Pak. j. life soc. Sci. (2025), 23(2): 1119-1128 E-ISSN: 2221-7630;P-ISSN: 1727-4915

Pakistan Journal of Life and Social Sciences

www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2025-23.2.00239

RESEARCH ARTICLE

Methodology and Implementation of a Secure Data Exchange System

Safaryan O.A[1], Alferova I.A[2], Buryakova O.S[3], Galichev N.A[4]

[1,2,3,4]Don State Technical University, Rostov-on-Don, Russian Federation

ARTICLE INFO

Received: Jan 12, 2025

Accepted: Mar 6, 2025

Keywords

Information Protection
Information Security
Commercial Secrecy
Secure Data Exchange

***Corresponding Author:**

safari_2006@mail.ru

ABSTRACT

The aim of this study is the software implementation of a secure data exchange system. This article discusses approaches to the organization of secure information exchange. An analysis of the ways in which information is protected during transmission to messengers is presented. Recommendations are given to improve data protection against unauthorized access

INTRODUCTION

Today, with the development of information technology, more organizations are using software to exchange information between employees. The vast majority of companies use instant messaging systems - messengers. This solution is justified by the ease of use and their prevalence. They allow free and fast data transfer between employees. However, given the growth in cybercrime, interest in confidential information of the company and its employees from both intruders and messengers, Cases of data transfer to the special services of the respective States and industrial espionage need to fundamentally reconsider the approach to the use of messengers within organizations. Frequent user data leaks, unreliable information protection methods, cooperation with special services of foreign states, closed source code pose serious threats to the security of the company using the data exchange application. The attacker who has access to the user account of a popular messenger will be able to view all preserved records and documents in corporate dialogues and conferences without problems. Also, at any point in time, the security policy of the application may not change for the better for the organization that uses and transmits its sensitive data through it.

One of the solutions, which can significantly increase the information security of the organization, may be to deploy its own software for the transmission of data within the contour of the enterprise. In this case, all the infrastructure is hosted on its own servers, serviced by in-house specialists. This approach provides its own demarcation of data access, full and continuous control over the servers, their flexible configuration and self-service. Thus, the on-premise approach overcomes the main shortcomings of conventional messengers, allowing you to develop your own software with reliable information protection methods and place it on your own server.

The purpose of this article is to implement a multi-user web application for data exchange.

1.Operating principles of instant messaging systems

With the increase in the number of Internet users and owners of smartphones, instant messaging systems, or messengers, have become very popular for quick and convenient exchange of information as between people for private purposes, between staff members within organizations

for commercial purposes. Instant messaging system is a special program designed and intended for the transmission of text information, audio and video data, files and documents in real time through the Internet. This system is used for implementation of chat, broadcasts, conferences, games and other applications where it is necessary to instantly process and display the information that users share with each other. The operation of the instant messaging system is based on a client-server architecture. The user is provided with a client program with an interface, thanks to which he can enter a message and send it to his interlocutor. Usually, the application shows the user the status of the conversation participant in the network, the process of entering text into the field and the status of the message after it has been sent. The server part to which a client is connected is responsible for processing and storing data about users and information they send. In the modern world there are a large number of messengers used by people and corporations all over the world [1,2].

Most companies developing rapid messaging systems use closed protocols for secure data transfer between customers, which on the one hand allows companies to keep secret details of the mechanisms that ensure the security of their application, in order to gain a commercial advantage, on the other hand, lowers the potential level of security of the messenger due to the secrecy of the source code and its inaccessibility to a large community of developers who can confirm or deny the existence of vulnerabilities in this software. Therefore, some developers of messengers use open and time-tested protocols [3-5].

Every messenger has its own security flaws: a closed source code, the presence of backdoors in the system, unreliable methods to protect transmitted and stored data. This is causing mistrust among potential users, especially organizations planning to use instant messaging systems for commercial purposes.

Choosing a reliable cryptographic protocol plays an important role in ensuring the security of the instant messaging system [6-9]. Cryptographic protocol includes various cryptographic algorithms, their description and rules of application. The participant, or subject, in the protocol is a person, group of people, application or organization, that is, any entity capable of performing an active or passive role in the operation of the protocol. The encryption protocol performs functions such as:

- creation of an electronic signature to verify the integrity of data and confirm their authorship;

- generation of cryptographic keys;

- key exchange;

- key splitting;

- System access delineation;

- ensuring the integrity of information and connections;

- Authentication of participants;

- ensuring data confidentiality;

- fail-safe.

