Universal Caesar Cipher

Universal Caesar Cipher is a classic encryption method that belongs to symmetric methods, where the same key is used for both encryption and decryption. The basic principle of this cipher is that each letter in the text is shifted by a certain number of positions in the alphabet. This means, for example, if we shift the letter "A" one position forward, it becomes "B", and if we shift it by two positions, it becomes "C", and so on. The shift can be both to the left and to the right in the alphabet.

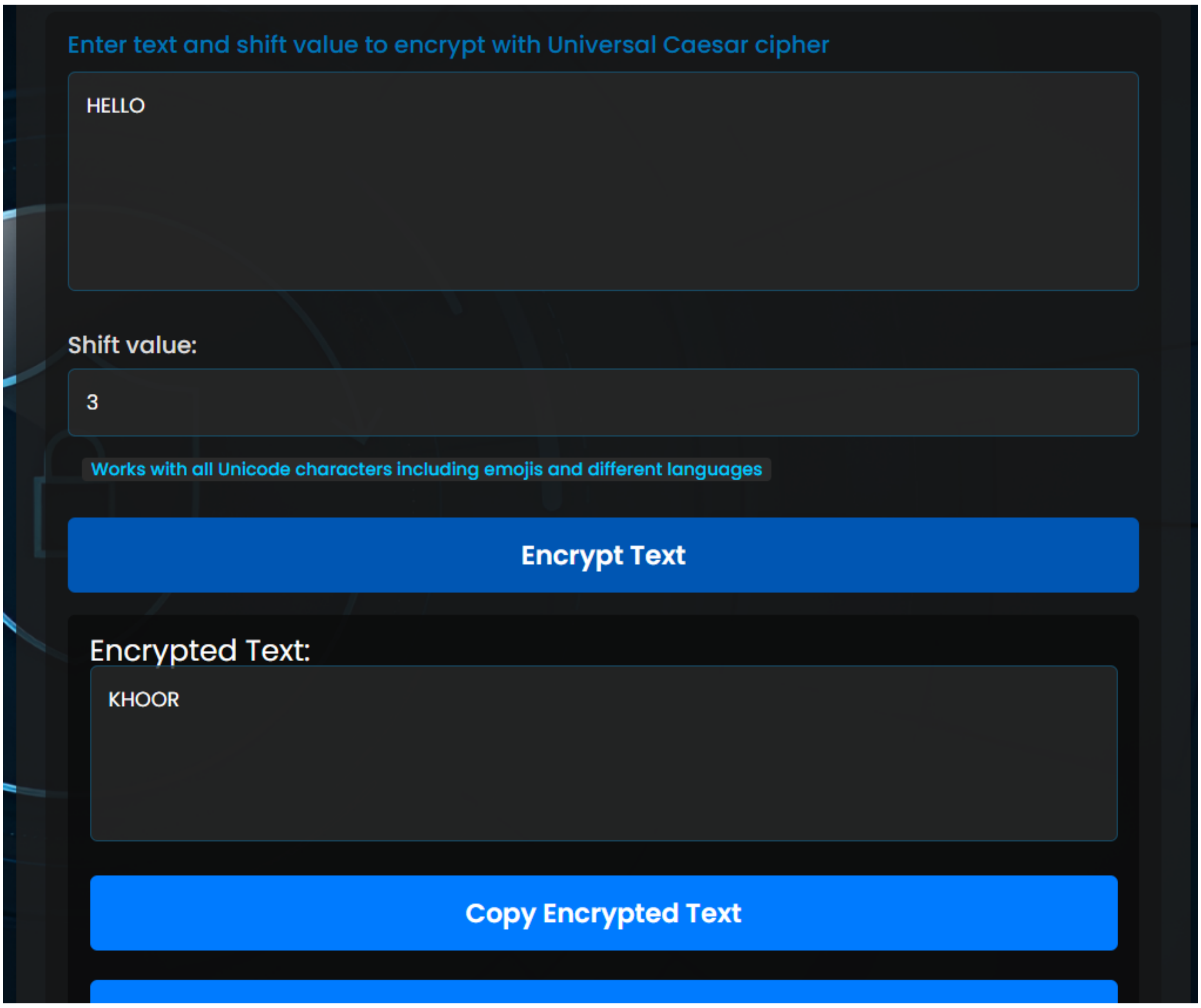
In the Universal Caesar Cipher, this method is extended to all Unicode characters, which means encryption can work not only with letters but also with other characters such as numbers, spaces, punctuation marks, and even special characters. This makes the cipher much more universal and suitable for encrypting various types of texts, regardless of their content.

How to use the Universal Caesar Cipher:

1. Enter text: First, you need to enter the text you want to encrypt or decrypt. This can be any text containing letters, numbers, special characters, or spaces.
2. Specify the key: The key for encryption in this method is a number that indicates how many positions each character in the entered text should be shifted. For example, if you choose a key of 3, this means each character will be shifted three positions forward in the Unicode table. The larger the key, the greater the shift of characters.
3. Click the "Encrypt" or "Decrypt" button: After entering the text and defining the key, you can click one of two buttons:
    - Encrypt: This will create an encrypted text where each character will be shifted by the specified number of positions.
    - Decrypt: If you have encrypted text and want to return it to its original form, you need to specify the same key and click "Decrypt". This will return the text to its original form, as the encryption and decryption process is the same — just a reverse shift.

The Universal Caesar Cipher is a simple but effective method for learning the basics of cryptography. Although this method is not very reliable for protecting confidential data, it is still widely used in various cryptographic educational programs and for basic encryption needs.

An example of encryption is shown in Figure 1.1.



