Lightweight secure message transfer protocol based on Ecc in Internet of Things equipped with satellite communications

Mahdi Baghaei Jezehei (✉ baghaee90@gmail.com)
Islamic Azad University South Tehran Branch

Seyed Ahmad Olamaei
Islamic Azad University South Tehran Branch

Ali Broumandnia
Islamic Azad University South Tehran Branch

Research Article

Keywords: ECC Elliptic Curve Cryptography, Internet of Things, Cryptographic Protocols, Satellite and Information security, Satellite Communications

Posted Date: May 16th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2926208/v1

Additional Declarations: No competing interests reported.

Lightweight Secure Message Transfer Protocol Based On Ecc

In Internet Of Things Equipped With Satellite Communications

Mahdi Baghaei Jezehei [1] Dr. Seyed Ahmad Olamaei [2] And Dr. Ali Broumandnia [3]

1 Islamic Azad University-South Tehran Branch / Baghaee90@Gmail.Com
2 Islamic Azad University-South Tehran Branch / Sa_Olamaee@Azad.Ac.Ir
3 Islamic Azad University-South Tehran Branch / Broumandnia@Gmail.Com

Abstract

With the expansion of Internet of Things (IOT) services and the use of satellite communications, according to the regional or continental extent of these services, the need for lightweight encryption has increased. In satellite communications, due to long distances, there are limitations in applying security, so heavy encryption algorithms such as RSA[1] cannot be trusted for security. ECC[2] elliptic curve cryptography provides a lighter alternative by invoking a mathematical problem called the ECDLP[3] elliptic curve discrete logarithm problem that cannot be solved in sub exponential time. Here, we propose a new strategy for secure IOT data communication between a satellite link and a terrestrial link that uses the principles of ECC elliptic curve cryptography and the NIST P-256 standard for key agreement and encryption for transmitting messages over the satellite communication platform.

Keywords ECC Elliptic Curve Cryptography, Internet of Things, Cryptographic Protocols, Satellite and Information security, Satellite Communications

1 Introduction

Internet of Things equipped with S-IOT[4] satellite communications are growing as an important part of services in the field of Internet of Things. This technology can be used in many applications, including smart environments, healthcare, drones, military centers, etc. An important point that is important in the provision of S-IOT services is that the security in this area is very fragile due to the long distance between the source and the destination, and the authentication mechanisms must be selected in such a way that the low capacity of the Internet of Things devices, low energy and power Low processing and delay in sending and receiving messages to the minimum possible. To ensure the safety of data, there is a need for encryption, so the cryptographic algorithm used in a satellite communication must be complex, low-power, and overall lightweight. Attacks on the Internet of Things equipped with satellite communication are conceivable. Usually, asymmetric algorithm such as RSA algorithm (which is based on integer factorization problem) and DSA[5] (which is based on discrete logarithm problem) is used for data transmission in terrestrial communication. As a result, to establish a relative security in a terrestrial connection, the key size for both RSA and DSA algorithms is recommended to be at least 2048 bits. Therefore, the system that uses these algorithms has a long key length and a lot of calculations. In S-IOT devices due to limited resources, there is a need for a lightweight encryption algorithm. Elliptic curve cryptography ECC provides a lightweight port function based on the discrete logarithm problem of ECDLP elliptic curve. The key size in the ECC algorithm is significantly smaller than many other encryption algorithms such as RSA. Elliptic curve cryptography is a public key encryption method that uses smaller keys for encryption than other encryption techniques that use relatively larger keys. As a result, the keys used for ECC are much smaller compared to the keys used by the alternatives. ECDSA[6] is a popular method used in many applications for authorization and user identification, but the proper exploitation of the

[1] Rivest-Shamir-Adleman
[2] Elliptic Curve Cryptography
[3] Elliptic Curve Discrete Logarithm Problem
[4] Satellite Internet of Things
[5] Digital Signature Algorithm
[6] Elliptic Curve Digital Signature Algorithm

ECDSA standard in a satellite communication requires changes and improvements. to avoid the possibility of revealing the private key when two communication links are connected. Here, using the random selection of integers, and using the NIST P-256 standard and improving the efficiency of the ECC encryption algorithm in satellite communications, the proposed algorithm can create higher reliability for authentication. Whenever the random integer key is reused, it resists MITM[7] attacks [13][12][1].

2 Leo Orbit Communication Satellites

Satellites are moving around the earth in a closed path, which is called an orbit. Generally, satellites are placed on four types of orbits that depend on the type of satellite application:

- LEO Low Earth Orbit
- Polar orbit POLAR
- GEO Earth Station Orbit
- Elliptical orbit

Elliptic curve algorithm behaviors allow them to generate unique sequences that are in no way inferior to modern cryptographic programs.

In this article, there is research done on LEO orbit satellites. Low Earth Orbit (LEO) satellites are called low earth orbit satellites. The highest height of this type of satellites is between 300 and 1200 km from the earth's surface. The movement path of these satellites is from west to east and in the same direction as the earth's cycle. The time for one revolution around the earth in these orbits is about 90 minutes. These orbits are located at a relatively low altitude, as a result, relatively heavy objects can be placed in those orbits with a simple launcher system. These orbits are usually used for observation, satellite communication and military satellite activities.

