International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 10
ISSN: 2454-4248
86 – 91

Securing IoT with Trusted Authority Validation in Homomorphic Encryption Technique with ABE

Mr. Suraj U. Rasal
Assistant Professor,
Department of Computer Engineering, Bharati Vidyapeeth
University College of Engineering Pune, India
surasal@bvucoep.edu.in

Mr. Raghav Agarwal
Department of Computer Engineering, Bharati Vidyapeeth
University College of Engineering Pune, India
agarwalraghav01@gmail.com

Mr. Yash Agarwal
Department of Computer Engineering, Bharati Vidyapeeth
University College of Engineering Pune, India
agarwal.yash276@gmail.com

Mrs. Varsha S. Rasal
Department of Computer science & Engineering, Nehru
College of Engineering & Research Centre, Thrissur, India
vs.rasal@yahoo.com

Abstract— Existing security system includes levels of encryption. IoT access is very important aspect. Failure of IoT security can cause more risks of physical and logical damage. IoT contain both functionalities including physical or computational process. In proposed approach, levels of encryption are enhanced by increasing levels of security. User can access IoT through central trusted authority only. Instead of actual data like user credentials or I/O functionality of Internet of things, encrypted data is delivered. Trusted authorities are been involved in secured IoT access structure by considering their credentials. Trusted authority is selected randomly, based on randomized selection algorithm. Based on secured logic, decryption key will be delivered to the IoT through separate channel by trusted authority. Session management has been added by considering initial and waiting time after which all encryption or decryption data will be expired. Homomorphism is applied in encryption process where proposed logic is applied on considered data after which again RSA algorithm is applied. Overall, proposed logical approach, homomorphism, session management, secured access structure and trusted authority involvement improves the security level in IoT access process.

Keywords- *Key from trusted authority TK, I/O functions request Funct_n_Req( ), Initial time it, Waiting time wt, proposed logic AL , Attribute Based Encryption ABE.*

*****

I. INTRODUCTION

With the help of Internet of things, machine can communicate with each other. IOT provides connectivity for everything and everyone. But there are many security flaws in IOT. Hence in proposed approach advanced crypto graphical techniques are considered like ABE (Attribute based Encryption), homomorphism, session management, decentralized authorities approach and crypto graphical algorithms to cover as many security flaws. Attribute based Encryption or the log encryption schemes are used that represent user attributes as a monolithic set in keys instead of encrypting each log file with diff key. To access a particular file, user need the specific attribute to decrypt that file .So ABE encryption can use on the databases, as databases are one of the most important as it contains very sensitive and personal data. A first step in addressing this problem of trust is to only store information in encrypted form. However, data access is not static as employees are hired, fired or promoted so every time it cannot be changed the credentials of the database. But it will be necessary to change the authority that can access the data. SAP Hana is one most used databases now days by automobile and other companies [1]. They have personal info about their customer. Anybody may be able to penetrate the server and bypass authentication by exploiting software vulnerabilities. So the solution to this is Cryptography is user can access the control through Attribute based Encryption (ABE) [2]. For Instance in the Ubiquitous IoT application, such as smart city, data is gathered by many people of different domain. The data may be out without the knowledge of the user and transmitting in the plain text. There may be many departments and the data may be transferred inter department that can be accessed by an unauthorized user and can cause serious problem. In proposed approach, main area of Interest is to exploit the heterogeneous nature of the IoT to make best possible use of Attribute based Encryption schema in different environment. For Instance in the Ubiquitous IoT application such as smart city, data is usually gathered by many people of different in domains. The data may be out without the knowledge of the user and transmitting in the plain text, there may be many departments and the data may be transferred inter department that can be accessed by an unauthorized user and can cause serious problem [3]. Main area of Interest Proposed approach is to exploit the heterogeneous nature of the IoT to make best possible use of crypto graphical schema with different security approaches.

 ISSN: 2454-4248
86 – 91

II. EXISTING SECURITY TECHNIQUES

A . *Attribute & identity-based encryption*

Attribute-based encryption (ABE) is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, for example email address of the receiver [4].

