НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання кваліфікаційного дослідження

ДОСЛІДЖЕННЯ СИСТЕМ ЗАХИСТУ ЗАХИЩЕНИХ МЕСЕНЖЕРІВ ТИПУ TELEGRAM, VIBER, WHATSAPP, SIGNAL

Виконали студенти
групи ФІ-32мн
Баєвський Констянтин,
Шифрін Денис,
Кріпака Ілля

Київ — 2025

1 ДОСЛІДЖЕННЯ СИСТЕМ ЗАХИСТУ ЗАХИЩЕНИХ МЕСЕНДЖЕРІВ ТИПУ TELEGRAM, VIBER, WHATSAPP, SIGNAL

Telegram, Signal, WhatsAPP, Viber є одними з найпопулярніших месенджерів у світі, якими користуються мільйони людей. Як і кількість особистої інформації та кількість користувачів продовжує зростати щодня. Все більше людей проводять конфіденційні розмови в месенджерах, сподіваючись, що ця інформація залишиться приватною і недоступною для сторонніх осіб. Захищені месенджери допомагають запобігти кіберзлочинам, таким як шпигунство, крадіжка особистої інформації, фішинг або викрадення облікових даних. Тож напевне треба знати які месенджери варто використовувати. У даному досліджені розглянемо популярні месенджери та їх механізми захисту інформації.

1.1 Telegram

Telegram - хмарна, крос-платформена соціальна мережа, що має можливість швидким обміном повідомлень. Він дозволяє користувачам обмінюватися повідомленнями, ділитися файлами, проводити приватні групові, а також публічні трансляції. На даний час він став одним із найбільш завантажуваним додатком та має понад 500 мільйонів активних користувачів станом на 2021 рік.

Для того, щоб зрозуміти, чому Telegram не є E2E зашифрованим за замовчуванням треба згадати, що основою частиною функціоналу месенджера є:

– Канали (one-to-many), які поширюють інформацію на загал. Канали у свою чергу можуть:

– бути публічними чи приватними;

– мати необмежену кількість учасників;

– мати адміністраторів, що надсилають повідомлення всім, хто підписався.

Якщо узагальнити, то кожен може бачити повідомлення, а канал, по суті, діє як стрічка публікацій від адміністраторів. До прикладу, на каналі «Мілітарний» станом на кінець 2024 року є 63 тисячі підписників.

– Групові чати, що підтримують до 200 тисяч користувачів, які працюють як у більшості інших месенджерів.

– Особисті *хмарні* чати, у яких є компанія може показувати та синхронізувати

повідомлення в програмах для комп'ютера та смартфона в реальному часі. Це також означає, що повідомлення, які надсилаються, зберігаються на її серверах – компанія каже, що повідомлення в хмарних чатах «теоретично» доступні.

– Особисті *секретні* чати, що використовують E2E шифрування та не мають можливості синхронізації між пристроями, адже доступні лише на мобільних пристроях.

– Особисті голосові та відео дзвінки, що використовують протоколи які мають деяку схожість із секретними чатами. Вони не мають функції запису дзвінків, лише комунікація у реальному часі.

Спочатку поговорімо про **хмарні чати**.

MTProto v1 — це спеціальний мобільний протокол, розроблений командою Telegram [1], що був розкритикований численними експертами з криптографії. Організація вирішила не йти уже протоптаним шляхом та зробила свій стек протоколів для обміну повідомленнями за допомогою своєї комбінації шифрування. Додамо, що на даний час цей протокол не використовується, але був у використанні до 2021 року.

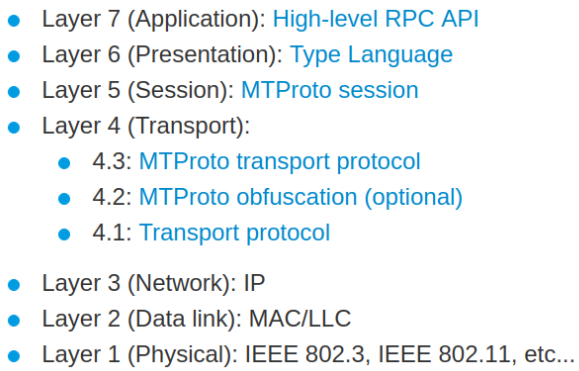


Рисунок 1.1 – Схема рівнів для MTProto v1.

Зауваження. Як кажуть у суспільстві: «Don't roll over your crypto», але Telegram зробив по іншому.

У даній роботі було наведено посилання [2] на роботи криптографів, що будували різні атаки на даний протокол. Цікаво, що у одній із них MTProto було визнано не є IND-CCA безпечним. Додамо, що даний протокол використовує RSA алгоритм для створення ключів та симетричний алгоритм AES.

Із приводу конфіденційності, то можна сказати, що Telegram:

– може збирати пристойну кількість особистої інформації, яку він може зберігати до 12 місяців;

– відповідно до їх політики конфіденційності, вони, можуть такі метадані, як IP-адреса, пристрої та програми Telegram, якими ви користуєтеся, історія змін імені користувача тощо.

Окрім того, компанія має можливість читати будь-які ваші повідомлення Cloud Chat, щоб розслідувати спам та інші порушення їхніх умов обслуговування. Вони можуть ділитися деякими вашими особистими даними з іншими користувачами Telegram, з якими ви вирішите

