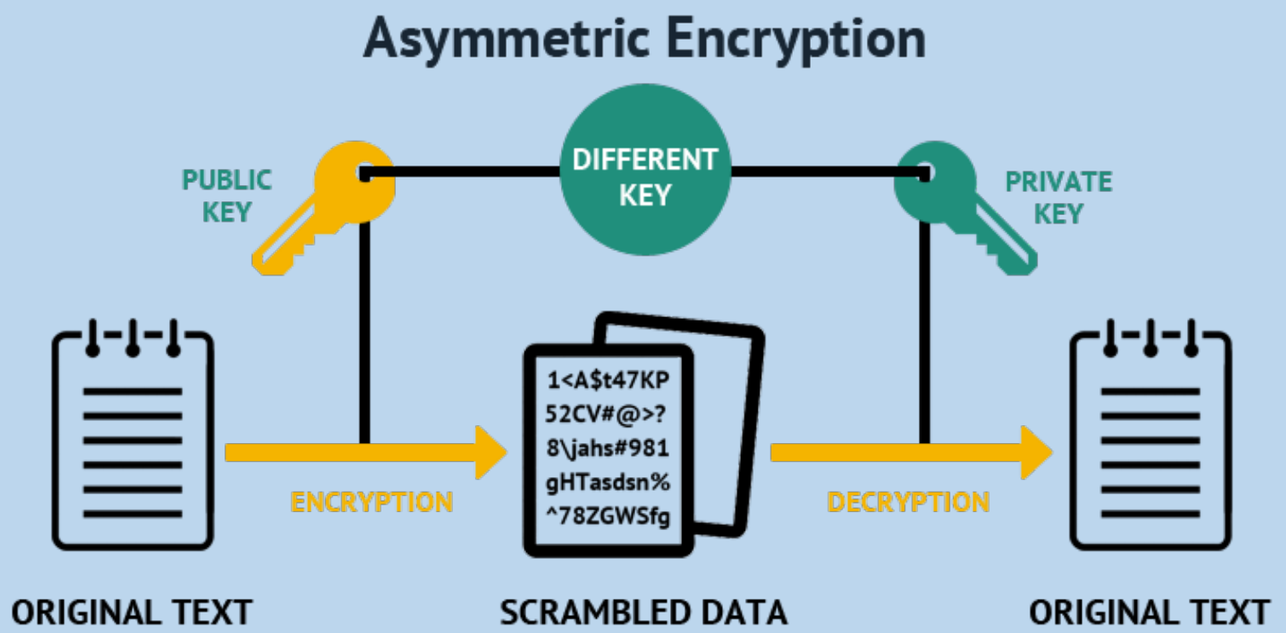
ЛЕКЦІЯ 6

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Асиметричні криптосистеми



План

1. Ідея криптосистеми з відкритим ключем
2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)
3. Алгоритм RSA
4. Алгоритм Ель-Гамаля
5. Алгоритм обміну ключами Діффі-Хелмана
6. Симетричні шифри vs асиметричні шифри

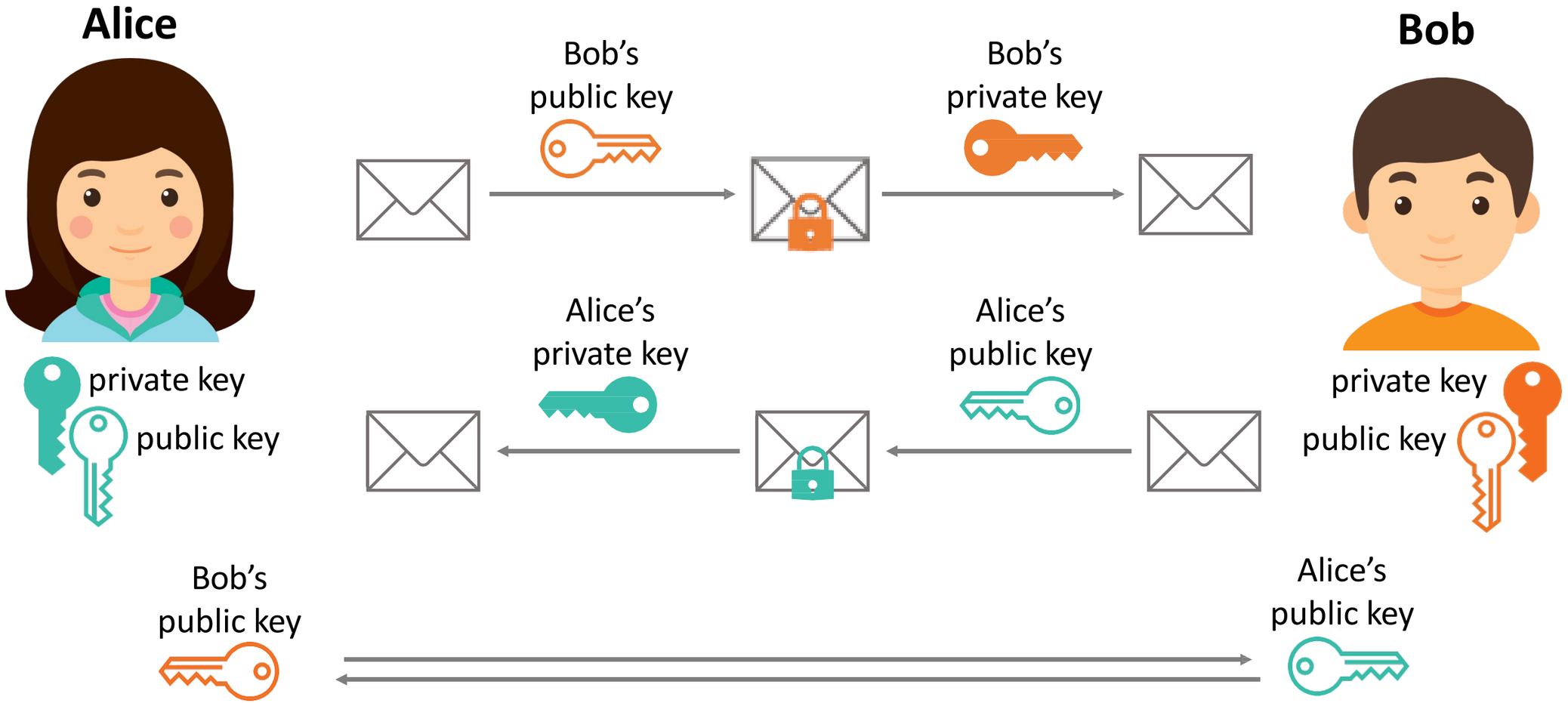
1. Ідея криптосистеми з відкритим ключем

