End-to-end Encrypted Native Software Using the Sieve of Eratosthenes and Binet Formula MA_T CHAT / MA_T SOHBET

Mehmet Akif Karakaş and Taylan Yalçın, İzmir Fen Lisesi, yalcin.taylan02@gmail.com

ARTICLE INFO

Gold Medalists in IMSEF 2021 , Izmir Turkey and Silver Medalists in ISAC Olympiad 2021Tehran, Iran
Accepted by Ariaian Young Innovative Minds Institute,AYIMI

http://www.ayimi.org , info@ayimi.org

ABSTRACT

If the need-to-know principle is seen as the basic principle of opposing intelligence, the question "Is it possible for scientists to correspond in a secure and location-independent manner on subjects with a high degree of confidentiality?" may arise. Our MA_T Chat software, which was created with the Python Programming Language, is proof that it is possible to securely communicate between two points without data loss, with a system consisting of two separate encryption, two separate hashing algorithms and shelling over the TCP/IP Reference Model.

Key words: *Encryption, Hash, Eratosthenes, Binet, Algorithm*

1 Introduction

The principle of need-to-know, defined in the three paragraphs of Article 4 of the Defense Industry Security Regulation, published in the Official Gazette No. 27601 and dated 04.06.2010, states that we must encrypt our highly confidential information while moving it from one point to another.

In the conversation between a scientist at T point and a scientist at MA point, if information with a high confidentiality level is shared, end-to-end encryption may be preferred here.

Although Ermoshina, Musiani and Halpin created an overview of end-to-end encryption protocols in 2016, in 2017 Musiani and Ermoshina asked “What is a good secure messaging tool?” In their article on the question “If your keys are stolen, will past communications be safe?” With the question of the theft of keys used in encryption, it has opened up discussion. AIM reviewed BlackBerry Messenger, BlackBerry Protected, Ebuddy XMS, Facebook Chat, and ChatSecure+Orbot. Only in ChatSecure+Orbot did the theft of the key not compromise the security of past communications. In other words, we can say that only end-to-end encryption is not enough.

In 2016, Yılmaz and Ballı compared the confidentiality, integrity, authentication, non-repudiation, performance and security features of Symmetric and Asymmetric Encryption Algorithms in their work on Developing an Intelligent Selection System for the Use of Data Encryption Algorithms. They determined that the security feature depends on the key length. They also revealed that symmetric algorithms have faster performance than asymmetric algorithms.

In 2018, Chakraborty, Jana, Mandal, and Kule showed that neural network-based RSA provides better results than Standard RSA scheme in performance analysis. A Comparative Study of Different Techniques for Basic Testing in RSA Implementation by Banerjee, Mandal, and Das, published in 2020, and Comparative Studies of Rathod, Sreenivas, Chandavarkar, from Beginning to Present, Between RSA Algorithm and Variants in RSA Application, published in 2020, still continue to face the dilemma of security and performance in asymmetric encryption algorithms. reveals that.

To solve this current situation, it is necessary to work on encryption algorithms that can simultaneously provide the performance speed of symmetric encryption and the strong security of asymmetric encryption. It is also necessary to research for a correspondence software that uses these algorithms.

While doing this research, it should be taken into account that while some scientists are dealing with cryptography or steganography, some scientists are dealing with cryptanalysis or steganalysis. In their article Cryptography and Cryptanalysis: A Review published in 2013, Tiwari, Nandi and Mishra used encryption (cryptography) and encryption analysis (cryptanalysis) methods, and in their article published in 2014 named Review of various steganalysis techniques, Kaur and Kaur used text hiding (steganography) and hidden text. They talked about finding (steganalysis) techniques.

Safitri, Ali and Ibrahim used the graph regarding the pre-cryptanalysis in their article entitled Cryptology Techniques and Methodologies (Fig. 1).

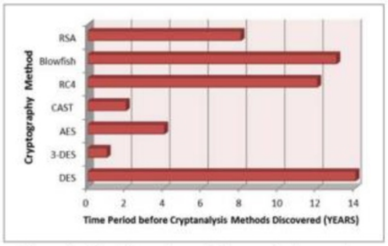


Fig.1: Visual Overview of Cryptanalysis over time

Carroll and Martin (1986) The automated cryptanalysis of substitution ciphers. With their work called Cryptologia, they emphasized that the password can be cracked according to the usage frequencies of the characters.

There are three types of cryptanalysis that tries to decrypt only ciphertext, known ciphertext and raw text pairs, and selected raw text or selected ciphertext.

Considering all these, it may be worth examining the question of what would be the result of developing a system that can be resistant to cryptanalysis, instead of creating and using an encryption algorithm, and using encryption and hashing algorithms together.

Starting from this point, we started our research on T encryption and T hashing algorithm and shelling of MA encryption and MA hashing algorithms.

