УДК 004.056

Дмитро ПЕРЕРВА

Західноукраїнський національний університет

АЛГОРИТМИ ШИФРУВАННЯ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ ОБМІНУ ПОВІДОМЛЕННЯМИ

Вступ. У сучасному світі безпечний обмін повідомленнями є ключовою складовою цифрової комунікації. Зростання кількості кіберзагроз, атак на мережеві сервіси та спроб перехоплення даних вимагає використання надійних методів криптографічного захисту. Алгоритми шифрування є основою безпечних комунікацій у більшості месенджерів і протоколів – від TLS до Signal. Від ефективності цих алгоритмів залежить не лише конфіденційність даних, але й довіра користувачів до цифрових сервісів.

Мета. Дослідження сучасних алгоритмів шифрування, їхньої архітектури, взаємодії компонентів і практичної реалізації для підвищення рівня безпеки обміну повідомленнями.

1. Аналіз криптографічних підходів у системах обміну повідомленнями

На сьогодні існує два базових підходи до шифрування – симетричний та асиметричний. Симетричні алгоритми (наприклад, AES, ChaCha20) використовують один спільний ключ для шифрування і розшифрування повідомлень. Вони мають високу швидкодію і застосовуються для обробки великих обсягів даних. Натомість асиметричні методи (RSA, ECC, Curve25519) використовують пару ключів – відкритий і приватний, що дозволяє безпечно передавати ключі та забезпечує автентичність учасників обміну.

Більшість сучасних протоколів, таких як Signal Protocol чи OMEMO, поєднують обидва підходи. Асиметричні алгоритми відповідають за створення сеансових ключів, тоді як симетричні – за безпосереднє шифрування даних. Така гібридна модель гарантує і конфіденційність, і високу продуктивність системи. Для побудови власної системи шифрування в рамках цього дослідження обрано поєднання алгоритмів AES-GCM і Curve25519, що відповідає сучасним стандартам безпеки, зокрема рекомендаціям NIST і Signal Foundation. На рисунку 1. наведена структурна схема алгоритму Curve25519 [1].

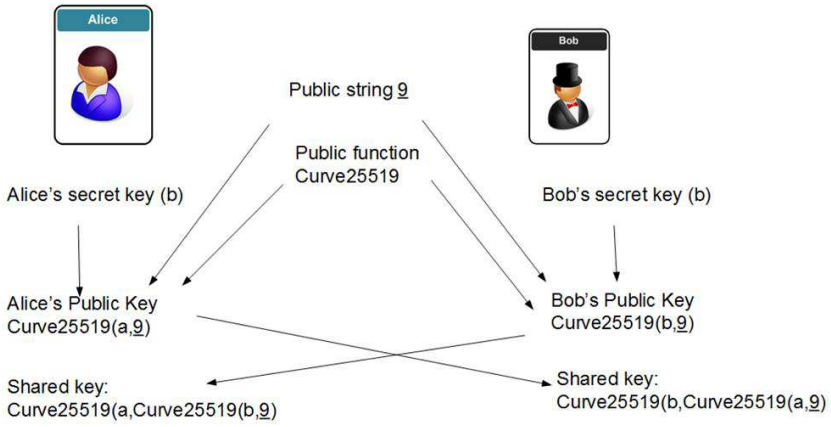

Рисунок 1 - Структурна схема алгоритму Curve25519

З огляду на сучасні тенденції розвитку цифрової комунікації, головною вимогою до будь-якої системи обміну повідомленнями стає забезпечення конфіденційності, цілісності та автентичності даних. Використання криптографічних алгоритмів дозволяє створити стійкі до атак канали зв'язку, однак ефективність таких систем залежить не лише від вибору алгоритму, але й від його коректного впровадження. Останніми роками у світі відбувається активний перехід до асиметричних схем на еліптичних кривих, зокрема на базі Curve25519, яка демонструє високу швидкодію при мінімальному споживанні обчислювальних ресурсів. Вона широко застосовується у популярних месенджерах – Signal, WhatsApp, Session, Element, – де реалізована в рамках протоколів безпечного обміну ключами. Її перевагою є відсутність необхідності у складних сертифікаційних механізмах, що суттєво спрощує інтеграцію в мобільні та десктопні застосунки.

Таким чином, поєднання AES-GCM та Curve25519 утворює гібридну криптосистему, у якій швидкість симетричного шифрування поєднана з безпечним розподілом ключів через асиметричну схему. Особливої актуальності така комбінація набуває в умовах постійного зростання кількості кібератак на месенджери, хмарні сервіси та соціальні платформи. Зловмисники дедалі частіше використовують методи аналізу трафіку, підміни відкритих ключів або атаки типу «людина посередині» (Man-in-the-Middle). Тому реалізація ефективних механізмів шифрування на основі перевірених алгоритмів є ключовим елементом сучасних систем безпечного обміну повідомленнями[2].

