ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 9, September 2017

A Secured Public key Exchange Technique for Elliptic Curve Cryptography

[1] A.Vithya Vijayalakshmi, [2] A. Dalvin Vinoth Kumar, [3]Er. Karthigai Priya Govindrajan, [4] Dr. L. Arockiam
[1][2] Research Scholar, [3] Software consultant, [4] Associate Professor
[1][2][4] Department of Computer Science
[1][2][4]St. Joseph's College (Autonomous), Tiruchirapalli, Tamil Nadu, India
[3] United States.

***Abstract* - Security is the most important factor in Internet of Things (IoT). The public key cryptography systems like RSA, DSA are not suitable for IoT Devices because of the key size and energy consumption. ECC cryptography system is more suitable for IoT lightweight devices. ECC provides same level of security as like RSA with small key size and less energy computation. Key exchange is challengeable task due to Routing transmission attacks like DoS attack. The proposed technique is to enhance key exchange system in ECC based security protocols. The public key is encrypted before shared into network. The path selection algorithm is used to choose two different paths. One is for encrypted public key sharing and another is for unlock-key for encrypted public key sharing. The possibility of attacks during key sharing is investigated; the key size and energy consumption is compared. This technique is suitable for Tmote Sky and MICAz nodes.**

***Key Words*— ECC, IoT security, Encryption, Curve 25519, Public key, Tmote.**

I. INTRODUCTION

The Internet of Things visualizes a real world in which every object is connected to the Internet. Things used in our day-to-day life form a part of the virtual world, whereby they can be controlled and remotely accessed in a spontaneous way. The aspect of the Internet of Things (loT) is to make computing ubiquitous. Because, loT is being seen as the forthcoming vision for both the economy as well as the individuals. Sensors and loT enabled devices work with the help of their networking capabilities and they are able to communicate with each other as well as interact with people to be accessed, monitored and managed remotely as shown in figure 1. Data collected from these sensors and devices allows individuals and industries to make better decisions and take advantage of new products and services. But, every new device that connects to a network is possibly exposed to viruses, malware, and other attacks that could result in security issues. However, with all the benefits of IoT, there are also many risks and security issues involved which pose an immense technical and social challenge.

Security is one of the most fundamental issues that need attention. There are several challenges in the IoT which are still under research. The IoT challenges and open problems are elevated due to two main reasons. The first one is related to the enormous information that is gathered by RFID about large of things which are found at IoT system. The second is security and the privacy of information due to wireless transmission media [1]. The terminal security issues of the IoT are sensitive information leakage, tampering, copying, air interface information leakage, terminal virus and other issues. Sensor network security problems of the IoT are counterfeit attacks and malicious code attacks and security risks in information transmission. To secure the information in wireless transmission there are some safety measures such as certification, access control, data encryption etc. Data encryption is an important means of protecting data security. The role of encryption is to prevent information from being decrypted when it is captured by the attackers. Information Encryption can solve the problem of eavesdropping, but needs a flexible and robust key exchange and management programs which are easy to deploy and suitable for the IoT enabled devices.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 9, September 2017

