АСИМЕТРИЧНІ КРИПТОАЛГОРИТМИ

Загальні відомості про асиметричні криптоалгоритми

Симетричні криптосистеми, що мають множину переваг, володіють одним суттєвим недоліком, що виникає, коли спілкування між собою проводять не три-чотири людини, а сотні і тисячі користувачів. В цьому випадку для кожної пари, що переписується між собою, необхідно створювати свій секретний симетричний ключ. Це у результаті призводить до існування в системі з N користувачів $N^2/2$ ключів, що є дуже великою кількістю. Крім того, при порушенні конфіденційності будь-якої робочої станції зловмисник отримує доступ до всіх ключів цього користувача і може відправляти, нібито від його імені, повідомлення всім абонентам, з якими "жертва" вела листування.

Своєрідним рішенням цієї проблеми є використання асиметричної криптографії. Ця область криптографії дуже молода в порівнянні з іншими представниками. Перша така схема, що мала прикладну цінність, була запропонована всього близько 40 років тому. За цей час асиметрична криптографія перетворилася на один з основних напрямів криптології, і використовується в сучасному світі також часто, як і симетричні схеми.

Асиметрична криптографія спочатку розроблялась як засіб передачі повідомлень від одного об'єкту до іншого (а не для конфіденційного зберігання інформації, яке забезпечують тільки симетричні алгоритми). Тому наступний зміст використовує терміни "відправник" – особа, шифрує, а потім відправляє інформацію за незахищеним каналом і "одержувач" – особа, що приймає і відновлює інформацію в її початковому вигляді. Основна ідея асиметричних криптоалгоритмів полягає в тому, що для шифрування повідомлення використовується один ключ, а при дешифруванні – інший.

Крім того, процедура шифрування обрана так, що вона є незворотною навіть за відомим ключом шифрування – це друга необхідна умова асиметричної криптографії. Тобто, знаючи ключ шифрування і зашифрований текст, неможливо відновити початкове повідомлення – прочитати його можливо тільки за допомогою другого ключа – ключа дешифрування. Т. ч., ключ шифрування для відправки листів будь-якій особі можливо взагалі не приховувати – знаючи його все одно неможливо прочитати зашифроване повідомлення. Тому, ключ шифрування називають в асиметричних системах "відкритим ключем", а ключ дешифрування одержувачу повідомлень необхідно тримати в секреті – він називається "закритим ключем". На питання: "Чому, знаючи відкритий ключ, не можливо обчислити закритий ключ ?" висуває третю необхідну умову щодо асиметричної криптографії – алгоритми шифрування і дешифрування створюються так, щоб знаючи відкритий ключ, неможливо обчислити закритий ключ.

У цілому система листування при використанні асиметричного шифрування виглядає наступним чином. Для кожного з N абонентів, що ведуть листування, обрана своя пара ключів: "відкритий" Ej і "закритий" Dj, де j – номер абонента. Всі відкриті ключі відомі всім користувачам мережі, кожний закритий ключ, навпаки, зберігається тільки у того абонента, якому він належить. Якщо абонент, наприклад під номером 7, збирається передати інформацію абоненту під номером 9, він шифрує дані ключем шифрування E_9 і відправляє її абоненту 9. Не зважаючи на те, що всі користувачі мережі знають ключ E_9 і, можливо, мають доступ до каналу, за яким передається зашифроване повідомлення, вони не можуть прочитати початковий текст, оскільки процедура

шифрування незворотна за відкритим ключем. Тільки абонент №9, одержавши повідомлення, проводить над ним перетворення за допомогою відомого тільки йому ключа D_9 і відновлює початковий текст. Якщо повідомлення потрібно відправити в протилежному напрямі (від абонента 9 до абонента 7), то потрібно буде використовувати вже іншу пару ключів (для шифрування ключ E_7, а для дешифрування – ключ D_7).

_Т. ч., по-перше, в асиметричних системах кількість існуючих ключів пов'язана з кількістю абонентів лінійно (в системі з N користувачів використовуються 2N ключів), а не квадратична, як в симетричних системах. По-друге, при порушенні конфіденційності k-ої робочої станції зловмисник довідається тільки ключ Dk : це дозволяє йому читати всі повідомлення, що надходять до абоненту k, але не дозволяє видавати себе за нього при відправці листів. Окрім цього, асиметричні криптосистеми володіють ще і іншими корисними можливостями.

