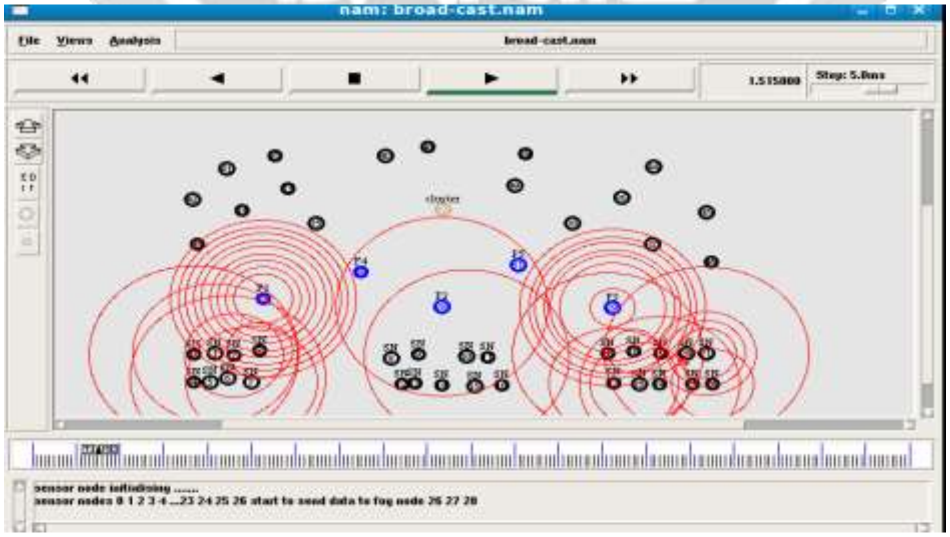


Fig.3 Node creation in NAM

The cloud is shown as a cluster node. The fog nodes form the intermediate layer between the cloud and the IoT sensor nodes. The figure 3 shows the transfer of signal among the nodes. The user sends the requests to the cloud and fog nodes. The fog nodes collect the requests, aggregate it and transfers it to the sensor nodes through orchestration.



```
sensor node initializing .......
sensor nodes 0 1 2 3 4 ...23 24 25 26 start to send data to fog node 26 27 28
```

Fig.4 Data transfer from fog nodes to sensor nodes

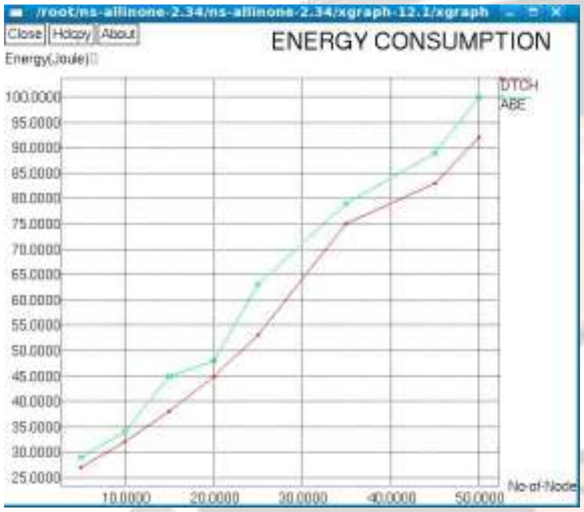
The initialization and authentication are done prior to the data transfer. Once the nodes are authenticated by using the DTCH they are encrypted using the homomorphic encryption. This gives double layer security and prevents the data going into wrong hands. The figure 4 shows transfer of data in an open environment that is subjected to many attacks. The packets are dropped due to the high rate of attacks that try to hack the information.

5.1 GRAPHICAL OUTPUTS

The data transmission between the nodes is done using the double trap door chameleon hash function-based encryption. The values computed from the end results were collected to depict the results as a graph. The following three graphs are plotted from the readings got by the data transmission that was done using DTCH cryptosystem.

5.1.1 Energy Consumption

The figure 6.4 shows the graphical output of the proposed DTCH security method in terms of energy consumption. The proposed encryption has reduced the energy consumption of the nodes during the transfer of data in a secured way than the existing ABE method. The x axis is the number of nodes taking part in the transmission and y axis is the amount of energy consumed by the nodes during the secured process.



