АНАЛІЗ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ, ЩО ЗНАХОДЯТЬ ВИКОРИСТАННЯ У СУЧАСНИХ МЕСЕНДЖЕРАХ

Білокуров О.О., Майба М.А., Шлома О.К., Дробяз М.О.

Кафедра «Інфокомунікаційної інженерії ім. В.В. Поповського»,
Харківський національний університет радіоелектроніки, Україна

E-mail: belokurovalex70@gmail.com,
mykola.maiba@nure.ua,
mykhailo.drobiaz@nure.ua,
oleksandr.shloma@nure.ua

Abstract

Currently, mobile technologies have become an integral part of everyday life. With their development, specialized instant messaging services have become available, gradually replacing standard short text messaging services. Unlike simple SMS, messengers use the public Internet. Most users choose messenger for simplicity and ease of use. In addition, as often happens, users think less about security issues. In general, they do not mention this at all when choosing and using messengers. Therefore, the paper analyzes the information protection methods used in modern messengers.

Наразі мобільні технології стали невід'ємною частиною повсякденного життя. З їх розвитком нам стали доступні спеціалізовані послуги для миттєвого обміну повідомленнями, які поступово витісняють стандартні послуги обміну короткими текстовими повідомленнями. На відміну від простих SMS месенджери використовують загальнодоступний Інтернет. Більшість користувачів вибирають месенджер з погляду простоти та зручності використання. Зазвичай про питання безпеки користувачі замислюються меншою мірою. А в загальній масі зовсім не згадують про це під час вибору та використання месенджерів.

У даній роботі будуть розглянуті методи захисту інформації, які використовують сучасні популярні месенджери. Під месенджером розумітимемо систему для обміну повідомленнями в режимі реального часу через Інтернет, голосовий та відеозв'язок, обміну файлами, організації групових відеоконференцій (групових чатів), та відповідне програмне забезпечення (програми, мобільні застосунки або вебсервіси). Найбільшу популярність у даний час отримали месенджери WhatsApp, Viber, Skype, Telegram, ICQ, Facebook Messenger, Hangouts (що є стандартним сервісом Google). При цьому в середньому кожен абонент користується трьома різними системами обміну повідомлень [1].

Для захисту переданих даних месенджери зазвичай використовують E2E (кінцеве або наскрізне) шифрування. Підтримка наскрізного шифрування гарантує, що тільки відправник та отримувач зможуть розшифрувати та прочитати інформацію. E2E вважається основним атрибутом будь-якого месенджера, що позиціонує себе безпечним. Це означає, що криптографічні ключі генеруються та зберігаються на кінцевому пристрої (пристрої користувача), а не на серверах миттєвих повідомлень. Тому ніхто, крім отримувача, навіть сервер системи, не зможе прочитати вміст зашифрованих повідомлень. З іншого боку листування буде не доступне і самому абоненту, якщо він перейде на інший пристрій. Тому синхронізація пристроїв або відновлення у разі втрати пристрою (тобто отримання доступу до архіву листування) спільно з використанням кінцевого шифрування неможливо без депонування особистого ключа поза пристроєм.

Оскільки кінцеве шифрування реалізується на верхніх рівнях мережної архітектури, адресні дані мають бути доступні в незашифрованому вигляді у проміжних вузлах (тобто серверах системи). Це означає, що метадані комунікацій користувачів (хто кому дзвонив чи писав, та коли) залишаються відкритими.

Багато месенджерів у даний час реалізують кінцеве шифрування на основі протоколу Signal, розробленого некомерційною організацією для однойменного месенджера. Протокол ретельно документований [2] та має бібліотеки, що його реалізують, з відкритим вихідним кодом мовами Java, C і JavaScript. На сьогоднішній день протокол Signal використовується месенджерами WhatsApp, Google Allo, Facebook Messenger та Skype. Реалізація шифрування Viber використовує концепцію протоколу Signal, але інші криптографічні алгоритми.

Протокол використовує криптографію з відкритим ключем на еліптичній кривій Curve25519 або Curve448 для цифрових підписів, узгодження ключів на основі модифікації протоколу Діффі-Хеллмана (DH), шифрування повідомлень за допомогою симетричного алгоритму AES-256 та перевірку цілісності за допомогою коду автентифікації. Крім того, для зміни ключів шифрування протягом сеансу зв'язку використовується функція диверсифікації ключа (KDF) на основі HMAC-SHA256 та HMAC-SHA512 (HKDF). Всі криптографічні примітиви, що використовуються протоколом, добре відомі та рекомендовані для застосування в системах захисту інформації. Безпека протоколу проаналізована у [3, 5-7].

