РОЗДІЛ 2. Теорія чисел і криптографія

Тема 8. Криптосистеми з відкритим ключем

План лекції

- **Асиметричні криптосистеми (криптосистеми з відкритим ключем)**
  - **Симетричні та асиметричні криптосистеми**
  - **Система шифрування *RSA***
  - **Обґрунтування коректності системи *RSA***
  - **Чому криптосистема *RSA* підходить для криптографії з відкритим ключем?**
- **Криптографічні протоколи**
  - **Обмін ключем**
  - **Цифрове підписання**
- **Довідка при сучасні симетричні криптосистеми**

Криптосистеми з відкритим ключем.

Симетричні та асиметричні криптосистеми

Усі класичні шифри, зокрема зсуву, афінні – це криптосистеми з секретним ключем. Відмінна особливість таких криптосистем полягає в тому, що кожний, кому відомий шифрувальний ключ, швидко може знайти ключ дешифрувальний. Отже, знання як зашифровано повідомлення з використанням секретного ключа дає змогу розшифрувати повідомлення, яке було зашифроване за допомогою цього ключа. Тому класичні криптосистеми називають *симетричними*.

Нині широко застосовують криптосистеми з відкритим ключем. Такі системи називають *асиметричними* – для шифрування й розшифрування вони використовують різні ключі. Ключ, що його використовують для шифрування, є відкритим (публічним) і може бути повідомлений усім бажаючим надіслати секретне повідомлення. Ключ для розшифрування – закритий (секретний, приватний) і зберігається таємно одержувачем шифрованих повідомлень. Навіть знання всього зашифрованого повідомлення й відкритого ключа не дає змоги дешифрувати повідомлення (без знання закритого ключа).

Система шифрування RSA

У 1976 р. дослідники з Массачусетського технологічного інституту Рональд Райвест (Ronald Rivest), Аді Шамір (Adi Shamir) та Леонард Адлеман (Leonard Adleman) запропонували систему шифрування з відкритим ключем, нині відому як **система RSA**, за першими буквами прізвищ її винахідників.

Коротко опишемо цю систему шифрування.

•**1.** Одержувач повідомлень здійснює генерування відкритого ключа (пара чисел n та e) і закритого ключа (число d). Для цього:

- вибирає два простих числа p і q;
- обчислює першу частину відкритого ключа $n = pq$;
- визначає другу частину відкритого ключа – вибирає невелике непарне число e, яке має бути взаємно простим з числом $(p-1)(q-1)$;
- обчислює закритий ключ d, який є цілим числом, оберненим до e за модулем $(p-1)(q-1)$: $d = e^{-1} \textbf{ mod}((p–1)(q–1))$, тобто $de \equiv 1 \ (\text{mod}(p–1)(q–1))$; таке обернене існує, бо gcd $(e, (p – 1)(q – 1)) = 1$.

•**2.** Повідомлення перетворюють у цифрову форму, тобто записують у вигляді послідовності цілих чисел. Щоб це зробити, ми спочатку замінюємо кожну букву повідомлення на двоцифрове число, використовуючи ту саму заміну, що й для шифру зсуву, але з однією відмінністю. А саме, ми включимо початковий нуль для букв від А до З (в українській абетці), отже А заміниться на 00, Б – на 01, …, З – на 09. Після цього об'єднують ці двоцифрові числа в цифровий рядок. Нарешті, розбивають цей рядок на рівного розміру **блоки** з $2N$ цифр, де $2N$ – **найбільше** додатне число таке, що 3232…32 (для українського алфавіту), чи 2525…25 (для англійського) із $2N$ цифр **не** перевищує n. У разі необхідності доповнюють повідомлення фіктивним символом так, щоб останній блок мав такий самий розмір, що й інші. Отже, повідомлення подано як послідовність блоків $M_1, M_2, \ldots, M_k$ для якогось цілого k. Шифрування полягає в трансформації кожного блоку M_i у зашифрований блок C_i. Це робиться з використанням функції

$$C = M^e \textbf{ mod } n.$$

Для виконання шифрування ми використовуємо алгоритм швидкого піднесення до степеня в модулярній арифметиці (алгоритм 4.1).

