International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

# Optimized ECDSA Algorithm for Secure and Efficient use in IoT Network

**Lalita Agrawal, Namita Tiwari**

***Abstract*: Internet of Thing (IoT) enhances the heterogeneous communication facility by providing Thing-to-Thing, Human-to-Thing, and Human-to-Human communication schemes. Various kind of threats and vulnerability cause sensor equipped IoT environment at a larger scale. So security and privacy are two important factors that must rely on IoT communication model. IoT network has some capability constraints that affect the deployment of realistic IoT application at a wider level like healthcare system and smart grid, smart city. One of the security concerns of the IoT network is the need of light-weighted encryption scheme. Elliptic Curve Cryptography (ECC) simulation with integrated encryption scheme is the most prominent solution for IoT devices to develop lightweight encryption technology. Here, Optimized Elliptic Curve Digital signature scheme is proposed to achieve secure communication between IoT sensor nodes. The results of optimized ECDSA algorithm are analyzed on Cooja simulator that is a IoT network simulator.***

***Keywords*: IoT, Security, ECC, Optimization, ECDSA, Simulation.***

## I. INTRODUCTION

Internet of Thing (IoT) network is comprised of various sensors (like PIR sensors, temperature sensors, RFID tag, RFID reader, etc), actuators, and transducers. IoT provides different pattern of communication schemes like Thing-to-Thing, Human-to-Thing, and Human-to-Human [1]. Smart city, public & defense services, smart surveillance system, smart grid, etc are the application of IoT. Protocols and standards of IoT network are provided by some standard institute or organization like ITU, IETF, and IEEE [2]. The main objective of these protocols is to meet the IoT light weight and low power constraint requirement. IoT network sensor nodes are capability constraint that is powered by some batteries that do not provide very long life to the nodes. So the nodes are required to have some sleeping routing for battery backup. IoT networks are opposed to conventional network in an aspect of communication mode since it provides Machine to Machine communication also [3]. So the security of IoT model causes by many attackers.

Trust management system is required for IoT network to assemble application specific data, reliability, accuracy, confidentiality and other security perspectives [4]. Sometimes, trust is one level above the security and privacy to ensure safety. Trust-worthy communication model also concerns about the other factor of an IoT entity.

## II. SECURITY MODEL OF IOT NETWORK

Security mechanism like encipherment (ensures data confidentiality, data integrity & authentication), digital signature (ensures non repudiation, data integrity & authentication) and key exchange are essential for the traditional network as well as IoT network to provide security services [5]. The CIA Triad of IoT network is presented as follow:

### A. Data Confidentiality

Communication over an unsecured channel causes data confidentiality. Any adversary can steal sensitive information over a public network and use that information for personal benefits. For military, public safety and defense purpose confidentiality becomes crucial. So data must only be transmitted from one authorized entity to other authorized entity to achieve data confidentiality. Encryption mechanism, where information is transmitted in the form of cipher text that is only accessible to the intended user only, can be used to achieve the data confidentiality services.

### B. Data Integrity

During the transmission of data, the attacker modifies the data or sometimes corrupts the data to cause data integrity which may result in system failure. The only authorized entity must have the right to update the data for security purpose using some valid and authenticate mechanism. Alteration of integrity is not only the result of adversary activity, sometimes it may be caused by system action like power failure, etc. Digital signature scheme and encipher technique provide the service of data integrity [6].

### C. Data Availability

The third security primitives of IoT security paradigm is data availability. This primitive is mainly caused by Denial-of-Service attack, where adversary sends lots of useless request to the server to make the server busy and unavailable to the other user. So the data availability is as much crucial as other security primitives. For an IoT security system, data availability must also be achieved.

Manuscript published on January 30, 2020.
* Correspondence Author
**Lalita Agrawal***, Faculty of Information & Technology, Parul University, Vadodara, India.
**Namita Tiwari**, Faculty of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal, India.

## III. RELATED WORK

Martinez et al. 2010 [7] presented a complete study of ECC algorithms, its mathematical simulation with existing algorithms. ECC simulation with integrated encryption scheme is proposed in detail like its hash function, standard, and MAC function. ECIES comparative survey with aspect of network model, standard and system specification is presented. But there is no motivation of using ECC in a sensor network and IoT. This has been seen in further research.