Алгоритм RSA

Алгоритм RSA було запропоновано трьома математиками; Рональдом Рівестом (R. Rivest), Аді Шаміром (A. Shamir) і Леонардом Адльманом (L. Adleman) у 1977-78 роках.

Першим етапом асиметричного алгоритму є створення пари ключів: відкритого і закритого і розповсюдження відкритого ключа всім бажаючим. Для алгоритму RSA етап створення ключів складається з наступних операцій:

1. обираються два прості (!) числа p і q;
2. обчислюється їх твір n = p q;
3. обирається довільне число e (e < n), таке, що НОД(e, (p - 1)(q - 1)) = 1, тобто e повинне бути взаємно простим з числом (p - 1)(q - 1);
4. методом Евкліду розв'язується в цілих числах (!) рівняння e d + (p - 1)(q - 1) y = 1, де невідомими є змінні d і y – метод Евкліда якраз і знаходить множину пар (d, y), кожна з яких є рішенням рівняння в цілих числах;
5. два числа (e, n) – публікуються як відкритий ключ;
6. число d зберігається в таємниці – це і є закритий ключ, який дозволить читати всі повідомлення, що зашифровані за допомогою пари чисел (e, n).

Для шифрування виконуються наступні дії:

7. відправник розбиває своє повідомлення на блоки, що дорівнюють $k = [\log_2(n)]$ біт, де квадратні дужки позначають отримання цілої частини від дробового числа.
8. такий блок може бути інтерпретовано як число з діапазону $(0; 2^k - 1)$. Для кожного такого числа (m_i) обчислюється вираз $c_i = ((m_i)^e) \bmod n$. Блоки c_i і є зашифроване повідомлення, їх можливо передавати за відкритими каналами, т. я. операція зведення в ступінь за модулем простого числа, є незворотнею математичною задачею. Зворотна їй задача носить назву "логарифмування в кінцевому полі" і є на декілька порядків складнішою задачею. Тобто навіть якщо зловмисник знає числа e і n, то за c_i прочитати початкові повідомлення m_i він не може ніяк, окрім як повним перебором значень m_i.

На приймальному боці процес дешифрування виконується за допомогою числа d, що зберігається в таємниці. За доказом теореми Ейлера слід, що якщо число n надано у вигляді двох простих чисел p і q, то для будь-якого x має місце рівність:

$$(x^{(p-1)(q-1)}) \bmod n = 1.$$

Така формула і застосовується з метою дешифрування RSA - повідомлень. Для цього підносяться обидві її частини до ступеня (-y) :

$$(x^{(-y)(p-1)(q-1)}) \bmod n = 1^{(-y)} = 1.$$

Далі помножуються обидві її частини на x :

$$(x^{(-y)(p-1)(q-1)+1}) \bmod n = 1x = x.$$

При створенні відкритого і закритого ключів, за допомогою алгоритму Евкліда, обирається таке d, що ed + (p - 1)(q - 1)y = 1, тобто ed = (-y)(p - 1)(q - 1) + 1. А отже в виразі $c_i = ((m_i)^e) \bmod n$ можливо замінити показник ступеня на число (ed). Звідси одержується $(x^{ed}) \bmod n = x$. Тобто для того, щоб прочитати повідомлення $c_i = ((m_i)^e) \bmod n$ достатньо піднести його за ступенем d по модулю m :

$$((c_i)^d) \bmod n = ((m_i)^{ed}) \bmod n = m_i.$$

Але операції зведення в ступінь великих чисел достатньо трудомісткі, навіть якщо вони проводяться по оптимізованих за часом алгоритмах. Тому весь текст повідомлення кодується блоковим шифром (що є набагато швидшим), але з використанням ключа сеансу, а ключ сеансу шифрується асиметричним алгоритмом за допомогою відкритого ключа одержувача і заноситься до початку файлу.

Технології цифрових підписів

Теорія асиметричного шифрування дозволяє вирішувати ще одну проблему інформаційної безпеки – перевірку достовірності автора повідомлення. Для вирішення цієї проблеми за допомогою симетричної криптографії була розроблена дуже трудомістка і складна схема. В той же час за допомогою, наприклад, того ж алгоритму RSA, створити алгоритм перевірки достовірності автора і незмінності повідомлення надзвичайно просто.

