

Універсальний шифр Цезаря

Універсальний шифр Цезаря — це класичний метод шифрування, що відноситься до симетричних методів, де для шифрування та розшифрування використовується один і той самий ключ. Основний принцип цього шифру полягає в тому, що кожен літеру в тексті зсувають на певну кількість позицій у алфавіті. Це означає, що, наприклад, якщо ми зсуваємо літеру "А" на одну позицію вперед, вона стане "Б", а якщо зсуваємо на дві — то "В" і так далі. Зсув може бути як вліво, так і вправо в алфавіті.

У Універсальному шифрі Цезаря цей метод розширюється до всіх символів Unicode, тобто шифрування може працювати не лише з літерами, але й з іншими символами, такими як числа, пробіли, знаки пунктуації та навіть спеціальні символи. Це робить шифр набагато більш універсальним і придатним для шифрування різних видів текстів, незалежно від їхнього вмісту.

Як користуватися Універсальним шифром Цезаря:

1. Введіть текст: Для початку потрібно ввести текст, який ви хочете зашифрувати або розшифрувати. Це може бути будь-який текст, що містить літери, цифри, спеціальні символи або пробіли.

2. Вкажіть ключ: Ключ для шифрування в цьому методі — це число, яке вказує, на скільки позицій потрібно зсунути кожен символ у введеному тексті. Наприклад, якщо ви обираєте ключ 3, це означає, що кожен символ буде зсунений на три позиції вперед у таблиці Unicode. Чим більший ключ, тим більше буде зсув символів.

3. Натисніть кнопку "Зашифрувати" або "Дешифрувати": Після введення тексту і визначення ключа, можна натискати одну з двох кнопок:

- **Зашифрувати:** Це створить зашифрований текст, в якому кожен символ буде зсунутий на задану кількість позицій.

- **Дешифрувати:** Якщо ви маєте зашифрований текст і хочете повернути його до початкового вигляду, вам потрібно вказати той самий ключ і натиснути "Дешифрувати". Це поверне текст до його оригінальної форми, оскільки процес шифрування і розшифрування однаковий — лише зворотний зсув.

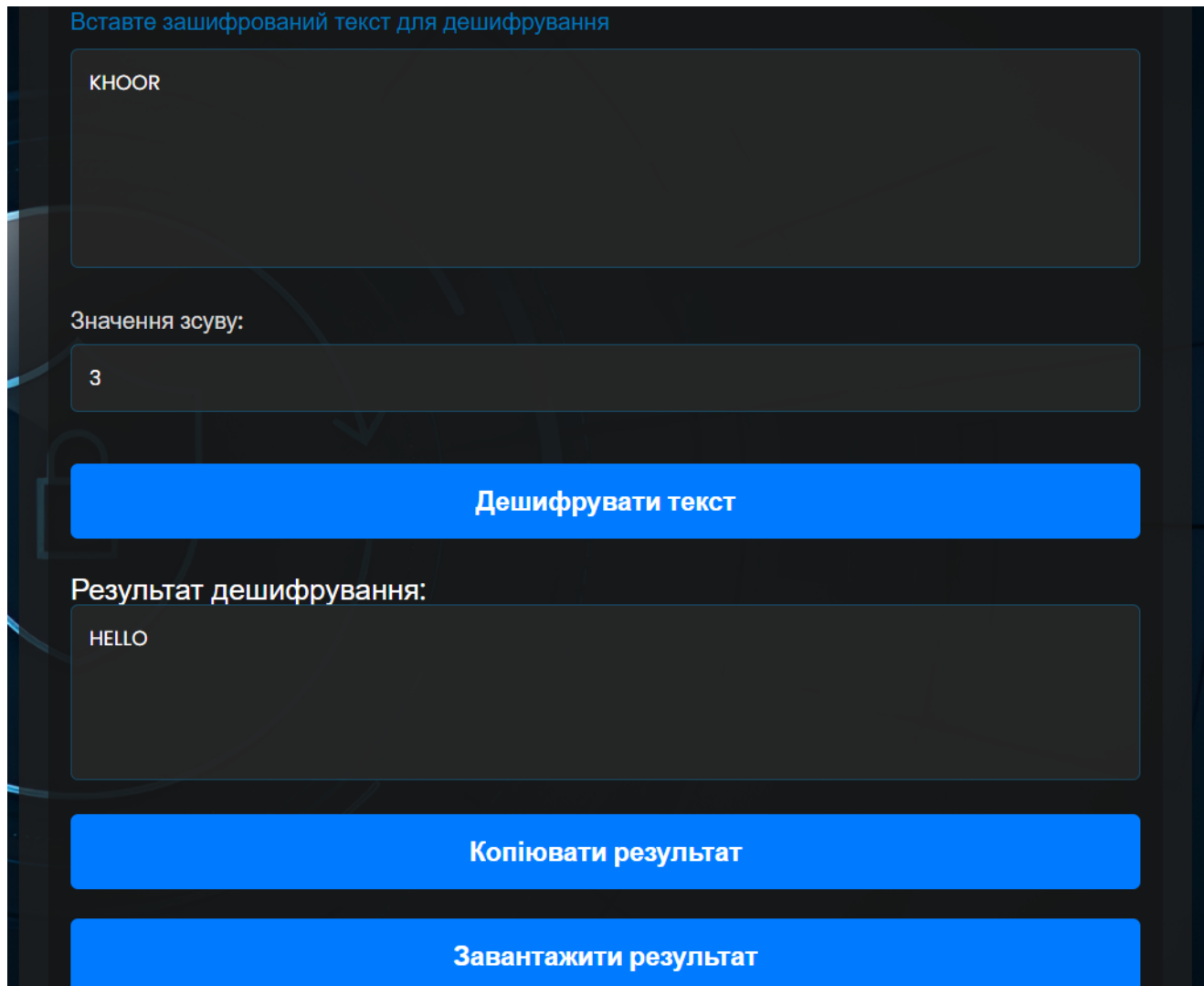
Універсальний шифр Цезаря — це простий, але ефективний метод для навчання основам криптографії. Хоча цей метод і не є надто надійним для захисту конфіденційних даних, він все ще широко використовується в різних криптографічних навчальних програмах і для базових потреб шифрування.

