Indian Journal of Science and Technology, *Vol 10(1), DOI: 10.17485/ijst/2017/v10i1/109387, January 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Design of Secured and Intelligent Architecture for Security in Perceptual Layer of the Internet of Things

V. Kamalakannan and S. Tamilselvan

Department of Electronics and Communication Engineering, Pondicherry Engineering College, Pillaichavady – 605014, Puducherry, India

Abstract

Background: Today's developing era data and information security plays an important role in unsecured communication between Internet of Things (IoT) elements. As the technologies are developing huge amount of information are exchanged with each other which increases the demand of a secure encryption standards. **Objectives**: To protect the information transmission between nodes in the perceptual layer of IoT, security is to be enhanced such that the Men in Middle (MIM) attack of information by intruders in the communication link is not possible. **Methods/Statistical analysis:** The security method is implemented using the concept of symmetric, asymmetric and hash algorithms between the sender and receiver. Elliptic Curve Cryptography (ECC) is performed for encrypting and sharing the Advanced Encryption Standard (AES) secret key between the sender and receiver. Elliptic Curve digital signature are added with the encrypted data for validation in transmission. Secured Hash Algorithm (SHA) is considered for generating Elliptical Curve Digital Signature Algorithm (ECDSA). All the algorithms are combined to enhance the security. **Findings:** Elliptic Curve Cryptography is generally applied for encrypting and sharing the secret key between the sender and receiver when Advanced Encryption Standard is used for encryption/decryption of data. A highly efficient architectures for asymmetric key encryption is applied with symmetric key encryption for encrypting the data transmitted between the devices present in the perceptual layers in the IoT. Due to security issues NIST selected SHA-3 Keccak-f[1600] algorithm. An authentication process based on ECC is applied to perceptual layer involving initialization and authentication phase apart from encryption and signature generation. Sender combines the encrypted data and key with signature together before transmitting to the receiver. The receiver verifies the signature and decrypts the key and obtains original information. Hence it is very difficult for the man in middle to access the information exchanged between the Perceptual Layer devices. **Application/Improvements:** This generally doesn't require an innovative technique but an approach to develop the method implemented successfully in the perceptual layer of Internet of Things. The key is encrypted and decrypted using a novel ECC algorithm with improved ECDSA.

Keywords: Elliptical Curve Cryptography (ECC), Elliptic Curve Digital Signature Algorithm (EDSA), Internet of Things (IoT), Keccak, Perception Layer, Secured Hash Algorithm (SHA)

1. Introduction

The data protection is an important issues when two systems are connected with each other. The security protocols are evolving parallel with the network evolutions. Security at the device as well as network level is crucial in Internet of Things operations.[1] At present networks of things have made a great breakthrough but it's been with us in different forms for many years, but it's the thing now. it is the ability to connect, communicate, manage networked and automated devices through the internet. This transition has raised the alarm of security

**Author for correspondence*

as we are depending on these intelligent, independent devices. To protect these devices cryptographic algorithms are necessary now. One of the Internet of Things elements is wireless sensor network.[2] Basic security of sensor network consists of key distribution, intrusion detection and encryption techniques. To enhance the security in wireless sensor network different security mechanisms are integrated together.[5] Data security can be provided using symmetric key encryption technique or asymmetric key encryption technique. The symmetric key encryption technique are simple having difficulty in maintaining keys as the key can be cracked using brute attacks. The asymmetric key encryption technique provides higher security but don't have the efficiency, thereby used to maintain key. A novel version of data security in Internet of Things is implemented based Elliptical Curve Cryptography-Secure Hash Algorithm hybrid encryption methodology. An introduction to IoT is provided in section 2 and the proposed security model is explained in section 3, Cryptography algorithms applied are discussed in section 4 is to encrypt/decrypt data. Finally conclusion is arrived in Section 5 and is followed by references.

