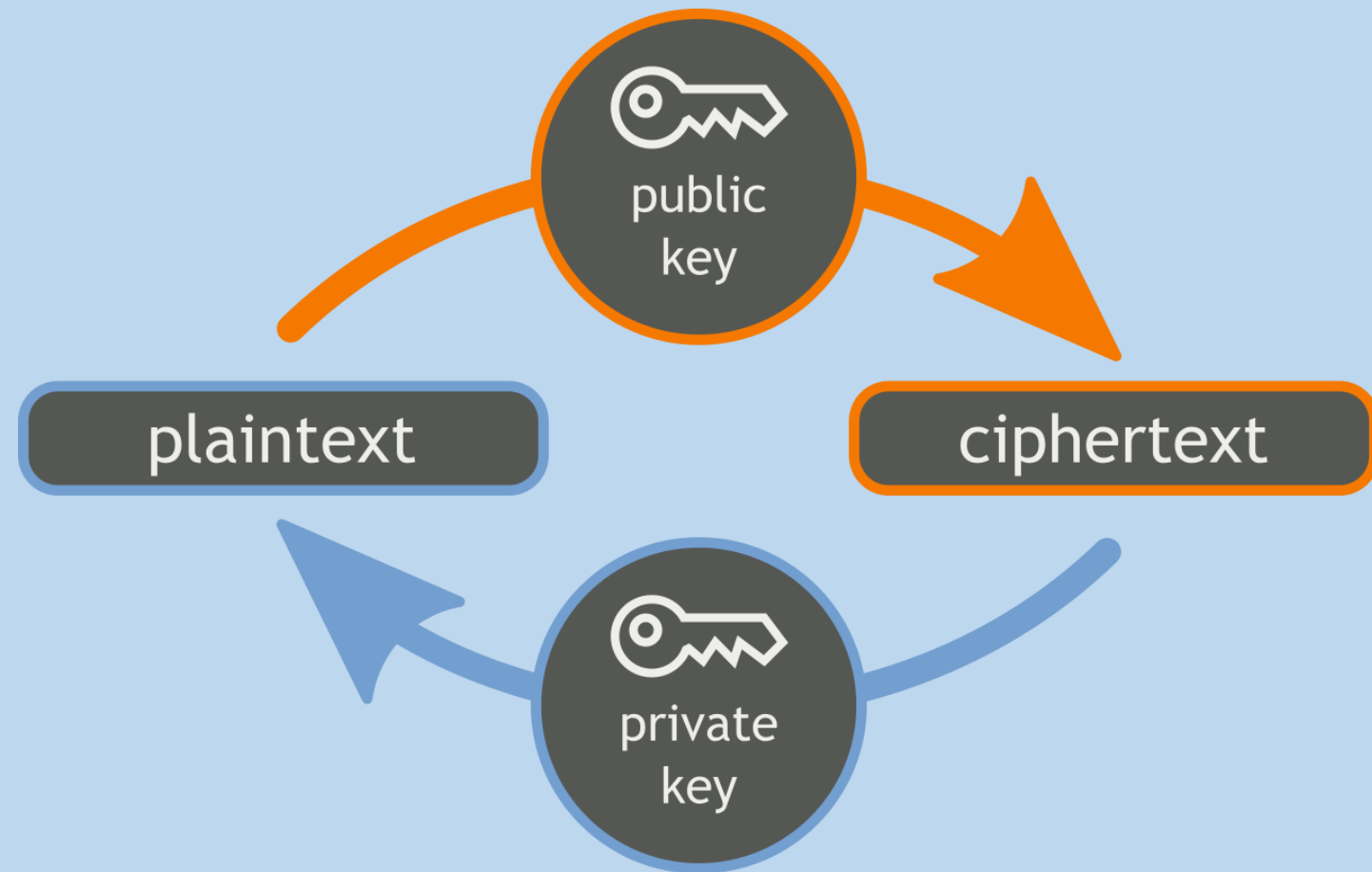


# Асиметричні криптосистеми



# План

1. Алгоритм RSA

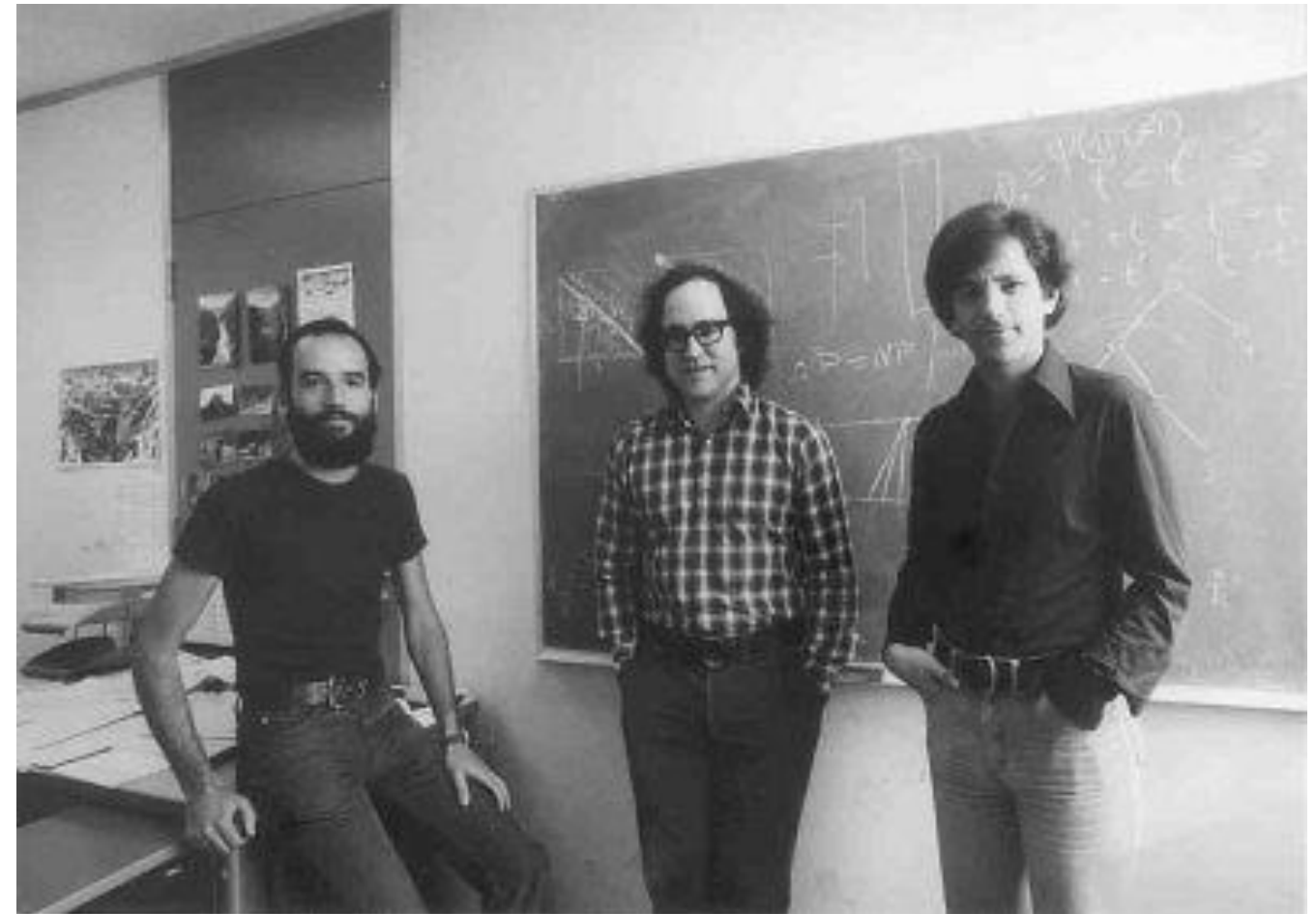
2. Алгоритм Ель-Гамала

3. Алгоритм обміну ключами Діффі-Хелмана

# 1. Алгоритм RSA

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

Безпека RSA заснована на складності розкладання на **множинки** великих чисел



Аді  
Шамір

Рон  
Рівест

Леонард  
Едлман

# 1. 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).



# 1. 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  $ed \equiv 1 \pmod{\varphi(n)}$
6.  $(e, n)$  публікується у якості **відкритого ключа**
7.  $(d, n)$  виконує роль **закритого ключа** і тримається таємниці

# 1. Алгоритм RSA

## Шифрування:

повідомлення  $m$   
розбивається на цифрові  
блоки, менші  $n$ ;  
кожен блок повідомлення  $m_i$   
зашифровують за формулою:

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

## Дешифрування:

для кожного зашифрованого  
блоку  $c_i$  обчислюють:

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

