Elliptical curve cryptography for data confidentiality with output feedback mode

ABSRTACT

With the increase in usage of the Internet, Data security is a critical issue to ensure safe transmission of information through the internet. As more and more users connect to the internet it attracts a lot of cyber-criminals. So, for faster transmission of data and to provide high security level, we propose a mutual authentication mechanism based on Elliptic curve cryptography. Other techniques like Diffie-Hellman and RSA cryptographic methods are based on the creation of keys by using very large prime numbers. Hence, key creation requires a lot of computational power. Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to other cryptography algorithms like RSA. This Elliptical Curve Cryptography is used along with output feedback mode as it is that any bit errors that might occur during transmission are not propagated to affect the decryption of subsequent blocks. ECC with output feedback mode is very efficient public-key cryptography mechanism as it provides privacy and security with lesser computation overhead and can provide faster encryption for large blocks of data. The reason behind keeping short key is the use of less computational power, fast and secure connection, ideal for Smartphone and tablet too. The advantages of the Output Feedback mode's insensitivity to transmission errors and the applicability to bulk encryption of multiple users' transmissions. One of the advantages of output feedback mode is that if some bits of the ciphertext are garbled, only those bits of plaintext get garbled. This paper presents the ECC with output feedback mode for data confidentiality.

Keywords- Network security, Elliptical Curve Cryptography, Encryption, Decryption, Output Feedback mode.

1.Introduction

Cryptography is the science of hiding information in order to conceal it from unauthorized access. It is a technique of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it and stop others from accessing the data. Cryptography deals with a set of methods which enable us to store and transmit information while safeguarding it from intruders. That is, we can use cryptography methods to keep information private.

Asymmetric cryptography is a branch of cryptography where a secret key can be divided into two parts, a public key and a private key. The public key can be given to anyone, while the private key must be kept secret. Encryption with asymmetric cryptography works in a slightly different way from symmetric encryption. Someone with the public key is able to encrypt a message, providing confidentiality, and then only the person in possession of the private key is able to decrypt it.

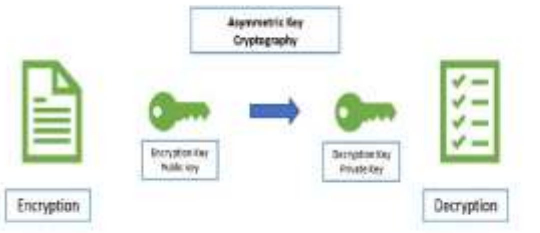

Asymmetric Key Cryptography
Encryption Key Public Key
Decryption Key Private Key
Encryption
Decryption

Figure 3.1: Asymmetric Key Cryptography

In this paper, we have used Elliptical Curve Cryptography (ECC) algorithm to encrypt the plain text. ECC is an algorithm is developed by Neil Kobiltz and Victor Miller in the 19th century. Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to other cryptography algorithms like RSA. The reason behind keeping short key is the use of less computational power, fast and secure connection, ideal for Smartphone and tablet too. And also, we have used Output feedback mode for encryption and decryption using Elliptical curve cryptography. Output feedback mode sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are sent instead of sending selected s bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases dependency or relationship of cipher on plaintext.

1.1Problem Definition

Big data require cloud that provides dynamically expanding data storage accessed through the internet. The outsourcing data in the cloud for storing makes the user data management easier and reduces the cost of maintaining data. Still organizations are not confident to store their data in cloud, because of security and privacy concerns.

However, existing encryption methods are able to protect data confidentiality, but it has some drawbacks of access patterns can also leak sensitive information.

Hence, we used Elliptical curve cryptography algorithm for encryption and decryption of data to improve the data security in cloud. This algorithm reduces the computational complexity and encrypted data efficiently.

1.2 Objective

To provide Network security we used ECC algorithm to encrypt the plain text. Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to other cryptography algorithms like RSA. ECC stands for Elliptic Curve Cryptography is the latest encryption method offers stronger security. If we compare to the RSA and DSA algorithms, then 256-bit ECC is equal to 3072-bit RSA key. The ECC certificates allow key size to remain small while providing a higher level of security and also, key creation method is entirely different from previous algorithms, while relying on the use of a public key for encryption and a private key for decryption. By starting small and with a slow growth potential, ECC has longer potential lifespan.

2. Literature Survey

The research papers help us to find the existing models and allow us to find the loop holes and guide us to develop a new thesis by overcoming the problems which have been found out in the survey.

2.1. Existing System

In Existing system, RSA (Rivest-Shamir-Adleman) algorithm is used to encrypt the data to provide security so that only the concerned user can access it. RSA is a block cipher, in which every message is mapped to an integer. It consists of public-key and private-key. The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially.

