WhatsUP

1st Santiago Valera Barreiros
Criptografía y Seguridad en Redes
Universidad Panamericana
CCDMX, México
0228654

***Abstract*—People are social beings who have a constant need to exchange information with other people and nowadays with technology people exchange messages with their smart devices, this information is not always protected and sometimes it becomes exposed so it is necessary to add protection so that the information cannot be read, modified or intercepted, for this there are cryptographic methods that help us cover this needs. To meet these needs, we will create a communications system with protective measures to preserve the authenticity, veracity and non-repudiation of the information contained in a message.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

With an increasingly connected world and with more technology, certain applications are used that help the communication of legal and physical persons, these applications carry information that are issued by a certain sender and received by its corresponding receiver, this information can be sensitive and of great value so it should not be seen by other people.

In this way, with messaging applications people have been able to establish contact with old friends, maintain conversations with loved ones who are far away and even start a new business by having a conversation with their customers with a messaging application.

The use of these technologies has become so common for people that sometimes they tend to send sensitive information such as photos of debit or credit cards, classified information, important documents, official credentials among other things in such a way that it has attracted the attention of ill-intentioned people who seek somehow to intercept these messages to get hold of the information or modify them according to their convenience, which has caused havoc in society causing scams, identity theft, money theft, sensitive information leakage, etc.

These problems occur mainly because communication networks are not the most secure, so certain protocols have been implemented to solve these problems, some examples of these would be:

- SSL which is the acronym for Secure Sockets Layer which is a standard technology to have a secure connection to the Internet [1].
- TSL is an acronym for Transport Layer Security, which is an improved and updated version of the previous protocol [2].
- HTTPS which stands for Hyper Text Protocol Security, this appears in the url address when a website is protected by an SSL certificate, besides this is the secure version of HTTP [3].

Through some channels there are specialized software called packet sniffers. A sniffer is a software tool that allows the user to monitor their Internet traffic in real time and capture all data traffic in and out of their computer. What ill-intentioned people called hackers do is that they use these programs to capture and inspect the packets that travel through the network, so if one of these hackers is on your network he will be able to see all the traffic and capture the datagrams that pass through and if one of those datagrams is a message that contains information in plain text the hacker will be able to read it without difficulty.

A. PROBLEM TO BE SOLVED

Although these types of protocols exist, they have not been sufficient to prevent third parties from stealing or stealing information from conversations, so to protect conversations and prevent third parties from obtaining information by intercepting datagrams, there are cryptographic methods, where cryptography is an ancient technique that seeks to encrypt text or information so that only the sender and receiver can decrypt it. It is based on complex mathematical algorithms that are responsible for encrypting messages.

There are two types of encryption: symmetric and asymmetric:

Symmetric encryption: uses the same key to encrypt and decrypt the message [4].

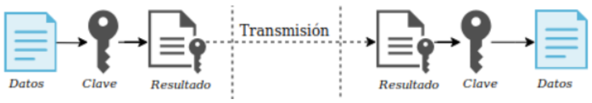


Fig. 1. Symmetric encryption

In the figure below we can see in an illustrated way how a symmetric encryption is performed using only one key.

Asymmetric encryption: uses 2 keys, the public key which can be distributed to people who need it and the private key which must never be disclosed [4].

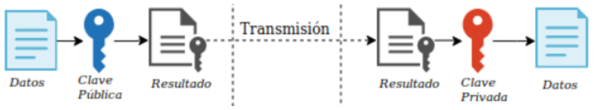


Fig. 2. Symmetric encryption

Figure 2 illustrates how asymmetric encryption is performed using two different keys.

Each of these types of encryption has its advantages and disadvantages such as symmetric encryption is much faster than asymmetric encryption, symmetric encryption is not as secure as symmetric encryption, it all depends on the needs that are present, which will determine which will be the best method to encrypt.

Already implementing these protocols and the encryption is intended to have secure documents which have three characteristics: integrity, authenticity and non-repudiation.

Integrity is to secure messages against unwanted modifications. Authenticity is to ensure that the communicating entities are who they claim to be.