Due to the close distance of these types of satellites from the earth's surface, the movement speed of these satellites is much higher than the speed of the earth's rotation around itself; sometimes their speed reaches 27,000 kilometers per hour.

Advantages Of Using Leo Orbit: Satellites need the lowest amount of energy compared to other orbits to be placed in LEO orbit. Among its other advantages is the provision of high data bandwidth and low communication delay. The LEO orbit is used by many communication services, such as the Iridium phone system. Some communication satellites use GEO geocentric orbit geographic station orbits, which move at a speed equal to the speed of the earth and are always on the same area. And they have a higher delay.

The proposed satellite communication networks use LEO low earth orbit constellations. Satellites in GEO orbit have a high propagation delay, but a few satellites are enough to communicate around the world. Satellites in LEO orbit has less propagation delay due to their lower altitude, but many satellites are needed to provide global service. GSO satellites in geostationary orbit have a propagation delay of about 500 milliseconds, and satellites in LEO orbit have a delay of 50 milliseconds. Below are the specifications of satellites in three main orbits [14][10][2].

Circuit type	Altitude of orbit (km)	The number of satellites required to cover the entire earth	Timer (milliseconds)
Geostationary (GEO)	36000	3	500
Middle Earth Orbit (MEO)	5000 to 20000	6	80
low earth orbit (LEO)	300 to 1200	100	50

3 Problem Statement

In general, encryption in a satellite communication requires the use of a lightweight encryption due to limitations such as long distance and high latency. Elliptic curve is a lightweight algorithm that, in addition to having a short key length compared to other asymmetric algorithms such as RSA, guarantees a higher level of security. This algorithm emphasizes secure management services and advanced authentication. Advanced behavior allows them to generate unique sequences that are in no way inferior to modern encryption programs. In S-IOT communication, there are several prerequisites regarding security that should be labeled. Arrangements that provide security prerequisites must be made for the protection and security of users. Some of them are: data authentication, data integrity, data confidentiality, access control, data non-repudiation and data availability. Encrypting and decrypting data using elliptic curve cryptography along with cognitive cryptographic exchanges to achieve security services.

4 Elliptic Curve Encryption (ECC)

Encryption is the transformation of a simple message into an encrypted form to make it impenetrable and undetectable to intruders. In 1985, Victor S. Elliptic curve cryptography, ECC, was independently described by Miller and Neil Ku blitz. Elliptic curve cryptography is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC encryption (based on simple Galois fields) to provide equivalent security. ECC has many advantages. A short key size that can maintain a level of security indistinguishable from various types of public key cryptography, e.g., RSA, DH[8], and DSA. Therefore, ECC is particularly useful for remote devices, which typically have limited CPU, power, and system availability. ECC relies on the hardness of the ECDLP elliptic curve discrete entry problem. In elliptic curve cryptography, there are different types of curves. In this article, the Wei stress curve is used in the form of equation (1).

Equation 1

[7] Man-In-The-Middle

[8] Diffie–Hellman

$$y^2 = x^3 + ax + b$$

Where a and b are constants.

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

Calculations in elliptic curve cryptography are for finite field or Galva field. Public key cryptography is based on not solving specific mathematical problems. Prime public key systems are secure by assuming that it is difficult to factor a large integer consisting of two or more large prime factors. For elliptic curve-based protocols, it is assumed that it is impossible to find the discrete logarithm of a random elliptic curve element with respect to a commonly known base point, known as the Elliptic Curve Discrete Logarithm Problem (ECDLP). The general equation for EC over each GF is as shown.

$$y^2 = \{x^3 + ax + b\} \bmod\{p\}$$

The main advantage promised by elliptic curve cryptography is the smaller key size, which reduces storage and transmission requirements, i.e., an elliptic curve group can provide the same level of security as that provided by RSA-based systems with a large modulus and correspondingly larger key. [9][3]

Special formulas are used for arithmetic operations with given points in GF. Which are as follows:

4-1 Point Addition

P (x_1, y_1) and Q (x_2, y_2) are distinct colons. The following calculation gives the value P+Q=R (x_3, y_3). And the operation (Point Addition) is shown graphically in Figure 3a.

$$x_3 = \{\lambda^2 - x_1 - x_2\} \bmod p$$

$$y_3 = \{\lambda\,(x_1 - x_3) - y_1\} \bmod p$$

It can also be said that:

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)}\ mod\ p$$

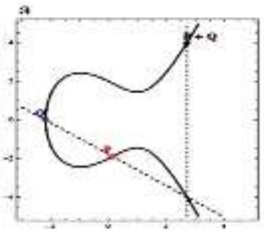


Figure 3a

4-2 Point Doubling

For the point P (x_1, y_1), when P+Q = R (x_3, y_3) is obtained by the following calculation. A graphical representation of the operation (point doubling) is shown in Figure 3b.