# 1. Алгоритм RSA

## Приклад 1.1 (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:  
 $1 < 3 < \varphi(n)$ ,  $\text{НСД}(3, 20) = 1$ .
4. Визначимо  $d$  – ключ дешифрування з рівняння  $3d \equiv 1 \pmod{20}$ .  
Для розв'язання рівняння використаємо розширений алгоритм Евкліда (див. вказівки до Лабб) та знайдемо  $d = 7$ .

# 1. Алгоритм RSA

## Приклад 1.2 (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  $C = 5\ 29\ 10\ 27\ 0$

$$c_1 = 14^3 \bmod 33 = ((14^2 \bmod 33) \cdot (14^1 \bmod 33)) \bmod 33 = (31 \cdot 14) \bmod 33 = 434 \bmod 33 = 5;$$

$$c_2 = 17^3 \bmod 33 = ((17^2 \bmod 33) \cdot (17^1 \bmod 33)) \bmod 33 = (25 \cdot 17) \bmod 33 = 425 \bmod 33 = 29;$$

$$c_3 = 10^3 \bmod 33 = 1000 \bmod 33 = 10;$$

$$c_4 = 3^3 \bmod 33 = 27 \bmod 33 = 27;$$

$$c_5 = 0^3 \bmod 33 = 0 \bmod 33 = 0.$$



# 3. Алгоритм RSA

## Приклад 1.3 (дешифрування):

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

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