Non-repudiation is the protection against denial of origin of the messages and that the sender should not be able to deny the authorship of the document. There are also the hash functions that take a message and transform it into characters that do not make any sense, in theory each message should derive in a different combination of characters and each time we put that message through the hash function should give us the same combination.

II. Definition of the Project Problem

A. *Definition of the problem to be solved*

The problem is that if some sniffer intercepts a datagram containing a message and this message is in plain text, it will be able to read it without problems.

B. *Hypothesis*

Simple substitution encryption is not sufficient to protect information. Electronic signature is a technique that guarantees the integrity, authenticity and non-repudiation of a message.

III. SOLUTION

Due to what has been shown above there is clearly a need for an application that uses cryptographic methods to protect the information, so it has been proposed to make a communication system that is secure and meets the three characteristics already mentioned, thus the idea of WhatsUp an application that allows users to choose between different types of encryption to protect their information and that third parties can not read it arose.

A. *OBJECTIVES*

To create a secure communication from sender to receiver. To offer different types of encryption methods. To make the information sent secure.

B. *SCOPE*

At the moment the project will have 2 applications, one where the client will be and the other where there will be a server.

The first stage will have limited cryptographic methods, which would be a simple substitution method called cesar encryption, which will be encrypted symmetrically and asymmetrically and finally there will be a mechanism that is capable of generating an encrypted summary to verify the authenticity of a message.

IV. METHODS

In the following, we will explain how simple substitution encryption also known as cesar encryption works and how the electronic signature works.

A. *Simple substitution*

In this type of encryption a set of characters is needed to be modified, the issue is that a set of chars is given and depending on the key each char will be changed by another letter depending on the original position of the char within the set of characters in this way as for example we have a set consisting of all the letters of the English alphabet, the word to encrypt is hello and the key 1 is assigned as a result will give ipmb, which would be the encrypted message and assuming that it was symmetrically encrypted if we decrypt it with the key 1 in result would be hello again [5].

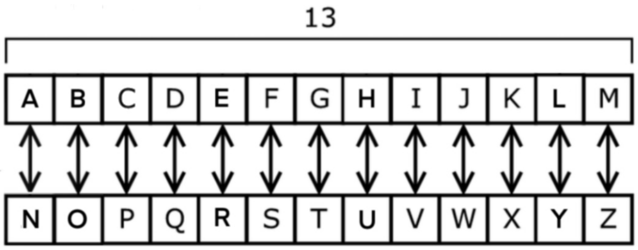


Fig. 3. Cesar Encryption

Fig. 3 shows an alphabet that is used to encrypt with the cesar method with a key of 13 where if your message has a letter ”a” it will be changed by an ”n” according to the key that tells us how many positions we should move or for example if your message has a ”z” that letter will be changed by an ”m”.

B. *Electronic signature*

This method offers authenticity and non-repudiation consists of once written a message which we will call original, this will be entered into a hash function to obtain a summary once obtained the summary this will be entered into an encryptor and given a key will be obtained an encrypted summary also known as digital signature, and the union of the original message plus the signature is known as a signed document, this set will be sent over the network and reach the receiver, this one will identify that the signed document conserves its two characteristics seeing that it has not been modified to make this identification the original message must be entered to the same function has that was used to obtain the first summary, to this component we will denominate it part a, later the encrypted summary that came next to the original message must be entered to a decryptor, the same that was used to encrypt, to this result we will denominate it b, to finalize a and b must be equal in case it could not make been that they modified the original message or that the key or the method to decrypt are not the correct ones. [6]

C. Digital envelope

A digital envelope is a secure electronic data container used to protect a message by encrypting and authenticating data. A digital envelope allows users to encrypt data with the speed of secret key encryption and the convenience and security of public key encryption. A digital envelope uses two layers for encryption: secret key and public key encryption. Secret key encryption is used to encode and decode messages. Public key encryption is used to send a secret key to a receiving party over a network. This technique does not require plain text communication. [7]

D. Digital Seal Certificate

A Digital Seal Certificate (CSD) is a digital file issued by the SAT in its capacity as Certification Authority, which contains the data of a taxpayer, individual or legal entity and is linked to a public key. [8]

In terms of Electronic Invoice, a CSD is composed of three parts:

1) A security certificate file (.CER)
2) A private key file (.KEY)
3) And the password of the latter.

