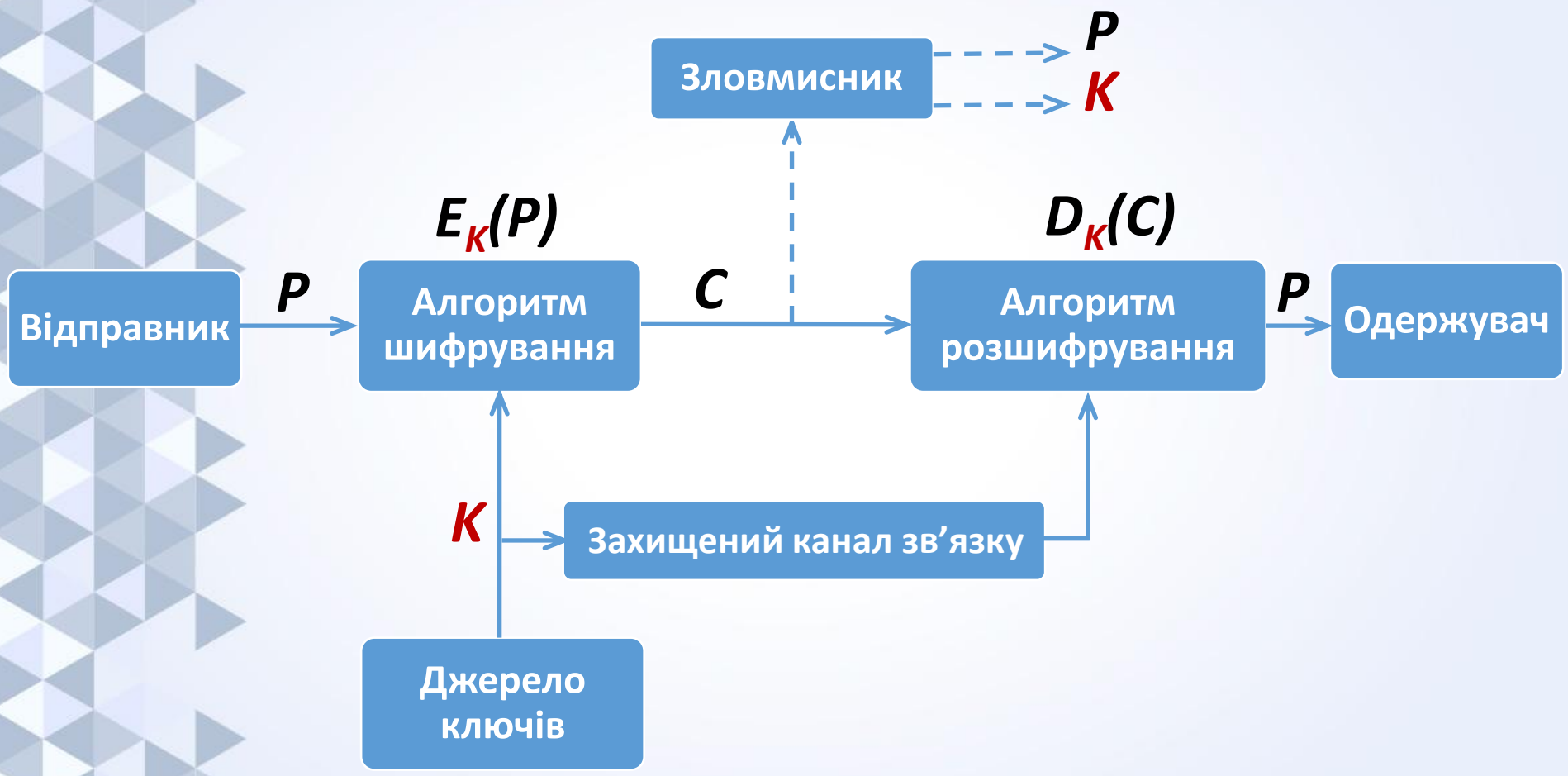
Алгоритми асиметричного шифрування

RSA

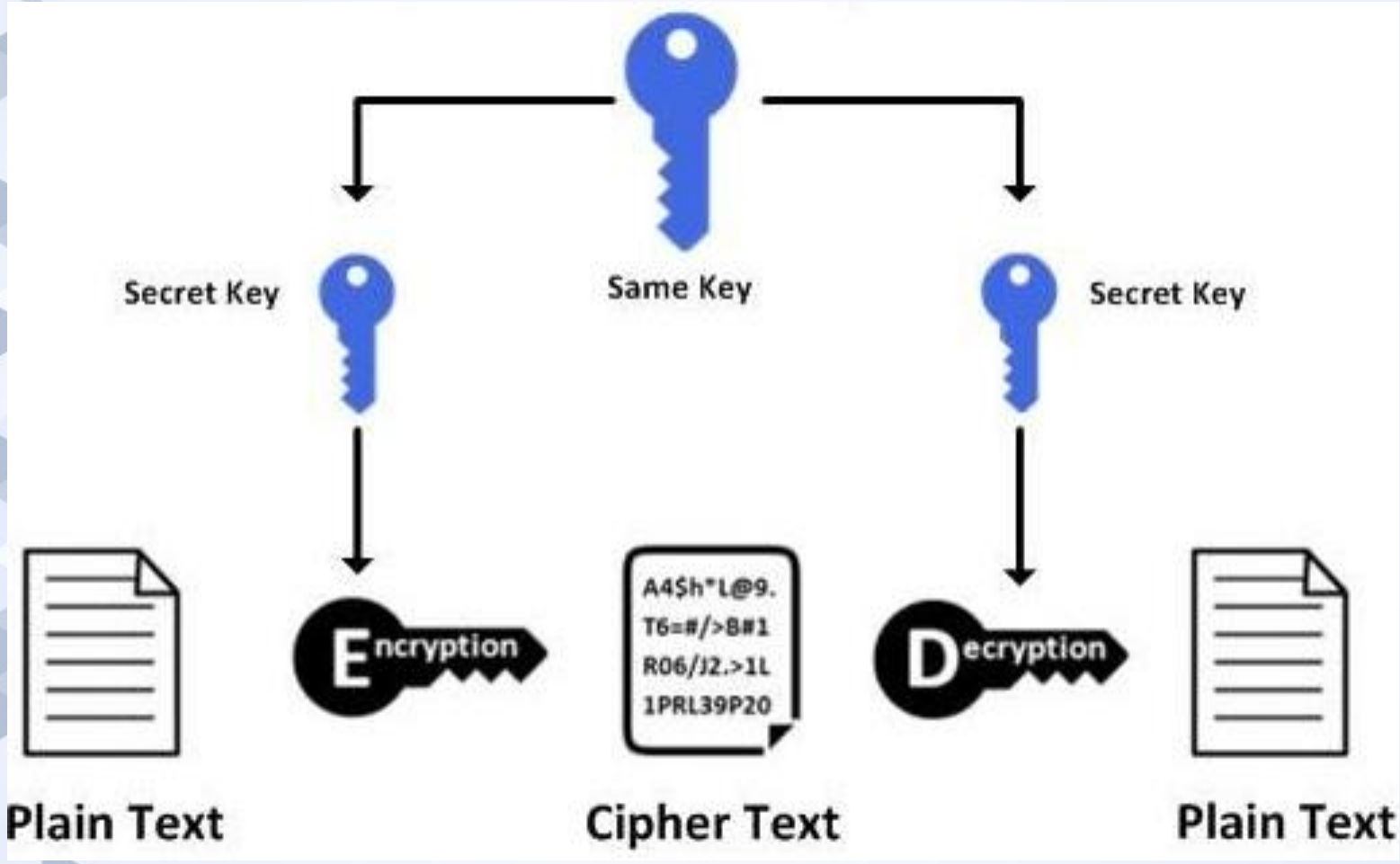
Розробила: старший викладач кафедри ПМА Бай Ю.П.

1. Організаційні питання
2. Симетричне і асиметричне шифрування
3. Створення RSA. Основні принципи
4. Реалізація RSA. Приклади

