УДК 004.63

SPECIFICS OF IMPLEMENTATION OF THE ASYMMETRIC ENCRYPTION ALGORITHM ON ELLIPTIC CURVES

Kalynovych M. S.[1], Golovko G. V.[2]

[1, 2] National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine
ORCID: [1] 0009-0002-1038-2711, [2] 0000-0002-1745-1321
E-mail: [1] kalinovichmaxym@gmail.com, [2] genvgolovko@ukr.net

DOI: 10.15673/atbp.v%vi%i.2484

Abstract: *The use of information technologies has become an integral part of the modern world, and the internet has played a vital role in facilitating access to various services, from ordering food to obtaining government services. However, with the increased reliance on the internet, the issue of information protection has become more pressing than ever. In the wake of the COVID-19 pandemic, remote work and online learning have become the norm, which has further emphasized the need for robust information protection mechanisms. One of the primary ways of ensuring information security is through the use of cryptographic algorithms. These algorithms have been in use for several decades and have provided an effective means of protecting sensitive information. However, with the exponential growth of computing power, the demands on these algorithms have increased, and some of them have become outdated and are no longer secure enough. As a result, there is a pressing need for the development of new cryptographic algorithms that use more complex mathematical principles to provide a higher level of security. One such class of algorithms that has gained popularity in recent years is elliptic curve cryptography. These algorithms use the principles of elliptic curves to provide greater security while using relatively fewer system resources. Elliptic curve cryptography has several advantages over other cryptographic algorithms. Firstly, it provides a higher level of security with shorter key lengths, making it ideal for use in devices with limited processing power and memory. Additionally, it offers greater resistance to attacks by quantum computers, which are expected to become more prevalent in the future. Moreover, elliptic curve cryptography has found applications in several areas, including secure communication protocols, digital signatures, and key exchange algorithms. These algorithms are part of the TLS protocol, which is used by the HTTPS protocol to ensure secure transmission of information over the Internet. The continuous development and integration of cryptographic algorithms into various information systems will be crucial for protecting against cyber threats in the future.*

Анотація: *Використання інформаційних технологій стало невід'ємною частиною сучасного світу. Інтернет відіграє важливу роль у полегшенні доступу до різних послуг, від замовлення їжі до отримання державних послуг. Однак із зростанням використання Інтернету, питання захисту інформації стало актуальнішим, ніж будь-коли. Після пандемії COVID-19, віддалена робота та онлайн-навчання стали нормою, що ще більше підкреслило потребу в надійних механізмах захисту інформації. Одним із основних способів забезпечення інформаційної безпеки є використання криптографічних алгоритмів. Ці алгоритми використовуються протягом кількох десятиліть і є ефективним засобом захисту конфіденційної інформації. Однак із експоненціальним зростанням обчислювальної потужності, вимоги до цих алгоритмів зросли, а деякі з них застаріли та стали недостатньо безпечними. Як наслідок, існує гостра потреба в розробці нових криптографічних алгоритмів, які використовують складніші математичні принципи для забезпечення вищого рівня безпеки. Одним із таких класів алгоритмів, які набули популярності в останні роки, є криптографія на основі еліптичних кривих. Ці алгоритми використовують принципи еліптичних кривих для забезпечення більшої безпеки при використанні відносно меншої кількості системних ресурсів. Криптографія на основі еліптичних кривих має кілька переваг перед іншими криптографічними алгоритмами. По-перше, ці алгоритми забезпечують вищий рівень безпеки з меншою довжиною ключа, що дозволяє використовувати їх в пристроях з обмеженою обчислювальною потужністю та пам'яттю. Крім того, вони мають більший опір атакам квантових комп'ютерів, які, як очікується, стануть більш поширеними в майбутньому. Крім того, криптографія на основі еліптичних кривих знайшла застосування в кількох областях, включаючи протоколи*

безпечного зв'язку, цифрові підписи та алгоритми обміну ключами. Ці алгоритми є частиною протоколу TLS, який використовується протоколом HTTPS для забезпечення безпечної передачі інформації через Інтернет. Постійний розвиток та інтеграція криптографічних алгоритмів у різні інформаційні системи буде мати вирішальне значення для захисту від кіберзагроз у майбутньому.
Keywords: Cryptography, Public-key cryptography, ECC, RSA, ASP.NET, Cybersecurity, Encryption.
Ключові слова: криптографія, криптографія з відкритим ключем, ECC, RSA, ASP.NET, кібербезпека, шифрування.

Introduction

Information is the result of human interaction with the environment. It is both the cause of actions and their consequences. And therefore, the possession of information and its processing determines and changes our world and people's lives. Nowadays, when information has become more accessible than ever, it is especially important to keep certain information secret. It is easy to assume that no company will like it if its corporate documents become public. This threatens multimillion-dollar damages and loss of reputation, or even worse when it comes to personal information of employees or secret developments.

