Scalable Asymmetric Security Mechanism for Internet of Things

Ayesha Siddiqa[1], Sohail Ahmed[2]
Department of Computer Science and Information
University of Lahore, Gujrat Campus
Pakistan

***Abstract*—The Internet of things stances rigorous demands on excellence of quality and the vitality of security. It becomes vital to provide an extremely reliable encryption algorithm with less complexity and computational expense in IoT paradigm. Most of the protocols designed in past for communication between sender and the receiver based on asymmetric cryptography algorithms poses high computational cost. Therefore, this paper presents a less complex and more secure and fast encryption algorithm for communication between devices i.e. Asymmetric Scalable Security between sender and the receiver of the information. We present a reliable, secure, scalable and efficient communication protocol that used asymmetric algorithm for securing the exchange of information between sender and the receiver. The proposed communication protocol is lightweight encryption method that does not require complex resources to perform the computations involved for using the asymmetric cryptography. The simulation results also show that the proposed method is efficient in terms of time and space and ensures confidentiality. Therefore, the proposed scheme is beneficial for providing the secure communication for the power and resource constrained IoT devices.**

***Keywords*—Asymmetric cryptography; confidentiality; internet of things; security**

I. INTRODUCTION

The Internet of things is one of the most demanding technology developments field in this era of digital world. The devices that comprises of sensors and actuators have sufficient functionality of supporting proficiencies of networking and other processing abilities. So these abilities make them communicate over the internet, to communicate with each other and provide services over the Internet [1]. The advent of big data analysis has brought tremendous advantage to the creation of a smart society so this realization poses many challenges such as getting authenticating on the network, encrypting the information shared between power and resource constrained devices [3,22]. These challenges require an efficient and lightweight communication mechanism, which is scalable and lightweight too for constrained devices [4, 25]. The IoT has a lot of diverse and heterogeneous devices and multiple purpose technologies that are manufactured and distributed by the different vendors so these devices may vary in their proficiencies. We can say that the Internet of things is comprised of different types of sensors and objects, which are called constrained devices; these are called constrained devices because they have constrained resources in terms of memory, power, processing power, communication and the user interfaces. They have very low power and very little memory. When these devices are used on the network; they also put constraints on the network as well; so the networks may expose to the large number of packet loses, less throughput rate and many types of advanced security facets. Hence, the first and foremost challenge in the world of IoT is to adapt to capabilities of these networks along with their integration with the traditional Internet Communication standards.

Conventionally the encryption algorithms comprises of symmetric and asymmetric key algorithms and most of the encryption algorithms which have been proposed in past for IoT devices involve symmetric key algorithms. The symmetric cryptography is being used in IoT devices as we have low powered and low memory equipment and same key is shared between sender and the receiver of the information. However, in recent years many algorithms have been proposed for secure communication, which is the target of this study [23].

Security parameters such as authentication, confidentiality, and integrity and access control are addressed by the different researchers and proposed many possible solutions to ensure security. The work of researchers includes the use of IP security i.e. IPSec, Internet Key Exchange protocols, using Transport layer security, using Datagram Transport Layer Security (DTLS) and key bootstrapping [20,28]. Although, these solutions have been proposed in an efficient way to ensure security in IoT core devices; but these solutions cannot be applied in the IoT constrained devices directly due to their limited computation capabilities and resources [32].

In this paper, we have presented an efficient and secure mechanism for communication in an IoT domain using the Asymmetric cryptography. We have used RSA as the PKC algorithm to secure the communication between sender and the receiver. The RSA is a block cipher algorithm [6] so it can be easily used with the IoT devices. The rest of the article is arranged as follows. Section II discusses the conceptual framework that includes the basics of public key cryptography and RSA algorithm. Section III describes the related work. Section IV elaborates the framework and defines the problem statement. Section V elaborates the proposed methodology and implementation details. Section VI explains the analysis and results of the proposed methodology while in Section VII we conclude the article.