Зашифроване повідомлення у вигляді послідовності блоків цілих чисел відправляють бажаному одержувачу. Тому що криптосистема RSA перетворює блоки букв у блоки букв, вона є **блоковим шифром**.

•3. Одержувач розшифровує повідомлення за допомогою закритого ключа d. Це здійснюють для кожного зашифрованого блоку C_i за допомогою функції

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

Зазначимо, що для дешифрування використовують той самий алгоритм швидкого піднесення до степеня в модулярній арифметиці (алгоритм 4.1), що й для шифрування.

Перед тим, як обґрунтувати коректність розглянутої системи шифрування, наведемо елементарний приклад. Для обчислень у модулярній арифметиці можна скористатися онлайн калькулятором.

Приклад.

Генерування ключів.

1. Вибираємо $p = 53$, $q = 67$.

2. $n = pq = 53 \cdot 67 = 3551$.

3. $(p - 1)(q - 1) = 52 \cdot 66 = 3432$, $e = 17$.

4. $d = e^{-1} \bmod (52 \cdot 66) = 17^{-1} \bmod 3432 = 1817$; для обчислення можна скористатись розширеним алгоритмом Евкліда та теоремою Безу або онлайн калькулятором.

Шифрування повідомлення. Нехай потрібно передати вказівку КУПИ. Спочатку перетворюємо повідомлення в цифрову форму, замінюючи кожну букву її двоцифровим номером в алфавіті (нагадаємо, що нумерація починається з 0: буква А заміниться на 00); отримаємо 14231910. З нашим модулем $n = 3551$ цифрове повідомлення розбивається на блоки по чотири цифри, бо $3232 < 3551 < 323232$:

1423 1910 (тобто $M_1 = 1423$, $M_2 = 1910$, $k = 2$).

Ми шифруємо кожен блок M_i, використовуючи функцію $C = M^e \bmod n$. Обчислення за допомогою алгоритму швидкого піднесення до степеня в модулярній арифметиці або за допомогою Modular Arithmetic Calculator дають $1423^{17} \bmod 3551 = 3153$, $1910^{17} \bmod 3551 = 2335$. Зашифроване повідомлення 3153 2335 (тобто $C_1 = 3153$, $C_2 = 2335$).

Розшифрування повідомлення. Кожний блок шифру C_i розшифровуємо, використовуючи функцію $M = C^d \bmod n$. Обчислення за допомогою алгоритму швидкого піднесення до степеня в модулярній арифметиці або за допомогою онлайн калькулятора дають $3153^{1817} \bmod 3551 = 1423$, $2335^{1817} \bmod 3551 = 1910$. Ми одержали вихідне повідомлення у цифровій формі:1423 1910. Повертаючись до букв українського алфавіту, одержуємо вихідне повідомлення КУПИ.

Обґрунтування коректності системи RSA

Нагадаємо, що $de \equiv 1 \pmod{(p-1)(q-1)}$, тому існує ціле j таке, що $de = 1 + j(p-1)(q-1)$. Звідси випливає, що

$$C^d \equiv \left(M^e\right)^d \equiv M^{de} \equiv M^{1+j(p-1)(q-1)} \pmod{n}.$$

Зазначимо, що $M^{1+j(p-1)(q-1)} = M \cdot \left(M^{p-1}\right)^{j(q-1)} = M \cdot \left(M^{q-1}\right)^{j(p-1)}$.

Подальше обґрунтування зробимо в додатковому припущенні, що

$$\gcd(M, p) = \gcd(M, q) = 1,$$

яке виконується за виключенням рідких випадків.

Тоді за малою теоремою Ферма можемо записати

$$C^d \equiv M \cdot \left(M^{p-1}\right)^{j(q-1)} \equiv M \cdot 1 = M \pmod{p},$$

$$C^d \equiv M \cdot \left(M^{q-1}\right)^{j(p-1)} \equiv M \cdot 1 = M \pmod{q}.$$

Оскільки $\gcd(p,q) = 1$, то з китайської теореми про остачі випливає, що $C^d \equiv M \pmod{pq}$.