- Authentication of participants;

- ensuring data confidentiality;

Fail-safe [4].

To ensure the security of the application, the protocol can use encryption, hashing, calculation of values of unilateral functions, generation of random numbers, reception and sending of messages, authentication of signatures, keys and certificates.

The degree of centralization of the application is also key to protecting the transmitted data. There are three stages in the centralization of instant messaging systems. Most popular messengers are centralized. They are classic client-server applications, where each user uses the client and their interaction takes place through a common server. In this case, additional servers can be used for data processing, but the point is that they are controlled by one company. Centralized systems are easy to design, implement and control. But this degree of centralization also has its drawbacks in the form of a possible blocking of the central point of entry by public services, which would make it impossible

to use the messenger. Another drawback has to do with the core of the system itself - there is only one owner, and nobody else knows how the server actually works and what's going on.

In a decentralized system, or peer to peer (P2P), each client is a server. That is, there is no server in the classical sense. Such a peer-to-peer network, in fact, can consist of and work with already two users. In addition to being fully block-resistant, this system does not require any user information including phone number or email. Therefore, it is not possible to prove authorship by a specific person. One user is one device. It stores all information related to correspondence and conversations with other members of the system. The data is also encrypted when transmitted. The disadvantages can be attributed to the complex implementation of this type of application and the lack of a system for restoring access to the account due to the principle of operation of the peer system.

OpenPGP is a cryptographic protocol based on Pretty Good Privacy (PGP), which is freely available and free [10-12]. Is the de facto standard for ensuring security of data exchange in electronic mail services. Also used in many applications to protect transmitted information through end-to-end encryption.

The OpenPGP protocol is based on symmetric and asymmetric algorithms. Each system participant has its own set of keys, consisting of a private and public key. The private key is kept secret, it is used to sign the sent message and decode the incoming message. The public key is freely distributed and shared with other conversation participants. This key encrypts the sent message and verifies the signature of the incoming message. But most often the asymmetric algorithm is not used to encrypt the transmitted information, but to create a protected transmission channel of the common secret symmetric algorithm, with which the encryption of the messages takes place. That is, using a symmetric algorithm to encrypt the transmitted data, and using an asymmetric algorithm to encrypt its key. OpenPGP is a reliable cryptographic protocol to ensure the protection of transmitted data through end-to-end encryption, and has a number of advantages such as:

- the user can generate their own PGP key pairs by choosing encryption algorithms;

- use of current and reliable cryptographic algorithms;

- the user can store several pairs of keys for their own purposes and independently choose a pair for a particular case;

- open-source code;

- is actually the standard for encryption in e-mail;

- providing end-to-end encryption for users of any operating system;

- free distribution and use;

- support for asymmetric keys up to 4096 bits;

- using the system for more than thirty years.

Thus, a secure data exchange system is an important tool for ensuring the security of data and improving employee interaction when sharing it.

2 Development of system modules

A Web-based data exchange application should meet all the general requirements for its safe operation in order to eliminate or reduce the risks of unauthorized access, theft of information, compromise, violation of confidentiality, accessibility and integrity of data. Therefore, it is necessary to form the security requirements of the application being developed and implement them. One of the main mechanisms to protect against unauthorized access to a web application for data exchange is the authentication and identification system. The system should implement authentication, based on the knowledge of the user of some secret information known only to him and the verifying party - the server [6]. In the case of a developed messenger this secret is password. The implementation of password protection should be based on the application of a repeated password by the user when authenticating and comparing the value of his hash with the stored value in the database. Passwords should not be kept in the open. Instead, the database stores hashes of passwords [5, 13].