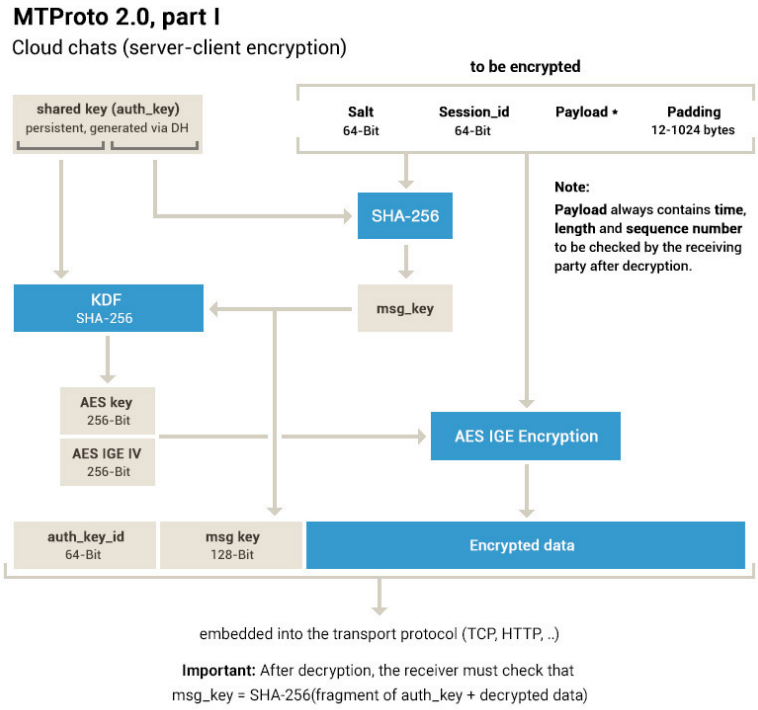


Рисунок 1.2 – Схема MTProto v1 для хмарних чатів.

спілкуватися, і компаніями в групі Telegram. Якщо вимагатиметься рішення суду, вони можуть надати вашу IP-адресу та номер мобільного телефону відповідним органам. Це важливо мати на увазі під час використання Telegram.

Наступним розглянемо протокол **MTProto2** [3], що на даний час використовується для секретних та звичайних хмарних чатів.

MTProto складається з кількох (під)протоколів, які обробляють початкову автентифікацію клієнтів зі створенням спільних ключів між клієнтами та сервером, створенням ключів сеансу між двома клієнтами для наскрізного шифрування в секретних чатах, повторним введенням секретних чатів і, звичайно, шифрування всіх повідомлень перед передачею через (можливо, незахищену) мережу.

Незважаючи на те, що MTProto 2.0 є повністю відкритим, а код клієнта відкритим, модель безпеки Telegram отримала широку критику. Перш за все, це через вибір нестандартного, спеціального протоколу та схеми шифрування викликав заперечення, тому що відсутність контролю може наражати систему на вразливість, потенційно підриваючи її безпеку. Крім того, усі повідомлення (навіть ті, що містяться в «секретних чатах») проходять через (хмару) пропрієтарні сервери із закритим кодом, де вони можуть зберігатися будь-який час. Ця архітектура зручна для користувачів, які можуть отримувати доступ до своїх повідомлень і синхронізувати їх із кількох пристроїв, а також надсилати й отримувати повідомлення навіть тоді, коли одноранговий пристрій відсутній, але з точки зору безпеки це означає, що сервер слід розглядати як ненадійну сторону.

Загалом MTProto v2 вважається безпечним і його більш детальний розбір можна знайти за посиланням [4]. Документ надає повністю автоматизоване підтвердження надійності

автентифікації MTProto 2.0, звичайного та секретного чатів та механізмів повторного введення ключів, а також щодо властивостей безпеки: автентифікації, цілісність, конфіденційність й ідеальну пряму секретність. Це частково усуває занепокоєння щодо недостатньої перевірки, одночасно підтверджуючи формальну безпеку останньої версії протоколу.

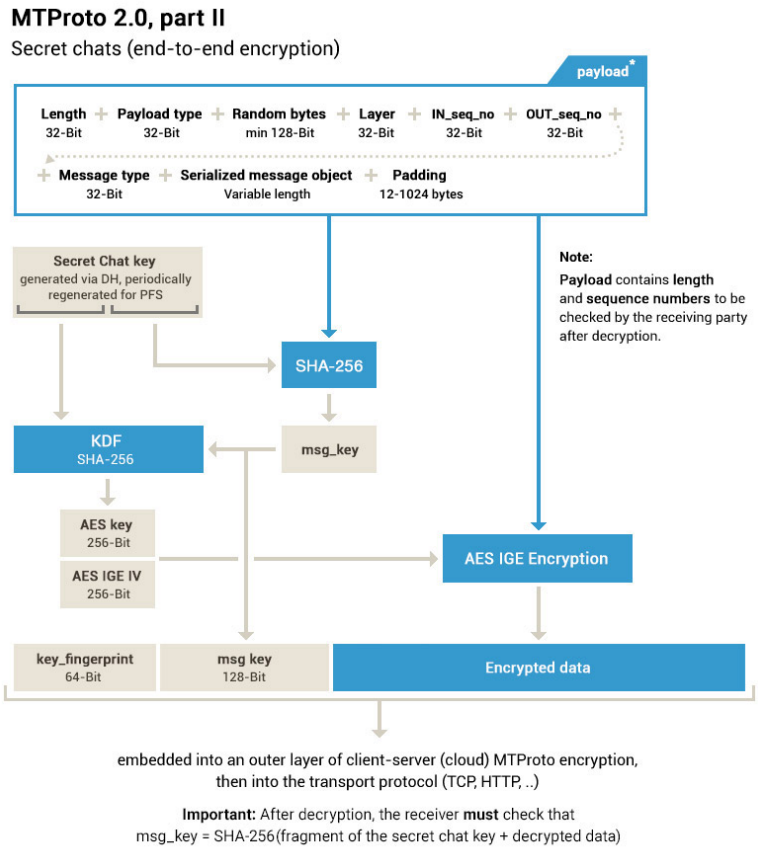


Рисунок 1.3 – Схема MProto v2.

Тобто як узагальнити, то месенджер використовує дискретний логарифм для створення спільного ключа та загалом надійну схему протокола, що наведена у наступній схемі із урахуванням рівнів.

Цю схему можна органічно поділити на 3 рівні, а саме:

– **Прикладний рівень** – визначає спосіб перетворення запитів і відповідей API до двійкових повідомлень. Цей компонент відповідає рівням OSI 7 (application) і 6 (presentation).

– **Криптографічні та авторизаційні компоненти** - визначають, як користувачі автентифікуються на сервері, а повідомлення шифруються перед передачею через транспортний протокол. Ці компоненти відповідають рівням OSI 5 (session) і 4 (transport).

– Авторизація: цей модуль надає функціональні можливості для початкової авторизації клієнта та автентифікації сервера. Він викликається під час першого запуску програми для отримання ключа авторизації (AK), довгострокового «головного» секрету, який надається лише серверу. Щоб встановити ключ авторизації, цей модуль виконує криптографічний протокол (в основному обмін DH) із сервером.

