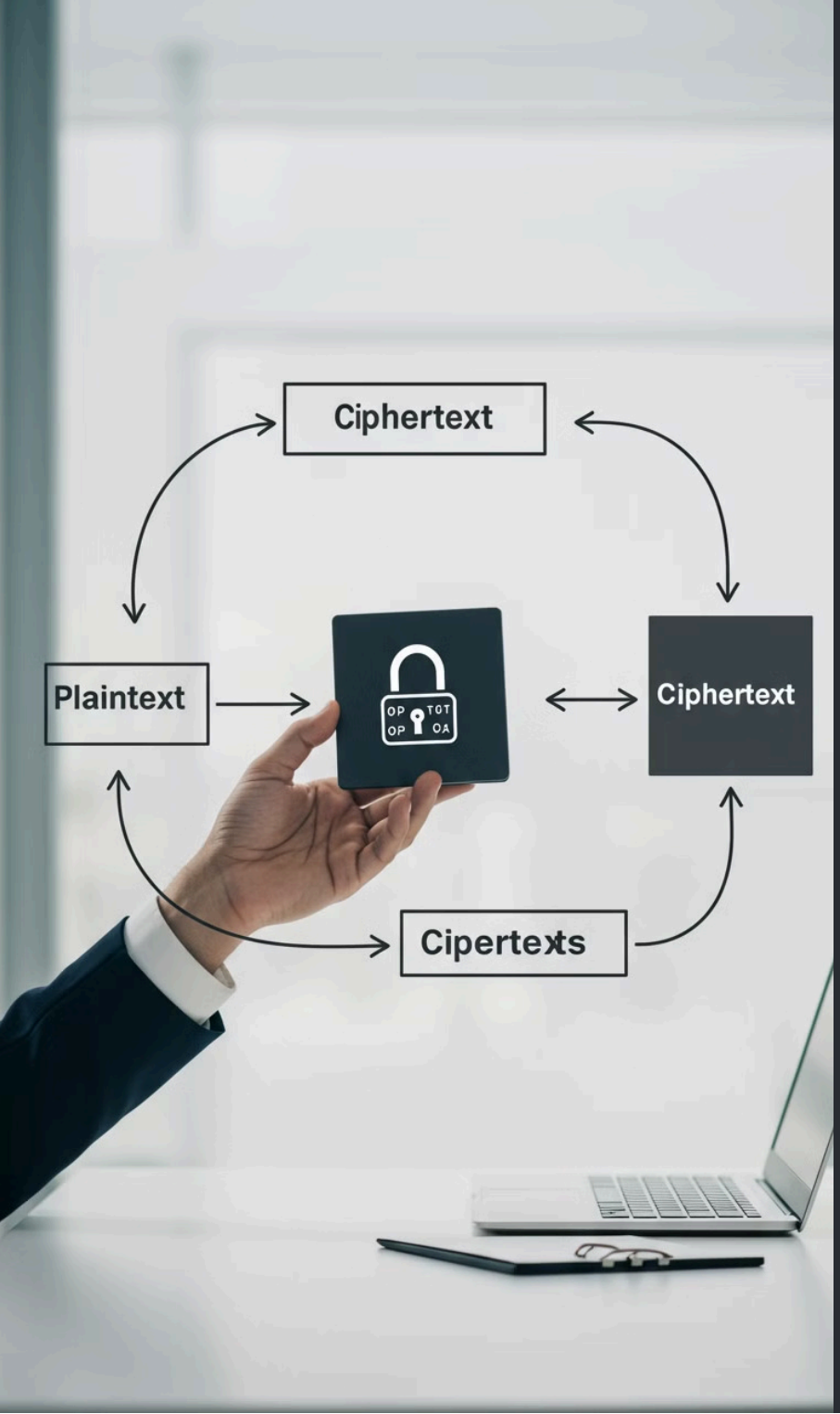
Основи Шифрування Повідомлень

Вітаємо вас на вступі до світу криптографії! У цій презентації ми розглянемо основні принципи шифрування та дешифрування повідомлень, використовуючи простий алгоритм шифрування. Ви дізнаєтесь, як захистити свої повідомлення від сторонніх очей та зрозумієте, як працюють базові криптографічні механізми.