2. Internet of Things

Internet of Things is superset of internet of network of computers. Nowadays focus is on devices with constrained resources (memory and energy) connected in Internet of Things[1]. IoT application data might be transmitted in plaintext for many reasons. One common reason is the poor design decision to treat only the most obviously private user information as sensitive. In a home automation system, sensor data like temperature readings might not be considered sensitive[2]. However, an eavesdropper monitoring such readings temporarily might be able to infer whether a user is at home by tracking sudden temperature changes or significant deviations from outside conditions (for instance, if the user starts the air conditioner). Another reason for transmitting unprotected data is the choice of hardware. Many IoT products are inexpensive components with limited memory and computational resources. Such devices might be unable to support the computationally intense cryptographic functions of public-key cryptography. Hence, they might be incapable of supporting the SSL/TLS protocol, which is the industry-standard transport protection mechanism. In our use case, even if system designers considered the privacy implications of unencrypted data, they would have limited encryption options because of the hardware platform. As a result, system designers have two choices: create their own lightweight security protocols or implement modified, stripped-down versions of well-known security protocols[3]. The first choice runs the risk that the new mechanism will be vulnerable in practice and incur significant development costs. However, the second choice carries a great likelihood of a security vulnerability. Thus, custom security schemes or hardware-adapted protocol implementations might result in data being transmitted without meaningful protection. Evidence suggests that such a modified protocol would run efficiently even on small single-board computers. Most security issues such as eaves dropping, message tampering, etc., existing in internet also exists in Internet of Things. The security architecture of Internet of Things is generally seen as four layer architecture consists of perceptual layer, network layer, support layer and application layeras shown in figure 1.

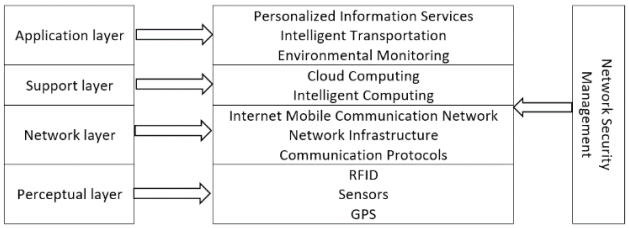

Figure 1. IoT Security Architecture.

The lowest layer in the Internet of Things, Perception layer captures and identifies the device's information with the help of RFID tags and Sensors and passes the information to the network layer[4]. Application layer processes huge data by segregating it as it is available from different types of sources. In the perceptual layer, wireless sensor node forms an Adhoc network. Wireless Sensor Network consists of sensor devices which are generally of low cost and small in size. They have limited storage and energy resources. Lot of security challenges have to be met which are crucial in there operation. They have limited storage and energy resources. Lot of security challenges have to be met which are crucial in there operation. The threat that are present in the perceptual layer are due to the limited computational resources and in authentication due to distributed environment. Thereby the intruders can access the data easily in the Perceptual Layer devices. Hence the security has to be implemented to overcome these issues in the IoT.

3. Proposed Security Model

In the proposed model AES and ECC are compared and analyzed and found that in AES speed of operation is good compared to other symmetric encryption techniques, but the drawback is with respect to the key management which is unsecured[5-7]. Thereby asymmetric key encryption technique, ECC is applied to solve the key management and sharing. This paper ECC algorithm is applied for key management by encrypting and sharing the AES key with the sender and receiver. The AES performs the encryption/decryption of the data. The data security is enhanced by applying additional ECDSA for signature generation and verification. The hash function in ECDSA uses the SHA-3 Keccak algorithm. These three algorithms mutually guarantees data security and these algorithms are combined in transmitter shown in the figure 2 for transmission of data after performing encryptions and the reverse process is performed in in the receiver are represented in figure 3.

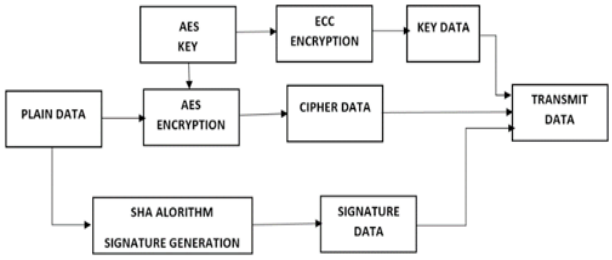

Figure 2. Transmission of data.

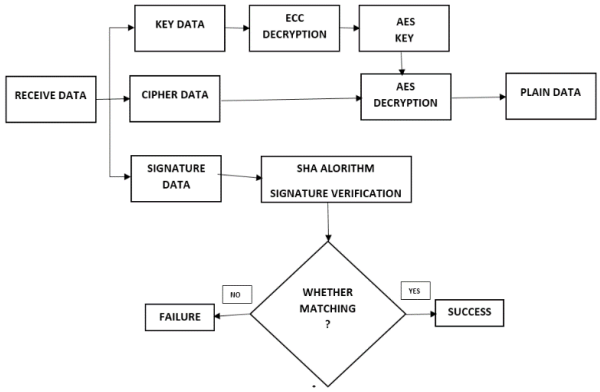

Figure 3. Reception of data.