And its functions are the following:

Generate the seal of the CFDi, security element that prevents the falsification or manipulation of data of the same one. To include information of the Certificate, and therefore of the issuer, inside the receipt.

E. Public Key Infrastructure (PKI)

A public key infrastructure (PKI) is a system of resources, policies and services that supports the use of public key encryption to authenticate parties to a transaction.

There is no single standard that defines the components of a PKI, but a PKI typically consists of Certificate Authorities (CAs) and Registration Authorities (RAs). CAs provide the following services:

- Issuance of digital certificates.
- Validation of digital certificates.
- Revocation of digital certificates.
- Distribution of public keys.

The X.509 standard provides the basis for the industry standard Public Key Infrastructure. See Digital Certificates for more information on digital certificates and Certificate Authorities (CAs). RAs verify the information provided when digital certificates are requested. If the RA verifies this information, the CA can issue a digital certificate to the requestor.

A PKI can also provide the tools to manage digital certificates and public keys. A PKI is sometimes described as a trusted hierarchy for managing digital certificates, although most definitions include additional services. Some definitions include encryption and digital signature services, but these services are not essential to the operation of a PKI.. [9]

The term public key infrastructure is derived from public key cryptography, the technology on which PKI is based. Public key cryptography has unique characteristics that make it indispensable as the basis for security functions in distributed systems.

In a PKI environment, two different cryptographic keys are used for encryption and decryption purposes, known as public and private keys. The private key is kept secret by the user or in the system, and the public key can be shared and distributed. The keys are mathematically related and cannot be deduced from each other. Data encrypted with one key can only be decrypted with the other complementary key and vice versa. PKI encompasses a complex set of technologies; the main components are listed below:

- Policy and practices: define the requirements and standards for the issuance and management of keys and certificates and the obligations of all PKI entities, and is used to determine the level of trust provided by the certificate.
- Certification Authority (CA): in a PKI environment, a digital certificate would be required, which usually contains the individual's public key, information about the certificate authority and additional information about the certificate holder. The certificate is created and signed (digital signature) by a trusted third party, the certificate authority (CA). The identity of the individual is linked to the public key, where the CA assumes responsibility for the authenticity of that public key, to enable a secure communication environment. The main duties of the CA are to ensure a highly secure environment for cryptographic operations, as well as to issue and manage certificates.
- Registration Authority (RA): provides the interface between the user and the CA. It verifies the user's identity and transmits valid requests to the CA. The Registration Authority (RA) is the place where it verifies that the person or organization requesting the certificate is who they claim to be.
- Certificate distribution system: an optional component of PKI, but one that can be critical in private environments, is the certificate distribution system, which publishes certificates in the form of an electronic directory, database or by email so that users can find them. This is usually done through a directory service. A directory server may already exist within an organization or be provided as part of the PKI solution.

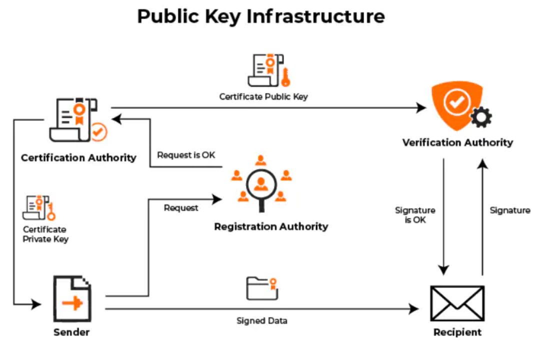


Fig. 4. Cesar Encryption

Fig. 4 shows an example of a Public Key Infrastructure and the information flow in each part .

