Desktop application for crypto-protected voice communication

Stefan M. Ivanović[a], *Marko* R. Marković[b], *Sava* S. Stanišić[c], *Kristina* R. Živanović[d], *Dimitrije* S. Kolašinac[e]

[a] Serbian Armed Forces, Center for Applied Mathematics and Electronics, Department for Cryptography in Telecommunications, Belgrade, Republic of Serbia,
e-mail: stefanivanovic1012@gmail.com, **corresponding author**,
ORCID iD: https://orcid.org/0009-0000-1788-2095

[b] Khaoticen, Department for System Administration, Belgrade, Republic of Serbia,
e-mail: markomarkovicati@gmail.com,
ORCID iD: https://orcid.org/0009-0007-8040-6260

[c] Serbian Armed Forces, Air Force and Air Defence, 98th Air Force Brigade, Lađevci, Republic of Serbia,
e-mail: sava.stanisic@vs.rs,
ORCID iD: https://orcid.org/0009-0002-3118-0537

[d] Serbian Armed Forces, Center for Applied Mathematics and Electronics, Department for Cryptography in Computer Networks, Belgrade, Republic of Serbia,
e-mail: kristinazivanovicc@gmail.com,
ORCID iD: https://orcid.org/0009-0004-3648-3400

[e] Serbian Armed Forces, Center for Applied Mathematics and Electronics, Department for Cryptography in Computer Networks, Belgrade, Republic of Serbia,
e-mail: dimitrije.kolasinac@vs.rs,
ORCID iD: https://orcid.org/0009-0008-9505-2482

DOI: https://doi.org/10.5937/vojtehg72-48208

FIELD: computer sciences, telecommunications, IT, cryptography
ARTICLE TYPE: original scientific paper

Abstract:

Introduction/purpose: All data exchanged over the Internet as well as other computer networks should be considered exposed to various types of security threats. In light of this, the transmission of voice over applications that do not use any type of crypto-protection allows anyone to discern the content of communication. Since voice transmission requires as little delay as possible, various protocols are used to enable crypto protected real-time communication. This paper presents one solution in a desktop application variant.

Methods: The essence in voice exchange systems as well as in other systems where real-time communication is necessary is the establishment of a crypto-protected session which is a virtual secure channel for

communication to which only the communicating parties have access. Voice sessions in the application are established with the SIP (Session Initiation Protocol) protocol. The sessions are further protected using the ZRTP (Zimmerman Real-time Transport Protocol) protocol. FusionPBX was used as the SIP server (registrar) for testing purpouses. The application is developed in C++ language using the Qt framework.

Results: The final version of the application demostrates that ZRTP and SIP protocols are well suited for establishing crypto protected voice communications with low delay.

Conclusion: This solution provides cryptographic functions for data secrecy and the management of cryptographic keys. Improving the solution with digital signatures and certificates will result in additional cryptographic functions: data integrity and personal identification. With this improvement, this solution will be able to withstand modern security threats with low delay.

Key words: crypto protected sessions, real-time communication, voice sessions, SIP protocol, ZRTP protocol, communications.

Introduction

The quality and speed of information exchange has grown with the technological development of mankind. Having the right information at the right time puts an individual or a group at a significant advantage over other entities, whether they are business competitors or the enemy in military conflicts. The telephone, one of important inventions of the modern age, enabled the transmission of voice to remote locations, which made business processes easier and faster. The importance of this invention is reflected in the rapid expansion of the use of telephones in the business sphere, as well as the continuous construction of telephone lines. The development of computer networks was much faster and more far-reaching compared to the development of telephone lines due to the fact that they enabled not only the transmission of voice but also other types of data. All this resulted in the development of VoIP (Voice over IP) technology, which enabled voice transmission using the infrastructure of an already existing computer network. This technology made it possible to exchange voice using devices of different standards as well as computers.

As in any system where data is exchanged, it is necessary to protect the integrity and authenticity of the data. The first use of classic telephones in war led to the development of various eavesdropping techniques in order to gain an advantage over enemies. In order to protect themselves from enemy eavesdropping, armies developed various encryption principles. As the systems became more complex, the attack techniques also became more complicated and multiplied, which was accompanied

by the development of defense techniques. The development of computers drastically accelerated data processing, which led to a revolutionary change in the field of cryptography. Voice is nothing but a type of data that can be transmitted over a computer network. The protection of voice transmission is therefore not much different from the protection of any other type of data, except that it requires low latency like other forms of real-time communication. In order to achieve this, various software tools and libraries have been developed to work with cryptographic algorithms and data protection techniques. In combination with the existing communication protocols and modern frameworks for the development of applications for different platforms, modern intuitive but also cryptographically protected applications can be implemented. This paper presents a proposal for an application solution for the cryptoprotection of speech in an IP environment.

Protocol suite

In order to provide the desired services for crypto-protected voice exchange, the implementation proposed in this paper implies the use of the SIP (Session Initiation Protocol) protocol to establish a session for voice transmission. The voice in the test application is transmitted using the ZRTP (Zimmermann Real-Time Transport Protocol) protocol that encrypts the data symmetrically, using the AES algorithm with a key exchanged using the Diffie-Hellman algorithm (Abdallah & Meshoul, 2023).

SIP protocol

SIP (Session Initiation Protocol) is an application-level protocol used to control, establish and terminate sessions. Its most common application is establishing communication in IP telephony (VoIP systems).

After establishing a session between two "agents", as all types of end points in communication are called in SIP (they consist of a client agent which generates requests, and a server agent which generates responses to requests), further data transmission is taken over by one of protocols responsible for the transmission of a given media type (video or sound).

