Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas

Eureka Journal of Education & Learning Technologies (EJELT)

ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

PUBLIC KEY EXCHANGE AND SYMMETRIC CHOIR ENCRYPTION ALGORITHM BASED ON ELLIPTIC CURVE CRYPTOGRAPHY

Muhamediyeva D. T.
Tagayev F. A.
Tashkent Institute of Irrigation and Agricultural
Mechanization Engineers National Research University

Abstract

This article considers the process of secure key exchange based on Elliptic Curve Cryptography (ECC) and the algorithm for encrypting data using this key. In the study, a mechanism for generating a public key of the Diffie–Hellman type based on the operations of adding and multiplying points on an elliptic curve is implemented in the Python programming language. A public secret key is calculated between Alice and Bob by generating public and private keys. The process of encrypting and decrypting text data based on the generated public key is performed using the XOR symmetric algorithm. The proposed model can be an effective tool for explaining the principle of operation of cryptographic protocols, teaching cryptography, and modeling information security systems. The results of the study show the mechanism of operation of elliptic curve cryptography through small-parameter models and help to understand the theoretical foundations of modern secure communication protocols.

Keywords: Elliptic curve cryptography, ECC, Diffie–Hellman key exchange, cryptography, information security, XOR encryption, Python programming, public key, public key cryptography.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

Introduction

Along with the development of modern information technologies, the problem of data protection is also gaining relevance. The increase in the volume of data transmitted over Internet networks and the increase in cybersecurity threats require the development and improvement of reliable cryptographic methods. In particular, the issue of secure transmission of secret keys when exchanging information over open networks is one of the important problems of cryptography. Public key cryptography systems are widely used to solve this problem. Among such systems, elliptic curve cryptography (Elliptic Curve Cryptography – ECC) is of particular importance because it provides a high level of security with a relatively small key length. ECC algorithms are based on the complexity of the discrete logarithm problem on an elliptic curve and are therefore used in many modern secure communication protocols. Among the algorithms based on elliptic curve cryptography, the Elliptic Curve Diffie–Hellman (ECDH) key exchange protocol is widely used, which allows two parties to securely generate a common secret key over an open channel. This public key is then used to protect data in symmetric encryption algorithms.

In this work, the main mathematical operations of elliptic curve cryptography — dot addition and scalar multiplication — are modeled in the Python programming language. The process of generating public keys and generating a public key is carried out between Alice and Bob. A simple XOR symmetric algorithm is used to encrypt and decrypt text data based on the generated public key. The main goal of the research is to demonstrate the mechanism of operation of elliptic curve cryptography through a software model and to explain the process of cryptographic key exchange using a practical example. A simplified mathematical model of the key exchange process based on elliptic curve cryptography was developed. The public and private key generation algorithm based on dot addition and scalar multiplication on the elliptic curve was implemented in the Python programming environment. A model for integrating

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

the public key generated using the elliptic curve Diffie–Hellman protocol with a symmetric encryption algorithm was proposed. The proposed model demonstrates the practical mechanism of secure key exchange and data encryption based on elliptic curve cryptography. A software model for data protection using the XOR encryption algorithm using a public key generated based on elliptic curves was developed.

2. Methods

Elliptic curve cryptography (ECC) is a public-key cryptographic system whose security is based on the complexity of the discrete logarithm problem on an elliptic curve. In ECC algorithms, all calculations are performed on finite fields, and the operation of points on the elliptic curve is the basic mathematical operation.

An elliptic curve on a finite fundamental field F_p is defined by the following general equation:

$$y^2 \equiv x^3 + ax + b \pmod{p},$$

where:

a and $b-$ are the curve parameters,

$p-$ a prime number (modulus),

$x, y \in \mathrm{F}_p$.

The following condition must be met for the curve to be non-degenerate:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}.$$

The set of all points on an elliptic curve and the point at infinity together form an abelian group. Adding points on an elliptic curve

The main operation in ECC algorithms is adding points on an elliptic curve. If $P(x_1, y_1)$ and $Q(x_2, y_2)$ are two points on an elliptic curve, their sum is determined $R(x_3, y_3) = P + Q$ by the following formulas.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

If $P \neq Q$:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod{p}.$$

If $P = Q$:

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{p}.$$

The coordinates of the resulting point are determined as follows:

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

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}.$$

These formulas allow us to add points on an elliptic curve and perform scalar multiplication.

