УДК 004.056

О. М. Козак

ЗАСІБ ЗАХИЩЕНОГО АУДІО ТА ВІДЕО ЗВ'ЯЗКУ

Вінницький національний технічний університет

Анотація. *В даній доповіді проведено аналіз різних реалізацій програмних засобів захищеного аудіо та відео зв'язку для операційних систем Android та iOS. Запропоновано та розроблено власний варіант засобу для встановлення захищеного аудіо та відео зв'язку. За допомогою мови програмування Dart та фреймворку Flutter було реалізовано користувацький інтерфейс із підтримкою таких операційних системи, як Android та iOS. Для обміну користувацькими даними, які необхідні для встановлення зв'язку було розроблено сервер на платформі Node.js з використанням фреймворку Express та бібліотеки socket.io. Основний протокол, який було використано для передавання медіа даних без третьої сторони – WebRTC.*

Ключові слова: *захищений аудіо та відео зв'язок, алгоритми передачі медіа даних, WebRTC, Dart, Flutter, Node.js.*

Abstract. *Various software implementations of secure audio and video communication for Android and iOS operating systems are analyzed in this report. We proposed and developed version of the tool for establishing secure audio and video communication. Using the Dart programming language and the Flutter framework, a user interface was implemented with a support of such operating systems as Android and iOS. A server was developed using Node.js platform utilizing Express framework and the socket.io library to exchange user data, which is necessary to establish a connection. The main protocol that has been used to transfer media data without a third party is WebRTC.*

Keywords: *secure audio and video communication, media data transfer algorithms, WebRTC, Dart, Flutter, Node.js.*

Вступ

У XXI столітті в багатьох сферах діяльності та в повсякденному житті, спілкування за допомогою інтернету стало набагато зручнішим за комунікацію в форматі живого спілкування, як наприклад розмова з близькими, що знаходяться в іншій країні чи бесіда між працівниками компанії на різних поверхах хмарочосу. Проте дане рішення призвело до виникнення інших проблем, а саме до втрати конфіденційності даних або ж втрати фактору приватного спілкування.

Наразі розроблено чимало програмного забезпечення для встановлення аудіо та відео зв'язку , які пропонують різні приклади захисту даних. Серед багатьох сучасних засобів виділяють загальну проблему, яка полягає у передаванні та можливому зберіганні даних на серверах, що створює вірогідність витоку даних у глобальну мережу, копіювання на носії інформації, тощо [1]. Тому необхідно удосконалити метод передавання інформації між користувачами та використати один з відомих та перевірених часом алгоритмів шифрування даних.

Метою даної роботи є покращення методів забезпечення конфіденційності інформації в програмних засобах для встановлення захищеного аудіо та відео зв'язку, шляхом розробки засобу, що зменшує вірогідність витоку даних через вищевказану проблему, через використання варіанту архітектурної системи Peer-to-peer(P2P). Завдяки даному рішенню, сервер для зберігання даних, які передаються між користувачами не потрібний, але потрібний сервер, який буде реалізовувати механізм встановлення зв'язку між користувачами, тобто Interactive Connectivity Establishment (ICE) сервер.

Результати дослідження

Сучасний ринок програмного забезпечення пропонує широкий спектр рішень для захищеного аудіо та відео зв'язку. Кожен програмний продукт має свої особливості, методи захисту та набір функціональних можливостей. Ці програмні засоби можна класифікувати за методами захисту, використовуваними технологіями та способом реалізації аудіо та відео зв'язку. Одним з найважливіших аспектів захищеного зв'язку є шифрування даних за допомогою різних алгоритмів, які можуть бути вбудовані в протоколи, які використовуються для передачі медіа даних. Шифрування перетворює дані в нечитабельний формат, який може бути розшифрований лише авторизованими користувачами.

Для передавання аудіо та відео даних використовуються різні протоколи. Деякі з найпоширеніших протоколів включають:

- Real-time Transport Protocol (RTP) [2] використовується для передачі аудіо та відео даних в реальному часі;
- Real-time Transport Control Protocol (RTCP) [3] використовується для контролю та моніторингу RTP-сеансів;
- Secure Real-time Transport Protocol (SRTP) [4] використовується для шифрування RTP-даних.

Зазвичай ці протоколи працюють разом, для забезпечення більш стійкого зв'язку під час передавання аудіо та відео даних.

Програмний засіб, який було розроблено, реалізує поставлену мету, а саме покращує захист конфіденційності даних, які передаються під час встановленого аудіо або відео зв'язку шляхом виключення непотрібної ланки, а саме сервера, який використовується для прийняття та надсилання даних від одного користувача до іншого. В основі покращеного методу лежить встановлення Peer-to-Peer зв'язку між користувачами, що забезпечує впевненість в тому, що дані обробляються тільки авторизованими пристроями користувачів, що здійснюють спілкування.