The AES encrypts the plain text to cipher text using secret key. This is shared between the sender and receiver using Elliptic Curve Diffie Hellman (ECDH) Key Exchange algorithm. The SHA algorithm generates the signature data to be added with the cipher text. The cipher text, key data and signature data are transmitted to the receiver. The receiver performs the inverse operation of decryption of cipher text and Key. With signature verification decision is arrived whether the data received successfully or failure in transmission. The proposed model consists of initialization stage and authorization stage.

In the proposed model different processes are involved to increase the security of the perceptual layer in the IoT. This proposed scheme plays an important role by providing security in the communication between the elements present in the perceptual layer by applying the methods such as initialization stage and authorization stage.

3.1 Initialization stage

The initialization stage, a common generator point 'G' is computed to generate P, 2P….. kP which are to be mapped with key data. Sender and receiver generates their generator points G_S and G_R individually based on the respective Elliptical Curve equations. The sender and receiver exchange information between them and generate a common base point as shown in the Figure 4.

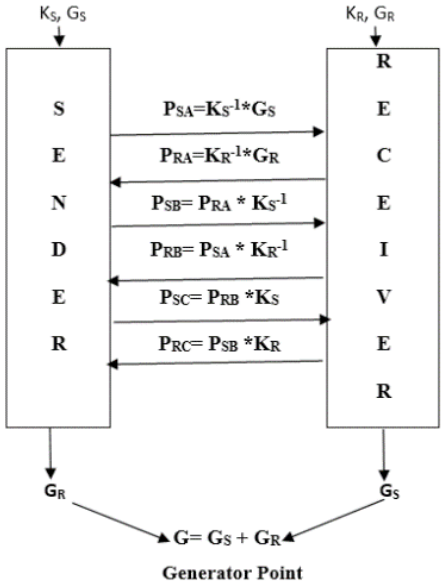

Figure 4. Initialization stage.

This base point or generator point is used for mapping in cryptography algorithm. The sender randomly selects an integer $K_S \in (1, 2\ldots\ldots p_1)$ and receiver also randomly selects an integer $K_R \in (1, 2\ldots\ldots p_2)$ which are referred as the secret keys and then computes the inverses of K_S and K_R. The sender later generates public key P_{SA} and using the equation 1.

$$P_{SA} = K_S^{-1} * G_S \quad (1)$$

Similarly receiver also generates public key P_{RA} using the equation 2

$$P_{RA} = K_R^{-1} * G_R \quad (2)$$

Both the public keys P_{SA} and P_{RA} are shared with each other by multiplying it with the inverses of private keys. The sender generates the key to be transmitted to the receiver using the public key of receiver and is given in the equation 3.

$$P_{SB} = K_R^{-1} * G_R * K_S^{-1} \quad (3)$$

The receiver generates the key to be transmitted to ender using the sender's public key and is given in the equation 4.

$$P_{RB} = K_S^{-1} * G_S * K_R^{-1} \quad (4)$$

The private keys of the sender and the receiver are multiplied with the keys P_{SB} and P_{RB} to generate P_{SC} and P_{RC} given in equation 5 and equation 6.

$$P_{SC}=P_{RB} * K_S=K_S^{-1} * G_S * K_R^{-1} * K_S = G_S * K_R^{-1} \quad (5)$$

$$P_{RC}=P_{SB} * K_R=K_R^{-1} * G_R * K_S^{-1} * K_R = G_R * K_S^{-1} \quad (6)$$

When P_{SC} received by receiver and P_{RC} received by sender, then they are multiplied with K_S and K_R to obtain G_R and G_S. At transmitter the generator obtained is expressed in equation 7.

$$P_{RC} * K_S=K_S^{-1} * G_R * K_S=G_R \quad (7)$$

At receiver the generator obtained is expressed in equation 8.

$$P_{SC} * K_R=K_R^{-1} * G_S * K_R=G_S \quad (8)$$

Now sender and receiver both have the generator points G_S and G_R and both generate a common generator point by adding the two generator points given in equation 9.

$$G = G_S + G_R \quad (9)$$

3.2 Authorization stage

In this stage a random number is generated by sender and receiver represented a K_S and K_R. The public key of sender and receiver are P_S and P_R. The sender considers public key P_R and private key K_S generates P_{SR} and transmits it to receiver. The receiver similarly considering public key P_S and private key K_R generates P_{RS} to be transmitted to sender. This process is best visualized by the figure 5.