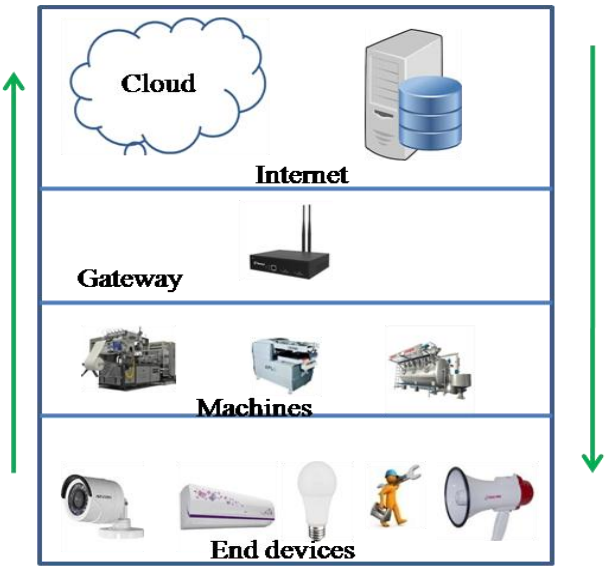

Figure 1. IoT Architecture

Data encryption technologies are used to protect the confidentiality and integrity of information. Cryptographic algorithms are mainly used for data encryption. They are divided in two parts i.e. Symmetric (private key or secret key) algorithms and Asymmetric (public key) algorithms [2]. Private-key cryptography holds both stream and block cipher. Block cipher includes AES, DES, RC2, 3DES, RC6 and other. Whereas, public-key cryptography includes DSA, RSA, ECC and someother algorithms. RSA (Rivest, Shamir and Adelman) algorithm is widely used as public key cryptographic algorithm for cryptosystems. This algorithm is based on factorization of integers [3]. It provides a strong security; therefore an adversary should not be able to break RSA by factoring due to its complexity and large keys [4].While the DSA (Digital Signature Algorithm is based on that of the discrete logarithm problem in integer fields [5]. DSA is standardized and wide utilized in IoT applications. The remaining paper is organized as follows; Section II explains ECC, Section III gives IoT related works. Section IV, explains about the proposed work, with figures and tables and finally Section V describes the conclusion of the work.

II. ELLIPTIC CURVE CRYPTOGRAPHY

ECC works on the concept of elliptic curve, an elliptic curve is a group of finite field and ECC uses this group for its working [6]. Elliptic Curve Cryptography uses the curve equation: $y2 = x3 + ax + b$, where a and b are constants, some widely used curves and their factors are tabulated in table 1. It is considered to be more suitable for building up lightweight public key crypto-systems (PKC) mainly due to its key size [7].

Table 1. Cofactor and Twist factors of the curves

Curve	Cofactor	Twist factor
Curve 25519 [8]	8	4
256 [9]	1	34905
E-382 [10]	4	4
M383	8	4
Curve 41417 [11]	8	8
M 511	8	4

Elliptic Curve Digital Signature Algorithm (ECDSA): The Elliptic Curve version of the Digital Signature Algorithm (DSA) is a variant on the ElGamal signature scheme. The algorithm allows the use of ECC to generate and verify signatures for messages. It provides greater security with smaller key sizes. It is mostly suitable for machines having low bandwidth, low computing power, less memory. It has easier hardware implementations.

III. RELATED WORKS

Hemant et al. [12] carried out the comparative analysis of security mechanisms such as DES, AES, RSA and ECC to enhance the security over internet. The outcome of these comparison showed that ECC secures the data being transmitted over the Internet with low computation, shorter keys, low cost and faster execution. Tarun Kumar Goyal et al. [13] focused on lightweight and low power algorithm such as Diffie-Hellamn, RSA and ECC and proposed an improvement in ECC methodology. The comparative analysis was done based on various parameters such as power, area, and performance. Experimental outcomes proved that the Elliptic Curve Diffie Hellman (EC-DH) is the best algorithm for Internet of Things in terms of power and area.
Sindhu et al. [14] discussed ECDH security algorithm for securing Internet of Things and presented a new scheme without modular inversion process in Signature Verification algorithms. The scheme is compared with the existing system and proved that the proposed digital signature scheme is more secured. Zhe Liu et al. [15]

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 9, September 2017

discussed the security system of Internet of Things and presented that the elliptic curve cryptography is the main component for securing IoT. Generation of MoTE curve is explained, which is an elliptic curve includes both Montgomery model and twisted - edwards model. They presented the design of a scalable, regular, and highly-optimized ECC library for both MICAz and Tmote Sky nodes, which supports both widely-used key exchange and signature schemes. The Flash memory and Random Accesses memory usage for different key size are tabulated in table 2.

Table 2. Flash memory and RAM usage for various key size

Bits	Affine	Homogeneous	Jocobin	Bignit.o	ECC.o
160 [16]	1504	1652	1904	4186	5373
192 [17]	1464	1592	1852	4186	5361
256 [18]	1400	1544	1796	4186	5403

Oriol Pinol et al. [19] described ECC as a strong algorithm to provide security in the IoT environments and presented a lightweight implementation and evaluation of ECC for the Contiki OS. They discussed the ECC mathematical foundations, operations and key generation etc. Experimental conducted on particular platform proved that jacobian coordinate gives better performance in less storage space. The implementation is released under BSD licensed and unified in much security protocol trust on ECC.

