Тема 10. ОСНОВНІ ПОНЯТТЯ АСИМЕТРИЧНОЇ КРИПТОГРАФІЇ

1. Основні поняття асиметричних криптосистем

При практичному використанні моделі Шеннона необхідність реалізації захищеного каналу для ключового обміну породжує так звану проблему безпечного розповсюдження ключів. Крім того, виникає проблема підтвердження істинності тієї чи іншої інформації.

Наприклад, це може бути необхідно при перевірці того, що ключ дійсно належить особі від імені якого він надходить у систему, оскільки існує можливість так званого нав'язування ключа.

Обидва ці завдання без використання захищеного каналу зв'язку вдалося вирішити в рамках моделі криптосистеми з відкритим ключем, запропонованої Діффі та Хеллманом у 1976 році.

Відмінність моделі Діффі-Хеллмана від моделі Шеннона в тому, що вона є асиметричною в тому сенсі, що користувачі по відношенню до секретного параметру не рівноправні. Ключ відомий повністю тільки одержувачу повідомлення і являє собою пару *(e,d)* де підключ *e* (відкритий ключ) служить ключем зашифрування, а підключ *d* (секретний, особистий ключ) служить для розшифрування. Ключ *d* відомий тільки одержувачі повідомлень, які відправники повинні шифрувати використовуючи ключ *e*. Такі криптосистеми називаються *асиметричними* або *системами з відкритими ключами*.

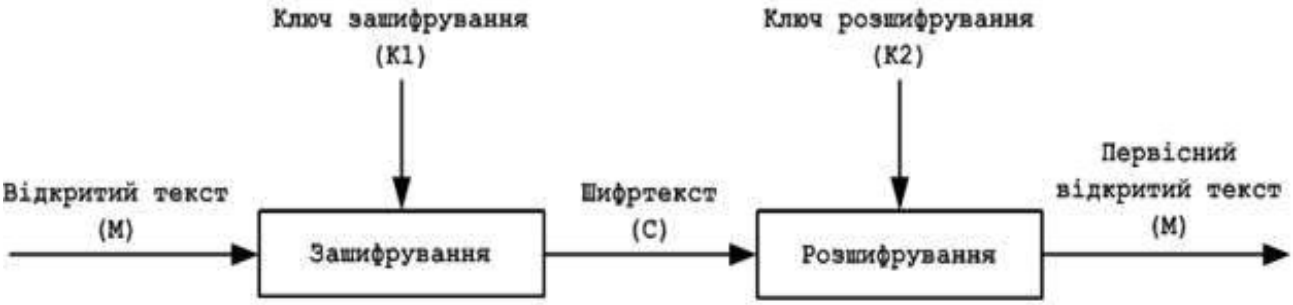
В асиметричних криптосистемах процедури прямого і зворотного криптоперетворення виконуються на різних ключах і не мають між собою очевидних і легко відсліджуваних зв'язків, що дозволяють за одним ключем визначити інший. В такій схемі знання тільки ключа зашифрування не дозволяє розшифрувати повідомлення, тому він зазвичай публікується учасником обміну для того, щоб будь-хто бажаючий міг послати йому зашифроване повідомлення.

Принцип функціонування асиметричної криптосистеми полягає у наступному:

- ✓ користувач А генерує два ключі – відкритий і секретний, і передає відкритий ключ по незахищеному каналу користувачу Б;
- ✓ користувач Б шифрує повідомлення, використовуючи відкритий ключ шифрування використовуючи відкритий ключ шифрування користувача А;
- ✓ користувач Б посилає зашифроване повідомлення користувачу А по незахищеному каналу;
- ✓ користувач А отримує зашифроване повідомлення і дешифрує його, використовуючи свій секретний ключ

Пара (відкритий ключ; секретний ключ) обчислюється за допомогою спеціальних алгоритмів, при чому жоден ключ не може бути виведений з другого.