Зауваження. Ми довели коректність системи RSA при додатковому припущенні $\gcd(M, pq) = 1$. Можна довести, що конгруенція $C^d \equiv M \pmod{pq}$ справджується також і при $\gcd(M, pq) > 1$. Для цього потрібно виписати конгруенції за модулем p і за модулем q і застосувати китайську теорему про остачі. Зазначимо, що коли $\gcd(M, p) > 1$),то $\gcd(M, q) = 1$), бо $M < pq$, а p і q – прості. Аналогічно, коли $\gcd(M, q) > 1$), то $\gcd(M, p) = 1$).

Чому криптосистема RSA підходить для криптографії з відкритим ключем?

По-перше, можна швидко побудувати відкритий ключ, знайшовши два великих простих числа p і q, кожне з яких має більш ніж 200 цифр, і знайти ціле число e, взаємно просте з $(p-1)(q-1)$. Коли ми знаємо розклад n на множники, тобто, коли ми знаємо p і q, ми можемо швидко знайти d, яке є цілим, оберненим до e за модулем $(p-1)(q-1)$. [Це робиться за допомогою алгоритму Евкліда: знаходять коефіцієнти Безу s і t для e і $(p-1)(q-1)$, тоді s – обернене до e за модулем $(p-1)(q-1)$.] Знання d дає змогу розшифрувати повідомлення, відправлені за допомогою нашого ключа. Однак, невідомий спосіб розшифрування повідомлень, який не заснований на пошуку факторизації n (тобто розкладу n на прості множники). Факторизація вважається важким завданням, на відміну від знаходження великих простих чисел p і q, **яке може бути зроблено швидко за допомогою імовірнісних методів**. Найбільш відомі (станом на 2018 р.) ефективні методи факторизації вимагають мільярди років для факторизації 400-значних чисел. Отже, якщо p і q 200-розрядні прості числа, то вважається, що повідомлення, зашифровані за допомогою $n = pq$, не можуть бути розшифровані в розумний період часу, за виключенням ситуації, коли прості числа p і q відомі.

Звичайно, коли числа p і q невеликі, як-от у нашому навчальному прикладі, задача факторизації n не є складною і такий шифр неважко зламати.

Криптографічні протоколи

Досі ми вивчали, як криптографію можна використати, щоб засекретити повідомлення. Проте, є й інші важливі застосування криптографії. Одне з них – криптографічні протоколи, які дають змогу досягти певного рівня безпеки при обміні повідомленнями між сторонами або учасниками протоколу. Під протоколом ми будемо розуміти послідовність узгоджених приписів, згідно з якими відбувається обмін повідомленнями. Зокрема, ми покажемо, як можна використати криптографію, щоб дати змогу двом сторонам обмінюватись секретним ключем через незахищений канал зв'язку. Ми покажемо також, як криптографію можна використати для відправлення підписаних секретних повідомлень таким чином, щоб одержувач міг бути впевненим, що повідомлення прийшло від передбачуваного відправника.

Обмін ключем

Класична симетрична система захисту конфіденційності листування ґрунтується на наявності надійного каналу для обміну секретним ключем. Канал цей може бути набагато повільнішим, ніж канал для обміну повідомленнями, але безумовно він має бути захищеним від посягань суперника. У класиці такий канал реалізовували за допомогою кур'єра.

В асиметричних криптосистемах проблеми пересилання ключа не існує, бо закритий ключ є особистою власністю кожної сторони, а відкритий ключ перебуває у відкритому доступі. Зазначимо однак, що з появою асиметричних криптосистем симетричні системи не вийшли зі вжитку, бо вони є набагато швидкішими. Фактор швидкості шифрування або дешифрування стає визначальним при пересиланні великих обсягів інформації. Проте асиметричні криптосистеми відкривають нові можливості для обміну ключами при використанні криптосистем симетричних. Наприклад, практичним є пересилання ключа тим же каналом зв'язку, що й звичайних повідомлень, але зашифрованого за допомогою асиметричної криптосистеми. І хоча швидкодія криптосистеми з

відкритим ключем нижча, для цієї мети вона достатня, адже ключ буде посилатися значно рідше, ніж звичайні повідомлення.