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

$$\begin{aligned} m_2 &= 29^7 \bmod 33 = ((29^4 \bmod 33) \cdot (29^3 \bmod 33)) \bmod 33 = \\ &= (((29^2))^2 \bmod 33) \cdot (29^2 \bmod 33) \cdot (29 \bmod 33) \bmod 33 = (25 \cdot 16 \cdot 29) \bmod 33 = 11600 \bmod 33 = 17; \end{aligned}$$

$$\begin{aligned} m_3 &= 10^7 \bmod 33 = ((10^4 \bmod 33) \cdot (10^3 \bmod 33)) \bmod 33 = \\ &= (((10^2))^2 \bmod 33) \cdot (10^2 \bmod 33) \cdot (10 \bmod 33) \bmod 33 = (1 \cdot 1 \cdot 10) \bmod 33 = 10 \bmod 33 = 10; \end{aligned}$$

$$\begin{aligned} m_4 &= 27^7 \bmod 33 = ((27^4 \bmod 33) \cdot (27^3 \bmod 33)) \bmod 33 = \\ &= (((27^2))^2 \bmod 33) \cdot (27^2 \bmod 33) \cdot (27 \bmod 33) \bmod 33 = (9 \cdot 3 \cdot 27) \bmod 33 = 729 \bmod 33 = 3; \end{aligned}$$

$$m_5 = 0^7 \bmod 33 = 0 \bmod 33 = 0.$$

# 3. Алгоритм RSA

## Приклад 1.4:

**p**

12131072439211271897323671531612440428472427633701410925634549312301964  
37304208561932419736532241686654101705736136521417171171379797429933487  
1062829803541