Yao et al. 2014 [8], Identity-based encryption (IBE) and attribute-based encryption (ABE) has been proposed as a lightweight standard for IoT environment. Features of ECC and ABE are combined to propose ECDDH with ABE based on no pairing technique. Two variant of ABE that is KP-ABE (key policy) and CP-ABE (cipher-text policy) are also presented. Result analysis is based on an overhead metric (that are communication and computation description) for both KP-ABE and CP-ABE for comparison purpose with the limitation of generalization and poor flexibility.

Elliptic curve cryptography has been used as a security platform for IoT applications to enhance the reliability and usability of the smart parking system [9]. The objective is to preserve the privacy using data perception trust and also make it suitable for light-weighted encryption. For a smart healthcare system, authentication services are provided using ECC for IoT environment [10].As the patient's medical report is very sensitive information so it must only be available to a person after proper authentication using public key infrastructure. So RFID based authentication architecture is proposed for the healthcare environment. Many of the research are still on-going to fulfill the requirement of light-weighted encryption for IoT environment.

## IV. PRELIMINARIES

Elliptic curve cryptography provides same level of security with a smaller key size unlike other public-key cryptosystem depicted in Table I.

**Table I: RSA and ECC Key size comparison over same level of security**

| Level of Security | Key size (in bits) |
|---|---|
| 80 | RSA-1024 |
| | ECC-160 |
| 112 | RSA-2048 |
| | ECC-224 |
| 128 | RSA-3096 |
| | ECC-256 |

Original cryptosystem has a key size of 1024 bits which is much larger than 160 bits key size of ECC for 80-bit security level [11]. But the computation cost and time of ECC operation (Point addition and scalar multiplication) are higher than asymmetric-key cryptosystem. There are some ECC optimization techniques that can be used to reduce the computation time and cost of ECC operation. Some of the ECC optimizations are discussed as follows:

### A. Sliding Window Optimization for scalar multiplication [12-14]

Since point multiplication operation is the most commonly used operation of elliptic curve cryptography. So this optimization technique is used to reduce the computation time of point multiplication operation. Here window width is shown as **w**. For ***kP*** scalar multiplication, **k** is a positive integer. First of all non-adjacent form of **k** is computed as:

$$\mathbf{NAF}(k) = \sum_{i=0}^{l-1} k_i * \mathbf{pow}(2, \mathbf{i})$$

This optimization requires some pre-computation values of the form $P_i = iP$ for **i= 1, 3----$2^w$-1.** While scanning NAF (*K*) from left to right, one point doubling operation is performed when the scanned bit is '1'and a point addition operation is also performed at that time. At a time w bits are scanned, and only one point doubling operation is performed for w bits set. So this decreases the computation time of scalar multiplication. But for pre-computation value, extra memory space is required.

### B. Shamir Trick Optimization [12-14]

Shamir trick is used to reduce the verification time of the ECDSA signature algorithm. The verifying step of ECDSA algorithm has the equation of the form **T(x, y) = A*P + B*Q** where P and Q are two points on elliptic curve and A & B are t-bit numbers. Firstly we have to pre-compute **iP + jQ** for all **i, j = 0, 1, ------$2^w$-1**. Initially, **R** is initialized to infinity as an intermediate value. For each bit scan performs two steps that is:

$$R = 2^wR$$

$$R = R + (K^iP + L^iQ)$$

So there is only one point multiplication operation is required instead of two point multiplication operation that reduces the computation. More RAM and ROM are utilized by Shamir trick for pre-computation values.

## V. PROPOSED SYSTEM

An Optimized Elliptic Curve Digital Signature Scheme is proposed to achieve the secure and light-weighted requirement of capability constraint IoT network. The proposed system involves three steps that are key generation, signing, and verification. Fig.1 presents the working of each step of Optimized ECDSA Algorithm.

### A. Key Generation:

1. Select an elliptic curve $\mathbf{E_p(a, b)}$ with **p** prime number and **q** is another prime number.
2. Select an integer **d** and $\mathbf{e_1(x_1, y_1)}$, a point on the elliptic curve.
3. Calculate $\mathbf{e_2(x_2, y_2) = d*e_1(x_1, y_1)}$ using sliding window optimization
4. Declare $\mathbf{(a, b, p, q, e_1, e_2)}$ as public key and **d** as private key.

