КРИПТОГРАФІЧНІ АЛГОРИТМИ ШИФРУВАННЯ ТА ЕЦП

Сенюк Є. О., студ. (гр. ІО-11, ФІОТ КПІ ім. Ігоря Сікорського);
Мітюк Л. О., к.т.н., доц., Праховнік Н. А., к.т.н., доц. (каф. ОППЦБ КПІ ім. Ігоря Сікорського)

Анотація. Розглянуті питання забезпечення конфіденційності, автентичності та цілісності при обміні чутливими даними.

Ключові слова: шифрування, симетричне шифрування, асиметричне шифрування, ЕЦП, шифр гамування, RSA.

Abstract. Issues of ensuring confidentiality, authenticity and integrity when exchanging sensitive data are considered.

Keywords: encryption, symmetric encryption, asymmetric encryption, electronic digital signature, jamming cipher.

Вступ. Криптографія є однією з ключових технологій сучасного світу, забезпечуючи надійний захист даних та інформації в цифровому середовищі. У наш час, коли обсяг переданих даних зростає з неймовірною швидкістю, шифрування та дешифрування відіграють вирішальну роль у збереженні конфіденційності та цілісності інформації.

Аналіз стану питання. Багато популярних месенджерів, таких як WhatsApp чи Telegram, використовують шифрування для забезпечення безпеки особистих повідомлень користувачів. Це допомагає гарантувати, що тільки відправник і отримувач зможуть прочитати повідомлення, уникаючи стороннього доступу.

Мета роботи: запропонувати методи та алгоритми забезпечення конфіденційності, автентичності та цілісності при обміні чутливими даними.

Методики, матеріали і результати досліджень. Шифрування – це процес перетворення інформації в такий вигляд, який може бути прочитаний лише за допомогою спеціального ключа. Дешифрування, у свою чергу, дозволяє повернути зашифровану інформацію до її оригінальної форми, щоб її зміг прочитати лише той, кому вона призначена.

Симетричне шифрування

Симетричне шифрування використовує один ключ для шифрування та дешифрування даних. Це означає, що відправник та отримувач повинні мати однаковий секретний ключ для успішної передачі зашифрованих повідомлень. Приклади симетричних алгоритмів включають AES (Advanced Encryption Standard), DES (Data Encryption Standard) та Blowfish.

Шифр гамування (або потокове шифрування) – це метод симетричного шифрування, при якому кожен біт або байт відкритого тексту по черзі комбінується з бітами або байтами ключової послідовності (гамою). Завдяки цьому шифру можна досягти високого рівня криптографічного захисту, особливо для передачі потокових даних у режимі реального часу.

Принцип роботи шифру гамування

Гамування базується на генерації псевдовипадкової послідовності (гами), яка має бути такого ж розміру, як і повідомлення. Шифрування здійснюється шляхом побітового або побайтного складання відкритого тексту з гамою, зазвичай за допомогою операції XOR (ексклюзивного «або»). Дешифрування відбувається аналогічно: зашифрований текст знову об'єднується з гамою за допомогою операції XOR, що дозволяє отримати оригінальний текст.

Основні етапи шифру гамування:

1. **Генерація гами:** Спочатку генерується псевдовипадкова послідовність, яка називається гамою. Для її генерації часто використовується спеціальний алгоритм або генератор випадкових чисел.
2. **Шифрування:** Кожен біт або байт відкритого тексту комбінується з відповідним бітом або байтом гами за допомогою операції XOR.
3. **Дешифрування:** Процес дешифрування є ідентичним шифруванню – зашифрований текст комбінується з тією ж гамою за допомогою XOR для отримання початкового тексту.

Формально, процес можна представити наступним чином:

- **Шифрування:** $C=P \oplus K$, де P – відкритий текст, K – гама (ключова послідовність), C – шифротекст.
- **Дешифрування:** $P=C \oplus K$

Асиметричне шифрування

Асиметричне шифрування використовує пару ключів: відкритий (публічний) та закритий (приватний) ключі. Відкритий ключ доступний усім, і він використовується для шифрування даних. Закритий ключ тримається в секреті і використовується для дешифрування. Лише власник закритого ключа може розшифрувати повідомлення, зашифровані відповідним відкритим ключем. Найбільш відомі асиметричні алгоритми – це RSA (Rivest–Shamir–Adleman), DSA (Digital Signature Algorithm) та ECC (Elliptic Curve Cryptography).

RSA (Rivest–Shamir–Adleman) – це один з найпоширеніших алгоритмів асиметричного шифрування, розроблений у 1977 році Рональдом Рівестом, Аді Шаміром та Леонардом Адлеманом. RSA базується на властивостях простих чисел і є фундаментальною основою для багатьох сучасних систем безпеки, таких як захищені веб-з'єднання, цифрові підписи та шифрування даних.