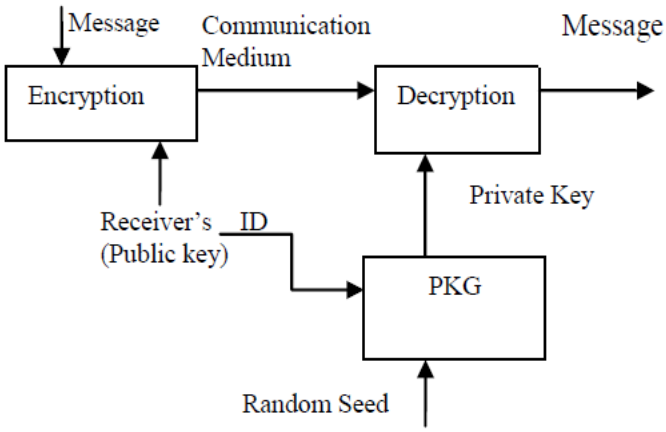

Fig.1 Identity Based Encryption [4]

ABE goes one step further and defined the identity not atomic but as a set of attributes for example roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (cipher ext-policy ABE - CP-ABE). The key issue is someone should only be able to decrypt a cipher text if the person holds a key for matching attributes where user keys are always issued by some trusted party

B. Ciphertext-Policy ABE

1) With Attribute Based Encryption

In cipher text-policy attribute-based encryption (CP-ABE), a user's private-key is associated with a set of attributes and a cipher text specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a cipher text, if and only if his/her attributes satisfies the policy of the respective cipher text [5]. Policies may be defined over attributes using conjunctions, disjunctions and (k,n) (k,n)threshold gates, that kk out of nn attributes have to be presented It is assumed that the universe of attributes is defined to be {A,B,C,D}{A,B,C,D} and considered user received a key to attributes {A,B}{A,B} and another user to attribute {D}{D}. If a cipher text is encrypted with respect to the policy (A∧C)∨D(A∧C)∨D, then receiving user will able to decrypt, while sending user will not be able to decrypt. CP-ABE thus allows realizing implicit authorization, i.e. authorization is included into the encrypted data and only user who satisfies the associated policy can decrypt data. Another nice feature is, users can obtain their private keys after data has been encrypted with respect to the policies. So data can be encrypted without knowledge of the actual set of users that will be able to decrypt, but only specifying the policy which allows decrypting. Any future users that will be given a key with respect to attributes such that the policy can be satisfied will be able to decrypt the data [6].

2) Key-Policy ABE

KP-ABE is the dual to CP-ABE in the sense that an access policy is encoded into the users secret key, for example (A∧C)∨D(A∧C)∨D, and a cipher text is computed with respect to a the set of attributes, for example {A,B}{A,B. In this example the user would not be able to decrypt the cipher text but would be able to decrypt a cipher text with respect to {A,C}{A,C}. An important property which has to be achieved by both, CP- and KP-ABE is called collusion resistance. It means it should not be possible for distinct users to pool their secret keys such that they could together decrypt a cipher text that neither of them could decrypt on their own (which is achieved by independently randomizing users' secret keys) [6].

3) Beyond ABE

ABE is just one type of the more general concept of functional encryption (FE) covering IBE, ABE and many other concepts such as inner product or hidden vector encryption its example can be searchable encryption, decentralized policy etc [5]. It is a very active and young field of research and has many interesting applications like in the field of cloud computing.