Приклад шифрування продемонстрований на рисунку 1.1.

The image shows a web application interface for the Caesar cipher. At the top, a blue header contains the text "Введіть текст та значення зсуву для шифрування універсальним шифром Цезаря". Below this, there is a large text input field containing the word "HELLO". Underneath the input field, the text "Значення зсуву:" is followed by a smaller input field containing the number "3". A small blue tooltip below the shift field says "Працює з усіма символами Unicode, включаючи емодзі та різні мови". A prominent blue button labeled "Зашифрувати текст" is positioned below the inputs. Underneath the button, the text "Зашифрований текст:" is followed by a text area displaying the encrypted result "KHOOR". At the bottom of the interface, there are two more blue buttons: "Копіювати зашифрований текст" and "Завантажити зашифрований файл".

Рисунок 1.1 – Приклад шифрування універсальним методом Цезаря

Приклад розшифрування продемонстрований на рисунку 1.2.



The image shows a web application interface for decrypting text using the Caesar cipher. The interface is dark-themed with blue buttons and text. It includes a text input field for the encrypted text, a numeric input field for the shift value, and a button to perform the decryption. Below the button, there is a text area showing the decrypted result, and two more buttons: one to copy the result and another to download it.

Вставте зашифрований текст для дешифрування

KHOOR

Значення зсуву:

3

Дешифрувати текст

Результат дешифрування:

HELLO

Копіювати результат

Завантажити результат

Рисунок 1.2 – Приклад розшифрування універсальним методом Цезаря

AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) — це сучасний і широко використовуваний симетричний блоковий шифр, який застосовується для захисту даних в багатьох сферах, від банківських операцій до захисту персональної інформації. Симетричне шифрування означає, що для шифрування та розшифрування використовується один і той самий ключ. Це робить процес шифрування швидким і ефективним, але важливим є збереження конфіденційності ключа.

Принцип роботи:

AES працює на принципі блочного шифрування. Дані шифруються не поодинокими символами, а цілими блоками, розмір яких становить 128 біт (або 16 байт). Ключі для шифрування в AES можуть мати різну довжину: 128 біт, 192 біт або 256 біт. Чим довший ключ, тим складніший і, відповідно, надійніший захист. У процесі шифрування AES використовує кілька раундів математичних операцій, таких як заміни, перестановки та шифрування під ключем, щоб перетворити початкові дані в зашифрований текст.

Один з головних аспектів AES — це його ефективність. Він здатний працювати швидко навіть при великих обсягах даних, завдяки чому став стандартом для багатьох застосунків, що потребують високого рівня захисту.