2 Main Aim

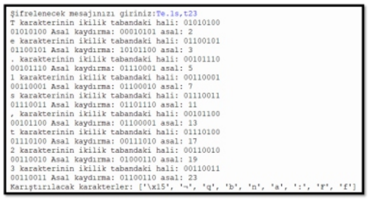
With the current version of our MA_T Chat software created with Python Programming Language, it is aimed to prove that it is possible to securely communicate between two points without data loss, with a system consisting of two separate encryption, two separate hashing algorithms and shelling that we created over TCP/IP Reference Model. Our second aim is to produce prime numbers by using the Sieve of Eratosthenes and to examine the effect of using these numbers in character-level encryption and shifting algorithms on the complexity of the operation.

Our final goal is to analyze a chat software created with the Caesar encryption algorithm in terms of frequency analysis with the MA_T Chat software we created.

3 Method

3-1 T Encryption Method

The T encryption algorithm encrypts the raw data as shown in the screenshot (Fig. 2).

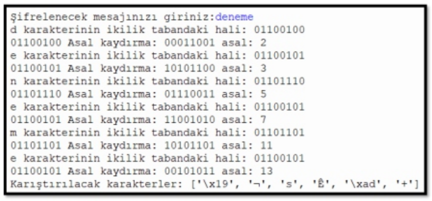


```
Şifrelenecek mesajınızı giriniz:Te.1s,t23
T karakterinin ikilik tabandaki hali: 01010100
01010100 Asal kaydırma: 00010101 asal: 2
e karakterinin ikilik tabandaki hali: 01100101
01100101 Asal kaydırma: 10101100 asal: 3
. karakterinin ikilik tabandaki hali: 00101110
00101110 Asal kaydırma: 01110001 asal: 5
1 karakterinin ikilik tabandaki hali: 00110001
00110001 Asal kaydırma: 01100010 asal: 7
s karakterinin ikilik tabandaki hali: 01110011
01110011 Asal kaydırma: 01101110 asal: 11
, karakterinin ikilik tabandaki hali: 00101100
00101100 Asal kaydırma: 01100001 asal: 13
t karakterinin ikilik tabandaki hali: 01110100
01110100 Asal kaydırma: 00111010 asal: 17
2 karakterinin ikilik tabandaki hali: 00110010
00110010 Asal kaydırma: 01000110 asal: 19
3 karakterinin ikilik tabandaki hali: 00110011
00110011 Asal kaydırma: 01100110 asal: 23
Karıştırılacak karakterler: ['\x15', '¬', 'q', 'b', 'n', 'a', ':', 'F', 'f']
```

Fig.2: Screenshot 01 – Application of T Encryption Method

A prime number equal to the number of characters of the data to be encrypted is generated. This prime number sequence starts from 2 and grows without skipping prime numbers, forming a set with as many elements as the number of characters in the data. Each character in the data is matched with the prime number sequence on the condition that their indices (orders) remain the same. For each character, first the character is converted to binary. It is shifted bitwise to the right by the prime number to which it is mapped. The resulting new binary number is converted to a character in the ASCII table. These characters are stored in a list. Thus, our data is encrypted.

While the standard algorithm used when creating prime numbers has O(n2) complexity, this complexity is reduced to O(nloglogn) by using the Sieve of Eratosthenes in the T encryption algorithm. This increases the performance speed. Since each character will match a different prime number according to its location, its image will be a different character. Thus, it will not be possible to easily match in frequency analysis during cryptanalysis. In addition, the space characters that separate the words from each other may remain as spaces in the encrypted data after the data is processed, while some of them may turn into a different character. Thus, word detection will not be possible by looking at the spaces (Fig.3).

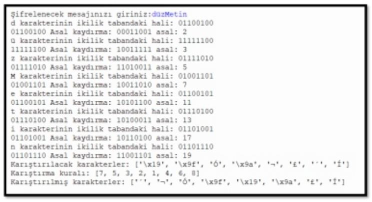


```
Şifrelenecek mesajınızı giriniz:deneme
d karakterinin ikilik tabandaki hali: 01100100
01100100 Asal kaydırma: 00011001 asal: 2
e karakterinin ikilik tabandaki hali: 01100101
01100101 Asal kaydırma: 10101100 asal: 3
n karakterinin ikilik tabandaki hali: 01101110
01101110 Asal kaydırma: 01110011 asal: 5
e karakterinin ikilik tabandaki hali: 01100101
01100101 Asal kaydırma: 11001010 asal: 7
m karakterinin ikilik tabandaki hali: 01101101
01101101 Asal kaydırma: 10101101 asal: 11
e karakterinin ikilik tabandaki hali: 01100101
01100101 Asal kaydırma: 00101011 asal: 13
Karıştırılacak karakterler: ['\x19', '¬', 's', 'Ê', '\xad', '+']
```

Fig. 3: Screenshot 02 – The letters "e" in the word "deneme" are paired with different characters according to their place in the word. ['¬','Ê','+']