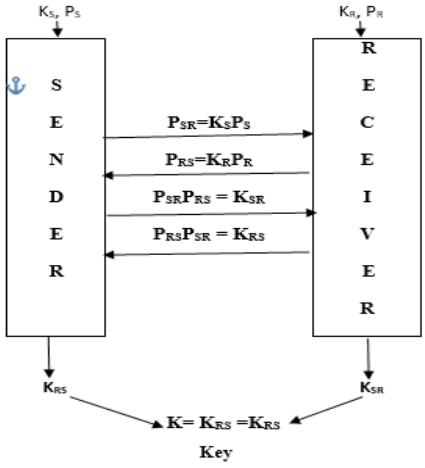

Figure 5. Authorization Stage.

These two values are exchanged between the sender and the receiver to perform the authentication process. The sender then generates K_{SR} in equation 10.

$$K_{SR} = P_{SR} * P_{RS} \quad (10)$$

The receiver generates K_{RS} in equation 11.

$$K_{RS} = P_{RS} * P_{SR} \quad (11)$$

The key is generated and applied for encryption/decryption and signature generation/signature verification is specified in equation 12.

$$K= K_{RS} = K_{SR} \quad (12)$$

3.3 Matrix Mapping Methodology

A modified mapping method based on matrices and elliptic curve is proposed. The alphanumeric characters of the key are mapped on to the points of the elliptic curve.

Here it is assumed that both the sender of the message and the receiver of the message to know the following relationships from the initialization stage and authorization stage:

S: the set of the mapping points generated by the proposed algorithm.

A: the encoded matrix is constructed in such a way that: A is nonsingular and has only integer entries.

A^{-1}: Matrix inverse of A. In our case, we select the entries of A in such a manner that A is nonsingular diagonal (for simplicity).

It is defined the mapping F: C⟶ ⟶ S, as specified rule of correspondence between sets of symbols which are composed message and a set of points on elliptic curve.

It is assumed that the embedding system m ⟶ ⟶ P_m , which imbed the original message on an elliptic curve E exist.

Step 1: Transform the alphanumeric characters into points on elliptic curve.

$[P_1(x_1, y_1), P_2(x_2, y_2) \ldots P_u(x_u, y_u)]$. The original message M of length u is taken for consideration. If u is not divided by 4, then the points have been padded with Ω Ω which represent space.

Step 2: Create the matrix of 3 *x r* shown in equation 14 with entries are points on EC (Step 1):

$$M=\begin{vmatrix} P_1 & P_2 & P_3 & \ldots\ldots & P_r \\ P_{r+1} & P_{r+2} & P_{r+3} & \ldots\ldots & P_s \\ P_{s+1} & P_{s+2} & P_{s+3} & \ldots\ldots & P_t \\ P_{t+1} & P_{t+2} & P_{t+3} & \ldots\ldots & P_u \end{vmatrix} \quad (14)$$

With $= \frac{u}{4}, s = \frac{2u}{4}, r = \frac{3u}{4}$ and $s = \frac{4u}{4}$

Step 3: Choose a matrix A of order 4 x 4in equation 15, such that A is non-singular diagonal.

$$A=\begin{vmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{vmatrix} \quad (15)$$

Then, using addition and doubling of points to compute Q = AM represented in equation 16.

$$Q=\begin{vmatrix} P_1 & P_2 & P_3 & \ldots\ldots & P_r \\ P_{r+1} & P_{r+2} & P_{r+3} & \ldots\ldots & P_s \\ P_{s+1} & P_{s+2} & P_{s+3} & \ldots\ldots & P_t \\ P_{t+1} & P_{t+2} & P_{t+3} & \ldots\ldots & P_u \end{vmatrix} \quad (16)$$

Step 4: The resulting set of points is: $[Q_1(x_1, y_1), Q_2(x_2, y_2) \ldots Q_u(x_u, y_u)]$ Once the mapping of the all-alphanumeric characters onto the curve is completed, these points are encrypted by using elliptic curve encryption technique which are transmitted through an insecure channel. The message is retrieved from the encoded data by using the elliptic curve decryption technique and the inverse of matrix.

