

Development of a secure video chat based on the WebRTC standard for video conferencing

Elena Revyakina^{1*}

¹Don State Technical University (DSTU), Rostov-on-Don, Russian Federation

Abstract. The article describes the process of developing a secure video chat based on the WebRTC standard for establishing a connection and exchanging information between end users during a video conference. The analysis of a variety of technologies when creating a video chat was performed, theoretical information on the WebRTC standard was studied. As a result of the analytical review, the shortcomings of the WebRTC technology were revealed - disclosure of the private IP address of the system user. Therefore, an application architecture was proposed that solves this problem. Conventionally, the architecture consists of a number of entities - sending a message to the signal and STUN servers to obtain the necessary information for making a video call and a loop that, using an open socket connection, listens, transmits and reproduces audio and media streams between system participants. A software product was also written as part of this work, which allows you to use video chat without the risk of revealing a private IP address. This product was also compared with existing analogues and a number of features were identified. The developed product can be used for online conferences. To organize a broadcast, you only need to create a room and share a link with its participants. The advantage of this development is security because data transmission occurs only over HTTPS, media streams are encrypted (DTLS and SRTP), the technology does not require the installation of plug-ins. In addition, there is no need to install additional software, the service is cross-platform, supported by most browsers.

1 Introduction

With the development of information technology, global communications such as video chats have become available. Video chats are the transmission of a video stream received from a webcam. At first, such video chats were so-called photo chats: due to the low bandwidth of the channels, not a video stream was sent, but an array of images at certain intervals, which made it possible to observe the interlocutor quite well and was a significant breakthrough.

The first video chats worked using Flash technology. But this technology was vulnerable in terms of security. Therefore, there was a need to create a technology or standard that would describe all kinds of processes for establishing a secure connection. And WebRTC has become such a standard. WebRTC is a standard technology that allows you to establish a

* Corresponding author: revyelena@yandex.ru

connection between two or more clients, transfer both streaming video and audio data, as well as test messages and media data using a browser [1-3].

WebRTC opens up new ways of communication for organizing video conferences, online meetings and other events. The relevance of this work is due to the fact that at present video chats based on WebRTC technology are a new stage in the development of Internet communications. This standard allows you to imagine the browser as a complete video chat terminal, which gives an advantage over other technologies due to the fact that communication between video chat participants takes place in real time without installing any extensions and plug-ins [2].

2 Materials and methods

2.1 Analysis of the relevance of webRTC technology

WebRTC (Web Real Time Communications) is a standard that describes the transfer of streaming video data, audio data and content from the browser of one client to the browser of another in real time without installing plugins or other extensions. The standard provides the ability to turn the browser into a video conferencing terminal. To do this, you just need to open a web page to start chatting.

Data transfer is carried out using the SRTP (Secure Real-time Transport Protocol) protocol. WebRTC uses SRTP-DTLS for authentication, encryption, and message integrity, and to protect against replay attacks. This gives privacy by encrypting the RTP traffic and authentication. SRTP is one of the components for security, it is very convenient for developers who are looking for a reliable and secure API [4-9].

SRTP (Secure Real-time Transport Protocol) is a security system that extends the RTP (Real-time Transport Protocol) protocol with a set of security mechanisms. The protocol was published by the IETF (Internet Engineering Task Force) in RFC 3711, in March 2004.

SRTP provides privacy by encrypting the RTP payload without including RTP headers. It also supports authentication, which is widely used as a security mechanism in RTP. SRTP may not be used with the full set. WebRTC uses two audio codecs, G.711 and Opus, as well as VP8 and H.264 video codecs. In most cases, developers prefer to use Opus audio codec and VP8 video codec due to their characteristics [10-15].

2.2 Development of a software product

To develop a software product that implements the establishment of a video connection between end users, the JavaScript programming language was chosen, since it allows writing fairly fast and cross-platform applications.

As a framework for creating interfaces, a powerful tool was chosen - vue.js, which allows you to focus only on the application architecture and its logic during development, without worrying about how the application will work with the state.