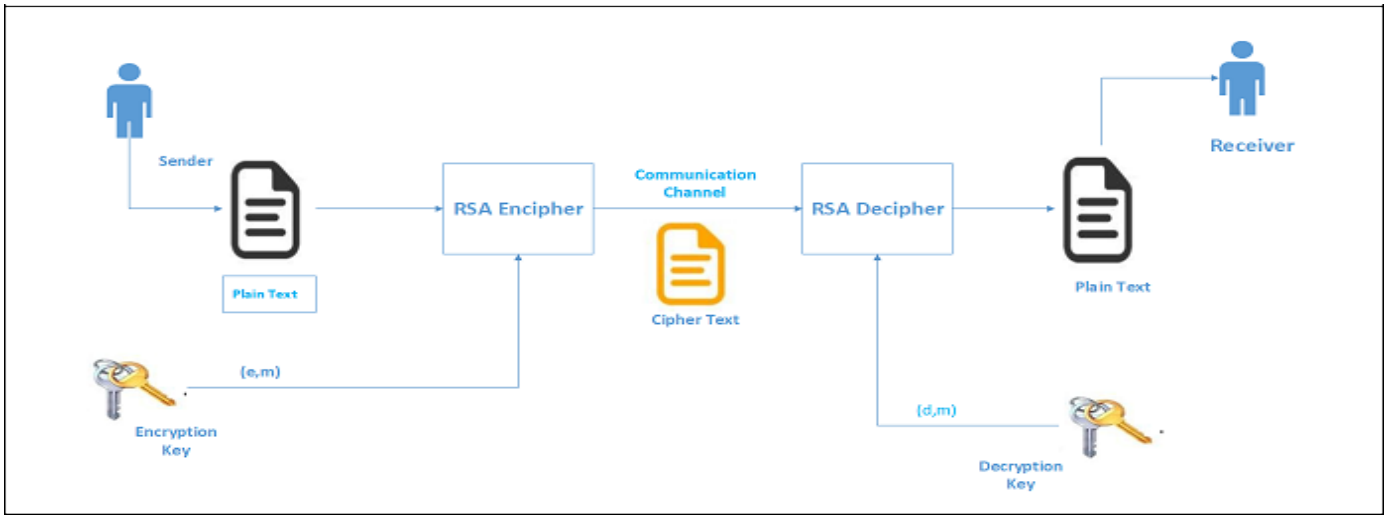

Fig. 1. Scenario of Public Key Cryptography.

II. Conceptual Framework

A. Public Key Cryptography

Public Key Cryptography is used for encryption. Fig. 1 explains the scenario of PKC; different keys are used for encryption and decryption. Hence, PKC is considered to be more secure as compared to symmetric cryptography. PKC is used in IoT domains to transport secret keys which are encrypted using asymmetric cryptography because in this case smaller blocks are encrypted and the process of encryption is needed only once[6]. These kinds of algorithms are mainly based on the RSA, ECC and Diffie Hellman (DH) for key exchange and negotiation purposes. The RSA algorithm relies on the rigid mathematical problems of prime factorization while ECC algorithm is based on the elliptic curve discrete logarithm problem and DH (Diffie Hellman) security is based on the discrete logarithm problem [33]. The Diffie Hellman is a key exchange algorithm but it suffers from the Man-in-the-Middle attack. However, the RSA algorithm provides highest security but it is computationally extensive algorithm. The ECC algorithm is widely used for the shortest length of key as compared to the RSA algorithms but these both algorithms provide the equal strength of security. These algorithms are used widely in IoT devices for the authentication purposes.

B. Basics of RSA Algorithm

RSA algorithm was named after the names of the designers who have proposed it i.e. Rivest, Shamir and Andleman so it is named as RSA algorithm. It was initially proposed in 1978 [16]. It is one of the most powerful public key algorithms used. It has been widely used in public key cryptography and PKI (Public Key Infrastructure) [8]. The mathematical base makes it most suitable to be used for the Public key cryptography. It is also used in certificate mode of security. Its mathematical background is the theorem of Euler and it relies on the integer prime factorization i.e. IFP. The whole algorithm bases on the selection of the prime numbers that are generated randomly and this is the strength of the algorithm.

In order to achieve the end-to-end security a protocol named MIKEY in [7,29] is used. It supports multiple modes of security such pre shared keys, public keys and key exchanges. It was specifically designed for real time and multimedia applications but this can also be used with the constrained devices such as sensors and actuators as it has attributes similar to the attributes of constrained devices. The Public Key Cryptography schemes provide high scalability and more resilience to attacks, which may raise the energy requirements and may involve complex computations [24, 26, 31]. However, they are more suitable when security is the major concern. In this paper, we present a secure way of communication using Asymmetric Cryptography.

The whole algorithm of RSA can be decomposed into three steps that are as follows:

1) Key Generation
2) Encryption
3) Decryption

In RSA algorithm the most computationally extensive part is the generation of key pairs because here the large prime numbers are selected such that they are not easily guessed and then the product of two large prime numbers is computed which will also a very large number as explained in Fig. 2. In fact two large prime numbers are selected and the minimum length of the modulus N is 1024 bits which is the minimum bit length in case of RSA algorithm for providing security [8].

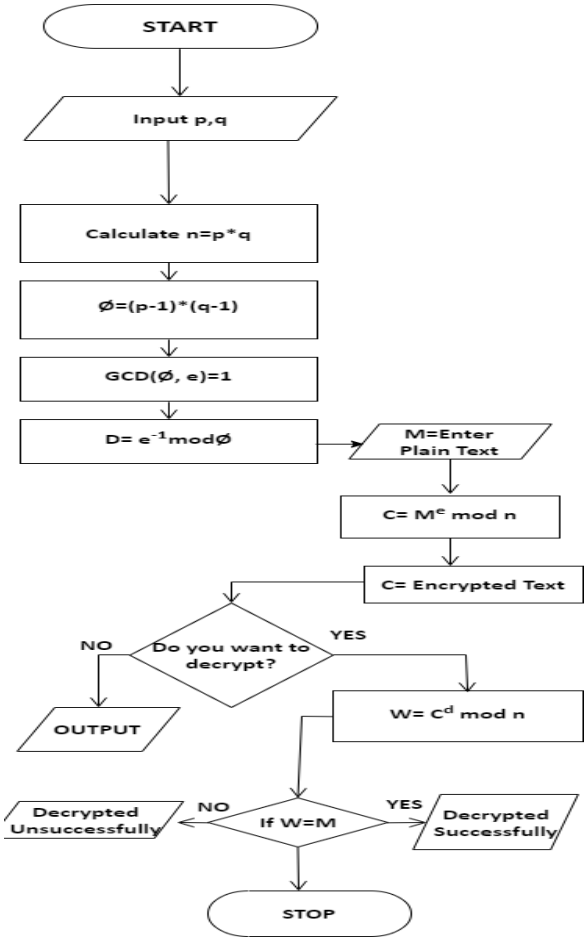

Fig. 2. The Structure of RSA Algorithm.