Для реалізації було використано мову програмування Dart [5], оскільки найбільш необхідний параметр для застосунку такого виду, це швидкодія. Обрана мова програмування має велику перевагу у вигляді строгої типізації, що забезпечує надійне, швидке та безпечне виконання програмного застосунку, також підтримує динамічну компіляцію перед виконанням та компілюється у більш ефективний машинний код, що дозволяє збільшити швидкодію. Оскільки, ще одна характеристика, яка має бути втілена у застосунку це кросплатформеність, необхідно обрати фреймворк Flutter [6], за допомогою якого, можна розробити додаток для таких операційних систем, як Android, iOS та Windows, при цьому, розроблений застосунок не буде сильно поступатись у швидкодії тим, які були розроблені за допомогою нативних мов програмування. Обраний стек технологій забезпечить велику швидкодію розробленого застосунку, та буде займати менше об'єму дискового простору на пристрої користувача. Для передачі потоку медіа даних між користувачами було обрано протокол WebRTC [7], який був розроблений компанією Google. Було обрано саме його, оскільки це відносно новий протокол, який надає змогу реалізувати адаптивний стрімінг у режимі реального часу, також надає можливість встановити з'єднання безпосередньо між співрозмовниками. Схема передавання медіа-потоків за допомогою WebRTC проілюстровано на рисунку 1.

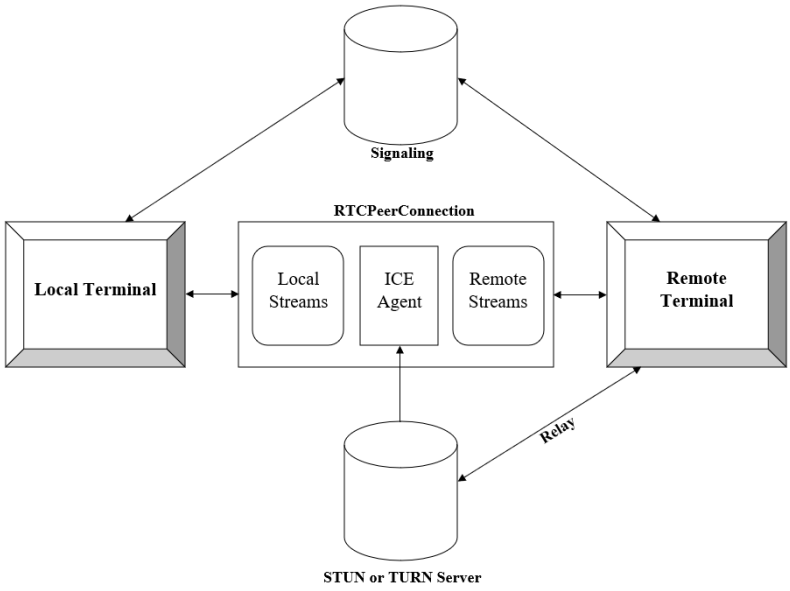

Рисунок 1 – Схема передавання медіа потоків WebRTC

На основі розробленої структури модуля захищеного зв'язку, було розроблено алгоритм роботи даного модуля. Початок виконання даного модуля передбачає формування SDP-пакету користувачем, який виступає ініціатором встановлення зв'язку. Після цього, відбувається надсилання сформованих даних іншому користувачу за допомогою сигнального серверу. Після цього починається процес формування аудіо- та відео-потоків для передачі медіа даних віддаленому користувачу. Під час цього,

віддалений користувач отримав повідомлення від сигнального серверу про ініціювання зв’язку та почав виконувати ідентичні дії, а саме формування SDP-пакету, надсилання його на сигнальний сервер та формування потоків для передачі медіа даних. Після підключення користувачів до однієї кімнати та обміну SDP-пакетів формується P2P тип з’єднання. Далі виконується генерація та розподіл секретних ключів між користувачами за допомогою протоколу Diffie-Hellman. Після отримання секретного ключа, користувачі шифрують медіа дані та надсилають їх іншим користувачам. При отримані аудіо- та відео-потоків відбувається розшифрування та виведення користувачу віддалених медіа даних. Схема розробленого алгоритму модуля захищеного зв’язку наведено на рисунку 2.