The most commonly used protocol for VoIP in combination with SIP is RTP (Realtime Transport Protocol) (Merit & Ouamri, 2012).

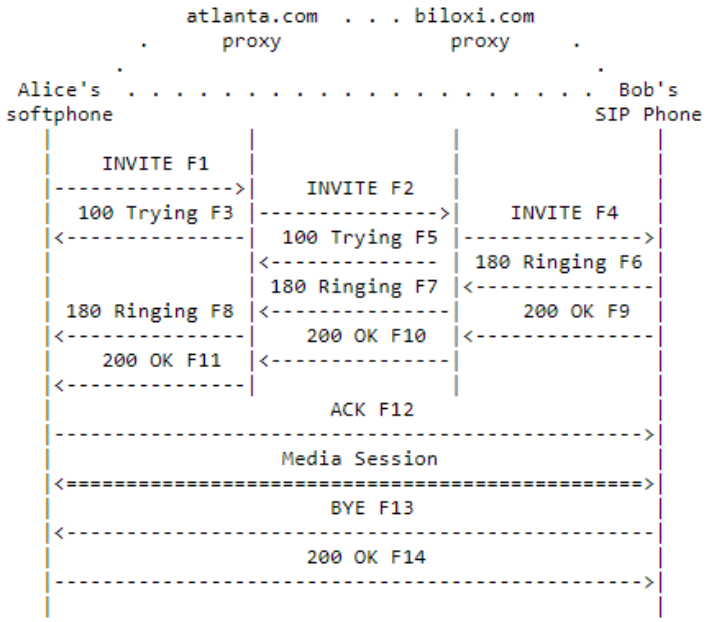

Figure 1 – An example of SIP messages exchange for establishing the session

The syntax of SIP is similar to other protocols that use a request/response system such as HTTP. SIP requests must start with a line containing the name of the method used, the URI to which the request is directed, the name, and the version of the protocol.

The basic methods supported by SIP requests are as follows:

– REGISTER - serves to log a certain agent to the server so that it is generally visible and available for establishing sessions,
– INVITE - a request to send an invitation to a specific address to establish a session.
– ACK - serves as a signal to the requested party that its confirmation has been received and that the session can begin,
– CANCEL – a request to cancel the session,
– BYE – a request to terminate the current session, and
– OPTIONS – a request sent to the SIP server to inquire about its capabilities.

Here is a example of an SIP message with the INVITE method:

```
INVITE sip:bj@bjsip.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bj@bjsip.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

In the SIP response, the first line represents the status line which contains the status code. A status code is a three-digit integer value that indicates the result of an attempt by the requested party to understand and respond to the request. Similar to HTTP, the first digit of the status code divides the codes into classes. Unlike the HTTP protocol which has 5 code classes, the SIP protocol has an additional 6th class which represents Global failures statuses (Boruchinkin, 2015).

ZRTP protocol

ZRTP (Zimmerman Real-time Transport Protocol) is a protected version of the RTP (Real-time Transport Protocol) protocol, which is a transport layer protocol intended for real-time communication, usually visual and voice communication. It uses the Diffie-Hellman algorithm for key exchange, which builds on SRTP (Secure Real-time Transport Protocol) which uses the AES algorithm to maintain an encrypted session (Boruchinkin, 2015). The protocol implements the negotiation of security parameters through small time windows where participants agree on algorithms and cryptographic settings. During this process, a Short Authentication String (SAS) code is generated and can be used to verify the user's identity (Qi et al, 2018).

ZRTP is used to protect against a variety of attacks, including Man-in-the-Middle attacks, where someone tries to intercept and manipulate communications. During parameter exchange, users can compare the

SAS code over a secure channel, such as face-to-face with a contact or through another communication channel. Data is encrypted using a symmetric cryptosystem, such as AES, which provides efficient and strong encryption. There is also the possibility of using user keys for additional security. Taking into account these technical aspects, ZRTP contributes to the creation of secure and confidential communication channels on the Internet, ensuring data integrity and privacy during information exchange (Eltengy, 2021).

Development tools

The test application was implemented in the C++ programming language using the Qt framework, specifically the QtQuick approach for developing graphical interfaces. QtCreator was used as the development environment. The basic principles of object-oriented programming as well as the functionalities provided by the programming languages C++, JavaScript and QML were used.

Qt framework

Qt is a cross-platform, open-source software development framework that enables the creation of applications for a wide variety of operating systems. It is developed by the Qt Company and is one of the most popular GUI development frameworks in the world of programming. The advantages of using Qt are: independence from the platform, having various tools for developing graphical user interfaces and a large number of classes for working with files, databases, network, multimedia, etc. Qt has its own integrated environment for developing applications called Qt Creator. Qt Creator offers a wide choice of types of projects when developing applications, where there are two main types for applications with a graphical user interface: Qt Widgets and Qt Quick. Other project types are variations of these two types (Mondal & Sharma, 2019).

The test application was developed using QT Quick approach to create graphical interfaces. QT Quick is a newer approach to creating user interfaces in QT, based on declarative graphical programming (QML) and imperative graphical programming (JavaScript). QML (Qt Modeling Language) is a declarative language used in the Qt Quick environment for creating user interfaces. QML allows the declarative definition of the user interface using simple scripts, which describe the appearance and behavior of elements. QML supports a component approach which means that reusable components can be created and embedded in different parts of the user interface making it easier to structure and maintain the code.

This makes UI development faster and more efficient, as it requires fewer lines of code compared to QT Widgets (Krasnowski & Lebrun, 2022).

Each user-defined component is defined in separate files. The syntax of the QML language resembles JSON (JavaScript Object Notation), where objects are graphic components, and object key-value pairs represent the attributes of that graphic object. In the example, the flexibility of the QML language can be observed, because attribute values can be bound to the values of other attributes or to the values of JavaScript expressions. Comparing QML with HTML, it is found that QML is more readable and more flexible, it reduces code repetition with a component approach and offers wider possibilities for customization of components (Mondal & Sharma, 2019). Below is an example of a user-defined QML component from a test application, specifically a user-defined button:

