Тема 23. Шифрування

23.1. Шифрування

Шифрування – це кодування даних з метою захисту від несанкціонованого доступу. Процес кодування називається **шифруванням**, а процес декодування – **розшифруванням**. Саме кодоване повідомлення називається **шифрованим**, а застосований метод називається **шифром**.

Основна вимога до шифру полягає в тому, щоби розшифрування (і, можливо, шифрування) були можливі тільки при наявності санкцій, тобто деякої додаткової інформації (або пристрою), яка називається **ключем** шифру. Процес декодування шифровки без ключа називається **дешифруванням**.

Галузь знань про шифри, методи їх побудови та розкриття називається **криптографією**. Властивість шифру протистояти розкриттю називається **криптостійкістю** або **надійністю** і звичайно визначається складністю алгоритму дешифровки.

У практичній криптографії криптостійкість шифру оцінюється з економічних міркувань. Якщо розкриття шифру коштує (в грошовому еквіваленті, включаючи необхідні комп'ютерні ресурси, спеціальні пристрої тощо) більше, за саму зашифровану інформацію, то шифр вважається достатньо надійним.

23.2. Симетричні криптосистеми

Означення 23.1. **Симетричні криптосистеми** (**симетричне шифрування**) – спосіб шифрування, в якому для шифрування та дешифрування використовуються один й той самий криптографічний ключ.

Ключ шифрування має зберігатись у секреті обома сторонами та має бути обраним до початку обміну повідомленнями.

Одним з прикладів симетричного шифрування є алгоритм **гамування**. Він заснований на випадкових числах. Нехай маємо датчик псевдо-випадкових чисел, який працює за деяким визначеним алгоритмом. Часто використовують наступний алгоритм:

$$T_{i+1} = (a \cdot T_i + b) \bmod c,$$

де T_i – попереднє псевдо-випадкове число, T_{i+1} – наступне псевдо-випадкове число, а коефіцієнти a, b, c сталі та добре відомі. Зазвичай $c = 2^n$, де n – розрядність процесору, $a \bmod 4 = 1$, b – непарне. В цьому випадку послідовність псевдо-випадкових чисел має **період** c.

Процес шифрування визначається наступним чином. Шифроване повідомлення представляється у вигляді послідовності слів $S_0, S_1, \ldots$, кожне довжини n, які додаються за модулем 2 зі словами послідовності $T_0, T_1, \ldots$, тобто

$$C_i = S_i \oplus T_i.$$

Послідовність $T_0, T_1, \ldots$ називається **гамою шифру**.

Процес розшифрування заключається в тому, щоби ще раз скласти шифровану послідовність з тією самою гамою шифру:

$$S_i = C_i \oplus T_i.$$

Ключем шифру є початкове значення T_0, яке є секретним та має бути відоме тільки відправнику та адресату шифрованого повідомлення. Якщо період послідовності псевдо-випадкових чисел достатньо великий, щоби гама шифру була довша повідомлення, то дешифрувати повідомлення можна тільки підбором ключа. При збільшенні n експоненційно збільшується криптостійкість шифру.

Описаний метод має суттєвий недолік. Якщо відома хоча б частина висхідного повідомлення, то все повідомлення може бути легко дешифроване. Дійсно, нехай відоме одне висхідне слово S_i. Тоді:

$$T_i = C_i \oplus S_i,$$

і далі вся права частина гами шифру визначається за вказаною формулою датчика псевдо-випадкових чисел.

Більшість симетричних шифрів використовують складну комбінацію великої кількості підстановок та перестановок. Багато таких шифрів виконуються у декілька (до 100) проходів, використовуючи на кожному проході **ключ проходу**. Множина «ключів проходів» для всіх проходів називається **розкладом ключів**. Зазвичай, він утворюється з ключа шляхом виконання над ним певних операцій, в тому числі перестановок та підстановок.

Найважливішими параметрами всіх алгоритмів симетричного шифрування є:

- стійкість;
- довжина ключа;
- кількість раундів;
- довжина блоку, якій оброблюється;
- складність апаратно/програмної реалізації;
- складність перетворень.

До переваг симетричної системи можна віднести:

- порівняно високу швидкість (приблизно на 3 порядки вищу ніж у асиметричних систем);
- простота реалізації (за рахунок більш простих операцій);
- менша необхідна довжина ключа для відповідної стійкості;

Але є також суттєві недоліки, які практично призводять до того, що дана система майже не використовується на даний час.