Enter text and shift value to encrypt with Universal Caesar cipher

HELLO

Shift value:

3

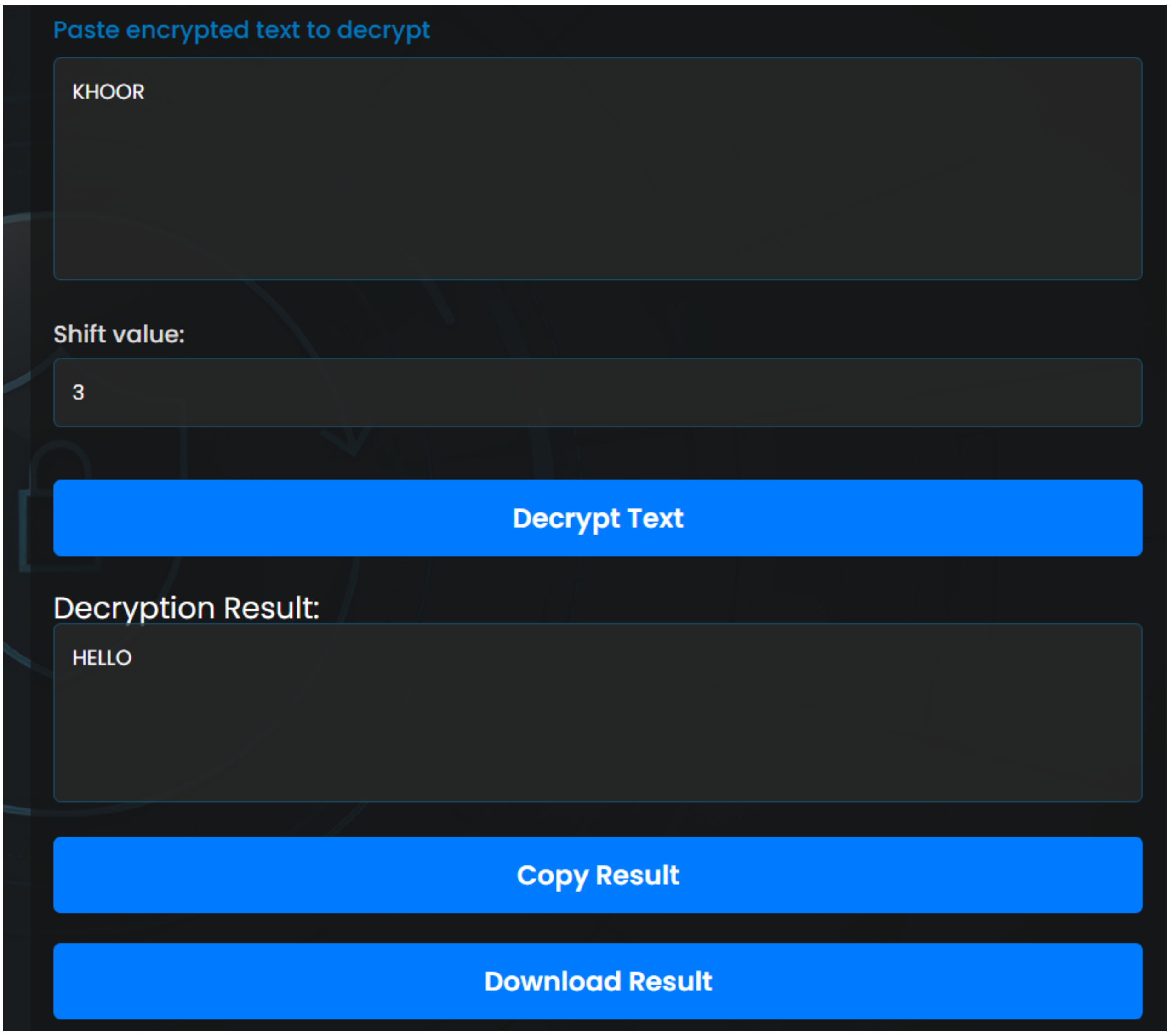
Works with all Unicode characters including emojis and different languages

Encrypted Text:

KHOOR

Figure 1.1 – Example of encryption with the Universal Caesar method

An example of decryption is shown in Figure 1.2.


Paste encrypted text to decrypt

KHOOR

Shift value:

3

Decryption Result:

HELLO

Figure 1.2 – Example of decryption with the Universal Caesar method

AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) is a modern and widely used symmetric block cipher that is applied to protect data in many areas, from banking operations to personal information protection. Symmetric encryption means that the same key is used for both encryption and decryption. This makes the encryption process fast and efficient, but it is important to maintain the confidentiality of the key.

Operating Principle:

AES works on the principle of block encryption. Data is encrypted not by individual characters, but by entire blocks with a size of 128 bits (or 16 bytes). Keys for encryption in AES can have different lengths: 128 bits, 192 bits, or 256 bits. The longer the key, the more complex and, accordingly, more reliable the protection. In the encryption process, AES uses several rounds of mathematical operations, such as substitutions, permutations, and encryption under the key, to transform the original data into encrypted text.

One of the main aspects of AES is its efficiency. It can work quickly even with large amounts of data, which is why it has become a standard for many applications that require a high level of protection.

How to use:

1. Enter text for encryption or decryption: This is the first step where you enter the text that needs encryption or needs to be decrypted. The text can be anything: from simple strings to complex messages.
2. Enter the key (password) that will be used for encryption: The key is a very important element for the security of your encryption. You must enter a password or key that will be used to create the encrypted text. The key should be reliable, so it is recommended to use a combination of uppercase and lowercase letters, numbers, and special characters. This will ensure the resistance of encryption to possible attacks.
3. Click the appropriate button for encryption or decryption: After entering the text and key, you can click one of two buttons:
    - Encrypt: If you want to encrypt your text using AES, click this button. The system will apply the AES algorithm to encrypt the text using the entered key.

    - Decrypt: If you have encrypted text and you want to return it to its original form, you need to click the "Decrypt" button and enter the same key that was used during encryption.

Note:

For ensuring the reliability of your encryption, it is important to use a strong key. Weak keys, such as simple passwords (for example, "12345" or "password"), are easily broken with brute force attacks. To avoid this, create complex keys that include different types of characters, and be sure to store them in a safe place.