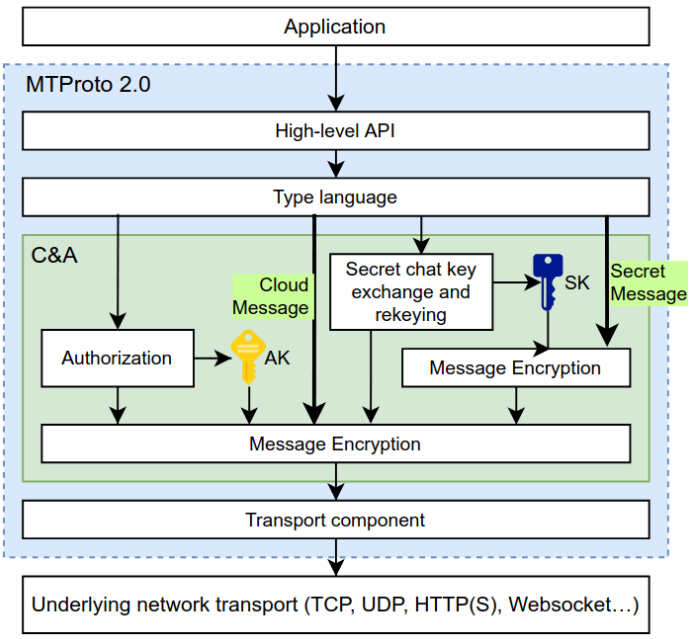


Рисунок 1.4 – Рівні протоколу MProto v2.

– Обмін секретним ключем чату та зміна ключа: цей модуль надає функції для встановлення спільного секретного ключа сеансу (SK) між двома клієнтами. Він виконується один раз на початку секретного чату та після 100 обміну повідомленнями між двома сторонами (або протягом тижня) для встановлення нового ключа. В обох випадках цей модуль виконує обмін Діффі-Геллмана з одноранговим клієнтом (через сервер).

– Шифрування повідомлень: усі повідомлення між клієнтом і сервером шифруються за допомогою симетричного шифру з використанням ефемерного ключа, отриманого з AK. Повідомлення в секретних чатах також шифруються за допомогою ефемерного ключа, отриманого від SK. Схема шифрування однакова, але використовується двічі (з різними ключами) для повідомлень у секретних чатах.

– **Транспортна компонента** – визначає, як клієнт і сервер фактично обмінюються повідомленнями через існуючий транспортний протокол, такий як UDP, TCP, HTTP, HTTPS, WebSocket тощо. Варто зауважити, що також підтримуються незахищені протоколи без з'єднання.

- A->B : (generates a and) sends g_a_hash := hash(g^a)
- B->A : (stores g_a_hash, generates b and) sends g_b := g^b
- A->B : (computes key (g_b)^a, then) sends g_a := g^a
- B : checks hash(g_a) == g_a_hash, then computes key (g_a)^b

Рисунок 1.5 – Схема автентифікації користувачів MProto v2 для дзвінків.

На останок додамо пару слів про аудіо та відео дзвінки [5]. У Telegram вони реалізовані за допомогою трохи модифікованої версії MTProto v2, що використовується у секретних чатах. У ній додано додаткову автентифікацію користувача за умови входження в онлайн, що є за

замовчуванням, адже це є дзвінок. Обмін повідомленнями для автентифікації виглядає наступним чином 1.5.

1.2 Viber

Viber, це крос-платформне програмне забезпечення для передачі голосу через IP (VoIP) і обміну миттєвими повідомленнями (IM), яке належить японській транснаціональній компанії Rakuten.

Голосові дзвінки, відеодзвінки, групові чати, стікери, соціальні канали, чати, що самознищуються, наскрізне шифрування, багато спаму та цільова реклама. Viber пропонує всі найкращі практики світу із програм обміну повідомленнями. Він схожий на суміш WhatsApp і Telegram із додаванням Snapchat. І це величезний додаток на понад мільярд користувачів у всьому світі. Під час пандемії Viber розширив функції групового відеочату, щоб дозволити більше користувачів, що призвело до появи цікавих речей, як-от вчителі в Угорщині використовували його, щоб підтримувати зв'язок зі своїми учнями.

1.2.1 Протоколи

Viber досить добре захищає конфіденційність, адже у 2016 році за замовчуванням було прийнято наскрізне шифрування, окрім групових дзвінків. Тобто, якщо підсумувати, то наскрізне шифрування [6]:

- використовується для
  1) особистих повідомлень та дзвінків між двома абонентами;
  2) груп.
- не використовується для
  - спільнот;
  - каналів;
  - чатів із ботами;
  - групових дзвінків;
  - платних дзвінків із Viber Out.

E2E побудовано таким чином, що кожен пристрій користувача генерує 256-бітну Curve-25519 пару ключів, що зветься ID Key. Зокрема, за необхідності користувач генерує PreKey пару ключів для поширення їх на сервер та для іншого користувача.

Сесії користувачів між пристроями створюються тоді, коли пристрій перший раз реєструється. Тільки одна сесія потрібна для того, щоб синхронізувати будь-яку кількість

повідомлень між двома акаунтами.

У випадку, коли користувач А хоче встановити зв'язок із користувачем В, ініціатор отримує від сервера відповідні ID Key та PreKeys, що були попередньо згенеровані та відправлені на сервер іншим користувачем. Далі ініціатор генерує дві ключові пари по 256-біт Curve-25519 для рукостискання та для шифрування методом храповика відповідно.

$$\text{RootKey} = \text{SHA256}\ (\ \text{DH}(\text{ID}_{\text{Alice}}, \text{HS}_{\text{Bob}}) \parallel \text{DH}(\text{HS}_{\text{Alice}}, \text{ID}_{\text{Bob}}) \parallel \text{DH}(\text{HS}_{\text{Alice}}, \text{HS}_{\text{Bob}})\ )$$

Рисунок 1.6 – Схема одержання RootKey.

$$\text{TempKey} = \text{HMAC\_SHA256}(\text{RootKey}, \text{DH}(\text{Ratchet}_{\text{Alice}}, \text{Ratchet}_{\text{Bob}}))$$
$$\text{New RootKey} = \text{HMAC\_SHA256}(\text{TempKey}, \text{"root"})$$
$$\text{SessionKey} = \text{HMAC\_SHA256}(\text{TempKey}, \text{"mesg"})$$

Рисунок 1.7 – Схема одержання ключа для сесії.