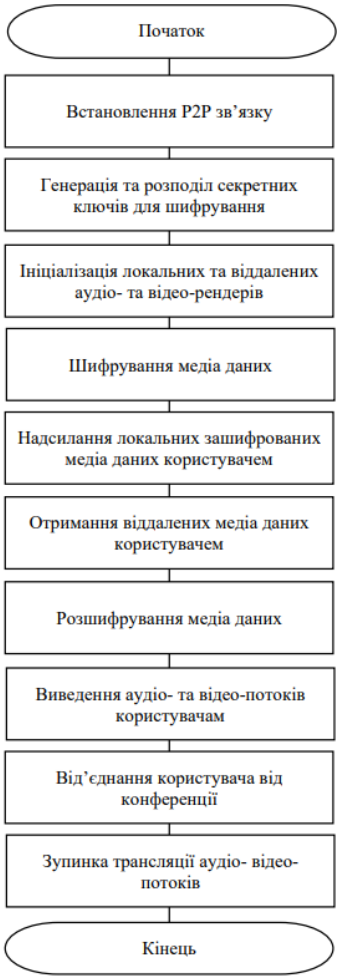

Рисунок 2 – Схема алгоритму роботи модуля захищеного зв’язку

На основі розробленого алгоритму роботи модуля захищеного зв’язку, було розроблено програмний застосунок для захищеної відео- та аудіо-комунікації. Робота розробленого засобу починається з екрану для введення ідентифікаційного номеру користувача та підключення до сигнального серверу на основі введеного номеру. При успішному підключенні до сигнального серверу, користувач переходить на наступний екран, який відповідає за введення ідентифікатора користувача, з яким необхідно встановити зв’язок, цей екран ідентичний до попереднього, основна відмінність це назва поля для вводу та назва кнопки. Також, на цьому екрані з’являється додаткове вікно, яке відповідає за відбій або прийняття вхідного дзвінка від віддаленого користувача, при цьому можна ідентифікувати користувача, від якого надходить запит за допомогою номеру, який також з’являється на цьому вікні. Наступним етапом роботи програми є екран очікування під’єднання віддаленого користувача, на цьому екрані можна побачити передпоказ свого відео-потоку, якщо користувач надав дозвіл на його передачу. Також, на цьому екрані можна вимкнути або ввімкнути передачу аудіо- або відео-потоку даних, або ж відхилити виклик. Наступний екран може з’явитись тільки при вдалому встановлені зв’язку із віддаленим користувачем, результатом роботи цього етапу є отримання та виведення на екран даних від віддаленого користувача, а також показ власного відео-потоку, у вигляді маленького прямокутника. Інтерфейс розробленого застосунку проілюстровано на рисунку 3.

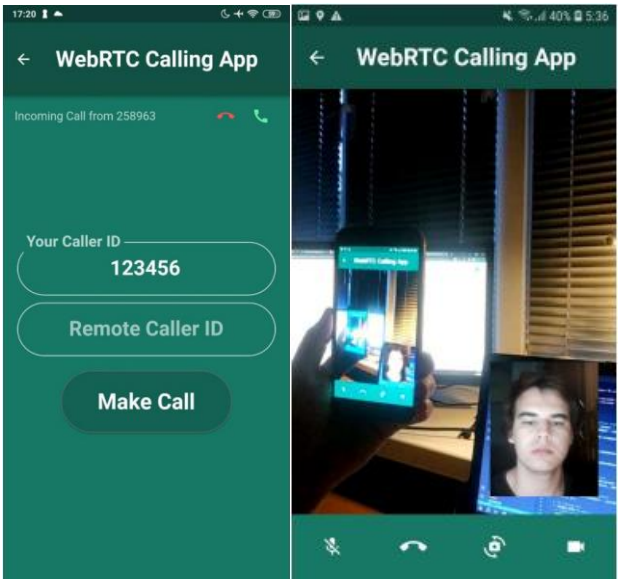

Рисунок 3 – Інтерфейс застосунку

Отже, розроблений програмний застосунок реалізує основну перевагу перед відомими аналогами подібних застосунків, а саме відсутність третьої сторони для передачі конфіденційної інформації у вигляді аудіо- та відео-потоків даних. Також, даний застосунок має перевагу у тому, що не використовує особисту інформацію користувача, оскільки зв'язок здійснюється за вільним ідентифікатором, який може генеруватись відповідно до певного алгоритму та бути унікальним для кожного користувача.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Most Popular Messaging Apps Worldwide 2023. URL: https://www.similarweb.com/blog/research/market-research/worldwide-messaging-apps/ (accessed: 14.03.2024);
2. Introduction to the Real-time Transport Protocol (RTP). URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Intro_to_RTP (accessed: 14.03.2024);
3. RTCP (RTP Control Protocol). URL: https://developer.mozilla.org/en-US/docs/Glossary/RTCP (accessed: 14.03.2024);
4. RTP (Real-time Transport Protocol) and SRTP (Secure RTP). URL: https://developer.mozilla.org/en-US/docs/Glossary/RTP (accessed: 14.03.2024);
5. Dart programming language. URL: https://dart.dev/ (accessed: 14.03.2024);
6. Flutter – Build apps for any screen. URL: https://flutter.dev/ (accessed: 14.03.2024);
7. Introduction to WebRTC protocols. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Protocols (accessed: 14.03.2024).

Козак Олександр Михайлович — студент групи 1БС-23М, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: sashakozak073@gmail.com.

Науковий керівник: Баришев Юрій Володимирович — к. т. н., доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, Україна.

Oleksandr Kozak — student of group 1BS-23M, , Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sashakozak073@gmail.com.

Scientific supervisor: Yurii Baryshev — Ph.D., Associate Professor of the Information Protection Department, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: yuriy.baryshev@vntu.edu.ua.