Нижче наводиться **інше** елегантне розв'язання проблеми, а саме, *протокол експоненційного обміну ключем*. Наголосимо, що тут йдеться про ключ **симетричної** криптосистеми. Двоє учасників протоколу – їх за усталеною традицією звуть Аліса і Боб – спілкуються через канал, що ймовірно прослуховується, і тому хочуть домовитися про спільний секретний ключ. Протокол, за яким вони діятимуть, містить такі кроки, де обчислення виконуються в Z_p.

(1) Аліса вибирає велике просте число p і первісний корінь r за модулем p, і **відкрито**, не роблячи з цього жодної таємниці, посилає p і r Бобові.

(2) Аліса вибирає **секретне** число k_1 у межах від 1 до $p-1$ включно, а Боб – **секретне** число k_2 у тих же межах.

(3) Аліса обчислює $r^{k_1} \bmod p$ і **відкрито** посилає це значення Бобові, а Боб обчислює $r^{k_2} \bmod p$ і теж **відкрито** посилає Алісі.

(4) Аліса обчислює число $\left(r^{k_2}\right)^{k_1} \bmod p$.

(5) Боб обчислює число $\left(r^{k_1}\right)^{k_2} \bmod p$.

Як результат – Аліса і Боб обчислюють одне і теж число

$$\left(r^{k_2}\right)^{k_1} \bmod p = \left(r^{k_1}\right)^{k_2} \bmod p = r^{k_1 k_2} \bmod p,$$

яке і приймають у якості **секретного** ключа.

Бачимо, що p, r, $r^{k_1} \bmod p$, $r^{k_2} \bmod p$ – передбачається як відкрита інформація, а k_1, k_2 та спільний ключ $r^{k_1 k_2} \bmod p$ – як інформація секретна. Для видобування секретної інформації з відкритої суперникові потрібно розв'язати конкретну задачу обчислення дискретного логарифму.

Справді, суперникові потрібно знайти k_1 і k_2 із r^{k_1} **mod** p і r^{k_2} **mod** p, відповідно. Жодний інший спосіб видобути цю секретну інформацію із відкритої невідомий. У свій час ми наголошували, що задача обчислення дискретного логарифму є практично нерозв'язною, коли числа p і r є достатньо великими. За досяжної нині потужності комп'ютерів ця система вважається незламною, коли p має більше 300 десяткових цифр, а k_1 і k_2 – більше 100 десяткових цифр кожне.

Приклад. Нехай Аліса вибрала $p = 97$, а $r = 5$ і переслала ці числа Бобові. Припустимо, що Аліса вибрала $k_1 = 12$, а Боб вибрав $k_2 = 63$ (ці числа не пересилаються). Тоді Аліса надсилає Бобові 5^{12}**mod** $97 = 42$, а Боб надсилає Алісі 5^{63}**mod** $97 = 75$. Обоє обчислюють одне й те саме число 75^{12}**mod** $97 = 42^{63}$**mod** $97 = 21$.

Цифрове підписання

Нині певні фінансові операції мають здійснюватись за короткий період часу, що унеможливлює використання традиційних засобів засвідчення платіжних документів на зразок великої гербової печатки та підпису головного бухгалтера. Але як тоді банкові вберегтися від злодія-інтелектуала, який добре знається і на фінансах, і на електроніці, і може від імені співробітника банку надіслати вимогу перевести гроші на власний підставний рахунок? Тут ми покажемо, як криптографію можна використати для того, щоб особа, яка отримала інформацію, була впевненою, що ця інформація отримана саме від відомої їй людини. Це питання вирішується за допомогою *протоколу цифрового підпису*. Ми розглянемо конкретну реалізацію такого протоколу на базі системи RSA.

Нехай (n, e) – відкритий ключ Аліси, а d – закритий. Аліса може шифрувати повідомлення x, використовуючи *шифрувальну* функцію $E_{(n,e)}(x) = x^e$ **mod** n і може розшифровувати шифроване повідомлення y, використовуючи *дешифрувальну* функцію $D_{(n,e)}(y) = y^d$ **mod** n.