IV. PROPOSED WORK

IoT is the interconnectivity of things, objects, etc, to Internet. The users in IoT are connected to cloud directly or indirectly. The security is the major concern in IoT. The famous algorithms like RSA, DSA are unable to fit in the small devices. The sensor nodes have only 8 / 16 bit microcontroller at a frequency of 10 MHz with small KB of ROM. There is a need of lightweight security algorithm for these kinds of lightweight devices. While compared to RSA, ECC is more suitable for generating a lightweight security algorithm. The elliptic curve algorithms ECDH and ECDSA also have some difficulties in key exchange.
The proposed algorithm uses curve 25519. The proposed work deals with four types of keys namely Personal key (Pe), Public key (Pa), Public Key Decryption (dPa) and Operational key (O). The personal key (Pe) is an integer number generated by the user. The common string (S) is an integer value which is known to all. The public key (Pa) is obtained by personal key and string as the coordinates to the curve (E). The generated public key is further encrypted as encrypted public key (ePa). The key for decrypting the encrypted public key and encrypted public key are sent to the receiver in two different paths from sender. The receiver generates his operational key using the public key of the sender. The sender generates the operational key using the public key of the receiver. The operational key of both sender and receiver are identical. The encryption and decryption are carried out using operational key. The sender encrypts data using operational key of sender, the receiver decrypts the data using the receiver's operational key. The pictorial representation is shown in figure 2, and step by step process of proposed algorithm is as given below:

Step 1: Choose the secure Elliptical curve (Ep)

Step 2: Generate the personal Key (Pe)

Step 3: The public key is generated using scalar multiplication to all points (P).

Step 4: Encrypt the public key

Step 5: Identify two different paths from source to destination (Neighbour of p1) ≠ (Neighbour of p2)

Step 6: Share the Encrypted public key of the sender to receiver in Path (p1)

Step 7: Share the key for decryption the public key in path (p2)

Step 8: Receiver encrypts its public key using the key sent by sender and sends the encrypted public key alone.

Step 9: Both the sender and receiver generates the operational key (O).

Step 10: Sender encrypts the data using operational key and transfer the data

Step 11: Receiver decrypts data using its operational key

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 9, September 2017

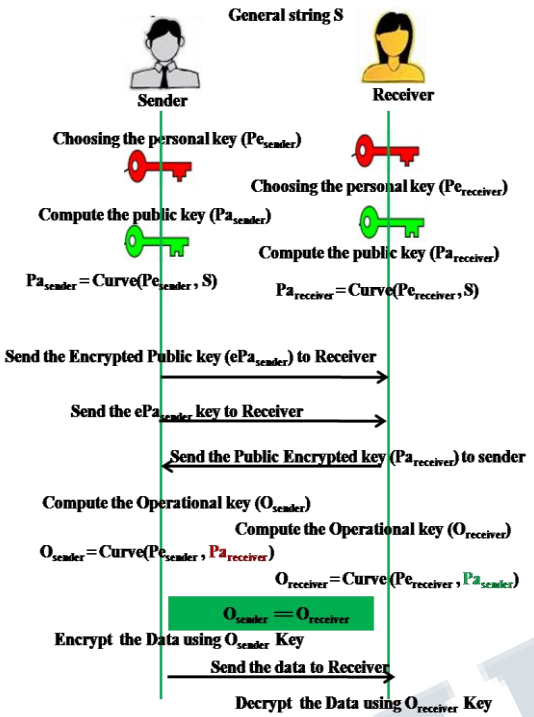

Figure 2. Key Exchange Mechanism

a. Curve Selection Algorithm:

In elliptic curve cryptography the curve selection is an important task. Montgomery and Trusted Edwards are the two widely used curves. Zheliu et al. [15] generates MoTE curves. MoTE curves are hybridization of Montgomery and twisted Edward curves. The MoTe curve uses Montgomery model and birationally equivalent twisted edwards model. The MoTE curves are generated as P159, P191 etc. whereas P stands for Pseudo-mersenne prime. The following digits are bit length. The curves are most commonly derived from wierstrass Equation. The Elliptic curve (E) is expressed as $Y^2 = X^3+AX+B$ where A, B, X, and Y are the elements of the Finite field (F). The elements in the finite field can be a real number (R), complex number (C), and rational number (Q). In ECC, the Curve (E) needs to be a member of finite group. The coordinates of curve (x y) where belongs to finite field (F). The projective coordinate system is used in proposed system. It has commonly two coordinates namely homogeneous and Jacobin. In homogeneous the coordinates (x y) is denoted as (θx, θy, θz) curve which is derived from $Y^2= x^3 + Axz^2 + z^3B$.In Jocobin the coordinates (x y) is denote as (α^2x, α^3y, αz) curve is derived from $Y^2= x^3 + Axz^4 + z^6B$. The point for generating the curve ($y^2 \equiv x^3+ x+1 \bmod 11$) is obtained as shown in table 3. The quadratic residue for the curve exists only when ($a^l \bmod p$) is 1. L value is calculated from (p-1) / 2. The point of the curve exists only when the curve has quadratic residue. These curves have the order of 4 so the value to be a multiple of 4 or 8.The equation$Y^2= x^3– x +1$ gives the curves as shown in figure 3.