A truly secure computer is one that hasn't connected to any network, even an electrical one, but it is impossible to use such a computer. The modern world is inextricably linked with information technologies. Large masses of information are constantly being transferred over the Internet, and it is not very difficult to intercept it therefore, there is a need to transmit information in a protected form so that even if was intercepted, an attacker cannot use it. Also, when transmitting information in open networks, such as the Internet, it is important to confirm the identity of the sender and recipient. Cryptography deals with these and other problems. There are hundreds of different encryption algorithms that use various mathematical principles to protect information.

Data protection is a set of measures not limited to cryptographic protection. Technical, engineering, and organizational protection are also distinguished. Technical protection includes hardware and technical measures to restrict access to the information carrier, such as routers, firewalls, antiviruses, etc. Engineering protection exists to prevent the physical destruction of the information carrier. Organizational protection limits access to information by third parties, for example, by using access control rules. [1]

Literature analysis

1. Classification of encryption algorithms

Modern encryption algorithms can be divided into two classes: symmetric and asymmetric. Symmetric encryption algorithms use one key to encrypt and decrypt information, so this key must be private and known only to those who exchange information. On the opposite, asymmetric algorithms (also public-key algorithms) use two keys: a public key for encrypting data and a private key for decryption. The feature of these algorithms is that generating a public key based on a private key is a simple task, but restoring the private key by knowing the public key is very difficult.

Each of these classes has its advantages and disadvantages. Yes, symmetric algorithms are much faster and more reliable because they use much simpler mathematical principles, but the main difficulty is the exchange of the private key. The most popular algorithms of this type are RC4, AES, DES, 3DES, and QUAD.

Asymmetric algorithms exchange data according to the following scheme. The User A generates his private key p_A and, based on it, calculates the public key P_A, similarly, the user B generates his own pair of keys p_B and P_B. Users exchange public keys over an insecure channel, after which the user A can encrypt information with the public key P_B, and only the user B can decrypt his messages with the private key p_B. Similarly, only the user A can decrypt a message encrypted with the public key P_A. Representatives of this class are the following algorithms: RSA, Diffie-Hellman, and ECC.

In practice, hybrid algorithms are often used, which take into account the shortcomings of both classes of algorithms. They are not a separate class of encryption algorithms but a combination of two classes. Their main point is to use an asymmetric algorithm to exchange a private key, which will be used to encrypt information using a symmetric algorithm [2].

2. Comparison of RSA and ECC

With the growth of computing capabilities, the need for more reliable algorithms grows. Also, over time, vulnerabilities are found in existing algorithms. So, DES and 3DES algorithms no longer meet modern security requirements and are hardly used. A partial solution to this problem is to increase the key size. For example, the standard key size for the asynchronous RSA algorithm is now 2048 bits. But there is the more reliable and relatively new algorithms class ECC, which stands for Elliptic Curve Cryptography, which is being used more and more often. In particular, this algorithm is used by various cryptocurrencies, including Bitcoin.

Due to its greater mathematical complexity, the ECC algorithm provides the same level of security with a smaller key size (Table 1) [3]. The relationship between the security level and key size of ECC, compared to RSA, is not linear, so ECC scales much better. Also, thanks to the significantly shorter key, ECC is suitable for use in systems where execution time and the amount of used memory are critical.

Table 1 – Comparison of the security level with different key sizes

Security bits	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Also, there is work that shows how much water can be boiled with the energy needed to hack an algorithm with a certain key length (Table 2) [4].

Table 2 – Intuitive security levels

Security level	Volume of water to bring to a boil	Bit-lengths		
		Symmetric key	Cryptographic hash	RSA modulus
Teaspoon security	0.0025 liter	35	70	242
Shower security	80 liters	50	100	453
Pool security	2 500 000 liters	65	130	745
Rain security	0.082km3	80	160	1130
Lake security	89 km3	90	180	1440
Sea security	3 750 000 km3	105	210	1990
Global security	1 400 000 000km3	114	228	2380
Solar security	-	140	280	3730

3. Summary of the section

In modern cryptography, two classes of encryption algorithms are distinguished: asymmetric and symmetric. Each class has its advantages and disadvantages, but the best result can be obtained from a combination of these algorithms. Over time, the need for more secure algorithms grows, but a simple increase in the key size leads to an increase in the load on the system. So, it is appropriate to use more complex algorithms that provide the same level of security with a much smaller key size. That is why the ECC algorithm has been gaining more and more popularity lately.

Object, subject, and methods of research

Research object: cryptography.

Subject of research: cryptography on elliptic curves.

Research method: implementation of the ECC algorithm for the organization of a secure data transmission channel over the Internet.

In general, elliptic curves are described by the following equation:

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

Where $4a^3 + 27b^2 \neq 0$ – this condition allows us to exclude special curves that cross the point $O(0;\ 0)$ and are not full-fledged elliptic curves.