Як користуватися:

1. Введіть текст для шифрування або розшифрування: Це перший крок, в якому ви вводите текст, що потребує шифрування або який потрібно розшифрувати. Текст може бути будь-яким: від простих рядків до складних повідомлень.

2. Введіть ключ (пароль), який буде використано для шифрування: Ключ є дуже важливим елементом для безпеки вашого шифрування. Ви повинні ввести пароль або ключ, що буде використовуватися для створення зашифрованого тексту. Ключ має бути надійним, тому рекомендується використовувати комбінацію великих і малих літер, цифр та спеціальних символів. Це забезпечить стійкість шифрування до можливих атак.

3. Натисніть відповідну кнопку для шифрування або розшифрування: Після введення тексту та ключа, ви можете натискати одну з двох кнопок:

- **Зашифрувати:** Якщо ви хочете зашифрувати ваш текст за допомогою AES, натискаєте цю кнопку. Система застосує алгоритм AES для шифрування тексту з використанням введеного ключа.

- **Дешифрувати:** Якщо у вас є зашифрований текст і ви хочете повернути його до оригінального вигляду, потрібно натискати кнопку "Дешифрувати" і ввести той самий ключ, який використовувався при шифруванні.

Примітка:

Для забезпечення надійності вашого шифрування важливо використовувати сильний ключ. Слабкі ключі, такі як прості паролі (наприклад, "12345" або "password"), легко піддаються зламуванню за допомогою брутфорс-атак. Щоб цього уникнути, створюйте складні ключі, що включають в себе різні типи символів, і обов'язково зберігайте їх у безпечному місці.

Приклад шифрування:

- Вхідний текст: "HelloWorld"
- Ключ: "MySecretKey123"

Коли цей текст зашифровується за допомогою AES з вказаним ключем, результат буде виглядати як набір зашифрованих символів, що неможливо розшифрувати без правильного ключа.

AES є основою для захисту даних в багатьох сучасних системах, таких як VPN, Wi-Fi захист, онлайн-банкінг та багато інших сервісів, де конфіденційність і безпека є критичними.

Приклад шифрування повідомлення методом AES-256 продемонстровано на рисунку 2.1.

Введіть текст і пароль для шифрування з AES-256

HelloWorld

Пароль:

.....

Підтримує шифрування тексту за допомогою AES-256 в режимі CBC

Зашифрувати текст

Зашифрований текст (Base64):