C Homomorphic Encryption

Homomorphic encryption is a type of encryption which allows us to computation over cipher text. This generates an encrypted result which matches the same result as the computation over plain text. Researchers have made homo morphic encryption because as it doesn't has to decrypt the file again and again to compute, but if user don't has homomorphic encryption towards his or her system, it makes the file vulnerable as soon as he decrypts it. Homomorphic encryption is still-mostly-theoretical advancement in the science of keeping secrets. It is described as the plausible construction of a fully homomorphic cryptosystem [7]. Gentry construction consist of several steps. He first constructed a "somewhat homomorphic" scheme that supports evaluating low degree polynomials on the encrypted data. It is GGH-type scheme over ideal lattices. It is also proved that with an appropriate key-generation procedure, the security of scheme can be reduced to the worst-case hardness of some lattice problem in ideal lattices [8]. Fully homomorphic encryption needs the concept of boot strapping.It allows an evaluation of class of functions below some complexity threshold. In his construction, homomorphic encryption scheme allows the homomorphic evaluation of any function whose polynomial representation has bounded degree. While using boot strapping techniques, it enforces the public key of the scheme to expand linearly with maximal depth of evaluated circuits. Size of the public key is made independent of circuit depth on one condition if homomorphic scheme can securely encrypt its own secret key.

D. *Symmetric & Asymmetric encryption*

1) Asymmetric encryption

Public key encryption or asymmetric cryptography is an encryption technique that uses pairs of keys as public key and private key. Public keys are distributed widely to the public. Private keys are kept secret and known only to the owner. This

 ISSN: 2454-4248
86 – 91

accomplishes two functions authentication and encryption. In an Asymmetric encryption technique, any person can encrypt a message using the public key of the receiver that was widely distributed, but the message can be decrypted only with the receiver's secret private key [9].

2) Symmetric encryption

Symmetric-key encryption refers to the encryption technique in which both the sender and receiver share the same key for the exchange of information that is to be kept secret though a message or group of messages may have a different key than others. In this, it is implemented as either block of ciphers or stream ciphers. The biggest disadvantage of symmetric ciphers is the necessity of Key Management to use them securely. With each different pairs of communicating parties there must be different keys and each cipher text exchanged as well. The number of keys required to accomplish that increases as the square of the number of network members, which are very quickly given rise to complex key management schemes to keep them all consistent and secret [9].

E. *Securing Iot Architecture*

The security architecture of IoT depends upon the previous security models and how the internet is implemented to IoT. The security model of the IoT is designed keeping in the mind that IoT neither has a standard execution environment nor it has high computational power. It is difficult to implement a standard secrete IoT architecture [10].

1) Existing security in IOT architecture

There are many security issues in the physical security layer of IoT model. Information processing consists of vulnerability at every layer. There exists layered security for every level. If it is identified in the perception level, it has RFIDS, sensors, gateway. All these are highly vulnerable for attack. The perception layer consist of short distance communication in external and internal ad-hoc network and the main issue is created by low-cost tags or sensors which are vulnerable and easy to exploit by the attacker to make it behave as per the attacker's decision. The attacker can misuse it to get the unauthorized access to anywhere. To secure this layer, perception layer, local security is added which uses crypto graphical algorithm and protocol to secure it. In the architecture of IOT, the second layer is network layer. Security is divided into two types, first one is access core network with information security and the second is perception layer network transmission with information security.

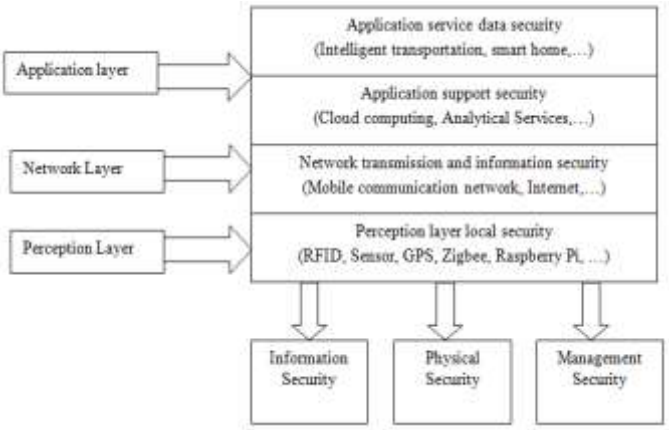

Fig.2 Layers of IoT architecture [11]