An important property of elliptic curves is that they are symmetrical about the x-axis (Fig. 1).

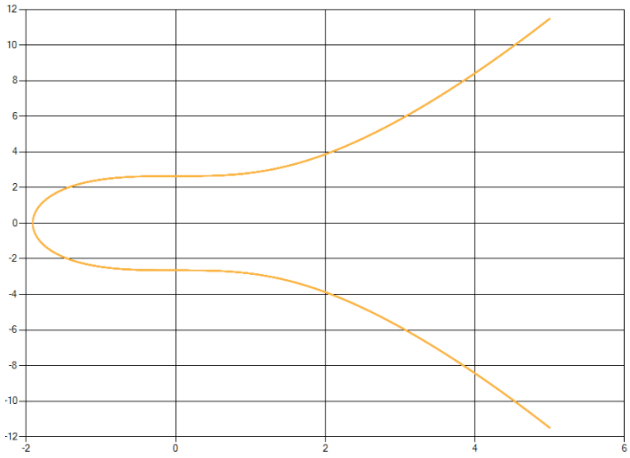

Fig. 1 – Elliptic curve when $a = 0, b = 7$

To further perform mathematical operations on the elliptic curve, it is necessary to define a group.

A group is a non-empty set G for which a binary operation has been defined. The group for which the addition operation "+" has been defined is called additive. The group must satisfy the following conditions:

1. Associativity: $a + (b + c) = (a + b) + c$
2. There exists a zero element O such that $a + O = O + a = a$
3. For any $a \in G$ there exists an opposite element b such that $a + b = O$. Which can be written as $a + (-a) = O$

If the commutativity condition $a + b = b + a$ is fulfilled, then such a group is abelian [5].

Let us define group G for the set of points of the elliptic curve. We define the binary operation "+" according to the following rule: the sum of three arbitrary, aligned non-zero elements P, Q, and R of the set G is equal to O:

$$P + Q + R = O$$

Where O – point at infinity or ideal point.

The element opposite to P is the element symmetrical about the x-axis $(-P)$.

Since the binary operation is defined for any elements, satisfying the condition, it can be proved that:

$$P + (Q + R) = (P + Q) + R = Q + R + P = O$$

Therefore, this binary operation has the property of associativity and commutativity. An abelian group is obtained.

Considering the properties of the group, the sum of two points can be rewritten in the following form:

$$P + Q = -R$$

It should be noted that any straight line crossing the elliptic curve intersects it in one, two, or three points (Fig. 2).

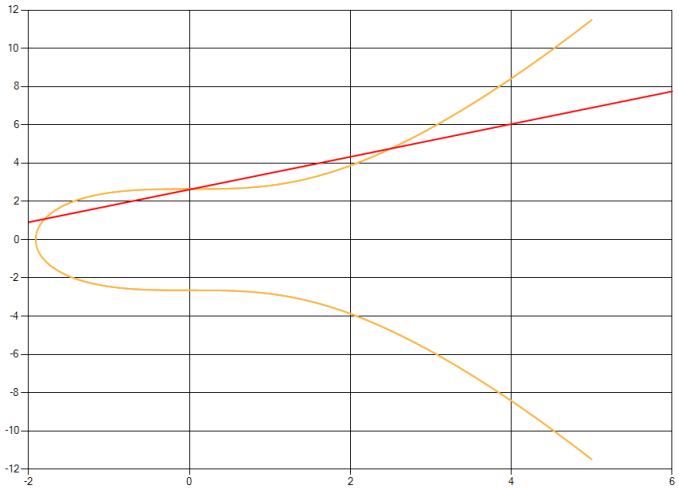

Fig. 2 – Intersection of the curve at 3 points

Vertical and tangent lines cross the curve at two points (Fig. 3-4).