III. Literature Review

The Symmetric key algorithms incur less overhead as pre-shared keys are used while the public key based cryptographic solutions are more scalable and more robust to assure key distribution to the masses of devices.

CoAP (Constrained Application Protocol) has been introduced by the IETF's (Internet Engineering Task Force) the core working group to ensure the unified transmissions [21,27]. It has been designed specifically to address the issues of low power and low memory devices along with the support of multicast messages and abridged consumption of energy [19]. CoAP uses the UDP as the transport layer protocol so the reliable transmission of packets.

To ensure the reliable transmission DTLS has been introduced to implement the security such as the TLS (Transport Layer Security) we have in the traditional networks. The purpose of using the DTLS is to provide the end-to-end security at the transport layer. However, the security of DTLS is not incorporated with the application layer protocol such as CoAP and it also does not support the message-oriented approach of security so there is a need of object model or message oriented approach at the application layer. An alternative approach is to incorporate security in the CoAP using an added security option [9,30]. However, the current specifications of CoAP shows three different security modes which are used with DTLS.

1) Pre-Shared Mode PSK: In this mode of security, the devices are per-shared with the symmetric keys, which can be used to secure the communication between devices. The keys are pre-programmed with the keys so this mode of security is useful for devices that cannot support the public key cryptography algorithms.

2) Raw Public keys (RPK): In this mode of security, the devices can use PKC but they are not the PKI (public Key Infrastructure). The devices are pre-configured with asymmetric keys, which are authenticated using out of band validation. The devices can obtain the identity from the Public key and the devices also contain a list of IDs and a list of nodes it has to communicate with.

3) Certificates: In this mode of security, certificates are used for authentication called PKI (public key infrastructure) so a kind of security infrastructure should be available which is still a challenge. The certificates used for binding the security and are also signed by the common trusted store so the device also holds a list of trusted stores which can be used for the validation of certificates.

An ECC-based signcryption method was introduced in "Henriques and Vernekar [5]", the public values were bound with the public keys so this is also a certificate based approach and it relies on the binding of certificates from a server which is known as the trusted server. In this approach the computational overhead of using certificates is solved but the use of public key encryption is stagnant challenge.

In "Chen [10]" the author has proposed a scheme called symmetric security with symmetric cryptography particularly for constrained devices. It uses the symmetric key cryptography because it is lightweight for IoT devices and the author has made it scalable for a variety of devices. This scheme uses a trust Anchor that is responsible for providing and establishing the connection between client and server. This approach uses the Trust Anchor to establish the connection using symmetric key cryptography using PSK mode only.

In "Rescorla and N. Modadugu [11]" a key agreement approach is used to establish the authentication scheme for IoT and the cloud and authentication is carried out using HTTP cookies. This is ECC based method but it is used to achieve the symmetric key session. ECC algorithm is used to perform the registration phase followed by the login and authentication phase. This approach specifically provides the authentication between cloud servers and embedded devices.

In "Chavan and Nighot [12]", the author has proposed an approach to use and implement RSA cryptography for sensor nodes in smart cities. The author has proposed an efficient way to implement RSA in sensor nodes using Montgomery multiplication instead of using Chinese Remainder Theorem for implementing RSA. This approach address the security issues in sensor nodes but does not specify the way of communication and performing encryption and decryption using RSA. Moreover, it does not interact with the application layer protocol and hence it is also not scalable to a large number of IoT devices. Therefore, this cannot be used security measure in IoT environments; instead, it implements the RSA in hardware; it cannot be used in client/server communication that is the bases for IoT environment.

A. Hardware based Approaches

In "Granjal, Monteiro, and Silva [13]", the authors have proposed a model for establishing authentication key-scheme which is signature based for IoT applications. It is a complex approach as the system of establishing secure signature based keys consists of the eight phases and all the phases involve some kind of mathematical computations, which is not suitable for IoT environment. Moreover, it is designed specifically for the future IoT devices, which may have more power and memory as compared to the devices used today so this scheme is not applicable to the current scenario of IoT application. Furthermore, it also does not specify that how communication between clients and resource servers will take place after being authenticated by using the described scheme of securing IoT applications using signature-based scheme for key establishment because this scheme is proposed for providing the authentication purposes "Zhou, Liu, Tang, and Tinashe [18]".

Many approaches have been proposed so far to accelerate the hardware of IoT devices such as AES. Although these kinds of methods seem to ensure the security requirements of IoT environment but they do require devices with more memory and power so these solutions are not suitable for IoT devices that are used and deployed in the current era; we may be capable of producing and manufacturing such devices in future which may have more memory and power as compared to the devices used today "Malik, Dutta, and Granjal [2]".