Симетрична криптосистема



Симетричне шифрування



Для шифрування і розшифрування використовується один і той самий ключ

Проблеми симетричного шифрування

Maxim

Bogdan

Daria

Svitlana

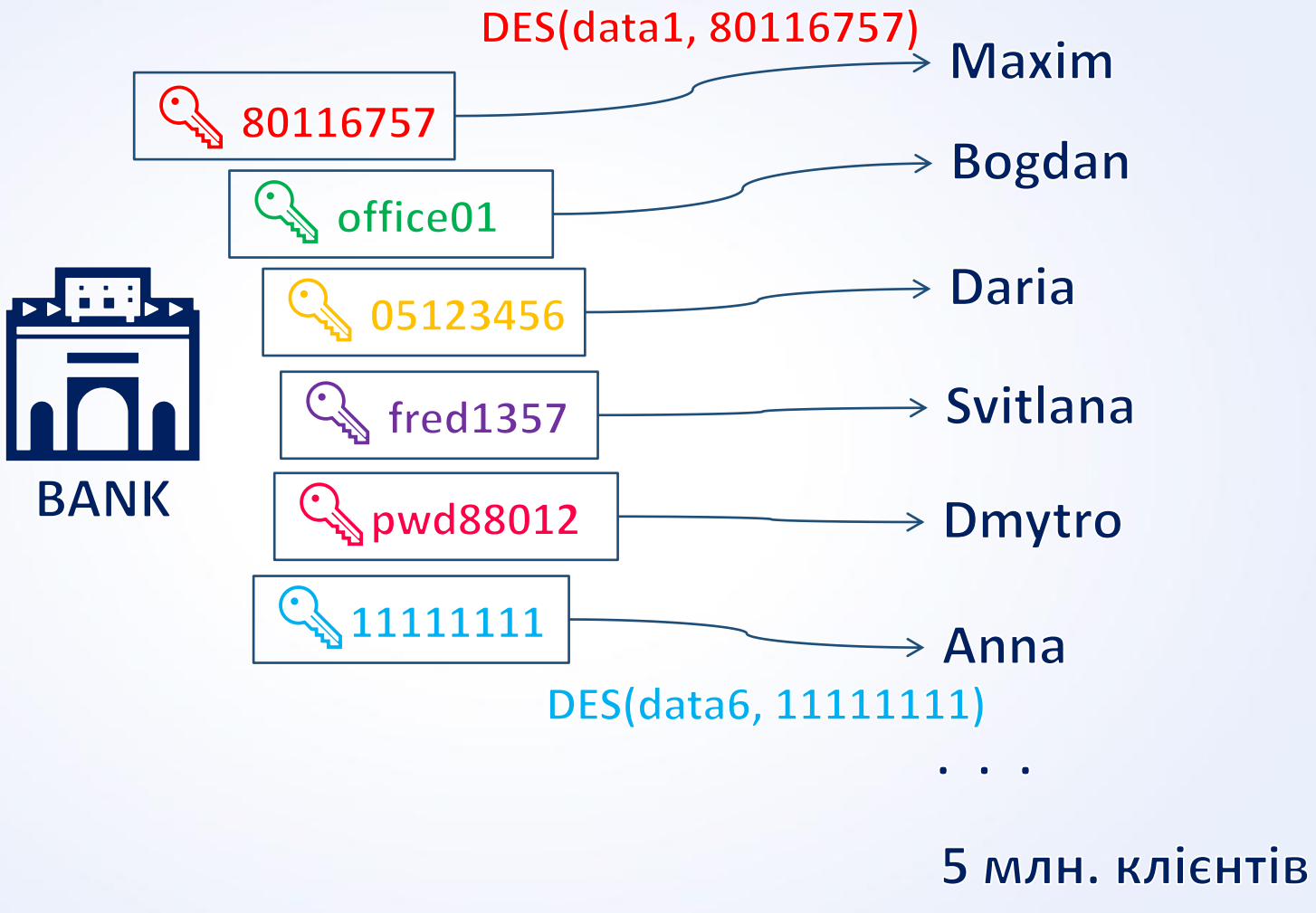
Dmytro

Anna

. . .

5 млн. клієнтів

Проблеми симетричного шифрування



Як надійно генерувати, зберігати і передавати ключі?

Проблеми симетричного шифрування

1) конфіденційність – як забезпечити секретність зберігання/передачі ключів?

2) аутентифікація – як перевірити, що ключ дійсно належить особі, від імені якої він надходить в систему? (*проблема нав'язування ключів зловмисником*)

Криптосистеми з відкритим ключем (алгоритми асиметричного шифрування)

В 1976 році **Діффі** та **Хелман** розробили метод, за допомогою якого вирішувались обидві вказані проблеми, і який кардинально відрізнявся від усіх раніше відомих підходів у криптографії за всю її 4-х тисячолітню історію існування.