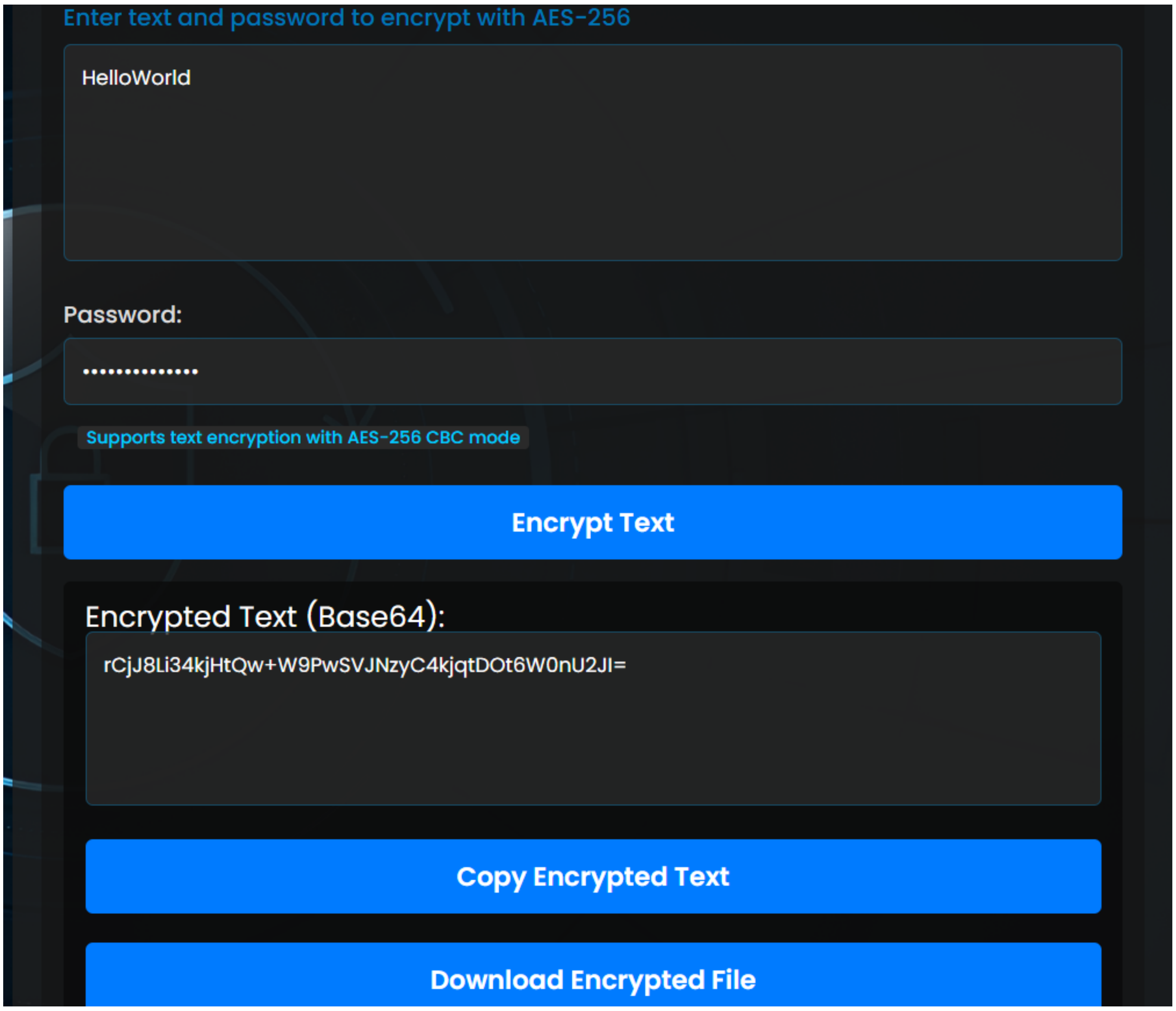
Example of encryption:

- Input text: "HelloWorld"
- Key: "MySecretKey123"

When this text is encrypted using AES with the specified key, the result will look like a set of encrypted characters that cannot be decrypted without the correct key.

AES is the foundation for data protection in many modern systems such as VPN, Wi-Fi protection, online banking, and many other services where confidentiality and security are critical.

An example of message encryption using the AES-256 method is shown in Figure 2.1.


Enter text and password to encrypt with AES-256

HelloWorld

Password:

••••••••••••••

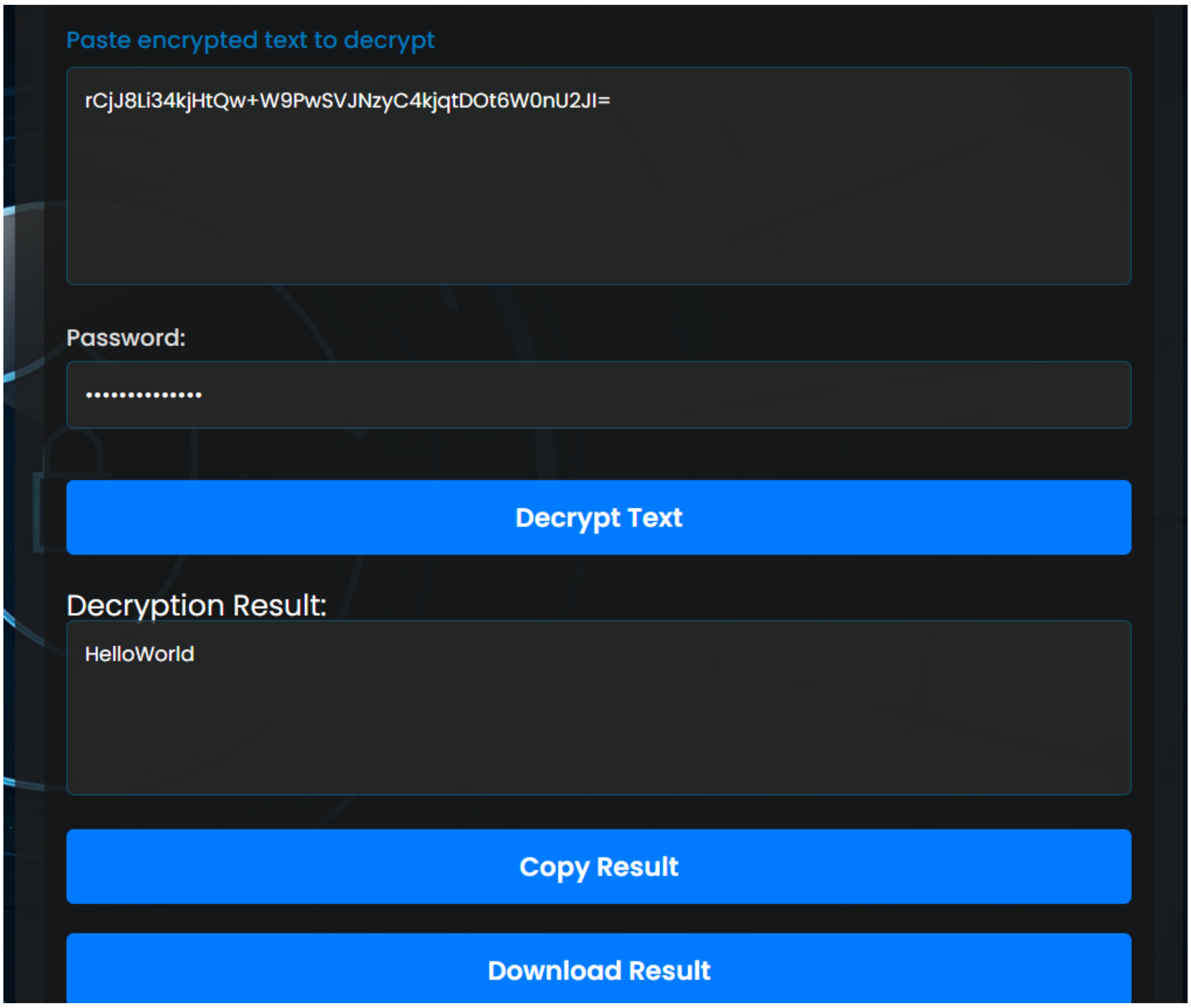
Encrypted Text (Base64):

```
rCjJ8Li34kjHtQw+W9PwSVJNzyC4kjqtDOt6W0nU2JI=
```

Figure 2.1 – Example of encryption using the AES-256 method

An example of decryption is shown in Figure 2.2.

