Comput Syst Sci & Eng (2020) 1: 51–63

International Journal of
Computer Systems
Science & Engineering

Multi level Key Exchange and Encryption Protocol for Internet of Things(IoT)

Poomagal C T[1]*, Sathish kumar G A[2], Deval Mehta[3]

[1] *Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Kanchipuram, INDIA*
[2] *Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Kanchipuram, INDIA*
[3] *SAC, ISRO, Ahmedabad, INDIA*

The burgeoning network communications for multiple applications such as commercial, IoT, consumer devices, space, military, and telecommunications are facing many security and privacy challenges. Over the past decade, the Internet of Things (IoT) has been a focus of study. Security and privacy are the most important problems for IoT applications and are still facing huge difficulties. To promote this high-security IoT domain and prevent security attacks from unauthorized users, keys are frequently exchanged through a public key exchange algorithm. This paper introduces a novel algorithm based on Elliptic Curve Cryptography(ECC) for multi-level Public Key Exchange and Encryption Mechanism. It also presents a random number generation technique for secret key generation and a new authentication methodology to enhance the security level. Finally, in terms of security, communication and computational overhead, the performance analysis of the proposed work is compared with the existing protocols.

Keywords: Elliptic Curve Cryptography, Key Exchange Mechanism, IoT, Elliptic Curve Discrete Logarithm Problem, Zero Knowledge Protocol, ElGamal Protocol.

1. INTRODUCTION

The Internet brings users closer and closer to the day-to-day web services. There are many reported instances of attacks on networks so-called Hackers attacks on cyber networks. With regard to security, the IoT will face more serious challenges because of the following reasons: 1) Networks viz mobile networks, sensor networks and traditional networks are currently extends to IoT, 2)'Internet' will be connecting every 'Thing' and 3)'Each other' will communicate through these 'Things', so the new privacy and security problems may emerge. There is a need for profound attention to the confidentiality, authenticity and integrity of information in the IOT and so the development of more complex cryptographic systems based on complicated mathematics is essential. The aim of this research is to ensure the secure transmission of information between the nodes of the IoT networks. Meanwhile in the IoT, due to the use of small devices and the ongoing connections with the Internet, the processing power requires to be much less and therefore this leads to the need for security mechanisms with less power consumption. Practically, Elliptic Curve Crypto systems has picked up expanding acknowledgment due to their appreciably smaller bit size of the operands compared to other public-key crypto systems. The computational complexity of RSA or the simple discrete logarithm system is lower than ECC, so ECC can be obviously selected for the high-performance public key applications. Despite the abundance of research on high-speed software and FPGA implementation of ECC from the mid-1990s, providing the high-performance ECC on promptly accessible (i.e., non-ASIC) platforms remains an open challenge. Due to standardization in Europe and the US, elliptic curves over prime fields are often selected over binary fields. ECC also provides the highest level of security and a way to achieve the unbreakable algorithm. Thus, ECC is appropriate for devices that are resource-conscious and are usually used in IoT. Security of the ECC is about solving the Elliptic Curve Discrete

*ctpoomagall@gmail.com

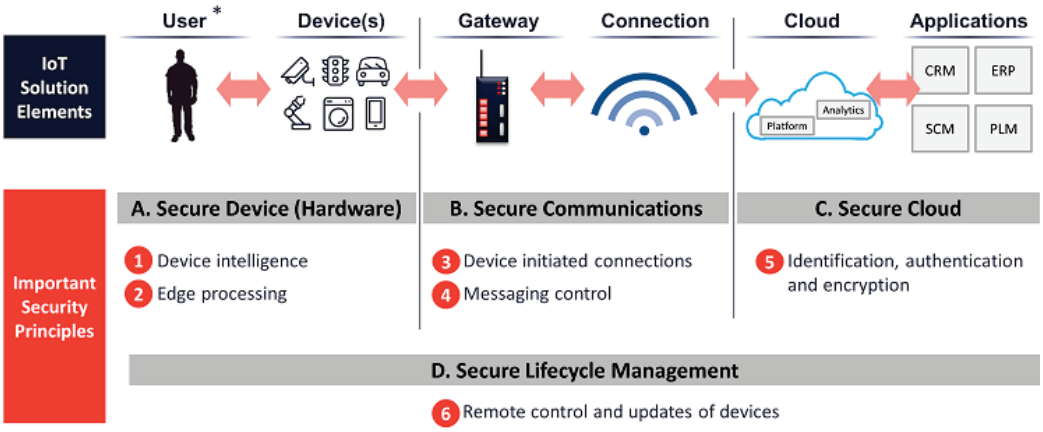

Figure 1 Multiple layers of the architecture and diverse aspects of information security in IoT.

Logarithm (ECDLP) problem. Although it has comparatively high theoretical complexity, it has some benefits over the other methods. Among these benefits, its implementation with lower keys is often significantly more effective [1]. With scalar multiplication, the Diffie-Hellman Key Exchange algorithm can be directly extended to the discrete logarithm over the elliptic curve. Simple scalar multiplication and point addition in Elliptic Curve Key Exchange(ECKE) protocols can provide a great deal of security over the public channels.

This paper proposes a new multilevel key exchange and encryption protocol using Elliptic Curve Cryptography with some more flavors of exponentiation operation, which will decrease the potential attacks that are still applicable in ECC algorithms. The Elliptic curve standards are currently being drafted throughout the world by various standardization organizations. Among these, NIST(National Institute of Standards and Technology) provides various standard curves to work with the Elliptic curves in case of improved security.

This paper organizes as follows: Section II reviews the key exchange mechanism and its associated vulnerabilities. Section III covers the basics and requirements of IoT; Section IV explains the proposed protocol for multi-level key exchange and encryption. Section V depicts the results and analysis of the proposed protocol. Lastly, Section VI outlines the concluding remarks.

2. RELATED WORKS

A comprehensive literature survey is made on various manuscripts of Authenticated Key Exchange Algorithm (AKE) and their potential vulnerabilities. A large number of AKE protocols have been discussed since Diffie and Hellman's seminal work in 1976 [1]. For instance in wireless mobile communications, Abdalla et al provided a three-party password authenticated key exchange (3PAKE) protocol[2]. Their scheme utilizes a trusted three-party server to authenticate the users prior to the exchange of session keys. Lu et al introduced a S-3PAKE protocol[3], but Chung et al[4] subsequently showed that they were in danger of being impersonation attack. Guo et al[5] also showed that the enhanced Chung et al protocol[4] has no guarantee for the protection from both impersonation and replay attacks. Chang et al[6] and Yoon et al[7] introduced a protocol not requiring a symmetric cryptosystem. The capacity to exchange only one session key per round limits a wireless mobile network implementation of the AKE protocol. Hence, there is a need of Multi-level key exchange protocols which, allows to exchange multiple keys at different levels in each round and so this paper presents a multi-level key exchange and encryption algorithm with a new authentication and a random key generation methodology for comparing its efficiency with [21], [22], [23], and [24], which has the similar optimization properties of the proposed algorithm.

3. BACKGROUND

3.1 Internet of Things(IoT)

In the emerging and active research field of IoT, with multiple layers of the architecture and from diverse aspects of information security as shown in Fig. 1, it is necessary to resolve various challenges. Below is a summary of the regular IoT security challenges and requirements.