Припустимо, що потрібно передати будь-який текст, не обов'язково секретний, але важливо те, щоб в нього при передачі за незахищеним каналом не були внесені зміни. До таких текстів звичайно відносяться різні розпорядження, довідки, і тому подібна документація, що не представляє таємниці. Обчислимо від цього тексту будь-яку хеш - функцію – це буде число, яке більш менш унікально характеризує такий текст.

Хоча можливо знайти і інший текст, який дає те ж саме значення хеш - функції, але змінити в початковому тексті десять - двадцять байт так, щоб текст залишився повністю осмисленим, та ще і змінився у вигідний зловмиснику бік (наприклад, зменшив суму оплати у два рази) – надзвичайно складно. Саме для усунення цієї можливості хеш - функції створюють такими ж складними як і криптоалгоритми – якщо текст з таким же значенням хеш - функції можливо буде підібрати тільки методом повного перебору, а множина значень складатиме як і для блокових шифрів $2^{32} – 2^{128}$ можливих варіантів, то для пошуку подібного тексту зловмиснику "потрібно" також мільйони років.

Таким чином, якщо можливо передати одержувачу захищеним від зміни методом хеш - суму від тексту, що пересилається, то у нього завжди буде можливість самостійно обчислити хеш - функцію від тексту вже на приймальній стороні і звірити її з надісланим значенням. Якщо хоча б один біт в обчисленій їм самостійно контрольній сумі тексту не співпадає з відповідним бітом в одержаному хеш - значенні, то, текст при передаванні зазнав несанкціоновані зміни.

Наведемо призначену до передачі хеш - суму у вигляді декількох k - бітових блоків h_i, де k – це розмір повідомлень за алгоритмом RSA. Обчислимо над кожним блоком значення:

$$s_i = ((h_i)^d) \bmod n ,$$

де d – закритий ключ відправника. Тепер повідомлення, що складається з блоків s_i можливо передавати по мережі. Ніякій небезпеці за відомими h_i і s_i знайти секретний ключ немає – це настільки ж складна задача, як і задача "логарифмування в кінцевому полі". А ось будь-який одержувач повідомлення може легко прочитати початкове значення h_i, виконавши операцію:

$$((s_i)^e) \bmod n = ((h_i)^{de}) \bmod n = h_i .$$

Відкритий ключ (e, n) є у всіх, а те, що зведення будь-якого числа в ступінь (ed) за модулем n дає початкове число, було доведено вище. При цьому ніхто інший, окрім відправника, не знаючи його закритого ключа d не може, змінивши текст, а отже, і хеш - суму, обчислити такі s'_i, щоб при їх зведенні в ступінь e вийшла хеш-сума h'_i, яка буде дорівнювати хеш - сумі фальсифікованого тексту.

Таким чином, маніпуляції з хеш - сумами тексту представляють з себе "асиметричне шифрування навпаки" : при відправці використовується закритий ключ відправника, а для перевірки повідомлення – відкритий ключ відправника. Подібна технологія одержала назву "електронний підпис". Інформацією, яка унікально ідентифікує відправника (його віртуальним підписом), є закритий ключ d. Жодна людина, що не володіє цією інформацією, не може створити таку пару (текст, s_i), щоб алгоритм перевірки дав би позитивний результат.

Подібний обмін місцями відкритого і закритого ключів для створення з процедури асиметричного шифрування алгоритму електронного підпису можливий тільки в тих системах, де виконується властивість комутативності ключів. Для інших асиметричних систем алгоритм електронного підпису або значно відрізняється від базового, або ж взагалі не може бути реалізовано.

Механізм розповсюдження відкритих ключів

Хоча асиметричні криптосистеми позбавлені одного з найголовніших недоліків симетричних алгоритмів – необхідності попереднього обміну сторонами секретним ключем за захищеною схемою (наприклад, особисто або за допомогою повіреного кур'єра), недостатнім вільне розповсюдження відкритих ключів. Але припустимо є відправник повідомлення і отримувач. Для того, щоб відправити зашифроване повідомлення, відправник повинен дізнатися відкритого ключа отримувача. Якщо отримувач не передав цей ключ відправнику особисто на змінному носії, то відправник отримав ключ з інформаційної мережі. При цьому не існує гарантії, що цей набір байт є саме відкритим ключем отримувача. Адже зловмисник може згенерувати довільну пару (закритий ключ, відкритий ключ), потім активно поширювати або пасивно підміняти при запиті відправника відкритий ключ отримувача на створеним їм. В цьому випадку при відправці повідомлення 1) відправник шифрує його тим ключем, який він вважає, що належить отримувачу; 2) зловмисник, перехопивши повідомлення дешифрує його парним закритим ключем, прочитає і більш того 3) може переслати далі, зашифрувавши

дійсно вже відкритим ключем отримувача. Так само, але за інверсною схемою, він може підмінити і електронний підпис відправника під його листом.