The developed application should be atomic for the user. The user should not be aware of the business logic that processes his requests to the server and clients. After a user request, regardless of success, the application must respond with results. Also, the application must be fault-tolerant, since there is always the possibility that the user will provide incorrect data, perform actions in the wrong order, or crash while the program is running.

The application architecture is a private closed network. The center of this network is a VPN server, which also serves as a STUN server. Unlike a regular STUN server, which issues external IP addresses to application users, which in itself means that if an external IP address falls into the hands of an attacker, he can easily determine the user's internal IP address.

However, this server provides a session IP address for the duration of the connection for video communication [16-17].

The first client wants to make a call to the second client. WebRTC gives you all the information you need to identify yourself. But the question remains how one browser can find another, how to send this meta-information, how to initialize the call. The first client generates meta-information and sends it to the signaling server using web sockets or HTTP.

The second client takes it, uses it, installs it for itself, generates a response, and using the signaling mechanism sends it to the signaling server, which, in turn, relays it to the first client. Thus, both clients currently have all the necessary data and meta-information.

The developed server includes the implementation of the STUN server. STUN is a client/server protocol for providing the external IP address of a NAT router. When determining the external address of the user, the developed server adds an entry to the routing table, where each external IP address is assigned a local IP address of the private network. This address will only be valid during the video call [18].

Conventionally, the architecture consists of a number of entities - sending a message to the signaling and STUN servers to obtain the necessary information for making a video call and a loop that, using an open socket connection, listens, transmits and plays audio and media streams between system participants. Figure 1 shows the proposed application architecture, which solves one of the main drawbacks of the technology - the disclosure of the internal IP address.

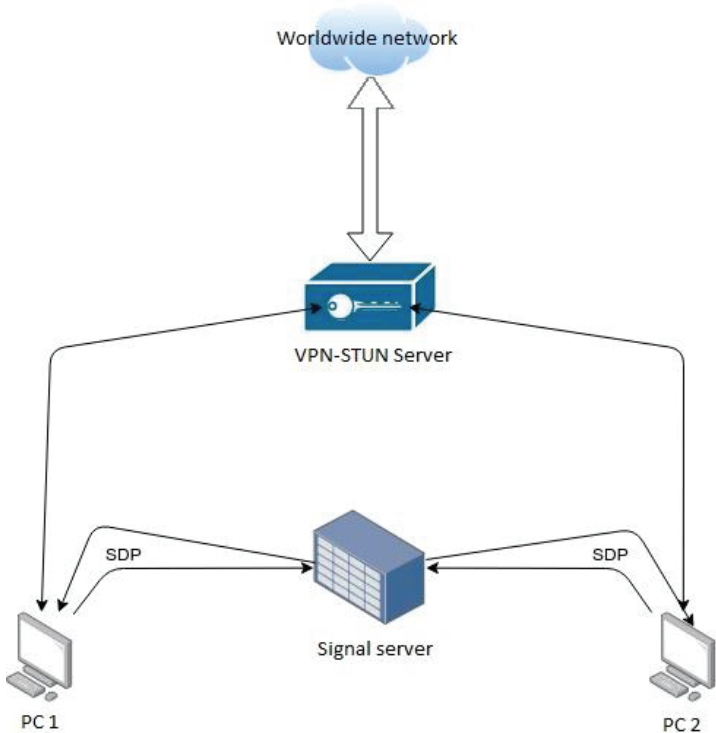


Fig. 1. Software product architecture (video chat).

2.3 Structure of software implementation

The main goal of implementing this software was to create a secure video chat based on the webRTC standard. The software implementation is a public web application and consists of two presentation parts: a form and a video container.

Conventionally, the program consists of a series of sequential actions - sending a message to the signaling and STUN servers to obtain the necessary information for making a video call and a loop that listens, transmits and plays audio and media streams between system participants using an open socket connection. The scheme of the program is shown in the figure. Figure 2 shows a block diagram of the software implementation.