Різні рядки, передані до функцій HMAC запевняють нас, що навіть якщо ключ сеансу зламано, то кореневий ключ не може бути отриманий назад з нього.

Потім ініціатор надсилає користувачу Б повідомлення про початок сеансу, що містить його власний публічний ідентифікатор ID Key, ідентифікатор попереднього PreKey користувача Б, який використовувався для цього сеансу, а також його власні публічні ключі рукостискання та храповий ключ ініціатора. Коли користувач Б підключається до Інтернету та отримує це повідомлення він може реконструювати той самий кореневий і сеансовий ключі за допомогою тієї самої процедури DH.

Для **шифрування повідомлень**, як було згадано, використовується метод храповика. Для досягнення цього генерується ефемерний одноразовий 128-бітний симетричний ключ, який використовується для шифрування тіла повідомлення за допомогою алгоритму шифрування Salsa20. Потім цей ефемерний ключ повідомлення шифрується за допомогою сеансового ключа кожного одержувача. Пристрій-відправник надсилає уніфіковане повідомлення на сервер, що містить один зашифрований шифротекст і набір зашифрованих ефемерних ключів. Розгортання на стороні сервера розрізає це повідомлення та доставляє відповідні частини до кожного цільового пристрою. Два пристрої по черзі просувають сеансові ключі в процесі, який називається храповим механізмом. Кожного разу, коли напрямок розмови змінюється, пристрій, чия черга, випадковим чином генерує нову пару ключів Ratchet і знову виконує таку послідовність:

З $\text{Ratchet}_{\text{this device}}$ є приватною частиною щойно отриманої пари ключів. Разом із кожним повідомленням також надсилається загальнодоступна частина $\text{Ratchet}_{\text{this device}}$. Одержувач запускає DH зі своїм останнім приватним храповиком разом із публічним

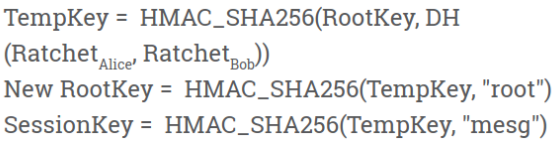


Рисунок 1.8 – Схема одержання ключа для шифрування методом храповика.

храповиком відправника.

Подвійний храповий механізм забезпечує дві речі:

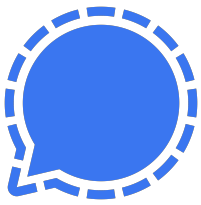
– пряму та зворотну секретність, тож навіть якщо ключі скомпрометовані, минулі та майбутні повідомлення неможливо розшифрувати;

– алгоритм підтримує автентифікацію пристрою користувача, оскільки DH-ланцюжок кореневих ключів починається з ID Keys обох пристроїв. Якщо ID Keys однорангового вузла є надійним у будь-який момент, то всьому ланцюжку можна довіряти.

1.2.2 Репутація

Загалом Viber має в Україні одну з найбільших спільнот користувачів, а саме 9 мільйонів. Не дивлячись на велику підтримку України, можна сказати, що це є тим месенджером, якому можна довіряти. Із іншого боку Сили Оборони України не довіряють йому і користуються альтернативними месенджерами. Зокрема, нещодавно була здійснена атака із викраденням 740 гігабайтів даних із месенджера **??**. У документації бракує детальнішого опису алгоритмів, щоб зрозуміти доцільність використання даного месенджера.

1.3 Signal



Signal - це відкрите програмне забезпечення із обміну зашифрованими миттєвими повідомленнями, голосовими дзвінками та відеодзвінками. Функція обміну миттєвими повідомленнями включає надсилання тексту, голосових нотаток, зображень, відео та інших файлів. Спілкування між користувачами може бути індивідуальним або передбачати груповий обмін повідомленнями.

Додамо, що Відповідно до Signal, він «ні в якому разі не продає, не здає в оренду та не монетизує ваші особисті дані чи контент». Signal реалізує принципи мінімізації даних - лише номер телефону використовується як джерело даних - і захисту даних за замовчуванням і за проектом.

1.3.1 Протоколи

Створення безпечного протоколу обміну повідомленнями не могло обійтися без використання складних схем шифрування та обміну даними. Його називають як стеком протоколів Signal і, як мінімум, використовує WhatsApp для свого функціонування. Обговоримо технічні деталі, що були використані для його реалізації.

– XEdDSA та VXEdDSA

XEdDSA та VXEdDSA — це схеми криптографічного підпису, розроблені для уніфікації форматів ключів для операцій еліптичної кривої Діффі-Геллмана (ECDH) і алгоритму цифрового підпису Едвардса (EdDSA). У деяких ситуаціях це дозволяє використовувати ту саму пару ключів для обох алгоритмів.

XEdDSA полегшує використання однієї пари ключів як для операцій ECDH, так і для EdDSA, спрощуючи керування ключами та зменшуючи потребу в кількох парах ключів. Також дозволяє створювати та перевіряти підписи, сумісні з EdDSA, використовуючи ключові формати, спочатку визначені для функцій Діффі-Геллмана на еліптичній кривій X25519 і X448.

VXEdDSA розширює XEdDSA шляхом включення керування версіями, дозволяючи схемі функціонувати як перевірена випадкова функція (VRF). На додаток до перевірки підпису, за успішного завершення, VXEdDSA повертає результат VRF, який гарантовано буде унікальним для повідомлення та відкритого ключа. Вихідні дані VRF для даного повідомлення та відкритого ключа неможливо відрізнити від випадкових для тих, хто не бачив підпис VXEdDSA для цього повідомлення та ключа.

Додамо, що детальніше нюанси реалізації можна побачити за цим посиланням [7].

– X3DH

X3DH (або Extended Triple Diffie-Hellman) - це схема узгодження ключів, що встановлює спільний секретний ключ між двома сторонами, які взаємно автентифікують одна одну на основі відкритих ключів. X3DH забезпечує пряму секретність і криптографічну заборону.