Зазначимо, що Аліса бажає надіслати повідомлення так, щоб кожний, хто його отримає, був упевнений, що це повідомлення саме від неї. Так само, як і під час RSA-шифрування, вона переводить букви повідомлення (незашифрованого) у цифрові еквіваленти і розділяє отриманий цифровий рядок на блоки m_1, m_2, …, m_k рівного розміру (розмір блоків визначають точно так, як і при RSA-шифруванні). Після цього вона застосовує **свою дешифрувальну** функцію $D_{(n,e)}$ до кожного блоку і дістає $D_{(n,e)}(m_i)$, де $i = 1, 2, \ldots, k$. Аліса посилає цей результат усім запланованим адресатам.

Коли будь-який адресат отримує її лист, він застосовує Алісину **шифрувальну** функцію $E_{(n,e)}$ до кожного отриманого блока цифр, – це доступно для будь-кого, бо Алісин відкритий ключ (n, e) – доступна інформація. Результат – блок повідомлення, яке пересилалось, бо $E_{(n,e)}(D_{(n,e)}(m_i)) = m_i$. Отже, Аліса має змогу надсилати свої листи багатьом адресатам, і, якщо вона діятиме за описаним протоколом, кожний адресат може бути впевненим, що лист прийшов саме від Аліси. Наступний приклад ілюструє цей протокол.

Приклад. Припустімо, що Алісин відкритий ключ системи RSA той самий, що й у попередньому прикладі, тобто $n = 53 \cdot 67 = 3551$ і $e = 17$. Її закритий ключ, як знайдено у попередньому прикладі, $d = 1817$. Нехай Алиса хоче передати повідомлення «ЗУСТРІЧ ВІДМІНЕНО». Що саме вона має послати?

Спершу Аліса перекладе повідомлення у блоки цифр й отримає таку послідовність блоків:

0923 2122 2011 2702 1105 1611 1706 1718.

Далі вона застосує **дешифрувальну** функцію $D_{(3551,17)}(y) = y^{1817} \bmod 3551$ до кожного блоку. Використовуючи швидке модулярне піднесення до степеня (з використанням комп'ютера) вона знайде, що $0923^{1817} \bmod 3551 = 0445$, $2122^{1817} \bmod 3551 = 1928$, $2011^{1817} \bmod 3551 = 3284$,

$2702^{1817}\ \mathbf{mod}\ 3551 = 0953$, $1105^{1817}\ \mathbf{mod}\ 3551 = 3501$, $1611^{1817}\ \mathbf{mod}\ 3551 = 1465$, $1706^{1817}\ \mathbf{mod}\ 3551 = 2188$, $1718^{1817}\ \mathbf{mod}\ 3551 = 3042$.

Отже, лист, розділений на блоки, який надішле Аліса, виглядає так:

0445 1928 5284 0953 3501 1465 2188 3042.

Коли її товариші отримають цей лист, вони застосують її (тобто Алісину) **шифрувальну** функцію (яка відкрита) $E_{(3551,17)}(x) = x^{17}\ \mathbf{mod}\ 3551$ до кожного з цих блоків. Коли вони зроблять це, то отримають блоки цифр оригінального листа, який легко можна перекласти українською мовою. Зокрема, $E_{(3551,17)}(0445) = 0445^{17}\ \mathbf{mod}\ 3551 = 0923$, що при перекладі дасть «ЗУ…» і т. д.

Зазначимо, що в протоколі цифрового підпису головну роль відіграють співвідношення

$$D_{(n,e)}\left(E_{(n,e)}(x)\right) = E_{(n,e)}\left(D_{(n,e)}(x)\right) = x.$$

Ці співвідношення зводяться до рівностей

$$\left(x^e\right)^d \mathbf{mod}\ n = \left(x^d\right)^e \mathbf{mod}\ n = x,$$

і виражають той факт, що шифрувальна функція $E_{(n,e)}$ і дешифрувальна функція $D_{(n,e)}$ є взаємно оберненими.