htG5TNT1IFb1XyzFho9gS7SdhajWhwgzzEWr9/ifpw=

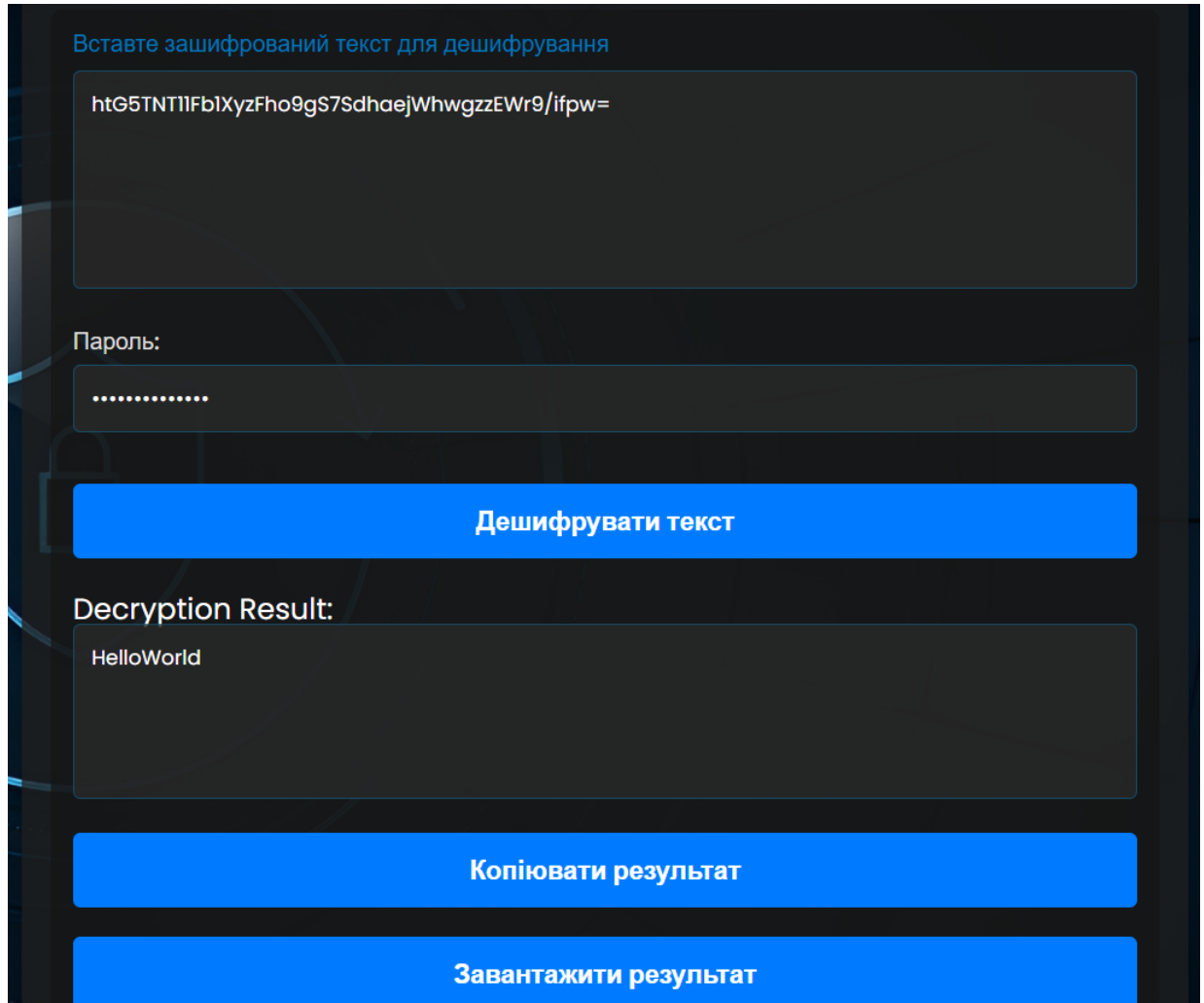
Копіювати зашифрований текст

Завантажити зашифрований файл

Рисунок 2.1 – Приклад шифрування методом AES-256

Приклад розшифрування продемонстрований на рисунку 2.2.

Тепер ми беремо наш зашифрований текст і ключ, вставляємо у відповідні поля на сайті і отримуємо розшифрований результат.



The image shows a web interface for AES-256 decryption. It features a dark background with blue text and buttons. The interface includes a text input field for the encrypted text, a password input field, a 'Decrypt text' button, a 'Decryption Result' section showing the decrypted text, and two buttons for handling the result: 'Copy result' and 'Download result'.

Вставте зашифрований текст для дешифрування

htG5TNTlIFbIXyzFho9gS7SdhaejWhwgzzEW9/ifpw=

Пароль:

.....

Дешифрувати текст

Decryption Result:

HelloWorld

Копіювати результат

Завантажити результат

Рисунок 2.2 – Приклад розшифрування методом AES-256

RSA (Rivest–Shamir–Adleman)

Принцип роботи: RSA — це асиметричний криптографічний алгоритм, який використовує пару ключів: відкритий для шифрування та закритий для розшифрування.

Як користуватися:

1. Згенеруйте пару ключів (відкритий та закритий) натиснувши на відповідні кнопки.
2. Для шифрування введіть текст та використайте відкритий ключ.
3. Для розшифрування введіть зашифрований текст та використайте закритий ключ.

Примітка: Зберігайте закритий ключ у безпечному місці та не передавайте його іншим особам.

Приклад генерування публічного і приватного ключів продемонстровано на рисунках 3.1 та 3.2. Коли ви працюєте з криптографією, особливо з системами шифрування, однією з важливих складових є генерація пари ключів: публічного та приватного. Щоб почати, вам потрібно вибрати розмір вашого ключа. Це може бути, наприклад, 1024 біти, 2048 біт або 4096 біт, залежно від бажаної безпеки вашого шифрування. Чим більший розмір ключа, тим складніше його буде зламати, але це також потребує більше ресурсів для обробки.

Після того як ви вибрали розмір ключа, система починає процес генерації вашої пари ключів. Це займає деякий час, оскільки система генерує два ключі, один з яких буде публічним, а інший — приватним. Публічний ключ можна вільно поширювати серед інших користувачів. Він використовується для шифрування повідомлень, тобто коли хтось хоче надіслати вам зашифроване повідомлення, вони використовують ваш публічний ключ.