**q**

12027524255478748885956220793734512128733387803682075433653899983955179  
85098879789986914690080913161115334681705083209602216014636634639181247  
0987105415233

**n**

14590676800758332323018693934907063529240187237535716439958187101987343  
87990053589383695714026701498021218180862924674228281570229220767469065  
43401224889672472407926969987100581290103199317858753663710862357656510  
507883714297115637342788911463535102712032765166518411726859837988672111  
837205085526346618740053

# 3. Алгоритм RSA

## Приклад 1.4:

**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

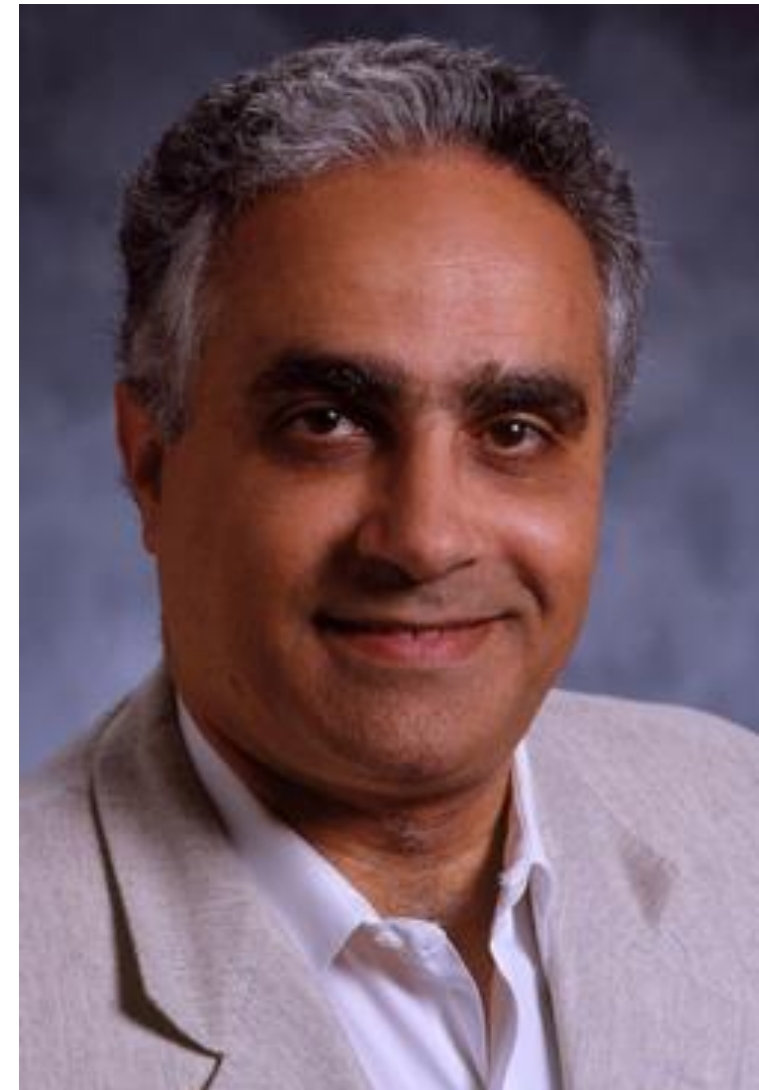
```
89489425009274444368228545921773093919669586065884257445497854456487674
83962981839093494197326287961679797060891728367987549933157416111385408
88132754881105882471930775825272784379065040156806234235500672400424666
65654232383502922215493623289472138866445818789127946123407807725702626
644091036502372545139713
```

## 2. 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

## 2. Алгоритм Ель-Гамала

### 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  $x$

## 2. Алгоритм Ель-Гамала

### 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

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

## 2. Алгоритм Ель-Гамала

### Приклад 2.1 (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  $x = 8$ .

## 2. Алгоритм Ель-Гамала

### Приклад 2.2 (шифрування):

Дано: повідомлення  $M = 5$ .