2.2 Disadvantages of Existing System

Even though RSA is highly secure and widely used, there are many problems in its implementation.

- The security of RSA is thought to be equivalent to the problem of factorizing the modulus n and the size of an RSA key usually is a function of the number of bits in the modulus.
- The brute-force attack over RSA can be easily overcome by increasing the key size but decryption time increases 8-fold as key size double. The entire processing time increases significantly as key size increase.
- Since, RSA is based on modular arithmetic with very long operands, the performance of RSA is quite slow on limited environments with low memory and processor power.

3. Proposed System

By considering the above existing systems, we used an Elliptical Curve Cryptography algorithm to encrypt the plain text. ECC is an approach to public-key cryptography based on the algebraic structure of elliptical curves over finite fields.

ECC requires small keys when compared to other cryptographic algorithms. The reason behind keeping the short key is the use of less computational power, fast and secure connection, ideal for smartphone and tablet too. We also used output feedback mode for encryption and decryption using ECC. It sends the encrypted output as feedback instead of actual cipher which is XOR output. In this mode, all bits of the block are sent instead of sending selected s bits.

3.1 Algorithms and Flowcharts

Key Generation:

Step 1: For user A, select a private key k(a), k(a)<n. And calculate public key P, P=k(a)*G

Step 2: For user B, select a private key k(b), k(b)<n. And calculate public key M, M=k(b)*G

Step 3: By using private keys and public keys of both users, generate a secret key K, where P1=K=k(a)*M and P2=K=k(b)*P.

Encryption:

Step 1: Divide the input message into block of equal size and store it as plain text.

Step 2: The initial vector is given as input along with secret key to encryption part of output feedback mode.

Step 3: The resultant output is given as input to the next step of encryption**.**

Step 4: And also, it is made XOR with plain text to generate cipher text.

Step 5: This process is continued until the end of plain text blocks.

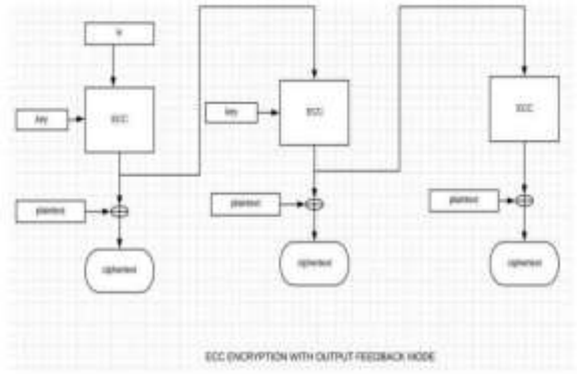


Figure. 3.1: ECC Encryption with Output Feedback Mode

Decryption:

Step 1: The generated cipher text is stored in blocks of equal size.

Step 2: The initial vector is given as input along with secret key to decryption part of output feedback mode.

Step 3: The output is given as input to the next step of decryption.

Step 4: And the resultant output made XOR with cipher text to generate plain text.

Step 5: This process is continued until the end of cipher text blocks

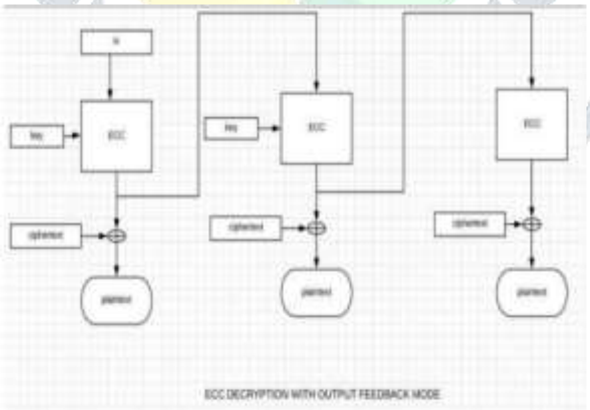
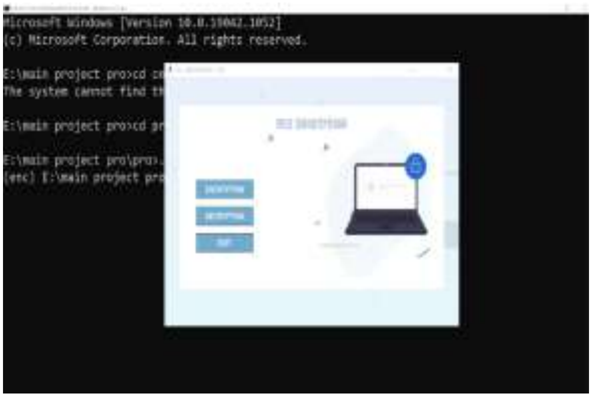