Після того як повідомлення зашифроване, тільки ви, маючи ваш приватний ключ, зможете його розшифрувати. Ваш приватний ключ є секретним і ніколи не повинен покидати ваш комп'ютер або пристрій. Це забезпечує високий рівень

безпеки, оскільки навіть якщо хтось отримає доступ до публічного ключа, вони не зможуть зламати шифрування без доступу до вашого приватного ключа.

Усі ці кроки — від вибору розміру ключа до генерування пари і використання публічного та приватного ключа для шифрування та розшифрування — є основою сучасних систем безпечної комунікації в Інтернеті, таких як електронна пошта, онлайн-банкінг і багато інших застосунків, де конфіденційність інформації має першочергове значення.

Згенерувати ключі

Згенеруйте пару ключів для використання при шифруванні/дешифруванні

Розмір ключа: 2048 bit ▾ Більші ключі безпечніші, але повільніші

1024 bit

2048 bit

4096 bit

Згенерувати пару ключів

Публічний ключ:

```
NTkINDM0MjUwOTAwMTk0MjM2MDcxODgyMTQ2Nzc4MjkyNTcwMTAxNDgxOTI5ODkx
NDgIMjg5MzUwNTA3MDA4MDI2MTMyNDM4NDc2OTUwNTYwNjA0NTk0ODQ4MTMwMMDQ5
NzAwMjAwMzgwNDk1MDEwMTUwNTc4ODY1Njg0NjgzNzAyMTIIMzM5NTAzMjY0NTki
LCJlIjo1NjUIMzcfQ==
-----END RSA PUBLIC KEY-----
```

Копіювати публічний ключ

Завантажити публічний ключ

Приватний ключ:

Рисунок 3.1 – Генерування публічного ключа

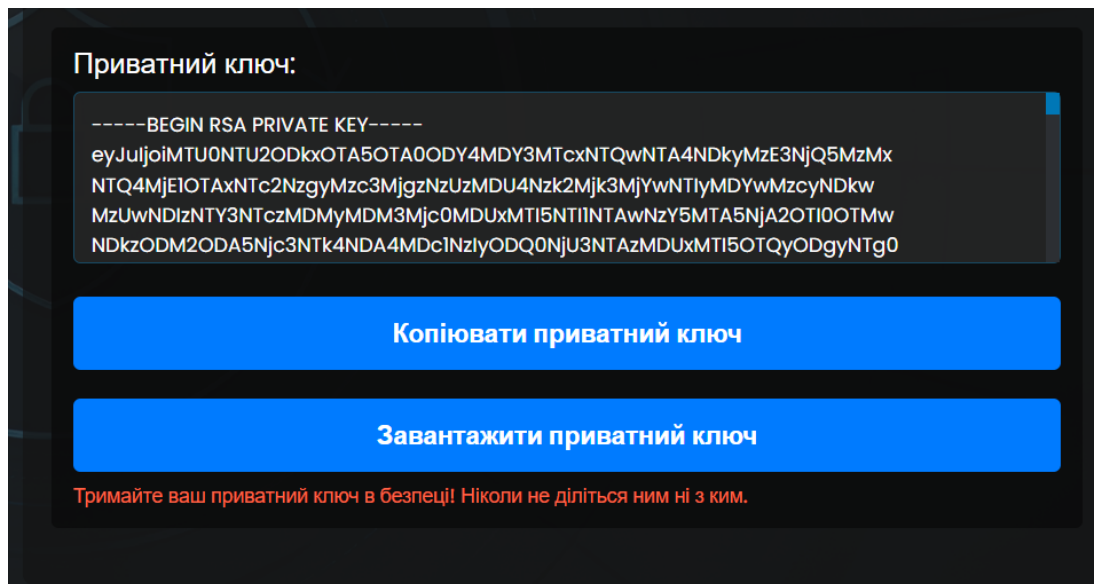


Рисунок 3.2 – Генерування приватного ключа

Приклад розшифрування повідомлення продемонстровано на рисунках 3.3 та 3.4.