4. Cryptographic Algorithms

Cryptographic algorithm are used provide achieve security in data transmission and are of generally categorized as symmetric key encryption using private key for cryptography, asymmetric key encryption using public and private key for cryptography and hash encryption algorithms[16]. Symmetric key cryptographic algorithm are generally having speed of execution faster than asymmetric key encryption methods. Asymmetric keys are known as public key and are used in session key exchanges between sender and receiver whereas symmetric key are known as private key and are used for encrypting data in communication[38]. The hash encryption generates a fixed size of data from variable size of data blocks and are much stronger against the brute force attacks, therefore they are applied for signature generation and verification. These three concepts of cryptography are explained.

4.1 Advanced Encryption Standard

The algorithm for Advanced Encryption Standard is given in figure 6. The hardware implementation for the encryption as well as decryption has very low complexity[8]. This particularly, is an advantage for WSNs given their stringent power requirements. For encrypting and decrypting process in this standard 4 byte oriented transformation steps are followed. They are byte substitution using a substitution table (S Box), shifting rows of state array, mixing the data within each column of the state array and adding a round key to state[9-11]. In the encryption process round key is added initially, once the round key is added the round function is repeated 14 times for 256 bit key size and in the last round mix column operation is not applicable[12-14]. Advanced Encryption Standard decryption is the inverse operation of each of transformation but is not identical. The decryption process is basically inverse of each transformation and has inverse-shift row, inverse-byte sub, add round key and inverse mix column transformations[15,16]. The sequence of transformations in encryption is different from the sequence of transformation in decryption with the same key schedule. In the decryption process inverse-shift row, inverse-byte sub are interchanged with add round key and inverse mix column interchange. There is a version of decryption process which is similar to encryption process with changes in the key schedule. Hence from the cipher text the information is obtained.

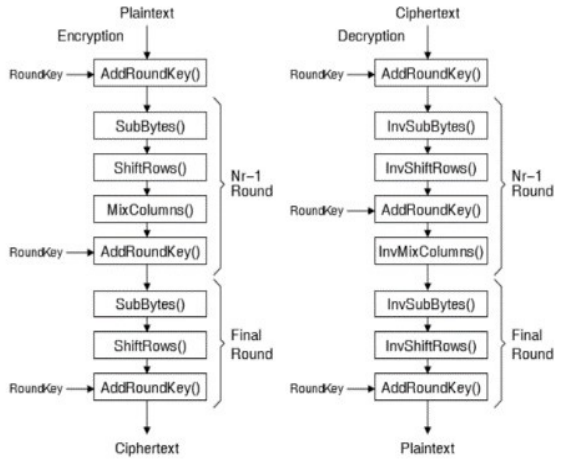

Figure 6. The Advanced Encryption Standard algorithm.

4.2 Novel Elliptic Curves Cryptography Algorithm

Most of the public key encryption techniques depend on finite mathematical analysis[17-19]. The finite field are generally prime field and binary field. The strength of the public key encryption using Elliptical curve cryptography depend on the complexity in solving elliptic curve discrete logarithmic problem (ECDLP)[20].The features of Elliptical curve cryptography, such as smaller key sizes and faster implementation results in selecting Elliptical Curve Cryptography for the security of data. Elliptic curve cryptography (ECC) is used in WSN because of smaller key compared to RSA. In Elliptical Curve Cryptography algorithm, the Generator point can be breached therefore a technique of hiding the generator point is designed to solve the MIM attack[21-25]. To rectify problem Elliptical Curve Cryptography is implemented with a Hidden generator point. This addresses the issue, when there is no Certificate Authority (CA) for Generator point sharing finalization stage process is performed. The Elliptic curve usually consists of a base point and is represented by Weierstrass Equation[26-30]. Which is later again simplified to get a simple equations for prime field and binary field. An elliptic curve E over a finite field F_q is specified by equation 17.