X3DH розроблено для асинхронних налаштувань, коли один користувач («Боб») перебуває в автономному режимі, але опублікував певну інформацію на сервері. Інший користувач («Аліса») хоче використати цю інформацію для надсилання зашифрованих даних Бобу, а також створити спільний секретний ключ для майбутнього спілкування.

Уточнимо, що IK - індентифікаційний ключ, SPK - попередньо підписаний попередній ключ, EK - ефемерний ключ, OPK - одноразовий передключ.

Додамо, що детальніше нюанси реалізації можна побачити за цим посиланням [8].

– PQXDH

PQXDH (або Post-Quantum Extended Diffie-Hellman) - протокол узгодження ключів, що встановлює спільний секретний ключ між двома сторонами, які взаємно автентифікують одна одну на основі відкритих ключів. PQXDH забезпечує постквантову пряму секретність і певну форму криптографічного заперечення, але все ще покладається на складність проблеми

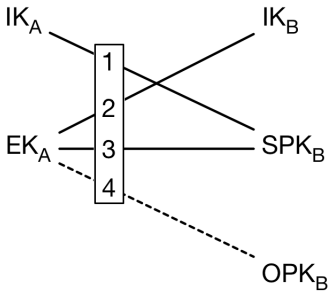


Рисунок 1.9 – Схема обміну ключами для X3DH.

дискретного логарифмування для взаємної автентифікації.

PQXDH розроблено для асинхронних налаштувань, коли один користувач («Боб») перебуває в автономному режимі, але опублікував певну інформацію на сервері. Інший користувач («Аліса») хоче використати цю інформацію для надсилання зашифрованих даних Бобу, а також створити спільний секретний ключ для майбутнього спілкування.

Додамо, що детальніше нюанси реалізації можна побачити за цим посиланням [9]

– Double Ratchet

Double Ratchet - криптографічний протокол, який використовується двома сторонами для обміну зашифрованими повідомленнями на основі спільного секретного ключа. Як правило, сторони використовують певний протокол узгодження ключів (наприклад, X3DH), щоб узгодити спільний секретний ключ. Після цього сторони використовуватимуть Double Ratchet для надсилання та отримання зашифрованих повідомлень.

Сторони отримують нові ключі для кожного повідомлення Double Ratchet, щоб попередні ключі не можна було обчислити з пізніших. Сторони також надсилають загальнодоступні цінності Діффі-Геллмана, додані до своїх повідомлень. Результати обчислень Діффі-Геллмана змішуються з похідними ключами, щоб пізніші ключі не можна було обчислити з попередніх. Ці властивості надають певний захист попереднім або пізнішим зашифрованим повідомленням у разі зламу ключів сторони.

Додамо, що детальніше нюанси реалізації можна побачити за цим посиланням [10]

– Sesame

Алогоритм Sesame - це протокол для керування сеансами шифрування повідомлень в асинхронних налаштуваннях із кількома пристроями.

Sesame був розроблений для керування сесіями Double Ratchet, створеними за допомогою угоди ключа X3DH. Він координує створення, видалення та використання сеансів шифрування на кількох пристроях, забезпечуючи безпечний і синхронізований зв'язок.

Sesame:

– підтримує активні сеанси шифрування між пристроями, сприяючи безперебійному обміну повідомленнями, навіть якщо деякі пристрої перебувають у режимі офлайн;

– ефективно обробляє сценарії, коли користувачі працюють з кількома пристроями, гарантуючи, що повідомлення належним чином зашифровані та доставлені на кожен пристрій;

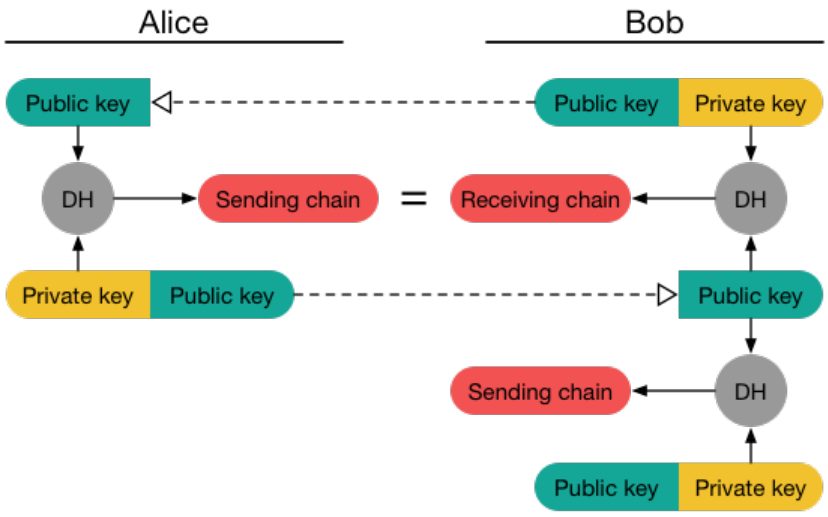


Рисунок 1.10 – Схема створення нових ключів для шифрування у Double Ratchet.

– гарантує, що кожен пристрій підтримує «активний» сеанс для кожного іншого пристрою, з яким він спілкується, гарантуючи, що повідомлення шифруються за допомогою правильних ключів сеансу;

– гарантує, що коли повідомлення отримано під час неактивного сеансу, цей сеанс стає активним, дозволяючи пристроям синхронізуватися та узгодити відповідний сеанс для використання;

– керує життєвим циклом сеансів, створюючи нові під час додавання пристроїв і видаляючи їх, коли пристрої видаляються, зберігаючи цілісність комунікаційної мережі.

Однак Sesame — це загальний алгоритм, який працює з будь-яким алгоритмом шифрування повідомлень на основі сеансу, який відповідає певним умовам.

Додамо, що детальніше нюанси реалізації можна побачити за цим посиланням [11]

Цікаво, але для Signal протоколів теж існують роботи із аналізу безпеки, а саме:

– Формальний аналіз безпеки Signal протоколу обміну[12]

– та інші . . .

1.3.2 Підсумки

Загалом це є одним із найкращих на даний момент месенджерів, що не збирає жодних даних і є орієнтований на анонімне використання. Звичайно, цей додаток не має жодних інтеграцій із бізнесами, що не дає поширювати комерційну інформацію серед них.