In "Raza, Seitz, Sitenkov, and Selander [14]", the author has proposed a model to be used with the distributed applications of IoT. This method provides authentication based mechanism which involves many phase just to initiate the communication between client and the server. This approach consists of the registration phase followed by the login phase and the key agreement phase.

The method proposed in "Chang, Wu, and Sun [15]" provides the security measures which consists of a lot of complex computation which involves the use of RSA as well as ECC along with some hash functions and some kind of MAC also. Although this method has introduced intense security measures, but due to its complexity, it is not applicable to all types of devices in the IoT paradigm.

B. Public Key Infrastructure

The research on PKI is mostly engrossed on the compression of protocol headers such as [16, 17]. The author has proposed the deployments of PKI at the DTLS layer. The author has analyzed the headers of DTLS and established that the size of DTLS header is too long and cannot fit into single packet of IEEE 802.15.4 for providing the end-to-end security. Therefore, the author proposed a scheme of compressing the header of 6LoWPAN for DTLS and they have further claimed to reduce the number of bits to 75% in the DTLS handshake header. However, the DTLS handshake is providing the automatic key management at the transport layer, providing authentication to the server and the client using and claimed to ensure end-to-end security. The author has not assessed the PKI and even not elaborated that how public keys will be transported.

All of these schemes are addressing the security issues in IoT domain as well as providing the solutions. The Hardware based approaches are efficient to provide essential security measures but they can also be exposed and do not provide robust security. PKI based approaches use DTLS header to provide secure communication but the approaches do not elaborate that how keys will be transported. Hence, studying all these proposed models we draw conclusion that we need a solution of providing security through a mechanism that should secure communication using dynamic keys assignment.

IV. Proposed Methodology

In this section, we propose a lightweight model for communication between sender and the receiver in a constrained devices environment. The proposed model uses asymmetric cryptography to secure the communication between CoAP client and the server; it includes the Trust Manager TM for the generation of asymmetric keys dynamically. The proposed model is inspired by the S3k and

includes four phases i.e. a) Key Generation phase that is performed by the Trust Manager TM, b) request sent by the client after getting the keys from the TM, c) Resource Server RS servicing the request of client and finally d) receiving the data from RS in encrypted form and performing decryption operation at the client side. The notations used in the proposed algorithm are listed in the Table I.

TABLE I. NOTATIONS

P	Large prime numbers of bit length
Q	Large prime number of bit length
N	Product of prime numbers p and p
E	Smallest integer in the range 1<e<n
Φ(n)	Product of p-1 and q-1
D	Exponent of private key
n,e	Pair of public key
d, n	Pair of private key
P	Plain text
C	Cipher text
Mod	Modulus

A. Terminologies used in the Proposed Methodology

1) Client: Client is a machine that wants to communicate with things i.e. sensors and actuators. The user is a client but the user will communicate using a machine that can be a laptop, smartphone or desktop computer. The user can connect to the Resource Server using one of the above-mentioned devices.

2) Trust Manager: Trust Manager is a kind of application that is obliged for generating the key pair for the client. The Trust Manager will generate keys i.e. private and public keys for client and the client will use these keys for the communication purposes. Only Client can connect to the Trust Manager and the Resource is kept apart from the Trust Manager to avoid the communication overhead from the RS.

3) Resource Server: It is a kind of Raspberry pi device, which is able to communicate with the different types of sensors. It has enough power and memory resources, which make it suitable for communicating with the different types of sensors efficiently. This device has different types of specifications according to the model of the Raspberry pi [22].

a) Phase I: Key Generation: The Client sends request to the TM for connection with the Resource Server RS; the TM generates the keys that are used for the communication purpose and sends the public and the private keys to the Client machine as shown in the Fig. 3. The pseudocode for the key generation process is as follows:

1) Generate a random number r using secure Random
2) Generate p and q two big prime numbers according to the bit length
3) Calculate n and Φ(n) and select an integer e
4) Compute d such that d= e.modInverse(Φ(n)).

Fig. 3. Phase I. Key Generation Request.

b) Phase II- Request Sent By Client: In this phase the client machine that can be desktop computer, laptop or smart phone generates the request.

1) Public key = n, e
2) Private Key= n, d
3) n,e→radix(16)……..(1)
4) Append (n, e)
5) Setting payload

Key Generation Algorithm is explained in the flow chart represented in Fig. 4.

c) Phase III- Request Processed By Rs

1) GetRequest (n+e)
2) Parse Request to get n and e
3) n+e → radix(10)
4) receive status s from sensor node
5) response.s^e mod n…….(2)
6) Response back to Client.

Fig. 5 shows the computations performed for processing the request at the client side as well as the RS side.