Now we take our encrypted text and key, insert them into the corresponding fields on the site, and get the decrypted result.


Paste encrypted text to decrypt

```
rCjJ8Li34kjHtQw+W9PwSVJNzyC4kjqtDOt6W0nU2JI=
```

Password:

••••••••••••••

Decryption Result:

```
HelloWorld
```

Figure 2.2 – Example of decryption using the AES-256 method

RSA (Rivest–Shamir–Adleman)

Operating Principle:

RSA is an asymmetric cryptographic algorithm that uses a pair of keys: a public key for encryption and a private key for decryption.

How to use:

1. Generate a pair of keys (public and private) by clicking on the corresponding buttons.
2. For encryption, enter the text and use the public key.
3. For decryption, enter the encrypted text and use the private key.

Note:

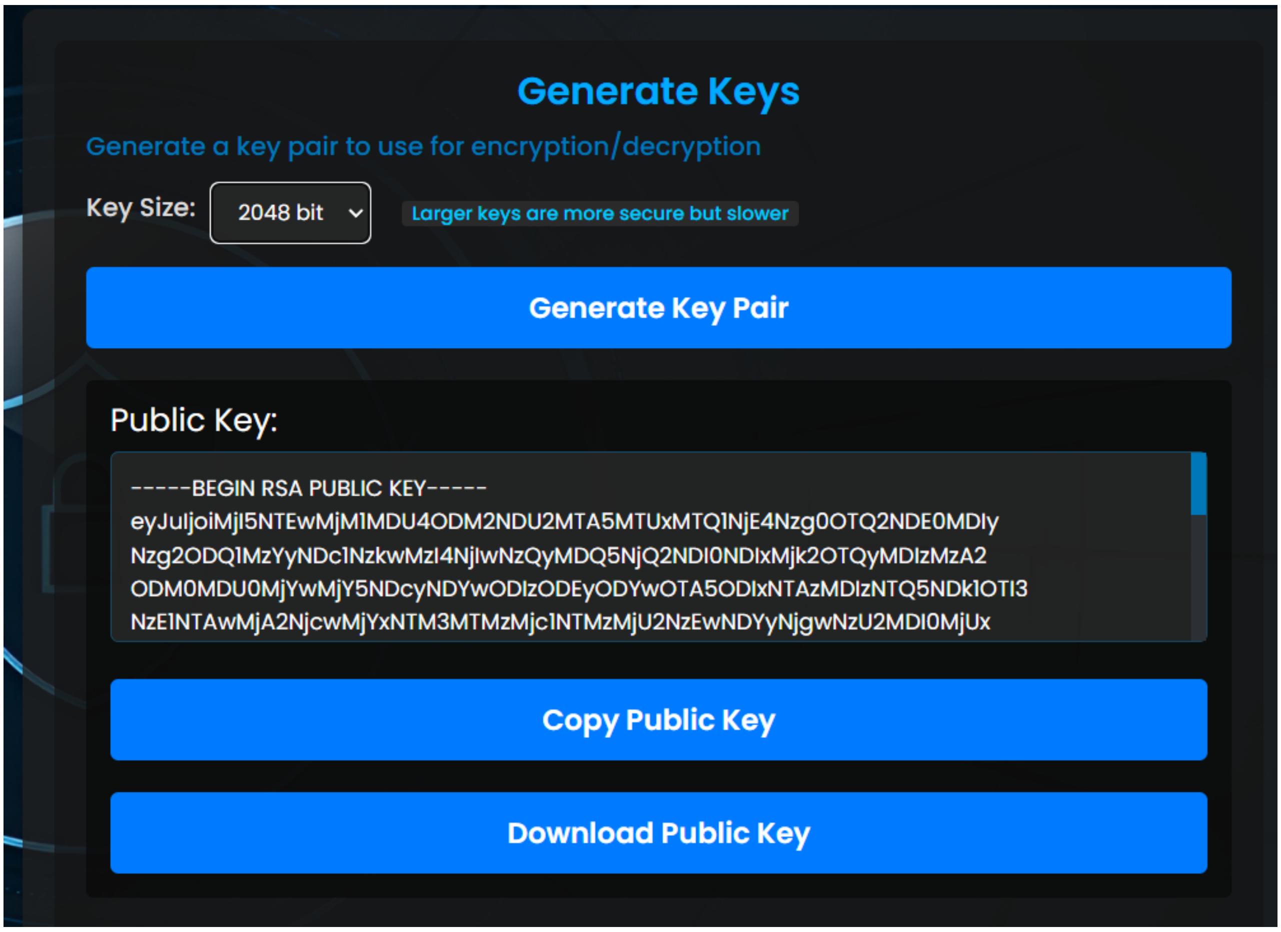
Keep the private key in a secure place and do not share it with other people.

An example of generating public and private keys is shown in Figures 3.1 and 3.2. When you work with cryptography, especially with encryption systems, one of the important components is the generation of a pair of keys: public and private. To start, you need to choose the size of your key. This can be, for example, 1024 bits, 2048 bits, or 4096 bits, depending on the desired security of your encryption. The larger the key size, the more difficult it will be to break, but it also requires more resources for processing.

After you have chosen the key size, the system begins the process of generating your pair of keys. This takes some time because the system generates two keys, one of which will be public and the other private. The public key can be freely distributed among other users. It is used to encrypt messages, meaning when someone wants to send you an encrypted message, they use your public key.

After the message is encrypted, only you, having your private key, will be able to decrypt it. Your private key is secret and should never leave your computer or device. This provides a high level of security because even if someone gains access to the public key, they will not be able to break the encryption without access to your private key.

All these steps — from choosing the key size to generating the pair and using the public and private key for encryption and decryption — are the foundation of modern secure communication systems on the Internet, such as email, online banking, and many other applications where information confidentiality is of paramount importance.