Ідея криптосистеми з відкритим ключем була висунута американськими криптографами Уітфілдом Діффі та Мартіном Хелманом (1976 рік), і окремо Ральфом Мерклом (1978 рік)

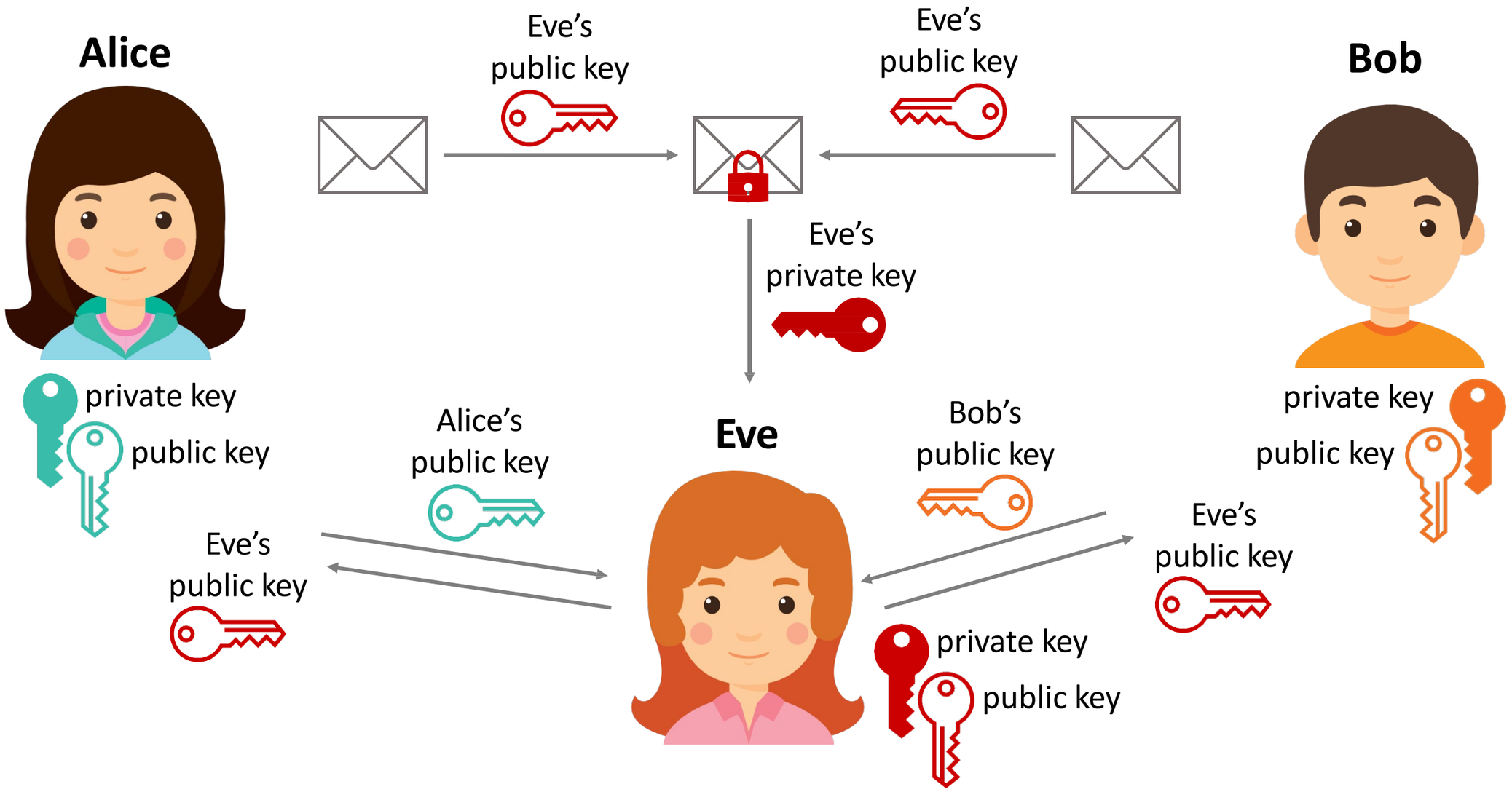
У асиметричних криптосистемах для шифрування використовується відкритий ключ (публічний), а для дешифрування – закритий (приватний)