Основні особливості протоколу Signal обумовлені тим, що другий бік комунікацій може бути недоступний (перебувати в автономному режимі, офлайн) в момент відправлення повідомлення. Тому стандартні протоколи автентифікації та обміну ключами (Authenticated Key Exchange, AKE) не можуть бути застосовані безпосередньо.

Для вирішення проблеми автономності однієї зі сторін Signal реалізує асинхронний протокол передачі Діффі-Хеллмана (X3DH), вимагаючи попереднього надсилання на проміжний сервер партії заздалегідь обчислених значень (відкритих ключів). Таке відправлення здійснюється під час реєстрації або пізніше (рис. 1). Коли абонент бажає надіслати повідомлення, він отримує необхідні для виконання AKE-подібного протоколу значення отримувача з проміжного сервера (який діє лише як буфер) і обчислює ключ шифрування повідомлення.

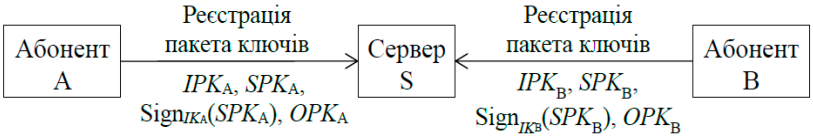

Рис. 1. Перший етап протоколу Signal - реєстрація ключової інформації користувачів

На етапі реєстрації до початку обміну повідомленнями кожен абонент формує три типи ключів асиметричної схеми: довготривалий ключ IK (для підпису), середньостроковий попередній ключ SK та набір одноразових попередніх ключів OK. При цьому на сервер надсилаються лише відповідні їм відкриті ключі IPK, SPK, OPK та підпис SignIK(SPK) – середньостроковий відкритий ключ SPK, що підписується довготривалим ключем IK абонента.

Абонент може періодично (наприклад, раз на тиждень або раз на місяць) оновлювати свій середньостроковий відкритий ключ SPK з підписом, а також у будь-який час завантажувати новий набір попередніх одноразових відкритих ключів OPK. Довготривалий ключ IPK є ідентифікаційним та реєструється абонентом одноразово. Нехай абонент A хоче розпочати сеанс зв'язку з абонентом B (рис. 2). Абонент A запитує сервер і отримує пакет ключів абонента B, а потім обчислює секретний ключ SK з декількох значень протоколу DH, отриманих на основі значень ідентифікаційних ключів обох абонентів IPKA, IPKB, середньострокового відкритого ключа отримувача SPKB та короткострокового відкритого ключа EPKA зі знову згенерованої відправником пари ключів асиметричного шифрування.

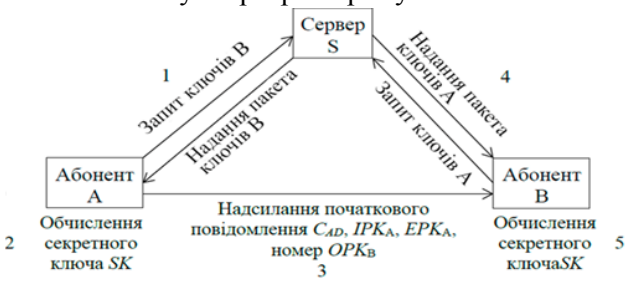

Рис. 2. Другий етап протоколу Signal – встановлення сеансу зв'язку

Ключ SK є основою для формування ланцюжка ключів шифрування повідомлень. Коли ключ шифрування створено, абонент A шифрує симетричним алгоритмом своє ідентифікаційне повідомлення AD, що складається зі значень

ідентифікаційних ключів відправника та отримувача, а також будь-якої додаткової ідентифікаційної інформації – імен абонентів, сертифікатів тощо.

Потім А надсилає отримувачу В криптограму CAD свій ідентифікаційний ключ IPKA, короткостроковий відкритий ключ EPKA та інформацію про те, який з попередніх ключів OPKB був використаний для отримання ключа шифрування повідомлення.

