JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 39 (2026) 161–166
Received: 3 April 2026
Accepted: 10 May 2026

Comparative analysis of the security of instant messaging apps

Natalia Pioterczak*, Maksymilian Potocki, Piotr Kopniak

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Communication through the Internet network is an unavoidable type of data exchanging in a modern society. Ensuring that user experiences a comfortable environment allowing for sending and receiving information is a need taken seriously by a lot of applications with the through Internet communication as their goal. This article is focused on the security solutions provided by the tested instant messengers. The research consists of eavesdropping the network traffic to check the applications' security during the communication, analysis of encryption algorithms and security of user's account. Each application was then examined in the context of all the previously listed features. Conclusions were drawn indicating the possibility of choosing an instant messenger that protects end user the most.

Keywords: instant messenger; security; encryption

*Corresponding author

Email address: **s95527@pollub.edu.pl** (N. Pioterczak)

1. Introduction

The omnipresent access to fast Internet and suitably efficient units allowed society to use fluid and unwavering communication. Developers of new modern software face the challenge of providing the right tools for communication between at least two end-points at an appropriate level of security, along with the growing demands of users.

Every end user interested in voice, text or image transmission should have access to information regarding most appropriate communication tool for their needs: emphasizing specific user security needs, every aspect which might caught user's attention when using said application. Users should have full knowledge about the way their data that they sent to the application is being processed, is there encryption mechanism involved and also to which degree their sent data is being isolated from undesirable sources.

The need for such data was an inspiration to conduct research that establishes hierarchy among applications used for Internet communication. Several instant messengers will be tested for ease of use (easiness and intuitiveness of the graphical interface, quality of transmitted voice and image and also security measures regarding getting access to that information by undesirable sources.

2. Literature review

This chapter is dedicated to a review of the literature on instant messengers, their security measures concerning information transmitted via them and ciphering algorithms used in the process of transferring data between users.

Social media has become an integral part of everyday life worldwide. Instant messengers are used globally in the information exchange process. Not only ordinary citizens enjoy the benefits of this technology for transmitting data to another person, but also people in high positions and public figures, where the privacy and transmission security must meet the highest standards.

The subject of privacy and security regarding exchanging messages, whether via text, voice, or video transmission, has always been a factor that raises doubt and uncertainty in society. Modern instant messenger applications face the challenge of ensuring the highest standards when it comes to encrypting data transmission between end users. In the article [1] authors mention that early Internet messengers did not use end-to-end message encryption techniques, whereas modern ones are committed to this. In another article [2], it is said that encryption techniques currently in use include: Kryptokaz, Encrypt, EnDe-Crypto and AES-Crypto.

Complete encryption, however, can be a matter of debate. In the article [3] and [4] the authors assume that in some cases, data transmitted between users may be used to intercept information about engaging in illegal activities. The option to delete the messages right after reading them can also make it difficult to prove someone's guilt in a certain situation.

When it comes to instant messaging, the issue of moderating content sent between users is also important. To ensure the greatest possible privacy, instant messengers try to automate the process of verifying the content processed by their own servers and use bots that analyze sent messages. In the article [5], it is presented how message in case of violation of the community guidelines and the terms of use of a given messenger can be deleted and the sender may lose access to their account.

The ongoing development of technology also means that end users want to communicate using a variety of different devices. While applications were once available for specific devices and systems, the article [6] mentions how the cross-platform support is now an important feature of a modern instant messenger. The user has the ability to make contact both on a computer with Windows or Mac OS, as well as on phones with Android or iOS, and also on other systems.

3. Purpose of work and hypotheses

The aim of the article is the aforementioned comparative analysis of the security of instant messaging apps which includes applications like Discord, Microsoft Teams, Zoom, Messenger, Signal, TeamSpeak 6 and WhatsApp. These applications will be installed and appropriately configurated to allow communication between two users. During testing each of the aforementioned applications will be tested for message encryption security and verification of security techniques protecting end users declared by developers.

The following hypotheses have been phrased and will be checked within the scope of this article:

H1: Signal is the golden standard in case of the voice communicators security.