Ральф Меркл | Мартін Хелман | Уітфілд Діффі

1. Ідея криптосистеми з відкритим ключем



1. Ідея криптосистеми з відкритим ключем



1. Ідея криптосистеми з відкритим ключем

How asymmetric (public key) encryption works

1. Ідея криптосистеми з відкритим ключем

Математична база

Ідея криптографії з відкритим ключем тісно пов'язана з ідеєю однобічних функцій (one-way function), тобто таких функцій *f(x)*, що по відомому *x* досить просто знайти значення *f(x)*, тоді як визначити *x* з *f(x)* важко

1. Ідея криптосистеми з відкритим ключем

Математична база

Також використовуються однобічні функції з лазівкою (one-way trap-door function).

Лазівка – це певний секрет, що допомагає розшифрувати. Тобто існує такий *y*, що знаючи *f(x),* можна обчислити *x*

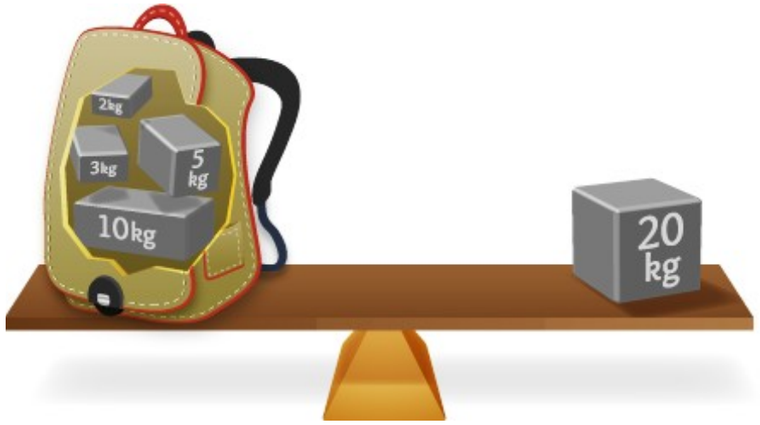
2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Задача рюкзака

Дано набір предметів різної маси. Чи можна покласти деякі із цих предметів у рюкзак так, щоб маса рюкзака дорівнювала певному значенню?

Задача: за вагою рюкзака визначити, які предмети поклали, а які ні

Наприклад, маси предметів 1, 5, 6, 11, 14 і 20. Можна спакувати рюкзак так, що його маса дорівнюватиме 22, використавши маси 5, 6 і 11. Неможливо спакувати рюкзак так, щоб його маса дорівнювала 24



2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Ідея шифрування повідомлення як розв'язання задачі рюкзака

Дано набір значень $M_1, M_2,\ldots, M_n$ і сума S, обчислити значення b_i, такі що $S = M_1 b_1 + M_2 b_2 + \ldots + M_n b_n$, $b_i \in \{0,1\}$

$M_1, M_2,\ldots, M_n$ – рюкзак;
$b_1, b_2,\ldots, b_n$ – відкритий текст;
S – шифротекст

Приклад 2.1:

Відкритий текст	111001	010110	000000	011000
Рюкзак	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20
Шифротекст	1+5+6+20=32	5+11+14=30	0=0	5+6=11

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Задача рюкзака

Легка

Якщо перелік мас предметів являє собою суперзростаючу послідовність, то задачу рюкзака легко розв’язати

Складна

Якщо перелік мас предметів являє собою нормальну послідовність, то задачу рюкзака розв’язати важко

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Суперзростаюча послідовність – це послідовність, у якій кожний елемент більший за суму усіх попередніх елементів

Нормальна послідовність – це послідовність, що містить довільні елементи

Наприклад, послідовність {1, 3, 6, 13, 27, 52} є суперзростаючою

Наприклад, послідовність {1, 3, 4, 9, 15, 25} не є суперзростаючою, тобто вона нормальна

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Алгоритм розв'язання задачі суперзростаючого рюкзака

1. Повну вагу рюкзака порівнюємо з найбільшим числом послідовності

2. Якщо повна вага менша за це число, то його не кладемо у рюкзак

3. Якщо повна вага більша або дорівнює цьому числу, то воно кладеться у рюкзак. Зменшуємо масу рюкзака на це значення.