To be more reliable, the password should have a set of minimum requirements: the length of the password should be 8 characters, and the password should contain one upper case character, one lower case character, one digit and one special character. When entering a password, the corresponding field should be masked by default. If you want, the masking can be disabled. Also, for the correct operation of password manager, the authorization form must be done on a separate page, excluding consecutive logins and then passwords. For security reasons, the password and all other user data required for authorization must be transmitted to the server by means of a POST request, which allows storing data in a request body that is much more difficult to access. Also in this case, the data does not remain in the browser history or server logs. In addition to authentication, it is necessary to define the requirements for user identification in the application under development. Identification is the process of identifying and assigning a unique identifier to an individual to uniquely identify a user within our web application. The ID must be unique. They can be email, username, phone number. Identification in the system to be developed will occur on a unique user name, or no-name. When registering, the user specifies it, at which time a check for the uniqueness of the entered name should take place. If the entered value is unique, this identifier is assigned to the user. After authorization, the user must be given a special access token or an authorization token to work safely with the application. Access token-based authentication simplifies the authorization process, speeds up system work and improves overall application security. In this case, you do not need to enter your account login and password each time. The token stored in the browser is sent to the server for each new request within one open session. In order to receive the access token, the user must have successfully registered or logged into the system. After sending the data to the server, they are validated. Upon successful validation, the server generates an access token based on user data including ID, time of creation and signs it with its private key, and only then sends it to the user where it is saved in the browser. It is important not to create a token based on the user's confidential information, such as a password, because the token is not encrypted but only signed. It is also necessary to set the life time for the token within 7 to 30 days. This measure can protect the user account when accessing the token by an attacker.

One way to provide a user access token is the JWT standard, which will be used as part of the web application being developed. JSON Web Token (JWT) is an open and popular RFC 7519 standard for generating access tokens [16]. It uses JavaScript Object Notation (JSON) to exchange tokens between the client and server, thus protecting data transfer between them. In addition to reliable and fast authentication, the token is also used to transmit additional information about the user account, such as ID or name. The token is passed in the HTTP request header and contains the necessary user data to avoid unnecessary requests to the database. JWT consists of three parts, presented in figure 1: - header (header), which must include the type of signature algorithm, and optionally the type of token and content; - payload (payload), which includes user data and other additional information, such as user ID, role, name, email, token lifetime and time of its creation; -signature (signature) based on header and payload

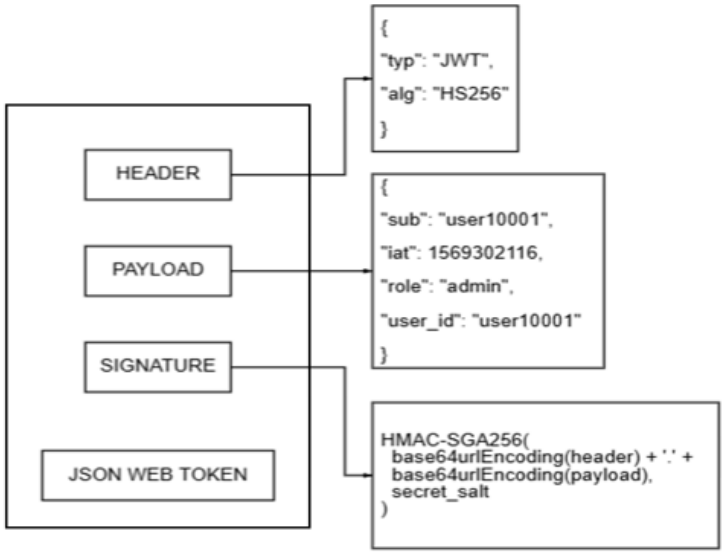

Figure 1 - Composition of JWT

Once signed, it is sent to the user and saved in the browser. Each time a user requests an application, a token is sent to the application where the server first verifies the signature with its public key and then provides access to the protected resource as needed, as shown in Figure 2.