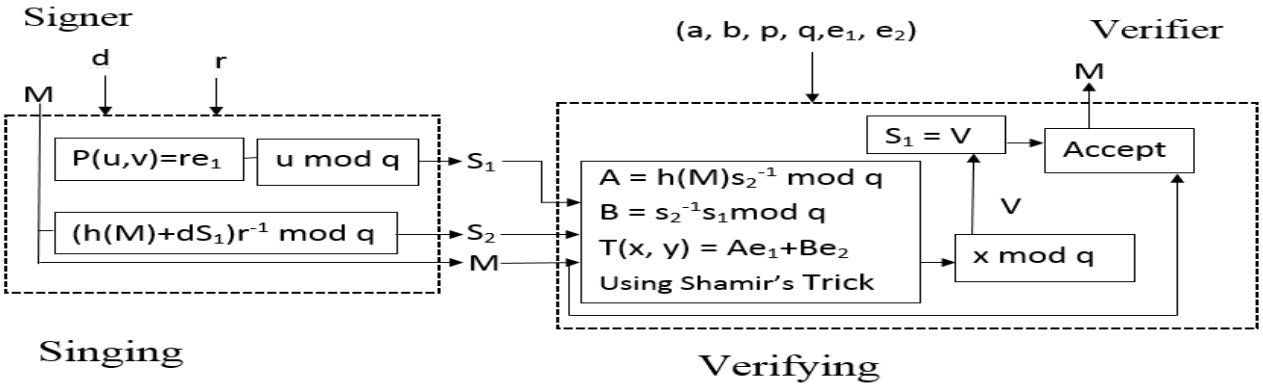



**Fig. 1 Block Diagram of Optimized ECDSA Algorithm**

**B. Signing:**

1. First of all, select **r**, a random number secret in between **{1, 2, 3, - - -, q-1}**.
2. Calculate **P(u, v) = r*e$_1$** using function ***f1***. **P** is also a point on the elliptic curve and sliding window optimization concept is also used.
3. Here, an extractor is used to extract the x-coordinate of point **P(u, v)**. Final signature **S$_1$** after modular arithmetic is like as **S$_1$= u mod q**.
4. To generate the second signature **S$_2$**, firstly create message digest **h(M)** and calculate **S$_2$** using **r, d, s$_1$, h(M)** like as: **S$_2$ = (h(M) + d*s$_1$)r$^{-1}$ mod q**.
5. Signer sends signature **S1, S2,** & message **M** to the receiver.

**C. Verifying:**

1. **A** and **B**, the intermediate result is calculated using **M, S$_1$, S$_2$** as:
   **A = h(M)S$_2^{-1}$ mod q & B = S$_2^{-1}$S$_1$ mod q**
   Both **A & B** are used to reconstruct the point **T** as:
   **T(x, y) = A*e$_1$ + B*e$_2$.**
   Here Shamir's trick is used for verification to reduce the computation cost and time of **T(x, y) = A*e$_1$ + B*e$_2$.**
2. Extract x-coordinate of point **T** and if **x= S$_1$ mod q**, the message is accepted otherwise it is not an authenticated message.

Hence this proposed algorithm reduces the computation of scalar multiplication and verification time of the ECDSA algorithm.

## VI. SIMULATION TOOL

IoT systems are a fully hardware-based system that uses specific kind of sensors. To build an IoT hardware architecture based on a proposed system or/and protocol approaches require preliminary testing to analyze the behavior of the proposed algorithm.

### A. Contiki OS

Firstly Contiki OS is designed and released in 2004 and written in a C programming language. The purpose of designing Contiki OS is to support capability constraint application of IoT system. Currently, it is designed to support many microcontrollers like AVR, MSP430 and many more [15]. We can easily port our application and algorithm to Contiki by dynamic uploading for result analysis. Its framework architecture is Contiki-Sec that is designed for light-weighted encryption.

### B. Cooja Simulator

Cooja is a network simulation tool that is provided with Contiki OS. Cooja is C/Java-based simulator that was last updated in 2017. It is highly recommended for generalization and practical implementation of low power sensors. Cooja is simulator as well as an emulator. Cooja simulator provides the facility of selecting mote types based on the requirement [15, 16].

The proposed optimized ECDSA algorithm is simulated on Cooja IoT network simulator supported by Contiki OS. For the simulation, Relic is used as cryptographic library. Tmote Sky & Cooja mote based simulation is used for implementation. All the simulation configuration details used for result analysis are shown in below Table II.