Всі асиметричні алгоритми ґрунтуються на використанні так званих **односторонніх функцій з секретом (*one-way trapdoor function*)**.

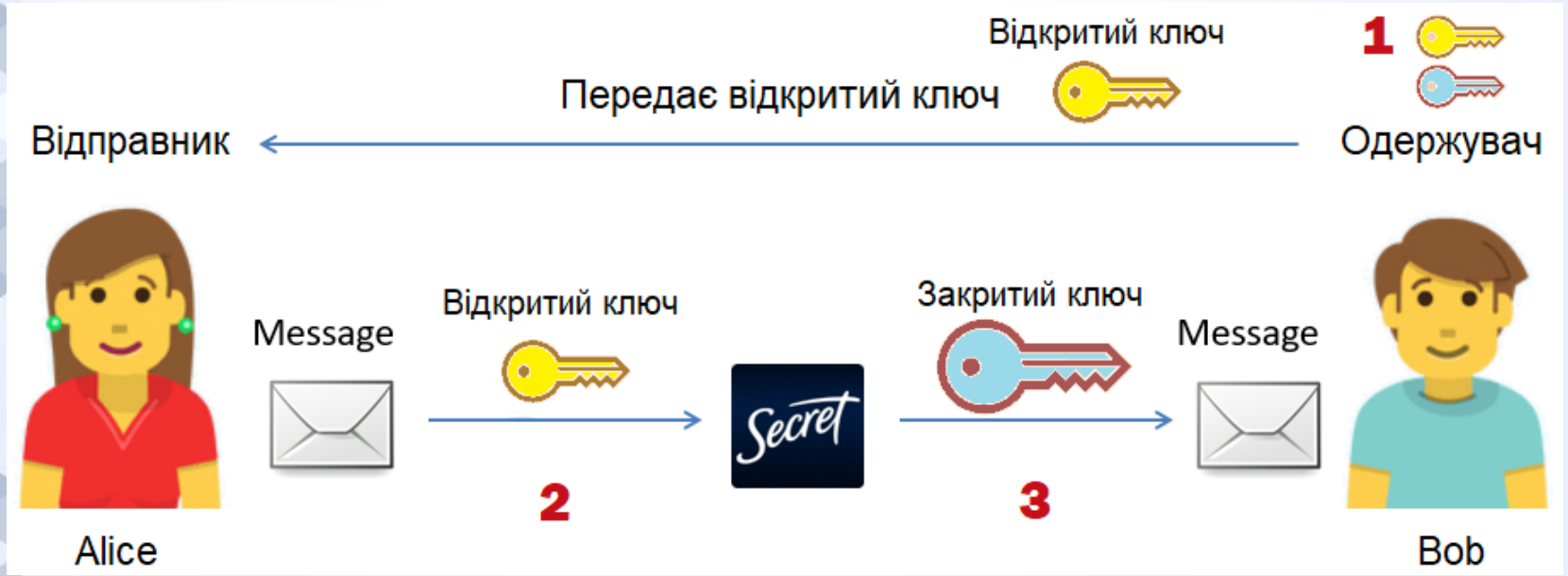
Одностороння функція з секретом (one-way trapdoor function)

1) f(x) є відомою відкритою функцією
2) для будь-якого x легко обчислити f(x)
3) для будь-якого y = f(x) складно обчислити x
4) f(x) легко обчислити, якщо відомий секрет t

Приклад: *$f(x) = x^3 \bmod n$, n – деяке задане число*

Криптосистеми з відкритим ключем (алгоритми асиметричного шифрування)

1) генеруються два різних ключі (**відкритий** і **закритий**), які пов'язані між собою алгоритмічно;
2) для шифрування інформації використовується відкритий ключ;
3) для розшифрування інформації використовується закритий ключ, який відомий лише одержувачу.



Криптосистеми з відкритим ключем (алгоритми асиметричного шифрування)

Алгоритми асиметричного шифрування використовують **два ключа**, які утворюють нерозривну пару.