- складність керування ключами у великій мережі. Це означає квадратичне збільшення кількості ключів, які необхідно генерувати, зберігати, передавати та знищувати у мережі. Для мережі з 10 абонентів потрібно 45 ключів, для 100 – вже 4950, для 1000 – 499500;
- складність обміну ключами. Для застосування симетричної системи необхідно вирішити проблему надійної передачі ключів до кожного абонента, тому що необхідний секретний канал для передачі кожного ключа обом сторонам.

23.3. Асиметричні криптосистеми

Означення 23.2. **Криптографічна система з відкритим ключем** (або асиметрична криптосистема, асиметричне шифрування) – це система шифрування, при якій відкритий ключ передається по відкритому (тобто не захищеному) каналу зв'язку та використовується для шифрування повідомлень. Для розшифрування повідомлень використовується секретний (або приватний) ключ.

Наявність двох ключів – відкритого та закритого – й робить цю систему асиметричною. Відкритий ключ розсилається всім, хто бажає відправляти повідомлення адресату, а приватний ключ зберігається адресатом і не повинен нікому відправлятись. Навіть якщо знати відкритий ключ та все відправлене розшифроване повідомлення, неможливо знайти приватний ключ.

Підхід у системі з відкритим ключем базується на існуванні односторонніх функцій – функцій $f(x)$, для яких по відомому x легко знайти значення функції $f(x)$, але для відомого значення функції $f(x)$ неможливо (або занадто важко) знайти значення аргументу x, тобто обчислити обернену функцію.

Розглянемо один з найвідоміших та найбільш поширених алгоритмів – RSA. Цей алгоритм, зокрема, використовується для передачі даних захищеними каналами зв'язку мережі Інтернет – HTTPS, SSH. Інший відомий алгоритм, який був першим з алгоритмів асиметричного шифрування – алгоритм Діффі-Хелмана (DH).

Алгоритм RSA базується на властивостях простих та взаємно простих чисел, а саме на задачі множення та розкладання складених чисел на прості співмножники. Ця задача є обчислювально однонаправленою. Тобто для заданих простих співмножників знайти складене число дуже легко – необхідно просто перемножити ці співмножники, а для заданого складного числа знайти його прості співмножники – набагато складніше. У реальних

системах, коли використовуються складені числа розмірності в 100 знаків та більше, час, необхідний для розв'язання задачі розкладання числа на прості співмножники, вимірюється роками.

У системі RSA кожен ключ складається з пари цілих чисел. Відкритий та закритий ключі утворюють узгоджену пару, тобто є взаємно оберненими з точки зору задачі шифрування.

Наведемо алгоритм RSA генерації закритого та відкритого ключів.

1. Обрати два випадкових простих числа p та q заданого розміру (наприклад 1024 біт).
2. Обчислити $n = pq$, яке називається **модулем**.
3. Обчислити $\varphi(n) = (p-1)(q-1)$.
4. Обрати ціле число e ($1 \le e \le \varphi(n)$), взаємно просте з $\varphi(n)$. Зазвичай у якості e беруть прості числа, які містять невелику кількість одиничних бітів у двійковому записі (наприклад 17, 257, 65537).

Число e називається **відкритою експонентою**. Занадто малі значення e потенційно можуть послабити криптостійкість шифру.

5. Обчислити число d, яке є мультиплікативно оберненим до числа e за модулем $\varphi(n)$, тобто число, яке задовольняє умові:

$$de = 1 \bmod (\varphi(n))$$

Число d називається **секретною експонентою**.

Пара $P = (e, n)$ публікується в якості відкритого ключа RSA. Пара $S = (d, n)$ відіграє роль закритого (приватного) ключа RSA.

Повідомлення мають бути менші за число n. Якщо повідомлення більше за n, то його б'ють на частини.

Процес обміну повідомленнями від сторони В стороні А полягає у таких кроках:

- сторона А генерує за вказаним алгоритмом пару ключів – $P = (e, n)$ та $S = (d, n)$ – і відправляє відкритим каналом стороні В свій відкритий ключ;
- сторона В шифрує повідомлення M за допомогою ключа $P = (e, n)$:

  $$C = M^e \bmod n$$

  та відправляє його стороні А;
- сторона А приймає зашифроване повідомлення C та розшифровує його своїм закритим ключем $S = (d, n)$:

  $$M' = C^d \bmod n.$$

Теорема 23.1 (без доведення). Шифрування з відкритим ключем є коректним, тобто $M = M'$.