Схема асиметричного алгоритму:



Слід зауважити, що ключі K_1 та K_2 не обов'язково мають бути різними, але в тому випадку, коли вони співпадають, втрачається багато переваг алгоритму з відкритим ключем.

Іноді повідомлення зашифровуються закритим ключем, а розшифровуються − відкритим. Від часу винайдення криптографії з відкритим ключем було запропоновано безліч криптографічних алгоритмів з відкритими ключами. Багато з них не є стійкими, а з тих які є, багато не придатних для практичної реалізації. Або вони використовують дуже великий ключ, або розмір отриманого шифротексту набагато перевищує розмір відкритого тексту.

Небагато алгоритмів є безпечними і практичними. Зазвичай ці алгоритми засновані на одній з важких проблем: деякі з них підходять тільки для

розподілу ключів, інші для шифрування, і треті корисні тільки для цифрових підписів.

Прикладами криптосистем з відкритим ключем є:

- ✓ Elgamal (названа на честь автора Ельгамаля);
- ✓ RSA (названа на честь винахідників Рона Рівеста, Аді Шаміра і Леонарда Адлмана);
- ✓ Diffie-Hellman і DSA (Digital Signature Algorithm) (винайдений Девідом Кравіцом);
- ✓ Rabin (Рабіна).

Але тільки три алгоритми добре працюють як при шифруванні так і для цифрового підпису: RSA, Elgamal, Rabin. Усі ці алгоритми повільні. Розшифровують і зашифровують данні набагато повільніше, ніж симетричні алгоритми. Зазвичай їх швидкість недостатня для шифрування великих обсягів даних.

Гібридні криптосистеми дозволяють прискорити події: для шифрування повідомлень використовується симетричний алгоритм з випадковим ключем, а для розшифрування алгоритм з відкритим ключем.

Головне досягнення асиметричного шифрування в тому, що воно дозволяє людям, що не мають існуючої домовленості про безпеку, обмінюватися секретними повідомленнями. Необхідність відправникові й одержувачеві погоджувати таємний ключ по спеціальному захищеному каналі цілком відпала. [10, 12, 14]

2. RSA – криптографічна система з відкритим ключем.

Незважаючи на досить велику кількість різних асиметричних шифросистем, найпопулярніша криптосистема RSA, яка розроблена в 1977 р. і набула назву на честь Рона Рівеста, Аді Шаміра і Леонарда Ейдельмана.

Систему побудовано на факті, що добуток великих простих чисел здійснюється легко, проте розкладання на множники добутку двох таких чисел практично неможливе. Доведено (теорема Рабіна), що розкриття шифру RSA еквівалентно такому розкладу. Тому для довільної довжини ключа дають

нижню оцінку кількості операцій для розкриття шифру, а з урахуванням продуктивності сучасних комп'ютерів оцінити і необхідний для цього час.

Можливість гарантовано оцінити захищеність алгоритму RSA стала однією з причин популярності цієї криптосистеми на фоні десятків інших схем. Тому алгоритм RSA використовується в банківських комп'ютерних мережах, особливо для роботи з віддаленими клієнтами (обслуговування кредитних карток).

Сьогодні алгоритм RSA використовується в багатьох стандартах, серед яких SSL, S – HHTP, S – MIME, S/WAN, STT і PCT.

Розглянемо математичні результати, які лежать в основі цього алгоритму.

Теорема 1. (*Мала теорема Ферма, згадувалася раніше*)

Якщо p – просте число, то $x^{p-1} = 1 \ (mod\ p)$ для будь – якого x, взаємно простого з p, і $x^{p} = x \ (mod\ p)$ для будь – якого x.

Визначення. Функцією Ейлера $\varphi(n)$ називається число додатних цілих, менших від n і простих відносно n.

N	2	3	4	5	6	7	8	9	10	11	12
$\varphi(n)$	1	2	2	3	2	6	4	6	4	10	4

Теорема 2.