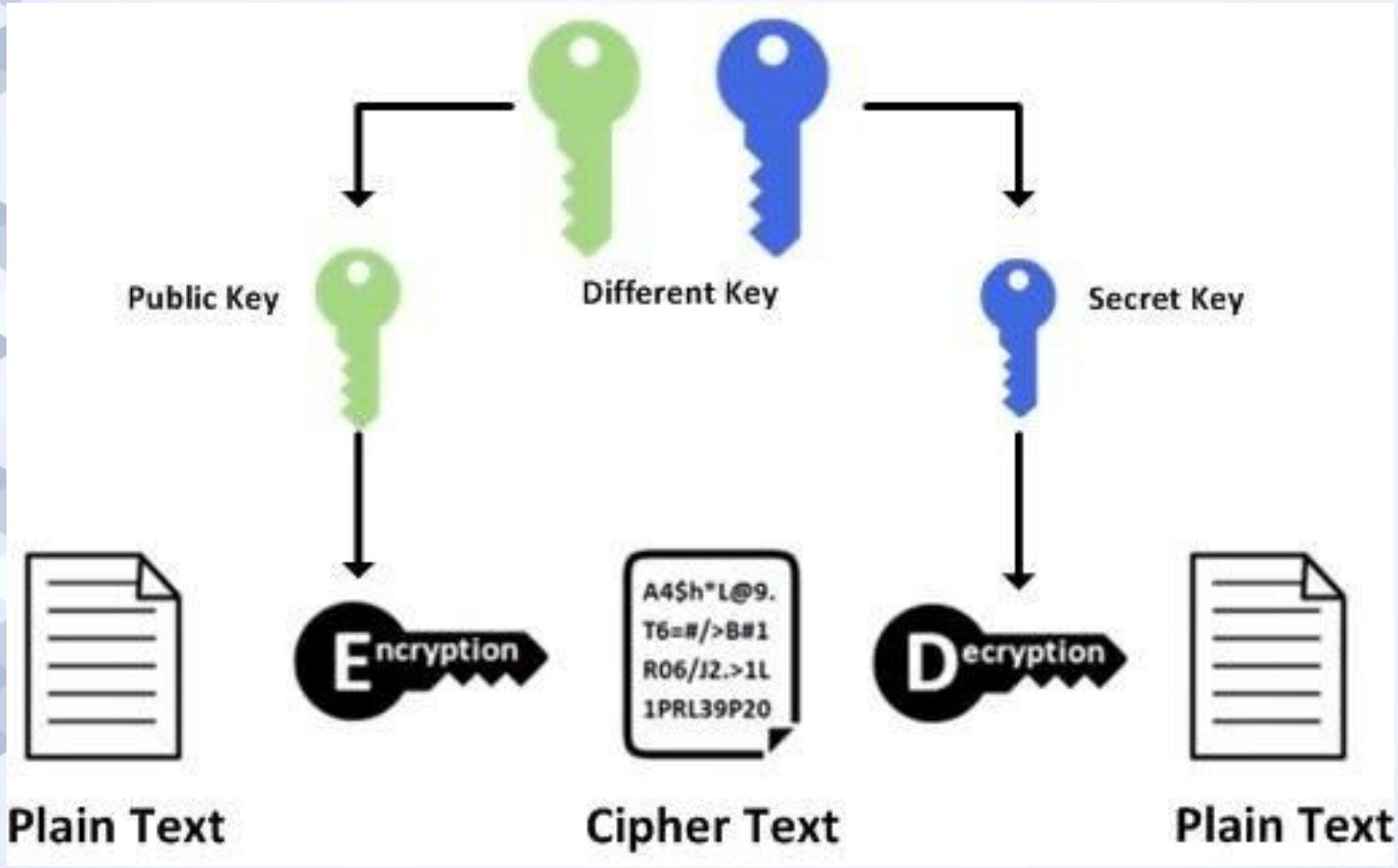
1) Одержувач генерує два ключі. Один ключ залишає собі, цей ключ називається **закритим** (private key). Другий ключ передається відправнику по відкритому каналу, він називається **відкритим ключем** (public key).

2) **Відкритий і закритий ключі пов'язані між собою алгоритмічно.**

3) Відправник знає лише відкритий ключ. За допомогою нього він шифрує повідомлення і надсилає його по відкритому каналу.

4) Розшифрувати повідомлення може лише одержувач за допомогою **закритого** ключа, який він тримає в секреті.

Асиметричне шифрування



Для шифрування і розшифрування використовуються два різні ключі

Алгоритми асиметричного шифрування є достатньо затратними за ресурсам і часом. На практиці для шифрування даних використовуються симетричні алгоритми, а розсилка ключів здійснюється з використанням алгоритмів асиметричного шифрування з **відкритим ключем одержувача даних**.