1.4 WhatsApp

WhatsApp Messenger — це служба обміну миттєвими повідомленнями і передачі голосу через IP, що належить технологічному конгломерату Meta. Це дозволяє користувачам надсилати текстові, голосові та відеоповідомлення, здійснювати голосові та відеодзвінки, а також ділитися зображеннями, документами, місцезнаходженням користувачів та іншим вмістом. Послуга вимагає номер стільникового мобільного телефону для реєстрації.[19] У січні 2018 року WhatsApp випустив окрему бізнес-програму під назвою WhatsApp Business, яка може спілкуватися зі стандартним клієнтом WhatsApp.

WhatsApp був заснований Яном Кумом і Брайаном Ектоном, які раніше 20 років разом працювали в Yahoo. WhatsApp приєднався до Facebook у 2014 році, але продовжує працювати як окрема програма з фокусом на створенні служби обміну повідомленнями, яка працює швидко та надійно в будь-якій точці світу.

1.4.1 Протоколи

WhatsApp використовує Signal протоколи, що були проаналізовані та протестовані усіма можливими дослідниками. Вони вважаються на даний час одними із найкращих. Додам, що даний месенджер використовує реалізацію протоколів, але реалізація додатку залишається за компанією. Відповдіно до запису у блозі Signal, WhatsApp перейшов на наскрізне шифрування аж у 2016 році, що каже про обізнаність розробників [13].

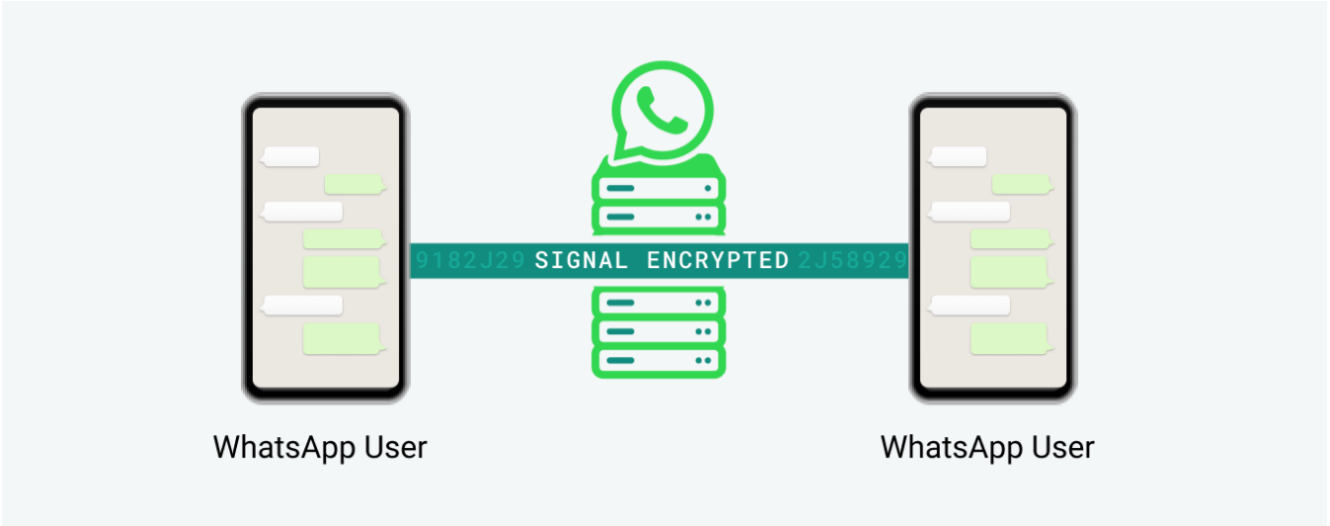


Рисунок 1.11 – Схема комунікації користувачів як через звичайну мережу так і через мережу бізнесів.

Неможливо не сказати про схему реалізації спілкування користувачів між користувачами та бізнесами. Перший рисунок показує схему спілкування звичайних користувачів із собі ж подібними на загальній мережі. Тобто всі повідомлення певним чином маршрутизуються та оброблюються розгорнутим сервером WhatsApp. У WhatsApp Business можна користуватися чатом, що має більше можливостей, ніж звичайний, для просування власних потреб у ньому 1.11.

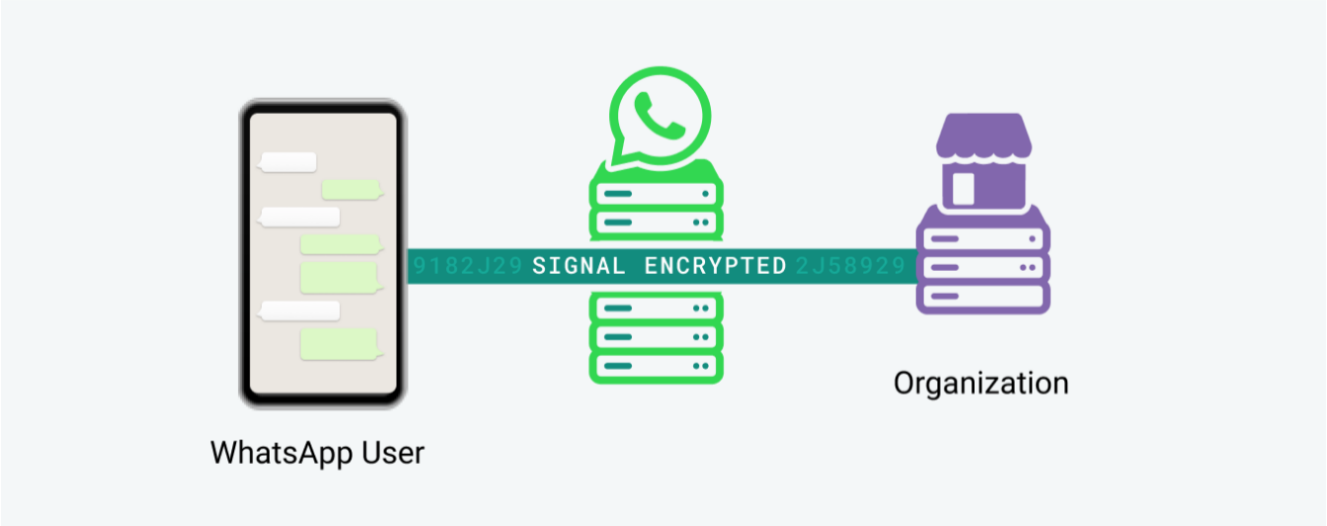


Рисунок 1.12 – Схема отримання даних для біснесу на їхні сервери.