```
import QtQuick
import "/scripts/Utils/changeBrightness.js" as ColorJs
Rectangle {
    property var onClick
    property alias text: textBox.text
    property string textColor: theme.textColor
    property string col
    height: 30
    radius: height / 4
    scale: mouseArea.containsMouse ? 1.02 : 1
    color: mouseArea.containsMouse ? ColorJs.changeBrightness(col, -0.05) :
col
    Text{
        //warning
        font.pointSize: Math.ceil(height * 0.7)
        id: textBox
        color: textColor
        font.family: globalFont
        anchors.centerIn: parent
    }
    MouseArea{
        id: mouseArea
        anchors.fill: parent
        cursorShape: Qt.PointingHandCursor
        onClicked: () =>{
            onClick()
        }
        hoverEnabled: true
    }
}
```

The main idea of the Qt Quick approach in the development of graphic interfaces is to define graphic components declaratively through QML, changing the appearance of graphic components imperatively through the JavaScript language, while more complex programming logic is performed in C++. This development pattern resembles the MVC (Model View Controller) pattern with the fact that only the more complex controller logic is implemented in the C++ programming language. In this way, the components of the MVC pattern are sufficiently separated as entities, while Qt provides several APIs to easily connect MVC components (Kara et al, 2023).

PJSIP library

PJSIP is a free, open-source multimedia communication library written in C that implements standard protocols such as SIP, SDP, RTP, STUN, TURN, and ICE. It combines the Signaling Protocol (SIP) with a rich multimedia framework and NAT traversal functionality in a high-level API that is portable and suitable for almost all types of systems ranging from desktop computers through dedicated systems to mobile phones. PJSIP is both compact and feature-rich. It supports audio, video, presence subscription and instant messaging and has extensive documentation. On mobile devices, it abstracts system-dependent functions and in many cases can use the device's native multimedia capabilities. The library makes it possible to implement softphone applications of very small size (up to 150KB) using lower-level APIs, while using higher-level functionality, applications can be "packaged" into footprints of only a few hundred kilobytes (Ntantogian et al, 2019).

In order to facilitate access and work with the library, higher-level programming interfaces have been implemented. The first API is the PJSUA high-level API for the C language that defines types, structures, macros, enumerations, and functions to facilitate operation and calling of library procedures. The next level of abstraction is PJSUA2, an object-oriented API that makes it even easier to call the PJSIP library from object-oriented languages. It was initially implemented for the C++ programming language, but it can be used through various interfaces and called from other higher programming languages such as C#, Python and Java (Merit & Ouamri, 2012).

The PJSUA2 API is implemented through various classes that represent the abstraction of various functions of the base library. The classes through which access to the main functionalities of PJSIP is enabled are:

– Endpoint - represents an abstraction of the endpoint in SIP communication; it is necessary to create exactly one instance of this class in each application because further initialization and settings of the library are performed through,
– Media - abstract base class that represents the media element of the system that can record/play media content; this class is inherited by classes like the AudioMedia class which in turn is inherited by the AudioMediaRecorder and AudioMediaRecorder classes,
– Call - serves to abstract the establishment of calls or other sessions; also most often subclasses are derived from it in order to redefine callback methods for reacting to changes in the status of the initiated call or the arrival of session instant messages,