$$x_3 = \{\lambda^2 - 2x_1\} \bmod p$$

$$y_3 = \{\lambda\,(x_1 - x_3) - y_1\} \bmod p$$

It can also be said that:

$$\lambda = \frac{(3x_1^2 - a)}{(2y_1)}\ mod\ p$$

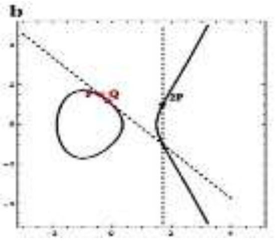


Figure 3b

4-3 Scalar Multiplication

The point of the elliptic curve P is multiplied. Repeated addition can be defined as scalar multiplication operation

$$k\,P = P + P + P + \cdots + P$$

In this article, the multiplication of the elliptic curve scale is indicated by a '¤' symbol. For example, k ¤ P is the scalar product of k with the point P.

4-4 Point at Infinity

When the points on the elliptic curve are x_1=x_2 and y_1=y_2=0 or x_1=x_2 and y_1=-y_2 the result is shown as infinity. The graphical representation (point at infinity) is shown in Fig 3c,3d.

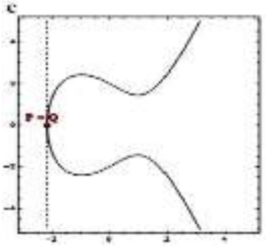


Figure 3c

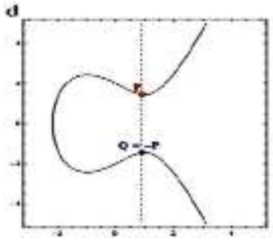


Figure 3d

5 Key Exchange Mechanism

Key exchange, also known as key generation, is a method in cryptography by which cryptographic keys are exchanged between two parties, enabling the use of a cryptographic algorithm.

If the sender and receiver want to exchange encrypted messages, each must be equipped to encrypt the messages sent and decrypt the messages received. The nature of the equipment they need depends on the encryption technique they may use. If they use the same code, they both need a copy of the same codebook. If they use a password, they need the appropriate keys. If the cipher is a symmetric key cipher, each pair requires a copy of the same key. If it is a key asymmetric encryption with the public/private key attribute, both require another public key.

The key cannot be sent via normal methods because the files sent between the two parties may end up in the wrong hands and thus be decrypted. Therefore, an alternative method should be easy to use, safe and highly scalable. It should also be designed for fast, connected, but highly insecure Internet highways. Otherwise, it would not be suitable for commercial use, as sensitive and high-volume transactions are often made on a daily or even hourly basis over very large intervals. There are different ways to send and receive keys, which can be mentioned below.

- Key exchange with SSL[9]
- Diffie-Hellman key exchange
- key exchange QKD[10]

Each of the above methods has advantages and disadvantages that can have a significant impact on creating security. The point that is important here regarding the S-IOT connection, considering the mentioned limitations, one should be careful in choosing the key that, in addition to solving the concerns in the satellite link, also has the minimum delay in sending and receiving information [15][6][5][4].

6 Analysis of Standard Elliptic Curve ECC (NIST P-256)

There are many elliptic curves for use in ECC proposed by different standards. The types of curves are basically classified based on the size of the first field and the shape of the curve. Some of the standards that are most used in the elliptic curve are given below:

1. M221 Curve
2. Nist P-224 Curve
3. Curve 25519 Curve
4. Bn (2,254) Curve
5. Brain pool P256t1 Curve
6. Nist P-256 Curve
7. Secp256k1
8. Secp256r1
9. Nist P-384 Curve
10. M-511 Curve

Each selected curve has different field sizes which are nothing but ECC key sizes. The selected curves are analyzed by performing two algorithms used in ECC, namely ECDH and ECDSA.

6-1 ECDH Algorithm

ECDH is a key agreement protocol that defines how keys are generated and exchanged between two parties. Each selected curve is used on its original field suggested by the existing standard. A random point P is selected from the curve.

To exchange the key between Alice and Bob using the ECDH algorithm, the following procedure is performed:
1) Alice and Bob generate their private and public keys.
2) Alice has a private key as $A(K_{Pr}) = a$ and a public key as $A(K_{Pb}) = a*P$. Bob has a private key as $B(K_{Pr}) = b$ and a public key as $B(K_{Pb}) = b*P$.
3) The public keys $A(K_{Pb})$ and $B(K_{Pb})$ of Alice and Bob are exchanged over an insecure channel.
4) On each side, to recover the shared keys, Alice calculates $S = a*(b*P)$ and Bob calculates $S = b*(a*P)$.

Therefore, the shared cipher S is the same for Alice and Bob [18].

6-2 ECDSA Algorithm

ECDSA is used to sign the message hash, which is short. The hash bit length is equal to n bit length. The truncated hash is an integer and is denoted by z. A prime number q, an elliptic curve E mod q, a base point from the curve G, Alice private key d and public key H_A are used to implement the ECDSA between Alice and Bob. Performed by Alice to sign the message, ECDSA works as follows:
1) Take a random integer k from {1,…,n−1},where n is still The subgroup order;
2) Calculate the point P=k G;
3) Calculate $r = x_P \bmod n$, where x_P is the x-coordinate of P;
4) If r = 0, then choose another k and try again;
5) Calculate the value of $s = (z + r d) / k \bmod n$;
6) If s = 0, then select another value of k and try again.