V. CODE

The WhatsApp UP project consists of 3 programs, the following is a general breakdown of the components of both components:

- WhatsUP clients: it has three parts, the first is the login screen, the second is the contact selection screen and the third is the chat screen.
- Server: it is a program that only has a single screen.
- AR: it is a program that only has a single screen.

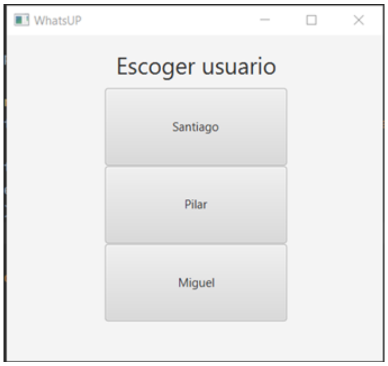


Fig. 5. Login screen

Fig 5 shows the first screen when running the program, where there are 3 buttons to choose which user you are, depending on the user you select the contacts will be different.

Once you have selected the user you are going to chat with, you cannot change it once selected, to select another user it is necessary to close the program and run it again to select another user.

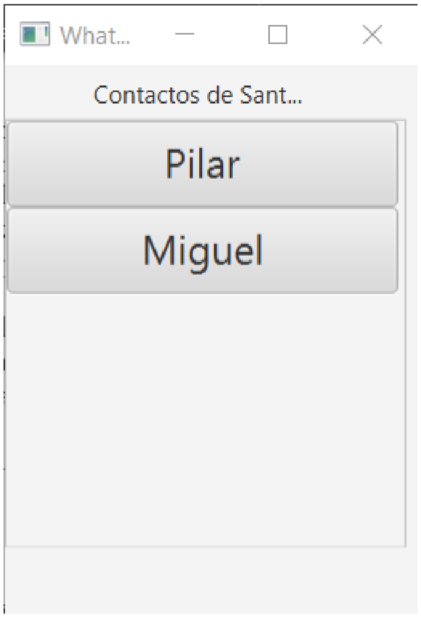
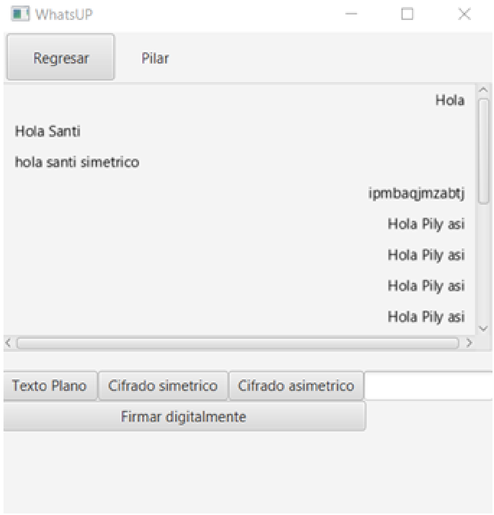


Fig. 6. Contacts screen

In fig 6 you can see the contacts of the selected user, each one of them is a button and when pressed it sends you to the chat screen, it is important to note that you cannot go back to the previous screen.



Hola

Hola Santi

hola santi simetrico

ipmbaqjmzabtj

Hola Pily asi

Hola Pily asi

Hola Pily asi

Hola Pily asi

Fig. 7. Chat screen

In fig 7 you will find the chat and the different buttons that help us to use the cryptographic methods available so far to protect the messages sent, the messages received by the contact will arrive encrypted if they have been sent in this way and with the same buttons in the images you will get the decrypted message, with the back button you can return to the previous screen.

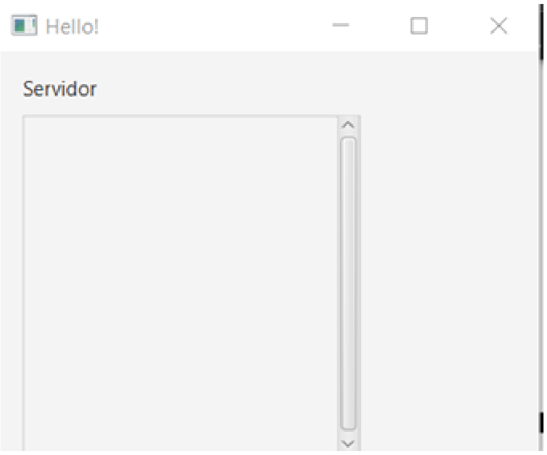


Fig. 8. Chat screen

Fig. 8 shows a simple interface that belongs to the screen of the server program, which is independent from the client one, where the messages that have arrived to the server will be shown, as well as who is the sender port and who is the receiver port.