Якщо $n=p\cdot q$ (p і q – відмінні одне від одного прості числа), то $\varphi(n)=(p-1)(q-1)$.

Теорема 3.

Якщо $n=p\cdot q$ (p і q – відмінні один від одного прості числа), x – взаємно просте з p і q, то $x^{\varphi(n)} = 1 \pmod n$.

Наслідок.

Якщо $n=p\cdot q$, (p і q – відмінні одне від одного прості числа), e – просте відносно $\varphi(n)$, то відображення $E_{e,n}: x \to x^{e} \pmod n$ буде взаємно однозначним.

Відомий і той факт, що коли e – просте відносно $\varphi(n)$, то існує ціле d – таке, що $ed = 1 \pmod{\varphi(n)}$.

На цих математичних фактах і заснований популярний алгоритм RSA.

Нехай $n=p\cdot q$, де p і q – різні прості числа. Якщо e і d задовольняють рівняння $ed = 1 \pmod{\varphi(n)}$, то відображення $E_{e,n}$ і $E_{d,n}$ будуть інверсіями $E_{d,n}$ і $E_{e,n}$, і легко обчислюються, якщо відомі e,d,p,q. Якщо відомі e і n, а p і q невідомі, то $E_{e,n}$ є односторонньою функцією і знаходження $E_{d,n}$ за заданим n рівнозначно розкладанню n. Якщо p і q – достатньо великі прості, то розкладання n практично нездійсненне. Це і закладено в основу системи шифрування RSA.

Користувач i вибирає пару різних простих p_i і q_i та розраховує пару цілих (e_i, d_i), які є простими відносно $\varphi(n)$, де $n_i=p_iq_i$. Довідкова таблиця містить ключі $\{(e_i,n_i)\}$. Припустимо, що вихідний текст $x=(x_0,x_1,\ldots,x_{n-1})$, $x \in Z_n$, $0 \le i < n$.

Користувач i зашифровує текст при передачі його користувачу j, застосовуючи до n відображення $E_{di,ni}$: $N \rightarrow E_{di,ni}$ $n=n'$.

Користувач j проводить дешифрування n', застосувавши $E_{ei,ni}$: $N' \rightarrow E_{ei,ni}$ $n'= E_{ei,ni}E_{di,ni}n=n$.

Самоочевидно, що для того, щоб знайти інверсію $E_{di,ni}$ до $E_{ei,ni}$, потрібно знати множники $n=p_iq_i$. Час виконання найкращих з відомих алгоритмів розкладу при $n=10^{100}$ на сьогодні виходить за межі реальних технічних можливостей. [7, 9]

3. Застосування алгоритму RSA.

Приклад. Зашифруємо повідомлення CAB. Для простоти використаємо маленькі числа (на практиці застосовують значно більші).

1. Виберемо p=3 і q=11.

2. Визначимо n=3·11=33.

3. Знайдемо $(p-1)(q-1)$=20. В якості d візьмемо взаємно просте з 20, наприклад d=3.

4. Виберемо число e. В ролі такого числа виступає будь – яке число, для якого задовольняється співвідношення $(e\cdot 3) \pmod{20}$=1, наприклад 7.

5. Запишемо шифроване повідомлення як послідовність цілих чисел за допомогою відображення: A→1, B→2, C→3. Тоді повідомлення набуде вигляду (3,1,2). Зашифруємо за допомогою ключа {7,33}. ШТ1=(3^7)(mod 33)=2187(mod 33)=9,

ШТ2=(1^7) (mod 33)=1(mod 33)=1,

ШТ3=(2^7) (mod 33)=128(mod 33)=29.

6. Розшифруємо одержане зашифроване повідомлення (9,1,29) на основі закритого ключа {3,33}:

ВТ1=(9^3)(mod 33)=729(mod33)=3,

ВТ2=(1^3)(mod 33)=1(mod33)=1,

ВТ1=(29^3)(mod 33)=24389(mod33)=2.