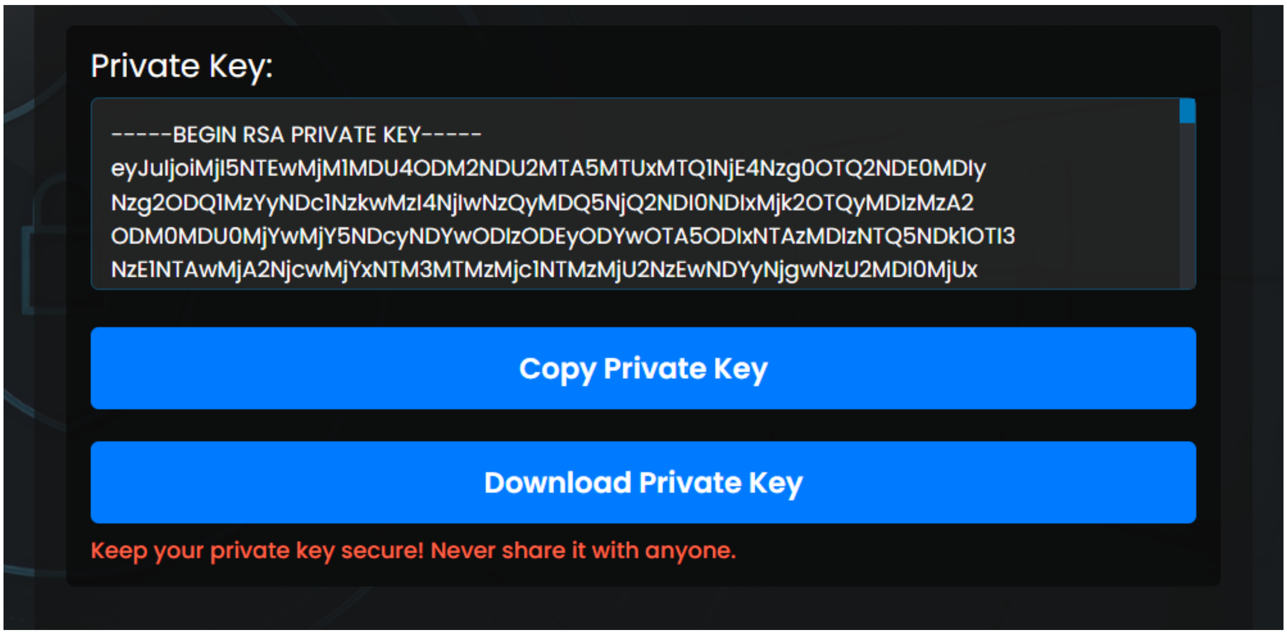
Key Size: 2048 bit

Larger keys are more secure but slower

Public Key:

```
-----BEGIN RSA PUBLIC KEY-----
eyJuIjoiMjI5NTEwMjM1MDU4ODM2NDU2MTA5MTUxMTQ1NjE4Nzg0OTQ2NDE0MDIy
Nzg2ODQ1MzYyNDc1NzkwMzI4NjIwNzQyMDQ5NjQ2NDI0NDIxMjk2OTQyMDIzMzA2
ODM0MDU0MjYwMjY5NDcyNDYwODIzODEyODYwOTA5ODIxNTAzMDIzNTQ5NDk1OTI3
NzE1NTAwMjA2NjcwMjYxNTM3MTMzMjc1NTMzMjU2NzEwNDYyNjgwNzU2MDI0MjUx
```

Figure 3.1 – Generating a public key



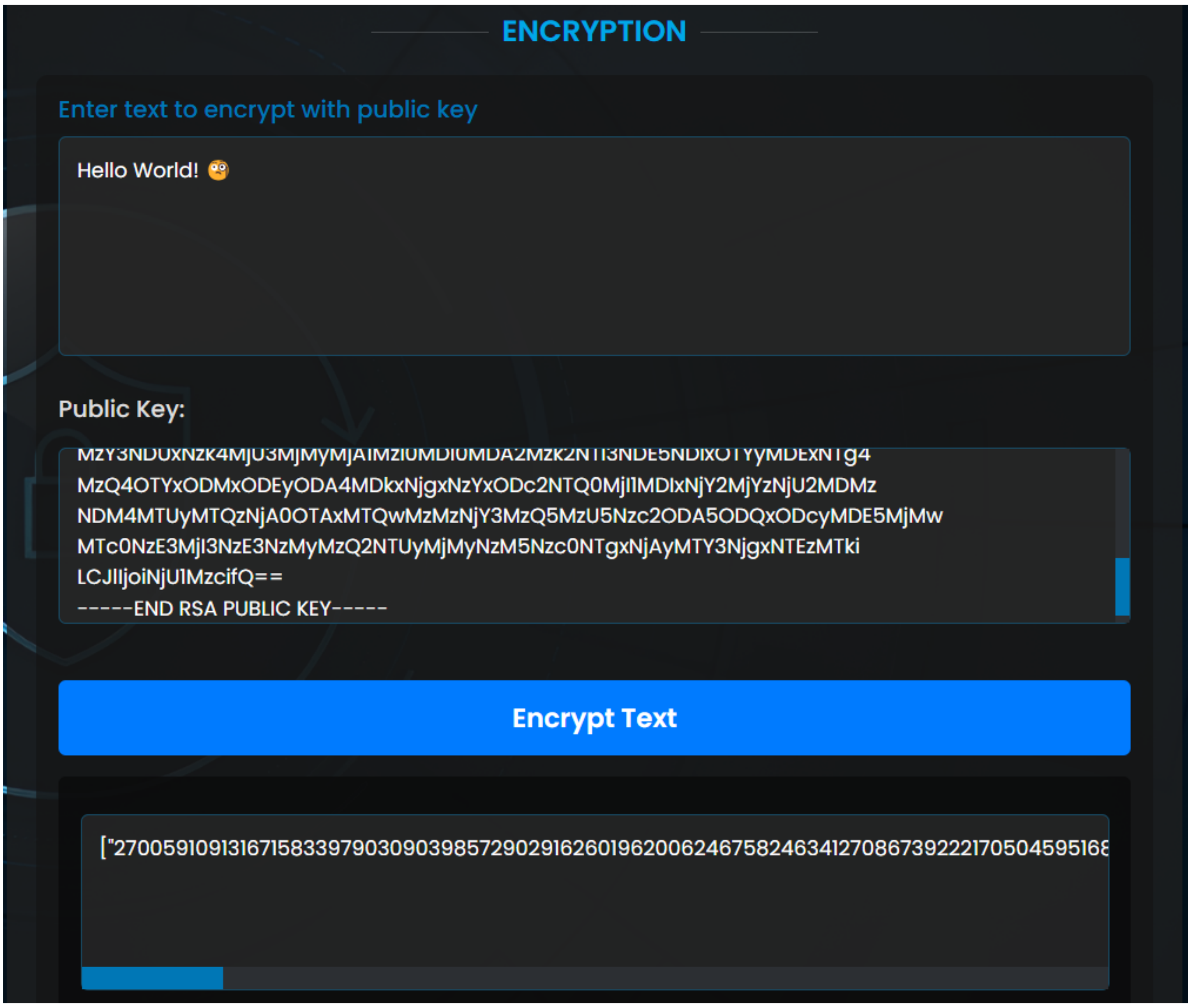
Private Key:

```
-----BEGIN RSA PRIVATE KEY-----
eyJuIjoiMjI5NTEwMjM1MDU4ODM2NDU2MTA5MTUxMTQ1NjE4Nzg0OTQ2NDE0MDIy
Nzg2ODQ1MzYyNDc1NzkwMzI4NjIwNzQyMDQ5NjQ2NDI0NDIxMjk2OTQyMDIzMzA2
ODM0MDU0MjYwMjY5NDcyNDYwODIzODEyODYwOTA5ODIxNTAzMDIzNTQ5NDk1OTI3
NzE1NTAwMjA2NjcwMjYxNTM3MTMzMjc1NTMzMjU2NzEwNDYyNjgwNzU2MDI0MjUx
```