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

Обчислюємо число

$$\begin{aligned} a &= g^k \bmod p = 2^9 \bmod 11 \\ &= 512 \bmod 11 = 6 \end{aligned}$$

Обчислюємо число

$$\begin{aligned} b &= y^k M \bmod p = 3^9 \cdot 5 \bmod 11 \\ &= 19683 \cdot 5 \bmod 11 = 9 \end{aligned}$$

Пара  $(6, 9)$  є шифротекстом.

### Приклад 2.3 (дешифрування):

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

Обчислюємо  $M$  за формулою:

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

Отримали початкове повідомлення

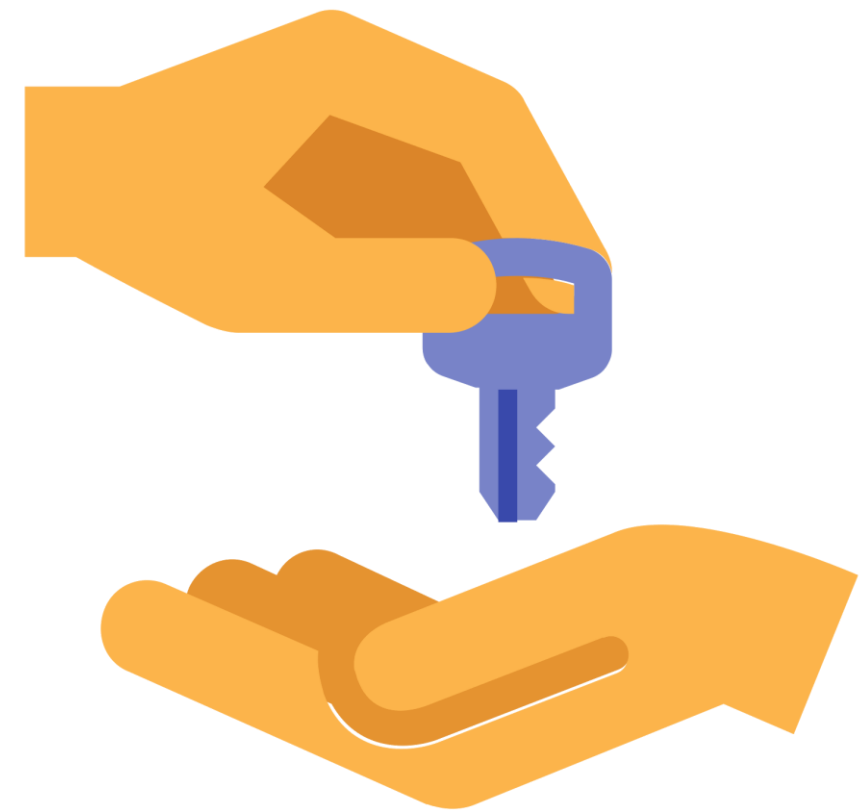
$$M = 5.$$



# 3. 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**



# 3. Алгоритм обміну ключами Діффі-Хелмана

## Алгоритм Діффі-Хелмана

1. Абоненти А і В спільно обирають просте число  $p$  і ціле число  $g$ , що є первісним коренем  $p$ .

2. Користувач А вибирає випадкове ціле число  $x < p$ , обчислює  $x_A = g^x \bmod p$  та відправляє його користувачеві В.