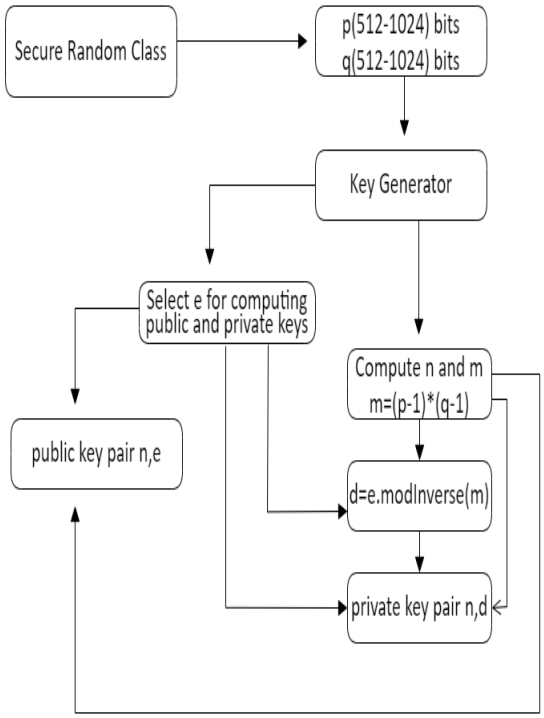

Fig. 4. Key Generation Algorithm.

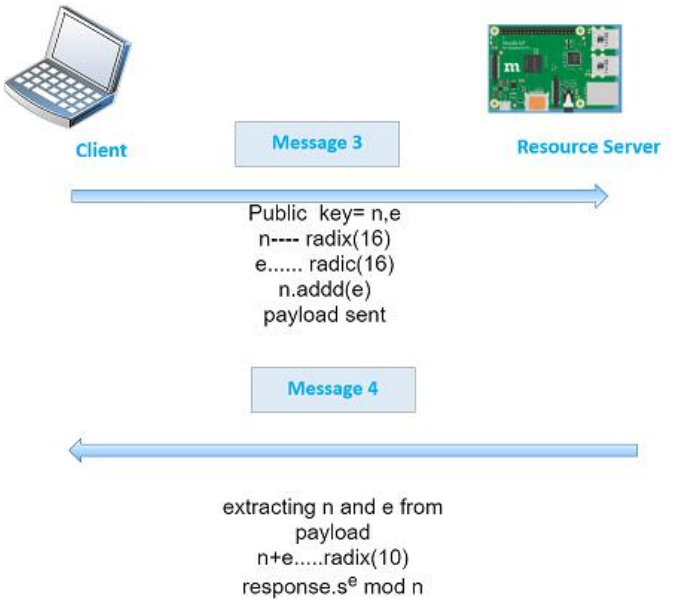

Fig. 5. Request Process Phase.

d) Phase Iv- Decryption At The Client Side: The response received is decrypted at the client side to extract the information hidden. The whole communication sent and received by the Client and the RS to compress the bits and to make it more secure to send over the channel. The pattern of flights between RS and the Client showed in the figures below.

The pattern of flights is shown between client and the RS when the client machine initiates the communication and the user from the client machine can access the devices attached to the Resource Server remotely by using the TM in a secure way. However, this solution presented here is one sided. As we know that, the RS is a kind of Raspberry Pi device and many sensors connected to the RS. The RS continuously gets the latest, updated command and takes suitable action. The above scenario applies to the critical situation where the device status matters a lot and immediately in a hospital, where intense care patient need immediate response from the doctor, so in this case, the RS can also initiate the communication and inform the client about the latest condition.

The RS would connect to the TM and request to generate the keys. The TM will generate the keys and send public and private key pair to the Resource Server. The RS sends the status of the device along with the public keys as depicted in Fig. 6.

The Client on the other end receives the status, issues the command accordingly after encrypting it with the public keys, and sends the encrypted message to the RS. The RS decrypts the message received and acts accordingly. The proposed scenario of the communication between RS and the client is explained in Fig. 7.

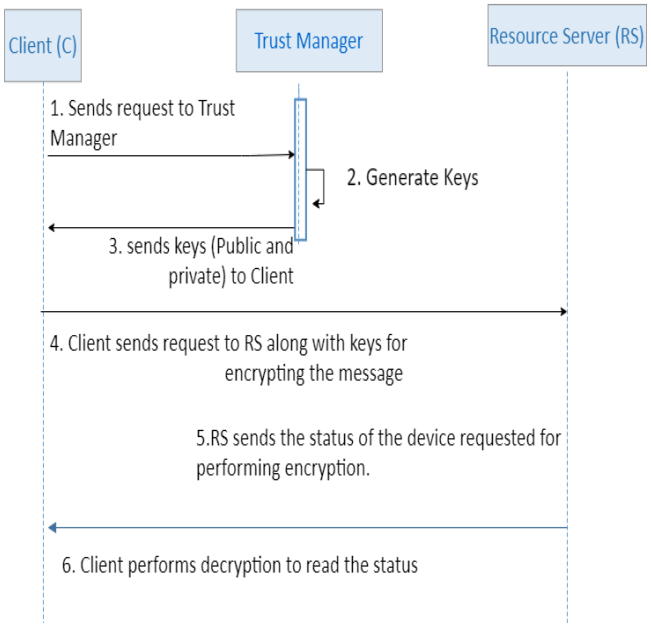

Fig. 6. Communication between Client and Resource Server.