Симетричне шифрування

Передбачає використання **однакових ключів** для шифрування і розшифрування

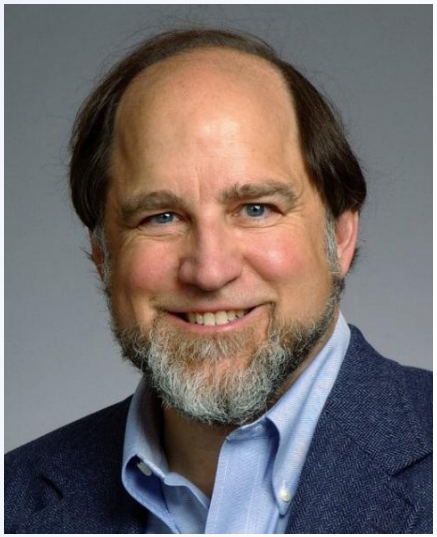
- DES
- Triple-DES
- IDEA
- BLOWFISH

Асиметричне шифрування

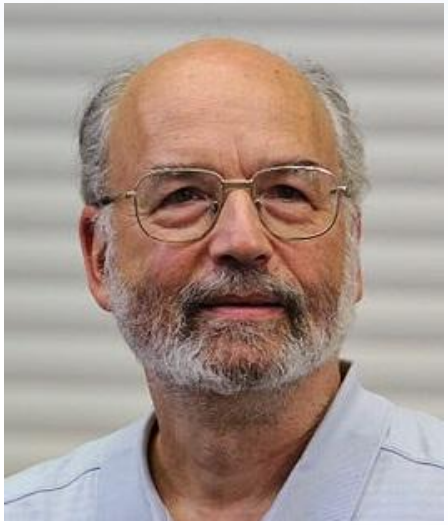
Алгоритм використовує **різні ключі** для шифрування і розшифрування

- RSA
- Diffie-Hellman
- ElGamal
- Eliptic-curve cryptography

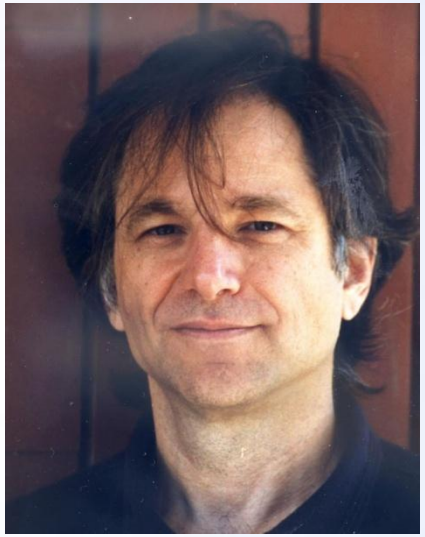
Алгоритм RSA (1977)



Ronald Rivest



Adi Shamir



Leonard Adleman

Математичне підґрунтя RSA

В основі алгоритму RSA полягає складність задачі факторизації великих чисел. **Факторизація числа** – це розклад числа на прості множники.

$p = 17,\ q = 19 \quad \rightarrow \quad n = p \cdot q = 323$

$n = 323 \quad \rightarrow \quad p = ?,\ q = ?$

$p = 3557,\ q = 2579 \quad \rightarrow \quad n = p \cdot q = 9173503$

$n = 9173503 \quad \rightarrow \quad p = ?,\ q = ?$ (назад складно)

Число *n* повинне мати розмір **не менше 512 біт**. На 2007 рік система шифрування на основі RSA вважалась надійною, починаючи з величини *n* в 1024 біти.

Етапи RSA

1) Генерація відкритого і закритого ключів.
2) Публікація відкритого ключа.
3) Шифрування за допомогою відкритого ключа.
4) Розшифрування за допомогою закритого ключа.

Етапи RSA

1) Одержувач ***B*** генерує ключі: **відкритий ключ** (*e, n*) та **закритий ключ** (*d, n*).
2) Одержувач ***B*** надсилає відправнику ***A*** відкритий ключ (*e, n*).
3) Користуючись ключем (*e, n*) відправник ***A*** шифрує повідомлення (це може бути, наприклад, 8-байтний ключ для DES) і надсилає одержувачу ***B*** по відкритому каналу.
4) Одержувач ***B*** розшифровує інформацію, отриману від ***A***, користуючись закритим ключем (*d, n*).