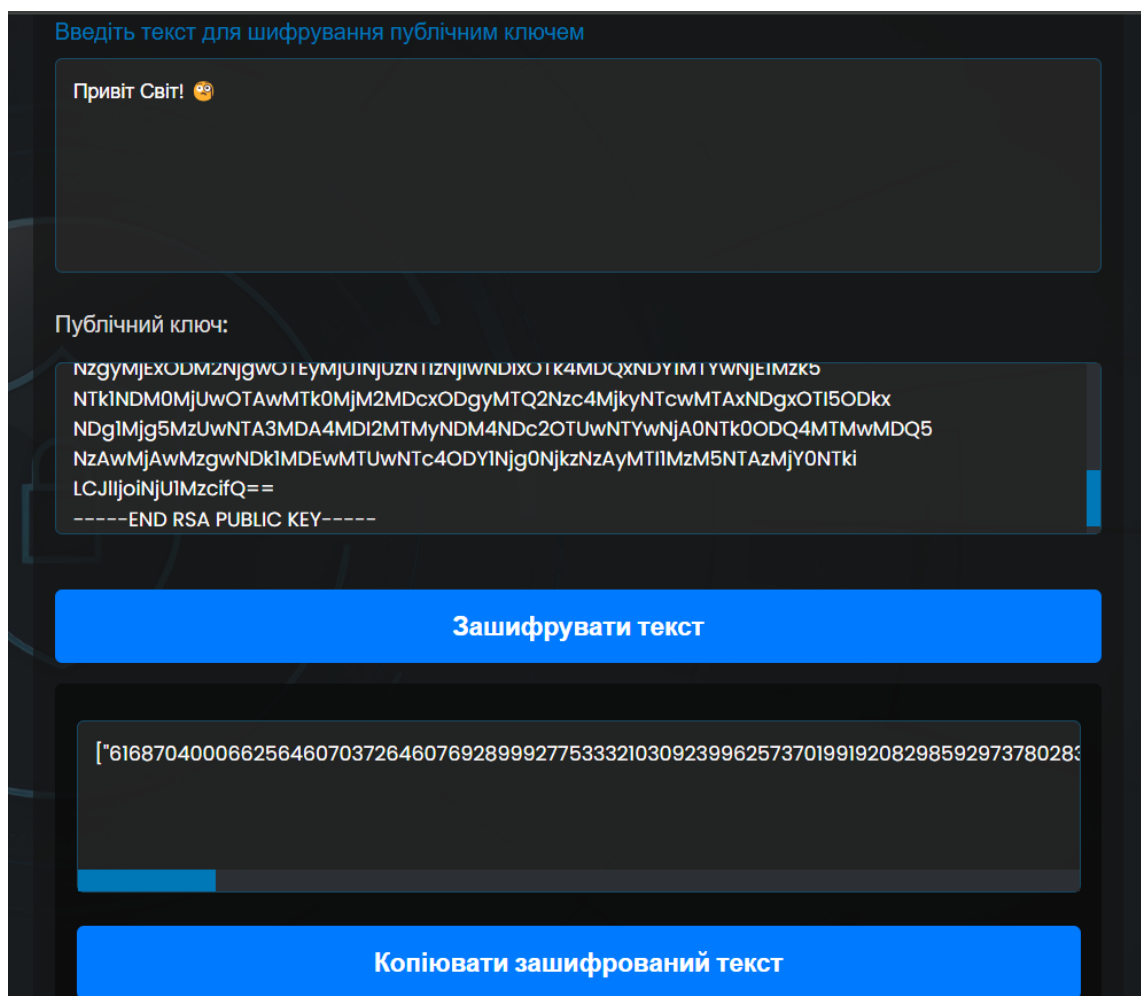


Рисунок 3.3 – Приклад шифрування за допомогою публічного ключа

Введіть зашифрований текст для дешифрування приватним ключем

```
32572629585420147199343484724008910146966414548943682319315708257430385682617961446611
28541954714806572358983660655122759807997372577769510746295308555884970469754598212
82667735252470206210988043066700691543412482932054457730533851838514853983474299222
2650537492210288055277721479041640284830464924824295601659899971907620649183117782227
4643517515364044073689070053718870384251925890095624699930153681319227702454831815486
24128970447240646516418576025832659254657697764010556046672119968117397043833311212288
49512132446249860347568888"]
```

Приватний ключ:

```
NZQ3MJY5MZYNZMXMTQXNDU3OTE4NZE4ODK3OTEWNJWUNDY4MTY4NDG4NDC3NIGI
NDUwMDI0MDg2Mzk4MzMzODgxMTEwNTMINDA2NzU5NTY4NzA5OTYwNDA2MjU0NDY0
MTAzMjY3MDExNTI2NjgxODc1MTU4OTE4Mjg1Mzc3OTc0MzM2MjI1MjM5MjM4MTQ1
NTk0OTkzNDI2MTMwMzA4MTY2ODY2Nzg3OTg4NDI1NDExMjU4MTY5Mzc2OTU1MTky
MDM2ODk5MyJ9
-----END RSA PRIVATE KEY-----
```