$$y^2=x^3+ax+b \text{ mop } q \quad (17)$$

Where a, b $\in$ $\in$ F_q and $4a^3+27b^2 \neq \neq 0$ (mod q). The EC generally acts as key distribution, encryption/decryption and digital signature algorithm

The method for encryption and decryption are as follows: From the initialization and Authentication stage we get equation 18 as

$$K_{RS}G= (K_1, K_2) = (K_x, K_y) \quad (18)$$

The message M to be transmitted after matrix mapping is (Qx_i, Qy_i)

The cipher text $(C1_i, C2_i)$ is computed by equation 19 and equation 20

$$C1_i = (Qy_iK_x + Qx_i) \text{ mod } q \quad (19)$$

$$C2_i = (Qy_i + Qy_iK_xK_y + Qx_iK_y) \text{ mod } q \quad (20)$$

The cipher text $(C1_i, C2_i)$ is decrypted using equation 21 and equation 22

$$Q2_i = (C2_i - K_yC1_i) \text{ mod } q \quad (21)$$

$$Q1_i = (C1_i - Qy_iK_x) \text{ mod } q \quad (22)$$

The proof of the decryption process is provided below

$$\begin{aligned} Qy_i &= C2_i - K_y\, C1_i \\ &= (Qy_i + Qy_iK_xK_y + Qx_iK_y) - K_y\,(Q_{yi}K_x + Qx_i) \text{ (mod q)} \\ &= Qy_i + Qy_iK_xK_y + Qx_iK_y - Q_{yi}K_{xi}K_{yi} - Qx_iK_y() \\ &= Qy_i \end{aligned}$$

$$\begin{aligned} Qx_i &= C1_i - Q_{yi}K_x \\ &= Q_{yi}K_x + Q_{xi} - Q_{yi}K_x \;= Qx_i \end{aligned}$$

4.3 Improved Elliptic Curve Digital Signature Algorithm

In this section, a very influential signature scheme based on Elliptical Curve Cryptosystem depends on the complexity of discrete logarithms over a finite field. The US Government's National Institute of Standards and Technology (NIST) proposed an algorithm for digital signatures known as DSA[36]. The DSA has become the US Federal Information Processing Standard186 (FIPS186). The DSA does not include key exchanges, and cannot be used for key distribution and encryption[37]. Digital signatures are generally used to verify the data of the sender considering data as a string of binary digits using the private key and public key of each use[39]. Private Key is used in signature generation by the sender and Public keys known by everyone, verifies the signature of a sender. Digital signature algorithms can be applied to data storage application requiring the integrity and originality of data. Digital signature are generated using Digital Signature Algorithm, the RSA algorithm, and the Elliptic Curve Digital Signature Algorithm (ECDSA)[40]. The signature generation process uses hash function to obtain a reduced data known as message digest. The hash function is also used in the verification process which is the

terms represented as the Secure Hash Algorithm (SHA)[40]. Signing the message digest improves the efficiency of the process The Elliptical Curve Digital Signature Algorithm module is represented in figure 7.

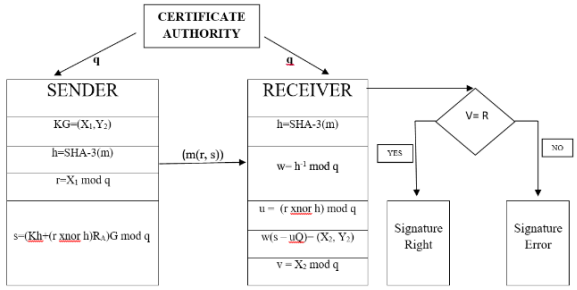

Figure 7. Signature generation and verification.

In this variant there is no need in finding inverse in both key generation and signing phase. This scheme embeds the information of signature into a point on the ellipse.

Steps involved in key pair generation:

Let A be the signatory for a message M. Entity A performs the following steps to generate a public and private key:

Step 1: Select a unique and unpredictable integer, R_A, in the interval [1, q-1]

Step 2: Q = (R_AG mod q)

Step 3: Sender A's private key is R_A

Step 4: Sender A's public key is the R_AG

Steps involved in Signature Generation:

Using As private key, A generates the signature for message M using the following steps:

Step 1: Select a unique and unpredictable integer k in the interval [1, q-1]

Step 2: KG =(x_1,y_1), where x_1,y_1 is an integer

Step 3: r = x_1 mod q; If r = 0, then go to step 1

Step 4: h = H(M), where H is the KeecakSHA-3.