4. Переходимо до наступного по величині числа послідовності

5. Будемо повторювати, поки процес не закінчиться. Якщо повна вага зменшиться до нуля, то розв’язок знайдений

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.2:

Повна вага рюкзака – 70, послідовність мас {2, 3, 6, 13, 27, 52}

1. Найбільша маса – 52 < 70 ⇨ кладемо 52 у рюкзак.
2. Віднімаємо: 70-52=18.
3. Наступна маса – 27 > 18 ⇨ 27 у рюкзак не кладемо.
4. Вага 13 < 18 ⇨ кладемо 13 у рюкзак.
5. Віднімаємо: 18-13=5.
6. Наступна маса – 6 > 5 ⇨ 6 не кладемо у рюкзак.

Продовження цього процесу покаже, що й 2, і 3 кладемо у рюкзак, і повна вага зменшується до 0, що повідомляє про знайдений розв’язок.

Відкритий текст: 110101

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Криптосистема Меркла-Хелмана

Закритий ключ – суперзростаюча послідовність

Відкритий ключ – нормальна послідовність

Генерування відкритого ключа із закритого

1. Генерується суперзростаюча послідовність

2. Обирається число *m* (модуль), більше за суму усіх чисел послідовності

3. Знаходиться *n* взаємно просте з *m*

4. Усі значення суперзростаючої послідовності множаться по модулю m на число *n*

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.3:

Дано: закритий ключ – суперзростаюча послідовність {2, 3, 6, 13, 27, 52}, m = 105, n = 31

Нормальною послідовністю буде:
2·31 mod 105 = 62
3·31 mod 105 = 93
6·31 mod 105 = 81
13·31 mod 105 = 88
27·31 mod 105 = 102
52·31 mod 105 = 37

Відкритий ключ – {62, 93, 81, 88, 102, 37}

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Шифрування у криптосистемі Меркла-Хелмана

1. Розбити повідомлення на блоки, рівні по довжині кількості елементів послідовності рюкзака.

2. Вважати, що у відкритому тексті одиниця вказує на присутність члена послідовності, а нуль – на його відсутність.

3. Обчислити повні маси рюкзака – по одному для кожного блоку повідомлення.

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.4:

Дано: повідомлення в бінарному виді 011000110101101110,
відкритий ключ – послідовність {62, 93, 81, 88, 102, 37}

Шифруємо: повідомлення = 011000 110101 101110
011000 відповідає 93 + 81 = 174
110101 відповідає 62 + 93 + 88 + 37 = 280
101110 відповідає 62 + 81 + 88 + 102 = 333

Шифротекст: послідовність {174, 280, 333}

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Дешифрування у криптосистемі Меркла-Хелмана

1. Спочатку визначають n^{-1}, таке що n (n^{-1}) ≡ 1 (mod m)

2. Кожне значення шифротексту множиться на n^{-1}mod m

3. Одержати значення відкритого тексту за допомогою закритого ключа – одиниця вказує на присутність члена послідовності, а нуль – на його відсутність

2. Алгоритм рюкзака (криптосистема Меркла-Хелмана)

Приклад 2.5:

Дано: шифротекст {174, 280, 333}, закритий ключ – {2, 3, 6, 13, 27, 52},
m =105, n = 31

Дешифруємо:

У нашому випадку n^{-1} дорівнює 61, тому значення шифротекста помножимо на 61 mod 105.

174 · 61 mod 105 = 9 = 3 + 6, що відповідає 011000

280 · 61 mod 105 = 70 = 2 + 3 + 13 + 52, що відповідає 110101

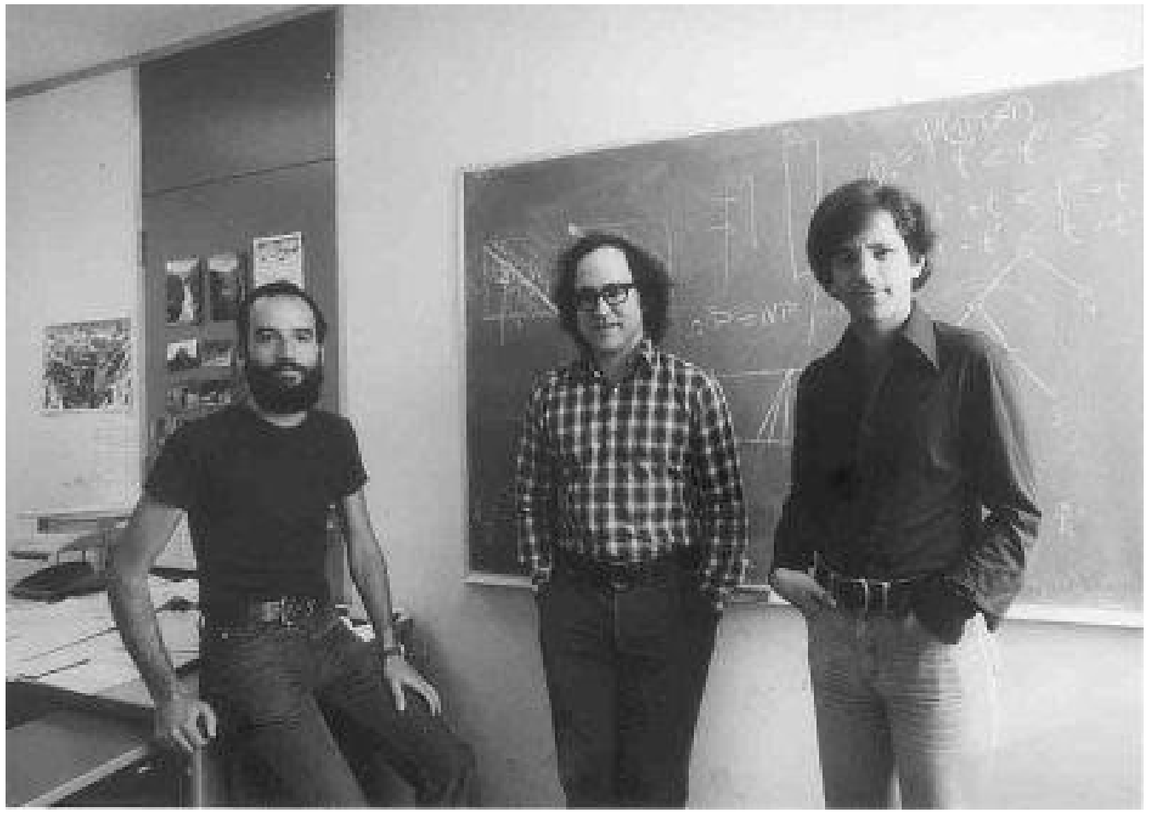
333 · 61 mod 105 = 48 = 2 + 6 + 13 + 27, що відповідає 101110