Scalar multiplication

One of the main operations of ECC algorithms is scalar multiplication of a point. If P is a point on an elliptic curve and k is an integer, then

$$Q = kP.$$

This is done by P adding the point to k itself:

$$kP = P + P + P + \ldots + P.$$

Scalar multiplication is the main cryptographic operation of ECC systems, and it is this operation that is based on the complexity of the elliptic curve discrete logarithm problem. If P and $Q = kP$ are known, finding k is very difficult.

Elliptic Curve-Based Key Exchange

In the Elliptic Curve Diffie-Hellman (ECDH) algorithm, two parties generate a shared secret key as follows.

1. Elliptic curve parameters $E(a, b, p)$ and starting point P are selected.
2. Alice chooses a private key d_A and calculates the public key:

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

$$Q_A = d_A P.$$

3. Bob chooses a private key d_B and computes the public key:

$$Q_B = d_B P.$$

4. The public secret key is generated as follows:

$$S = d_A Q_B = d_B Q_A.$$

As a result, Alice and Bob will have the same public key.

Using the public key in symmetric encryption

The generated public key is usually used in symmetric encryption. In this case, in a simplified form, the coordinates of the public key are used as a key in the XOR encryption algorithm. The XOR encryption algorithm is expressed by the following formula:

$$C = M \oplus K,$$

here

M — plaintext,

C — ciphertext,

K — sudden,

$\oplus$ — XOR money.

The decryption process is also performed through the same process.:

$$M = C \oplus K.$$

Thus, the public key generated using elliptic curve cryptography is used for secure encryption and decryption of data.

This section presents an algorithmic model of the process of public key exchange and data encryption using this key based on elliptic curve cryptography and its implementation in the Python programming language. The proposed model is based on the Elliptic Curve Diffie–Hellman (ECDH) principle and provides for the generation of a common secret key between two users - Alice and Bob. In the study, a scalar multiplication operation is performed using the operation of adding points on the elliptic curve. This operation forms the basis of the process of

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

generating public keys and generating a public key. The algorithm consists of the following steps:

In the Python program, adding points on the elliptic curve is performed using a separate function. This function accepts the coordinates of two points and returns the resulting point.

Algorithm steps:

If the points being added are equal, the point doubling formula is used.

If the points are different, the general addition formula is used.

Calculations are performed based on modular arithmetic.

As a result, new point coordinates are returned.

In the study, the algorithm is modeled in the Python programming language. The mathematical computing capabilities of the Python language and support for modular arithmetic make it convenient to create cryptographic algorithms.

The program consists of the following main modules:

1. Elliptic curve operations module. This module performs the operation of adding points on an elliptic curve.

2. Public key generation module. Includes functions for calculating public keys based on a secret key.

3. Public key calculation module. Generates a public secret key based on the ECDH protocol.

4. Encryption module. Performs the functions of encrypting and decrypting text data using the XOR algorithm. The XOR algorithm is one of the simplest forms of symmetric encryption. In this method, each byte is encrypted using the XOR operation with the key. The encryption algorithm consists of the following steps:

The text is converted into a sequence of bytes.

Each byte is recalculated by XORing with the key.

As a result, a sequence of encrypted bytes is formed.

The decryption process is also carried out using the same algorithm, since the XOR operation is its inverse. Sequence of operation of the algorithm

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

The proposed program works in the following sequence.
Determining the parameters of the elliptic curve.
Selecting the private keys of Alice and Bob.
Calculating the public keys.
Generating a public secret key.
Using the public key as an encryption key.
Encrypting the text using the XOR algorithm.
Decrypting the encrypted text.
As a result, a secure key exchange is performed between Alice and Bob, and using this key, data is transmitted in a protected manner.

3. Experimental Results and Analysis

This section analyzes the performance of the key exchange and encryption algorithm developed based on elliptic curve cryptography. The experiments were carried out in the Python programming environment, and the correct operation of the algorithm and the process of generating a public key were verified. The following parameters were selected in the experimental process (Table 1).

Table 1 Experimental process parameters

Parameter	Value
Elliptic curve parameter a	2
Modulus p	17
Starting point P	(5,1)
Alice's secret key dA	5
Bob's secret key dB	7

These parameters were chosen to be small values and used to illustrate the calculation process on an elliptic curve. In practical cryptographic systems, these parameters are very large values. After the algorithm is executed, the public keys for Alice and Bob are generated as follows (Table 2):

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

Table 2 Public key calculation results

User Public Key	User Public Key
Alice	$Q_A = (9,16)$
Bob	$Q_B = (0,6)$

These results were calculated by scalar multiplication of a point on an elliptic curve (Table 3). Alice and Bob calculated a shared secret key using each other's public key.