A. Login screen

In the signature you select which user is the one to be started so to speak, there will be three buttons where you can choose the user, once the user is chosen the sending ports will be assigned automatically.

Each button is a function called ”void escogerPersona(void)” where ’Person is a name that changes as the button appears with a certain name. Inside this function another function is invoked which sends by parameter a number that is related to the type of person that is chosen, the function that is invoked is ”void cambiarChat(int opcion)”.

”void cambiarChat(int opcion)” this function switches us to the chat selection screen and invokes a method of the second screen that allows us to assign the receiver ports and the sender according to the selected option, this way when listening to a chat the information will arrive to a determined port that corresponds with some name. The invoked method is ”void asignarPuertos(int opcion)” that corresponds to the class of the screen 2 ”HelloController”.

B. Contacts screen

In this screen are shown the contacts of the person that has been selected, each contact is a button which can be pressed to enter another screen called ”chat” to which you send a message to that person.

The first method to be used is ”public void asignarPuertos(int opcion)” method that is activated for the first time in the previous screen.

Public void asignarPuertos(int opcion)” what is done in this method is that an object of the class LogicaNegocio called user is instantiated, which has a parameter in its constructor and it is the selected user option, here it appears up to the top in the chat whose are the contacts and later the names of the people are assigned in the buttons obtaining from the user instance the receiver 1 and the receiver 2.

LogicaNegocio Class: This class is used to instantiate an object that will tell us who is the sender port and which are the receiver ports, where there is a sequence of else-if to determine which is the selected option and thus give the corresponding information.

There are 2 buttons with the name of each contact which can be seen in the image 1.2, inside these buttons an object of the Package class is called data giving the constructor a predefined message, who is the sender port and who is the receiver port that is linked to the name, this object ”data” will provide us with the information in the chat screen of which port the message is directed to and who is the sender port. Then we call the method ”void cambiarChat(String persona,Package datos)” to which we pass the name of the person we have chosen to talk to so that this name is shown up to the top of the next screen and we send the previously created target that contains the message address information, when clicking on this button we will change the screen and two methods of the ChatController class will be invoked. methods that are invoked within cambiarChat()

void cambiarNombre(String persona)” : which allows us to assign the name of the chat receiver in the next screen.

”void establecerUsuario(Package data)”. : the package address information is sent.

”void establecerUsuario(LogicaNegocio user)”: this method is used to save the option we selected in the previous screen, this variable is passed to the other screen so as not to lose the information and once we come back from the other screen we remember which was the user that was selected previously

”void recuperarPuertos(LoicaNegocio usuario)”: this method is invoked while in the chat screen when you want to return to the contacts screen to remember who is the person who has logged in and show their contacts.

C. Chat screen

The Chat screen is conformed by the ChatContoller class that has several global variables that would be the different elements of java fx, a global variable called ”data” that is of type ”Package” which will give us the information of which port we have to send the information to and the global variable user of type ”LogicaNegocio” this variable is used so that when we go back to the chat selection screen we remember who is the sender and who are the receivers.

Methods

”void establecerPackage(Package received)”: this method saves in the global variable the data of the sender and receiver port. It also loads the application data by calling the ”cargarMensajes()” method.

”cargarMensajes()”: This method is in charge of reading a txt file that stores an array with objects, once read they are saved in a global variable called ”mensajesBase” which is an array of objects, after the messages have been read the method ”cargarMensajesPantalla()” is invoked, which analyzes the array and depending on who is the sender port and the receiver decides from which side of the screen the message is going to be loaded.