Відкритий текст: 011000 110101 101110

3. Алгоритм RSA

Автори криптоалгоритму RSA (Rivest-Shamir-Adleman) – Рон Рівест, Аді Шамір і Леонард Едлман (1977 рік)

Безпека RSA заснована на складності розкладання на множники великих чисел



Аді Шамір | Рон Рівест | Леонард Едлман

3. Алгоритм RSA

Математична база

Ця система базується на таких двох фактах із теорії чисел:

- ✓задача перевірки числа на простоту є порівняно легкою;
- ✓задача розкладання на множники чисел вигляду $n = p \cdot q$ є складною, якщо ми знаємо тільки n, а p і q – великі прості числа (задача факторизації).

3. Алгоритм RSA

Генерація ключів

1. Вибираються два великих випадкових простих числа p і q
2. Обчислюється модуль системи – добуток: $n = p \cdot q$
3. Обчислюється функція Ейлера: $\varphi(n) = \varphi(pq) = (p-1)(q-1)$
4. Випадковим чином вибирається число e (ключ шифрування), таке що $1 < e < \varphi(n)$ та взаємно просте з $\varphi(n)$
5. За допомогою розширеного алгоритму Евкліда знаходиться число d (ключ дешифрування), таке що $ed \equiv 1 (\mathrm{mod}\ \varphi(n))$
6. (e, n) публікується у якості відкритого ключа
7. (d, n) виконує роль закритого ключа і тримається таємниці

3. Алгоритм RSA

Шифрування:

повідомлення m розбивається на цифрові блоки, менші n;

кожен блок повідомлення m_i зашифровують за формулою:

$$c_i = m^e \bmod n$$

Дешифрування:

для кожного зашифрованого блоку c_i обчислюють:

$$m_i = c^d \bmod n$$

3. Алгоритм RSA

Приклад 2.1 (генерація ключів):

Дано: повідомлення КНИГА, що складається із символів українського алфавіту та представляється як послідовність цілих чисел M = 14 17 10 3 0

1. Оберемо p = 3 і q = 11, тоді $n = p \cdot q = 3 \cdot 11 = 33$.
2. Обчислимо $\varphi(33) = 2 \cdot 10 = 20$.
3. Виберемо (випадково) $e = 3$ та перевіримо виконання умов:
   $1 < 3 < \varphi(n)$, НСД(3, 20) = 1.
4. Визначимо d – ключ дешифрування з рівняння $3d \equiv 1 (\mathrm{mod}\ 20)$.
   Для розв’язання рівняння використаємо розширений алгоритм Евкліда (див. вказівки до Лаб6) та зайдемо $d = 7$.

3. Алгоритм RSA

Приклад 2.2 (шифрування):

Отже відкритий ключ $e = 3$, закритий ключ $d = 7$.

Зашифруємо повідомлення M= 14 17 10 3 0, що складається із п’яти блоків m_i та отримаємо шифротекст C = 5 29 10 27 0

$c_1 = 14^3 \, mod \, 3 \, 3 = ((14^2 \, mod \, 3 \, 3) \cdot (14^1 \, mod \, 3 \, 3)) \, mod \, 3 \, 3 = (31 \cdot 14) \, mod \, 3 \, 3 = 434 \, mod \, 3 \, 3 = 5;$

$c_2 = 17^3 \, mod \, 3 \, 3 = ((17^2 \, mod \, 3 \, 3) \cdot (17^1 \, mod \, 3 \, 3)) \, mod \, 3 \, 3 = (25 \cdot 17) \, mod \, 3 \, 3 = 425 \, mod \, 3 = 29;$

$c_3 = 10^3 \, mod \, 3 \, 3 = 1000 \, mod \, 3 \, 3 = 10;$

$c_4 = 3^3 \, mod \, 3 \, 3 = 27 \, mod \, 3 \, 3 = 27;$

$c_5 = 0^3 \, mod \, 3 \, 3 = 0 \, mod \, 3 \, 3 = 0.$

3. Алгоритм RSA

Приклад 2.3 (дешифрування):

Для дешифрування потрібно також виконати піднесення до степеню, використовуючи ключ дешифрування 7.