3.1.1 Security Structure

Secure IoT solutions involve the development of end-to-end multiple layers through User, Device, Gateway, Connections, Cloud, Applications, and Lifecycle Management, which integrate together across IoT security and architecture. Therefore, building a security structure with the combination of these controls and information is a challenge and an important research area.

3.1.2 Key Management

Key management is a hot research area as it is the key basis for many security mechanism and is still the most demanding feature of cryptographic security.

3.1.3 Security Law and Regulations

Currently, security laws and guidelines are not yet the focus of principle, and there are no innovation standards for IoT. National security data, business insider facts and safety for the individuals mainly seeks the applications of IoT. The standardization

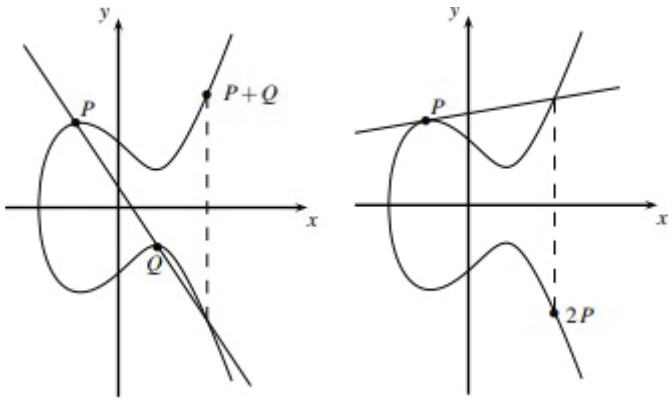

Fig. 2.(a) ***Fig. 2.***(b)

Figure 2 (a) is point addition operation and Fig 2(b) is point doubling operation on an elliptic curve over the real numbers.

organizations thus need the administrative perspective in order to promote the improvement of the IoT. Hence, the arrangements and guidelines are urgently required for the security laws of IoT.

3.1.4 Requirements for Flourishing Applications

With the advancements inIoTs, WSNs, Industrial internet, pervasive computing technology, smart cities and grids, and distributed real-time control theory is becoming a reality. A high-security system is necessary for this development with system performance. However, the security issues for the IoT are much severe and so the establishment of an efficient security system is necessary. In addition, key management in the real large-scale sensor network is always a challenging job with respect to IoT-related policies and regulations.

Over the most recent years, this rising space for the IoT has been attracting the huge intrigue, and will proceed for the upcoming years. Regardless of rapid advancements in IoT, it faces various security challenges and extreme difficulties to achieve the requirements of the security. Overall, the development of the IoT will bring more serious security problems, which are always the focus and the primary task of the research. And so, a strong key management and encryption mechanism has to be developed with the basis of Elliptic curve discrete logarithm problem, which works in a prime field elliptic curve.

3.2 Elliptic Curve Cryptography

Elliptic Curve Cryptography is a relatively new crypto system, suggested independently, from the second half of 19th century, by Neal Koblitz [9] and Victor Miller [10]. ECC has now been industrially recognized and adopted by a number of standardizing bodies such as ANSI, IEEE [11], ISO and NIST [12]. An elliptic curve can be defined as:

$$y^2 = x^3 + ax^2 + b \tag{1}$$

where ‘a’ and ‘b’ are the constants with

$$4a^3 + 27b^2 \neq 0 \tag{2}$$

3.3 Elliptic Curve Operations over Finite Field

For a finite field represented as Z_P, with $p > 3$, having set of all pairs $(x, y) \in Z_P$ forms an elliptic curve that fulfils $y^2 = x^3 + ax + b$ and an imaginary point of infinity [13]. If $(G, +)$ is a group and $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are the two points in the group. Then, the coordinates of an additional value of these points is given by another point in the same curve which is derived by the elliptic curve operations as in [13]. It defines the two arithmetic operations (i) Point Addition: $P + Q$, to compute R = P +Q, when P is not equal to Q. A line through P and Q will obtain a third point of intersection between the elliptic curve and the line. The Mirrored point of this third intersection point along the x-axis, is the point R and it is shown in. Fig.2(a) and (ii)Point Doubling: P + P is to compute P + Q when P = Q. Hence, R = P + P = 2P. A tangent line through P will obtain a second point of intersection between this line and the elliptic curve. And the mirrored point of the second intersection along the x-axis is the result R of the doubling, which is shown in Fig. 2(b).

3.4 Diffie-Hellman Protocol

This protocol is used to exchange keys [14]. To use this protocol in elliptic curves, consider that there are two parties which are Alice and Bob [13] and both the parties having an individual key pairs. To establish a shared secret key over a public insecure channel, it is a variant of the Diffie–Hellman protocol using elliptic-curve cryptography as reported in [14].

3.5 ElGamal Encryption over Elliptic Curves

The ElGamal encryption using elliptic curves is purely dependent on the discrete logarithm problem[15] used in asymmetric cryptosystems. This encryption reflects the plain text as the elliptic curve points and the process is as in [15].

3.6 Zero Knowledge Protocol

Informally, Zero Knowledge Protocol (ZKPs) enables one of the users to prove their knowledge of a secret to another party