3-2 T Hashing Method

How the raw data encrypted with the T encryption algorithm is mixed with the T hashing algorithm is shown in Figure (4).



```
Şifrelenecek mesajınızı giriniz:düzMetin
d karakterinin ikilik tabandaki hali: 01100100
01100100 Asal kaydırma: 00011001 asal: 2
ü karakterinin ikilik tabandaki hali: 11111100
11111100 Asal kaydırma: 10011111 asal: 3
z karakterinin ikilik tabandaki hali: 01111010
01111010 Asal kaydırma: 11010011 asal: 5
M karakterinin ikilik tabandaki hali: 01001101
01001101 Asal kaydırma: 10011010 asal: 7
e karakterinin ikilik tabandaki hali: 01100101
01100101 Asal kaydırma: 10101100 asal: 11
t karakterinin ikilik tabandaki hali: 01110100
01110100 Asal kaydırma: 10100011 asal: 13
i karakterinin ikilik tabandaki hali: 01101001
01101001 Asal kaydırma: 10110100 asal: 17
n karakterinin ikilik tabandaki hali: 01101110
01101110 Asal kaydırma: 11001101 asal: 19
Karıştırılacak karakterler: ['\x19', '\x9f', 'Ó', '\x9a', '¬', '£', '´', 'Î']
Karıştırma kuralı: [7, 5, 3, 2, 1, 4, 6, 8]
Karıştırılmış karakterler: ['´', '¬', 'Ó', '\x9f', '\x19', '\x9a', '£', 'Î']
```

Fig. 4: Screenshot 03 – T Encryption and then T Hash algorithm is applied to the word "düzMetin".

The T Hash Algorithm first creates a set of counting numbers from 1 to the number of characters of the encrypted data. In this set, if any, prime numbers are found first and they are ordered from largest to smallest. The remaining elements of the set are ordered from smallest to largest, and these two lists are combined. The character order of the encrypted text is rearranged according to the resulting list. Thus, our data is encrypted.

Since the presence of prime numbers in the hashing algorithm uses the already created prime number list for the T Encryption Algorithm, the complexity of the T Hash Algorithm is O(n).

Changing the number of prime numbers to be used in hashing according to the character length of the ciphertext and changing the spaces in the ciphertext will make the decryption more difficult compared to the unscrambled cipher during cryptanalysis.

3-3 MA Encryption Method

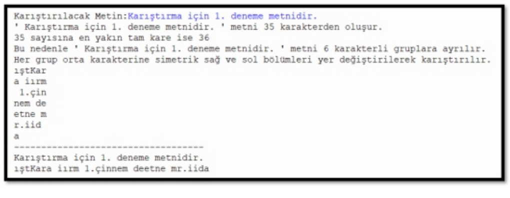
It is similar to the T Encryption algorithm. The only difference is that the Fibonacci sequence is used instead of the prime number sequence for shifting. If the calculation of the Fibonacci sequence is done iteratively, it is O(n) and if it is done recursively, it is O(2n) complexity. When we use the matrix form, the Fibonacci sequence is calculated with O(logn) complexity. We can find O(1) complexity by memorizing with dynamic programming. n. For the Fibonacci number, Binet (18th century French mathematician Jacques Binet) produces a number of O(1) complexity (Eq. 1).

$$\mathrm{Fib}(n) = \frac{\mathrm{Phi}^n - (-\mathrm{Phi})^{-n}}{\sqrt{5}} = \frac{\mathrm{Phi}^n - (-\mathrm{phi})^n}{\sqrt{5}} \tag{1}$$

The Fibonacci number calculation algorithm used in the MA Encryption Method is a dynamic programming and memorization algorithm.

3-4 MA Hashing Method

To shuffle it is necessary to create a new sequence. The integer number closest to the total number of characters is taken and the characters are grouped according to their square root. Afterwards, each group is mixed in itself so that the right and left sides are swapped. The data is mixed according to the new order created (Fig. 5).



```
Karıştırılacak Metin:Karıştırma için 1. deneme metnidir.
' Karıştırma için 1. deneme metnidir. ' metni 35 karakterden oluşur.
35 sayısına en yakın tam kare ise 36
Bu nedenle ' Karıştırma için 1. deneme metnidir. ' metni 6 karakterli gruplara ayrılır.
Her grup orta karakterine simetrik sağ ve sol bölümleri yer değiştirilerek karıştırılır.
ıştKar
a iırm
 1.çin
nem de
etne m
r.iid
a
-----------------------------------
Karıştırma için 1. deneme metnidir.
ıştKara iırm 1.çinnem deetne mr.iida
```

Fig. 5: Screenshot 04 – Example Application of MA Hashing Algorithm

3-5 Shelling of Encryption and Hashing Methods

For correspondence between point T and point MA, the original message is created at point T. The original message is encrypted with the T encryption algorithm using the sequence of prime numbers and mixed with the T hashing algorithm, taking its first form and sent to the MA point. The T hashing algorithm is applied in reverse, the message is encrypted with the MA encryption using the Fibonacci number sequence, mixed with the MA hashing algorithm, and sent back to the T point, taking its second form. At T point, first the MA hashing algorithm and then the T encryption application is applied in reverse, and the message is sent to the MA point with T scrambling in its third form. At the MA point, first the T hashing algorithm and then the reverse MA encryption algorithm are applied and the original message is reached without data loss.