Відкритий текст: M = 14 17 10 3 0 $\Rightarrow$ КНИГА

$m_1 = 5^7 \; mod \; 33 = ((5^4 \; mod \; 33) \cdot (5^3 \; mod \; 33)) \; mod \; 33 = (31 \cdot 26) \; mod \; 33 = 806 \; mod \; 33 = 14;$

$m_2 = 29^7 \; mod \; 33 = ((29^4 \; mod \; 33) \cdot (29^3 \; mod \; 33)) \; mod \; 33 =$

$= (((29^2))^2 \; mod \; 33) \cdot (29^2 \; mod \; 33) \cdot (29 \; mod \; 33)) \; mod \; 33 = (25 \cdot 16 \cdot 29) \; mod \; 33 = 11600 \; mod \; 33 = 17;$

$m_3 = 10^7 \; mod \; 33 = ((10^4 \; mod \; 33) \cdot (10^3 \; mod \; 33)) \; mod \; 33 =$

$= (((10^2))^2 \; mod \; 33) \cdot (10^2 \; mod \; 33) \cdot (10 \; mod \; 33)) \; mod \; 33 = (1 \cdot 1 \cdot 10) \; mod \; 33 = 10 \; mod \; 33 = 10;$

$m_4 = 27^7 \; mod \; 33 = ((27^4 \; mod \; 33) \cdot (27^3 \; mod \; 33)) \; mod \; 33 =$

$= (((27^2))^2 \; mod \; 33) \cdot (27^2 \; mod \; 33) \cdot (27 \; mod \; 33)) \; mod \; 33 = (9 \cdot 3 \cdot 27) \; mod \; 33 = 729 \; mod \; 33 = 3;$

$m_5 = 0^7 \; mod \; 33 = 0 \; mod \; 33 = 0.$

3. Алгоритм RSA

Приклад 2.4:

p

12131072439211271897323671531612440428472427633701410925634549312301964
37304208561932419736532241686654101705736136521417171171379797429933487
1062829803541

q

12027524255478748885956220793734512128733387803682075433653899983955179
85098879789986914690080913161115334681705083209602216014636634639181247
0987105415233

n

14590676800758332323018693934907063529240187237535716439958187101987343
87990053589383695714026701498021218180862924674228281570229220767469065
43401224889672472407926969987100581290103199317858753663710862357656510
50788371429711563734278891146353510271203276516651841172685983798867211
1837205085526346618740053

3. Алгоритм RSA

Приклад 2.4 (продовження):

e - the public key

65537 has a gcd of 1 with $\phi(n)$, so lets use it as the public key. To calculate the private key, use extended euclidean algorithm to find the multiplicative inverse with respect to $\phi(n)$.

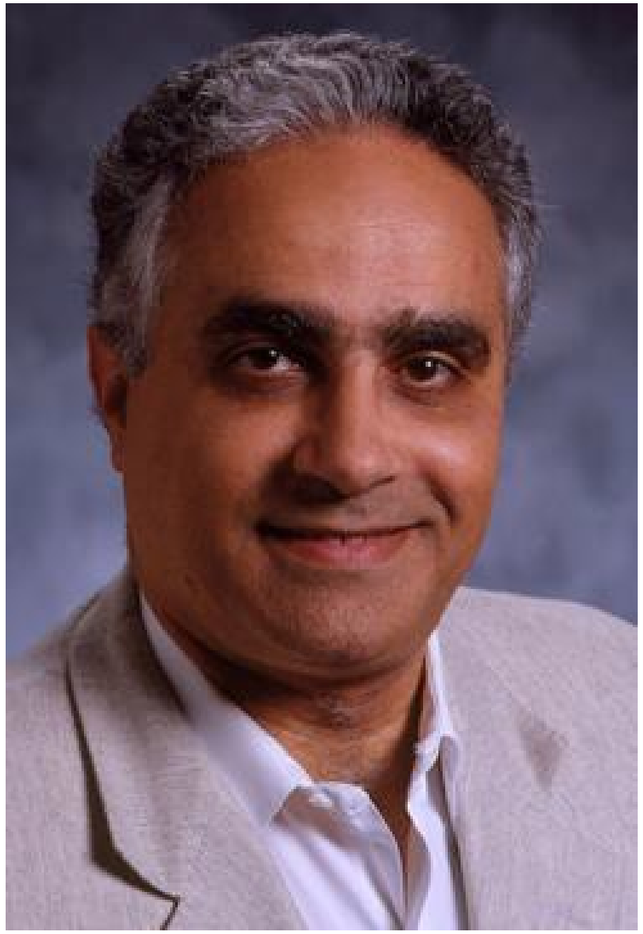
d - the private key

894894250092744443682285459217730939196695860658842574454978544564876748396298183909349419732628796167979706089172836798754993315741611138540888132754881105882471930775825272784379065040156806234235500672400424666656542323835029222154936232894721388664458187891279461234078077257026266440910365023725451397 13