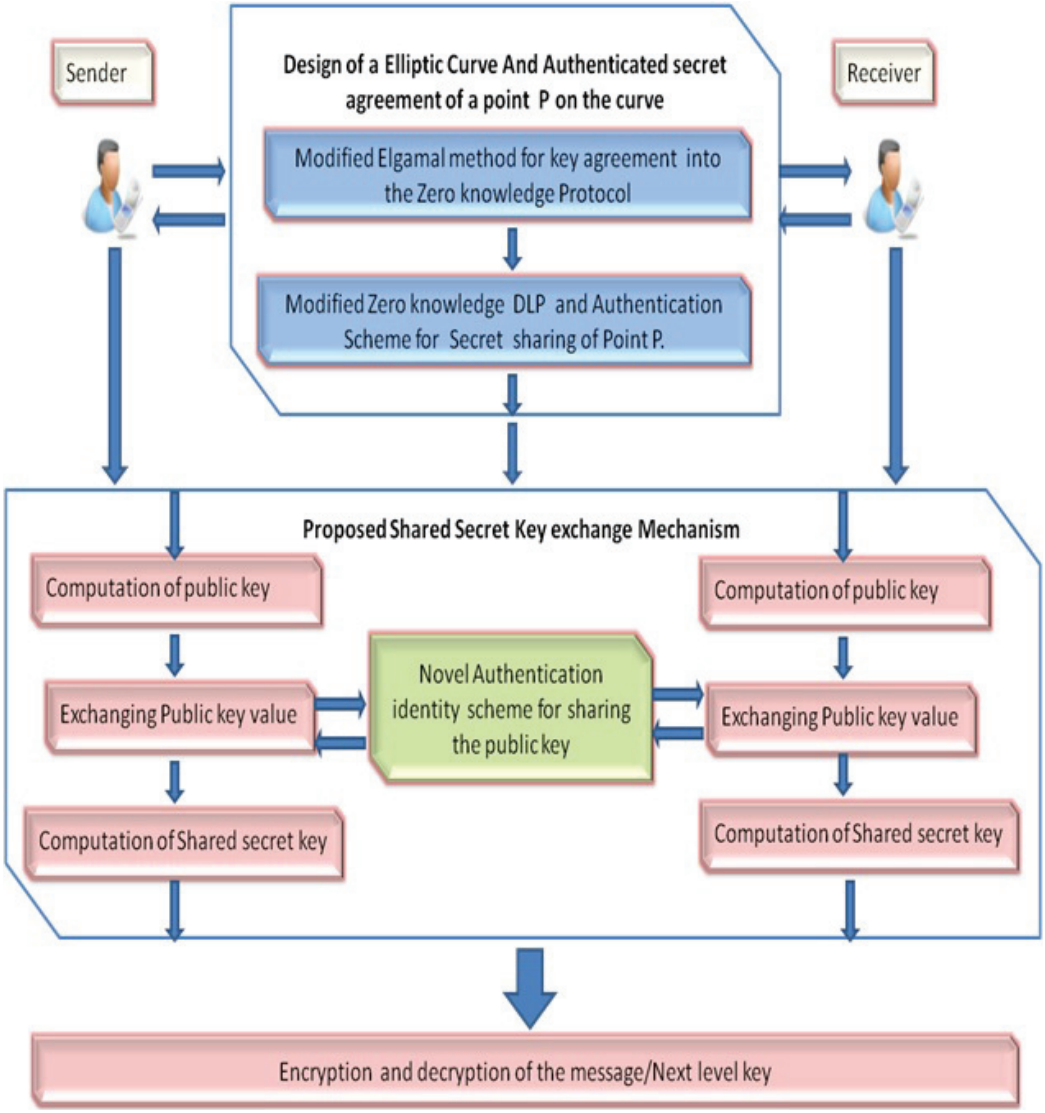

Figure 3 Flow of the Proposed Work.

without ever revealing the secret itself. Specifically, it has three characteristics [16].

1. Completeness: If the given condition is satisfied, the honest receiver will be contented by the honest sender.
2. Soundness: If the given condition is dissatisfied, sender cannot convince the receiver that it is authenticated, with some probable exceptions.
3. Zero-knowledge: If the given condition is satisfied, then the intruder can come to know that no other information can be collected from this obtained data.

Randomness is likewise a vital property of Zero knowledge protocol.

3.7 Authentication Scheme

Always, validating the data strives to maintain the user incognito from the eavesdropper. However, in economic and e-commerce transactions there are many authentication schemes that involve not only anonymity to the intruder, but also the server authentication.

4. PROPOSED WORK

The proposed work is a novel key exchange and encryption algorithm that contributes newly developed sub-modules and functions to build the complete security mechanism at each computation stage as shown in Figure 3.

Module I: Modified ElGamal Method

Based on the Diffie–Hellman key exchange method, the modified ElGamal encryption system provides an additional layer of security by asymmetrically encrypting the keys.

The proposed modified ElGamal method is used for the initial agreement of the secret values in the proposed key exchange protocols as shown in Fig. 4. An elliptic curve $E(a, b)$ over a field $G(p)$ of order 'q' is defined and a point on the curve

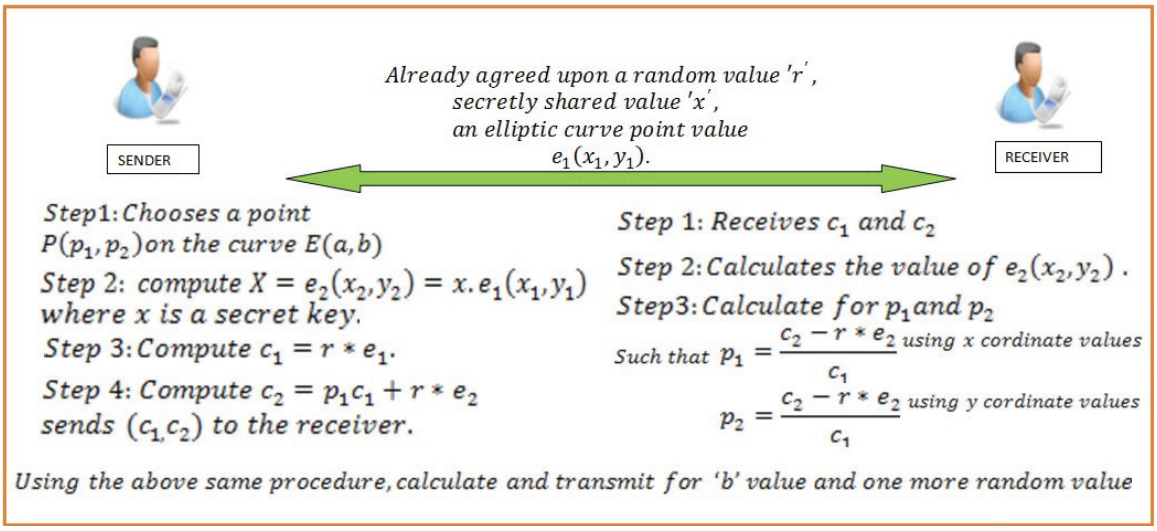

Figure 4 Modified ElGamal Method.

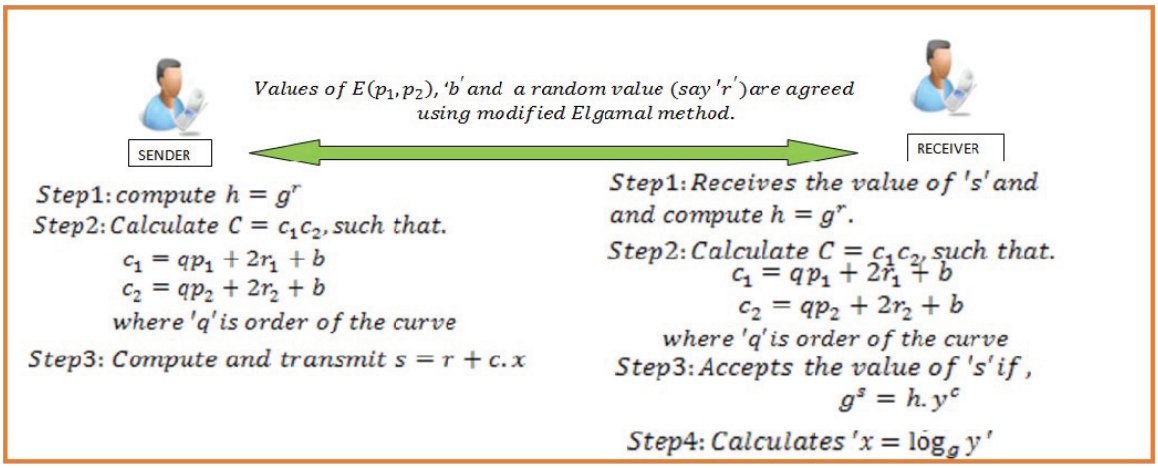

Figure 5 Modified Zero Knowledge Protocol.

$e_1(x_1, y_1)$ is chosen as a generator point. In this paper, the process involved in the modified ElGamal method is given as follows:

Sender Side:

Step1: Let $Z = e_1(x_1, y_1)$ and a random small number 'r' has to be chosen, where $r \in Z_q$ are publicly known.
Step 2: Sender has to compute $X = e_2(x_2, y_2)$ such that $X = e_2(x_2, y_2) = x, Z = x, e_1(x_1, y_1)$ where 'x' is a secret key selected by the sender.
Step 3: A point $P(p_1, p_2)$ has to be selected on the curve $E(a, b)$ which needs to be secretly shared to the receiver.
Step 4: Finally, the key pair is generated by the computation of c_1 and c_2 as $c_1 = r * e_1$ and $c_2 = p_1 c_1 + r * e_2$. For the computation and transmission of every value viz the point $P = (p_1, p_2)$, a random value and the value 'b', one of the coordinate value of the elliptic curve point can be used at a time. Then, the key pair (c_1, c_2) has to be sent through the public channel to the receiver.