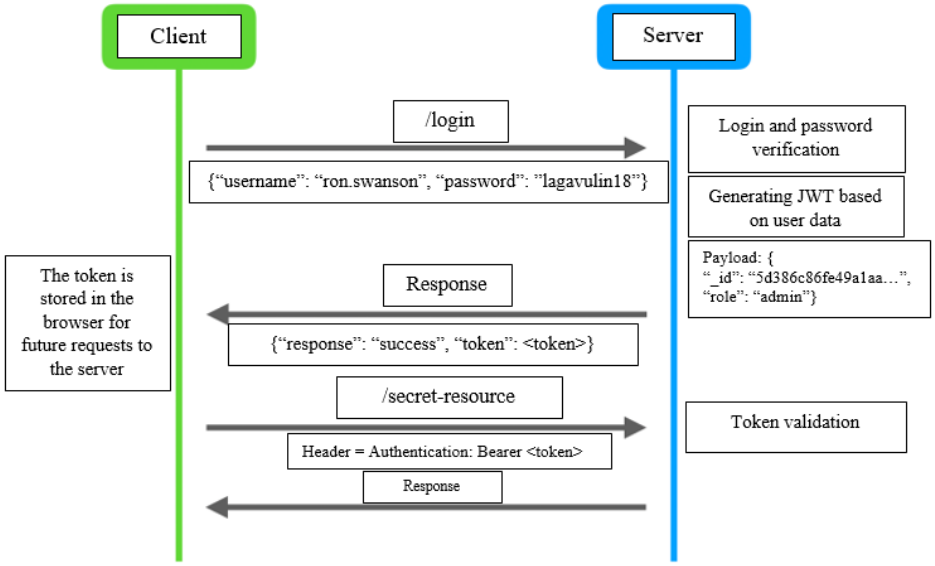

Figure 2 - JWT diagram in web application

Within the framework of the system under development, the payload of the token will contain a name, e-mail, username, user ID. The validity period, or life time, of the token will be set as 7 days, or one week. The limitation of the life of a token is due to the fact that when it is stolen, the attacker will not be able to use it for a long time. Once the token expires, the attacker will lose access to the user account.

For secure data exchange between the client and server deployed on the Internet, it is necessary to use network protocols that provide appropriate protection. Using HTTPS is the solution to this problem.

HTTPS is an extension of the Hypertext Transfer Protocol (HTTP), which implements secure data transmission over the network. HTTP is a network protocol that allows the client and server to interact on the Internet. This protocol is at the application level and carries out the exchange of information according to the scheme «request cloud». It uses the capabilities of TCP (Transmission Control Protocol) and its session, under which it can send requests. The default port is 80. HTTP is extensible, simple, fast and has no states, but the messages in it are transmitted unencrypted as shown in Figure 3, which compromises the security of using this protocol when transmitting sensitive information. For this purpose, a corresponding HTTPS (HyperText Transfer Protocol Secure) extension has been implemented, which is able to solve the problem with the protection of transmitted data on the Internet.

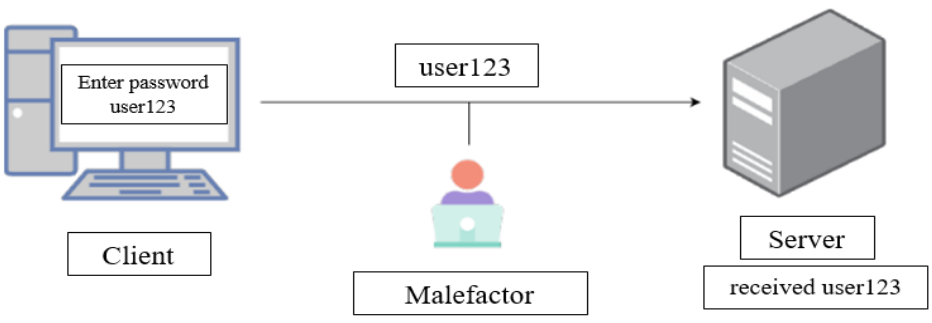

Figure 3 - HTTP data transfer scheme

HTTPS is an extension of the HTTP protocol, which encrypts data transmitted through it using cryptographic protocols TLS (Transport Layer Security) and SSL (Secure Sockets Layer) [23]. This extension uses 443 TCP ports. As shown in Figure 4, the data transfer is encrypted. When messages are intercepted by an attacker, it will only receive encrypted text that is useless without keys. The Web server requires an open key certificate to establish a secure connection, as symmetric and asymmetric algorithms are used for encryption. An asymmetric algorithm generates a common secret key, which is then used for symmetric encryption of transmitted messages. The open key certificate is sent to the client by the server when the connection is established. The certificate confirms that the public key belongs to the server. The private key is stored on the server and used to decrypt data. HTTPS can also be used for authentication, using certificates to grant access to the site only to specially authorized users. In this case, the administrator creates individual certificates for each user and passes them on to them. The certificate contains unambiguously identifiable information about the user, such as e-mail.