The pair (r, s) forms the signature of the ECDSA. To verify the signature, Bob needs Alice's public key H_A, the hash z and the signature (r, s). For the verification of the signature, Bob performs the following procedure:
1) Calculate $u_1 = s^{-1} z \bmod n$;
2) Calculate $u_2 = s^{-1} r \bmod n$;
3) Calculate the point $P = u_1 G + u_2 H_A$.

The signature is valid only if $r = x_p \bmod n$, where x_p is the x-coordinate of the computed point [18].

6-3 Standard Nist P-256

NIST P-256 is an elliptic curve cryptographic curve (ECC) defined by the National Institute of Standards and Technology (NIST). It is also known as secp256r1 or prime256v1. The curve is defined on the finite field of the first order $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ and its equation is $y^2 = x^3 - 3x + b$ where b is the y coordinate of the base point of the G base point with NIST coordinates P-256 is widely used in various cryptographic protocols and applications such as SSL/TLS, SSH. With a key size of 256 bits and relatively fast cryptographic operations, it offers a good balance between security and performance. The initial curve of the NIST P-256 standard is considered. Therefore, the elliptic curve is shown as equation (11). Which is given below:

Standard Nist P-256 Equation (11):

1) $y^2 = x^3 - 3x + 41058363725152142129326129780047268409114441015993725554835256314039467401291$
2) modulo $p = 2^{256} - 2^{224} - 2^{192} + 2^{96} - 1$
3) EP (a, b) =EP (-3, 410583637251521421293261297800472684091144 4101599372555483525631403946740129 1)

With the generating point:

4) $G = (G_X, G_Y)$

$G_X=48439561293906451759052585252797914202762949526041747995844080717082404635286$

$G_Y=36134250956749795798585127919587881956611106672985015071877198253568414405109$

6-4 Comparison of standard ECC(P-256) compared to RSA

[9] Secure Sockets Layer

[10] Quantum Key Distribution

Here are several advantages of using P-256 over RSA, including:

1- Smaller key size: P-256 uses a key size of 256 bits, while RSA usually requires a key size of at least 2048 bits to achieve the same level of security. This means that P-256 can provide the same level of security as RSA with smaller key sizes, resulting in faster cryptographic operations and smaller message sizes.

2- Faster cryptographic operations: ECC algorithms such as P-256 can perform cryptographic operations such as encryption, decryption, and signing faster than RSA for the same level of security. This is because the math operations used in ECC are simpler and faster than those used in RSA.

3- Lower energy consumption: The smaller key size and faster encryption operation of P-256 results in lower energy consumption in devices that use it. This makes the P-256 a good choice for applications that require low power consumption, such as mobile devices and IOT devices.

4- Resistance to certain types of attacks: ECC algorithms such as P-256 are resistant to certain types of attacks, such as attacks based on number field sifting algorithms that can be used to break RSA. This makes P-256 a good choice for applications that need protection against these types of attacks [17][16].

In Table 1, a comparison has been made between two asymmetric algorithms:

ECC(P-256)	RSA
A newer public key cryptography method compared to RSA	A method for public key encryption
Works on the mathematical representation of elliptic curves.	It works based on the principle of the first factorization method.
ECC requires more time as it is complex in nature.	RSA can run faster than ECC thanks to its simplicity.
ECC is more secure than RSA and is adaptive. Its use is expected to increase in the near future.	RSA is known to be vulnerable and is nearing the end of its tenure.
ECC requires much shorter key lengths compared to RSA.	RSA requires much larger key lengths to perform encryption.

Table 1: Application comparison of RSA and ECC

7 Suggested Work

Here, a security algorithm for data transmission of Internet of Things equipped with satellite communications with ECC elliptic curve encryption model is proposed. Security is used at the beginning of the connection using the Diffie-Hellman key exchange method and smart registration and authentication using the hash value of the IP address of the node. Also, using the NIST P-256 standard in the ellipse curve, the hidden values of the EC parameters and the corresponding generating point for two satellite communication nodes have been added before the start of the main data transmission. And public keys can be generated and shared by exchanged point values and private keys with mutual authentication that cannot be predicted or guessed by a third party. The data should be transmitted by the proposed encryption method over a public channel backed by a public key. So that no eavesdropper can break the security through the public channel and cannot penetrate the data in it. Due to the limitations that exist in a satellite communication and the need for the used protocol not to have a complex structure. The proposed protocol mentioned here has less overhead and thus can be regarded as a lightweight security protocol that provides minimum latency for data communication over satellite networks. This protocol has four different operation steps as follows:

7-1 The initiation or preparation stage
7-2 initial parameters agreement stage
7-3 key exchange step
7-4 elliptic curve encoding step

7-1 The Start Stage (Preparation)