Шифрування - це процес перетворення звичайного тексту в незрозумілий код, який можна прочитати лише за наявності спеціального ключа. Цей навик є фундаментальним у сучасному цифровому світі та використовується для захисту конфіденційної інформації.

by o d



Що Таке Шифрування?

Шифрування — це процес перетворення звичайного тексту (відкритого повідомлення) в незрозумілий набір символів (шифротекст) за допомогою певного алгоритму та ключа. Основна мета шифрування — забезпечити конфіденційність інформації, захищаючи її від несанкціонованого доступу.

У світі комп'ютерів шифрування базується на математичних функціях, які перетворюють вхідні дані таким чином, що відновити початкове повідомлення без знання ключа практично неможливо.

Ключ шифрування

Секретна інформація, що використовується для шифрування та дешифрування повідомлень

Шифротекст

Зашифроване повідомлення, яке виглядає як незрозумілий набір символів

Дешифрування

Процес перетворення шифротексту назад у відкритий текст за допомогою ключа

Алгоритм Шифрування

Наш алгоритм шифрування базується на перетворенні символів повідомлення в їхні ASCII-коди з подальшою модифікацією цих кодів за допомогою ключа. ASCII (American Standard Code for Information Interchange) — це стандартна таблиця, яка присвоює числові значення символам, які використовуються в комп'ютерах.

Кожен символ у вихідному повідомленні та ключі перетворюється в його ASCII-код. Потім код символу повідомлення додається до коду відповідного символу ключа, створюючи новий ASCII-код, який перетворюється назад у символ.