At Receiver Side:

Step 1: The key pairs c_1, c_2 is received for two times, first time is encrypted with the point values $P(p_1, p_2)$ and the next time is encrypted with a random value and the value 'b'.
Step 2: Receiver needs to calculate the value of $e_2(x_2, y_2)$ as $X = e_2(x_2, y_2) = x, Z = x, e_1(x_1, y_1)$.
Step 3: Later, separately the value of p_1, p_2 a random value and the value 'b' has to be retrieved from the obtained key pairs (c_1, c_2), to form the point $P(p_1, p_2)$ on $E(a, b)$ by using the encrypted value e_2 and the secret key 'x' and 'r', using the equation (3)

$$P = \frac{c_2 - r * e_2}{c_1} \tag{3}$$

4.1 Module II: Modified Zero Knowledge Protocol

The Modified Zero knowledge protocol proposed in this paper is based on homomorphic encryption and the schnor's protocol proves the knowledge of a discrete logarithm.

By using this protocol, a secret value can be shared with a proper authentication in a public channel. Modified Zero Knowledge Protocol is shown in Fig 5.

Over a cyclic group G_q of order q with generator g, the proposed Modified Zero Knowledge protocol of DLP module can prove and find the secret value as $x = \log_g y$.

1. By using the ***modified ElGamal*** method proposed in ***module I***, the prover and verifier must secretly agreed upon a secret value 'd' , and a point $P = (p_1, p_2)$ on an Elliptic curve.
2. The prover has to generate and send the value of 'h' using a random number 'r' as '$h = g^r$'.
3. The verifier and prover chooses a value 'C' such that '$C = c_1c_2$' [Based on homomorphic Encryption to share the secret values where, $c_1 = qp_1+2r_1+d$ and $c_2 = qp_2+2r_2+d$. From homomorphic encryption $c = qp+2r+d$, where r_1 and r_2 are small negligible noise, which can be ignored on applications].
4. Again the prover has to compute and send the value of 's' to verifier, where $s = r + Cx = r + c_1c_2x$.
5. Now the verifier may accept the value of 's' if $g^s = hy^c$ is satisfied, otherwise the verifier can reject the transmitted value.
6. Finally, the verifier has to calculate the value of 'x' as:

$$x = \log_g y \tag{4}$$

4.2 Module III: Novel Authentication Scheme

In this module, a novel authentication conspire is introduced with IP address and timestamp as the predominant chunks of this module. The proposed method of authenticating a user is as follows:

1. Users obtain the IP address and timestamp value and has to be manipulated to generate a single digit as a final value.
2. Then, the timestamp will be extracted and the hour's value will be discarded to take 9's complement for each digit in remaining bits.
3. Then the subsequent bits in minutes and seconds will be divided separately to results in one-digit each.
4. The prime P value used in Elliptic curve will be divided with minutes and seconds values separately.
5. And those divided values will be added to get a single value.
6. Finally, the IP address and timestamp with hours discarded and the above manipulated value have to be appended.
7. So this value will be mutually verified by the user(sender and receiver) to prove the authentication with each other.

4.3 Description of the Proposed Work

In a prime field of Z_p where P is prime, an elliptic curve of $E(a, b)$ $y^2 = x^3 + ax + b$ is defined. And a point P_m is shared secretly between sender and receiver using modified Zero knowledge protocol of DLP. Also, two novel functions named as Function1 and Function2 are defined for the proposed key exchange mechanism, shown in Fig 6.

They are represented as follows: For $Q = (q_1, q_2)$, a point on the finite field, the proposed public key generation function, $Function1$ $F(Q)^x = q * (x - 1)$ times scalar multiplication of $(q_1, q_2) = q(x - 1)Q$ can be used. For the proposed shared secret generation, the $Function2$ $F(Q)^x = (x-1)$ times scalar multiplication of $(q_1, q_2) = (x - 1)Q$ where 'q' is calculated from (q_1, q_2). And the proposed key exchange mechanism has different levels of operations viz:
Setup Stage: Random secret key generation.
Level 1:Computation of Public Key and authenticated exchanging.
Level 2: Computation of shared secret key.
Level 3: Sharing of original data/Next level of key.

4.4 Setup Stage:

This stage relies on two operations. Firstly, let any one of sender or receiver defines a cyclic group G_q of order q with generator g and a prime number p are agreed upon sender and receiver through public channel. A Point $P_m = (p_1, p_2)$ on the elliptic curve has to be shared mutually by modified Zero knowledge of DLP protocol proposed in section IV as Module I: Modified Elgamal Method. Secondly, Random secret key has to be generated, separately on both side (Sender and Receiver). The steps for proposed random secret key generation for both sender side and receiver side are given below.

Step 1: Initially one has to generate a random series of N-digits.

Step 2: Then the generated series must be circular shifted for $\lfloor \frac{N}{2} \rfloor$ times.

Step 3: Until getting a two digit residue, say R, difference of the subsequent digits from Most significant Bit(MSB) to Least significant Bit(LSB) in the generated series has to be computed.

Step 4: Converting the N and R values into binary equivalent, so that the hamming distance between N and R (bit-by-bit difference) may be computed which gives the random key value H. This value of H can be used as randomly generated secret key.

4.5 Level 1: Computation of Public Key and Authenticated Exchanging

This technique depends on the idea of adding commutative property with some more exponentiation flavors for enhanced security purposes. Only point addition and scalar multiplication operations are possible in the elliptic curve, but points cannot be directly multiplied or squared or power operation cannot be performed. Therefore, for the purposes of this protocol, two new functions, namely *Function 1* (for public key computing) and *Function 2* (for shared secret key computing), are proposed to implement scalar multiplication based on the power value and coordinate value of the elliptic curve points. There are four significant operations in the public key computation and key exchanging. They are:

1. Computation of L value: $L = \frac{(\lfloor \log p \rfloor + 1)^2}{2}$, where 'P' is the Prime number.
2. Calculation of a new point Q: From the already defined Elliptic curve and a point $P_m = (p_1, p_2)$ from it, a new