Дешифрувати текст

Результат дешифрування:

Привіт Світ! 🤖

Рисунок 3.4 – Приклад розшифрування приватним ключем

OTP (One-Time Pad)

OTP (One-Time Pad) — це один з найстаріших і водночас найнадійніших методів шифрування, який забезпечує абсолютну безпеку при правильному використанні. Головною особливістю цього методу є те, що для шифрування використовується одноразовий ключ, який має таку саму довжину, як і текст повідомлення. Після використання ключу для шифрування або розшифрування, він більше не застосовується, що робить його абсолютно непередбачуваним і не вразливим до атак.

Принцип роботи:

OTP працює на основі дуже простого, але ефективного принципу. Кожен символ в оригінальному повідомленні поєднується з відповідним символом з одноразового ключа за допомогою математичних операцій (наприклад, побітового додавання або XOR). Оскільки ключ має таку ж довжину, як і саме повідомлення, і використовується тільки один раз, шифрування є абсолютно безпечним, якщо ключ є випадковим і достатньо складним.

Цей метод шифрування не піддається розшифруванню без наявності ключа, оскільки кожен можливий ключ, що може бути використаний для розшифровки, буде математично непередбачуваним. Тому OTP є теоретично неушкодженим для атак, якщо дотримано правил його використання.

Як користуватися:

1. Введіть текст для шифрування або розшифрування: Спершу ви вводите текст, який потрібно зашифрувати або розшифрувати. Це може бути будь-яке повідомлення, яке ви хочете захистити.

2. Натисніть кнопку "Зашифрувати текст": Коли ви натискаєте цю кнопку, система автоматично генерує одноразовий ключ довжиною, що дорівнює довжині введеного тексту. Ключ повинен бути випадковим і не піддаватися прогнозуванню. Важливо, щоб кожен ключ був унікальним і використовувався тільки для одного повідомлення.

3. Натисніть кнопку для шифрування або розшифрування: Після того, як ви отримали ключ, можна натискати одну з двох кнопок:

- **Зашифрувати:** Це дозволяє створити зашифровану версію вашого повідомлення, використовуючи згенерований ключ.

- **Дешифрувати:** Якщо у вас є зашифроване повідомлення та відповідний одноразовий ключ, ви можете натискати "Дешифрувати", щоб відновити оригінальний текст.

Примітка:

- **Випадковість ключа:** Ключ, який використовується для ОТР, має бути повністю випадковим і не повторюватися для різних повідомлень. Це дуже важливе правило, яке забезпечує безпеку шифрування.

- **Одноразовість ключа:** Ключ може бути використаний лише один раз. Після того, як ви зашифруєте або розшифруєте повідомлення, цей ключ більше не може бути використаний для інших операцій.

ОТР є безумовно найнадійнішим методом шифрування з теоретичної точки зору, оскільки його безпека не залежить від складності алгоритму, а від абсолютної випадковості і одноразовості ключа. Однак для реального використання ОТР є певні обмеження, оскільки його впровадження на практиці потребує безпечного способу генерації та зберігання великих обсягів випадкових ключів, а також суворого контролю за їхнім використанням.

Приклад шифрування одноразовим паролем продемонстровано на рисунках 4.1 та 4.2.