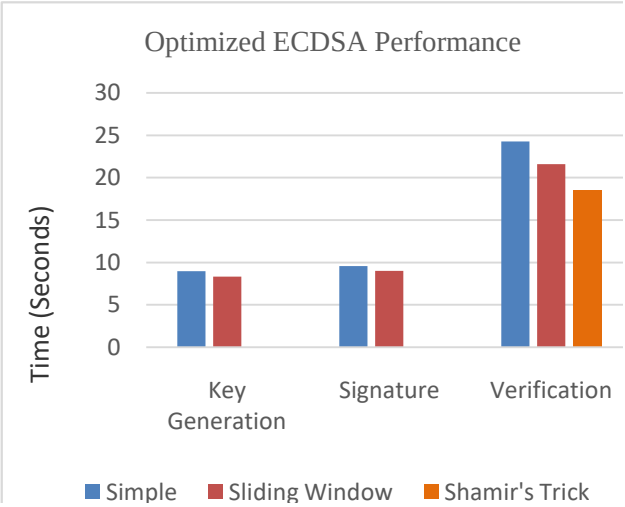
**Table II: Simulation Configuration detail**

| Parameter | Value |
|---|---|
| Simulator | Cooja |
| Cryptographic Library | Relic |
| Mathematical Library | Relic-Easy & GnuMP |
| Mote Type | Tmote Sky & Cooja |
| Network | 6LoWPAN |
| MAC Layer | IEEE802.15.4 |
| Host System | i5@2.53GHz, 4GB Memory |

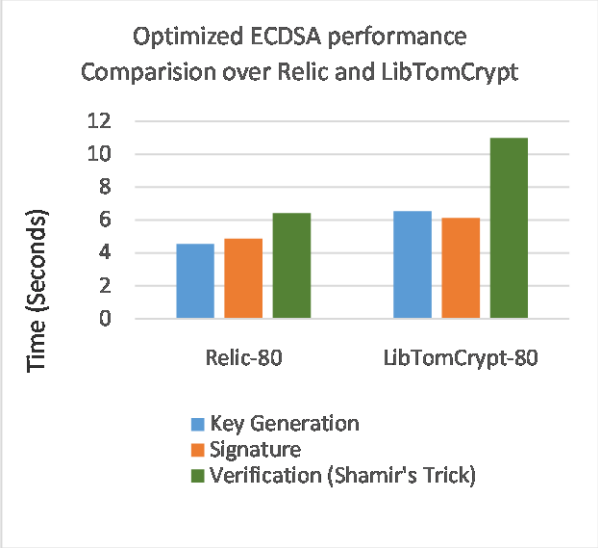
## VII. PERFORMANCE ANALYSIS

Performance of proposed optimized ECDSA algorithm is basically depend on two evaluation parameter that are execution time and

memory requirement. Contiki supports the real-time timer available for Cooja and it is hardware dependent. Since we are using Sliding window optimization technique and Shamir's trick for the optimization of existing ECDSA algorithm. Both the techniques require some pre-computed values that require extra memory space.




**Graph 1. Represent performance of Optimized ECDSA Algorithm in Key Generation, Signature, Verification step.**




**Graph 2. Represent the performance comparison of Optimized ECDSA Algorithm over Relic and LibTomCrypt libraries.**

**Table III: Execution Time (seconds) of proposed algorithm at each step with and without Optimization technique**

| Time(seconds) | Simple | Sliding Window | Shamir's Trick |
|---|---|---|---|
| **Key Generation** | 8.986 | 8.324 | - |
| **Signature** | 9.562 | 9.026 | - |
| **Verification** | 24.256 | 21.587 | 18.528 |

Optimized ECDSA algorithm has key generation, signature, and verification step so execution time is calculated at each step without optimization and with optimization (Sliding Window and Shamir's Trick) as shown in Table III and we have already discussed that Shamir's Trick is only applicable for verification time. Cooja Simulator supports many mathematical and cryptographic libraries. We have done result analysis only for two libraries (Relic and LibTomCrypt) according to the requirement of problem statement as shown in Table IV. Here we have used security level of 80 bits for the calculation of execution time over the libraries.

**Table IV: Execution Time (seconds) of proposed algorithm over Relic-80 and LibTomCrypt-80 libraries.**

| Time(seconds) | Key Generation | Signature | Verification (Shamir's Trick) |
|---|---|---|---|
| **Relic-80** | 4.562 | 4.864 | 6.425 |
| **LibTomCrypt-80** | 6.547 | 6.128 | 10.987 |

The estimation of memory is based on RAM and ROM utilization. Only some of the library has the facility of calculate the memory consumption. But still no inbuilt function is available for this purpose. Here all the result and comparison are based on execution time.