Keep your private key secure! Never share it with anyone.

Figure 3.2 – Generating a private key

An example of message decryption is shown in Figures 3.3 and 3.4.


ENCRYPTION

Enter text to encrypt with public key

Hello World! 😳

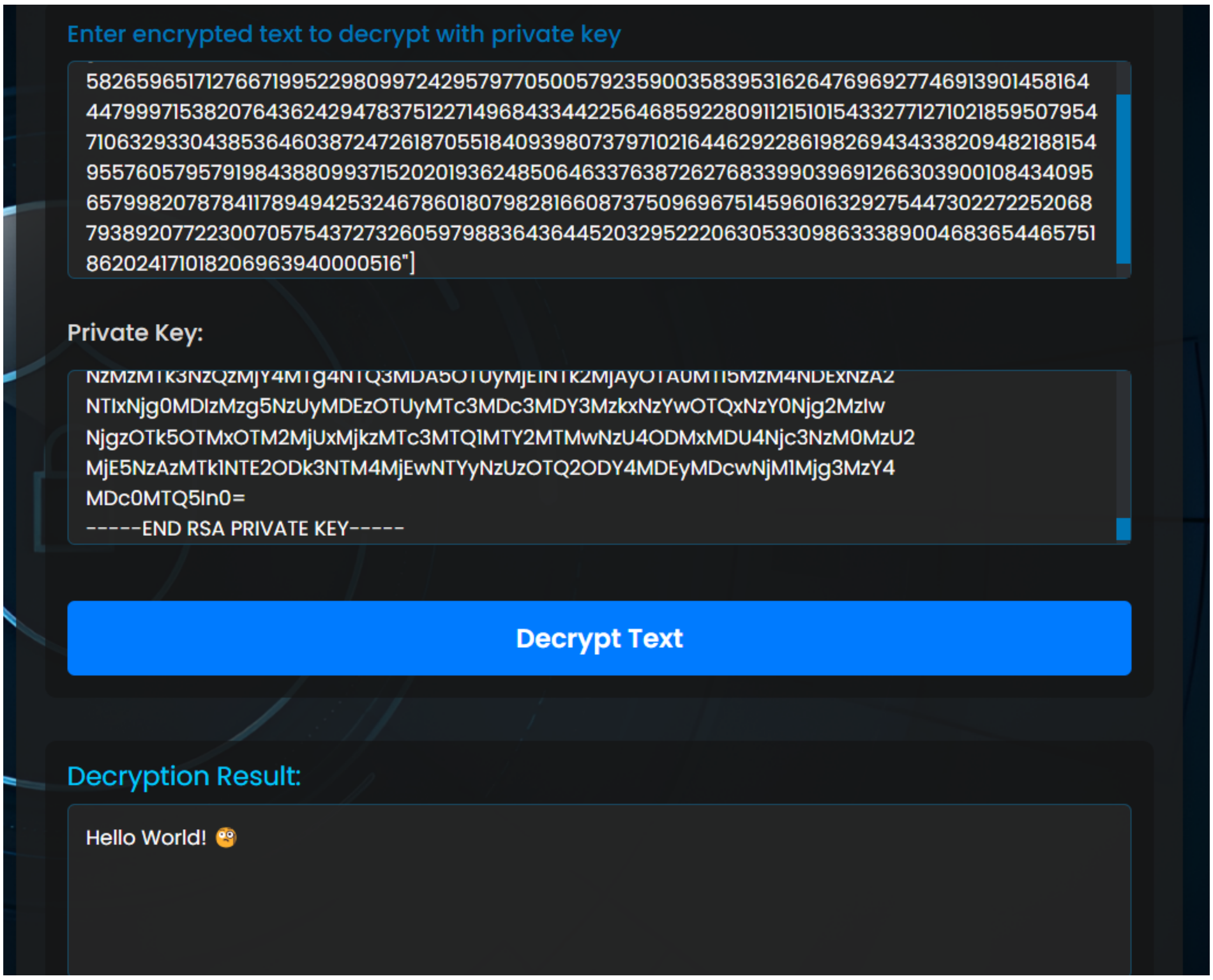
Public Key:

```
MzY3NDUxNzk4MjU3MjMyMjA1MzI0MDI0MDA2Mzk2NTI3NDE5NDIxOTYyMDExNTg4
MzQ4OTYxODMxODEyODA4MDkxNjgxNzYxODc2NTQ0MjI1MDIxNjY2MjYzNjU2MDMz
NDM4MTUyMTQzNjA0OTAxMTQwMzMzNjY3MzQ5MzU5Nzc2ODA5ODQxODcyMDE5MjMw
MTc0NzE3MjI3NzE3NzMyMzQ2NTUyMjMyNzM5Nzc0NTgxNjAyMTY3NjgxNTEzMTki
LCJlIjoiNjU1MzcifQ==
-----END RSA PUBLIC KEY-----
```

Encrypt Text

```
["27005910913167158339790309039857290291626019620062467582463412708673922217050459516
```