In the beginning phase, we will first introduce the parameters used in this article. The link or data sender reference (SA[11]) and data receiver reference (RA[12]) and transmission related parameters (CA_t[13]) and two EC parameters (a, b) as well as the converter point 'G' In the S-IOT network, there is an initial preparation stage. Key exchange parameters include Pub_{SA} public key of the sender (public key SA) and Pub_{RA} public key of the receiver (public key RA), also the private key is generated through SA and RA, and the shared encryption keys $E.Key_{SA}$ and $E.Key_{RA}$ can be calculated as follows:

7-1-1 Calculating the public key by using the private key "s" and "r" and then multiplying the score at the primary generating point G, Pub_{SA} = s ¤ G and Pub_{RA} = r ¤ G (¤ multiplication of the score) are calculated

7.1.2 Public keys are mutually shared between the SA and the RA over the public channel, so that even if a third party has access to the public keys, it cannot predict the secret key values.

7.1.3 When the public keys Pub_{SA} and Pub_{RA} are shared, SA and RA calculate the shared secret key $E.Key_{SA}$ and $E.Key_{RA}$, respectively, which is equal to SA and RA, so the secret key is not shared publicly. but is shared only with the SA and the secret key RA is shared by both ends to initiate secure transmission. At this stage, all the nodes that must be registered in the Internet of Things network equipped with the relevant S-IOT satellite communications will be the central SA and hub of the RA network. As a result, $E.Key_{SA}$ = S ¤ Pub_{RA} and $E.Key_{RA}$ = r ¤ Pub_{SA} are calculated in this way becomes

When registering and starting up the network, it is appended to the public keys of the IP address in the CA_t for further authentication. The central hub stores the IP hash values received from the registered nodes in a tabular format to verify the authenticity of the particular node in the future data transmission phase. Therefore, when transferring SA and RA public keys, they should be < Pub_{SA} ||, respectively ID > and ID < Pub_{SA} || ID > be sent to SA and RA.

[11] Sending Authority
[12] Receiving Authority
[13] Transmission Corresponding Authorities

Row	Abbreviation	Description
1	¤	Scalar multiplication operation in ECC
2	s, r	Sender and receiver secret key
3	Pub$_{SA}$	The public key of the sending node at the login stage
4	Pub$_{RA}$	The public key of the receiving node at the login stage
5	G = (g$_1$, g$_2$)	conversion point
6	a, b	Selected elliptic curve parameters
7	P$_{SA}$	The public key of the sender node
8	P$_{RA}$	The public key of the receiving node
9	P$_m$	A message to be sent via satellite communications
10	K	Unique hidden value
11	C$_1$, C$_2$	Cipher text
12	E.Key$_{SA}$	Shared encryption keys on the sender side
13	E.Key$_{RA}$	Shared encryption keys on the receiver side
14	ID	The corresponding IP address of the SA
15	h(ID)	The hash value of the corresponding IP address of the SA
16	⊕	Bitwise XOR operation

Table 2: Description of parameters in the proposed protocol

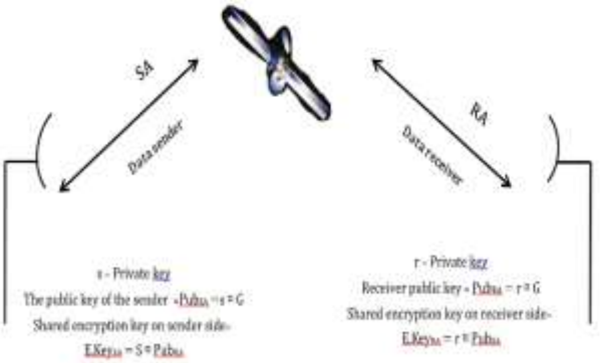

SA
Data sender
RA
Data receiver
s - Private key
The public key of the sender -Pub$_{SA}$ = s ¤ G
Shared encryption key on sender side-
E.Key$_{SA}$ = S ¤ Pub$_{RA}$
r - Private key
Receiver public key - Pub$_{RA}$ = r ¤ G
Shared encryption key on receiver side-
E.Key$_{RA}$ = r ¤ Pub$_{SA}$

Figure 1: The start phase of the proposed protocol

7-2 Initial Parameters Agreement Stage

In order to strengthen the security of Internet of Things equipped with S-IOT satellite communications, every time the message is transmitted, SA and RA must be transferred on an elliptic curve with a new generating point, which increases security, and these points and parameters must be transmitted over the public channel secretly. The elliptic curve has two parameters and b, and the generating point also has two values {for example, it has a coordinate bipoint (g$_1$, g$_2$)}, so twice the values must be agreed. Whenever these two values must be shared in common, it can be represented as "a" and "b" or (a, b).