– Buddy - represents a remote "friend" or contact whose status (present/not registered) can be subscribed to by the client; out-of-session instant messages can be sent through this object.

Application design and structure

Class diagram

The program logic of the application is placed in the C++ code of the application where the code is organized into several classes. In the class diagram (Figure 2), it can be seen that the classes BjEndpoint, BjAccount and BjCall inherit from the classes Endpoint, Account and Call respectively, which are classes from the PJSUA2 library. These classes enable basic functionalities related to VoIP communication. Through them, the user registers and logs out of the SIP server, makes a call, accepts a call, rejects a call and receives a status about the reason for the end of the call. The class that represents the core of the program logic is BjSip, which inherits from the QObject class, which is the basic class from the Qt framework and whose purpose is to enable the graphical representation of an object of a given class. The BjSip class has multiple purposes, but its primary purpose is to provide a link between other C++ classes and the graphical interface, that is, QML. Qt offers a number of APIs to bridge the gap between C++ and QML, the most notable of which is the Signals and Slots API, which also underlies all other APIs (Eltengy, 2021). Signals and slots are actually ordinary functions that are related by broadcasting; calling the signal function automatically calls the slot function. In this way, the functions from QML are connected with the functions from C++, in the concrete example with the methods of the BjSip class (Ivanović, 2023).

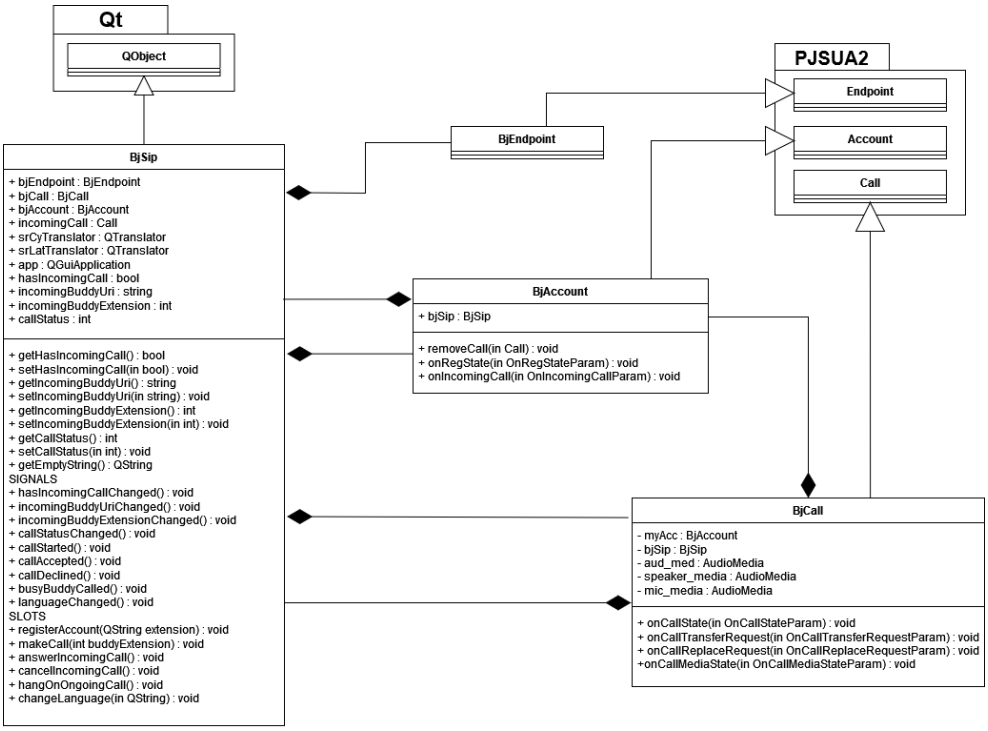

Figure 2 – Test application – Class diagram

Application activities

Registration activity

When starting the application, it is necessary for the user to log in to the predefined SIP server. To log in, it is necessary to first create a local account with a user name, password and extension (the number with which the user is identified on the SIP server), which can be done from the activity of creating a new user account.

After the account is created, the user can log in with a username and password. After entering the parameters for registration on the server, the user starts the process of registering the account on the server by pressing the "Log in" button.

Main activity

The main activity of the application is divided into two tabs: "call history" and "contacts". The "call history" tab is not implemented in this version of the application. On the "contacts" tab, there is a list of contacts created by the user, where each contact has an extension (a number on the SIP server) and a contact name. Next to each contact there are buttons for voice and video calls, although a video call is not implemented in this version of the application. There is also a button for creating a new contact on the "contacts" tab, which forwards the user to the form for entering a new user.

Figure 3 shows the main activity of the application and the "contacts" tab. The user can add new contacts by pressing the "Add new contact" button, which opens the form for adding a new contact (Figure 4).

On the form, the user should enter the extension of the new contact and the name of the contact, before pressing the "Add new contact" button, which adds the new contact to the contact list.

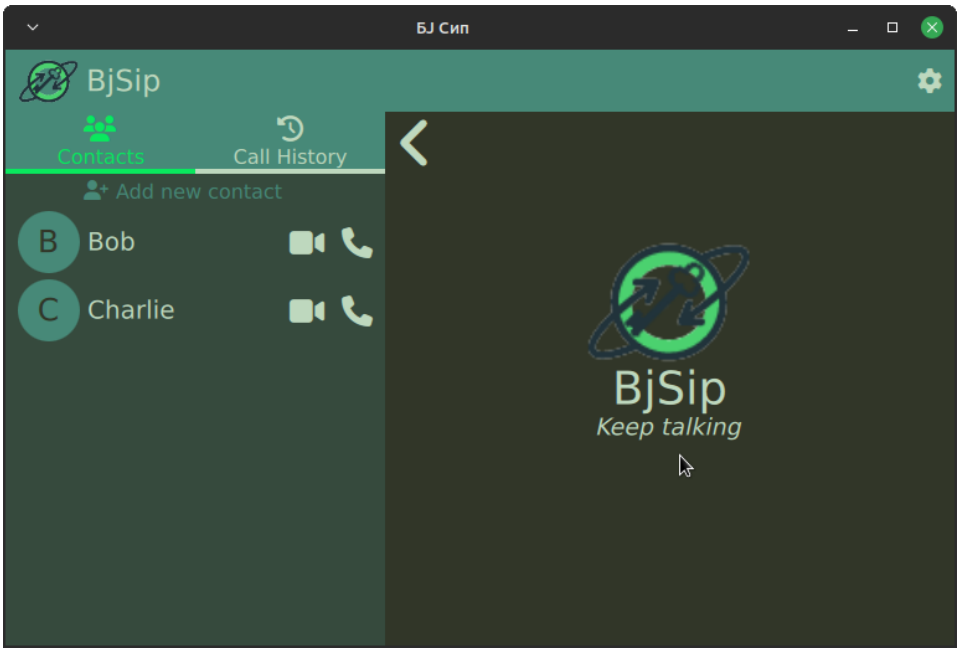

Figure 3 – Test application - Main activity in the dark mode in English

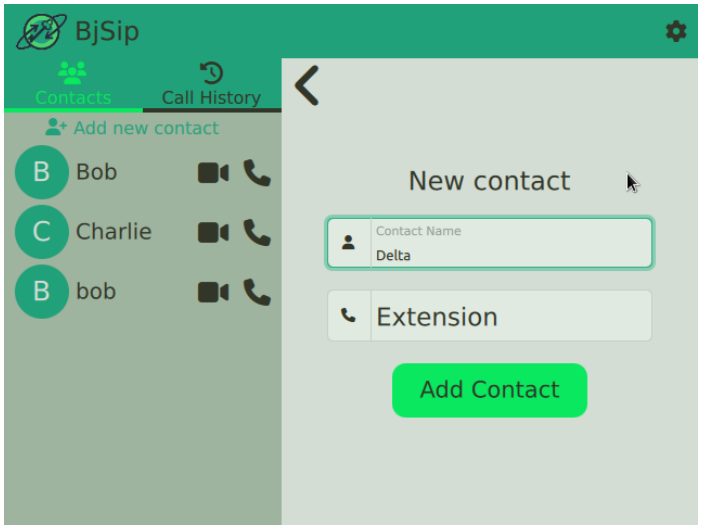

Figure 4 – Test application – Add the new contact activity in English

Call activities

In the current version of the application, only voice calls between two users are enabled. Pressing the voice call button in the contact list sends a call to the desired contact, where the user is forwarded to the call establishment activity (Figure 5). On a given activity, the user has the option to end the started call or to wait for the call to be established.

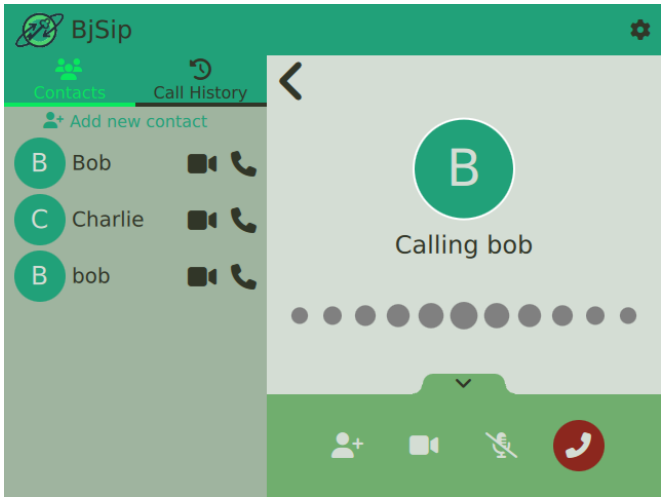

Figure 5 – Test application – Pending call activity in English