Отже, в реальних системах алгоритм RSA реалізується в такий спосіб: кожен користувач вибирає два великих простих числа p і q та, відповідно до описаного алгоритму, вибирає два простих числа e і d. Як результат добутку перших двох чисел (p і q) встановлюється n.

Відкритий ключ утворює $\{e,n\}$, а $\{d,n\}$ – закритий (хоч можна і навпаки). Відкритий ключ публікується і доступний кожному бажаючому надіслати власнику ключа повідомлення, яке зашифроване вказаним алгоритмом. Після шифрування, повідомлення неможливо розкрити за допомогою відкритого ключа. Власник закритого ключа має можливість розшифрувати прийняте повідомлення. [7, 9]

4. Система Діффі-Хеллмана та Ель – Гамаля.

Ця система є альтернативою до RSA і при однаковому значенні ключа забезпечує таку ж криптостійкість.

На відміну від RSA метод Ель – Гамаля заснований на проблемі дискретного логарифма. Цим він подібний до алгоритму Діффі – Хеллмана. Якщо підносити число до степеня в скінченному полі досить легко, то відновити аргумент за значенням (знайти логарифм) досить складно.

Система Ель-Гамаля

Основу системи становлять параметри p і g – числа, перше з яких – просте, а друге – ціле.

Далі Аліса генерує секретний ключ a і обчислює відкритий ключ $y=g^a \bmod p$. Якщо Борис хоче надіслати Алісі повідомлення m, то він вибирає випадкове число k, менше, ніж p, і обчислює $y_1=g^k \bmod p$ та $y_2=m \oplus y^k$, де $\oplus$ означає побітове додавання за модулем 2. Потім Борис надсилає (y_1,y_2) Алісі. Аліса, одержавши зашифроване повідомлення, відновлює його: $m=(y_1^a \bmod p) \oplus y_2$.

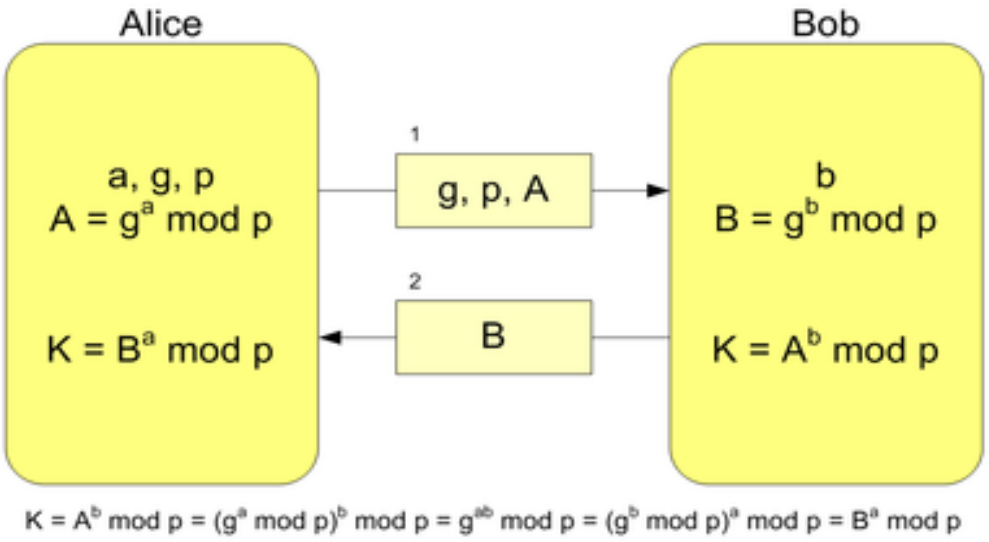
Система Діффі-Хеллмана

Схема обміну ключами Діффі-Хеллмана, винайдена в 1976 році при співробітництві Уитфилда Діффі і Мартіна Хеллмана, під сильним впливом роботи Ральфа Меркля (Ralph Merkle) про систему розповсюдження публічних ключів, стала першим практичним методом для отримання загального секретного ключа при спілкуванні через незахищений канал зв'язку. Для забезпечення стійкості, за порадою Джона Гілла (John Gill), була використана проблема дискретного логарифмування. За кілька років до цього ця ж схема була винайдена Малькольмом Вільямсоном з англійського штабу урядового зв'язку, але залишалася в секреті до 1997 року.