Figure 3.3 – Example of encryption using a public key


Enter encrypted text to decrypt with private key

```
5826596517127667199522980997242957977050057923590035839531626476969277469139014581644
4479997153820764362429478375122714968433442256468592280911215101543327712710218595079 54
7106329330438536460387247261870551840939807379710216446292286198269434338209482188154
9557605795791984388099371520201936248506463376387262768339903969126630390010843409 5
6579982078784117894942532467860180798281660873750969675145960163292754473022722520 68
7938920772230070575437273260597988364364452032952220630533098633389004683654465751
8620241710182069639400005 16"]
```

Private Key:

```
NzMzMTk3NzQzMjY4MTg4NTQ3MDA5OTUyMjE1NTk2MjAyOTA0MTI5MzM4NDExNzA2
NTIxNjg0MDIzMzg5NzUyMDEzOTUyMTc3MDc3MDY3MzkxNzYwOTQxNzY0Njg2MzIw
NjgzOTk5OTMxOTM2MjUxMjkzMTc3MTQ1MTY2MTMwNzU4ODMxMDU4Njc3NzM0MzU2
MjE5NzAzMTk1NTE2ODk3NTM4MjEwNTYyNzUzOTQ2ODY4MDEyMDcwNjM1Mjg3MzY4
MDc0MTQ5In0=
-----END RSA PRIVATE KEY-----
```

Decryption Result:

Hello World! 😳

Figure 3.4 – Example of decryption with a private key

OTP (One-Time Pad)

OTP (One-Time Pad) is one of the oldest and yet most reliable encryption methods, which provides absolute security when used correctly. The main feature of this method is that a one-time key is used for encryption, which has the same length as the message text. After using the key for encryption or decryption, it is no longer used, which makes it absolutely unpredictable and not vulnerable to attacks.

Operating Principle:

OTP works on a very simple but effective principle. Each character in the original message is combined with the corresponding character from the one-time key using mathematical operations (for example, bitwise addition or XOR). Since the key has the same length as the message itself and is used only once, the encryption is absolutely secure if the key is random and complex enough.

This encryption method cannot be decrypted without the key, as each possible key that can be used for decryption will be mathematically unpredictable. Therefore, OTP is theoretically invulnerable to attacks if the rules for its use are followed.

How to use:

1. Enter text for encryption or decryption: First, you enter the text that needs to be encrypted or decrypted. This can be any message you want to protect.
2. Click the "Encrypt text" button: When you click this button, the system automatically generates a one-time key of a length equal to the length of the entered text. The key must be random and not susceptible to prediction. It is important that each key is unique and used only for one message.
3. Click the button for encryption or decryption: After you have received the key, you can click one of two buttons:
   - Encrypt: This allows you to create an encrypted version of your message using the generated key.
   - Decrypt: If you have an encrypted message and the corresponding one-time key, you can click "Decrypt" to restore the original text.

Note:

- Randomness of the key: The key used for OTP must be completely random and not repeat for different messages. This is a very important rule that ensures the security of encryption.
- One-time use of the key: The key can be used only once. After you encrypt or decrypt a message, this key can no longer be used for other operations.

OTP is undoubtedly the most reliable encryption method from a theoretical point of view, as its security does not depend on the complexity of the algorithm, but on the absolute randomness and one-time use of the key. However, for real-world use, OTP has certain limitations, as its implementation in practice requires a secure way to generate and store large amounts of random keys, as well as strict control over their use.

An example of encryption with a one-time password is shown in Figures 4.1 and 4.2.

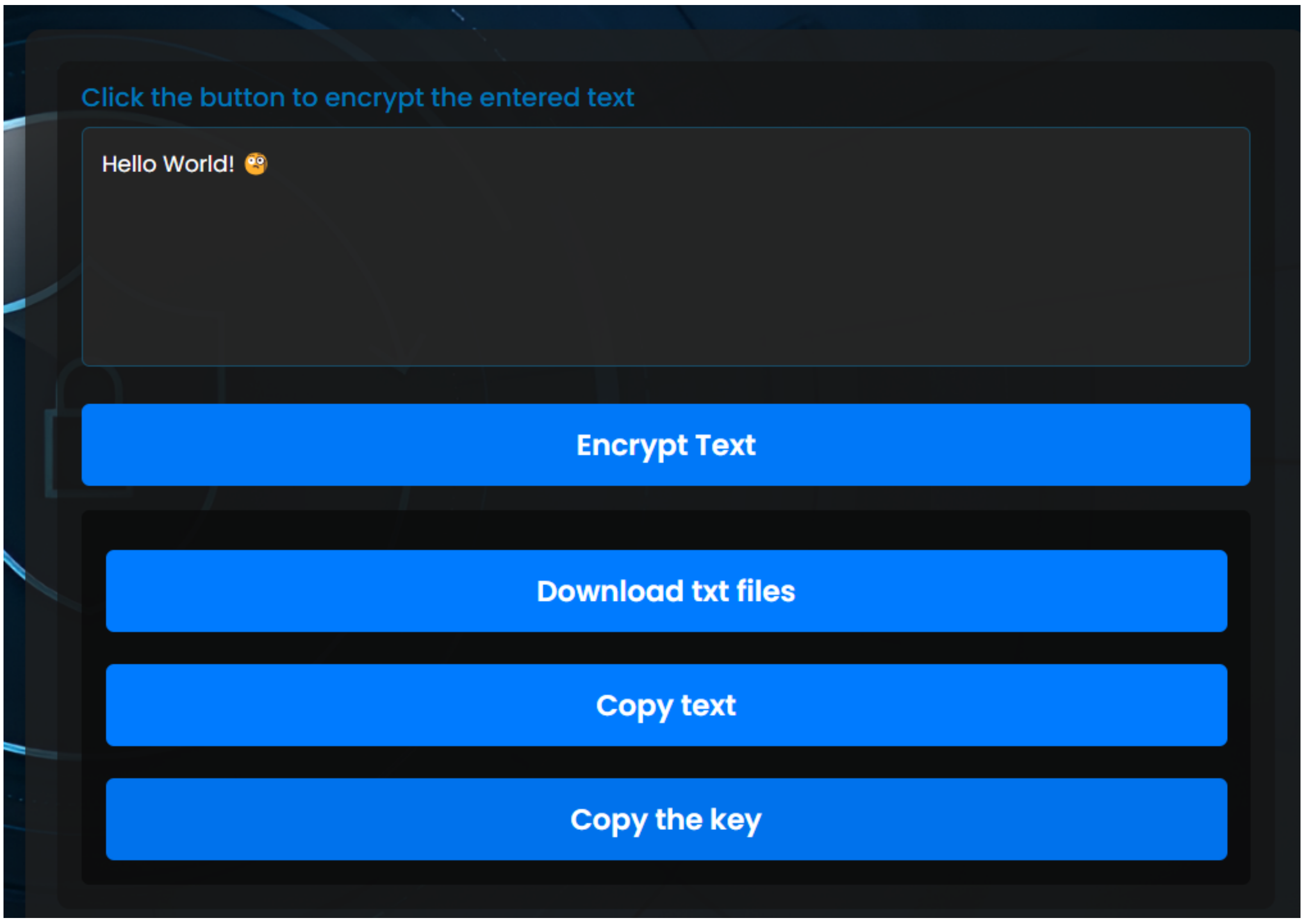
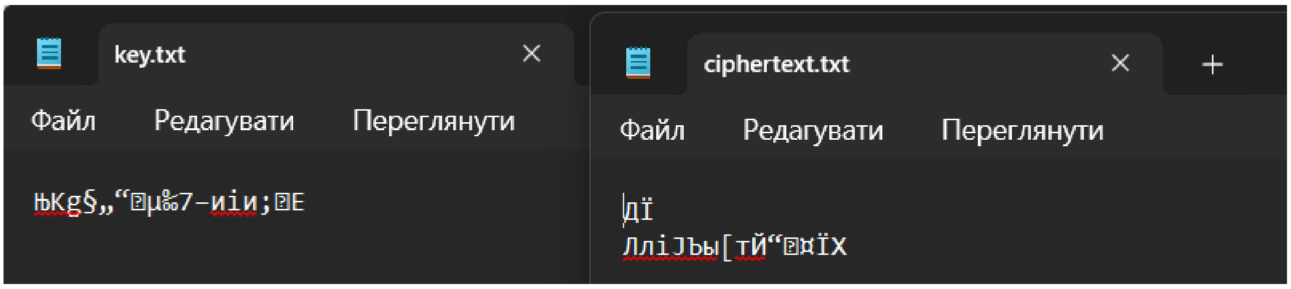