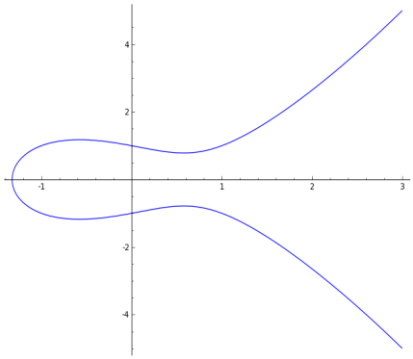

Figure 3. Elliptic curve for $Y^2= x^3– x +1$

Table 3. Points for curve with mod 11

x	a=(x3+ x+1 mod 11)	a power l mod 11	curve	points
1	3	1	y	(1,10)(1,1)
2	0	0	n	
3	9	1	y	(3,5)(3,6)
4	3	1	y	(4,2)(4,9)
5	10	10	n	
6	3	1	y	(6,4)(6,7)
7	10	10	n	
8	1	1	y	(8,3)(8,8)
9	10	10	n	
10	10	10	n	

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 9, September 2017

b. Key Generation:

The proposed work deals with four keys:

- o Personal Key (Pe)
- o Public Key (Pa)
- o Public Key Decryption (dPa)
- o Operational Key (O)

The personal key may any integer which returns the points (P, Q) in the curve. The personal key for both sender and receiver are different and it is not shared. The operational string is constant value which is known to all. The public key is generated individually by both sender and receiver using their personal key. The public key is shared between sender and receiver. In existing works public key of the sender and receiver are shared in the same path as shown in figure 4. The public key of sender Pasender is sent to receiver in path sender – 1- 2- receiver. The public key of the receiver Pareceiver also sent in the same path, there may a chance for hacker to snoop public key and generate the decryption key using common string (s) which is known to all. The proposed work uses a novel mechanism for public key exchange. The sender identifies two different paths (P1, P2) towards receiver. The nodes in Path (P2) need to be most possible least value of nodes in path (P1). Receiver node encrypts its public key using senders public key encryption key (ePa key). This novel path selection mechanism restricts attackers by snooping public key. Both the sender and receiver generate the operational key (O). The sender's operational key is generated using private key of sender (Pesender) and public key of the receiver (Pareceiver) as the coordinate value for cure and operational key is generated. This process is same to receiver as like sender. The operational key is used for encryption and decryption which is generated using the personal key and shared public key.

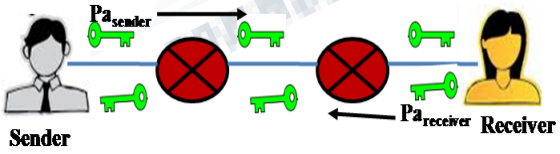

Figure 4. Key exchange in same path

a. Path Selection for Key Exchange

The path selection for key exchange is the new mechanism introduced in the proposed work. In IoT, the path is selected dynamically with the help of intermediate nodes. The public key is shared to the sender and receiver using the intermediate nodes. The intermediate node may be an attacker node. When the public node of both sender and receiver shared in the same path means the attacker node generate the operational code easily by snuffing the sender's public code and receiver's public code. The proposed work chooses the two different paths to send the encrypted public key and key for public key decryption. In figure 5 sender has two different paths Path (p1) and path (p2) towards the receiver.

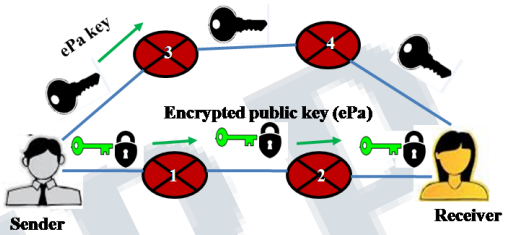

Figure 5. Key exchange in different paths

	DSA	RSA	DH	ECC	Proposed
Key Exchange	No	Yes	Yes	Yes	Yes
Encrypt/ Decrypt	No	Yes	Yes	Yes	Yes
Digital Signature	Yes	Yes	No	Yes	Yes
Key size	-	1024 bits	-	160 bits	160 bits
Routing attack vulnerable	yes	Yes	Yes	Yes	No

V. CONCLUSION

The proposed work is suitable for lightweight IoT devices like ATmega 128 microcontroller enabled devices and 8 bit AUK processor enabled devices. The proposed work provides additional security for the public key by encrypting the public key. The packet spoofing attack is resisted by the proposed mechanism. The usage of curve 25519 leads to high performance, reduced power consumption and less memory usage in IoT applications. The presence of pseudo-mersenne prime series makes the proposed system to work with high speed and memory efficient. ECC provides small key size. It is suitable for the industrial IoT, smart cities etc. where the data plays a major role. The proposed work will be implemented in 8 / 16 bit microcontroller to establish the real time IoT platform as part of future work.