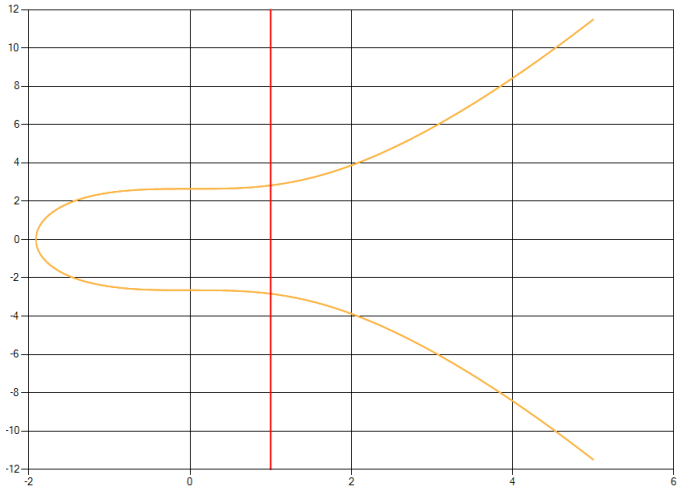

Fig. 3 – Intersection at 2 points

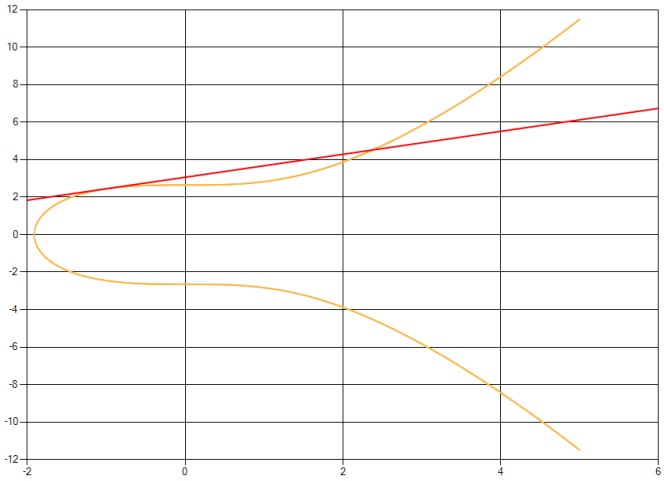

Fig. 4 – The tangent line intersects the curve at 2 points

So, we have an algorithm for finding the geometric sum of two nonzero points P, Q (Fig. 5).

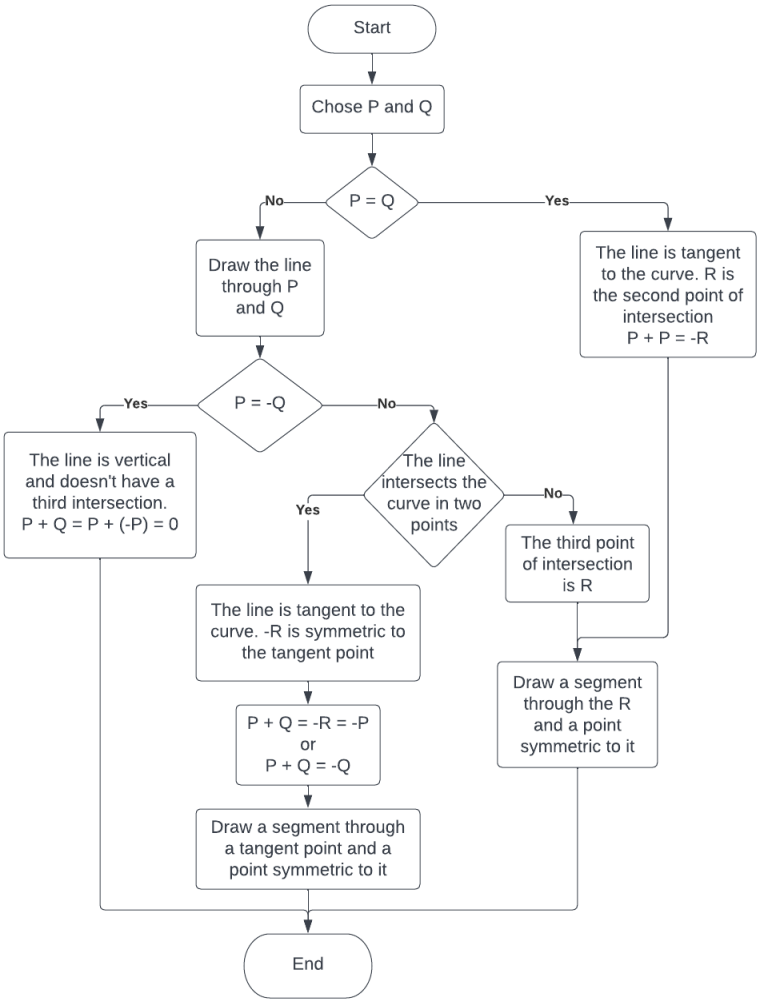

Fig. 5 – Algorithm for finding the geometric sum of two points

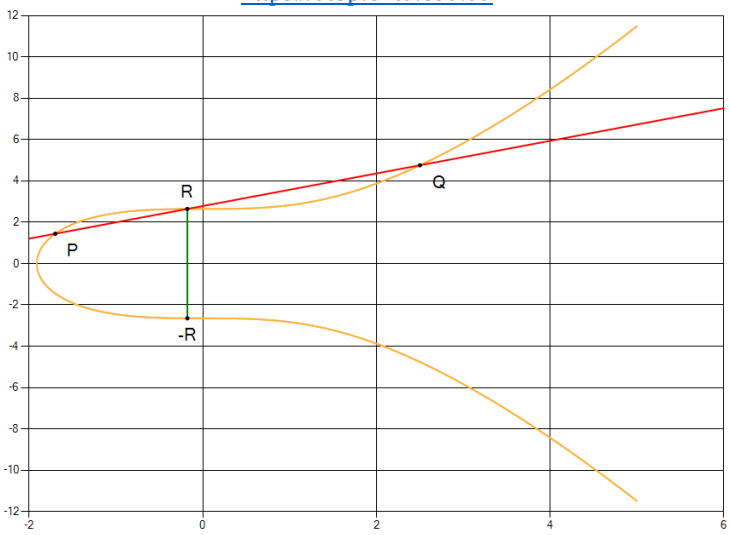

Fig. 6 – Geometric sum of points

Now it is possible to find the sum of two points on the graph (Fig. 6), but it is necessary to determine the algebraic sum for the numerical calculation of the coordinates of point $(-R)$.

