

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 4
з курсу: *«Безпека програмного забезпечення»*

Виконав:
студент 4-го курсу,
групи ТВ-21
Цвігун Богдан

Київ 2025

Практична робота № 4

Завдання:

1. Розробіть інтерфейс криптографічної системи для шифрування за допомогою DES з використанням всіх можливих режимів.
2. Ознайомтесь з описом класів `CryptographicServiceProvider` і `CryptoStream` бібліотеки `.NET Framework`.
3. Реалізуйте шифрування DES, використовуючи класи `.NET Framework`.
4. Виконайте тестування роботи системи.

Хід виконання:

Для розробки веб-додатку було використано фреймворк Flask, який забезпечує ефективну взаємодію між серверною частиною та інтерфейсом користувача. Інтерфейс реалізовано з використанням сучасних веб-технологій, зокрема CSS-фреймворку Bulma, що надає адаптивний і естетичний дизайн, а також шаблонізатора Jinja2, який відповідає за динамічне формування вмісту HTML-сторінок.

The screenshot shows a web application titled "DES Cipher" with a navigation bar at the top containing links: "DES Cipher", "Головна", "Про розробника", and "Вийді". The main content area is titled "Шифрування/розшифрування з використанням DES (ECB/CBC/CFB/OFB)". It features a form for inputting text and keys, a dropdown for selecting the mode (currently set to "ECB"), and buttons for "Шифрувати" (Encrypt) and "Розшифрувати" (Decrypt). The "Результат" (Result) section shows "Порожньо" (Empty). A "Додатково" (Additional) section provides more details about the encryption process. On the right side, there is a "Про розробника" (About the developer) section with a warning about the security of DES and a note about the developer's name.

DES Cipher

Головна Про розробника Вийді

Підтримка EN / UA

Шифрування/розшифрування з використанням DES (ECB/CBC/CFB/OFB)

Про розробника

Текст

Введіть або вставте текст...

При шифруванні приймається значення тексту. При розшифруванні — базові рядки.

Ключ (8 символів)

8 символів (ASCII)

Ключ повинен бути рівно 8 байт.

IV (для CBC/CFB/OFB, 8 символів)

IV (8 символів)

Потрібно лише для режимів, відмінних від ECB.

Режим

ECB

Шифрувати Розшифрувати

Результат

Порожньо

Додатково

Шифрування повертає базові рядки (щоб безпечно передавати двійкові дані). При розшифруванні подавайте базові рядки.

Поради та попередження

DES застарілий і небезпечний для реального використання — використовуйте AES для продуктивних рішень.

Ключ і IV мають бути точно 8 байт (символів ASCII). Переконайтесь, що немає зайвих пробілів.

Для CBC/CFB/OFB IV треба генерувати випадково й передавати/зберігати разом із шифротекстом.

Розробник: TB-21 Цайгун Богдан

DES — демонстраційний приклад симетричного шифрування

Шифр DES (Data Encryption Standard) є класичним алгоритмом симетричного блочного шифрування, розробленим у 1970-х роках компанією IBM та затвердженим Національним інститутом стандартів і технологій США (NIST) у 1977 році як федеральний стандарт. DES працює з блоками даних розміром 64 біти та використовує ключ довжиною 56 біт, за допомогою якого виконується послідовність із 16 раундів перестановок і підстановок.