In the mobile network, communication is wireless and it leads to access heterogeneity and causes security issues as it access different media switching technology. In this, the wireless channels can be tapped capturing even modifying, inserting, deleting or retransmitting messages transmitted through radio interface in order to fake user identity or deceive server. The complete security is given by the core network as of its traditional advantages of network information safety. But still it faces the traditional network security threats and vulnerabilities. Third and the final layer of the architecture of the IoT is the application layer. It has multiple application integration, various systems, and multiple formed data. It is targeted for the privacy breach for information. Mainly application layer is used for the protection of the privacy of the users from the unwanted access to the information. The main purpose of designing the security architecture of the trusting IoT is to provide information security protection for tag privacy, sensor data security, and data transmission [11].

a) *Embedded Privacy*

Embedded Privacy or Security is addressed throughout the device lifecycle. This can be achieved through a multi-layered approach that starts at the beginning when power is applied to establish a trusted computer baseline and all the way to the operational environment.

b) *Secure Booting*

The authenticity and integrity of the software on the device is verified when power is first introduced to the device by the means of cryptographically generated digital signatures. The Digital Signature attached to the software image is verified by the device which ensures that the only software containing this Digital Signature is authorized to run on that device and signed by the entity that authorized it, will be loaded. This ensures the foundation of trust has been laid, but the device still needs protection from various run-time threats and malicious intentions.

c) *Access Control*

IoT can be controlled using attribute based encryption with OTP [12]. It has been proposed that to make IoT access more secured, One Time Password security level has been added to the IoT access structure. Access Control is gained by establishing role-based access control which is built into the operating system that limits the privileges of device

components so that the applications access only the resources they need to do their jobs. This ensures that even if a component is compromised then the intruder still has a minimal access to other parts of the system.

d) *Securing Cloud*

Cloud computing has been envisioned as the next generation architecture of IT enterprise. In cloud computing the application software and the databases are move to large centralized data server which are not fully trustworthy , and the growth of this industries is increased day by day and gives us many security challenges so we have a numerous number of ways to secure cloud. [12]

F. *Digital Signature & Hash functions*

Digital signature is a mathematical function used to validate the authenticity and the integrity of a software and digital document. It can provide the added assurances of evidence of sender identity and status of an electronic document and get the acknowledging informed consent by the signer. Digital signatures do help in NON-Repudiation. So digital signature is be very useful in securing cloud as all cloud Consumers can verify the source and integrity of their cloud service via digital signature verification . The cloud provider must have a public signature key and the private key should not be given to unauthorized user. Anybody cannot manipulate anyone services as if he did his private key would not match the cloud provider and the users private key. Digital signature uses public key cryptography such as RSA &SHA [14],[15]. It has to generate two keys that are linked with each other as private and public. The private key is generated using hashing algorithm and encryption of hash with other information to make it more secure. By hashing it there is another way to assuring the integrity of the cloud service by using Cryptographic Hash such as SHA-256(Secure Hash Algorithm)[14].

G. *Message Authentication Code(Mac)*

MAC is an approach for the integrity. It is generated using a cryptographic function. It produces a message authentication code which is secret shared between the user and the provider. By MAC, all the customers will be able to verify the integrity of their cloud service via MAC verification at the time logging in the provider who will generate a secret code. Generated code will be shared with the user at that particular moment through any secure means of communication while logging in the user have to enter that secret key which was shared with the user [16] . Even if anybody knows the private key, it wouldn't be able to interrupt the services. This key is valid for a particular time period. Note that MAC is not a digital signature.

III. PROPOSED APPROACH

Internet of things covers almost measure work in the current life. Its appliances are increased in current tradition to cover all domains requirement. Internet of things is growing technology to segregate multi functionality environment which extends the complexity in access structure and working environment. Multilevel encryption techniques are needed to improve security. Proposed approach includes multilevel encryption to enhance the security level in internet of things

A. *Authorities and users*

Authorities are added to enhance the encryption level. Some authorities are selected randomley.

1) *Central Trusted Authority*

Internet of things platform is necessary for working environment. Platform is created based on IoT access structure laws and policies. Central Trusted Authority (CTA) manages all working IoT environment. CTA acts as indirectly the central system. Central Trusted Authority works with proposed logical and algorithmic approach. According to proposed approach, Trusted Authorities (TA) are registered in CTA. CTA provides surety to the registered user about security. Because multiple trusted authorities are registered in CTA to enhance the encryption level where CTA also doesn't has idea about logical and algorithmic approach applied at TAs. TAs provide their own security contribution to increase the security level. Central Trusted Authority is considered as central managing system which interacts with registered users, trusted authorities and data servers.