If $P \neq Q$ or $x_P \neq x_Q$, then a slope λ of the line, crossing two points is:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$

This line intersects the curve at a point $R(x_R; y_R)$:

$$x_R = \lambda^2 - x_P - x_Q$$

$$y_R = \lambda(x_R - x_P) + y_P = \lambda\big(x_R - x_Q\big) + y_Q \quad (1)$$

Then,

$$-R(x_R; y_R) = R(x_R; -y_R)$$

So, the equation (1) can be rewritten as:

$$-y_R = \lambda(x_P - x_R) - y_P = \lambda\big(x_Q - x_R\big) - y_Q$$

For the case $P = Q$, the slope λ is found as the first derivative of the curve:

$$\lambda = y' = \sqrt{x^3 + ax + b}' = \frac{3x^2 + a}{2\sqrt{x^3 + ax + b}}$$

Or,

$$\lambda = y'_P = \frac{3x_P^2 + a}{2y_P}$$

An algorithm for the algebraic sum of two non-zero points of the curve has been obtained. [6]

For further calculations, it is necessary to define the scalar multiplication operation:

$$nP = \sum_{i=1}^{n} P$$

But with large n, the calculations take a lot of time – the time complexity of the algorithm is $O(n)$. One of the algorithms for solving this problem is the double and add algorithm. It is best to describe this algorithm using an example. Let $n = 170$, which in binary equals to $n = 10101010$ or:

$$170 = 1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 2^7 + 2^5 + 2^3 + 2^1$$

Then nP can be written as follows:

$$nP = 170P = 2^7P + 2^5P + 2^3P + 2^1P$$

So, there are the following steps:

1) double P to get 2^1P;
2) double the result two more times to get 2^3P and add it to 2^1P;
3) double 2^3P twice to get 2^5P and add to the previous sum;
4) similarly get 2^7P and add it to the previous sum.

As a result, it is necessary to perform 7 doubling operations and 3 addition operations. Such an algorithm is much faster and has a complexity of $O(\log_2 n)$.

The peculiarity of the specified multiplication operation is that by knowing n and P, it is easy to find $nP = P'$, but to find n by P and P', it is necessary to solve a complex problem of discrete logarithms. No algorithm with polynomial time complexity has been found to solve this problem on a classical computer.

Before that, we dealt with curves defined on the set of real numbers R, but for the further implementation of the cryptographic algorithm, it is necessary to define curves on finite fields with integer points. A finite field consists of a finite set of objects called field elements together with the description of two binary operations — addition and multiplication.

A finite field containing q field elements only if q is a power of a prime number, and furthermore that for each such q there is precisely one finite field. The finite field containing q elements is denoted by F_q.

In cryptography, it is usually distinguished the prime finite field F_p containing p elements, and the characteristic 2 finite field F_{2^m} containing 2^m elements, where $m \geq 1$. In this work considered the prime finite field. F_p is a set of integers:

$$F_p = \{0, 1, \dots, p-1\}$$

Binary operations are defined as follows:

- Addition: If $a, b \in F_p$, then $a + b = r$ in F_p, where $r \in [0, p-1]$ is the remainder when the integer $a + b$ is divided by p. This is known as addition modulo p and written $a + b \equiv r(mod\ p)$.
- Multiplication: If $a, b \in F_p$, then $ab = s$ in F_p, where $s \in [0, p-1]$ is the remainder when the integer ab is divided by p. This is known as multiplication modulo p and written $ab \equiv s(mod\ p)$.

There is no exactly operation for subtraction and division, but we can describe additive and multiplicative inverses:

- Additive inverse: If $a \in F_p$, then the additive inverse (−a) of a in F_p is the unique solution to the equation $a + x \equiv 0(mod\ p)$.
- Multiplicative inverse: If $a \in F_p$, $a \neq 0$, then the multiplicative inverse a^{-1} of a in F_p is the unique solution to the equation $ax \equiv 1(mod\ p)$. Which can be easily calculated by extended Euclidean algorithm.