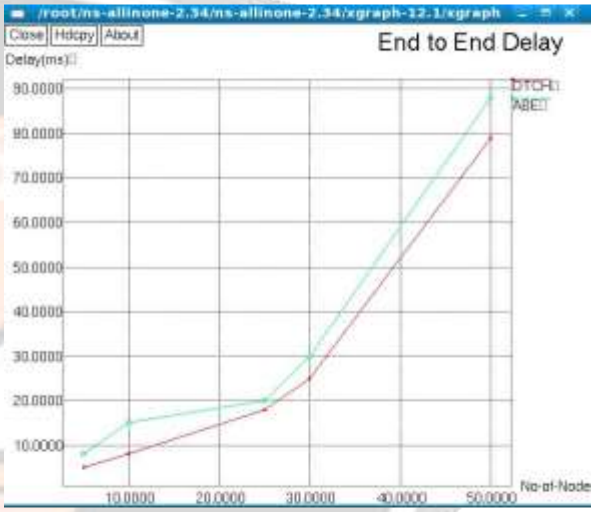


Fig. 5 Number of nodes vs. energy consumed **Fig.6** Number of nodes vs. delay

The concept of using only the nodes that are taking part in the network and the use of homomorphic encryption has helped in reducing the energy consumption lower than the ABE cryptosystem.

5.1.2 Delay

The figure 6 shows the graph that depicts the time taken between the request and response. It marks the time delay of the response given to the user requesting the data. The DTCH has reduced the delay and overall latency of the transmisson. In the ABE if a data needs to undergo a midway computation it has to be decrypted to take part in the calculations. But the proposed DTCH that works in conjunction with the homomorphic encryption can do the computations with the ciphertexts. This reduces the time taken for intermediate time delays for decryption. The x-axis shows the no.of users and the y-axis shows the time taken for a nodes to transfer the data i.e. delay.

The graphical results show that the DTCH has ensured the security and reduced the consumption of energy of nodes to deliver the messages, with reduced delay. This makes it better than the existing cryptosystem

6. CONCLUSION

The proposed data transfer using the DTCH encryption achieves better security, low cost, less delay and low power consumption. The experimental results got by deploying the fog like architecture in NS2.35, shows that the existing model has less security, high cost of computing and more energy consumption. But the proposed cryptosystem tested in a more open environment subjected to different kinds of attacks has achieved to better security and preserved the privacy of the data being manipulated in a wireless network. The energy consumption has been reduced to 8.435% , the delay reduced to 32.2% than the existing model. The drawback is that this system has achieved a lesser packet delivery ratio than ABE. The increased security has prevented larger number of attacks leading to a lesser PDR. In scenarios where the data is overheard by eavesdropping false codes and manipulating the data using wrong authentication, the DTCH combined with homomorphic encryption can be a best method to preserve the security of the data. The idea may be extended in achieving greater security without compromising on the packet delivery ratio in the future works.

7. REFERENCES

[1]. Alexandre Viejo and David Sánchez's ' Secure and privacy-preserving orchestration and delivery of fog-enabled IoT services' , 2018

[2]. A. Abduvaliev, S. Lee, and Y.-K. Lee, “Simple hash basedmessage authentication scheme for wireless sensor networks,” in Proceedings of the 9th IEEE International Symposium on Communications and Information Technology (ISCIT ’09), pp. 982–986, Incheon, South Korea, September 2009.

[3] I. F. Akyildiz,W. Su,Y. Sankarasubramaniam, and E.Cayirci, “A survey on sensor networks,” IEEE Communications Magazine,vol. 40, no. 8, pp. 102–114, 2002.

[4] I. F. Akyildiz and I. H. Kasimoglu, “Wireless sensor and actor networks: research challenges,” Ad Hoc Networks, vol. 2, no. 4,pp. 351–367, 2004.

[5]. Ng, W., Wu, H., Wu, W., Xiang, S. and Tan, K. (2012) ‘Privacy preservation in streaming data collection’, in: 2012 IEEE 18th International Conference on Parallel and Distributed Systems, pp. 810–815.

[6] Shafagh, H., Hithnawi, A., Burkhalter, L., Fischli, P. and Duquennoy, S., (2017) ‘Secure sharing of partially homomorphic encrypted iot data’, in: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems.

[7] Bonomi, F., Milito, R., Zhu, R. and Addepalli, S. (2012) ‘Fog computing and its role in the internet of things’, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, pp. 13–16.

[8] Carminati, B., Ferrari, E., Cao, J. and Tan, K. (2010) ‘A framework to enforce access control over data streams’, ACM Trans. Inf. Syst. Secured 13 (3).

[9] Fan, K., Wang, J., Wang, X., Li, H. and Yang, Y. (2017)‘A secure and verifiable outsourced access control scheme in fog-cloud computing’, Sensors 17 (7).