H2: Discord is mostly focused on integrity and comfort of use rather than the security aspects.

H3: Most of the tested applications aren't providing enough security covering the vulnerability of getting access to other user's public IP address.

4. Materials and methods

On the test workstations programs will be configured with the highest possible encryption and overall program security settings.

During experiment the following instant messaging apps will be tested: Discord, TeamSpeak, Messenger, Microsoft Teams, Zoom, Signal and WhatsApp.

4.1. Technologies and tools

Security of tested applications covers checking the security protocols these applications provide. Each of the instant messengers providing encryption uses selected algorithms and the appropriate type of encryption.

The encryption algorithms consist of symmetrical and asymmetrical algorithms. The former are based on the fact that the same key is used for both encryption and decryption processes whereas encryption and decryption functions are very similar to each other or even identical. Examples of algorithms from this family include AES, ChaCha20, and DES. This type of encryption is usually used in case of need for fast processing large amounts of information.

Asymmetrical algorithms on the other hand are characterized by diversity of encryption keys, which are divided into public key allowing for data encryption and is free to share and secret private key that allows a person with access to it to decrypt encrypted information. Examples of algorithms of this type include: RSA (Rivest-Shamir-Adleman Algorithm) and Diffie-Hellman. This encryption is used mainly for key management [7].

During research the use of the TLS (Transport Layer Security) protocol for the security of communication between users in terms of data exchange will be tested. It is a basic cryptographic protocol that authenticates information sent by the client to the server. Subsequent versions gradually improved in terms of checking the consistency of transmitted data.

The process of making a connection called handshake is the basis of the TLS protocol containing all the information regarding transmission of data: key negotiation methods used and encryption, TLS version and also authentication algorithm used. The 1.3 version of TLS uses asymmetrical algorithms for secure exchange of keys between transmission parties and symmetrical algorithms for data encryption. This combination of traits allows for an efficient and secure connection which translates into its broad use in various currently functioning IT systems [8].

Key feature of multimedia instant messengers is E2EE (end-to-end encryption). This model allows for decrypting an encrypted message only by its sender and receiver. It assumes that neither the internet service provider nor the server through which the communication takes place has access to such data. The cryptography is used exactly the same like with the TLS protocol – as combination of asymmetric and symmetric cryptography, which allows for both safe and efficient communication between users. This model is used in applications that required the highest level of privacy.

Signal Protocol [9] derived from the name of the application studied in this work is the most advanced protocol used in communication that uses E2EE in all its aspects:

- isolation of the proxy server in data transactions,
- changeability of keys to protect against their interception by an unauthorized source,
- security codes assigned to users,
- E2EE covering all forms of communication within the application.

For research purposes seven applications were chosen to determine their security abilities and choose the most suited combination of good security practices and the use of modern ciphering algorithms.

- **Discord** – instant messaging and VoIP social platform designed for private communication between two or more people as well as an application to manage a private server containing voice channels and text channels with multiple users able to simultaneously talk.
- **TeamSpeak 6** – VoIP service application that allows the end user to join a public server and create a password protected channel in such or join an existing channel to speak and chat with multiple users.
- **Messenger** – instant messaging service based on the communication between Facebook accounts that is required for the access to the application use. It gives users the ability to create group chats, individual calls and user-to-user messaging.
- **Microsoft Teams** – team collaboration platform based on a Microsoft cloud. Application serves the purpose of sharing the functionalities of video conferences and a workspace chat.
- **Zoom** – videotelephony software program basing on a videoconferencing communication and organizing closed video meetings requiring an access code from a participant to join.

- **Signal** – strictly encrypted instant messaging and VOiP service including one-to-one communication or group messaging. Requires having a phone number that identifies the end user.
- **WhatsApp** – instant messaging and VoIP service to send messages, make voice and video calls. Requires identifying user by his verified phone number.

The study is based on the packet traffic analyzed with an adequate application able to bring concrete results.

- **Wireshark** – network troubleshooter and a packet analyzer used to check the protocols in the network traffic. Main application used as a researching tool used to examine the security of all of the previously mentioned applications and services.