Step 5: s = (Kh + (r xnor h)R_A)G mod q

Step 6: If s = 0, then go to step1

Step 7: The signature of A for message M is the pair (r, s)

Steps involved in Signature Verification:

The receiver B can verify the authenticity of A's signature (r, s) for message M by performing the following:

Step 1: Obtain signatory A's public key

Step 1: Verify that values r and s are in the interval [1,q-1]

Step 1: h = H(M), where H is the same secure hash algorithm used by A

Step 1: w = h^{-1} mod n

Step 1: u = (r xnor h) mod q

Step 1: w(s - uQ) = (x_2, y_2)

Step 1: v = x_2 mod q

Step 1: The signature for message M is verified only if v = r

Proof of the scheme

Signature send by A to B is (r, s) and s can be generated only by because only A knows its private key RA.

- s = (Kh + (r xnor h)R_A)Gmodn
- s = (Kh +uR_A) Q
- sw = KG +uwQ
- KG = sw - uwQ= w (s-uQ) = (x_2, y_2)

Therefore (x_1, y_1) = (x_2, y_2), we know that r = x_1 mod q and v = x_2 mod q, thus v = r.

5. Conclusion

Most of the Internet of Things elements such as wireless sensors are inexpensive with limited resources and memory. Thereby light weight security protocols have to be used or modify the known security protocols. Hence the hybrid encryption/decryption module based on Advanced Encryption Standard-Elliptical Curve Cryptography-Secure Hash Algorithm architectures have been implemented for Security in Perceptual Layer of the Internet of Things. The module is secure against few potential attacks thereby providing better security requirements.In this a brief overview of symmetric and asymmetric encryption algorithm are applied for encryption and decryption of data where the strength of encryption depends on its key. The key is encrypted and decrypted using a novel Elliptic curve cryptography algorithm with improved ECDSA. It is clearly evident from the above that the security level is enhanced and it is not easy for the man in middle to access the information exchanged between the Perceptual Layer devices.

6. References

1. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems. 2013; (29):1645–60.
2. Vermesan O, Friess P. Internet of Things from Research and Innovation to Market Deployment. River Publishers Series in Communications. 2014. p. 8–15.