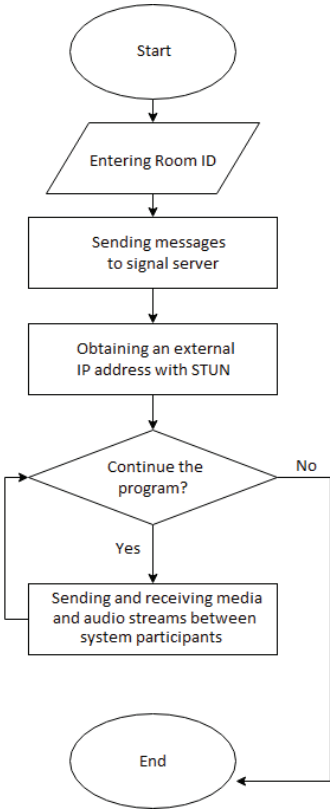


Fig. 2. Block diagram of the program.

Since the software tool is divided according to its functionality, all parts of the program are divided into certain modules [5].

The following describes the methods used to run the program.

getUserMedia(): Object - the method is used to get information about the user's device.

enumerateDevices(): Array - method to return a list of all media devices connected to the computer.

createOffer(): Object - the method is used to create an SDP object in order to start a new connection with a remote host.

setLocalDescription(sessionDescription: object): Promise - the method takes one parameter - the description of the sender's session and returns a promise that is fulfilled after the description is changed asynchronously.

setRemote Description(sessionDescription: object): Promise - the method takes one parameter - the description of the receiver's session and returns a promise that is executed after the description is changed asynchronously.

createAnswer(): Object - method creates an SDP response to an offer received from a remote participant during offer/answer negotiation for a WebRTC connection.

addStream(stream: Stream): Stream is a method for adding an audio and media stream.

onAddStream(): Stream - a method for subscribing to receive audio and media streams.

removeTrack(stream: Stream): void - The method instructs the other side of the connection to stop sending media from the specified track, without actually removing itself from the list of senders.

peerConnect(configuration: object): void - method for establishing a p2p connection.

closePeerConnection(): void - method to close the current peer connection.

2.4 Demonstration of the program

The HTML hypertext markup language and CSS style sheets provide ample opportunities for creating visual forms.

To use the application, users do not need to install any plugins or tools. Therefore, in order to start using the application, you must specify the room identifier, the input field for which is provided by the program interface

Obviously, the program interface will not look overloaded and is intuitive for the user who knows how and why the program works. Figure 3 shows the user form of the software solution.

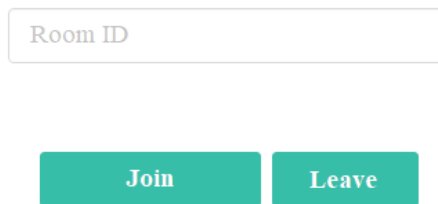
The image shows a simple web interface. At the top, there is a light gray rectangular box with the text 'Room ID' inside. Below this box, there are two teal-colored buttons. The left button has the word 'Join' in white text, and the right button has the word 'Leave' in white text.

Fig. 3. Program interface.

The interface consists of a field for entering the room ID and two buttons that allow you to either join the room or leave it. The room ID can be any value of the string type, which is preselected by the users.

After entering the identification number, the system participants press the “join” button, after which the connection becomes open. In order for a p2p connection to be established, it is necessary to wait for two or more participants to join the room.

In order to demonstrate the operation of the application and the creation of a conference, a computer with an Internet connection provided by one provider and a mobile phone with an Internet connection provided by another provider will be selected as client terminals. This way you can repeat the case when the participants of the videoconference are in different networks.

From a computer, you need to follow the application link. To join a call, you just need to enter the room ID number and click the "join" button. Figure 3 shows the interface of the program after connecting one client.

After the first participant has joined the room, the system is in a waiting state.

From the second device, you also need to follow the links and indicate the previous room identification number and join the room. After the second user has joined the room, a new container appears in the application window of the first user, into which the audio and video stream of another user is transmitted. Figure 4 shows the interface of the program after connecting the second client.