Отримавши всю інформацію, абонент *B* зможе сформувати ключ *SK* для розшифрування повідомлення. Якщо повідомлення розшифроване вдало та ідентифікаційна інформація є коректною, *B* продовжує використовувати ланцюжок ключів, отриманих зі *SK*, для шифрування свого повідомлення. Використані одноразові ключі *OPKB* видаляються сервером, а відповідні OKB – абонентом *B*.

Отже, у роботі виконано аналіз способу захисту інформації у разі використання месенджеру, розглянуто метод захисту інформації, який застосовується у найпоширеніших месенджерах. Виявлено основні переваги та недоліки цього протоколу захисту каналу зв'язку:

1. До переваг можна віднести відкритість вихідного коду, що дозволяє ретельно вивчити архітектуру Signal і переконатися в тому, що програма не має вразливостей та потенційно шкідливих функцій (наприклад, прихованого стеження за діями користувачів або збирання їх метаданих). Відкритий вихідний код програми для обміну миттєвими повідомленнями дозволяє здійснювати комплексний аудит безпеки. Розробники та аналітики, експерти з кібербезпеки можуть зробити складання програми, дослідити її функціональність та привернути увагу до слабких місць і вразливостей як у серверній, так і клієнтській частинах коду.

2. З іншого боку, вільний доступ до коду підвищує ризик того, що інформація щодо виявленої вразливості може використовуватися зловмисником. Отже, відкритість коду не може гарантувати безпеку даних користувача, але є важливим атрибутом її побудови. Переважна більшість незалежних аудиторів зацікавлені в еволюції надійності та безпеки коду месенджера, а цього можна досягти лише спільними зусиллями.

3. До недоліків протоколу можна віднести те, що взаємодія абонентів з сервером S та один з одним відбувається у недовіреному середовищі, тобто можлива реалізація загрози «людина посередині» (Man in the Middle, MITM), що дозволяє зловмиснику підміняти ключі під час передачі та видавати себе за будь-яку зі сторін комунікації. Зазвичай ця проблема вирішується за допомогою цифрової сертифікації та розгортання інфраструктури відкритих ключів (Public Key Infrastructure, PKI). Проте протоколом Signal вони не описуються, а проміжний сервер не має ролі центру сертифікації. Захист від реалізації атак MITM покликаний забезпечити використання на нижньому рівні захищеного транспортного протоколу (SSL/TLS), але на практиці його реалізації в мобільних додатках найчастіше вразливі до цього типу атак [4].

Література

1. Прес-релізи та звіти - ЯКІ МОБІЛЬНІ ДОДАТКИ Є НАЙБІЛЬШ ПОПУЛЯРНИМИ? *Домашня сторінка КМІС*. URL: https://www.kiis.com.ua/?lang=ukr&cat=reports&id=1072&page=1 (дата звернення: 09.11.2022).

2. Documentation. *Signal Messenger*. URL: https://signal.org/docs/ (дата звернення: 09.11.2022).

3. *Cohn-Gordon K., Cremers C., Dowling B., Garratt L., Stebila D.* A Formal Security Analysis of the Signal Messaging Protocol. Journal of Cryptology. 2020. Vol. 33, No. 4. P. 1914-1983. DOI: https://doi.org/10.1007/s00145-020-09360-1

4. *Кожухов Д.* Уязвимости SSL и TLS-протокола в небраузерном софте. *SPY-SOFT.NET*. URL: https://spy-soft.net/ssl-tls-vulnerabilities/ (дата звернення: 09.11.2022).

5. *Jain V., Sahu D.R., Singh Tomar D.* An Approach to Identify Vulnerable Features of Instant Messenger. 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP). 2020. P. 71-80. DOI: https://doi.org/10.1109/ISEA-ISAP49340.2020.235003

6. *Jain K., Ananth A., Honnavalli P.* Vulnerability Analysis of a Signal-based Messenger. 2021 IEEE Bombay Section Signature Conference (IBSSC). 2021. P. 1-6. DOI: https://doi.org/10.1109/IBSSC53889.2021.9673482

7. *Krishnapriya S., Priyanka V.S., Kumar S.S.* Forensic Extraction and Analysis of Signal Application in Android Phones. 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS). 2021. P. 1-6. DOI: https://doi.org/10.1109/FABS52071.2021.9702702