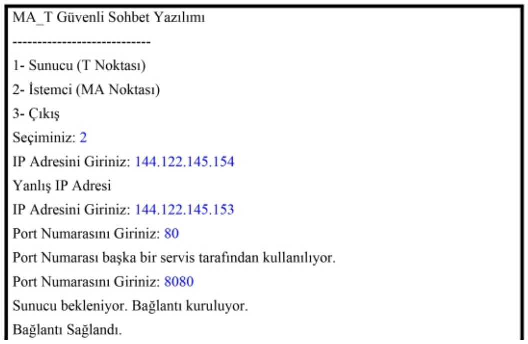
Table 1: MA_T Shelling Algorithm 01

Order of process	T Point	State of Data	MA Point
1	The user writes the message to be sent.	Raw Data	-
2	T Password is done.	T Encrypted Data	-
3	T Hashing is done.	T Encrypted and T Scrambled Data (FIRST STATE)	-
4	Data is sent from T Point.	FIRST STATE	Data is taken from T Point.
5	-	FIRST STATE	T Hashing is applied in reverse.
6	-	T Encrypted and MA Encrypted Data	MA Encryption is done.
7	-	T Encrypted, MA Encrypted and MA Hashed Data (SECOND STATE)	MA Hashing is done.
8	The data is taken from the MA Point.	SECOND STATE	Data is sent to T Point.
10	MA Hashing is applied in reverse.	T Encrypted and MA Encrypted Data	-
11	T Encryption is applied in reverse	MA Encrypted Data	-
12	T Hashing is done.	MA Encrypted and T Hashed Data (THIRD STATE)	-
13	Data is sent from T Point.	THIRD STATE	Data is taken from T Point.
14	-	THIRD STATE	T Hashing is applied in reverse.
15	-	MA Encrypted Data	MA Encryption is decrypted.
16	-	Raw Data	The user sees the message which is sent

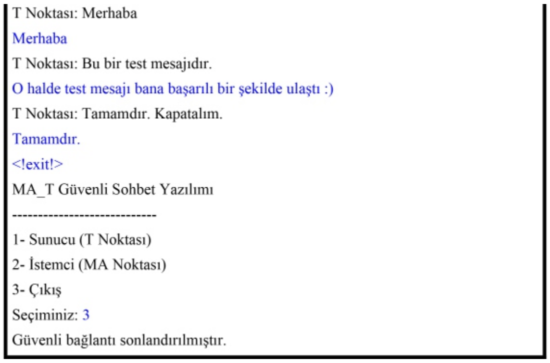
The table shows that assuming the first author has input T, the raw message has been sent and will display the MA news; but this is not required. The first message can also be sent from the MA. Such a preference is assumed to be MA and T is shelled.

4 MA_T Chat Softwareı

The program prepared using the Python programming language establishes the connection between the T point and the MA point with the TCP/IP Reference Model on the Internet using Microsoft Windows Socket (WinSock). After the connection is established, a secure correspondence is made with the MA_T shelling system. When the correspondence is completed, the program is closed without any recording (Fig.6).

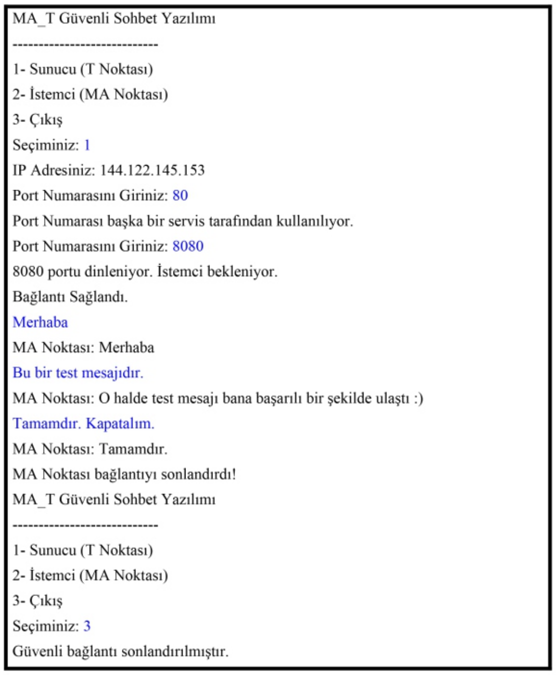


```
MA_T Güvenli Sohbet Yazılımı
----------------------------
1- Sunucu (T Noktası)
2- İstemci (MA Noktası)
3- Çıkış
Seçiminiz: 2
IP Adresini Giriniz: 144.122.145.154
Yanlış IP Adresi
IP Adresini Giriniz: 144.122.145.153
Port Numarasını Giriniz: 80
Port Numarası başka bir servis tarafından kullanılıyor.
Port Numarasını Giriniz: 8080
Sunucu bekleniyor. Bağlantı kuruluyor.
Bağlantı Sağlandı.
```



```
T Noktası: Merhaba
Merhaba
T Noktası: Bu bir test mesajıdır.
O halde test mesajı bana başarılı bir şekilde ulaştı :)
T Noktası: Tamamdır. Kapatalım.
Tamamdır.
<!exit!>
MA_T Güvenli Sohbet Yazılımı
----------------------------
1- Sunucu (T Noktası)
2- İstemci (MA Noktası)
3- Çıkış
Seçiminiz: 3
Güvenli bağlantı sonlandırılmıştır.
```