”agregarArchivo(int PuertoAmigo Package p)” In this method a new object containing the message is added to the variable ”baseMensajes” and then the txt file is overwritten with the new array that has a new message.

”void estabcerUsuario(LogicaNegocio us)”: this method allows us to save the user that tells us who was the person that entered the application and which are his contacts.

”public void initialize()”: this is a predefined method of the java fx programs which is executed before the components are loaded, here the thread1 is initialized and runs which contains a socket server that will be listening for messages.

”public void run()” this is a special method of the runnable class which provides the thread management, the thread used here runs in the initialize method.

Explanation of what happens in this method

Most of the code is inside a while loop except the creation of the server socket that is initialized using the sender port that provides the variable ”data” then inside the while loop we begin to accept messages from outside, once the message arrives and is saved in the variable data, a process of analysis of the information will begin, it will be determined if the message that arrives corresponds to the current chat or not, in case of not belonging the message will be saved in its corresponding txt file, in case if the message does belong to the conversation it will be analyzed if the message is encrypted or not, in case of not being encrypted the message will simply be shown on the screen, in case of being encrypted it is sent to call the method mensajeIzquierdoBtn(Package data) this method allows us to show the message encrypted or with electronic signature with a button next to it, when this button is pressed a popup window is sent which asks for a key to be entered, Once the key is requested, the type of encryption of the message is detected, when the type of encryption is detected a method will be invoked to decrypt or check the message as appropriate.

String decryptAsymmetric(String message,int key) this method is in charge of decrypting the message asymmetrically, it returns a string that would be the decrypted message.

String decryptSymmetric(String message,int key) this method decrypts symmetrically the message, it returns a string that would be the decrypted message.

Void comprobarFirmaDigital(String mensaje,int llave) what this method does is that it puts the message to a sha1 function and decrypts the summary with the key to verify that the digest is the same.Inside the chat there are 4 buttons to send messages, plain text, symmetric encryption, asymmetric encryption and send electronic signature.

Void sendServerPlano() this method creates an instance of a client socket that is sent to a server port, the message is sent as is.

Void sendServerSyimetrico() this method is executed when you press the button to send a symmetric encryption, a popup window asks for a key and then it is sent to call the method ”String ecripitadoCesarr(String message,int key)” which will be given a message and a key and gives as a result an encrypted message.

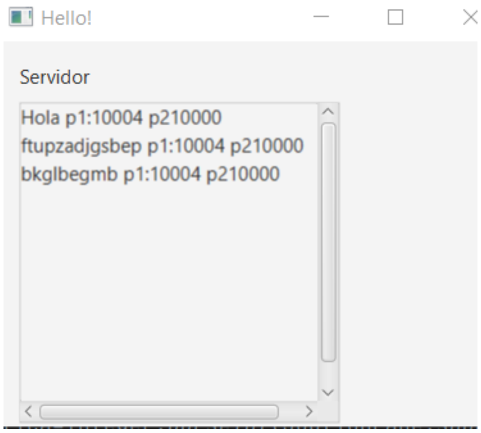
Void sendServerAsimetrico() this method is executed when you press the button to send an asymmetric encryption, it asks in a popup window for a key and then it is sent to call the method ”String encriptadoCesarr(String message,int key)” which will be given a message and a key and gives as a result an encrypted message.

”void sendFirmaDigital()”: this method first puts the message into a SHA1 to get the digest, then asks for a key to encrypt and encrypts the digest with cesar to get the encrypted digest. The three methods above in their last line of code invoke the method ”void sendEncriptado(String message, Package package)” this method is responsible for sending the information to the server and displaying the original message on the screen before being encrypted.

”void Regresar()” this last method is in charge of returning to the chat selection screen and sends back the user of the LogicaNegocio class so that the program knows who the logged in user was and displays his contacts.