Шифрування за допомогою закритого ключа називають *підписанням повідомлення*, а зашифроване за допомогою закритого ключа повідомлення – *цифровим підписом*. Процес підписання відбувається так само, як і шифрування, тільки використовується закритий (а не відкритий) ключ.

Маючи у розпорядженні дві операції – шифрування і підписання – ми можемо їх сумістити, одночасно шифруючи і підписуючи повідомлення. Це робиться у такий спосіб.

Спочатку, використовуючи свій закритий ключ, ви підписуєте повідомлення. Затим шифруєте повідомлення та підпис за допомогою відкритого ключа отримувача повідомлення. Отримувач спочатку розшифровує за допомогою свого закритого ключа ваше повідомлення і отримує відкритий текст і підпис. Потім отримувач розшифровує за допомогою вашого відкритого ключа підпис і перевіряє, чи збігається він з розшифрованим відкритим текстом повідомлення. Аліса і Боб поступають так.

1. Аліса генерує відкритий ключ (n_A, e_A) і закритий d_A. Вона надсилає (n_A, e_A) Бобові будь-яким зручним способом.

2. Боб генерує відкритий ключ (n_B, e_B) і закритий d_B. Він надсилає (n_B, e_B) Алісі будь-яким зручним способом.

3. Аліса підписує своє повідомлення M своїм закритим ключем: $C_1 = D_{(n_A, e_A)}(M)$.

4. Аліса за допомогою відкритого ключа Боба зашифровує своє повідомлення і підпис, отриманий на кроці 3, тобто зашифровує відкритим ключем Боба (M, C_1): $C_2 = E_{(n_B, e_B)}(M, C_1)$.

5. Аліса надсилає C_2 Бобові.

6. Боб розшифровує C_2 за допомогою свого закритого ключа: $(M, C_1) = D_{(n_B, e_B)}(C_2)$.

7. Боб перевіряє C_1, використовуючи відкритий ключ Аліси: $M = E_{(n_A, e_A)}(C_1)$?

Зауваження. Закритий ключ застосовується у дешифрувальний функції, оскільки $D_{(n, e)}(y) = y^d \bmod n$, а відкритий ключ – у шифрувальній функції $E_{(n, e)}(x) = x^e \bmod n$. Але шифрувати можна, використовуючи дешифрувальну функцію (закритий ключ), тоді дешифрування здійснюється за допомогою шифрувальної функції (відкритий ключ). Саме цю можливість використовують у цифровому підписанні.

Довідка при сучасні симетричні криптосистеми

Раніше ми вже говорили, що з появою асиметричних криптосистем симетричні системи не вийшли зі вжитку, бо вони є набагато швидкішими. Фактор швидкості шифрування або дешифрування стає визначальним при пересиланні великих обсягів інформації. Є чимало симетричних криптосистем. У 70-х роках XX століття компанія IBM розробила симетричну криптосистему **DES** (**D**ata **E**ncryption **S**tandard), яка була представлена Національному бюро стандартів США і використовувалась Агентством національної безпеки. Врешті-решт, **DES** перетворилась на стандартний алгоритм безпеки, який використовувався урядом Сполучених Штатів протягом 20 років**,** допоки distributed.net не об'єднався з Electronic Frontier Foundation і публічно дешифрували **DES** менш ніж за 24 години.

Державний інститут стандартів і технологій США розпочав розробку **AES** (**A**dvanced **E**ncryption **S**tandard), коли стало очевидним, що для **DES** потрібен новий наступник. Цей новий алгоритм був розроблений для якомога більш простого застосування в апаратному, програмному та обмеженому оточенні. **AES** є блоковим симетричним шифром і здатний захистити важливу державну інформацію від різних методів атаки. **AES** є щонайменше в шість разів швидшим, ніж навіть потрійний **DES**.

Порівняння DES і AES

	DES	**AES**
Рік розробки	1977	1999
Довжина ключа	56 біт	128, 192 або 256 біт
Тип шифру	Блоковий симетричний шифр	Блоковий симетричний шифр
Розмір блоку	64 біт	128 біт
Безпека	Ненадійний	Вважається безпечним