Опис алгоритму

Припустимо, що обом абонентам відомі деякі два числа g і p (наприклад, вони можуть бути “зашиті” в програмне забезпечення), які не є секретними і можуть бути відомі також іншим зацікавленим особам. Для того, щоб створити невідомий більш нікому секретний ключ, обидва абоненти генерують великі випадкові числа: перший абонент – число a, другий абонент – число b. Потім перший абонент обчислює значення $A = g^a mod\ p$ і пересилає його другому, а другий обчислює $B = g^b mod\ p$ і передає першому. Передбачається, що зловмисник може отримати обидва цих значення, але не модифікувати їх (тобто у нього немає можливості втрутитися в процес передачі). На другому етапі перший абонент на основі наявного в нього a і отриманого по мережі B обчислює значення $B^a mod\ p = g^{ab} mod\ p$, а другий абонент на основі наявного в нього b і отриманого по мережі A обчислює значення $A^b mod\ p = g^{ab} mod\ p$.

Неважко побачити, що в обох абонентів вийшло одне і те ж число: $K = g^{ab} mod\ p$. Його вони і можуть використовувати в якості секретного ключа, оскільки тут зловмисник зустрінеться з практично нерозв'язною (за розумний час) проблемою обчислення $g^{ab} mod\ p$ по перехоплених $g^a mod\ p$ і $g^b mod\ p$, якщо числа p, a, b вибрані достатньо великими.



Описані вище дії зображені на даному рисунку.

При роботі алгоритму, кожна сторона:

1. Генерує випадкове натуральне число a – *закритий ключ*.
2. Спільно з віддаленою стороною встановлює *відкриті параметри* p і g (зазвичай, значення p і g генеруються на одній стороні і передаються іншій), де

   p є *випадковим простим числом*

   g є *первісним коренем по модулю* p
3. Обчислює *відкритий ключ* A, використовуючи перетворення над *закритим ключем* $A = g^a\ mod\ p$.
4. Обмінюється *відкритими ключами* з віддаленою стороною.
5. Обчислює *загальний секретний ключ* K, використовуючи відкритий ключ віддаленої сторони B і свій закритий ключ a

   $K = B^a\ mod\ p$

   K виходить рівним з обох сторін, тому що:

   $B^a\ mod\ p = (g^b\ mod\ p)^a mod\ p = g^{ab} mod\ p = (g^a mod\ p)^b mod\ p = A^b mod\ p$

У практичних реалізаціях, для a і b використовуються числа порядку 10^{100} і p порядку 10^{300}. Число g не обов’язково має бути великим і зазвичай має значення в межах першого десятка.

Необхідно зазначити, що алгоритм Діффі-Хеллмана працює тільки на лініях зв'язку, надійно захищених від модифікації. Якби він був застосовний на будь-яких відкритих каналах, то давно зняв би проблему розповсюдження ключів і, можливо, замінив собою всю асиметричну криптографію. Однак, у тих випадках, коли в каналі можлива модифікація даних, з'являється можливість атаки людина посередині. Атакуючий замінює повідомлення переговорів про ключ на свої власні і, таким чином, отримує два ключі – свій для кожного з законних учасників протоколу. Далі він може перешифровувати листування між учасниками, своїм ключем для кожного, і, таким чином, ознайомитися з їх повідомленнями, залишаючись непоміченим. [2, 7, 9, 12, 13]