4.2. Test cases

Since every instant messenger may differ from the others in every aspect, communication between end users can look different in all cases. To prepare for examination regarding security of these applications a list of activities allowing for proper evaluation of the application was prepared:

- applications will be installed and configured,
- using Wireshark application security comparison will be made regarding the performance declared by their developers in the field of data transmission encryption,
- studied programs will be compared based on functions that allow users to secure their app account access and their stored data from cyber criminals trying to steal confidential information by eavesdropping on network packets and also from the application owner's side,
- based on tested aspects top application that perfectly executes the security needs placed before it will be selected.

Research scenarios are based on the developer's documentation and other official sources regarding its application:

- conducting test communication between two users in the field of voice and video calls, sending text messages and larger files, screen sharing,
- verification of the presence of E2EE,
- which protocols and version are they using during data transmission for example TLS for the transport layer security,
- what information is openly available when sniffing the network traffic,
- which user data is stored by the application server,
- alternatives for securing access to the account data by the user for example by security code, password or PIN (Personal Identification Number).

4.3. Testing environment

Research workstations used in testing that will have applications installed are characterized by the following specifications:

Table 1: Testing environment

	Workstation 1
CPU	Intel Core i7-2600 3,4GHz
RAM	12GB DDR3
Screen size	24” (1920x1080)
Operational system	Windows 10 Pro 22H2 64bit
Disk capacity	1TB SSD
Model	Dell Optiplex 790
	Workstation 2
CPU	Intel Core i7-4770HQ 2,2GHz
RAM	16GB DDR3
Screen size	15,4” (2880x1800)
Operational system	Windows 11 Pro 24H2 64bit
Disk capacity	256GB SSD
Model	Apple MacBook Pro A1398
	Workstation 3
CPU	Qualcomm SDM765G 5GHz 8core
RAM	12GB
Screen size	6,55” (1080x2400)
Operational system	Android 12 Snow Cone
Disk capacity	256GB (embedded)
Model	OPPO Reno4 Pro 5G CPH2089

Workstations marked by “Workstation 1” and “Work-station 2“ are the main workstations with Windows operational system installed on them which were used in communication between two users, whereas mobile phone workstation marked as “Workstation 3” was used only in situations where it was required to install applications on Android system in order to complete account registration process or change account settings.

5. Results

The analysis of the research results consists of each security issue examined separately for each application. The research was based on what the application author promised and declared in the relevant technical documents or whitepapers and what the research demonstrated in reality.

Each of the applications had different level of user security declared which is not necessarily caused by negligence of the creators, but rather emphasis on certain functionalities and the very nature of the application.

Security research analyzes tested applications from both a theoretical perspective - developer's declarations - as well as from a practical standpoint usually by scanning

network traffic with Wireshark and analyzing the security performance of the application itself.

The scope of practical tests using Wireshark program covered checking if the communication between application users uses E2EE, the visibility of users IP addresses of members of the call for the person intercepting network traffic and possible protection against the risks associated with it. The visibility of packets and metadata in network traffic associated with current nature of conversation in a given instant messenger was also crucial. Difference in size and type of packages depending on given activity will also be taken into consideration - whether the user was receiving a file from another user, establishing a voice call, a shared screen or webcam connection with him. The information about whether the user uses said instant messenger is also visible in Wireshark, but not in all cases and this has also been tested.

In the presented results all the applications are compared to each other in the purpose of finding the best one among all tested with traffic network monitor application Wireshark.