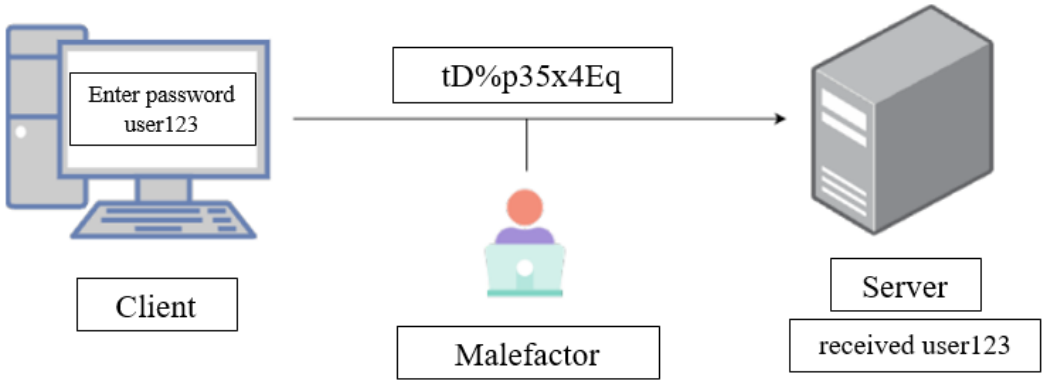

Figure 4 - Data transfer scheme over HTTPS

As a result of the consistent execution of all necessary actions, a secure connection between client and server is established. Transmitted messages are encrypted with a symmetric algorithm using a session key. After closing or interrupting the session, the secret key is destroyed.

3 Algorithmic software development

The main purpose of the web application being developed is to secure information exchange between users, reliable data storage and safe use of software. Based on this, it is possible to determine the necessary system functionality for realizing this goal and build its algorithm of operation. Web application should have the following functionality:

- registration and authorization of users in the system;

- exchange of messages between interlocutors in real time;

- generation of a pair of cryptographic keys for the user;

- import keys into the system for later use;

- export of keys from the system and keep them protected; - verification of integrity of transmitted information;

- confirmation of message authorship;

- use of end-to-end encryption;

- keeping the history of information exchange.

Registration in the application is done with a username, full name of the registrant, e-mail and password. To validate the data on the client part, all fields must be filled in and the password must meet the minimum requirements. After validating on the client side, the data is sent to the server where another validation takes place. The server also verifies that all fields are filled in, that the user's name is unique and that the e-mail address is correct. If the validation is not successful, the

server will send an error response and information about it to the user. During validation, the user and information about him is stored in the MongoDB database.

The Web server interacts with the database through the Mongoose library. Mongoose is an Object-Relational Mapping (ORM). It allows you to model web application data using schematics and interact with MongoDB while doing validation, type checking and other logic for processing requests. Instead of the user's password, its hash is saved in the database. Cryptographic hash function b crypt, based on the Blowfish cipher, is used to compute the hash. The number of rounds for generating the salt needed to protect against attacks is 10.

Next, the access token is generated with JWT. It is based on information about the user to whom the token is given: identifier, name (nickname), e-mail address and full name. Token has a life span of seven days. An asymmetric RS256 algorithm is used to generate and verify the signature, which assumes both open and private keys. Hash is calculated using the SHA-256 hash function and signed with a pre-generated RSA private key. After signing the token is sent to the client where it is placed in a cookie with a lifetime of one week. As a result, the user successfully registers and accesses the application under their account. The user token will be used to access requested web application resources. The complete logging algorithm in the system is shown in Figure 5.