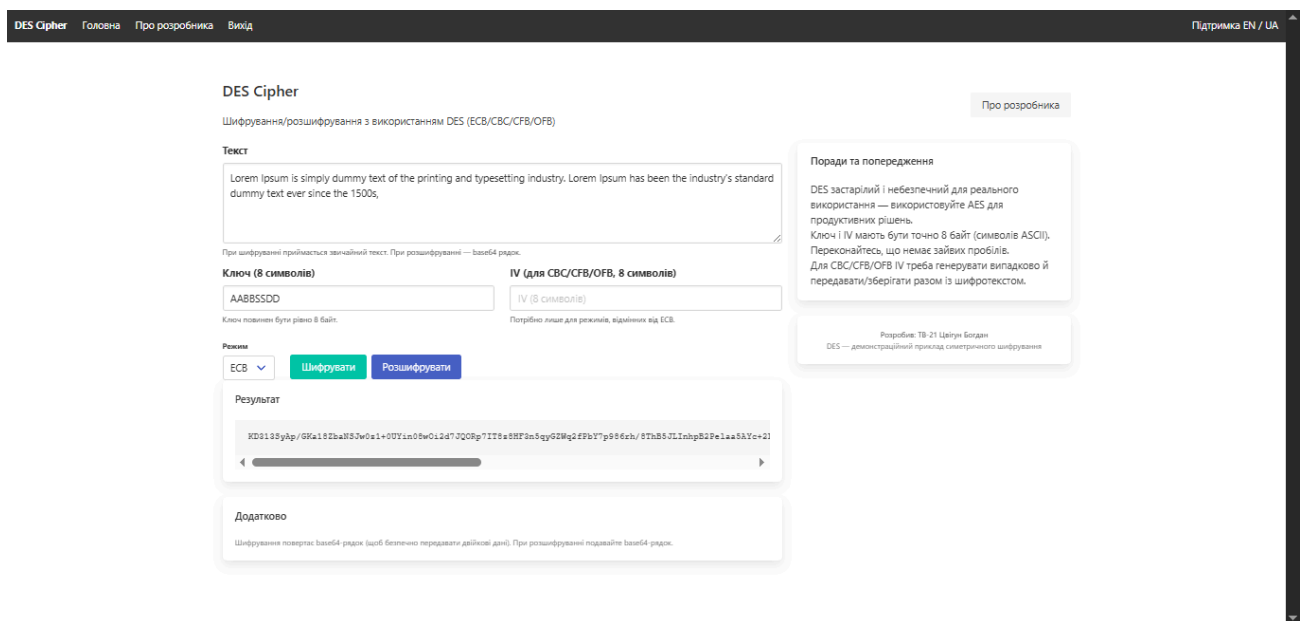
Основний принцип DES полягає у поєднанні операцій заміни (S-box) та перестановки (P-box), що забезпечує як дифузію, так і плутанину — ключові властивості надійного шифрування.

Попри історичну важливість і довготривале використання у фінансових та урядових системах, DES нині вважається криптографічно застарілим, оскільки його ключ можна зламати за допомогою сучасних обчислювальних засобів методом повного перебору.

Для підвищення безпеки було створено модифікацію Triple DES (3DES), у якій алгоритм виконується тричі з різними ключами. Проте сьогодні для практичного використання рекомендується застосовувати сучасні стандарти, зокрема AES (Advanced Encryption Standard), який забезпечує значно вищий рівень криптостійкості.

```
116
117 @app.route("/des/", methods=["GET", "POST"])
118 def des_cipher():
119     result = None
120     error = None
121     formdata = {}
122
123     if request.method == "POST":
124         formdata = request.form.to_dict()
125         text = request.form.get("text", "")
126         key = request.form.get("key", "")
127         iv = request.form.get("iv", "")
128         mode = request.form.get("mode", "ECB")
129         action = request.form.get("action", "encrypt")
130
131         try:
132             # Перевірка ключа (рівно 8 байт)
133             if len(key.encode()) != 8:
134                 raise ValueError("Ключ має бути рівно 8 символів (8 байт).")
135
136             # Вибір режиму DES
137             mode_map = {
138                 "ECB": DES.MODE_ECB,
139                 "CBC": DES.MODE_CBC,
140                 "CFB": DES.MODE_CFB,
141                 "OFB": DES.MODE_OFB
142             }
143
144             if mode not in mode_map:
145                 raise ValueError("Невідомий режим DES.")
146
147             # Ініціалізація шифра
148             if mode == "ECB":
149                 cipher = DES.new(key.encode(), mode_map[mode])
150             else:
151                 if len(iv.encode()) != 8:
```

Повний вихідний код проекту доступний у репозиторії, що буде додано до завдання.



Ендпойнт `/des/` реалізує веб-інтерфейс для симетричного блочного шифру DES (Data Encryption Standard) у межах Flask-застосунку. Його основне призначення полягає у забезпеченні взаємодії між користувачем і серверною частиною для виконання процесів шифрування та розшифрування тексту. Під час обробки запиту методом POST ендпойнт отримує з форми дані — вхідний

текст, ключ, ініціалізаційний вектор, режим роботи та обрану дію. Перед початком обчислень виконується перевірка коректності введених параметрів: ключ повинен містити рівно вісім байт, а ініціалізаційний вектор (IV), якщо він використовується, також має відповідати цій довжині.

Далі відбувається створення об'єкта шифра відповідно до вибраного режиму, серед яких підтримуються ECB, CBC, CFB та OFB. У режимі шифрування текст доповнюється до довжини, кратної розміру блоку, після чого шифрується і кодується у формат Base64 для коректного відображення в інтерфейсі. У режимі розшифрування виконується зворотна процедура — вхідні дані декодуються, розшифровуються та очищуються від заповнення.

Результат операції або повідомлення про можливі помилки передаються у шаблон DES.html, який відображає відповідну інформацію користувачеві. Така реалізація забезпечує зручну демонстрацію принципів роботи блочного симетричного шифру DES та дозволяє на практиці спостерігати відмінності між різними режимами його роботи.