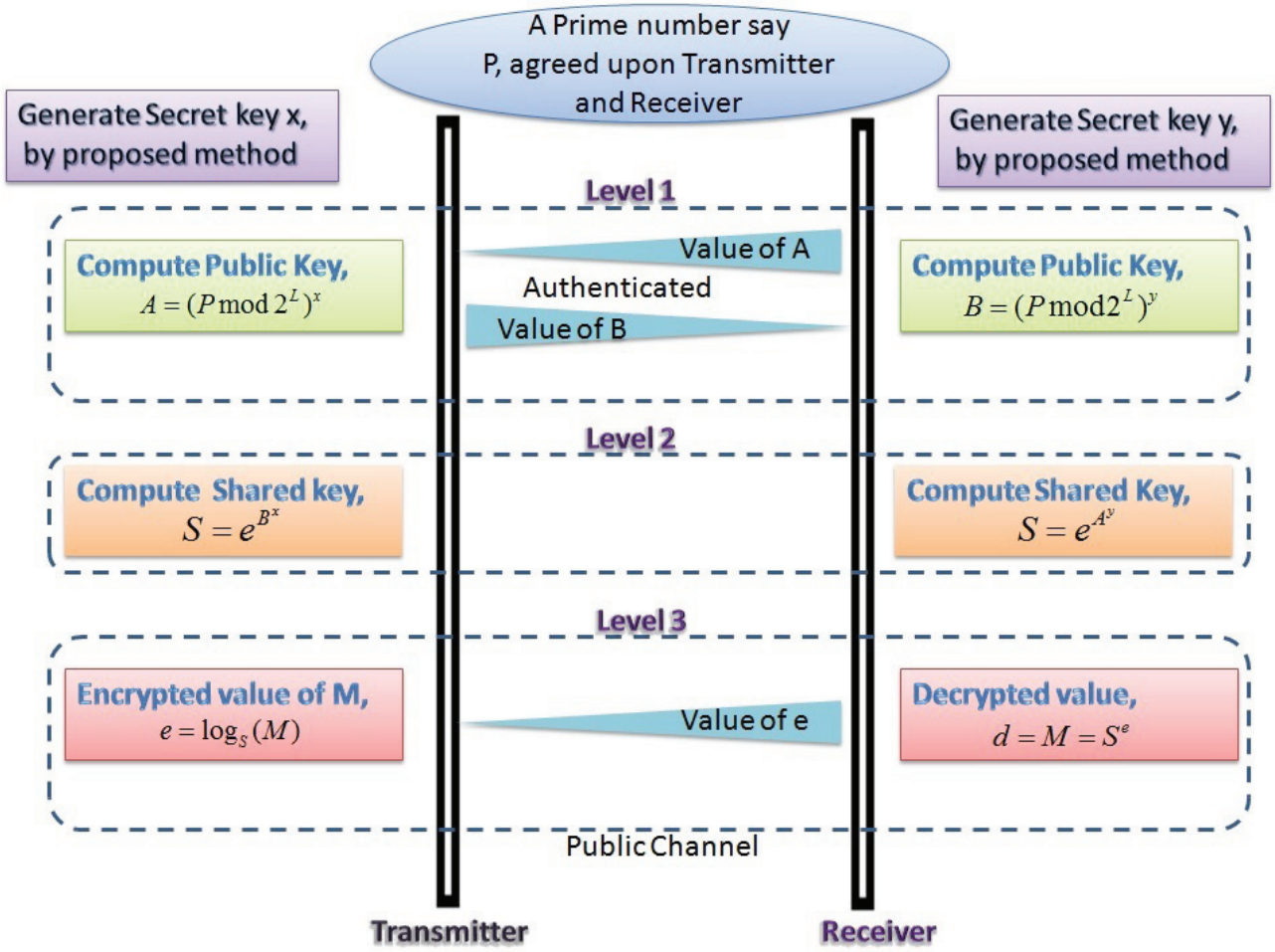

Figure 6 Proposed Key Exchange and Encryption Technique.

point $Q = (q_1, q_2) = P_m \mod 2^L = (p_1, p_2) \mod 2^L$ can be calculated.

3. Computation and exchange of Public key: $A = (Q)x$ and $B = (Q)y$, which is given by the Function 1 that involves the scalar multiplication operations depending on the value of secret keys and the co-ordinate values of the point Q. Here x and y are the secret key values generated from the previous stage(setup stage).

4. During the Public key transmission, the proposed novel authentication scheme explained in Module III can be used.

4.6 Level 2: Computation of Shared Secret Key

1. Sender and receiver must calculate the value of the shared secret key, $S = (s_1, s_2) = e^{B^x}$ and $S = (s_1, s_2) = e^{A^y}$ respectively and independently after the mutual transmission of the public key values A and B.

2. Shared secret key computation: The value of A^y and B^x has to be calculated using a novel scheme given by the Function 2 proposed in Module II, which is different from the technique used in public key computation.

3. The exponentiation function used here is to provide more security and this exponentiation function is calculated for the individual co-ordinate values of the obtained point, which gives again a point $S = (s_1, s_2)$ on the field of defined Elliptic curve.

After the calculation of Shared secret key, the value of '$A^y = B^x = q(x-1)(y-1)Q$' is shared between sender and receiver without letting known to the intruder. And this value can be used as an encryption key value for the encryption of future data transmission.

4.7 Level 3: Sharing of Original Data/Next Level of Key

1. With the consideration of a message(say M) to be encrypted using the shared secret key value '$S = (s_1, s_2)$' that is agreed between sender and receiver, The encryption and decryption operation are done as pursues:

2. Calculation of 's' value: For the shared secret value $S = (s_1, s_2)$ the value of 's' has to be calculated as $s = (s_1 + s_2)$, this 's' value is used in encryption and decryption operation by the sender side and receiver side respectively.

Figure 7 Public key and Shared Secret Key using the curve P-256.

3. Encryption of data M: At the sender end, the encrypted value can be represented as 'e' is calculated as,

$$e = log_s M \tag{5}$$

4. Decryption of original data: At the receiver end, the decrypted value can be represented as 'd' which is calculated as,

$$d = M = s^e \tag{6}$$

5. RESULTS AND ANALYSIS

5.1 Simulation Results

The proposed algorithms are evaluated using standard NIST curves on sample text data and the following statistics demonstrate the test results of the proposed algorithms with the two standard NIST curves as shown below.

5.1.1 NIST P-256 Curve

For the curve *NIST P-256* having the parameters of prime $P = 257; a = -3; b = 41058363725152142129326129780047268$ $34027684091144410159937255548352563140394674012 91$ forms the Elliptic curve equation

$$y^2 = x^3 - 3x + 41058363725152142129326129780047268 34027684091144410159937255548352563140394674012 91(mod257) \tag{7}$$

And the generator point and random secret key values chosen are *E= [255, 36]; x=9; y=8* with results of the proposed algorithm are explicated in *Fig. 7* and *Fig. 9*.

In the first stage of this work, the Novel key exchange algorithm computes the Public key and secondly the Shared Secret key using the curve P-256, which is shown in the *Fig.7*. Also, a sample text file shown in *Fig.8* is encrypted and decrypted to original data using the proposed key exchange and encryption algorithm in the curve *P-256*, so as to visualize that the file after decryption is same as that of the original input file, the hash value of both inputs of encryption and outputs from decryption algorithm using curve *NIST P-256* is shown in *Fig.9*.

5.1.2 NIST P-384 Curve

For the standard curve *NIST P-384*having the parameters of prime $P = 257; a = -3; b = 27580193559959705877849011840389048093056905856361568521428707301988689241309860865136260764883745107765439761230575$ forms the Elliptic curve equation

$$y^2 = x^3 - 3x + 27580193559959705877849011840389048093056905856361568521428707301988689241309860865136260764883745107765439761230575(mod257) \tag{8}$$

And the generator point and random secret key values chosen are: $E = 26247035095799689268623156744566981891852923491109213387815615900925518854738050089022388053$