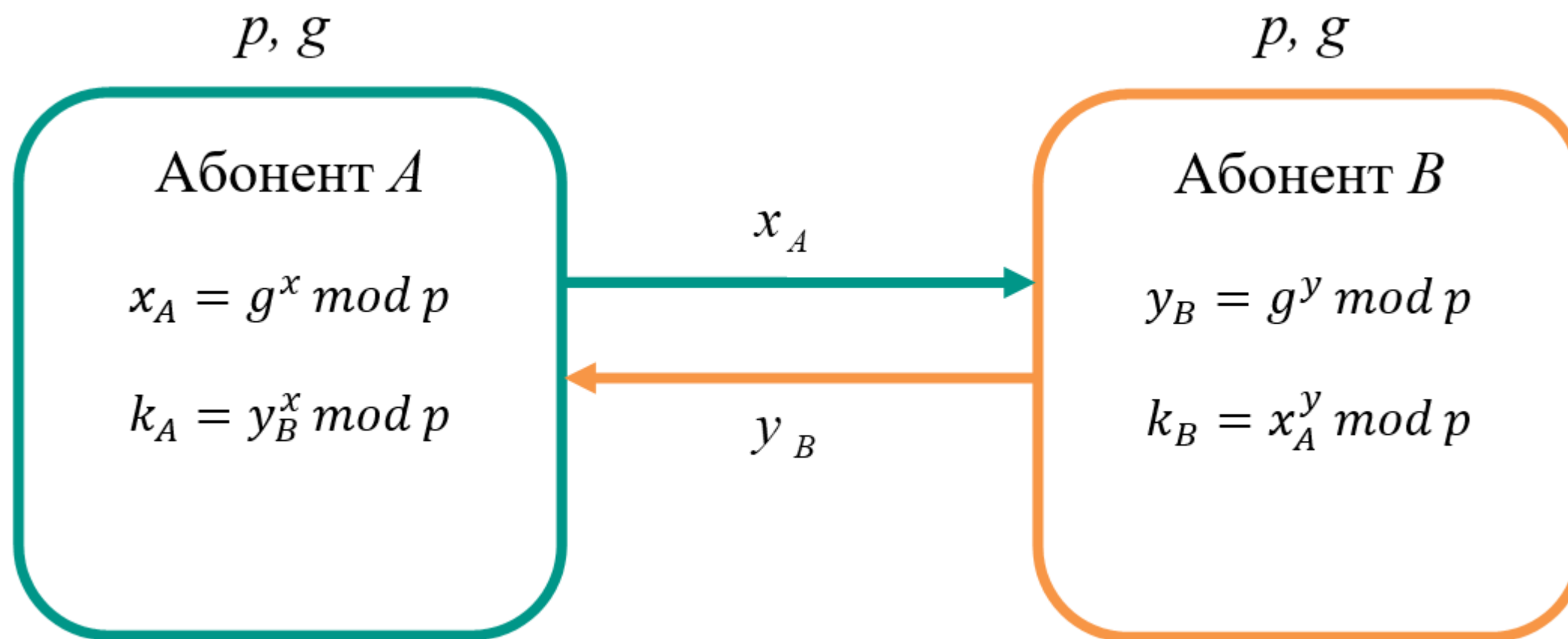
3. Користувач В вибирає випадкове ціле число  $y < p$ , обчислює  $y_B = g^y \bmod p$  та відправляє його користувачеві А.

4. Користувач А обчислює закритий ключ за формулою  $k_A = y_B^x \bmod p$ .

5. Користувач В обчислює закритий ключ за формулою  $k_B = x_A^y \bmod p$ .

# 3. Алгоритм обміну ключами Діффі-Хелмана

## Схема обміну ключами Діффі-Хелмана



# 3. Алгоритм обміну ключами Діффі-Хелмана

## Приклад 3.1:

1.  $p = 11, g = 2.$

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.  $k_A = 9^4 \bmod 11 = (9^2)^2 \bmod 11 = 4^2 \bmod 11 = 16 \bmod 11 = 5.$

5.  $k_B = 5^6 \bmod 11 = (5^3)^2 \bmod 11 = 4^2 \bmod 11 = 16 \bmod 11 = 5.$

**Секретний ключ**, обчислений обома сторонами – 5.