Figure. 3.2 ECC Decryption with Output Feedback Mode

4. Output Screens



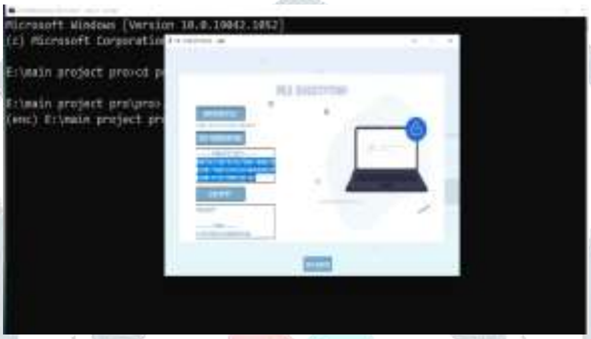
```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

E:\main project pro>cd cr
The system cannot find th

E:\main project pro>cd pr

E:\main project pro\pro>.
(enc) E:\main project pro
```

Screen 4.1: Main Interface

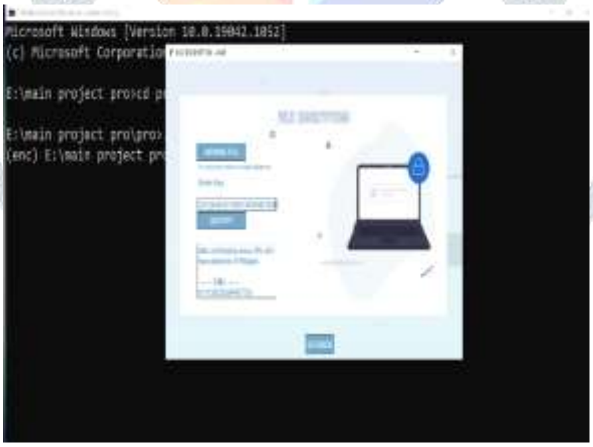


```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporatio

E:\main project pro>cd p

E:\main project pro\pro>
(enc) E:\main project pr
```

Screen 4.2: Time taken for Encryption



```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporatio

E:\main project pro>cd p

E:\main project pro\pro>
(enc) E:\main project pr
```

Screen 4.3: Time taken for Decryption

5. Conclusion

In this paper, we have provided a new cryptographic algorithm using elliptic curve cryptography using output feedback mode. We introduce a fast and secure way to transfer data using authenticated key agreement protocol based on elliptic curve cryptography with output feedback mode and provides mutual authentication and explicit key establishment.

In Elliptical Curve Cryptography, key size is less compared to other cryptographic algorithms and provides more security level than other algorithms. ECC is thus used for authentication on low computational devices because of its high security level with relatively less key length. Output feedback mode is used to encrypt data which is divided into blocks as the unauthorized user cannot easily decrypt the data. Hence ECC with output feedback mode provides better security level with less hardware requirements which can be used in low end devices.

6. Reference

[1] Victor S. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology-CRYPTO’85 Proceedings, Springer, vol. 218, pp. 417–426, December (2000).

[2] Neal Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, vol. 48, issue 177, pp. 203–209, January (1987).

[3] Neal Koblitz, Alfred Menezes and Scott Vanstone, The State of Elliptic Curve Cryptography, Designs, Codes and Cryptography, vol. 19, issue 2–3, pp. 173–193, (2000).

[4] Darrel Hankerson, Alfred Menezes and Scott Vanstone, Guide to Elliptic Curve Cryptography, Springer (2004).

[5] Lawrence C. Washington, Elliptic Curves Number Theory and Cryptography, Taylor & Francis Group, Second Edition (2008).

[6] Jorko Teeriaho, Cyclic Group Cryptography with Elliptic Curves, Brasov, May (2011).

[7] S. Maria Celestin Vigila and K. Maheswaran, Implementation of Text based Cryptosystem using Elliptic Curve Cryptography, International Conference on Advanced Computing, IEEE, pp. 82–85, December (2009).

[8] D. Sravana Kumar, Ch. Suneetha and A. Chandrasekhar, Encryption of Data Using Elliptic Curve Over Finite Fields, International Journal of Distributed and Parallel Systems (IJDPS), vol. 3, no. 1, January (2012).

[9] K. Jarvinen, Helsinki and J. Skytta, On Parallelization of High-Speed Processors for Elliptic Curve Cryptography, VLSI Systems, IEEE Transaction, vol. 16, issue 9, pp. 1162–1175, August (2008).

[10] M. Amara and A. Siad, Elliptic Curve Cryptography and its Applications, 7th International Workshop on Systems, Signal Processing and their Applications, pp. 247–250, May (2011).