4. Алгоритм Ель-Гамаля

Автор – американський вчений єгипетського походження
Тахер Ель-Гамаль
(1985 рік)

Безпека алгоритму заснована на складності обчислення дискретних логарифмів у скінченному полі



Тахер Ель-Гамаль

4. Алгоритм Ель-Гамаля

Генерація ключів

1. Генерується просте випадкове число p
2. Вибирається генератор g, таке що $1 < g < p - 1$ та $g^{p-1} \bmod p = 1$.
3. Вибирається випадкове число x, таке що $1 < x < p - 1$
4. Обчислюється $y = g^x \bmod p$
5. Відкритими даними є p, g, y
6. Закритим ключем є x

4. Алгоритм Ель-Гамаля

Шифрування:

Повідомлення M шифрується таким чином: вибирається сесійний ключ – випадкове число k, таке що $1 < k < p - 1$;

потім обчислюються

$$a = g^k \bmod p$$

$$b = y^k M \bmod p$$

Пара чисел (a, b) є шифротекстом

Дешифрування:

для дешифрування (a, b) обчислюється

$$M = b(a^x)^{-1} \bmod p$$

або

$$M = b(a^x)^{-1} \bmod p = \\ = b \cdot a^{(p-1-x)} \bmod p$$

4. Алгоритм Ель-Гамаля

Приклад 3.1 (генерація ключів):

1. Нехай $p = 11, g = 2.$
2. Виберемо $x = 8$ – випадкове ціле число x таке, що таке що $1 < x < p - 1$.
3. Обчислимо $y = g^x \bmod p = 2^8 \bmod 11 = 3.$
4. Отже, відкритим даними є трійка є 11, 2 та 3, закритим ключем є число $x = 8.$

4. Алгоритм Ель-Гамаля

Приклад 3.2 (шифрування):

Дано: повідомлення $M = 5$.

Вибираємо випадкове ціле число $k = 9$ таке, що $1 < k < p - 1$.

Обчислюємо число

$$a = g^k \bmod p = 2^9 \bmod 11$$
$$= 512 \bmod 11 = 6$$

Обчислюємо число

$$b = y^k M \bmod p = 3^9 \cdot 5 \bmod 11$$
$$= 19683 \cdot 5 \bmod 11 = 9$$

Пара $(6, 9)$ є шифротекстом.

Приклад 3.3 (дешифрування):

Шифротекст $(6, 9)$, закритий ключ $x = 8$.

Обчислюємо M за формулою:

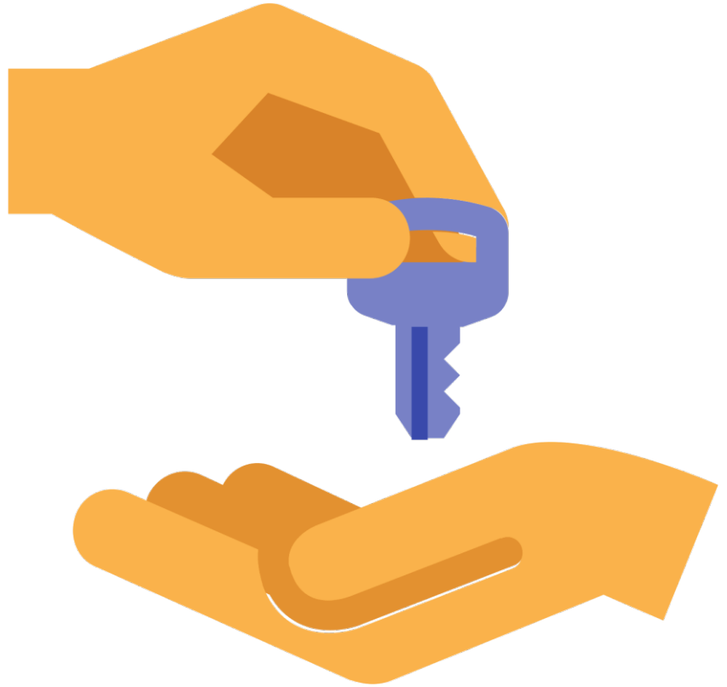
$$M = b(a^x)^{-1} \bmod p =$$
$$= b \cdot a^{(p-1-x)} \bmod p$$
$$= 9 \cdot 6^{(11-1-8)} \bmod 11 = 5$$

Отримали початкове повідомлення

$$M = 5.$$

5. Алгоритм обміну ключами Діффі-Хелмана

Алгоритм обміну ключами Діффі-Хелмана дозволяє двом сторонам отримати спільний секретний ключ, використовуючи незахищений від прослуховування, але захищений від модифікації канал зв'язку



Алгоритм заснований на складності обчислень дискретних логарифмів

5. Алгоритм обміну ключами Діффі-Хелмана

Алгоритм Діффі-Хелмана

1. Абоненти A і B спільно обирають просте число p і ціле число g, що є первісним коренем p.

2. Користувач *A* вибирає випадкове ціле число $x < p$, обчислює $x_A = g^x \ mod \ p$ та відправляє його користувачеві *B*.