2) *Parallel Central Authority*

Parallel Central Authority (PCA) is installed in different location than Central Trusted Authority to overcome with server down and disaster problems. It will be working in parallel and same manner as Central Trusted Authority.

3) *Registered User*

Ceiling working environment required for internet of things is internet. Internet provides a platform to create working environment for internet of things. Registered user can access internet of things by entering user and security credentials on access structure. Access structure provides user interface.

4) *Trusted Authorities*

Trusted Authorities are considered to increase the level of encryption and follow legal ethics. Trusted Authorities contribute their encryption level to enhance the level of security and trust between user and IoT service provider. Security policies and logical approaches like encryption and decryption are known to the registered trusted authorities only. Algorithmic and logical approaches vary trusted authorities to trusted authorities.

B. *Proposed Homomorphism*

Trusted authority receives the data delivered by Central Trusted Authority. Data is already encrypted by Central Trusted Authority. Re encryption is applied to the same received data delivered by the Central Trusted Authority. Trusted Authorities apply their own encryption techniques based on their logical and algorithmic approach.

1) *Initialization request*

User sends the request through internet to initiate the IoT appliance I/O. Rather than delivering request to the IoT appliance directly, it is delivered to the Central Trusted Authority. CTA continues user's request to deliver it to the IoT through internet.

 ISSN: 2454-4248
86 – 91

2) *Trusted Authority selection*

When request is forwarded to the IoT, it is attached with stipulated time data within which other approval request needs to be retrieved at the IoT appliance side. Simultaneously Central trusted Authority sends the same request delivered by the client to the Trusted Authority. Here Trusted Authority selection is based on randomized selection algorithm working in CTA. The respective TA delivers it own key to the IoT appliance.

User logins to the Central Trusted Authority (CTA) through which IoT appliance can be logged in. User credentials are used to login which defines first security level. Based on user account, temporary user id is allocated as Temp_Uid. Here randomized selection RS alogorithm is applied to select Trusted Authority (TA) from available and registered trusted authorities. Trusted authorities are worked according to own identificatio protocols. Which means that trusted authorities delivers their decryption key according to their identification policy only. Trusted authority knows which trusted authority has been selected for decryption perpose.

Fig.3 IoT Access Structure

3) *Attribute Based Encryption with cipher policy*

All trusted authorities includes same decryption A_L logical policy. But the measure difference is their personal identification number allocated with respect to current transaction time only. If user is logged in to the CTA on 19Aug2016 then on the same day, trusted authority generates their own identification number in the form of key which expires after waiting time. If TA2 is selected based on randomzed selection algorithm, TA2 will deliver its current personal identification key T_K ($T_{9@Xy*}$) to the CTA. A_L is logical algorithm applied to generate secured data without including real data. This secured data will be applied with cipher and attribute based encryption policy. In the example id user has been allocated with some user id Temp_Uid as XY12AAS. Due to which actual user details will be hided. Through CTA, user can provide 'n' number of inputs. Based on user inputs, respective Funct_n_Req( ) is generated as FQ which is added to the required component set of encrytion process.

TABLE. I IoT I/O with defined function request

IoT I/O	Funct_n_Req( )
I/O_1	Funct1_Req( )
I/O_2	Funct2_Req( )
I/O_3	Funct3_Req( )
……	……

Encryption process requires components like Temp_Uid, I_time, W_time, TK and Funct_n_Req( ). AL algorithm is applied to form encrypted cipher data. ED is encrypted data and DE is decryption key formed by trusted authority based on data provided by central trsuted authority.

CTA includes logarithmic approach AL which considers components seperately. While decryption, all these components are extracted as it is to veryfy the data. If data generated by CTA is AB1C9X including initial and waiting times as it_and wt respectively, d_AB1C9X is user request data delivered and ED is encrypted data. Here d_AB1C9X is formed with Temp_Uid, I_time, TK and Funct_n_Req( ). Evrything happens with respect to the current date. Here AL is applied on components to form single data.