Fig. 6: Screenshot 05 – Example of running MA_T Chat Software as a client



```
MA_T Güvenli Sohbet Yazılımı
----------------------------
1- Sunucu (T Noktası)
2- İstemci (MA Noktası)
3- Çıkış
Seçiminiz: 1
IP Adresiniz: 144.122.145.153
Port Numarasını Giriniz: 80
Port Numarası başka bir servis tarafından kullanılıyor.
Port Numarasını Giriniz: 8080
8080 portu dinleniyor. İstemci bekleniyor.
Bağlantı Sağlandı.
Merhaba
MA Noktası: Merhaba
Bu bir test mesajıdır.
MA Noktası: O halde test mesajı bana başarılı bir şekilde ulaştı :)
Tamamdır. Kapatalım.
MA Noktası: Tamamdır.
MA Noktası bağlantıyı sonlandırdı!
MA_T Güvenli Sohbet Yazılımı
----------------------------
1- Sunucu (T Noktası)
2- İstemci (MA Noktası)
3- Çıkış
Seçiminiz: 3
Güvenli bağlantı sonlandırılmıştır.
```

Fig. 7: Screenshot 06 – Example of running MA_T Chat Software as a server

MA_T Chat Software can communicate between T point and MA point with known IP and Port number. With this type of connection, the software neither records the connection nor the entries. If there is no active listening on the network (sniffing) or the keyboard listening system (keylogger) software is not running on the computer, the encrypted messages of the MA_T Chat software cannot be accessed.

Table 2: MA_T Chat Software Client Tests

Tests to be made	Variable	Explanation
Connection Test	IP Adress	Entering Wrong IP Address (For example, merhaba.122.153.tr)
Connection Test	IP Adress	Entering an unused IP Address
Connection Test	IP Adress	Entering the IP Address of the server that does not accept external connections
Connection Test	IP Adress	Entering the IP Address of the server accepting external connection
Connection Test	Port	Entering a port that is not open on the server
Connection Test	Port	Entering a port where a different service/service is running on the server (conflict)
Connection Test	Port	Entering an open and unused port on the server
Connection Test	Port	Entering the port opened for the MA_T Chat software on the server
Connection Test	Network Card	Entering the IP Address of the server that accepts external connection when the network card is disabled
Connection Test	Network Card	Entering the IP Address of the server that accepts external connections when the network card is active

Connection Test	MODEM	Entering the IP Address of the server that accepts external connection when the MODEM is turned off and the network card is active
Connection Test	MODEM	Entering the IP Address of the server that accepts external connection when the MODEM is on and the network card is active
Connection Test	Phone Line	Entering the IP Address of the server that accepts external connection when the telephone line is disconnected but the MODEM and Network card are active
Connection Test	Phone Line	Entering the IP Address of the server that accepts external connection when the telephone line is open but MODEM and Network card are active
Connection Test	Disconnection	Disabling the client computer's network card during correspondence
Connection Test	Disconnection	Disabling the network card of the server computer during correspondence
Message Test	Message	Entering only uppercase letters as a message
Message Test	Message	Entering only lowercase text as a message
Message Test	Message	Entering a text consisting of numbers only as a message
Message Test	Message	Entering only special characters (such as space, period, comma, root sign) as a message
Message Test	Message	Entering mixed-character text (24-4) as a message
Message Test	System Message	Entering a message containing the <!exit!> phrase (at the beginning, middle, or end of the phrase)
Message Test	System Message	Entering the <!exit!> block

Table 3: MA_T Chat Software Server Tests

Tests to be made	Variable	Explanation
Connection Test	IP Adress	Can't get IP Address
Connection Test	Port	Entering a port that is not open on the server
Connection Test	Port	Entering a port where a different service/service is running on the server (conflict)
Connection Test	Port	Entering an open and unused port on the server
Connection Test	Handshake	Client not sending request from long listening port
Connection Test	Network Card	Server's IP Address request with network card disabled
Connection Test	MODEM	Server's IP Address request when MODEM is off and Network card is active
Connection Test	MODEM	Server's IP Address request when MODEM is on and Network card is active
Connection Test	Phone Line	Server's IP Address request with phone line down but MODEM and Network card active
Connection Test	Phone Line	Server's IP Address request when phone line is on but MODEM and Network card are active
Connection Test	Disconnection	Disabling the client computer's network card during correspondence
Connection Test	Discconection	Disabling the network card of the server computer during correspondence
Message Test	Message	Entering only uppercase letters as a message
Message Test	Message	Entering only lowercase text as a message
Message Test	Message	Entering a text consisting of numbers only as a message
Message Test	Message	Entering only special characters (such as space, period, comma, root sign) as a message
Message Test	Message	Entering mixed-character text (2^4-4) as a message
	Message	beginning, middle, or end of the phrase)
Message Test	System Message	Entering the <!exit!> block