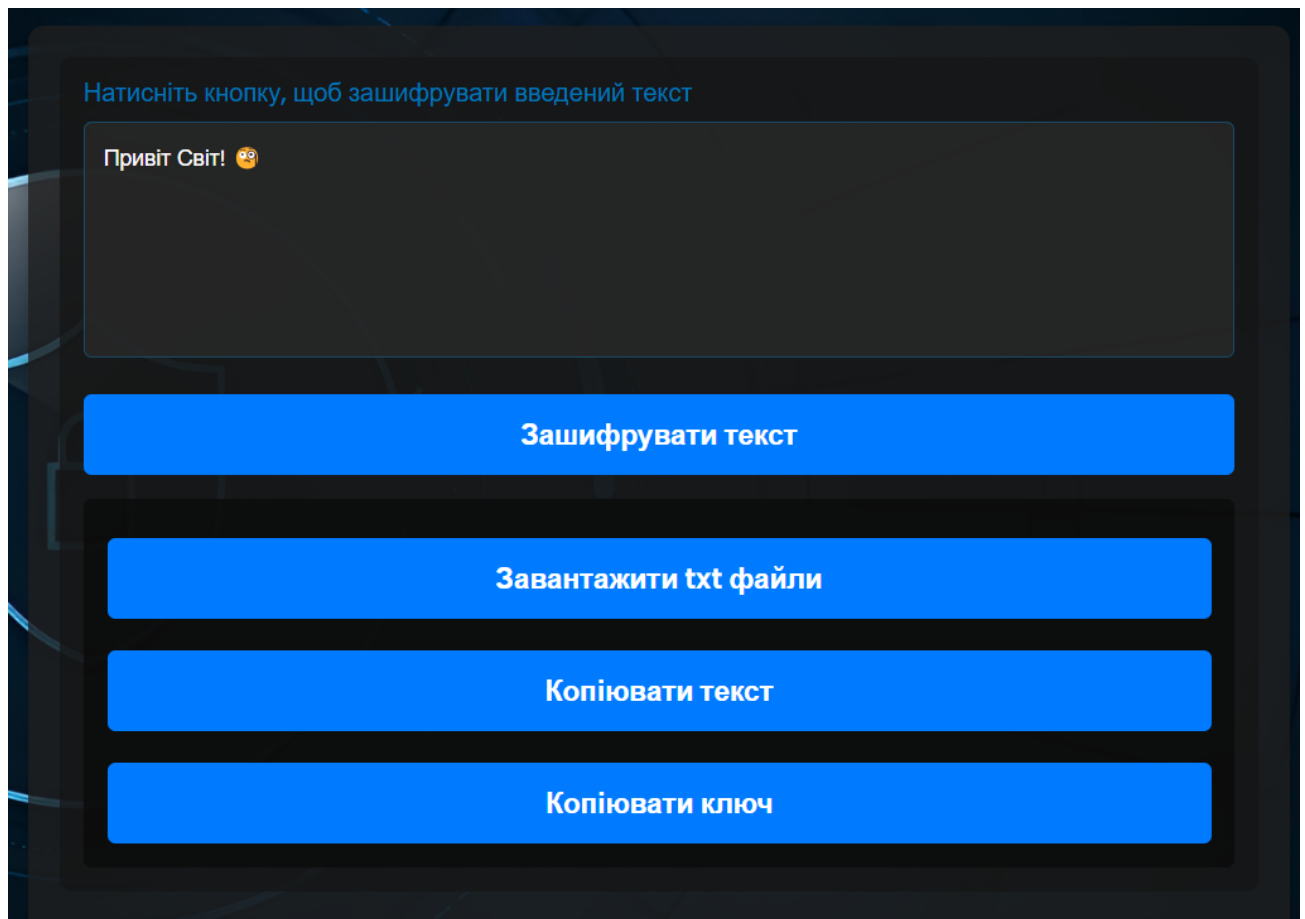


Рисунок 4.1 – Приклад шифрування тесту одноразовим паролем

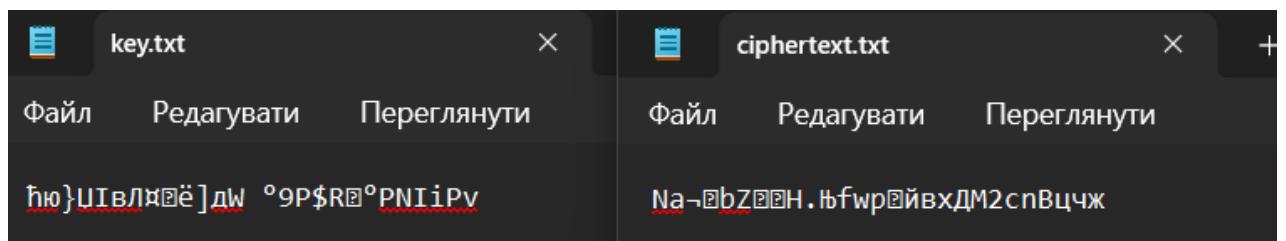


Рисунок 4.2 – Результат шифрування одноразовим паролем

Приклад розшифрування одноразовим паролем продемонстровано на рисунку 4.3.

Дешифрування

Вставте txt документ з інформацією, яку ви хочете дешифрувати:

Вибрати файл ciphertext.txt

Вставте txt документ з ключем дешифрування:

Вибрати файл key.txt

Дешифрувати

Дешифрування

Привіт Світ! 🤖

Копіювати текст

Завантажити txt файли

Рисунок 4.3 – Приклад розшифрування за допомогою ключа і шифротекста в txt файлах