REFERENCES

[1] Gonzalez, Enrique, Raul Peña, Cesar Vargas-Rosales, Alfonso Avila, and David Perez-Diaz de Cerio. "Survey of WBSNs for pre-hospital assistance: trends to maximize the network lifetime and video transmission techniques." Sensors, vol. 15, no. 5, 2015, pp. 11993-12021.

[2] Bellare, M., and Hoang, "V. T. Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model", In Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015, pp. 627-656.

[3] Burnett, Steve and Stephen Paine. "The RSA Security's Official Guide to Cryptography", McGraw-Hill, Inc., 2001, pp. 51-53.

[4] Singh, D., Nand, P., Astya, R., and Dixit, "Improved DSA Cryptographic Protocol and its Comparative Study with RSA Protocol", IEEE, International Conference on Computing, Communication & Automation (ICCCA), 2015, pp. 755-759.

[5] Al Imen Ali, "Comparison and evaluation of digital signature schemes employed in NDN network", International Journal of Embedded systems and Applications (IJESA), vol. 5, no. 2, 2015, pp. 15-29.

[6] Said, Omar, and Mehedi Masud. "Towards Internet of Things: Survey and Future Vision", International Journal of Computer Networks (IJCN), vol. 5, no. 1, 2013, pp. 1-17.

[7] Dewan, and Surbhi. "Comparative Study of Security Protocols to Enhance Security", Advanced Computing & Communication Technologies (ACCT), 2015, pp. 1-6.

[8] Bernstein, and Daniel J. "Curve25519: new Diffie-Hellman speed records", International Workshop on Public Key Cryptography, Springer Berlin Heidelberg, 2006, pp. 207-228.

[9] Gueron, Shay, and Vlad Krasnov. "Fast prime field elliptic-curve cryptography with 256-bit primes", Journal of Cryptographic Engineering, vol 5, no 2, 2015, pp. 141-151.

[10] Aranha, Diego F., Paulo SLM Barreto, CCF Pereira Geovandro, and Jefferson E. Ricardini. "A note on high-security general-purpose elliptic curves", IACR Cryptology ePrint Archive. 2013 pp. 647- 653.

[11] Guerrini, Eleonora, Laurent Imbert, and Théo Winterhalter, "Randomized Mixed-Radix Scalar Multiplication", https://eprint.iacr.org/2016/1022.pdf, Accessed on: 01.02.2017.

[12] Hemant, Kumar, and Archana Singh. "Internet of Things: A Comprehensive Analysis and Security Implementation through Elliptic Curve Cryptography", International Journal of Current Engineering and Technology, vol. 6, no. 2, 2016, pp. 498-502.

[13] Goyal, Tarun Kumar, and Vineet Sahula, "Lightweight security algorithm for low power IoT devices", Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1-7.

[14] Sindhu, B., and R. M. Noorullah. "Secure Elliptic Curve Digital Signature Algorithm for Internet of Things", Global Journal of Computer Science and Technology, vol. 16, no.3, 2016, pp. 1-5.

[15] Liu, Zhe, Xinyi Huang, Zhi Hu, Muhammad Khurram Khan, and Lu Zhou. "On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age", IEEE Transactions on Dependable and Secure Computing, 2016, pp. 1-5.

[16] Anoop, M. S. "Elliptic Curve Cryptography", An Implementation Guide, 2007, pp.51-55.

[17] Souza, E. C. C., and E. N. S. Muccillo. "Effect of Solvent on Physical Properties of samaria-doped ceria prepared by homogeneous precipitation", Journal of Alloys and Compounds, vol. 473, no.1, 2009, pp. 560-566.

[18] Qin, Ying, Chengxia Li, and ShouZhi Xu. "A fast ECC digital signature based on DSP", IEEE, International Conference on Computer Application and System Modeling (ICCASM), 2010, vol. 7. 2010, pp. 77-83.

[19] Oriol Pinol Piñol, Shahid Raza, Joakim Eriksson and Thiemo Voigt, "BSD-based Elliptic Curve Cryptography for the Open Internet of Things", IEEE, International Conference on New Technologies, Mobility and Security (NTMS), 2015, pp. 1-5.