Table 4: MA_T Chat Software Algorithm Tests

Yapılacak Testler	Değişken	Açıklama
Encryption Test	T Encryption	Checking whether messages containing characters that can be entered with the keyboard in the ASCII Table can be encrypted
Encryption Test	T Encryption	Checking the reversal of generated ciphertexts without data loss
Encryption Test	MA Encryption	Checking whether messages containing characters that can be entered with the keyboard in the ASCII Table can be encrypted
Encryption Test	MA Encryption	Checking the reversal of generated ciphertexts without data loss
Encryption Test	T ve MA Encryption	Checking the reversal of generated ciphertexts without data loss
Hashing Test	T Hashing	Checking whether messages containing keyboard-enterable characters in the ASCII Table can be shuffled
Hashing Test	T Hashing	Checking the rendering of generated hashed texts without data loss
Hashing Test	MA Hashing	Checking whether messages containing keyboard-enterable characters in the ASCII Table can be shuffled
Hashing Test	MA Hashing	Checking the rendering of generated hashed texts without data loss
Shelling Test	Whole Shelling System	Checking that messages containing characters that can be entered with the keyboard in the ASCII Table are shelled and decoded without data loss
Complexity Test	Big Data	Testing algorithm performance
Complexity Test	Prime Numbers	Comparison of the standard method and the Sieve of Eratosthenes method
Complexity Test	Fibonacci	Comparison of the standard method and the Binet method
Strenght Test	Cryptanalysis	Testing the strength of the algorithm

Table 5: Project Deed-Date Schedule

	MONTHS										
Deed	March	April	May	June	July	August	Septemb	October	Novemb	Decemb	January
Programming Tutorials	X	X	X	X							
Literature Review			X	X	X	X					
Programming of the Software						X	X	X	X		
Performing Software Tests, Recording of Data and Analysis									X	X	
Writing the Project Report											X

5 Results

Table 6: MA_T Chatting Software Client Tests Results (Part 1)

Test Performed	Variable	Result
Connection Test	IP Adress	When IP Addresses that are not between 0.0.0.0 – 255.255.255.255 IP Address block are entered, the software asks to enter the IP Address again. When the entered address is in this block range but an unused IP Address, the software first asked for the port, then tried to establish a connection 3 times, and when it could not establish the connection (pinged), it asked to enter the IP Address again. The same situation occurred in the case of entering a server IP Address that does not accept external connections. In case the IP Address of the server that accepts external connection is entered, port tests are started. If the IP address of any server that accepts external connection is entered, not the server to be connected, it has been determined that the software does not work properly. A different handshake protocol should be developed between the client and the server, other than TCP/IP Protocol.
Connection Test	Port	In case a port that is not open on the server is entered, the client software tried to establish a connection 3 times and when it could not establish the connection (pinged), it asked to enter the IP Address again. It has been determined that the software does result was seen when an open and unused port was entered on the server. A different handshake protocol should be developed between the client and the server, other than TCP/IP Protocol. If the port opened for MA_T Chat software is entered on the server, it has been determined that the software works properly. Network Card tests have been passed.
Connection Test	Web Card	While the network card was disabled, the client software tried to establish a connection 3 times, and when it could not establish the connection (pinged), it asked to enter the IP Address again. Pinging the 127.0.0.1 ip address of the software and getting a response should be checked and the error that there is a problem with the network card should be given to the user. The software has been found to work properly with the network card enabled. MODEM tests were started.

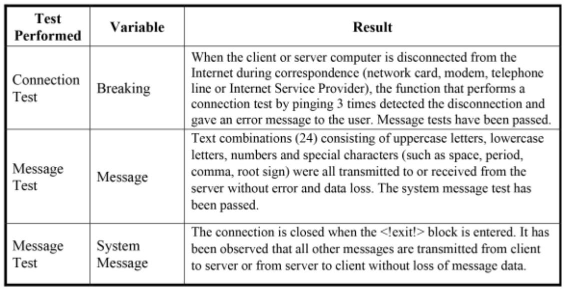
Table 7: MA_T Chatting Software Client Tests Results (Part 2)

Test Perforned	Variable	Result
Connection Test	MODEM	While MODEM was turned off, the client software tried to establish a connection 3 times and when it could not establish the connection (pinged), it asked to enter the IP Address again. It has been determined that the software is working properly when MODEM is on. Telephone line tests have been started.
Connection Test	Phone Line	While the phone line was down, the client software tried to establish a connection 3 times, and when it could not establish the connection (pinged), it asked to enter the IP Address again. The software has been found to work properly when the phone line is open. The rupture tests have been passed.
Connection Test	Breaking	When the Internet connection of the client computer or the server computer is lost during correspondence (network card, modem, telephone line or Internet Service Provider), the function that performs a connection test by pinging 3 times a minute in the background detected the disconnection and gave an error message to the user. Message tests have been passed.
Message Test	Message	Text combinations (24) consisting of uppercase letters, lowercase letters, numbers and special characters (such as space, period, comma, root sign) were all transmitted to or received from the server without error and data loss. The system message test has been passed.
Message Test	System Message	The connection is closed when the <!exit!> block is entered. It has been observed that all other messages are transmitted from client to server or from server to client without loss of message data.