Therefore, subtraction and division operations accordingly can be written as:

$$a - b \equiv a + (-b)\ (mod\ p)$$

$$\frac{a}{b} \equiv a(b^{-1})\ (mod\ p)$$

The equation of the curve over prime finite field F_p takes the following form:

$$y^2 \equiv x^3 + ax + b\ (mod\ p)$$

Where $4a^3 + 27b^2 \not\equiv 0$.

Graphically, the curve $y^2 \equiv x^3 + 7\ (mod\ 127)$ looks as follows (Fig. 7). The symmetry is preserved about the line $y = \frac{p}{2}$.

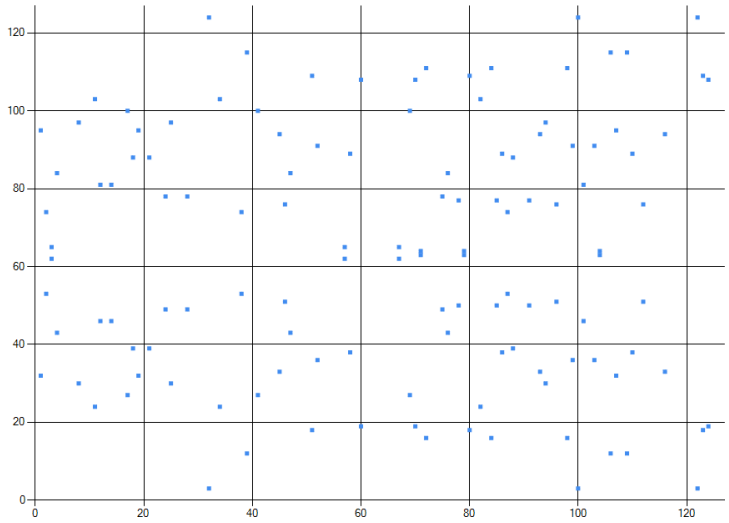

Fig. 7 – Elliptic curve on the finite field F_{127}

We are still able to find geometric sum of points on the graph (Fig. 8).

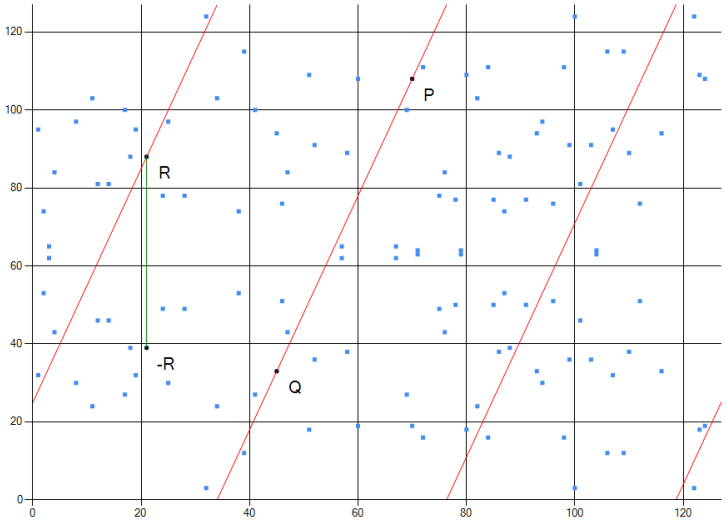

Fig. 8 – Sum of points P, Q of curve

The method is similar to the one described above with the difference that the line drawn through points P and Q "repeats" itself on the plane.

Summarizing the above, we can make changes to the equations of point R:

$$x_R \equiv \lambda^2 - x_P - x_Q \ (mod\ p)$$

$$-y_R \equiv \lambda(x_P - x_R) - y_P \ (mod\ p) \equiv \lambda\big(x_Q - x_R\big) - y_Q \ (mod\ p)$$

Where:

$$\lambda \equiv \begin{cases} (y_Q - y_P)\big(x_Q - x_P\big)^{-1}(mod\ p), & if\ x_P \neq x_Q \\ (3x_P^2 + a)(2y_P)^{-1}(mod\ p), & if\ x_P = x_Q \end{cases}$$

Elliptic curve domain parameters over F_p precisely specify an elliptic curve and base point. This is necessary to precisely define public-key cryptographic schemes based on ECC.

Elliptic curve domain parameters over F_p are a sextuple:

$$T = (p, a, b, G, n, h)$$

Where p – prime integer, specifying the finite field F_p.

a, b – parameters of a curve.

G – a base point.

n – order of G.
h – cofactor of the subgroup.
This information is sufficient for rough implementation of algorithms on elliptic curves [7].

Results

The selection of curve parameters is a very important stage. There are a number of vulnerable curves that cannot be used in cryptographic problems. But, elliptic curves with recommended parameters were defined and standardized.