T

Розбиття тексту

Розділяємо повідомлення та ключ на окремі символи

01
10

Конвертація в ASCII

Перетворюємо кожен символ у відповідний ASCII-код

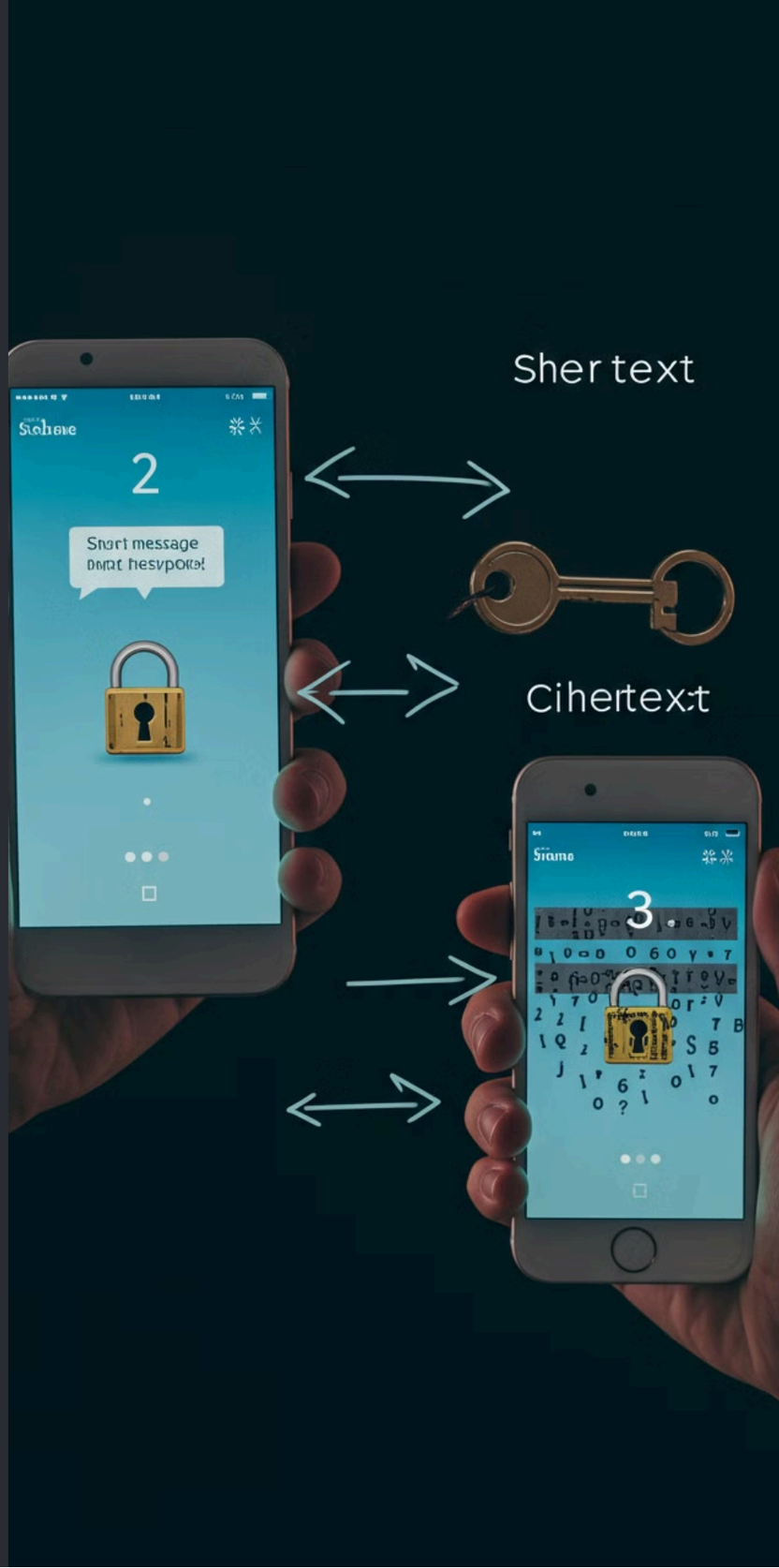
Математична операція

Додаємо ASCII-коди повідомлення та ключа

A

Конвертація назад

Перетворюємо нові ASCII-коди назад у символи



Необхідні функції

Для реалізації нашого алгоритму шифрування потрібні такі JavaScript функції:

- **split('')** — розбиває текст на масив символів
- **charCodeAt(0)** — перетворює символ у ASCII-код (0-255)
- **map()** — застосовує функцію до кожного елемента масиву
- **String.fromCharCode()** — конвертує ASCII-код назад у символ
- **join('')** — об'єднує масив символів у рядок

Ці функції працюють послідовно, трансформуючи повідомлення у шифротекст через математичні операції з ASCII-кодами. При шифруванні коди додаються, при дешифруванні — віднімаються.

Функції Шифрування та Дешифрування

Для роботи з шифруванням нам потрібні дві основні функції: одна для шифрування повідомлення, інша для його дешифрування. Обидві функції працюють із символами та їхніми ASCII-кодами, але виконують протилежні операції.

При шифруванні ми додаємо ASCII-коди, а при дешифруванні віднімаємо їх. Це дозволяє нам перетворити відкритий текст у шифротекст і навпаки, щоб отримати початкове повідомлення.

Функція шифрування

```
function encryptMsg(msg, key) {
  const msgChars = msg.split('');
  const msgASCI = msgChars.map(
    item => item.charCodeAt(0));
  const keyASCI = key.split('').map(
    item => item.charCodeAt(0));
  const encryptedMsgASCI = msgASCI.map(
    (value, index) =>
    value + keyASCI[index]);
  const encryptedMsg = encryptedMsgASCI
    .map(value =>
    String.fromCharCode(value))
    .join('');
  return encryptedMsg;
}
```

Функція дешифрування

```
function decryptMsg(msg, key) {
  const msgChars = msg.split('');
  const msgASCI = msgChars.map(
    item => item.charCodeAt(0));
  const keyASCI = key.split('').map(
    item => item.charCodeAt(0));
  const decryptedMsgASCI = msgASCI.map(
    (value, index) =>
    value - keyASCI[index]);
  const decryptedMsg = decryptedMsgASCI
    .map(value =>
    String.fromCharCode(value))
    .join('');
  return decryptedMsg;
}
```

Приклад Роботи Алгоритму

Розглянемо приклад роботи нашого алгоритму шифрування на практиці. Припустимо, у нас є повідомлення "hello World" і ключ "hello Its my key". Процес шифрування відбуватиметься за описаним алгоритмом.