Table 8: MA_T Chatting Software Server Test Results (Part 1)

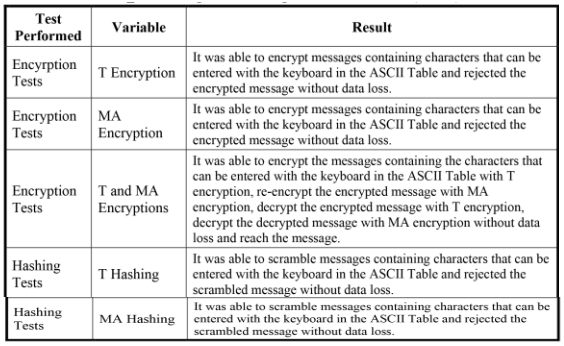
Test Performed	Variables	Result
Connection Test	IP Adress	Server software can obtain IP Address; but when it receives with IP addresses such as 192.168.XXX.XXX, 127.0.0.1 or 10.0.XXX.XXX, it may not be healthy to connect with the client. Since it passed the IP Address acquisition test, these healthy connection failures were examined in other tests. Port tested.
Connection Test	Port	If a port that is not open on the server is entered, the server software activated that port and listened. The same result was seen when an open and unused port was entered on the server. It has been determined that the software does not work properly in case of entering a port (conflict) where a different service/service is running on the server. Handshake tests have been passed.
Connection Test	Hand Shaking	As a result of the client not sending a request from the listening port (while there is a connection) for a long time (3 minutes), the connection was interrupted and an error message was given to the user. It's on to the next test.
Connection Test	Web Card, MODEM, Phone Line	The server software received an ip address when the network card was disabled. It is necessary to write a function that checks whether the received IP Address can be accessed by the client over the Internet. The same situation was found in the MODEM and Telephone Line tests.

Table 9:MA_T Chatting Software Server Test Results (Part 2)



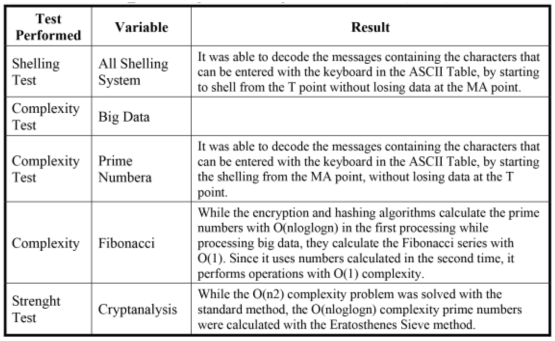
Test Performed	Variable	Result
Connection Test	Breaking	When the client or server computer is disconnected from the Internet during correspondence (network card, modem, telephone line or Internet Service Provider), the function that performs a connection test by pinging 3 times detected the disconnection and gave an error message to the user. Message tests have been passed.
Message Test	Message	Text combinations (24) consisting of uppercase letters, lowercase letters, numbers and special characters (such as space, period, comma, root sign) were all transmitted to or received from the server without error and data loss. The system message test has been passed.
Message Test	System Message	The connection is closed when the <!exit!> block is entered. It has been observed that all other messages are transmitted from client to server or from server to client without loss of message data.

Table 10: MA_T Chatting Software Algorşitm Test Results (Part 2)



Test Performed	Variable	Result
Encyrption Tests	T Encryption	It was able to encrypt messages containing characters that can be entered with the keyboard in the ASCII Table and rejected the encrypted message without data loss.
Encryption Tests	MA Encryption	It was able to encrypt messages containing characters that can be entered with the keyboard in the ASCII Table and rejected the encrypted message without data loss.
Encryption Tests	T and MA Encryptions	It was able to encrypt the messages containing the characters that can be entered with the keyboard in the ASCII Table with T encryption, re-encrypt the encrypted message with MA encryption, decrypt the encrypted message with T encryption, decrypt the decrypted message with MA encryption without data loss and reach the message.
Hashing Tests	T Hashing	It was able to scramble messages containing characters that can be entered with the keyboard in the ASCII Table and rejected the scrambled message without data loss.
Hashing Tests	MA Hashing	It was able to scramble messages containing characters that can be entered with the keyboard in the ASCII Table and rejected the scrambled message without data loss.