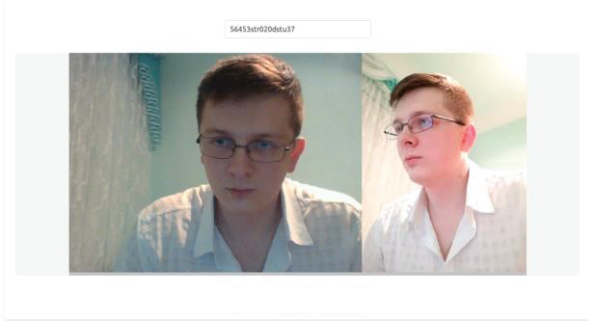


Fig. 4. Video conference between two participants.

After the second user has joined, another container is also displayed on the screen of the mobile device, which contains streaming video and audio data.

3 Comparison of the written application with analogues

To date, one of the most popular software products, the basis of which is written in accordance with the webRTC standard, is the Skype and Google Hangouts media services [6]. The table below shows the results of the study, which compared the software product written in the framework of this study and existing analogues. For ease of comparison, you should give the name to the application written as part of the work - "Video Chat" table 1

Table 1. Comparison of the application with popular services.

Criteria	Videoconference	Skype	Hangouts
Delay	0.2 sec.	1.4 sec.	1.05 sec.
Max. quality	720p	1080p	1080p
Max. Frame Rate	60 frame./sec.	60 frame./sec.	60 frame./sec.
Audio codec	OPUS	AVC	AVC
Video codec	VP8	H.264	H.264
The need for registration	No	Yes	Yes
The need to install software	No	Yes	No

Based on the data in the table, we can conclude that Skype and Google Hangouts are focusing on quality, while the delay reaches high values. Since these services are predominantly entertainment content, the delay is not the main factor for them.

To achieve the minimum delay in the written application, the quality is reduced to 720p, while Skype and Google Hangouts can afford to broadcast in 1080p quality. Another main advantage of the service is that video conference participants do not have to register, which gives them an advantage over the group of people who, for whatever reason, do not want to identify themselves. It is also worth noting that Twitch and Google Hangouts use the H.264

codec for video encoding, which is closed from free use due to the imposed patent law. The disadvantage of this codec is that it requires much more computing resources than the free-to-use analog VP8 [7].

Also, when developing this software product, it was noted that with an increase in the number of participants, the consumption of system resources, namely the system's random access memory (RAM), increases. For the test in each of the programs, conferences ranging in size from two to five participants were created. Figure 5 shows a graph of the dependence of system participants on performance resources.

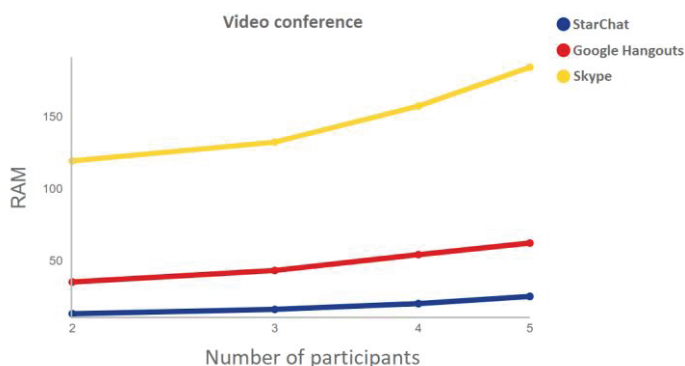


Fig. 5. Influence of the number of system participants on system resources.

The graph shows that with the addition of a new user to the conference, the consumption of system resources, in particular RAM, increases. Based on the graph, we can conclude that the application written as part of the study outperforms its counterparts in terms of system RAM consumption, which plays an important role when using the product.