Table 2: Testing hiding message in packet traffic

App	Hides messages from eavesdroppers
Discord	Yes
TeamSpeak 6	Yes
Messenger	Yes
Microsoft Teams	Yes
Zoom	Yes
Signal	Yes
WhatsApp	Yes

Table 3: Testing the presence of E2EE

App	E2EE
Discord	Yes (not mandatory)
TeamSpeak 6	Yes (always)
Messenger	Yes (in new conversations)
Microsoft Teams	Yes (if both sides turn on)
Zoom	Yes (if both sides turn on)
Signal	Yes (always)
WhatsApp	Yes (always)

Table 4: Testing the ability of hiding user's public IP address

App	Hides users IP
Discord	Yes – through the server
TeamSpeak 6	Yes – through the server
Messenger	No – P2P communication
Microsoft Teams	No – P2P communication
Zoom	No – P2P communication
Signal	Yes – through the server (if set)
WhatsApp	Yes – through the server (if set)

Table 5: Testing if the application shows its domains name in the packets content

App	Shows application domains during the time of use
Discord	Yes
TeamSpeak 6	Yes
Messenger	No
Microsoft Teams	Yes
Zoom	No
Signal	No
WhatsApp	Yes

Table 6: Analyzing the TLS versions

App	TLS version
Discord	1.3
TeamSpeak 6	1.3
Messenger	1.2 and 1.3
Microsoft Teams	1.2 and 1.3
Zoom	1.2
Signal	1.3
WhatsApp	1.3

In the scope of the aforementioned research there is also mentioning of checking up the multi-factor authentications (MFA) or other forms of protecting the end user from unauthorized sources getting access to his account data, conversations and even gaining the control over the account.

The studies are going to be based around the documentations provided by the authors of the tested applications. Discord's whitepaper mentions multiple ways of protecting your account with MFA: passkeys and security codes being an additional form of verification based on the fingerprint or face scanning technology, a classic authenticator app that stores user's 2-Factor-Authentication codes to user's applications or SMS code sent to the user's provided phone number. There is also a security method such as a „backup code" that lets user regain access to their accounts if they lost it.

TeamSpeak 6 also offers an MFA as a code sent to the authenticator app on the mobile phone.

Messenger provides MFA in a form of security codes (like a fingerprint), using a mobile app storing authentication code to 2FA or receiving an SMS code on accounts with assigned mobile phone numbers. There is also possibility to generate up to 10 security codes to regain access to a lost account.

Microsoft Teams – as it's connected to a Microsoft account – gives the ability to a two-step verification after turning such option on the Microsoft account settings. Its scope ranges from an e-mail sent on a backup e-mail address, an SMS on a phone number or using an authentication app.

Zoom has an MFA feature like all the other applications but unlike the others there it's not possible for a regular, free account. The scope though embraces factors like an SMS code, authenticator app or business e-mail address.

Signal security is also providing the possibilities of using an authenticator app or the backup codes generated in the application.

WhatsApp on the other hand provides an option to add a PIN code and an e-mail address that lets their users reset the code which is an additional account security.

The summary of all the information about the additional security options concludes the research results with another table:

Table 7: Checking additional MFA protection methods

App	Additional MFA protection
Discord	Security codes, authenticator app, SMS code, backup code
TeamSpeak 6	Authenticator app
Messenger	Security codes, authenticator app, SMS code, backup code
Microsoft Teams	E-mail codes, SMS codes, authenticator app,
Zoom	For professional accounts: SMS code, authenticator app, business e-mail codes
Signal	Authenticator app, backup code
WhatsApp	PIN code, e-mail verification

Cryptography is one of the most important factors in a secure communication process as it's all about protecting the end user from getting their conversation data stolen. Instant messaging apps' security is based on the key exchanging mechanism, a common practice to enhance the security of user's conversations. Instant messengers tested in this study differ in this case so it's another factor to compare when it comes to the protection of the end user.

All applications with their designated key exchanging methods are presented in table 8.

Table 8: Checking key exchanging methods

App	Key exchanging methods
Discord	DHKEM with Curve25519
TeamSpeak 6	ECDH with Curve25519
Messenger	X3DH (in E2EE conversations)
Microsoft Teams	Through the TLS channel
Zoom	Between users' devices
Signal	X3DH
WhatsApp	X3DH

6. Conclusions

Analyzing the table of security test results, it is clear that applications are following the security standards regarding the confidentiality of messages and the inability to access its direct, unencrypted content from the perspective of a network packet sniffer. The research also showed obvious influence of the nature of communication via the instant messenger on the sizes of packets sent between users, which makes it easier, for person who is scanning internet traffic, to identify the communication type (chat, audio or video). E2EE is also the standard when it comes to instant messengers. However, the matter becomes more complicated from now on because not all instant messengers support E2EE in all communication cases. Signal, TeamSpeak 6 and WhatsApp strive for this standard by declaring full encryption.