The curve of the secp256k1 standard [8] with the following parameters was used in this work:

$$T = (p, a, b, G, n, h)$$

Where:

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

$$a = 0$$

$$b = 7$$

$$G = 04\ 79BE667E\ F9DCBBAC\ 55A06295\ CE870B07\ 029BFCDB\ 2DCE28D9\ 59F2815B\ 16F81798\ 483ADA77\ 26A3C465\ 5DA4FBFC\ 0E1108A8\ FD17B448\ A6855419\ 9C47D08F\ FB10D4B8$$

$$n = FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ FFFFFFFE\ BAAEDCE6\ AF48A03B\ BFD25E8C\ D0364141$$

$$h = 1$$

Knowing the parameters of the elliptic curve, we can implement any algorithm on elliptic curves. Many classical cryptographic algorithms have analogs on elliptic curves. The most popular of them are the Elliptic Curve Digital Signature Algorithm (ECDSA) – an algorithm that allows authenticating the identity of the sender, and the Elliptic-curve Diffie–Hellman (ECDH) – which is a complete analog of the private key exchange algorithm.

But in this work, we are interested in the asynchronous encryption algorithm, which is written as follows:

1. User A chooses a random private key $d_A \in [1, n-1]$.
2. Based on the private key d_A, the user generates the public key $P_A = d_A G$ and transfers it to user B.
3. User B similarly finds his key pair d_B, P_B and gives user A his public key P_B.
4. User A chooses the plaintext m that he wants to encrypt and finds the point of the curve $P_m(m; y_m)$.
5. User A chooses a random number $k \in [1, n-1]$ and uses the public key P_B to encrypt the message, then send the result to user B as a pair of points:

$$G_m = \{kG,\ P_m + kP_B\}$$

6. User B decrypts the message using his private key d_B:

$$(P_m + kP_B) - d_B kG = P_m + k(d_B G) - d_B kG = P_m$$

Since the message is encoded by a point on the curve, it can be represented as an arbitrary number $m \in [1, n-1]$. In this work, the bit length of the key is 256 bits, so it is possible to encrypt a message up to 32 bytes long at one time.

The ECC algorithm was implemented by using the programming language C#. There was created a console server and a website in case to demonstrate the algorithm at work. The website was created by using the ASP.NET framework [9]. Also, the Microsoft SQL Express database was created, which contains accounts of registered users and their access levels. User passwords are stored in a hashed form in the database to protect them in case of data leakage.

This software can be used to build a secure system in the company. All computers can be connected to a local network with a separate computer on which the site is hosted, so there is no need for an expensive server room. The server can be located anywhere on the Internet. Since computers are connected to a local network and connected to the Internet through a router, the structure of the network is not visible from the outside. Connection to the server is only possible through the website, and only registered and authorized users have access to the information.

For demonstration purposes, a five-level system of access mediation was created. But this can be changed in the database with administrator rights. An example of partial filling of the database is shown in Fig. 9.

AccountId	Login	Password
1	admin	8c6976e5b5410415bde908bd4dee15...
2	ceo	618402424220c92bcb61bd07148525...
3	chiefaccountant	1c200f7054544e609ca2ad667201abe...
4	engineer 1	488acff6c8256dbe92a4860b8251c99...

Fig. 9 – Example of filling the database

Registration of new users is performed by the administrator. Registered users can log in using their login and password. Authorized users have access to data depending on their access level and can view any available files, download them from the server, and upload new files to the server.

The authorization form is shown in Fig. 10.

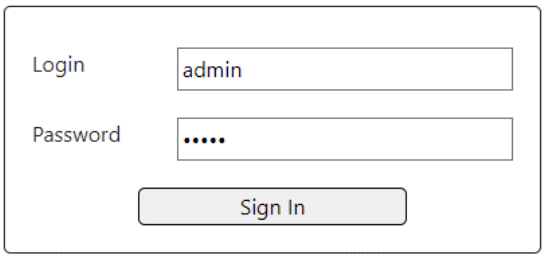

Fig. 10 – Authorization form

The list of all available files is displayed to the authorized user, depending on their access level (Fig. 11-12).