Зверніть увагу, що довжина ключа має бути не меншою за довжину повідомлення. У нашому випадку ключ довший, що забезпечує надійне шифрування кожного символу повідомлення.

Вхідні дані

Повідомлення: "hello World"

Ключ: "hello Its my key"

Розбиття на символи

Повідомлення: ['h','e','l','l','o',' ','W','o','r','l','d']

Ключ: ['h','e','l','l','o',' ','I','t','s',' ','m','y',' ','k','e','y']

Конвертація в ASCII

Повідомлення: [104,101,108,108,111,32,87,111,114,108,100]

Ключ: [104,101,108,108,111,32,73,116,115,32,109,121,32,107,101,121]

Додавання ASCII-кодів

[208,202,216,216,222,64,160,227,229,140,209]

Результат шифрування

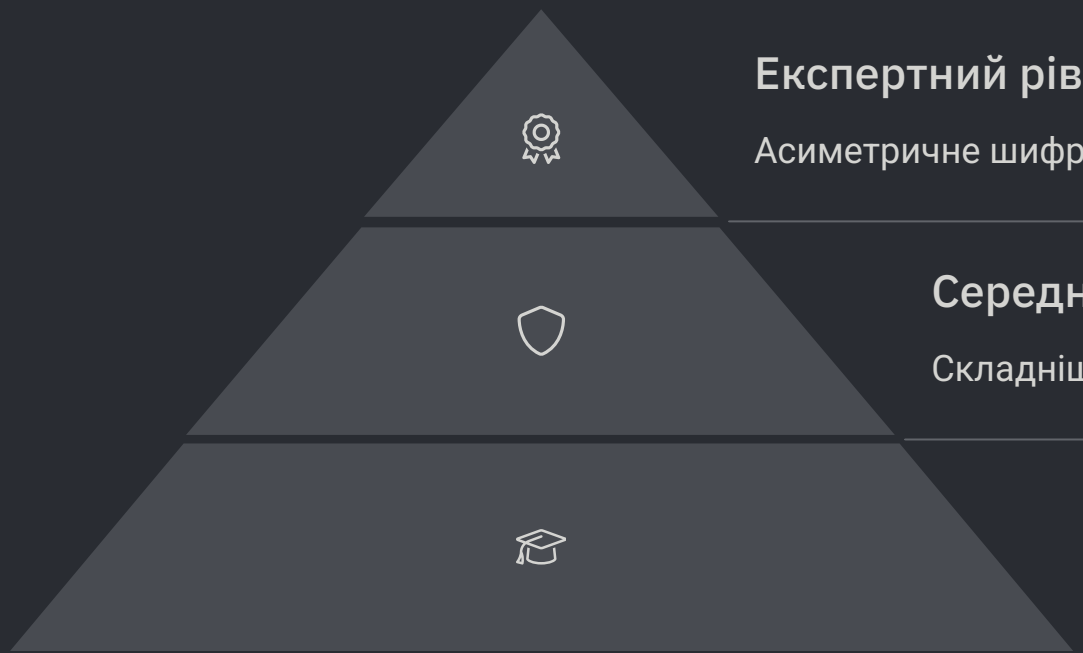
Зашифроване повідомлення: "ÐÊØØÞ@ åå"

Після дешифрування: "hello World"

Практичне Застосування та Висновки

Шифрування відіграє критичну роль у захисті конфіденційності та безпеки інформації в цифровому світі. Хоча наш алгоритм є простим прикладом, він демонструє фундаментальні принципи, на яких базуються сучасні криптографічні системи.

Важливо розуміти, що реальні системи шифрування використовують набагато складніші алгоритми і ключі. Проте, вивчення основ допомагає краще зрозуміти, як працює захист інформації.



Експертний рівень

Асиметричне шифрування та криптографічні протоколи

Середній рівень

Складніші алгоритми з ключами

Початковий рівень

Базове шифрування з ASCII-кодами