In the initial parameter agreement phase, it uses the power operation in terms of a variable numerical value, and the EC parameter (a, b) and the converter point G = (g$_1$, g$_2$) are separately agreed here. So, it is usually represented as (a, b) which is shared twice for EC and Generator parameter. Hence (a, b) was secretly shared through the public channel. First, (a, b) will be the EC parameter, and secondly, (a, b) will be the generator point. These values should be shared publicly along with the corresponding IP address hash values. The RA then checks the received hash against the previously stored hash values in the hub. If those values are present in the checklist, it accepts the received data and processes it for further communication, otherwise it rejects it and reports about the node to the hub. The work steps are as follows:

Initial stage:

SA **Data sender authority**	**Available parameters include a, b, G**	**RA** **Data receiving reference**
s = Private key Receiver public key: Pub$_{SA}$ =s ¤ G Shared encryption key on sender side: E.Key$_{SA}$ = S ¤ Pub$_{RA}$	Pub$_{SA}$ ‖ ID ⟶ Public channel ⟵ Pub$_{RA}$ ‖ ID	r = Private key Receiver public key: Pub$_{RA}$ = r ¤ G Shared encryption key on receiver side: E.Key$_{RA}$ = r ¤ Pub$_{SA}$

Agreement stage of initial parameters:

a, b Power in terms of values		

X =(x$_k$)a mod P
Y =(y$_k$)b mod p
	X, Y ‖ h(ID)
	⟶
	Public channel
		a = log(X-x$_k$)
		b = log(Y-y$_k$)

Key exchange step:

s = Private key		**r = Private key**

The public key of the sender node
PSA =s ¤ G
Pm = E(M)
C$_1$ = E(M) ⊕ K ¤ PRA
C$_2$ = K ¤ G
	P$_{SA}$ ‖ h(ID)
	⟶
	P$_{RA}$ ‖ h(ID)
	⟵
	Public channel
	C$_1$, C$_2$
		The public key of the receiving node
		P$_{RA}$ =r ¤ G
		Decoding (C$_1$, C$_2$)to find Pm

7-3 Key Exchange Step

Before data can be transferred between satellite links, public keys must be shared between the SA and RA, using which the original data must be fully encrypted with the new encryption method. The parameter EC and the generator point agreed in the previous step SA and RA should create a new elliptic curve by which the corresponding public keys P$_{SA}$ and P$_{RA}$ are calculated and respectively as P$_{SA}$|| h(ID) and P$_{RA}$|| h (ID).

7-4 Elliptic Curve Encoding Step

In most cases, information must be transmitted through hubs in the S-IOT network, which consist of signal messages and data messages. The message to be transmitted is coded with any encryption method, and then the coded message is XORed with a secret value and public key to generate cipher texts. Now a cipher text C$_1$ and C$_2$ is generated and transmitted through the channel. At the receiving end, the cipher text (C$_1$, C$_2$) is received and decoded as in the encryption method, which reconstructs the original data, resulting in the original data being reconstructed without being disclosed to third parties or eavesdropping.

7-5 Security Analysis of The Proposed Protocol

The start (preparation) step of calculating the shared secret key in SA and RA is E.Key$_{SA}$ and E.Key$_{RA}$, respectively. In other words:

$$E.Key_{SA} = s*Pub_{RA}$$
$$E.Key_{SA} = s*r\ G$$
$$E.Key_{SA} = (x ‹y)$$
$$E.Key_{RA} = r\ Pub_{SA}$$

$$E.Key_{RA} = r*s\ G$$

$$E.Key_{RA} = (x،y)$$

where $Pub_{SA} = s ¤ G$ and $Pub_{RA} = r ¤ G$ and the product of the field is the displacement elliptic curve because every operation is on the displacement field. So, the secret key is shared at both ends without being revealed to the public channel. This shared secret key can be the basis for further information transfer in the S-IOT network, so sharing a secret key with other users in the network will be the initialization stage of the network.

In the initial parameter agreement step, if (a, b) is the generating point or parameter of the elliptic curve where the data is transmitted securely, it should be secretly transmitted from SA to RA.

So, we have:

1) SA is calculated X and Y which is $X =(x_k)a \bmod P$ and $Y =(y_k)b$ mod.

2) RA is calculated as a and b as $a = \log (X – x_k)$ and $b = \log (Y – y_k)$.

Proof:

$$X =(x_k)a \bmod P$$

$$\text{Log } X = a.\ \log (x_k) \bmod P$$

$$\frac{\log X}{\log x_k} = a.\bmod P$$

$$a = \log (X – x_k)$$

$$Y =(y_k)b \bmod P$$

$$\text{Log } Y = b.\ \log(y_k) \bmod P$$

$$\frac{\log Y}{\log y_k} = b.\bmod P$$

$$b = \log (Y – y_k)$$

In the key exchange and encryption step, the original message M is encrypted as cipher text C_1 and C_2.