DES Cipher

Головна

Про розробника

Вийді

Підтримка EN / UA

DES Cipher

Шифрування/розшифрування з використанням DES (ECB/CBC/CFB/OFB)

Про розробника

Текст

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,

При шифруванні приймається значайний текст. При розшифруванні — base64 рядок.

Ключ (8 символів)

IV (для CBC/CFB/OFB, 8 символів)

AAB855DD

DD588AA

Ключ повинен бути рівно 8 байт.

Потрібно лише для режимів, відомих від ECB.

Режим

CFB

Шифрувати

Розшифрувати

Результат

i09V4ya1s0CKTCu0d18Q0B96Mx1ngX0ck/baka9eW8MMe5T/116.3Wew9Tn5/g8qQD9oKXEdhWuv7R12vnc7n7bQ0Zv9u8FFt

<

>

Додатково

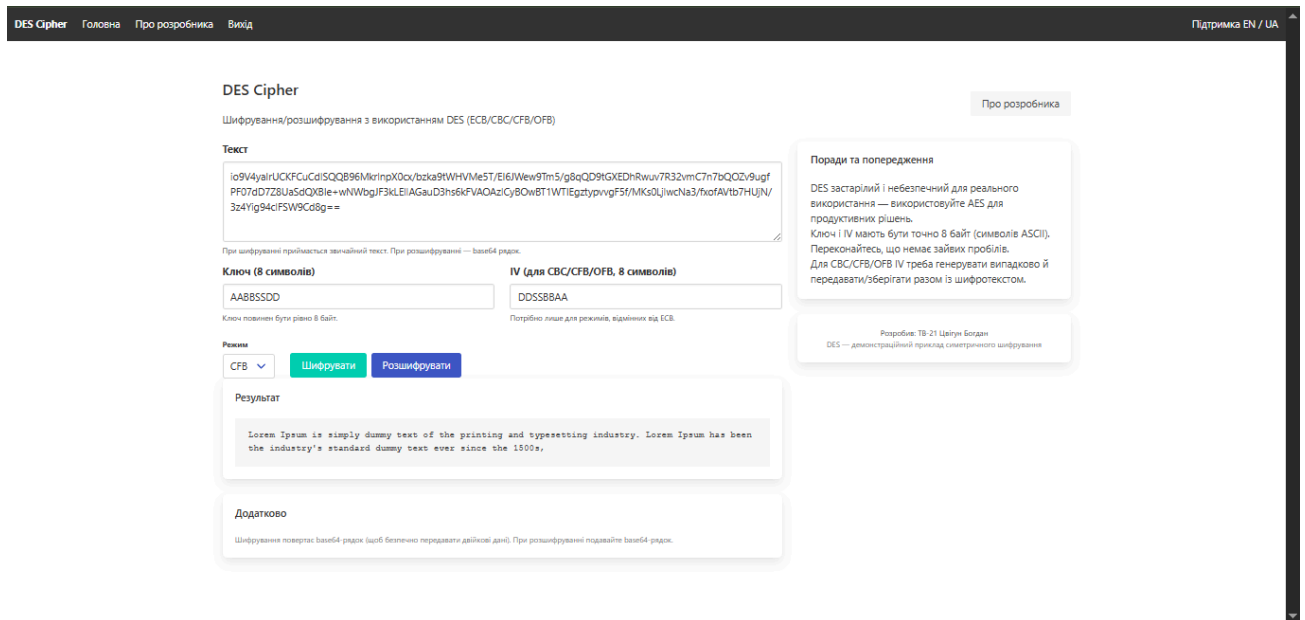
Шифрування повертає base64 рядок (щоб безпечно передавати довгий дані). При розшифруванні подається base64 рядок.

Поради та попередження

DES застарілий і небезпечний для реального використання — використовуйте AES для продуктивних рішень. Ключ і IV мають бути точно 8 байт (символів ASCII). Переконайтесь, що немає зайвих пробілів. Для CBC/CFB/OFB IV треба генерувати випадково й передавати/зберігати разом із шифротекстом.

Розробник: ТБ-21 Цілий Богдан

DES — демонстраційний приклад симетричного шифрування



Висновок:

У результаті виконання практичної роботи було створено веб-інтерфейс для симетричного блочного шифрування за допомогою алгоритму DES з підтримкою всіх основних режимів роботи (ECB, CBC, CFB, OFB). Розроблена система забезпечує коректну обробку ключів та ініціалізаційних векторів, виконує шифрування і розшифрування тексту та відображає результат користувачеві у зручному форматі. Реалізація на базі Flask і використання сучасних веб-технологій дозволяє продемонструвати принципи роботи DES у навчальних цілях, а також надає можливість тестування різних режимів шифрування та їхнього впливу на результат, забезпечуючи практичне розуміння роботи симетричних криптографічних алгоритмів.