$U_{id} + T_K + F_Q \xrightarrow{A_L} d_{\_AB1C9X}$

Again RSA algorithm is applied on $d_{\_AB1C9X}$ to form encrypted data R_{Dn}.

$d_{\_AB1C9X} \xrightarrow{RSA} R_{D1}$

Same logic is updated in IoT appliance structure which continuously synchronizes at both ends.

RDn + it + wt *AL* ED

TABLE. II Data segments with actual and encrypted data

$d_{\_n}$	R_D	U_{id}	T_K	F_Q
$d_{\_AB1C9X}$	R_{D1}	XY12AAS	$T_{9@Xy*}$	Funct1_Req( )
$d_{\_QW2D7Z}$	R_{D2}	VSR19SS	$T_{*9@Sr\$}$	Funct3_Req( )
…….	……...	……	…….	……...

A_L is proposed cipher policy algorithm. It considers data as different segments from database table. If R_{D1} is RSA applied data, i_t is initial time and w_t is waiting time, then values will be considered separately. Again RSA algorithm is applied to form reencrypted data as R_E. So it has three level encryption including first applied RSA for $d_{\_AB1C9X}$, second A_L and third is again applied RSA. A_L is applied to retain segmentwise data.

$E_D \xrightarrow{RSA} R_E$

 ISSN: 2454-4248
86 – 91

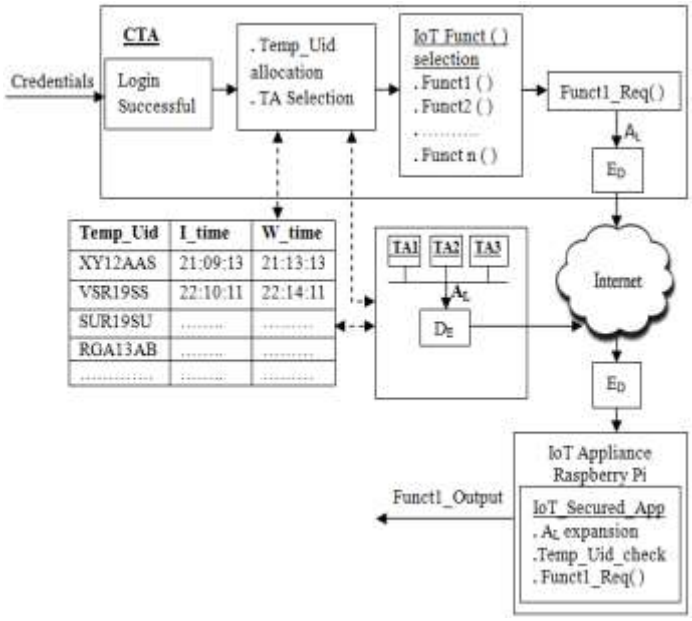

Fig.4 IoT access structure with secured homomorphism

Actual data will be traveleld as R_E. So even if anybody reqtrives R_E data, he or she cant retrieve the original data. Same approach applied in the reverse order to get original data.

$R_E \xrightarrow{RSA} E_D$

Here when A_L is applied on E_D and R_{Dn}, segment wise data will be retrieved.

$E_D \xrightarrow{A_L} R_{Dn} + i_t + w_t$

$R_{D1} \xrightarrow{RSA} d_{\_AB1C9X}$

$d_{\_AB1C9X} \xrightarrow{A_L} U_{id} + T_K + F_Q$

Actual retrieved data is considered for verification. Indirectly, actual data includes user details, initial time, waiting time, function request and trusted authority key. Due to different locality, time zone may vary. To overcome with this problem, IoT secured app and CTA-TA synchronises timely so that they can set their time module with same time constants. Homomorphic encryption is applied to increase the level of encryption. Proposed cipher policy based A_L algorithm is applied on U_{id}, T_K & F_Q at first level to seggregate the segmentwise data ito single file $d_{\_AB1C9X}$. In second level, RSA is applied on formed data in the first level to form encrypted data R_{Dn}. In third level, two components are considered as i_t & w_t with first leveled encrypted data R_{Dn} .New formed encrypted data E_D is again is encrypted using RSA algorithm in final level with respect to with homomorphic encryption. Same algorithms are applied in reverse order to decrypt the data. So level of encryption enhances the security level.