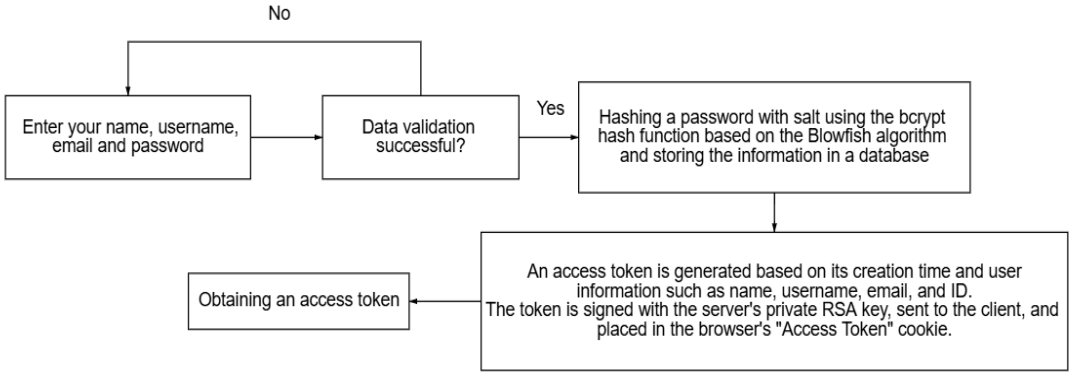

Figure 5 - Registration architecture in the application

After registration, the user must generate or import a pair of open and closed PGP keys to be able to exchange messages using end-to-end encryption. If you have keys, you can import them into the system in compressed ZIP format. Otherwise, you need to generate the key. For this purpose, the user must enter a secret phrase with which the generated private key will be encrypted for safe storage. The minimum requirements for a secret phrase are the same as for a user password: 8 characters, 1 lower- and upper-case character, 1 digit and special character. PGP key generation is based on unique user IDs: ID and e-mail. After entering the secret phrase, the keys are generated in RSA format and 4096 bits long. They will be encoded using Base64. The private key will be encrypted with a previously entered secret phrase. The keys will be stored in a local browser storage (local storage) and will be used for cross-encryption in the future The public key will be sent to the server and stored in the database. If a browser's local storage is accidentally cleared, the keys will be removed. Without a key, the user will not be able to access messages sent and received previously, so it is suggested to export generated keys to disk. They will be uploaded to the device in the ZIP archive. For authorization in the application, the user must enter a name and password. After the client sends data, it is validated on the server side. The user's name is searched in the database. If the user with this name is not found, the server sends an error message to the corresponding client. Otherwise, the server calculates the hash of the received password and compares it with the hash from the database. If the hashes do not match, the server returns a bad password error. If the hash matches, the access token is generated in the same sequence as during registration. After the client receives a token, it will access the account and open a new session. The access token is sent by the client to the server with each request. The server verifies the identity of a token by verifying its signature with its public key. If the token is verified, the user gets access to the requested resource. If not, the server redirects the user to the authorization page. After a successful authorization, the application automatically searches for PGP keys in the browser's local storage. If keys are not found, the user is prompted to generate or import them. If keys are found, they will be used for end-to-end

encryption. If you open a new application window or refresh the page during a session, you will be prompted for a secret phrase to decrypt the closed PGP key. After successful entry of the phrase, access to the key will be obtained and the user can continue to exchange messages with interlocutors. In case the user has forgotten a secret phrase, the application will suggest to create a new pair of keys.

4 Develop a software product

Application for data exchange using end-to-end encryption based on OpenPGP protocol includes registration and authorization of users, generation of PGP key pairs, their export and import, and sending messages to the interlocutor using end-to-end encryption.

For registration you need to enter username (nickname), full name of the person registering, e-mail address and password. The password must meet minimum requirements in order to protect the account from attack by a complete firewall. To verify the entered password, you can disable the character hiding mode in the corresponding field.

After successful registration, the user is prompted to create a new PGP key pair, or import their existing one as shown in Figure 6.