4 Conclusion

In this work, an analytical review was carried out and the shortcomings of the webRTC technology were identified - the disclosure of the private IP address of the system user. Therefore, an application architecture was proposed that solves this problem. A software product was also written as part of this work, which allows you to use video chat without the risk of disclosing a private IP address. This product was also compared with already existing analogues and a number of features were identified.

The developed product can be used for online conferences. To organize a broadcast, you only need to create a room and share a link with its participants.

The advantage of this development is safety. data transfer occurs only via HTTPS, media streams are encrypted (DTLS and SRTP), the technology does not require the installation of plugins. In addition, there is no need to install additional software, the service is cross-platform, supported by most browsers.

References

1. G.A. Buzov, S.V. Kalinin, A.V. Kondratiev, Protection against information leakage through technical channels (Hotline-Telecom: Moscow, Russia, 2005)
2. *Temporary methodology for assessing the security of fixed technical means and systems ...: Regulatory and methodological document*. Collection of temporary

- methods for assessing the security of confidential information from leakage through technical channels (M.: State Technical Commission of Russia, 2002)
3. E. Kostyuchenko, I. Rakhmanenko, M. Lapina, *Evaluation of a method for measuring speech quality based on an authentication approach using a correlation criterion*. 17th International Conference on Intelligent Environments (IE), p. 1-7. IEEE (2021). DOI: 10.1109/IE51775.2021.9486435
 4. GOST R 51275-2006. Data protection. Informatization object. Factors affecting information. General provisions. Input. 02/01/2008 (M.: Standartinform, 2006)
 5. O. Lukmanova, et. al., *Simulation of the Passive Protection Device in the Acoustoelectric Leakage Channel*. IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), p. 2381-2384. IEEE (2021). DOI: 10.1109/ElConRus51938.2021.9396169
 6. S.L. Emelyanov, Information processing systems 3, 20–23 (2010)
 7. A.P. Zaitsev, A.A. Shelupanov, Technical means and methods of information protection (M.: Mashinostroenie, 2009)
 8. L. Cherkesova, et. al., *A System for Monitoring the Effective Operation of Employee Access to Computer Systems Based on Biometric Authentication*. Networked Control Systems for Connected and Automated Vehicles: vol. 1. Cham : Springer International Publishing, pp. 1683-1693 (2022) DOI: 10.1007/978-3-031-11058-0_171
 9. A.A. Titov, Engineering and technical protection of information: A textbook for students of the specialties "Organization and technology of information protection", "Integrated protection of informatization objects" and "Information security of telecommunication systems" (Tomsk: Tomsk. state University of Control Systems and Radioelectronics, 2010)
 10. D. Korochentsev, L. Cherkesova, P. Razumov, E3S Web of Conf. **224**, 01042 (2020). <https://doi.org/10.1051/e3sconf/202022401042>
 11. A.V. Ivanov, S.R. Salimov, J. of Phys.: Conf. Ser. **1791(1)**, 012046 (2021). DOI 10.1088/1742-6596/1791/1/012046
 12. A.A. Khorev, N.V. Tsarev, Vestnik VSU. Series: System Analysis and Information Technologies **1**, 57–67 (2017)
 13. Model of threats and a violator of the security of personal data processed in special information systems of personal data of the industry. Ministry of Telecom and Mass Communications of the Russian Federation. Moscow (2010) [Electronic resource]
 14. M. Pasha, F. Shahzad, A. Ahmad, ArXiv preprint arXiv:1701.09182 (2017). <https://doi.org/10.48550/arXiv.1701.09182>
 15. A. Zelensky, et. al., E3S Web of Conf. **371**, 01056 (2023). 10.1051/e3sconf/202337101057
 16. S. Karo-Karo, et. al., IOP Conf. Ser.: Mat. Sci. and Eng. **725(1)**, 012135 (2020). DOI 10.1088/1757-899X/725/1/012135
 17. S. Northcut, J. Novak, Detection of security breaches in networks. Third edition. (Translation from English: Williams Publishing House, 2003)
 18. R. Anderson, *Why information security is hard-an economic perspective*. Seventeenth Annual Computer Security Applications Conference. IEEE (2001)