3. Aggarwal M, Singh M. Smart city based on NDNoT: The future of IoT. Indian Journal of Science and Technology. 2016 Sep; 9(36):1–8. Doi: 10.17485/ijst/2016/v9i36/89557.
4. Bormann C, Ersue M, Keranen A. Terminology for Constrained-Node Networks. RFC 7228 (Informational), C. Internet Engineering Task Force, May 2014. [Online]. Available from: http://www.ietf.org/rfc/rfc7228.txt
5. Nithya S, George E, Raj P. Survey on Asymmetric key Cryptography Algorithms. Journal of Advanced Computing Technologies. 2014; 2(1).
6. Rajam STR, Kumar SBR. Enhanced elliptic curve cryptography. Indian Journal of Science and Technology. 2015 Oct; 8(26):1–6. Doi: 10.17485/ijst/2015/v8i26/80444.
7. Diffie W, Hellman M. New Directions in Cryptography. IEEE Transactions on Information Theory. 1976; (22):644–54.
8. AES, Available from: http://www.nist.gov/CryptoToolkit
9. Daemen J, Rijmen V. AES Proposal: Rijndael, AES Algorithm. Submission 1999, available at '8'.
10. Agrawal H, Sharma M. Implementation and analysis of various symmetric cryptosystems. Indian Journal of Science and Technology. 2010 Dec; 3(12):1173–6.
11. Menezes A, van Oorschot P, Vanstone S. Handbook of Applied Cryptography, CRC Press, New York. 1997; 81–3.
12. Amruta R, Dumane N, Narole G, Wanjari P. Design of advanced encryption standard on soft-core processor. World Conference on Futuristic Trends in Research and Innovation for Social Welfare. 2016. p. 1–5.
13. Pendli V, Pathuri M, Yandrathi S, Razaque A. Improvising performance of Advanced Encryption Standard algorithm. Proceedings of Second International Conference on Mobile and Secure Services (MobiSecServ). 2016. p. 1–5.
14. Garcia DF. Performance Evaluation of Advanced Encryptio n Standard Algorithm. Proceedings of Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI). 2015. p. 247–52.
15. Joshi A, Dakhole PK, Thatere A. Implementation of S-Box for Advanced Encryption Standard. Proceedings of IEEE International Conference on Engineering and Technology (ICETECH). 2015. p. 1–5.
16. Miller VS. Use of elliptic curves in cryptography. Advanced in Cryptology, Proceedings of Crypto85, Lecture note in Computer Science, Springer Verlag. 1986; 417–26.
17. Koblitz N. Elliptic curve cryptosystem. Mathematics of Computation. 1987; (48):203–9.
18. Shau PK, Chhotray RK, Jena G, Pattnaik S. An Implementation of Elliptic Curve Cryptography. International Journal of Engineering Research and Technology. 2013; 2(1).
19. Menezes A, Vanstone S. Elliptic curve cryptosystem and their implementation. Journal of Cryptography. 1993; 6(4):209–24.
20. EI Gamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. Advanced in Cryptology. Proceedings of Crypto84. Springer Verlag. 1988; 10–8.
21. ELGamal T. A public cryptosystem and signature scheme based on discrete logarithms. IEEE Trans on Info Theory(S0018-9448). 1985; 31(4):469–72.
22. Rivest RL, Shamir A, Adleman LM. Method for Obtaining Digital Signatures and Public-key Cryptosystems. Communications of the ACM. 1978; 21:120–6.
23. Diffie W, Hellman ME. New directions in cryptography. IEEE Transactions on Information Theory. 1976; 22:644–54.
24. Nagaraj S, Raju GSVP. Image security using ECC approach. Indian Journal of Science and Technology. 2015 Oct; 8(26):1–5. Doi: 10.17485/ijst/2015/v8i26/81185.
25. Kurt M, Yerlikaya T. A New Modified Cryptosystem Based on Menezes Vanstone Elliptic Curve Cryptography Algorithm that Uses Characters' Hexadecimal Values. TAEECE, Turkey. 2013.
26. Kurt M, Duru N. Encryption with Changing Least Significant Bit on Menezes Vanstone Elliptic Curve Cryptosystem. 2014; 1–3.
27. Geetha G, Jain P. Implementation of Matrix based Mapping Method using Elliptic Curve Cryptography. International Journal of Computer Applications Technology and Research. 2014; 3(5):312–17.
28. Amounas F, El Kinani EH. Cryptography with elliptic curve using Tifinagh characters. Journal of Mathematics and System Science. 2012; 139–44.
29. Amounas F, El Kinani EH. An Efficient Elliptic Curve Cryptography protocol Based on Matrices. International Journal of Engineering Inventions. 2012; 1(9):49–54.
30. Pan W, Zheng F, Zhao Y. An Efficient Elliptic Curve Cry ptography Signature Server with GPU Acceleration. IEEE Transactions on Information Forensics and Security. 2017; 111–22.
31. Reddy AG, Das AK, Yoon E-J, Yoo K-Y. A Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography. IEEE Access. 2016; 4:4394–407.
32. Available from: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
33. Stallings W. Cryptography and Network Security, principles and practices, 4th Edition.
34. Computer Security Objects Register (CSOR). Available from: http://csrc.nist.gov/csor/
35. Yalcin T. Compact ECDSA engine for IoT applications. IET Electronics Letters. 2016; 52(16).
36. El hadjyoussefwajih, Mohsen M, Rached T. A Secure Elliptic Curve Digital Signature Scheme for Embedded Device. International Conference on Signals, Circuits and Systems. 2008. p. 1–6.

37. Knezevic M, Nikov V, Rombouts P. Low-Latency ECDSA Signature Verification-A Road Toward Safer Traffic. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2016; 24(11):3257–67.
38. Zhang Q, Li Z, Song C. The Improvement of digital signature algorithm based on elliptic curve cryptography. 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC). 2011. p. 1689–91.
39. Junru H. The improved elliptic curve digital signature algorithm. International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT). 2011. p. 257–9.
40. Lamba S, Sharma M. An Efficient Elliptic Curve Digital Si gnature Algorithm (ECDSA).International Conference on Machine Intelligence and Research Advancement. 2013. p. 179–83.