Table 11: MA_T Chatting Software Algoritm Test Results (Part 2)



Test Performed	Variable	Result
Shelling Test	All Shelling System	It was able to decode the messages containing the characters that can be entered with the keyboard in the ASCII Table, by starting to shell from the T point without losing data at the MA point.
Complexity Test	Big Data	
Complexity Test	Prime Numbera	It was able to decode the messages containing the characters that can be entered with the keyboard in the ASCII Table, by starting the shelling from the MA point, without losing data at the T point.
Complexity	Fibonacci	While the encryption and hashing algorithms calculate the prime numbers with O(nloglogn) in the first processing while processing big data, they calculate the Fibonacci series with O(1). Since it uses numbers calculated in the second time, it performs operations with O(1) complexity.
Strenght Test	Cryptanalysis	While the O(n2) complexity problem was solved with the standard method, the O(nloglogn) complexity prime numbers were calculated with the Eratosthenes Sieve method.

6 Discussions

It has been determined that our MA_T Chat software does not work properly. It is foreseen that these problems can be solved by developing a different handshake protocol between client and server computers other than TCP/IP Protocol.

It has been determined that our MA_T Chat software does not work properly in cases where the server computer receives an IP address that cannot be reached over the Internet in the connection to be made from the server to the client, or a service/service is selected for the connection (conflict). It is envisaged that these problems can be eliminated by adding a function that checks port conflict and a function that checks whether the IP address can be accessed over the Internet or not.

In that case, it has been proved in the current tests that our MA_T Chat software prepared with Python Programming Language, apart from the connection problems between the server and the client, can safely communicate between two points with the encryption and hashing algorithms and the shelling system over the TCP/IP Reference Model without any data loss. We can say it does.

In addition, the production of prime numbers by utilizing the Sieve of Eratosthenes used during encryption O(nloglogn) and the production of the terms of the Fibonacci series with the Binet formula is O(1), and the stored and reused numbers of the produced numbers are O(1) complexity.

It has been observed that encryption and scrolling at the character level affects the complexity of the operation as much as the number of characters. In that case, it can be said that the encrusted system is O(n) complexity.

If MA_T Chat software used only Caesar encryption, the key would be detected and the communication could be decrypted due to the matching of domain and image set in frequency analysis tests. As a result of using T and MA mixing and encryption algorithms with shelling in the current version of MA_T Chat software, it has been observed that the mapping of Domain and Image Set could not be detected in frequency analysis. The software then allowed secure correspondence.

7 Suggestions

Research on generating a handshake protocol such as password (Moon) – sign (star) between client and server can improve the server – client connection of MA_T Chat software. Doing research on checking port conflicts and checking whether the IP address is reachable over the Internet can improve the server-client connection of MA_T Chat software.

Apart from frequency analysis, the degree of security of MA_T Chat software can be examined with different password cracking (cryptanalysis) studies.

Apart from correspondence, research can be made on how to increase the functions of the MA_T Chat software by researching on the sharing of picture, video and audio files between T and MA points.

A new interface can be designed for the MA_T Chat software and compared with the existing interface, and the usability of these interfaces by various target groups can be investigated.

References

[1] Savunma Sanayii Güvenliği Yönetmeliği (2010). T.C. Resmi Gazete (27601, 4 Haziran 2010).

[2] Dooley, J. F. (2018). History of cryptography and cryptanalysis: Codes, Ciphers, and their algorithms. Springer.

[3] Musiani, F., & Ermoshina, K. (2017). What is a good secure messaging tool? The EFF secure messaging scorecard and the shaping of digital (usable) security. Westminster Papers in Communication and Culture, 12(3).

[4] Carroll, J. M., & Martin, S. (1986). The automated cryptanalysis of substitution ciphers. Cryptologia, 10(4), 193-209.

[5] Tiwari, G., Nandi, D., & Mishra, M. (2013). Cryptography and Cryptanalysis: A Review. International Journal of Engineering Research & Technology, 2(10), 1898-1902.

[6] Kaur, M., & Kaur, G. (2014). Review of various steganalysis techniques. International journal of computer science and information technologies, 5(2), 1744-1747.
Safitri, C., Ali, H. S., & Ibrahim, J. B. A Study: Cryptology Techniques and Methodologies.

[7] Agrawal, M., & Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. International Journal on Computer Science and Engineering, 4(5), 877.

[8] Ermoshina, K., Musiani, F., & Halpin, H. (2016, September). End-to-end encrypted messaging protocols: An overview. In International Conference on Internet Science (pp. 244-254). Springer, Cham.

[9] Yılmaz, M., Ballı, S. (2016). VERİ ŞİFRELEME ALGORİTMALARININ KULLANIMI İÇİN AKILLI BİR SEÇİM

SİSTEMİ GELİŞTİRİLMESİ. Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, 2 (2) , 18-28.

[10] Chakraborty, M., Jana, B., Mandal, T., & Kule, M. (2018, July). An Performance Analysis of RSA Scheme Using Artificial Neural Network. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[11] Rathod, U., Sreenivas, S., & Chandavarkar, B. R. (2021). Comparative Study Between RSA Algorithm and Its Variants: Inception to Date. In ICCCE 2020 (pp. 139-149). Springer, Singapore.

[12] Banerjee, K., Mandal, S. N., & Das, S. K. (2020). A Comparative Study of Different Techniques for Prime Testing in Implementation of RSA. American Journal of Advanced Computing, 1(1), 1-7.