$$P_m = E(M)، \{C_1 = E(M) \oplus KP_{RA} \text{ و } C_2 = KG\}$$

Proof to find Pm:

$$P_m = C_1 \oplus (C_2\ r)$$

$$P_m = E(M) \oplus K\ P_{RA} \oplus (C_2\ r)$$

$$P_m = E(M) \oplus K\ r\ G \oplus (K\ G\ r)$$

$$P_m = E(M) \text{ decoded at the other end}$$

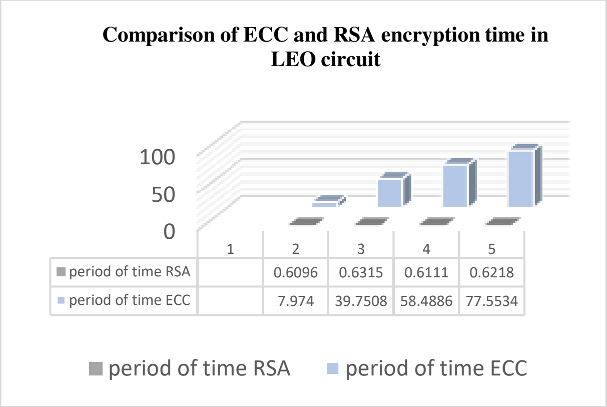
Hence, the original message M is securely decrypted at the other end. Here there is no chance of interruption of data transfer by intrusive third party. As a result, the message is sent with minimum delay and safely.

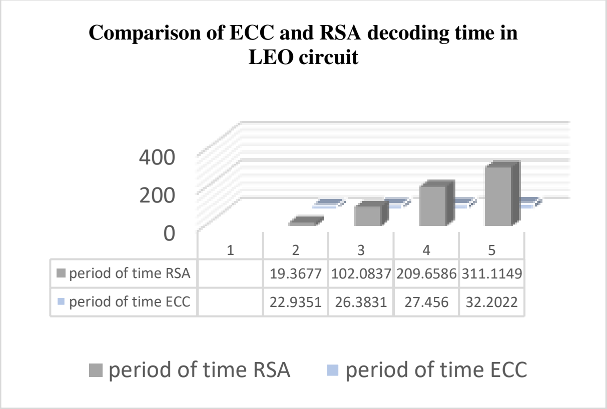
7-6 Comparison of The Proposed Protocol ECC (NIST P-256) With RSA Cipher Algorithm

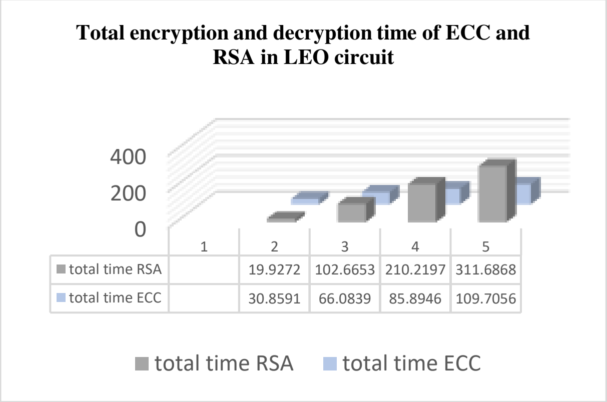
Here we implement RSA and the proposed ECC algorithm with NIST P-256 standard for information confidentiality with 256-bit data input and random private keys in LEO circuit. The efficiency of the proposed algorithm compared to RSA is shown in table 3 and graphs 1, 2 and 3. Based on the experiment, it was observed that RSA is very efficient in encryption but slow in decryption while the proposed algorithm is slow in encryption but very efficient in decryption. In general, the proposed algorithm is more efficient and safer than RSA, considering that in satellite communications, due to the long distance, we must apply the minimum time required for security. The algorithm, which is designed based on the elliptic curve, has a higher security level and less delay in sending than other asymmetric algorithms. And it is received.

Entrance :256 bits with LEO satellite orbit delay						
Security bit level	Encryption		decoding		total time	
	period of time ECC	period of time RSA	period of time ECC	period of time RSA	total time ECC	total time RSA
80	7.9240	0.5596	22.8851	19.3177	30.8091	19.8772
112	39.7008	0.5815	26.3331	102.0337	66.0339	102.6153
128	58.4386	0.5611	27.4060	209.6086	85.8446	210.1697
144	77.5034	0.5718	32.1522	311.0649	109.6556	311.6368

Table 3: 256 bits - encoding, decoding and total time (in seconds)







8 Conclusion

Message security is very important during its transmission in a satellite communication. An asymmetric message security cryptographic technique is provided here. To reduce the problems of key distribution and management and to ensure the confidentiality and integrity of a message, asymmetric key encryption with NIST P-256 standard with Diffie-Hellman key exchange mechanism is proposed. In this article, a comparative analysis was also presented between RSA and ECC. This test was performed to find the time interval during encryption, decryption of the message on the 256-bit input event with random keys based on the NIST P-256 standard. Based on this experiment, it was found that ECC is better than RSA in terms of operational efficiency and security with fewer parameters. In this project, we proposed a lightweight and secure communication algorithm for the S-IOT network-based node that uses elliptic curve cryptography for lower overhead costs with higher security. The security analysis of the presented algorithm shows that this method is more secure compared to the security scheme of the existing works. Our scheme provides additional features, such as a dynamic node addition step, mutual authentication between each node in the network, and secret key exchange. The performance analysis presented in this paper explains that the proposed work is lightweight, requires low communication cost and moderate computational cost. Practical analysis of the correctness and working interpretation of the presented algorithm is analyzed under different parameters of the network. Therefore, the security methods implemented in this paper will be satisfactory.