Принцип роботи RSA

Алгоритм RSA використовує пару ключів: відкритий та закритий. Процес роботи алгоритму можна описати такими основними кроками:

1. **Генерація ключів:**
   - Вибираються два великих простих числа, позначені як p і q.
   - Обчислюється їх добуток: $n=p \times q$, яке називається модулем. Значення n використовується в обох ключах.
   - Обчислюється функція Ейлера: $\varphi(n)=(p-1)(q-1)$.

○ Обирається число «e», яке є взаємно простим з φ(n)\varphi(n)φ(n) і задовольняє умову 1<e<φ(n). Це число стає відкритою експонентою.

○ Знаходиться d, обернене до e за модулем φ(n) (тобто d×e≡1 mod φ(n)d). Значення d стає закритою експонентою.

2. **Відкритий і закритий ключі:**

○ **Відкритий ключ** складається з пари чисел (e, n). Його можна відкрито публікувати і використовувати для шифрування даних.

○ **Закритий ключ** складається з пари чисел (d, n). Він тримається в секреті і використовується для дешифрування.

3. **Шифрування:**

○ Для зашифрування повідомлення M використовується формула $C = M^e mod\ n$, де C – це зашифроване повідомлення (шифротекст).

○ Відправник використовує відкритий ключ отримувача, щоб зашифрувати повідомлення. Це дозволяє лише отримувачу з його закритим ключем розшифрувати його.

4. **Дешифрування:**

○ Для дешифрування отриманого шифротексту C використовується формула $M = C^d mod\ n$, де M – це вихідне повідомлення.

○ Оскільки тільки власник закритого ключа має значення d, лише він може відновити оригінальне повідомлення.

Електронний цифровий підпис (ЕЦП) також є важливим компонентом у забезпеченні безпеки даних. ЕЦП підтверджує автентичність документа чи повідомлення, гарантує, що вони були надіслані від конкретного відправника та не змінені під час передачі. У сучасному бізнесі ЕЦП активно використовується для підписання важливих документів, таких як контракти та фінансові звіти, що зменшує ризик підробок і зміцнює довіру між сторонами

Як працює цифровий підпис за допомогою RSA

Процес підписання повідомлення за допомогою RSA включає наступні основні кроки:

1. **Хешування повідомлення:**

○ Спочатку оригінальне повідомлення хешується за допомогою криптографічної хеш-функції, наприклад, SHA-256. Хешування зменшує повідомлення до фіксованої довжини, що дозволяє підписувати його швидше.

○ Хеш-функція гарантує, що навіть незначна зміна в повідомленні призведе до значно іншого хешу.

2. **Шифрування хешу:**

○ Хеш підписується шляхом його шифрування за допомогою закритого ключа відправника. Це і є власне цифровий підпис.

○ Формула для підписання виглядає так: $S = H(M)^d mod\ n$, де S – цифровий підпис, H(M) – хеш повідомлення, d – закрита експонента, а n – модуль.

3. **Відправка повідомлення та підпису:**

○ Підписане повідомлення разом із цифровим підписом передається отримувачу.

4. **Перевірка підпису:**

○ Отримувач спочатку обчислює хеш отриманого повідомлення за допомогою тієї ж хеш-функції, що використовувалася відправником.

○ Далі отримувач розшифровує цифровий підпис за допомогою відкритого ключа відправника для отримання хешу, який відправник підписав. Це можна представити як $H'(M) = S^e mod\ n$

○ Потім отримувач порівнює обчислений хеш з розшифрованим хешем. Якщо вони співпадають, то підпис вважається дійсним, а повідомлення не було змінено.

Висновки. Загалом, криптографія, шифрування, дешифрування та ЕЦП є невід'ємними елементами захисту інформації в цифрову епоху, граючи вирішальну роль у безпечному функціонуванні багатьох систем та комунікацій. Завдяки їм ми можемо безпечно спілкуватися, передавати дані та здійснювати фінансові транзакції в мережі.

Література

1. Diffie, W., & Hellman, M. E. (1976). New Directions in Cryptography. Retrieved from https://www-ee.stanford.edu/~hellman/publications/24.pdf.
2. Тернопільський національний технічний університет імені Івана Пулюя. (n.d.). Алгоритм RSA. Вікі Тернопільського національного технічного університету. Retrieved from https://wiki.tntu.edu.ua/Алгоритм_RSA.
3. Studfile.net. (n.d.). Криптографія та алгоритм RSA. Retrieved from https://studfile.net/preview/6047915/page:5/.