Деякі організації можуть використовувати локальний WhatsApp Business API додаток 1.12, який можна розгорнути як кінцеву точку WhatsApp на сервері. API дозволяє цим організаціям програмно надсилати й отримувати повідомлення.

WhatsApp вважає зв'язок із локальними користувачами WhatsApp Business API, які керують кінцевою точкою API на серверах, наскрізно зашифрованим, оскільки немає стороннього доступу до вмісту між кінцевими точками.

Підсумовуючи, наведемо список із знайдених досліджень щодо стійкості імплементованого месенджера:

- Документ, що дає зрозмути які алгоритми шифрування використовуються [14].
- Побудова атак на протокол реєстрації супутнього пристрою в WhatsApp [15].
- Порівняння аналізу безпеки між Signal, WhatsApp і Telegram [16].
- Аналіз документів безпеки WhatsApp [17].
- Інший аналіз безпеки від International Journal of Science and Research [18].
- Аналіз безпеки протоколу наскрізного шифрування резервного копіювання [19].

1.4.2 Підсумки

Як узагальнемо, то WhatsApp є надійним месенджером, що він надає надійні гарантії безпеки як і для бізнесу так і для особистого спілкування. Єдине на що треба зважати, це збір даних який все таки трохи відбувається через Meta для покращення власних сервісів.

1.5 Різниця Signal та WhatsApp

Не дивлячись на те, що WhatsApp також використовує Signal протокол для обміну повідомленнями, проте вони мають у собі певні різниці:

- **Право володіння**
  - ***Signal***

Належить некомерційній організації Signal Foundation, яка діє як неприбуткова організація та фінансується за рахунок пожертвувань та грантів. Немає жодної реклами, відстеження чи монетизації даних користувачів.

  - ***WhatsApp***

Належить Meta (або раніше Facebook). Бізнес-модель передбачає інтеграцію з ширшою екосистемою Meta, включаючи потенційний обмін даними (хоча чати WhatsApp залишаються зашифрованими).

- **Збирання даних**
  - ***Signal***

Збирає мінімум метаданих (до прикладу, номер телефону та час створення облікового запису). Не збирає та не зберігає історію чату чи дані користувача.

  - ***WhatsApp***

Збирає більше метаданих: номери телефонів, контакти, інформацію про пристрій і статистику використання. Ділиться деякими даними з Meta для аналітики та бізнес-інтеграції.

- **Безпека**
  - ***Signal***

Використовує свій відкритий протокол сигналу для наскрізного шифрування (E2EE). Повідомлення, дзвінки, групові чати та мультимедійні дані є зашифрованими. У месенджері немає резервних копій або хмарної синхронізації повідомлень, якщо вони явно не були експортовані користувачем.

  - ***WhatsApp***

Також використовує протокол сигналу для E2EE. Повідомлення, дзвінки та групові чати шифруються за замовчуванням. Резервні копії, що зберігаються в хмарних службах, не шифруються, якщо користувач не ввімкне резервне копіювання з наскрізним шифруванням.

- **Особливості та фокус**

– ***Signal***

Орієнтований на конфіденційність і простоту. Пропонує такі функції, як зникнення повідомлень, анонімний обмін повідомленнями (без спільного доступу до контактів) і безпечні PIN-коди. Немає широких бізнес або соціальних функцій.

– ***WhatsApp***

Призначений для більш широкого використання, включаючи соціальне та ділове спілкування. Пропонує такі функції, як WhatsApp Business [20], канали, оновлення статусу та зашифровані резервні копії. Поєднує безпеку зі зручністю для користувача та соціальною залученістю.

– **Відкритість вихідних кодів**

– ***Signal***

Повністю відкритий вихідний код, тобто його код є загальнодоступним для вивчення та модифікації.

– ***WhatsApp***

Частково з відкритим вихідним кодом, але більшість його коду та внутрішніх операцій залишаються запатентованими.

– **База користувачів**

– ***Signal***

Приваблює користувачів, які надають пріоритет конфіденційності, безпеці та некомерційній етиці. Менша база користувачів порівняно з WhatsApp.

– ***WhatsApp***

Один із найпоширеніших додатків для обміну повідомленнями в усьому світі з мільярдами користувачів. Більш широке впровадження та інтеграція з бізнесом.

– **Signal протокол**

– ***Signal***

Signal переважно використовує реалізацію протоколу Signal (libsignal) [21], що імплементована на Rust у його сучасних версіях. Перевага віддається Rust через його продуктивність, гарантії безпеки та строгу систему типів.

– ***WhatsApp***

Відомо, що WhatsApp використовує реалізацію Java протоколу сигналів (libsignal-protocol-java) [22] у своїх Android і серверних системах. Бібліотека Java добре інтегрується з існуючою інфраструктурою WhatsApp, яка значною мірою покладається на Java для серверних служб і розробки Android.

2 КОНФІДЕНЦІЙНІСТЬ ЗА ПРОЕКТУВАННЯМ

На останок можна згадати роботу дослідників, що сформували документ у якому містяться пункти про те, як треба робити безпечні додатки. Розберемо коротко 7 основоположних принципів конфіденційності за проектуванням [23]:

– **Проактивний, а не реактивний; Профілактичний, а не лікувальний**

Це означає, що програмне забезпечення має залишатися підтримуваним, а розробники - зобов'язаними, як з огляду на законодавство, так і з огляду на суспільну культуру, дотримуватися найкращих практик захисту інформації. Важливо діяти на випередження, а не обмежуватися реактивними заходами.

– **Конфіденційність за замовчуванням**

Означає використання мінімізації зібраних особистих даних, детальне аргументування для чого вони потрібні, відкриття певних даних лише у законних цілях за проханням держави і безпечно утилізовані.