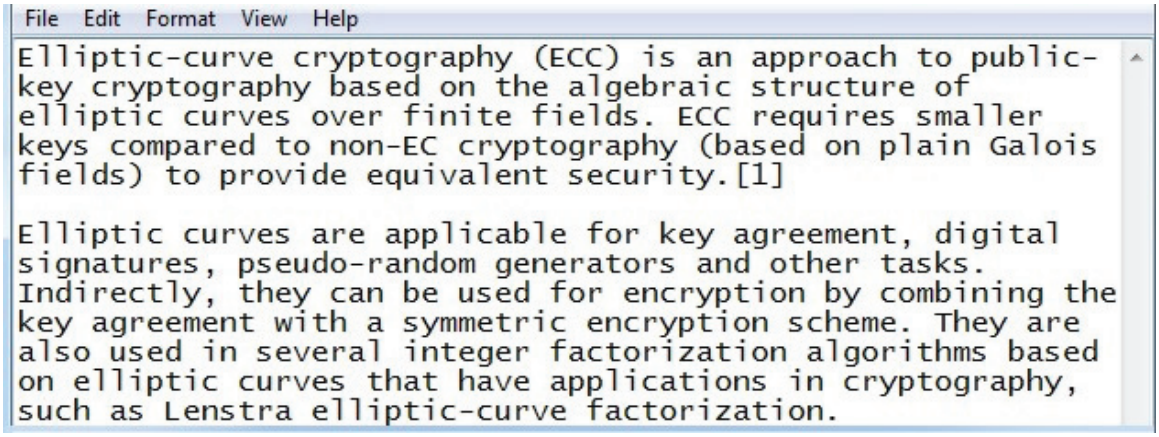

Figure 8 Sample data/text file used for encryption.

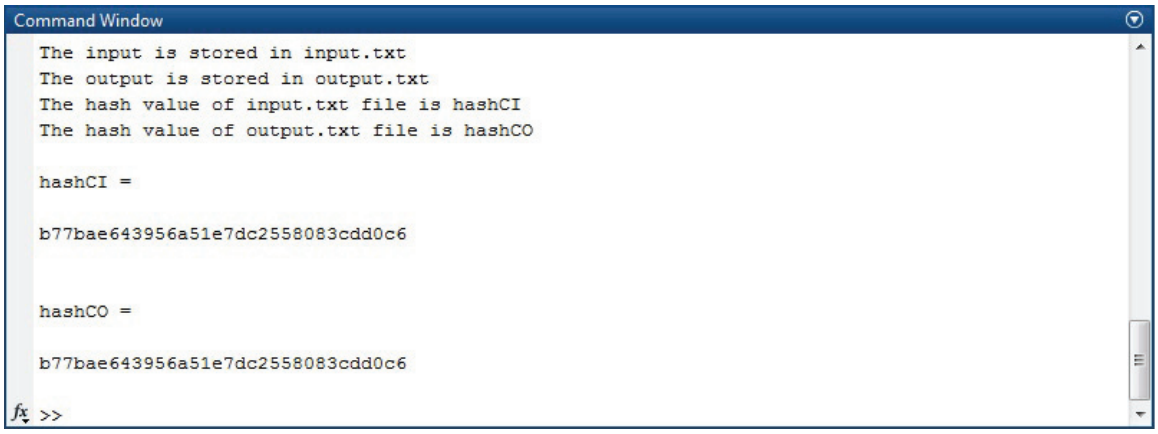

Figure 9 Hash values of Sample Text file and the output file from proposed protocol for curve P-256.

975719786650872476732087, 832571096148902998554675128952010817928785304886131559470920590248050319988441922443864376039294733307808651162787 1; $x = 92$; $y = 89$ with results of the proposed algorithm are given in *Fig.10*and *Fig.11*.

This Novel key exchange algorithm computes the Public key and shared secret key using the curve P-384, which is shown in the *Fig.10*. Also, a sample text file shown in figure 8 is encrypted and decrypted to original data using the proposed key exchange and encryption algorithm in the curve *NIST P-384*. To visualize that the file after decryption is same as that of the original input file, the hash value of both input to encryption and output from decryption algorithm is exhibited in *Fig.11*.

5.2 Security Analysis

Key exchange protocols are depicted to give at least two authorized parties to convey over an open communication channel with a common and asymmetrically secret key which may successively utilized to accomplish some cryptographic objectives, for example, secrecy or data integrity. Secure and verified key exchange conventions are essential as a successful substitution for conventional key exchange methods to overcome the following security attributes [17] and the comparison of the security parameters concerning with different protocols are shown in the table 1.

5.2.1 Implicit Key Authentication Attack

One of the features of the key exchange protocol is that the secret keys can access only by the guaranteed entities. An authentication mechanism has been introduced in the proposed protocol to check and verify the identity of each and every entity to communicate with another entity and thus the implicit authentication attack cannot applicable to this protocol.

5.2.2 Known-Key Security Attack

Nobody can interrupt the public key or other communication between the entities until the third party intruder knows the secret key [18]. A random secret key generation procedure is introduced in this proposed protocol, which can provide secret keys to sender and receiver with sufficient randomness to the secret key values used in the protocol, by which the intruder is unable to know about the secret key values.

5.2.3 Forward Secrecy Attack

Forward secrecy (FS) or perfect forward secrecy (PFS) is an attribute of key agreement protocols to ensure that the session keys won't be imperilled even if the private/secret key is endangered [19]. In the proposed work, for every user initiating session, a random key will be produced to avoid data loss. Regardless of whether the single specific session key is endangered, it will not impact some other information that is

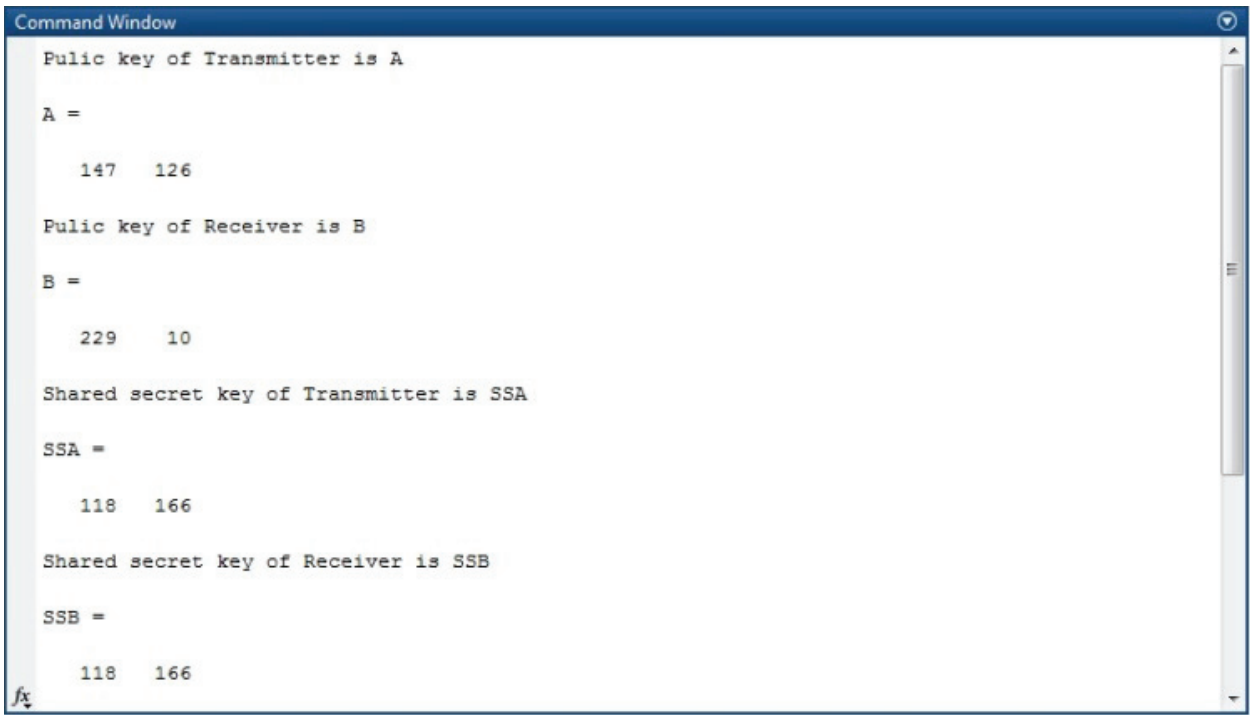

Figure 10 Public key and Shared Secret Key using the curve P-384.