Таким чином, якщо між відправником і одержувачем немає конфіденційної схеми передачі асиметричних ключів, то виникає серйозна небезпека появи зловмисника-посередника. Але асиметрична криптографія знайшла спосіб дуже значного зниження ризику подібної атаки. Теза, що між відправником і одержувачем немає гарантованої лінії зв'язку, не є повністю вірною. Поза сумнівом у будь-кого знайдеться троє-четверо надійних знайомих в столиці або закордоном, у них в свою чергу також знайдеться безліч знайомих в багатьох точках країни і світу. Врешті-решт, кожна особа користуєтеся програмним забезпеченням фірм, представництва яких знаходяться в тій країні або в тому місті, куди необхідно відправити листа. Проблема тільки в тому, що починаючи, з другої від відправника ланки людина особисто не знаєте його і вірогідність того, що вона, або більш того, велика компанія, будуть якось мотивовані, дуже мала.

Але, якщо безліч однодумців об'єднаються з метою створити надійну мережу розповсюдження ключів, то це буде ймовірно. А сама асиметрична криптографія допоможе їм в цьому таким чином: в дійсності нікуди ходити зі змінним носієм не потрібно, для передання відкритого ключа відправника одержувачу. Адже особа спілкуєтеся з власним знайомим, тому у відправника є його відкритий ключ, одержаний будь-яким надійним способом. А отже, він може передати відправнику цей відкритий ключ одержувача, підписавши повідомлення власним електронним підписом. А від відправника вимагається лише надіслати цей ключ далі по ланцюжку у напрямі отримувача, підписавши вже власним електронним підписом. Таким чином, минувши дещо перепідписань, відкритий ключ дійде від місця відправлення до місця вимоги за надійним шляхом. У принципі від відправника навіть може не вимагатися ніяких дій – йому потрібно лише встановити на власній ЕОМ спеціальний сервер розповсюдження ключів, і він всі зазначені дії виконуватиме автоматично.

На сьогоднішній день не існує єдиної мережі розповсюдження відкритих ключів, що пов’язано з «війною» стандартів. Тепер розвиваються декілька незалежних систем, але жодна з них не одержала переваги над іншими.

Необхідно відзначити, що ланцюжок розповсюдження ключів в реальних випадках не дуже великий. Звичайно він складається з двох-чотирьох ланок. Із залученням до процесу розповсюдження ключів крупних фірм-виробників програмних продуктів він стає ще коротшим. Якщо на носії з придбаним програмним забезпеченням вже знаходиться відкритий ключ цієї фірми, а сама вона має крупний ринок збуту, то ланцюжок складатиметься або з однієї ланки (якщо ПЗ цієї ж фірми встановлено і у потенційних отримувачів), або з двох (другим стане будь-який інший концерн, ПЗ якого встановлено у отримувача, між собою всі великі компанії обмінялися ключами електронних підписів достатньо давно). Відкритий ключ, підписаний будь-якою третьою стороною, називається завіреним за допомогою сертифікату. Сертифікатом називається інформаційний пакет, що містить будь-який об'єкт (зазвичай ключ) і електронний підпис, підтверджуючий цей об'єкт від імені чиєї-небудь особи.

Обмін ключами за алгоритмом Діффі - Хеллмана

Алгоритм названо за прізвищами його розробників Діффі (Diffie) і Хеллмана (Hellman). Припустимо, що двом абонентам необхідно здійснити конфіденційне листування, а в їх розпорядженні немає спочатку обумовленого конфіденційного ключа. Проте, між ними існує канал, захищений від модифікації, тобто дані, що надходять за ним, можуть прослуховуватись, але не можуть бути змінені. В цьому випадку дві сторони можуть створити однаковий таємний ключ, жодного разу не передавши його по мережі, за наступним алгоритмом.