D. Server

The server has a graphical interface that shows all the messages that pass through it, this part of WhatsUp is simpler since it only has a single thread and a function, a serversocket is created, then within a thread begins to accept the messages that arrive once the message is received it is displayed on the screen to see what reaches the server, the message received has information of the receiver, so a client socket with data.getPuertoReceptor(), in this way the message arrives to whoever it corresponds. The thread is inside a while loop which allows it to be constantly listening for messages.



```
Hola p1:10004 p210000
ftupzadjgsbep p1:10004 p210000
bkglbegmb p1:10004 p210000
```

Fig. 9. Chat screen

The figure 9 shows the server interface in operation, where you can see the messages that pass through, both those that are sent as plain text and those that are sent with encryption and whose information is not readable by the server.

E. AR

The AR program knows the certificate paths, it acts as a certificate database. It works as follows: A WhatsUP client in

3 moments will establish a connection with the AR, when it receives a digital signature and will check it, when it sends an envelope where it requires to search for a certificate to use a public key and when it verifies an envelope where it also requires a public key to check the content, when the connection is established the AR is sent the certificate that is sought, if it contains it sends it back to the client, otherwise a connection is established with the other AR to search for the certificate if it finds it, it is sent to the AR that made the request and this in turn sends it to the client.

Fig. 10. Chat screen

Fig. 10 shows a simple interface that belongs to the screen of the AR program.

VI. Conclusions

In conclusion, we can define that encryption is the part of cryptology that deals with the techniques, applied to art or science, that alter messages, by means of encryption or encoding techniques, to make them unintelligible to intruders who intercept those messages. Therefore, the sole objective of cryptography is to achieve confidentiality of messages. In the application of this project, integrity, authenticity and non-repudiation were maintained with the application of the signature and the application of the digital envelope, allowing users to send messages in a relatively secure way, where the relative security depends on the strength of the encryption used. This project should be taken into account for desktop or cellular applications whose main purpose is to send messages, since currently many of these applications do not have security or the one they have is very poor, which makes their users exposed to cybercriminals. As observed in the implementation of this project, it increases the privacy and security of users by applying cryptography and using public key infrastructures. It should be noted that although encryption is very heavy cybercriminals are very ingenious and will always seek to circumvent security so it is necessary to constantly improve the security of messages.

References

[1] SSL Certificates — Website Security — DigiCert. (s. f.). Recuperado 14 de septiembre de 2022, de https://www.websecurity.digicert.com/

[2] TSL Certificates — Website Security — DigiCert. (s. f.). Recuperado 14 de septiembre de 2022, de https://www.websecurity.digicert.com/

[3] HTTPS Certificates — Website Security — DigiCert. (s. f.). Recuperado 14 de septiembre de 2022, de https://www.websecurity.digicert.com/

[4] Conecta Software. (2020, 13 enero). Métodos criptográficos - una introducción. Recuperado 14 de septiembre de 2022, de https://conectasoftware.com/ciberseguridad/metodos-criptograficos-una-introduccion/

[5] González, A. (2020, 27 octubre). ¿Qué es el cifrado César y cómo funciona? Ayuda Ley Protección Datos. Recuperado 14 de septiembre de 2022, de https://ayudaleyprotecciondatos.es/2020/06/10/cifrado-cesar/

[6] DocuSign, C. de. (2022c, marzo 23). ¿Qué es la firma electrónica? DocuSign. Recuperado 23 de octubre de 2022, de https://www.docusign.mx/blog/que-es-la-firma-electronica

[7] Techopedia. (2011, 2 diciembre). Digital Envelope. Techopedia.com. Recuperado 23 de octubre de 2022, de https://www.techopedia.com/definition/18859/digital-envelope

[8] ¿Qué es un Certificado de Sello Digital? - Factura Fácilmente de México S.A. de C.V. (2021, 26 julio). Factura Facilmente de Mexico S.A. de C.V. Recuperado 23 de octubre de 2022, de https://www.facturafacilmente.com/que-es-un-certificado-de-sello-digital/

[9] Infraestructura de claves públicas (PKI). (s. f.). © Copyright IBM Corp. 2018. https://www.ibm.com/docs/es/ibm-mq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-sec-doc-q009900–htm