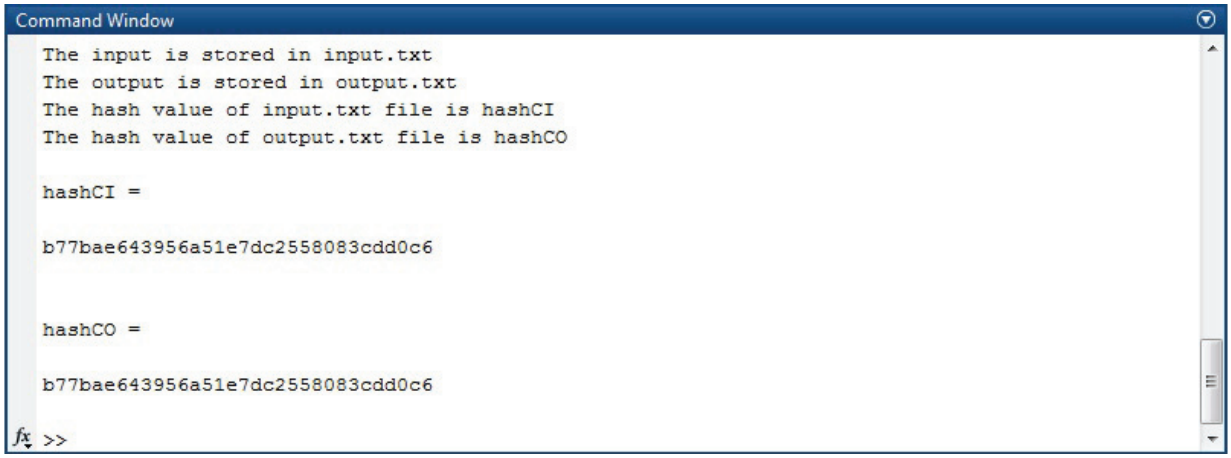

Figure 11 Hash values of Sample Text file and the output file from proposed protocol for the curve P-384.

traded in that specific session ensured by that specific key. So this protocol can maintain the forward secrecy along the data to be secured.

5.2.4 Key-Compromise Impersonation Attack

By the proposed scheme, all the nodes of a communicating network are given a novel method of authentication to share their identity with each other nodes. This information is utilized to authenticate other nodes mutually in the key agreement step. So that even if the key is impersonated or changed by third party intruder over the communication, the identity of intruder will not match with the already shared identity of authenticated nodes which in turn will help to find the impersonation in key values. Therefore, impersonation cannot succeed in the proposed protocol.

5.2.5 Unknown Key-Share Attack

As explained in [20], there is a possibility of intruders can intentionally coerces between two honest parties in a network to establish a new secret key, where at least one nodes does not know that the secret key is sharing with the intruder and this is called unknown key-share(UKS) attack. In the proposed work, it has been designed a multilevel key exchange and encryption methods so that multiple levels of authentication is also carried out to avoid the interruptions due to intruders and so this method is totally resist from this UKS attack.

5.2.6 Securely Change/Update Shared Secret Key

There is an arrangement for the nodes to refresh or change their secret key for a particular time gap, so that on changing every secret key the shared secret key needs to update for the following

Table 1 Comparison of Security Attributes.

Parameters/Protocols	*Proposed Work*	*Wang et al Protocol [21]*	*Law et al protocol [22]*	*Strangio Protocol [23]*	*Song et al Protocol [24]*
Implicit Key authentication	Yes	Yes	No	Yes	No
Known-key security	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes	Yes
Unknown key-share	Yes	Yes	No	Yes	Yes
Key-compromise impersonation	Yes	Yes	Yes	No	No
Update Shared Secret Key	Yes	No	Yes	Yes	Yes

Table 2 Comparison of Computational Attributes.

Operations/Protocol	*Proposed Work*	*Wang et al Protocol [21]*	*Law et al protocol [22]*	*Strangio Protocol [23]*	*Song et al Protocol [24]*
Scalar Multiplication	3	3.5	3	5	4
Field Inversion	0	1	1	0	0
Hash Function	0	2	0	2	0
Exponentiation	1	0	0	0	0

Table 3 Computational Cost Analysis.

Parameters/Protocol	*Proposed Work*	*Wang et al Protocol [21]*	*Law et al protocol [22]*	*Strangio Protocol [23]*	*Song et al Protocol [24]*
Total Computation cost for all the operations of the protocols.	$T_e + 3T_{SM}$	$3.5T_{SM} + 1T_{FI} + 2T_H$	$3T_{SM} + 1T_{FI}$	$5T_{SM} + 2T_H$	$4T_{SM}$
Computation cost (in Seconds)	0.20845	0.2289675	0.19679	0.316015	0.2523

time interval. To be specific, the nodes specified in this paper can generate another secret word to the connected network for specific time gap and afterward the hub registers new estimation of secret shared key and stores them for the ongoing session.

5.3 Communication Cost Analysis

Some of the attributes of security protocols also impact the communication cost, which can be clearly studied by analysing the parameters affecting the communication. Some of the parameters are depicted below.

5.3.1 Minimal Number of Passes

The number of passes represents the number of messages exchanged. In a key exchange or encryption algorithm, the more the levels of exchanging keys and data, more the communication cost. Hence in the proposed protocol, even it gives multilevel security it costs only three passes at key exchange level and one pass for encryption level.

5.3.2 Low Communication Overhead

Total number of bits transmitted between the nodes should be lesser for an optimized performance of a network is defined as low communication cost. So an efficient algorithm should have a low communication overhead similar to the proposed algorithm, which uses the Elliptic curve parameters to carry the data and secret keys so that *NIST P-256* curve and*NIST P-384* curve will consume a 128-bit and 192-bit data for each passes respectively.

5.4 Computational Cost Analysis

The total computation time needed for each phase of the given algorithm is outlined as the computational cost. And this computational cost analysis can be illustrated using the Table 2, 3 and using the following attributes.

5.4.1 Low Computation Overhead

Total number of arithmetic operations involved in an algorithm is preferred as low as possible. Because if the number of operations is more, then computation cost gets increased implies that increase in the computation time and energy of the machine in which the algorithm is running. But unfortunately, the elliptic curve field arithmetic requires more computation than other similar protocols. Even though the proposed protocol is utilizing the elliptic curve computations for maintaining the security, this issue has been considered while designing the protocol and makes with low computational overhead than other related protocols using the Elliptic curve filed arithmetic.