If the contact is unavailable for any reason, the user is forwarded to the end-of-call activity shown in Figure 6, where the reason for the termination of the initiated call is printed. On a given activity, the user has a button to restart the same call, a button to return to the initial activity, as well as the written name of the contact with whom he participated in the call and the reason for ending the call.

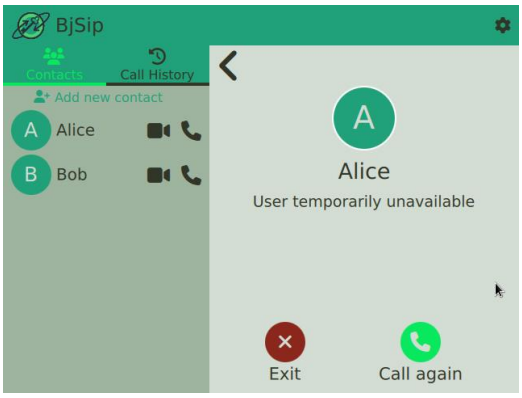

Figure 6 – Test application – Call ended activity in English

In case the called contact accepts the call, both users are forwarded to the voice call activity (Figure 7). On a given activity, users have buttons to add a new participant, switch to a video call, mute the microphone, and end the call. In the current version, only the button to end the call is implemented. An established call continues until one party ends the call by pressing the end call button, which forwards both users to the end-of-call activity (Figure 6).

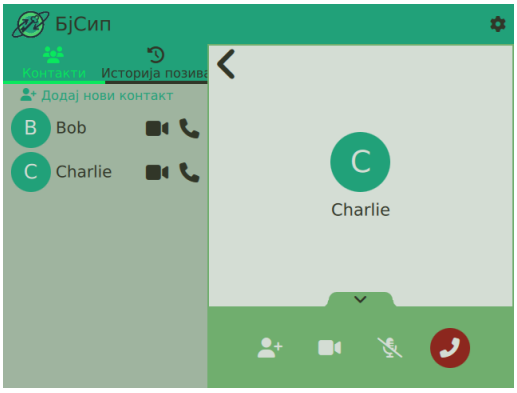

Figure 7 – Test application – Outgoing call activity in Serbian

If the server forwards an incoming call to the application, the user is shown a pop-up menu (Figure 8) in the lower right corner of the application window with the information about the user who made the call and the buttons for accepting and rejecting the call. If the application detects an incoming call while a call is in progress at that moment, it refuses to give the call by passing the status to the caller that the user is busy, which is displayed on the end-of-call activity (Figure 6).

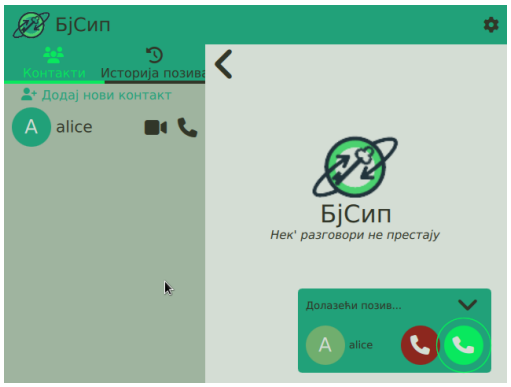

Figure 8 – Test application – Incoming call popup in Serbian

Settings activity

There is a settings activity in the application (Figure 9) that currently has two settings that are there to improve the user experience. One setting allows the selection of the color mode between light and dark mode, while the other setting allows the selection of the language. Currently, the application supports Serbian (Latin and Cyrillic) and English. Figure 9 shows the appearance of the main activity of the application in the light color mode in Serbian.

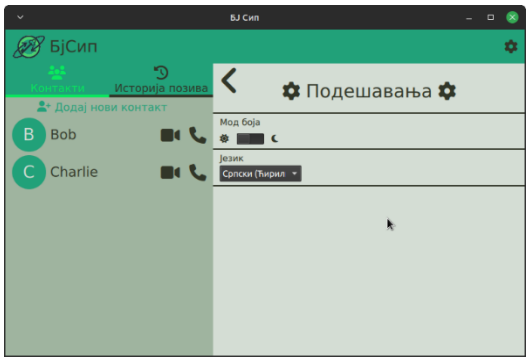

Figure 9 – Test application – Settings activity in the light mode in Serbian

Test environment

In order to test the functioning of the application, it was necessary to create a test environment in which there is a SIP server to which the test application connects. The network environment is simulated by a router with Wi-Fi access points through which a laptop device running three client processes and a virtual machine running a server are connected to a wireless LAN network. The FusionPBX SIP server is running on a Linux Debian 11 virtual machine, with its network settings set to Bridge, so that the server can be accessed from other physical machines on the network. FusionPBX can be configured via web interface (Figure 10). If the application functions as intended in this environment, it can be concluded that it will function in any network that operates according to TCP/IP network protocols (Tot et al, 2021).

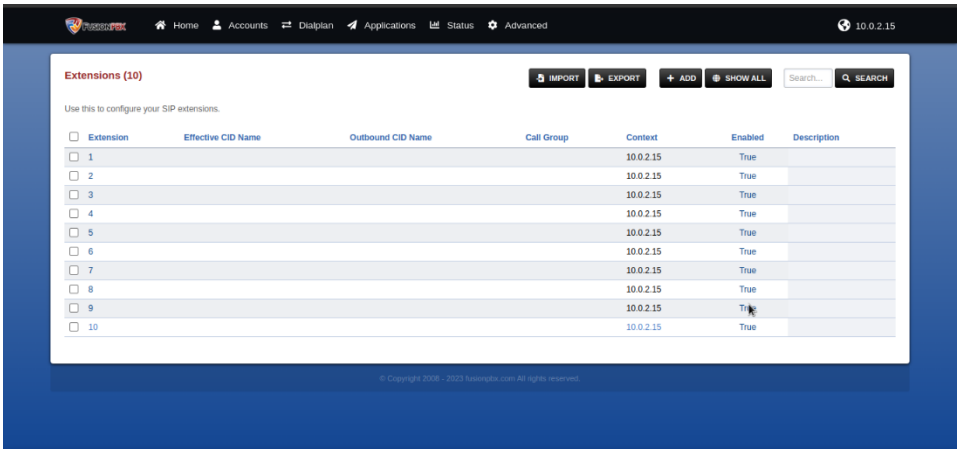

Figure 10 – Test environment – Web interface for FusionPBX

Conclusion

Exchanging any type of content over unprotected media is practically unacceptable in today's world. With the advancement of technology, the demands for faster and better communication have increased, which must also be protected due to today's security challenges. Through the integration of cryptographic protocols such as the ZRTP protocol, the application enables the creation of secure communication channels and the exchange of secret keys between the users. This provides encryption and authentication, contributing to the impenetrability of unwanted activities. Using the SIP protocol of the application level, the way of mutual