TLS 1.3 is also a common standard although some applications also use the older TLS 1.2, and in some cases using both - each for different purpose.

Applications such as Discord and TeamSpeak do not share users' external IP addresses in their packets during connections, which distinguishes them from Messenger, Microsoft Teams and Zoom, that communicate in a peer-to-peer (P2P) form. In Signal and WhatsApp applications users' IP addresses are hidden provided that the option to hide addresses from other users is enabled in the application settings – then packets are sent through the application server.

It is worth mentioning that the domain names associated with a given application are also visible while using said applications. It allows someone monitoring network traffic to determine that someone is currently using said application.

TeamSpeak 6, Signal and WhatsApp are the applications that are providing the best security practices among all tested instant messengers – each one of them hides the content of the messages and IP addresses, uses TLS 1.3 and full E2EE. Signal also differs from them in the confidentiality of domain names in the packets. Together with regularly improved protocol, Signal is undoubtedly the favorite when it comes to user security in online messaging apps.

Referring to our hypothesis H1 it turns out that Signal Protocol and everything that surrounds the instant messaging app called Signal is actually the safest one in terms of the user security and gives the end-user everything to keep privacy during the communication with other application users. Hypothesis H2 is something the research also predicted to be the case and the aftermath of this confirms the definitive difference between Discord's security methods and the Signal's ones that are superior to any other applications, even the ones using their security protocols. Hypothesis H3 in case of the current research turned out not to be true what is caused by the choice of tested instant messaging applications because 4 out of 7 are immune or are giving users the abilities to secure their IP address and hide from the traffic eavesdropper.

References

[1] J. Son, Y. W. Kim, D. B. Oh, K. Kim, Forensic analysis of instant messengers: Decrypt Signal, Wickr, and Threema, Forensic Science International: Digital Investigation 40 (2022) 301347, https://doi.org/10.1016/j.fsidi.2022.301347.

[2] V. B. Liwandouw, A. D. Wowor, The existence of cryptography: A study on instant messaging, Procedia Computer Science 124 (2017) 721–727, https://doi.org/10.1016/j.procs.2017.12.210.

[3] B. Pribadi, S. Rosdiana, S. Arifin, Digital forensics on Facebook Messenger application in an Android smartphone based on NIST SP 800-101 R1 to reveal digital crime cases, Procedia Computer Science 216 (2023) 161–167, https://doi.org/10.1016/j.procs.2022.12.123.

[4] Y. Su, C.-H. Shih, T. Ou Yang, Investment fraud cases study in Chinese context of instant messaging software, Procedia Computer Science 246 (2024) 391–402, https://doi.org/10.1016/j.procs.2024.09.418.

[5] M. Warner, A. Strohmayer, M. Higgs, L. Coventry, A critical reflection on the use of toxicity detection algorithms in proactive content moderation systems, International Journal of Human-Computer Studies 198 (2025) 103468, https://doi.org/10.1016/j.ijhcs.2025.103468.

[6] T. Dargahi, A. Dehghantanha, M. Conti, Forensics analysis of Android mobile VoIP apps, In: K. R. Choo, A. Dehghantanha, (eds) Contemporary Digital Forensic Investigations of Cloud and Mobile Applications, Syngress (2017) 7–20, https://doi.org/10.1016/B978-0-12-805303-4.00002-2.

[7] C. Paar, J. Pelzl, T. Güneysu, Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms, 2nd ed., Springer, Berlin, 2024.

[8] R. Oppliger, SSL and TLS: Theory and Practice, 2nd ed., Artech House, Boston, 2016.

[9] H. Beguinet, C. Chevalier, T. Ricosset, H. Senet, Formal Verification of a Post-quantum Signal Protocol with Tamarin, Lecture Notes in Computer Science 14368 (2024) 105–121, https://doi.org/10.1007/978-3-031-49737-7_8.