The computation time of the presented algorithm and other related algorithm is calculated based on the finite field scalar multiplication, field inversion, hash operation and exponen-

tiation operation. An ECC scalar multiplication requires 0.063075 seconds [26-28], field inversion takes 0.007565 seconds [29]. And the hash function requires 0.00032 seconds and the modular exponentiation operation involved in this novel algorithm takes 0.0192 seconds [30]. In this paper, the computation time for normal multiplication operation can be neglected, which involves negligible time duration than the other operations. In table 3, the gross computation time for all the operations of related protocols and the proposed protocol are depicted with the notations as T_e T_{SM} T_{FI} T_H which denotes the Computation time requires for Exponentiation operation, Scalar Multiplication operation, Field Inversion, Hash function respectively. The protocol in [21], [23] and [24] takes more time than the proposed algorithm and the techniques used in [22] takes lesser time than the proposed one but lags more in security. So the proposed key exchange protocol works well with added flavors of exponentiation operation and authentication mechanism for key authentication along with a novel secret key generation technique.

5.4.2 Possibility of Pre-Computation

To avoid the on-line delay due to computation time, the pre-computation is required. The proposed protocol has multiple levels of key exchange phase and each level of computation depends on the data shared in previous levels. Hence, the pre-computation is not possible in the proposed protocol.

6. CONCLUSION

In this paper, the proposed multi level key exchange and encryption protocol has been implemented for two standard NIST prime curves viz *P-256* and *P-384* and the results are given correspondingly with sample text data. Therefore, this can be used as a secured data transfer protocol in application layer of the network for various security requirements of IoT applications and other lightweight applications because the usage of ECC in multilevel key exchange and encryption costs with lesser communication and computational overhead shields the protocol from possibilities of leakage of secured data.

Also, a novel random key generation technique and an authentication scheme are presented in this paper, which reinforces the algorithm to be even more secured system. This work can withstand different possible attacks elaborated in section V and achieves a better overall performance and security compared to the well-known standardized protocols in [21], [22], [23] and [24]. Future work includes the hardware implementation of proposed protocol in real time applications.

7. ACKNOWLEDGMENTS

This work was supported by Indian Space Research Organization(ISRO) under the RESPOND scheme of Projects with the grants No: ISRO/RES/3/750/17-18.

REFERENCES

1. W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, Vol. IT-22, 1976, pp. 644–654.
2. M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password based authenticated key exchange in the three-party setting,” in International Workshop on Public Key Cryptography, Springer, 2005, pp. 65–84.
3. R. Lu and Z. Cao, “Simple three-party key exchange protocol,” Computers & Security, vol. 26, no. 1, pp. 94–97, 2007.
4. H.-R. Chung and W.-C. Ku, “Three weaknesses in a simple three-party key exchange protocol,” Information Sciences, vol. 178, no. 1, pp. 220–229, 2008.
5. H. Guo, Z. Li, Y. Mu, and X. Zhang, “Cryptanalysis of simple three-party key exchange protocol,” Computers & Security, vol. 27, no. 1, pp. 16–21, 2008.
6. T.-Y. Chang, M.-S. Hwang, and W.-P. Yang, “A communication-efficient three-party password authenticated key exchange protocol,” Information Sciences, vol. 181, no. 1, pp. 217–226, 2011.
7. E.-J. Yoon and K.-Y. Yoo, “Cryptanalysis of a simple three party password-based key exchange protocol,” International Journal of Communication Systems, vol. 24(4), pp. 532–542, 2011.
8. W. Stalling, Cryptography and Network Security, Principles and Practice, 5th edition New Jersey: Prentice Hall, 2011.
9. N. Koblitz, “Elliptic Curve Crytosystems,” Mathematics of Computation, Vol.48, pp 203–209, 1987.
10. V. S. Miller, “Use of Elliptic Curves in Cryptography,” Advances in Cryptology - CRYTO ’85, Lecture Notes in Computer Science, vol. 128, Springer Verlag, pp 417–426, 1985.
11. IEEE P1363, “Standard Specifications for Public Key Cryptography,” 2000.
12. Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186–2, National Institute of Standards and Technology. 2000.
13. C. Paar, and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, 1st ed. Springer Publishing Company, 2009.
14. D. Hankerson, A. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, New York, 2004.
15. K. Kim, T. Matsumoto, Advances in Cryptology - ASIACRYPT ’96: International Conference on the Theory and Applications of Crypotology and Information Security, Kyongju, Korea, November 3 - 7, 1996.
16. J. Binder, H.P. Bischof, “Zero knowledge proofs of identity for ad hoc wireless networks.” 2003.
17. S. Tavares and H. Meijer, “Authenticated Diffie Hellman Key Agreement Protocols”, SAC’98, Springer-Verlag Berlin Heidelberg, LNCS 1556, pp. 339–361, 1999.
18. Knudsen, L.R., Rijmen, V.: Known-key distinguishers for some block ciphers. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324. Springer, Heidelberg (2007).
19. Menzies, Alfred, van Oorscot, Paul C, Vanstone, SCOTT, “Handbook of Applied Cryptography” CRC Pres. ISBN 0–8493-8523–7, 1997.
20. Blake-Wilson, S, Menezes, A, "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", Public Key Cryptography, Lecture Notes in Computer Science, 1560, Springer, pp. 154–170, 1999.
21. S Wang, Z Cao, M A Strangio, L Wang, “Cryptanalysis and Improvement of an Elliptic Curve Diffie-Hellman Key Agreement Protocol,” IEEE Communications Letters, Vol. 12(2), February 2008.
22. L. Law, A. Menezes, M. Qu, J. Solinas, and S. A. Vanstone, “An efficient protocol for authenticated key agreement,”

Des. Codes Cryptography, vol. 28, no. 2, pp. 119–134, 2003.
23. M. A. Strangio, "Efficient Diffie–Hellmann two-party key agreement protocols based on elliptic curves," in Proc. 20th ACM Symposium on Applied Computing (SAC), pp. 324–331, 2005.
24. B. Song and K. Kim, "Two-pass authenticated key agreement protocol with key confirmation," in Proc. Indocrypt'00, LNCS 1977, pp. 237–249, 2000.
25. B. Ustaoglu, "Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS," Cryptology ePrint Archive, Report 2007/123, 2007.
26. Li C.-T, Hwang M.-S, and. Chu Y.-P, 2008, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," Computer Communications. 31(12): 2803–2814.
27. Lee J.-S and Chang C.-C, 2007, "Secure communications for cluster based ad hoc networks using node identities," Journal of Network and Computer Applications. 30(4): 1377– 1396.
28. Li W.-M, Wen Q.-Y, Su Q and Jin Z.-P, 2012, "An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network," Computer Communications. 35(2): 188–195.
29. Zhi Li, John Higgins, M Clement, "Performance of Finite Field Arithmetic in an Elliptic Curve Cryptosystem", MASCOTS 2001, IEEE Xplore, August 2002.
30. Lee C.-C, Chen C. T, Wu P.-H, and Chen T.-Y, 2013, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices", IET Computer Digital Technology. 7(1): 48–55.

About the Authors

Poomagal, C.T received the M.E. degree in Applied Electronics from Anna University, Chennai, in 2014. She is currently pursuing the Ph.D. degree in Information and communication Engineering at Anna University, Chennai, INDIA.. Her research interest includes VLSI design circuits, information security and Cryptography.

Sathish Kumar, G.A. obtained his M.E degree from Anna University, India. He has completed PhD in Anna University, Chennai. Currently, he is working as professor in Sri Venkateswara College of Engineering, Sriperumbudur. His research interest is Network Security, Image Processing, and VLSI Signal Processing Algorithms.

Deval Mehta working as a Head of the Satellite Communication Technology Division, SAC, Indian Space Research Organisation(ISRO), Ahmedabad. His research interests are Public key Cryptosystems, Hardware Security.