communication between users and servers is standardized. The development of an application for cryptographic protection of speech is of great importance in the context of a growing need for secure communications both in business and private spheres as well as in the military. Challenges such as compatibility with different devices and operating systems, and compliance with the existing privacy standards and regulations need to be addressed.

The level of protection as well as the range of information aspects that are protected in the proposed solution are not sufficient for applications in real systems. In order to further improve the security and protection of messages, it is necessary to explore the possibilities of using additional cryptographic mechanisms, such as digital signatures, random session keys, and digital envelopes for the implementation of additional security services. This would preserve the integrity of the data and the authenticity of the data source as well as the sender's non-repudiation, and the secrecy of the data itself would be more clearly protected.

The application test shows that the SIP protocol is a suitable and good solution for this type of application. In order for this application to be more comprehensive, it is necessary to expand it in the ways mentioned in the previous chapter and test it in a larger and more complex environment.

References

Abdallah, H.A. & Meshoul, S. 2023. A Multilayered Audio Signal Encryption Approach for Secure Voice Communication. *Electronics*, 12(1), art.number:2. Available at: https://doi.org/10.3390/electronics12010002.

Boruchinkin, A.Yu. 2015. Secure voice communication system with hardware encryption of data on hands-free headset. In: *SIN '15: Proceedings of the 8th International Conference on Security of Information and Networks*, Sochi, Russian Federation, pp.76-79, September 8-10. Available at: https://doi.org/10.1145/2799979.2800030.

Eltengy, A.H. 2021. Encryption Of Voice Calls Using CryptoBin Algorithm. In: *2021 International Telecommunications Conference (ITC-Egypt)*, Alexandria, Egypt, pp.1-5, July 13-15. Available at: https://doi.org/10.1109/ITC-Egypt52936.2021.9513963.

Ivanović, S. 2023. *Razvoj aplikacije za kriptozaštitu govora u IP okruženju*. BS thesis. Belgrade, Serbia: University of Defence (in Serbian).

Kara, M., Merzeh, H.R.J., Aydın, M.A. & Balık, H.H. 2023. VoIPChain: A decentralized identity authentication in Voice over IP using Blockchain. *Computer Communications*, 198, pp.247-261. Available at: https://doi.org/10.1016/j.comcom.2022.11.019.

Krasnowski, P. & Lebrun, J. 2022. Exchanging Keys with Authentication nd Identity Protection for Secure Voice Communication without Side-channel. *arXiv:2211.07186.* Available at: https://doi.org/10.48550/arXiv.2211.07186.

Merit, K. & Ouamri, A. 2012. Securing Speech in GSM Networks using DES with Random Permutation and Inversion Algorithm. *International Journal of Distributed and Parallel Systems (IJDPS)*, 3(4), pp.157-164, Available at: https://doi.org/10.5121/ijdps.2012.3416.

Mondal, S. & Sharma, R.K. 2019. Application of Advanced Encryption Standard on Real Time Secured Voice Communication using FPGA. In: *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT),* Kanpur, India, pp.1-6, July 06-08. Available at: https://doi.org/10.1109/ICCCNT45670.2019.8944857.

Ntantogian, C., Veroni, E., Karopoulos, G. & Xenakis, C. 2019. A survey of voice and communication protection solutions against wiretapping. *Computers & Electrical Engineering*, 77, pp.163-178. Available at: https://doi.org/10.1016/j.compeleceng.2019.05.008.

Qi, D., Longmei, N. & Jinfu, X. 2018. A Speech Privacy Protection Method Based on Sound Masking and Speech Corpus. *Procedia Computer Science*, 131, pp.1269-1274. Available at: https://doi.org/10.1016/j.procs.2018.04.342.

Tot, I., Trikoš, M., Bajčetić, J., Lalović, K. & Bogićević, D. 2021. Software Platform for Learning about Brain Wave Acquisition and Analysis. *Acta Polytechnica Hungarica*, 18(3), pp.147-162. Available at: https://doi.org/10.12700/APH.18.3.2021.3.8.

Aplicación de escritorio para comunicación de voz criptoprotegida

Stefan M. Ivanović[a], *Marko* R. Marković[b], *Sava* S. Stanišić[c], *Kristina* R. Živanović[d], *Dimitrije* S. Kolašinac[d]