Припустимо, що обом абонентам відомо деякі два числа v і n. Вони, втім, відомі і всій решті зацікавлених осіб. Наприклад, вони можуть бути просто фіксовано «зашиті» до програмного забезпечення. Для того, щоб створити секретний ключ, обидва абоненти генерують випадкові або псевдовипадкові прості числа: перший абонент – число x, другий абонент – число y. Потім перший абонент обчислює значення $(v^x) \bmod n$ і надсилає його другому, а другий обчислює значення $(v^y) \bmod n$ і надсилає його першому. Якщо зловмисник отримає обидва ці значення, то модифікувати їх (втрутитися в процес передачі) не він не зможе. На другому етапі перший абонент на основі того, що у нього є x і одержане по мережі значення $(v^y) \bmod n$ обчислює значення $(((v^y) \bmod n)^x) \bmod n$, а другий абонент на основі власного значення y і одержаного по мережі $(v^x) \bmod n$ обчислює значення $(((v^x) \bmod n)^y) \bmod n$. Така операція зведення в ступінь може здійснюватись через операцію узяття модуля за простим числом (комутативна у кінцевому полі), тобто у обох абонентів розраховано одне і те ж число: $((v^{xy}) \bmod n$. Його вони і можуть використовувати як секретний ключ, оскільки тут зловмисник знову зустрінеться з проблемою RSA при спробі з'ясувати за перехопленими $(v^x) \bmod n$ і $(v^y) \bmod n$ власно числа x і y – це дуже ресурсоємна операція, якщо числа v, n, x, y обрано достатньо великими.

Алгоритм Діффі - Хеллмана працює тільки за лініями зв'язку, що надійно захищені від модифікації. Якщо його застосовувати на будь-яких відкритих каналах, то це вирішило б проблему розповсюдження ключів і, можливо, замінило б собою всю асиметричну криптографію. Проте, в тих випадках, коли в каналі можлива модифікація даних, з'являється очевидна можливість втручання до процесу генерації ключів "зловмисника - посередника" за тією ж самою схемою, що і для асиметричної криптографії.

Загальна схема асиметричної криптосистеми

Загальна схема асиметричної криптосистеми зображена на рис. 10.

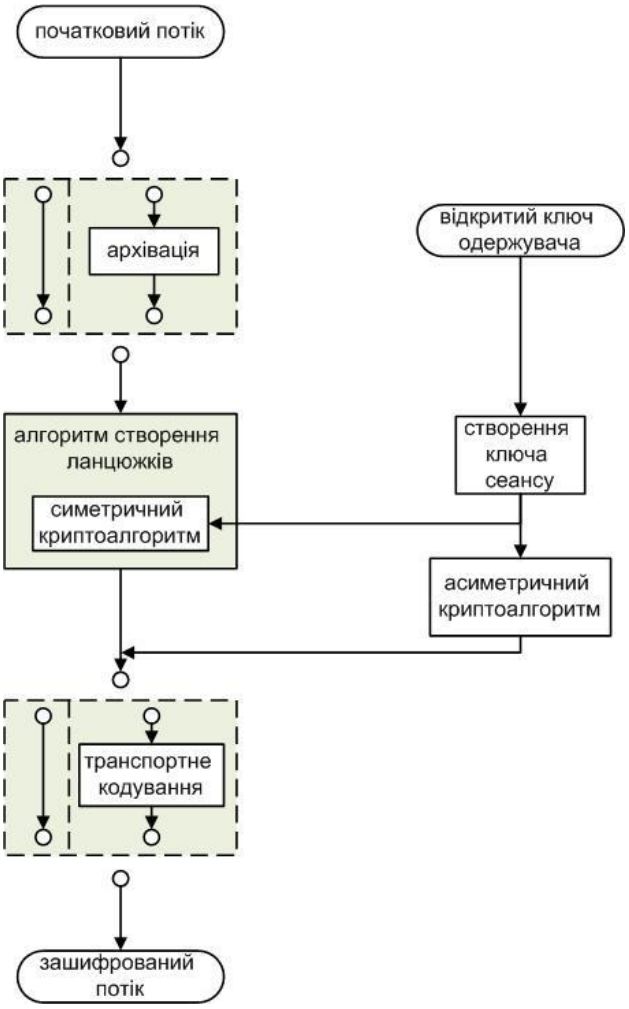

Рис.10. - Узагальнена схема асиметричної криптосистеми

За структурою вона практично ідентична симетричній криптосистемі (рис. 9) з ключем сеансу.