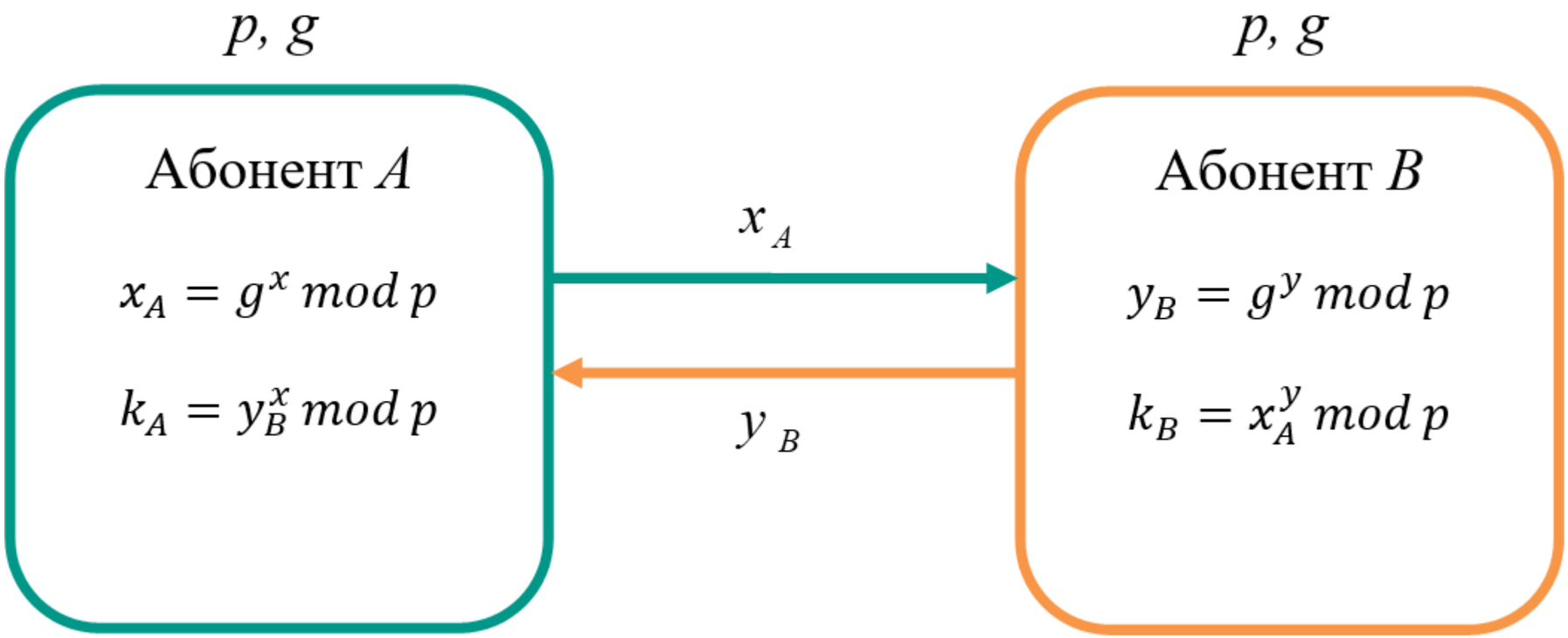
3. Користувач *B* вибирає випадкове ціле число $y < p$, обчислює $y_B = g^y \ mod \ p$ та відправляє його користувачеві *A*.

4. Користувач *A* обчислює закритий ключ за формулою $k_A = y_B^x \ mod \ p$.

5. Користувач *B* обчислює закритий ключ за формулою $k_B = x_A^y \ mod \ p$.

5. Алгоритм обміну ключами Діффі-Хелмана

Схема обміну ключами Діффі-Хелмана



5. Алгоритм обміну ключами Діффі-Хелмана

Приклад 4.1:

1. $p = 11, \mathrm{g} = 2.$
2. $x = 4$, обчислимо $x_A = 2^4 \bmod 11 = 16 \bmod 11 = 5.$
3. $y = 6$, обчислимо $y_B = 2^6 \bmod 11 = 64 \bmod 11 = 9.$
4. $k_A = 9^4 \, mod \, 11 = (9^2)^2 \, mod \, 11 = 4^2 \, mod \, 11 = 16 \, mod \, 11 = 5.$
5. $k_B = 5^6 \, mod \, 11 = (5^3)^2 \, mod \, 11 = 4^2 \, mod \, 11 = 16 \, mod \, 11 = 5.$

Секретний ключ, обчислений обома сторонами – 5.

6. Симетричні шифри vs асиметричні шифри

Характеристика	Симетричні шифри	Асиметричні шифри
Ключ	Один і той самий ключ використовується для шифрування та дешифрування	Один ключ (відкритий) використовується для шифрування, інший (закритий) – для дешифрування
Обмін ключами	Потрібен секретний канал для передачі ключа або інший надійний механізм обміну ключами	Відкритий ключ доступний всім, але його справжність має перевірятися центром сертифікації ключів
Математична складність	Відносно прості математичні операції	Складні математичні обчислення
Швидкість роботи	Висока	Низька
Криптографічна стійкість	Задовільна	Достатня
Вид захисту	Конфіденційність	Конфіденційність, цілісність, автентичність, невідмовність