## VIII. CONCLUSION

It is observed that there is dire need of light-weighted security technique for IoT environment. Many of the researchers have discussed their work regarding this objective. By observing their work, we came across to use elliptic curve cryptography as a light-weighted and compatible technique for IoT network. In the previous research related to the use of ECC in IoT, we have found some limitation of computational overhead (due to scalar multiplication and verification). So here we have proposed an optimized ECDSA algorithm to overcome these limitations and we have analyzed better result from previous work by simulating our optimized ECDSA algorithm on Cooja simulator (an IoT network simulator).

## IX. FUTURE SCOPE

Optimized ECDSA algorithm is proposed as a light-weighted technique in this project with the result of lesser computation time than the previously existing work. But still, there is a requirement of further improvement of the proposed algorithm because while decreasing the computational overhead it is observed that the extra memory (in form of RAM or/and ROM) is required for pre-computed values. Here, we are using Cooja simulator that has the constraint on the number of nodes creation.So, the next target is to overcome these limitations and found out the solution that focuses on lesser memory utilization along with lesser computational overhead. We will also try to apply our algorithm on different-different IoT network simulator that has no constraint on the number of nodes.

## REFERENCES

1. Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi, "Internet of things security: a survey ", Journal of Network and Computer Application, vol. 88, pp. 10-28, (2017).
2. Jorge Granjal, Edmundo Monteiro, and Jorge Sa Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues", IEEE Communication Surveys & Tutorials., vol. 17, no. 3, pp. 1294 -1312, (2015).
3. Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, and Hicham Lakhlef, "Internet of things security: A top-down survey", Computer Network, vol. 141, pp. 199-221, (2018).
4. Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos, "A survey on trust management for Internet of Things", Journal of Network and Computer Application, vol. 42, pp. 120–134, (2014).

5. M. U. Farooq, Muhammad Waseem, Anjum Khairi, and Sadia Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)", International Journal of Computer Application (IJCA), vol. 111, no. 7, (2015).
6. Ashvini Balte, Asmita Kashid, and Balaji Patil, "Security Issues in Internet of Things (IoT): A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, Issue 4, ISSN: 2277 128X, pp. 450-455, (2015).
7. V. Gayoso Martínez, L. Hernandez Encinas, and C. Sanchez Ávila, "A survey of the elliptic curve integrated encryption scheme", Journal of Computer Science & Engineering, vol. 2, Issue 2, (2010).
8. Xuanxia Yao, Zhi Chena, and Ye Tian, "A lightweight attribute - based encryption scheme for the Internet of Things", Future Generation Computer System, vol. 49, pp. 104–112, (2015).
9. Ioannis Chatzigiannakis, Andrea Vitaletti, and ApostolosPyrgelis, "A privacy-preserving smart parking system using an IoT elliptic curve based security platform ", Computer Communications, vol. 89-90, pp. 165-177, (2016).
10. Debiao He, and Sherali Zeadally, " An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography ", IEEE Internet of Things Journal, vol. 2, no. 1, pp. 72-83, (2015).
11. Behrouz A. Forouzan,: Cryptography and Network Security. Tata McGraw-Hill., (2007).
12. Henk C.A. van Tilborg, "Encyclopedia of Cryptography and Security", Springer.
13. Kerry A. McKay, Larry Bassham, Meltem Sönmez Turan, and Nicky Mouha, "Report on Lightweight Cryptography", National Institute of Standards and Technology Internal Report 8114, (2017).
14. Nan Li , Dongxi Liu, and Surya Nepal, " Lightweight Mutual Authentication for IoT and Its Applications ", IEEE Transactions on Sustainable Computing, vol. 2, no. 4, (2017).
15. M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds", IEEE Internet of Things Journal, vol. 5, pp. 1637-1647, (2017).
16. Zhe Liu, Johann Grobschadl, Zhi Hu, Kimmo Jarvinen, Husen Wang, and Ingrid Verbauwhede, " Elliptic Curve Cryptography with Efficiently Computable Endomorphisms and Its Hardware Implementations for the Internet of Things ", IEEE Transactions on Computers, vol. 66, no. 5, (2017).

## AUTHORS PROFILE

**Lalita Agrawal,** currently working as an assistant professor, Department of Information & Technology, Parul University, Vadodara, Gujrat, India. Qualification: M.Tech (Advanced Computing ), MANIT, Bhopal.

**Nanita Tiwari,** currently working as an assistant professor, Department of Computer Science & Engineering, MANIT, Bhopal, India.