Figure 4.1 – Example of text encryption with a one-time password

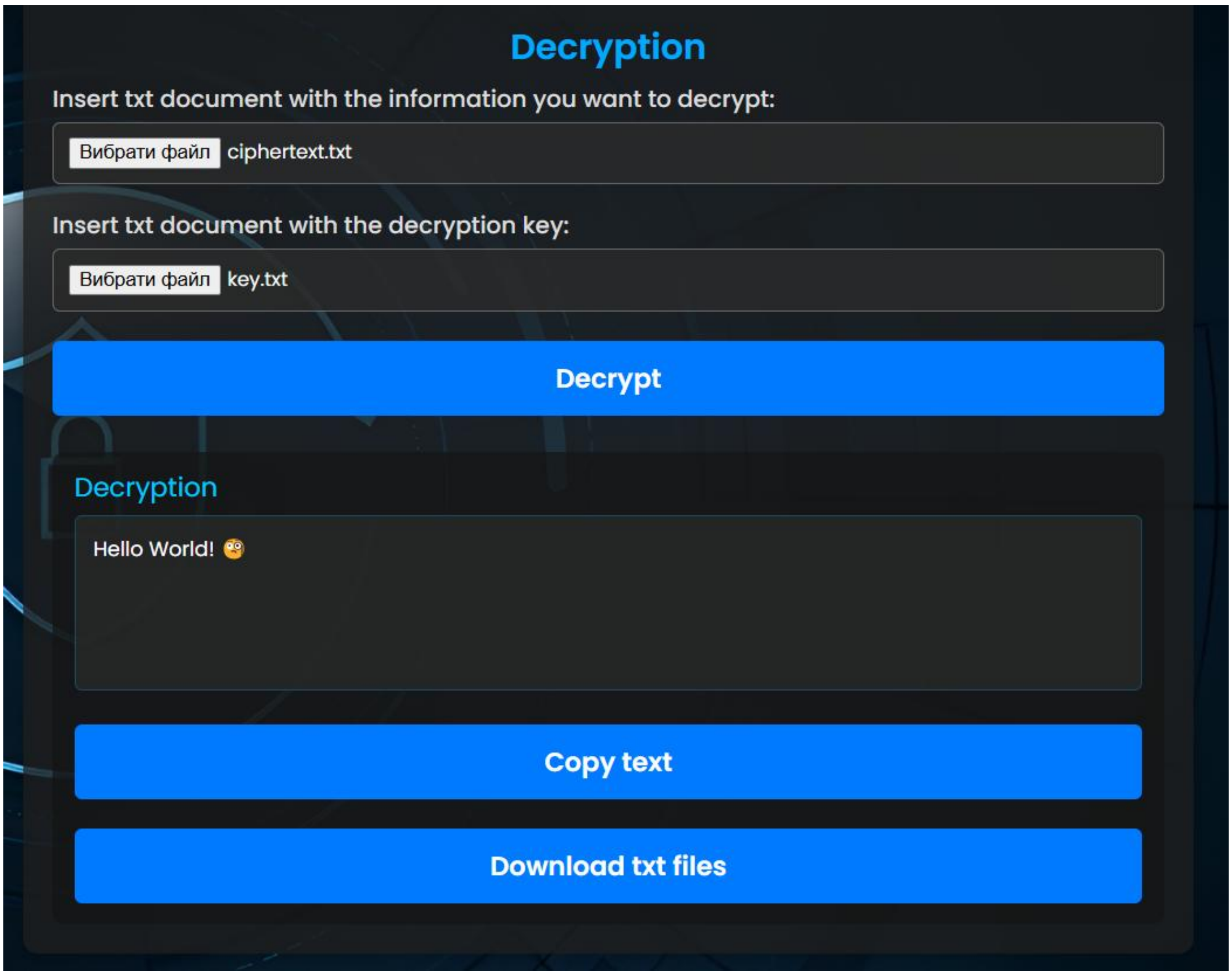


```
ђKg§„“⍰µ‰7–иіи;⍰E
```

```
дї
ЛліЈЪы[тЙ“⍰¤ЇХ
```

Figure 4.2 – Result of encryption with a one-time password

An example of decryption with a one-time password is shown in Figure 4.3.


Decryption
Insert txt document with the information you want to decrypt:
Вибрати файл
ciphertext.txt
Insert txt document with the decryption key:
Вибрати файл
key.txt
Decrypt
Decryption
Copy text
Download txt files

Hello World! 😳

Figure 4.3 – Example of decryption using the key and ciphertext in txt files