2. Реалізація та взаємодія компонентів алгоритму AES-GCM з Curve25519

Для забезпечення конфіденційності повідомлень у сучасних комунікаційних системах важливо застосовувати криптографічні алгоритми, які поєднують високу швидкодію, надійність і можливість реалізації у програмних продуктах з обмеженими ресурсами. У даній роботі розглядається використання симетричного алгоритму AES у режимі Galois/Counter Mode (GCM) у поєднанні з асиметричним механізмом обміну ключами Curve25519. Такий підхід реалізовано у багатьох сучасних протоколах безпечного обміну повідомленнями, зокрема у Signal Protocol та Session, завдяки його здатності забезпечувати Forward Secrecy і високу стійкість до атак. У програмному коді (рисунок 2) реалізовано модуль AESGCM, який виконує основні операції шифрування та дешифрування даних, а також генерує симетричний ключ на основі обміну відкритими ключами за алгоритмом X25519. Код написано мовою Kotlin і використовує бібліотеки javax.crypto для базових криптографічних операцій та Curve25519 для еліптичної криптографії.

```
internal fun encrypt(plaintext: ByteArray, hexEncodedX25519PublicKey: String): EncryptionResult
    val x25519PublicKey = Hex.fromStringCondensed(hexEncodedX25519PublicKey)
    val ephemeralKeyPair = Curve25519.generateKeyPair()
    val symmetricKey = generateSymmetricKey(x25519PublicKey, ephemeralKeyPair.secretKey.data)
    val ciphertext = encrypt(plaintext, symmetricKey)
    return EncryptionResult(ciphertext, symmetricKey, ephemeralKeyPair.pubKey.data)
}
```

Рисунок 2 - Фрагмент коду реалізації алгоритму AES-GCM з Curve25519 (Kotlin)

Основною метою модуля є забезпечення безпечного шифрування повідомлень з автентифікацією даних. Режим AES-GCM дозволяє одночасно виконувати шифрування та контроль цілісності, що усуває потребу у додаткових механізмах перевірки автентичності.

У процесі шифрування генерується вектор ініціалізації (IV) розміром 12 байтів, який додається до зашифрованих даних, що забезпечує унікальність кожної операції. Тег автентичності (GCM Tag) довжиною 128 біт додається до результату шифрування, дозволяючи виявляти будь-які зміни у переданих даних. Функція generateSymmetricKey() реалізує генерацію симетричного ключа через обмін публічними та приватними ключами з використанням алгоритму Curve25519. Для формування остаточного симетричного ключа застосовується функція HMAC-SHA256, яка забезпечує криптографічно стійке перетворення проміжного секрету (shared secret). Цей підхід дозволяє уникнути зберігання ключів у відкритому вигляді та підвищує стійкість системи до атак типу перехоплення ключа. Під час шифрування даних функція encrypt() створює випадковий IV, ініціалізує об'єкт Cipher у режимі GCM, виконує операцію шифрування та поєднує IV з результатом шифрування.

Процес дешифрування реалізований у функції decrypt(), яка виділяє IV з початку блоку даних, ініціалізує Cipher у режимі дешифрування та відновлює оригінальне повідомлення. Комбінація AES-GCM і Curve25519 утворює надійну гібридну криптосистему, у якій симетричне шифрування відповідає за швидкість, а асиметричний обмін ключами – за безпеку. Це дозволяє будувати протоколи з властивістю проспективної секретності (Forward Secrecy), тобто навіть у разі компрометації ключів у майбутньому зловмисник не зможе розшифрувати вже передані повідомлення. Розглянутий модуль може бути використаний як основа для реалізації безпечного обміну повідомленнями у мобільних або десктопних застосунках [3].

Висновок. Проведене дослідження показало, що поєднання симетричних і асиметричних алгоритмів у межах одного криптографічного рішення забезпечує високий рівень безпеки при збереженні швидкодії. Реалізований механізм шифрування на основі AES-GCM та Curve25519 гарантує конфіденційність, цілісність і автентичність даних у системах обміну повідомленнями.

Підхід може бути використаний як база для подальшої інтеграції у месенджери або корпоративні платформи з підтримкою E2EE.

Перелік використаних джерел.

1. Структурна схема алгоритму Curve25519. [Електронний ресурс]. – Режим доступу: https://asecuritysite.com/encryption/go_25519ecdh2

2. Signal Protocol Documentation. [Електронний ресурс]. – Режим доступу: https://signal.org/docs/

3. RFC 5116 – An Interface and Algorithms for Authenticated Encryption. [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc5116