IV. CONCLUSION

IoT is accessed and controlled via internet. Its security is very important aspect. IoT is widely used for logical functioning or physical work. If it is not secured, its damage can cause severe issues. Proposed approach enhances the level of encryption where instead of actula data, encrypted data is delivered. Multiple authorities has been invoved in the level of security to hide the information and protocol policies. Attribute encryption and cipher policy improves the data identity. Homomorphism has been applied to enhance the level of encryption. So proposed approach includes four levels of encryption to make IoT access structure more secure.

REFERENCES

[1] Rasal, S.U., Gupta, K., Mulik, V.T. and Shelar, S.T., 2016. Improving Security in SAP-HANA Cloud by Applying Multiple Encryption Policies. International Journal of Science Technology & Engineering, pp.196-200.

[2] Rasal, S., Relan, S. and Saxena, K., 2016. OTP Processing using UABE & DABE with Session Management. International Journal of Advanced Research in Computer Science and Software Engineering, 6(5), pp.57-59.

[3] Theodoridis, E., Mylonas, G. and Chatzigiannakis, I., 2013, July. Developing an iot smart city framework. In Information, intelligence, systems and applications (iisa), 2013 fourth international conference on (pp. 1-6). IEEE.

[4] Shamir, A., 1984, August. Identity-based cryptosystems and signature schemes. In Workshop on the Theory and Application of Cryptographic Techniques (pp. 47-53). Springer Berlin Heidelberg.

[5] Ramesh, D. and Priya, R., 2016, January. Multi-authority scheme based CP-ABE with attribute revocation for cloud data storage. In Microelectronics, Computing and Communications (MicroCom), 2016 International Conference on (pp. 1-4). IEEE.

[6] Bethencourt, J., Sahai, A. and Waters, B., 2007, May. Ciphertext-policy attribute-based encryption. In Security and Privacy, 2007. SP'07. IEEE Symposium on (pp. 321-334). IEEE.

[7] Gentry, C., 2009, May. Fully homomorphic encryption using ideal lattices. In STOC (Vol. 9, No. 2009, pp. 169-178).

[8] Gentry, C. and Halevi, S., 2011, May. Implementing Gentry's fully-homomorphic encryption scheme. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 129-148). Springer Berlin Heidelberg.

[9] Fujisaki, E. and Okamoto, T., 2013. Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology, pp.1-22.

[10] Zhou, L. and Chao, H.C., 2011. Multimedia traffic security architecture for the internet of things. IEEE Network, 25(3).

[11] Suo, H., Wan, J., Zou, C. and Liu, J., 2012, March. Security in the internet of things: a review. In Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on (Vol. 3, pp. 648-651). IEEE.

[12] Varsha S Rasal, Suraj U Rasal, Shraddha T Shelar. (2016). Enhancing Privacy And Security Through Mediator Using DCP-ABE with OTP. The IIOAB Journal. 7 (1), p277-283.

[13] Wind Rive. (2015). White paper on SECURITY IN THE INTERNET OF THINGS Lessons from the Past for the Connected Future. Wind. 1 (01), p1-6.

[14] Kalpana, P. and Singaraju, S., 2012. Data security in cloud computing using RSA algorithm. IJRCCT, 1(4), pp.143-146.

[15] Lee, J., Chang, D., Kim, H., Lee, E., Hong, D., Sung, J., Hong, S. and Lee, S., 2005, November. A New 256-bit Hash Function DHA-256: Enhancing the security of SHA-256. In Cryptographic Hash Workshop hosted by NIST.

[16] Bernstein, D.J., 2005, February. The Poly1305-AES message-authentication code. In International Workshop on Fast Software Encryption (pp. 32-49). Springer Berlin Heidelberg.