Table 3 Results of public key generation

Calculation	Result
Key calculated by Alice	$S_A = (10,6)$
Key calculated by Bob	$S_B = (10,6)$

The results show that both parties have the same public key. This indicates that the Elliptic Curve Diffie-Hellman algorithm is working correctly. The results show that both parties have the same public key. This indicates that the Elliptic Curve Diffie–Hellman algorithm is working correctly. The x coordinate of the generated public key was used as the key for the XOR encryption algorithm (Table 4).

Table 4 Encryption and decryption results

Process	Natija
Plaintext	Hello, this is a secure message!
Encryption key	10
Encrypted text	Baytlar ketma-ketligi
Decrypted text	Hello, this is a secure message!

The results show that the data encrypted using the XOR algorithm can be successfully recovered using the key. To demonstrate the effectiveness of the algorithm, the results of scalar multiplication on the elliptic curve were observed

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

for different secret key values. The following table lists the coordinates of the kP points for different k values (Table 5).

Table 5 Coordinates of the points kP

k	kP (point coordinate)
1	(5,1)
2	(6,3)
3	(10,6)
4	(3,1)
5	(9,16)

These results indicate that the process of multiplying points on an elliptic curve has a periodic nature. The experimental results showed the following conclusions: Public keys were successfully generated using the operations of adding points on an elliptic curve and scalar multiplication. Public keys independently calculated by Alice and Bob gave the same result. Text data is encrypted using the XOR algorithm based on the public key and then successfully decrypted. It is possible to show the mechanism of operation of the algorithm using small parameters, but in real systems very large modulus values are used to ensure security. Thus, the proposed software model demonstrates the principles of operation of elliptic curve cryptography in a practical way and allows modeling the process of secure key exchange.

4. Conclusion

In this study, a software model of the public key exchange process based on elliptic curve cryptography and a data encryption algorithm using this key were developed. During the study, a key generation mechanism based on the addition of points on the elliptic curve and scalar multiplication operations was implemented in the Python programming language. Experimental results showed that it is possible to successfully generate a public secret key between Alice and

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

Bob through the exchange of public keys. The fact that the public keys calculated by both parties have the same value confirms the correct operation of the elliptic curve Diffie–Hellman algorithm. The coordinate of the generated public key was used as a key for the XOR symmetric encryption algorithm, and the plaintext data was successfully encrypted and subsequently restored. The results of the study show that elliptic curve cryptography allows for a high level of security with a small key length. The proposed software model can serve as an effective tool for studying the mechanism of operation of cryptographic algorithms, for use in educational processes on information security, and for modeling cryptographic protocols. In the future, it is desirable to increase the security level of the system by using large-scale elliptic curve parameters for the development of this research, as well as by integrating it with AES or other modern symmetric encryption algorithms.

References

1. Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation. 1987, 48(177), 203–209.
2. Miller V. Use of elliptic curves in cryptography. Advances in Cryptology – CRYPTO. 1986, pp. 417–426.
3. Hankerson D., Menezes A., Vanstone S. Guide to Elliptic Curve Cryptography. Springer, 2004.
4. Stallings W. Cryptography and Network Security: Principles and Practice. Pearson, 2017.
5. Paar C., Pelzl J. Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2010.
6. Menezes A., Vanstone S., Oorschot P. Handbook of Applied Cryptography. CRC Press, 1996.
7. Bernstein D., Lange T. Elliptic curve cryptography. IEEE Security & Privacy. 2017, 15(3), 48–56.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Education & Learning Technologies (EJELT)
ISSN 2760-4918 (Online) Volume 2, Issue 3, March 2026

https://eurekaoa.com/index.php/2

8. Washington L. Elliptic Curves: Number Theory and Cryptography. CRC Press, 2008.
9. Boneh D., Shoup V. A Graduate Course in Applied Cryptography. Stanford University, 2020.
10. Katz J., Lindell Y. Introduction to Modern Cryptography. CRC Press, 2014.
11. Smart N. Cryptography Made Simple. Springer, 2016.
12. Bernstein D., Buchmann J., Dahmen E. Post-Quantum Cryptography. Springer, 2009.
13. Diffie W., Hellman M. New directions in cryptography. IEEE Transactions on Information Theory. 1976, 22(6), 644–654.
14. Johnson D., Menezes A., Vanstone S. The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security. 2001, 1(1), 36–63.
15. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley, 2015.