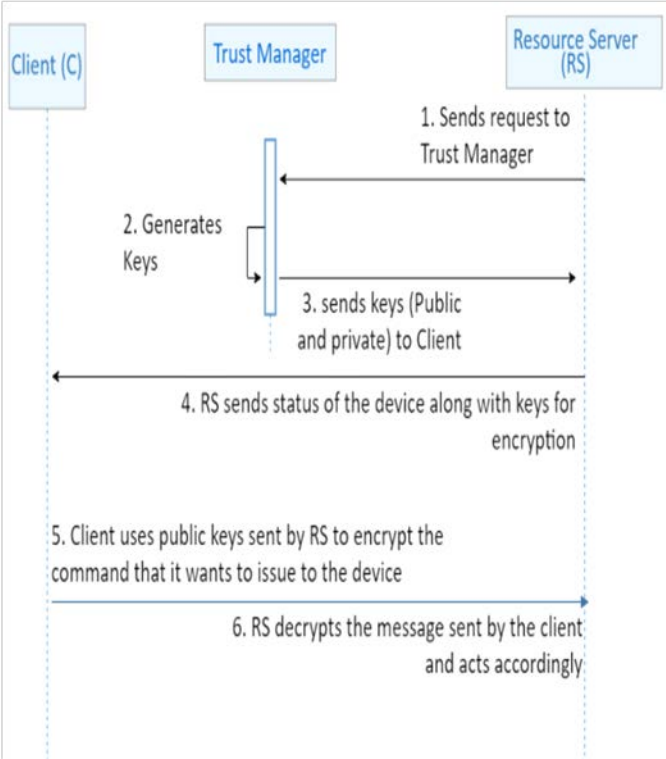

Fig. 7. Communication between Resource Server and Client.

V. Analysis of Results and Discussion

In the following section, we discuss about the implementation detail of our work, the results obtained and the analysis of results. We have simulated our proposed framework using Californium library, which is available as open source for implementing in Eclipse using Java language. We have setup the environment for the testing purpose of our methodology using the following experimental setup. We changed the data sets against different approaches, which are being used for ensuring secure communication in IoT domain. We carried out equal number of tests for each approach and obtained the results using the same machine and other specifications being same as well as described in Table II. Results are attained for time required to perform the encryption and the decryption operation as well as using the more secure to communicate.

A. Time Consumption

In this section, we have evaluated the results to compute the time utilized by our proposed framework to encrypt the data that is being sent and the decrypt the received data at the other end. We have recorded the time as when the request payload is formed and the keys for encryption are also sent along with the request; the RS receives the request, encrypt the status and sends it to the client and the client performs the decryption operation to check the status of the device. However, we assume that the RS is continuously receiving the statuses of the devices it has been attached with. We have obtained the results and then compare them with the well-known symmetric algorithms i.e. AES and the DES as they have been implemented in past. Our purpose was to reduce the time used to perform encryption and decryption along with enhancing the security. The results are shown in the following figures.

The results show that the proposed framework takes less time as compared to AES so we can say that the proposed methodology is efficient in terms of time consumption. Moreover, we recorded the time by changing the packet size and the size of the key as well, but the results still show that our proposed work is more efficient. We have also evaluated the results against DES algorithm because DES is also a well-known symmetric key algorithm. Table IV shows the observations recorded for DES and the proposed framework.

Table III shows the comparison between our proposed framework and the AES scheme.

The data set was remained constant for recording the time for AES and the proposed framework and obtained results are shown in a graph in Fig. 8.

In the previous approaches, the symmetric algorithms were used to ensure the secure communication between nodes. Moreover, these symmetric algorithms were used either by using PSK i.e. pre-shared keys or by embedding the keys in the hardware as in [8], [9], [10], [12] and [13]. However, if we want to use PKC in constrained nodes then it requires more resource for the generation of keys and keeping both the public and private keys. Our proposed methodology has used asymmetric cryptography, which is more secure and robust by ensuring that it does not create computational overhead for the constrained devices.

The corresponding graph between the proposed framework and the DES algorithm is constructed using the observation recorded in Table IV. The graph is shown in Fig. 9.

B. Security

The most important and challenging task was to secure the constrained devices to protect them from the security threats and attacks. We have implemented RSA, which is more robust, and secure. Whenever the client or RS wants to communicate with each other, the nodes send request to the Trust Manager for the generation of keys so the client and the RS do not know the keys before initiating the communication. They are assigned keys dynamically each time i.e. every time a new key pair is generated which ensures that the same key is not repeated and moreover, the communicating parties are also kept aside for the generation of keys; this makes our proposed approach more secure which is the strength of our work.

TABLE II. Experimental Setup

Parameters	Values
Transmission Bits of Data	
Sets	1024, 2048, 3072
No of tests on data sets	25

TABLE III. Comparison between Proposed Framework and AES

	Time in msec	
	Framework	**AES**
1Kb	79.96	1008.39
2Kb	287.76	914.03
3Kb	765.88	1075.64

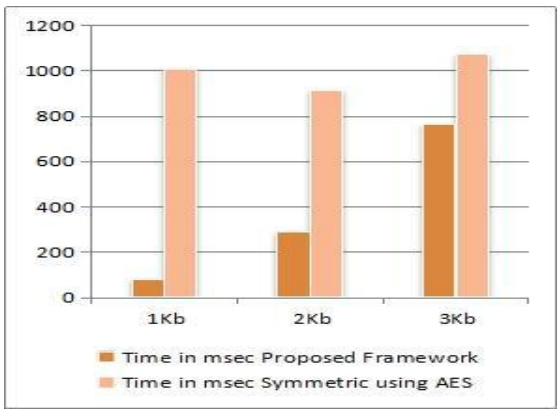

Fig. 8. Encryption and Decryption Time using Proposed Framework and AES.