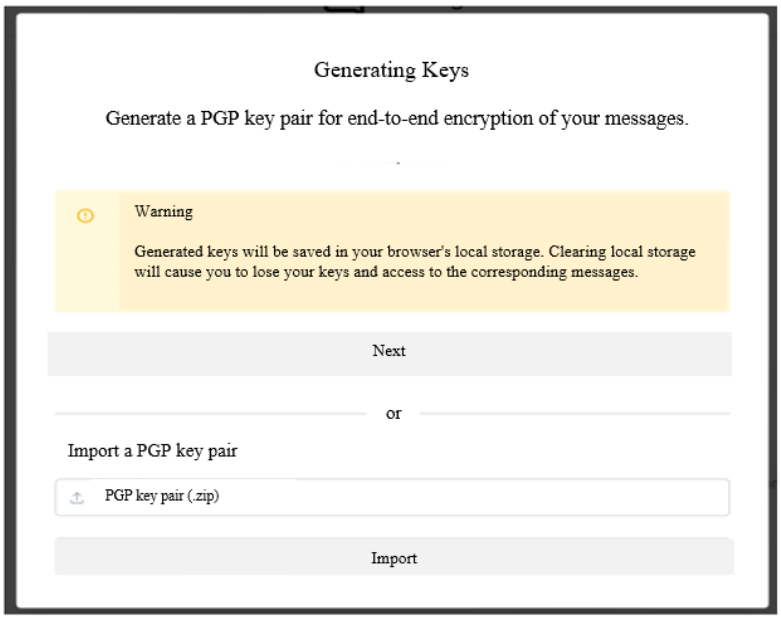

Figure 6 - Creating and importing a pair of keys

For security reasons, the private key must be kept encrypted. In order to encrypt it, you must enter a secret phrase that should meet the minimum requirements for protection against a complete overdraft attack. The generated pair of keys will be stored in the browser's local storage, but for greater security you can download it to a disk or any other safe storage location. It will be archived in a protected format. If the keys have already been exported, or do not need it, you can skip this step by selecting the appropriate item. If the keys are lost, it is not possible to recover them. Access to messages that used this pair of keys will also be lost forever.

The authorized user must enter a secret phrase each time they open or restart an application tab in their browser to access a private key that decodes messages sent to them and signs their own. Figure 16 shows a window for entering the required secret phrase.

After entering the secret phrase, the user is presented with the initial screen of the application with the possibility to select a interlocutor and start a dialogue with him. You can select a conversation partner from the existing dialogs or by searching for their name. After selecting a new interlocutor, a new dialogue with him is initiated. When you select a user from the list of existing dialogs, a chat will open with the corresponding user and history of sent and received messages. Sent messages are displayed in the dialog on the right, received - on the left. Each message is marked with a tag that shows the time of creation. The exchange of messages between users takes place in real time.

Next to the name of the interlocutor is information that the dialogue is protected by through encryption. When you click on the appropriate label, you can see detailed information about your interlocutor's public key and view the public key itself as shown in Figure 7. The user interface shows detailed information about your private and public key. It is also possible to export keys or generate new ones. When you open a session, you can sign out of your user account. When you log out of your account, the user is directed to the authorization page in the web application. The authorization window is also displayed by default whenever a user logs on to an application without opening an active session. If the user does not have an account, they are invited to register a new one.