Reference:

[1] Ana Goulart, Anitha Chennamaneni, Damiano Torre, Byul Hur and Fadhil Y Al-Aboosi. (2022). On Wide-Area IoT Networks, Lightweight Security and Their Applications—A Practical Review Electronics 2022, 11, 1762. https://doi.org/10.3390/electronics11111762

[2] Juan A. Fraire, Oana Iova, Fabrice Valois,"Space-Terrestrial Integrated Internet of Things:Challenges and Opportunities," IEEE Communications Magazine (IF 9.03) Pub Date: 2022-09-12 , DOI:10.1109/mcom.008.2200215

[3] Yuhan Yan, "The Overview of Elliptic Curve Cryptography (ECC)" Journal of Physics: Conference Series 2386 (2022) 012019, DOI: 10.1088/1742-6596/2386/1/012019

[4]Nan Li, " Research on Diffie-Hellman Key Exchange Protocol" International Conference on Computer Engineering and Technology , DOI: 10.1109/ICCET.2010.5485276

[5]Jaime Díaz Arancibia Vicente Ferrari Smith Julio López Fenner 2019" On-The-Fly Diffie-Hellman for IoT" International Conference of the Chilean Computer Science Society (SCCC) DOI: 10.1109/SCCC49216.2019.8966440

[6] Turkanović, M., Brumen, B., & Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. Ad Hoc Networks, 20, 96–112. https ://doi.org/10.1016/j.adhoc .2014.03.009.

[7] Wazid, M., Das, A., Kumar, N., Odelu, V., Reddy, G., Par, K., et al. (2017). Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks. IEEE Access, 5, 14966–14980. https ://doi.org/10.1109/ACCES S.2017.27232 65.

[8] Xue, K., Ma, C., Hong, P., & Ding, R. (2012). A temporal credential-based mutual authentication and key agreement scheme for wireless sensor networks. Journal of Network and Computer Applications, 36, 316–323. https ://doi.org/10.1016/j.ins.2015.02.010.

[9] Yeh, H. L., Chen, T. H., Liu, P. C., Kim, T. H., & Wei, H. W. (2011). A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors, 11(5), 4767–4779. https ://doi.org/10.3390/s1105 04767.

[10] Bin Li, Zesong Fei, Caiqiu Zhou, Yan Zhang, "Physical Layer Security in Space Information Networks: A Survey," IEEE Internet of Things Journal (IF 10.238) Pub Date: 2020-01-01, DOI:10.1109/jiot.2019.2943900

[11] Yingying Chen, Minghu Zhang, Xin Li, Tao Che, Rui Jin, Jianwen Guo, Wei Yang, Baosheng An, Xiaowei Nie, "Satellite-Enabled Internet of Remote Things Network Transmits Field Data from the Most Remote Areas of the Tibetan Plateau," Sensors (IF 3.847) Pub Date: 2022-05-13, DOI:10.3390/s22103713

[12] Pietro Tedeschi, Savio Sciancalepore, Roberto Di Pietro, "Satellite-based communications security: A survey of threats, solutions, and research challenges," Computer Networks (IF 5.493) Pub Date: 2022-08-03, DOI: 10.1016/j.comnet.2022.109246

[13] Matthias G. Schraml, Robert T. Schwarz, Andreas Knopp, "Multiuser MIMO Concept for Physical Layer Security in Multibeam Satellite Systems," IEEE Transactions on Information Forensics and Security (IF 7.231) Pub Date: 2020-11-26, DOI:10.1109/tifs.2020.3040884

[14] Yan Zhang, Yong Wang, Yihua Hu, Zhi Lin, Yadi Zhai, Lei Wang, Qingsong Zhao, Kang Wen, Linshuang Kang, "Security Performance Analysis of LEO Satellite Constellation Networks under DDoS Attack," Sensors (IF 3.847) Pub Date: 2022-09-26, DOI:10.3390/s22197286

[15] Abdellahi Ahmed , Mohamedade Farouk Nanne , Bamba Gueye, "The effectiveness of a hybrid Diffie-Hellman-RSA-AES model," International Conference on Computer Communication and Informatics (ICCCI) Pub Date:2022-03-31,DOI: 10.1109/ICCCI54379.2022.9740762

[16] Mingxuan Ma " Comparison between RSA and ECC," 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT) Pub Date: 2021-01-29, DOI: 10.1109/AINIT54228.2021.00129

[17] E. Vidhya , S. Sivabalan , R. Rathipriya, "Hybrid Key Generation for RSA and ECC," International Conference on Communication and Electronics Systems (ICCES) Pub Date: 2019-07-19 , DOI: 10.1109/ICCES45898.2019.9002197

[18] Javed R. Shaikh, Maria Nenova, Georgi Iliev and Zlatka Valkova-Jarvis, "Analysis of Standard Elliptic Curves for the Implementation of Elliptic Curve Cryptography in Resource-Constrained E-commerce Applications," 2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS) Pub Date: 2017-11-13, DOI: 10.1109/COMCAS.2017.8244805