TABLE IV. Comparison between Proposed Framework and DES

	Time in msec	
	Framework	**DES**
1Kb	79.96	877.86
2Kb	287.76	882.75
3Kb	765.88	897.92

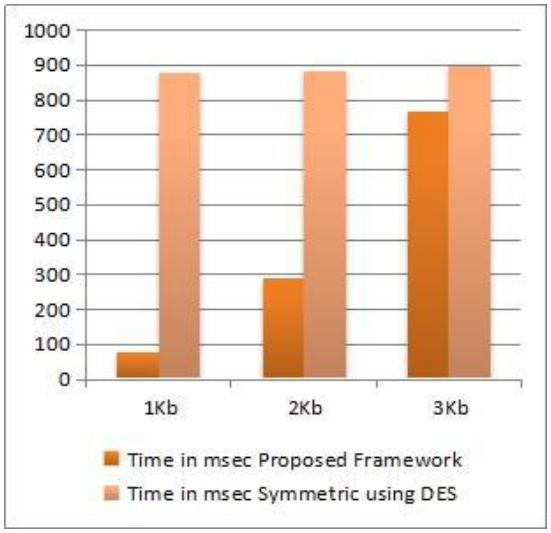

Fig. 9. Encryption and Decryption Time using Proposed Framework and DES.

C. Dynamicity

The proposed methodology ensures the dynamicity as well because the nodes connect to the Trust Manager before initiating the communication and the Trust Manager does not store the keys statically in memory, instead, it generates each time a new key pair by selecting the prime numbers randomly using cryptographically secure random number generator.

D. Computational Overhead

We demonstrate the computational cost as the computations needed to perform by the constrained devices to ensure the security. As we want to implement Asymmetric algorithm so it is computationally complex and our constrained devices cannot support such computationally complex algorithms. Our proposed framework also ensures that the constrained devices should not be involved in the computations of the complex algorithm of RSA. The RS performs the encryption whenever it wants to send the status of device in encrypted form so the only operation it has to perform is encrypting the status with the key it has received from client. The operation of encryption s^e mod n is simple; it completes this task in 12 millisecond, 17 millisecond and 39 millisecond on average for message length of 1Kb, 2Kb and 3Kb respectively. According to the specification of RS in [21], and this time does not create any kind of computational overhead for RS so it can perform this operation easily and efficiently.

VI. Conclusion

In the proposed framework, we have implemented the Asymmetric key algorithm i.e. RSA to ensure the confidentiality of the data shared between RS and the Client. The experimental results show that our proposed work is more secure as compared to the symmetric key approach used earlier. Moreover, we have used Trust Manager, which is responsible for the generation of asymmetric keys because the most computationally extensive part of RSA algorithm is to generate the keys i.e. public and private key pair. We used the Trust Manager for this purpose to reduce the computational overhead from the constrained nodes; this increases the efficiency of our proposed algorithm. Our Proposed framework is more secure because asymmetric approach is being used which provides more security than the symmetric key algorithms. We also provided a protocol suite that can be used for the communication between client and the RS. This protocol suite is applicable to both of the scenarios when the client initiates the communication as well as when the RS initiates the communication with the Client. The present work deals with the security features at the application layer and the transport layer. We have used CoAP as the application layer protocol in our work. This work can be extended in future and security features at the other layers of protocol stack of IoT can be added. Moreover, we have ensured the secure communication between client and the server i.e. Confidentiality and in future other features of security such as integrity, authentication and non-repudiation can also be added to make it more robust.

Acknowledgment

Mr. Sohail Ahmed is greatly thanked and acknowledged for his perceptive and constructive suggestions during this research work. His supportive attitude and generous willingness to give his precious time is much appreciated.

References

[1] J. D. de Hoz, J. Saldana, J. Fernandez-Navajas, and J. Ruiz-Mas, “IoTsafe, Decoupling Security from Applications for a Safer IoT,” IEEE Access, no. 1, pp. 1–1, 2019.

[2] M. Malik, M. Dutta, and J. Granjal, “A survey of Key bootstrapping protocols based on Public Key Cryptography in the Internet of Things,” IEEE Access, no. 1, pp. 1–1, 2019.

[3] M. G. Samaila, M. Neto, D. A. B. Fernandes, M. M. Freire, and P. R. M. Inácio, “Security challenges of the Internet of Things,” Internet of Things, no. 9783319507569, pp. 53–82, 2017.

[4] V. Adat and B. B. Gupta, “Security in Internet of Things: issues, challenges, taxonomy, and architecture,”Telecommun. Syst., vol. 67, no. 3, pp. 423–441, 2018.

[5] M. S. Henriques and N. K. Vernekar, “Using symmetric and asymmetric cryptography to secure communication between devices in IoT,” IEEE Int. Conf. IoT its Appl. ICIOT 2017, 2017.

[6] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, “Public Key Authentication and Key Agreement in IoT Devices with Minimal Airtime Consumption,” IEEE Embed. Syst. Lett., vol. 9, no. 1, pp. 1–4, 2017.