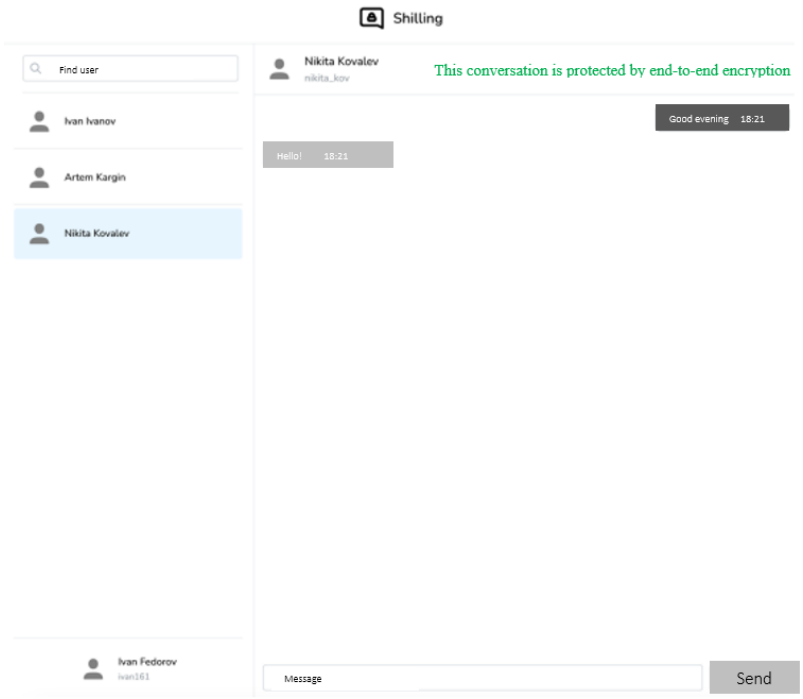

Figure 7 – Application window

Next to the name of the interlocutor is information that the dialogue is protected by through encryption. When you click on the appropriate label, you can find out details about the public key of your interlocutor and see the public key itself. In the user interface you can see detailed information about your private and public key. It is also possible to export keys or generate new ones. When you open a session, you can sign out of your user account. When you log out of your account, the user is directed to the authorization page in the web application. The authorization window is also displayed by default whenever a user logs on to an application without opening an active session. If the user does not have an account, he is invited to register a new

The practical task of ensuring information protection has been accomplished. The information protection mechanism includes various means and methods, measures and activities to ensure the security of the information transmitted, stored and processed.

CONCLUSION

In conclusion, it should be noted that the protection of information is characterized by a complex of activities aimed at ensuring its safe storage and hacking.

As a result of the work implemented multi-user system for data exchange using end-to-end encryption based on OpenPGP protocol. Special attention was given to the protection methods in messengers, in particular the importance of choosing a reliable cryptographic protocol, implementation of end-to-end encryption and degree of centralization of the application.

The developed web application allows to exchange data with users in real time, reliably protecting the secret of correspondence from third parties thanks to the implemented in it through encryption.

REFERENCES

1.Orellana Reyes G. Evaluación de privacidad y seguridad de datos de usuarios en redes sociales / G. Orellana Reyes, G. Cotera Ramirez // Revista Científica Arbitrada Multidisciplinaria PENTACIENCIAS. – 2025. – Vol. 7, № 5. – P. 87–110. DOI: 10.59169/pentaciencias. v7i5.1641

2.Hendler J. Security of WhatsApp and the role of metadata in maintaining privacy / J. Hendler // arXiv:1701.06817 [cs.CR]. – 2017. – 8 p. URL: https://doi.org/10.48550/arXiv.1701.06817

3.Sharma A. Information Security Policy Compliance / A. Sharma, A. Koohang, S. Pal Singh // Journal of Global Information Management. – 2025. – Vol. 33, № 1. – P. 1–32. DOI: 10.4018/JGIM.389715

4.Unger, N. SoK: Secure Messaging – San Jose: Symposium on Security and Privacy, 2015. – C. 232–249.

5.Mobile Protocol: Detailed Description. // URL: https://core.telegram.org/mtproto/description

6.Ashanin A.O. On the Specific Features of Legal Regulation of the Use, Processing, and Transmission of Restricted Information // Bulletin of the Dimitrovgrad Engineering and Technology Institute. 2023. No. 2 (30). Pp. 71-78.

7.Chamzo O.D. Confidential information and methods of its protection // Bulletin of the Magistracy. 2023. No. 2-1 (137). Pp. 58-59.

8.Sharipova A.A. Methods of Protecting Confidential Information // Young Scientist. 2024. No. 42 (541). Pp. 203-204.

9.A new usage control protocol for data protection of cloud environment // Kefeng Fan [and oth.] // EURASIP Journal on Information Security. 2016. № 7. pp. 1–7.

10.Anooplal. K. S, Girish S. An Infallible Method to Transfer Confidential Data Using Delta Steganography|| International Journal of Engineering Research and Technology (IJERT). 2015. Vol.4. Issue 2. February. pp. 1060 – 1063.

11.Anooplal. K.S, Girish S., Arunlal. K.S An Infallible Method to Hide Confidential Data in Video Using Delta Steganography// International Journal of Engineering Research and Technology (IJERT). 2015. Vol.3. Issue 4. February. pp. 228 – 234.

12.Gartner Identifies the Top Cybersecurity Trends for 2025. URL: https://www.gartner.com/en/newsroom/press-releases/2025-03-03-gartner-identifiesthe-top-cybersecurity-trends-for-2025