[a] Fuerzas Armadas de Serbia, Centro de Matemáticas Aplicadas y Electrónica, Departamento de Criptografía en Telecomunicaciones, Belgrado, República de Serbia, **autor de correspondencia**

[b] Khaoticen, Departamento de Administración de Sistemas, Belgrado, República de Serbia

[c] Fuerzas Armadas de Serbia, Fuerza Aérea y Defensa Aérea, 98.a Brigada de la Fuerza Aérea, Lađevci, República de Serbia

[d] Fuerzas Armadas de Serbia, Centro de Matemáticas Aplicadas y Electrónica, Departamento de Criptografía en Redes Informáticas, Belgrado, República de Serbia,

CAMPO: ciencias de la computación, telecomunicaciones, TI, criptografía
TIPO DE ARTÍCULO: artículo científico original

Resumen:

Introducción/objetivo: Todos los datos intercambiados a través de Internet, así como de otras redes informáticas, deben considerarse expuestos a

diversos tipos de amenazas a la seguridad. En vista de esto, la transmisión de aplicaciones de voz en off que no utilizan ningún tipo de criptoprotección permite que cualquiera pueda discernir el contenido de la comunicación. Dado que la transmisión de voz requiere el menor retraso posible, se utilizan varios protocolos para permitir la comunicación en tiempo real criptoprotegida. Este artículo presenta una solución en una variante de aplicación de escritorio.

Métodos: La esencia en los sistemas de intercambio de voz, así como en otros sistemas donde es necesaria la comunicación en tiempo real, es el establecimiento de una sesión criptoprotegida que es un canal virtual seguro para la comunicación al que sólo tienen acceso las partes que se comunican. Las sesiones de voz en la aplicación se establecen con el protocolo SIP (Session Initiation Protocol). Las sesiones están además protegidas mediante el protocolo ZRTP (Protocolo de transporte en tiempo real de Zimmerman). FusionPBX se utilizó como servidor SIP (registrador) para fines de prueba. La aplicación está desarrollada en lenguaje C++ utilizando el framework Qt.

Resultados: La versión final de la aplicación demuestra que los protocolos ZRTP y SIP son muy adecuados para establecer comunicaciones de voz criptoprotegidas con bajo retardo.

Conclusión: Esta solución proporciona funciones criptográficas para la secrecía de los datos y gestión de claves criptográficas. Mejorar la solución con firmas y certificados digitales dará como resultado funciones criptográficas adicionales: integridad de datos e identificación personal. Con esta mejora, esta solución podrá resistir las amenazas de seguridad modernas con poca demora.

Palabras claves: sesiones criptoprotegidas, comunicación en tiempo real, sesiones de voz, protocolo SIP, protocolo ZRTP, comunicaciones.

Десктопное приложение для криптозащиты голосовой связи

Стефан М. Иванович[а], *Марко Р*. Маркович[б], *Савва* С. Станишич[в], *Кристина* Р. Живанович[г], *Дмитрий* С. Колашинац[г]

[а] Вооруженные силы Республики Сербия, центр прикладной математики и электроники, департамент криптографии в телекоммуникациях, г. Белград, Республика Сербия, **корреспондент**

[б] «Khaoticen», отдел системного администрирования, г. Белград, Республика Сербия

[в] Вооруженные силы Республики Сербия, Военная авиация и противовоздушная оборона, 98-ая авиационная бригада, Ладжевци, Республика Сербия

[г] Вооруженные силы Республики Сербия, центр прикладной математики и электроники, департамент криптографии в компьютерных сетях, г. Белград, Республика Сербия

РУБРИКА ГРНТИ: 20.15.05 Информационные службы, сети, системы в целом,
28.21.19 Теория кодирования,
49.33.35 Надежность сетей связи и защита информации,
81.93.29 Информационная безопасность. Защита информации
ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: Безопасность всех данных, которыми обмениваются через интернет, а также другие компьютерные сети подвергаются различным видам угроз. Таким образом, передача голоса через приложения, которые не используют криптозащиту, обеспечивает возможность любому желающему перехватить голосовое сообщение. Поскольку передача голоса должна осуществляться с минимальной задержкой, используются различные протоколы для обеспечения криптозащиты связи в режиме реального времени. В данной статье представлено решение в виде десктопного приложения.