[7] K. T. Nguyen, N. Oualha, and M. Laurent, “Novel lightweight signcryption-based key distribution mechanisms for MIKEY,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9895 LNCS, pp. 19–34, 2016.

[8] G. Singh, “A Study of Encryption Algorithms (RSA , DES , 3DES and AES) for Information Security,” Int. J. Comput. Appl., vol. 67, no. 19, pp. 33–38, 2013.

[9] R. I. Emori, “Scale models of automobile collisions with breakaway obstacles - Investigation indicates that scale models can be used to show the motion of breakaway signposts and lightposts after being struck by automobiles,” Exp. Mech., vol. 13, no. 2, pp. 64–69, 1973.

[10] X. Chen, “Constrained Application Protocol for Internet of Things,” Wirel. Mob. Netw., vol. 857, pp. 1–12, 2014.

[11] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security Version 1.2,” RFC 6347, 2012.

[12] A. A. Chavan and M. K. Nighot, “Secure CoAP Using Enhanced DTLS for Internet of Things,” Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ., vol. 3297, no. 12, pp. 7601–7608, 2007.

[13] J. Granjal, E. Monteiro, and J. S. Silva, “Application-layer security for the WoT: Extending CoAP to support end-to-end message security for

internet-integrated sensing applications," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7889 LNCS, pp. 140–153, 2013.

[14] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K:Scalable Security with Symmetric Keys - DTLS Key Establishment for the Internet of Things," IEEE Trans. Autom. Sci. Eng., vol. 13, no. 3, pp. 1270–1280, 2016.

[15] C. C. Chang, H. L. Wu, and C. Y. Sun, "Notes on 'Secure authentication scheme for IoT and cloud servers,'"Pervasive Mob. Comput., vol. 38, pp. 275–278, 2017.

[16] L. Qiu, Z. Liu, G. C. Geovandro, and H. Seo, "Implementing RSA for sensor nodes in smart cities," Pers. Ubiquitous Comput., vol. 21, no. 5, pp. 807–813, 2017.

[17] S. Challa et al., "Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications," IEEE Access, vol. 2016, no. 2016, pp. 1–16, 2017.

[18] Y. Zhou, T. Liu, F. Tang, and M. Tinashe, "An Unlinkable Authentication Scheme for Distributed IoT Application," IEEE Access, vol. 7, no. c, pp. 14757–14766, 2019.

[19] S. Raza, T. Helgason, P. Papadimitratos, and T. Voigt, "SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things,"Futur. Gener. Comput. Syst., vol. 77, pp. 40–51, 2017.

[20] S. Raza, D. Trabalza, and T. Voigt, "6LoWPAN compressed DTLS for CoAP," Proc. - IEEE Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2012, pp. 287–289, 2012.

[21] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt "Lithe: Lightweight secure CoAP for the internet of things,"IEEE Sens. J., vol. 13, no. 10, pp. 3711–3720, 2013.

[22] H. H. Hadwan and Y. P. Reddy, "Smart Home Control by using Raspberry Pi & Arduino UNO," Int. J. Adv. Res. Comput. Commun. Eng., vol. 5, no. 4, pp. 283–288, 2016.

[23] C. Hennebert and J. Dos Santos, "Security protocols and privacy issues into 6LoWPAN stack: A synthesis," IEEE Internet Things J., vol. 1, no. 5, pp. 384–398, 2014.

[24] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI," J. Sens. Actuator Networks, vol. 2, no. 3, pp. 589–613, 2013.

[25] IEEE Computer Society, IEEE Standard Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), vol. 2011, no. September. 2011.

[26] Ahmad Mujtaba, "IPv6 over Low-power Personal Area Network (6LoWPAN)," RFC 4919, vol. 67, no. 6, pp. 14– 21, 2007.

[27] P. K. Kamma, C. R. Palla, U. R. Nelakuditi, and R. S. Yarrabothu, "Design and implementation of 6LoWPAN border router," IFIP Int. Conf. Wirel. Opt. Commun. Networks, WOCN, vol. 2016-Novem, pp. 2–6, 2016.

[28] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,"IEEE Internet Things J., vol. 4, no. 5, pp. 1125–1142, 2017.

[29] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the internet of things: A standardization perspective," IEEE Internet Things J., vol. 1, no. 3, pp. 265–275, 2014.

[30] K. K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities," IEEE Trans. Ind. Informatics, vol. 14, no. 8, pp. 3567–3569, 2018.

[31] N. Sklavos and I. D. Zaharakis, "Cryptography and security in internet of things (IoTs): Models, schemes, and implementations," 2016 8th IFIP Int. Conf. New Technol. Mobil. Secur. NTMS 2016, pp. 3–4, 2016.

[32] M. G. Samaila, M. Neto, D. A. B. Fernandes, M. M. Freire, and P. R. M. Inácio, "Security challenges of the Internet of Things," Internet of Things, no. 9783319507569, pp. 53–82, 2017.

[33] O. P. Pinol, S. Raza, J. Eriksson, and T. Voigt, "BSD-based elliptic curve cryptography for the open Internet of Things,"2015 7th Int. Conf. New Technol. Mobil. Secur. - Proc. NTMS 2015 Conf. Work., 2015.