– **Конфіденційність влаштована у дизайн проекту**

Конфіденційність повинна бути влаштована у дизайн цілісної системи, щоб при помилці, взламі мінімізувати витік даних. Зокрема повинно бути пом'якшено ризики, що зв'язані із можливими векторами атак на конфіденційність додатку.

– **Повна функціональність**

Конфіденціність за проектуванням за своєю природою забезпечує подвійні можливості, дозволяючи повну функціональність, а саме реальні практичні результати та вигідні результати для кількох сторін. Мається на увазі, що користувачам повнно бути надано повний функціонал додатку без урізання можливостей із захисту даних.

– **Наскрізне шифрування**

Конфіденційність має постійно захищати в усьому домені та протягом життєвого циклу даних. Організації повинні нести відповідальність за безпеку особистої інформації відповідно до ступеня чутливості протягом усього життєвого циклу відповідно до стандартів. І звичайно треба дотримуватися правильності видалення/зберігання/розповсюдження даних.

– **Видимість та порозорість**

Означає використання підзвітності щодо того як дані є зшифровані чи куди були передані для зберігання, відкритості задля легкої перевірки, відповідності дій відповідно до задокументованих процедур доступу до інформації.

– **Повага до конфіденційності користувачів**

Означає урахування інтересів і потреб окремих користувачів, які найбільше зацікавлені в управлінні своїми особистими даними. Надання можливості суб'єктам даних відігравати активну роль в управлінні своїми власними даними.

ЛІТЕРАТУРА

1. Telegram. *MTProto Mobile Protocol Documentation* Accessed: 2024-11-30. 2024. https://core.telegram.org/mtproto.
2. MTPsym. *Security Analysis of Telegram (Symmetric Part)* https://mtpsym.github.io/. Accessed: 2024-11-21. 2024.
3. Telegram. *End-to-End Encryption API Documentation* Accessed: 2024-11-30. 2024. https://core.telegram.org/api/end-to-end.
4. Miculan, M. & Vitacolonna, N. *Automated Symbolic Verification of Telegram's MTProto 2.0* в *Proceedings of the 18th International Conference on Security and Cryptography, SECRYPT 2021* Accessed: 2024-11-30 (SCITEPRESS, 2021), 185—197. ISBN: 978-989-758-524-1. arXiv: 2012.03141v2 [cs.CR]. https://arxiv.org/abs/2012.03141v2.
5. Telegram. *End-to-End Encryption for Video Calls Documentation* Accessed: 2024-11-30. 2024. https://core.telegram.org/api/end-to-end/video-calls.
6. Viber Media S.à r.l. *End-to-end encryption in chats* Accessed: 2024-12-30. n.d. https://help.viber.com/hc/en-us/articles/8909167863453-End-to-end-encryption-in-chats.
7. (editor), T. P. *The XEdDSA and VXEdDSA Signature Schemes* https://signal.org/docs/specifications/xeddsa/. Revision 1, [PDF]. Жовт. 2016.
8. Marlinspike, M. & (editor), T. P. *The X3DH Key Agreement Protocol* https://signal.org/docs/specifications/x3dh/. Revision 1, [PDF]. Листоп. 2016.
9. Kret, E. & Schmidt, R. *The PQXDH Key Agreement Protocol* https://signal.org/docs/specifications/pqxdh/. Revision 3, Last Updated: 2024-01-23, [PDF]. Трав. 2023.
10. (editor), T. P. & Marlinspike, M. *The Double Ratchet Algorithm* https://signal.org/docs/specifications/doubleratchet/. Revision 1, [PDF]. Листоп. 2016.
11. Marlinspike, M. & (editor), T. P. *The Sesame Algorithm: Session Management for Asynchronous Message Encryption* https://signal.org/docs/specifications/sesame/. Revision 2, [PDF]. Квіт. 2017.
12. Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L. & Stebila, D. *A Formal Security Analysis of the Signal Messaging Protocol* https://eprint.iacr.org/2016/1013. Accessed: 2025-01-04. 2016.
13. moxie0. *WhatsApp's Signal Protocol integration is now complete* https://signal.org/blog/whatsapp-complete/. Accessed: 2025-01-04. Квіт. 2016.
14. WhatsApp. *WhatsApp Encryption Overview: Technical white paper* https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf. Вер. 8. Серп. 2024.

15. Немкович, О. М. *Побудова атак на протокол реєстрації супутнього пристрою в WhatsApp* https://ela.kpi.ua/items/e6faa995-b1f0-44ea-9b21-ca2a99cdc035. Advisor: Фесенко, Андрій Вячеславович. 2023.

16. Bogos, C.-E., Mocanu, R. & Simion, E. *A security analysis comparison between Signal, WhatsApp and Telegram* https://eprint.iacr.org/2023/071.pdf. Accessed: 2025-01-04. 2023.

17. Li, C., Sanchez, D. & Hua, S. *WhatsApp Security Paper Analysis* https://courses.csail.mit.edu/6.857/2016/files/36.pdf. Accessed: 2025-01-04. 2016.

18. AL-Hameed, M. R. Whatsapp Security. *International Journal of Science and Research (IJSR)* **7.** Licensed Under Creative Commons Attribution CC BY. ISSN: 2319-7064. https://www.ijsr.net/archive/v7i11/ART20192911.pdf (2018).

19. Davies, G. T. *та ін. Security Analysis of the WhatsApp End-to-End Encrypted Backup Protocol* https://eprint.iacr.org/2023/843.pdf. Accessed: 2025-01-04. 2023.

20. WhatsApp Business. *WhatsApp Business Platform* https://business.whatsapp.com/. Accessed: 2025-01-04.

21. Signal Foundation. *libsignal: Signal Protocol Library* https://github.com/signalapp/libsignal. Home to the Signal Protocol as well as other cryptographic primitives which make Signal possible. Accessed: 2025-01-04.

22. Signal Foundation. *libsignal-protocol-java: Signal Protocol Implementation in Java* https://github.com/signalapp/libsignal-protocol-java. Accessed: 2025-01-04.

23. Cavoukian, A. *Privacy by Design: The 7 Foundational Principles* Accessed: 2024-11-30. 2011. https://privacy.ucsc.edu/resources/privacy-by-design---foundational-principles.pdf.