Розглянемо приклад генерації ключів та шифрування/розшифрування повідомлень методом RSA.

1. $p = 3$, $q = 11$.
2. $n = pq = 33$;
3. $\varphi(n) = (p-1)(q-1) = 20$;
4. $e = 7$
5. $d = 3$: $de = 3 \cdot 7 \bmod 20 = 21 \bmod 20 = 1$.

Таким чином, відкритий ключ $P = (7, 33)$, закритий – $S = (3, 33)$.

Нехай повідомлення складається з трьох чисел: $M_1 = 3$, $M_2 = 1$, $M_3 = 2$. Відповідні шифри, отримані за допомогою ключа (7, 33):

$C_1 = 3^7 \bmod 33 = 2187 \bmod 33 = 9$;

$C_2 = 1^7 \bmod 33 = 1 \bmod 33 = 1$;

$C_3 = 2^7 \bmod 33 = 128 \bmod 33 = 29$.

Перевіримо розшифрування закодованих повідомлень закритим ключем (3, 33).

$M'_1 = 9^3 \bmod 33 = 729 \bmod 33 = 3$;

$M'_2 = 1^3 \bmod 33 = 1 \bmod 33 = 1$;

$M'_3 = 29^3 \bmod 33 = 24389 \bmod 33 = 2$.

Розглянемо переваги та недоліки асиметричної криптосистеми.

До переваг можна віднести:

- не потрібно передавати закритий ключ будь-якими каналами зв'язку;
- на противагу симетричній криптосистемі, секретний ключ зберігається тільки у одній стороні;
- у симетричній криптосистемі варто змінювати ключ після кожного сеансу передачі даних; у асиметричній ключі можна тримати незміненими достатньо довгий час;
- у більшості мереж кількість ключів при асиметричному шифруванні набагато менша, ніж при симетричному.

Недоліки:

- хоча повідомлення шифруються надійно, але самі сторони «засвічуються» самим фактом передачі (на чому може бути побудована атака);
- асиметричні алгоритми використовують набагато довші ключі ніж симетричні;
- у чистому вигляді асиметричні системи вимагають значних обчислювальних ресурсів, тому на практиці вони частіше використовуються разом з іншими алгоритмами.

Наведемо декілька слів відносно криптостійкості асиметричного шифрування. Одна з можливих атак на таку систему полягає в наступному. Припустимо, що сторони A та B обмінюються повідомленнями, як це було описано вище. Якщо третя сторона C захоче втрутитись у цей процес, то вона може прослуховувати канал зв'язку між A та B і підставляти свої повідомлення замість оригінальних. Так, спочатку, коли A відправляє B свій відкритий ключ P_A, сторона C може перехопити цей ключ та надіслати до B власний ключ P_C. Тоді B буде кодувати повідомлення за допомогою неправильного ключа P_C. Перехопивши ці повідомлення, C зможе їх розкодувати своїм приватним ключем, а замість оригінального повідомлення направити до A якесь інше некоректне повідомлення, закодоване вже ключем P_A. Описаний тип атаки має назву «men-in-the-middle».

23.4. Цифровий підпис

Також асиметрична система з відкритим ключем використовується у системах цифрових підписів. Цей процес ґрунтується на властивості комутативності операцій шифрування та розшифрування:

$$M = (M^e)^d \bmod n = M^{ed} \bmod n = M^{de} \bmod n = (M^d)^e \bmod n = M.$$

Нехай сторона A відправляє стороні B повідомлення M. Для того, щоб підтвердити, що повідомлення дійсно відправлено від A, ця сторона супроводжує його підписом $C = M^d \bmod n$, таким чином надсилаючи до B пару (M, C). Сторона B, отримавши це повідомлення з підписом, кодує повідомлення M відкритим ключем P_A: $C' = M^e \bmod n$. Якщо $C = C'$, то повідомлення M дійсно прийшло від сторони A неушкодженим. Інакше, повідомлення було пошкоджено або замінено третьою стороною.