Методы: Суть систем голосового обмена, а также других систем, где необходима связь в режиме реального времени, заключается в установлении криптозащищенного сеанса. Криптозащищенный сеанс представляет собой виртуальный защищенный канал связи, доступ к которому имеют только общающиеся стороны. Голосовые сеансы в приложении устанавливаются с помощью протокола SIP (Session Initiation Protocol). Сеансы дополнительно защищены с помощью протокола ZRTP (Zimmerman Real-time Transport Protocol). FusionPBX использовался в целях тестирования SIP-сервера (регистратор). Приложение разработано на языке C++ с использованием фреймворка Qt.

Результаты: Окончательная версия приложения демонстрирует, что протоколы ZRTP и SIP обеспечивают криптозащищенную голосовую связь с минимальной задержкой.

Выводы: Данное решение предоставляет криптографические функции при обеспечении конфиденциальности данных и управления криптографическими ключами. Усовершенствование решения с помощью цифровых подписей и сертификатов приведет к появлению дополнительных криптографических функций: целостности данных и идентификации личности. Благодаря такому усовершенствованию данное решение сможет противостоять современным угрозам безопасности с минимальной задержкой.

Ключевые слова: криптозащищенные сеансы, связь в реальном времени, голосовые сеансы, протокол SIP, протокол ZRTP, коммуникации.

Десктоп апликација за размену криптозаштићеног говора

Стефан М. Ивановић[а], *Марко* Р. Марковић[б], *Сава* С. Станишић[в], *Кристина* Р. Живановић[г], *Димитрије* С. Колашинац[г]

[а] Војска Србије, Центар за примењену математику и електронику, Департман за криптографију у комуникацијама, Београд, Република Србија, **аутор за преписку**

[б] Khaoticen, Департман за системску администрацију, Београд, Република Србија

[в] Војска Србије, Ратно ваздухопловство и противваздухопловна одбрана, 98. ваздухопловна бригада, Лађевци, Република Србија

[г] Војска Србије, Центар за примењену математику и електронику, Департман за криптографију у рачунарским мрежама, Београд, Република Србија

ОБЛАСТ: рачунарске науке, телекомуникације, ИТ, криптографија
КАТЕГОРИЈА (ТИП) ЧЛАНКА: оригинални научни рад

Сажетак:

Увод/циљ: Сви подаци размењени преко интернета, као и преко других рачунарских мрежа, подложни су разним типовима безбедносних претњи. Узимајући то у обзир, пренос гласа уз коришћење апликација које не користе никакав вид криптозаштите омогућава било коме да пресретне садржај комуникације. Како пренос гласа захтева што мање кашњење, користе се разни протоколи за криптозаштиту за комуникацију у реалном времену. Овај рад представља решење у варијанти десктоп апликације.

Методе: Суштина размене гласа, као и комуникације у реалном времену, јесте у обезбеђењу криптозаштићене сесије, која представља виртуелни заштићени канал за комуникацију коме могу приступити само легални учесници. Гласовне сесије успостављене су у апликацију коришћењем SIP (Session Initiation Protocol) протокола. Сесије су заштићене коришћењем ZRTP (Zimmerman Real-time Transport Protocol) протокола. FusionPBX је коришћен као SIP сервер за тестирање. Апликација је развијена у C++ језику коришћењем Qt фрејмворка.

Резултати: Коначна верзија апликације демонстрирала је да ZRTP и SIP протоколи обезбеђују криптозаштићену гласовну комуникацију са минимумом кашњења.

Закључак: Ово решење обезбеђује криптографске функције за тајност података и размену криптографских кључева.

Унапређивање решења, коришћењем дигиталних потписа и сертификата, резултирало би додатним криптографским функцијама: интегритетом података и аутентикацијом. Овим унапређењем решење би било отпорно на модерне претње уз мало кашњење.

Кључне речи: криптозаштићене сесије, комуникација у реалном времену, гласовне сесије, SIP протокол, ZRTP протокол, комуникације.

Paper received on: 11.10.2023.
Manuscript corrections submitted on: 03.03.2024.
Paper accepted for publishing on: 04.03.2024.