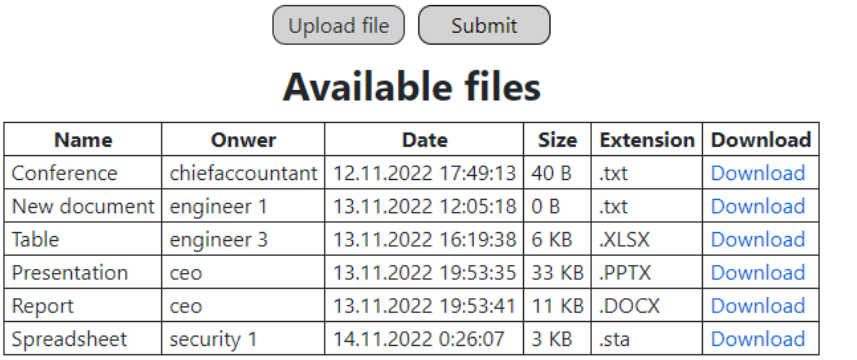

Fig. 11 – List of files available to a user with access level 5

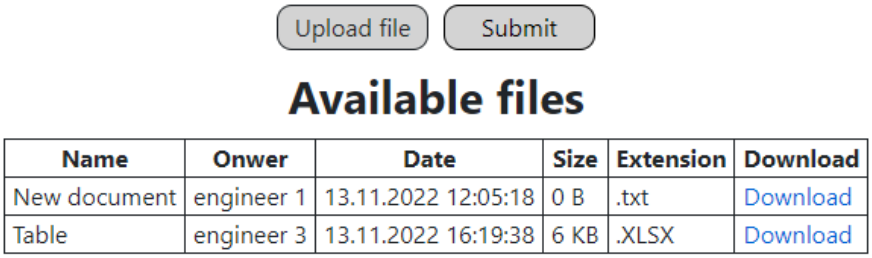

Fig. 12 – List of files available to a user with access level 2

The server records all user actions (Fig. 13).

```
127.0.0.1:12128: connected
127.0.0.1:12128: user admin logged in
127.0.0.1:12128: files listed
127.0.0.1:12128: user engineer 1 logged in
127.0.0.1:12128: files listed
```

Fig. 13 – Log of user actions

The algorithm for working with the server is as follows:

1. When the website is initialized, the connection to the server and the generation of encryption keys are performed.
2. The server and the website exchange public keys over an insecure channel. After this stage, data exchange goes in a protected form.
3. The user enters the authorization data, which are sent in encrypted form.
4. The server decodes the message and compares the user data with the data in the database. If the user is found, the server determines their access level and sends an authorization confirmation to the user. Otherwise, if the entered data is not valid, the authorization request is rejected, return to point 3.
5. The user sends a request to display the list of available files.

6. The server returns to the user information about all files with an access level equal to or lower than the user's access level.

7. The user selects the file and confirms its sending. The file is split into 32-byte fragments and sent to the server as an encrypted stream.

8. The server receives information about the file, creates it in its own file system, and writes the decoded stream of bytes to this file.

9. The user sends a request to download a file from the server.

10. The server sends the file to the user in the form of an encrypted stream.

11. The user receives the file stream, decrypts it, and saves it to a file on his own system.

The implemented asynchronous encryption algorithm works stably for any data of arbitrary size but is expectedly very slow. And such factors as the execution of a request to the database, data transfer via the Internet, and insufficiently optimized implementation of the algorithm further reduce its speed. In this example, with an Internet connection speed of ~90 Mbit/s, the speed of file transfer by this algorithm was ~2 KB/s.

Conclusions

In this work were considered the main principles of building a secure system. Different types of cryptographic algorithms were described. Compared two asymmetric algorithms – RSA and ECC. The basics of cryptography on elliptic curves have been described, and the encryption algorithm on elliptic curves is implemented. A demonstration software has been created, which includes cryptographic and organizational measures for building a secure system. Experiments show that such an algorithm is very slow (~2 KB/s), which makes it inapplicable when transferring large files. This can be partially resolved by analyzing the code and identifying weaknesses in the implementation of the algorithm. But considering that the algorithm is complex and requires a lot of calculations, it still will be slow. That is why it is more efficient to use a hybrid encryption algorithm. For example, make an exchange of a private key using ECDH, and then use this key to encrypt data with any modern symmetric algorithm, for example, AES.

References

[1] V. O. Khoroshko, M. V. Kapustian. Zakhyst informatsii. Instytut entsyklopedychnykh doslidzhen NAN Ukrainy, 2010 https://esu.com.ua/article-15872

[2] Bellare, Mihir; Rogaway, Phillip. Introduction to Modern Cryptography. University of California at Davis, 2005.

[3] Maletsky K. RSA vs. ECC Comparison for Embedded Systems, 2020. https://ww1.microchip.com/downloads/en/DeviceDoc/00003442A.pdf

[4] Arjen K. Lenstra, Thorsten Kleinjung, and Emmanuel Thome. Universal security from bits and mips to pools, lakes – and beyond. Cryptology ePrint Archive, Paper 2013/635, 2013. https://eprint.iacr.org/2013/635

[5] Joseph J. Rotman. An Introduction to the Theory of Groups. Fourth Edition. Springer-Verlag New York, Inc., 1995.

[6] Atkin, A. O. L. and Morain, F. Elliptic Curves and Primality Proving. American Mathematical Society, 1993.

[7] Daniel R. L. Brown. SEC 1: Elliptic Curve Cryptography, 2009. https://www.secg.org/sec1-v2.pdf

[8] Daniel R. L. Brown. SEC 2: Recommended Elliptic Curve Domain Parameters, 2010. https://www.secg.org/sec2-v2.pdf

[9] Daniel R., Rick A., and Shaun L. Overview of ASP.NET Core, 2022. https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0