Secured Audio Player

With Encryption and Decryption functions

Agenda

- Technology used
- Details of the design
- Demonstration of the implementation

Technology used

Front-end

- React + Vite
- CSS

Deploy

- Firebase

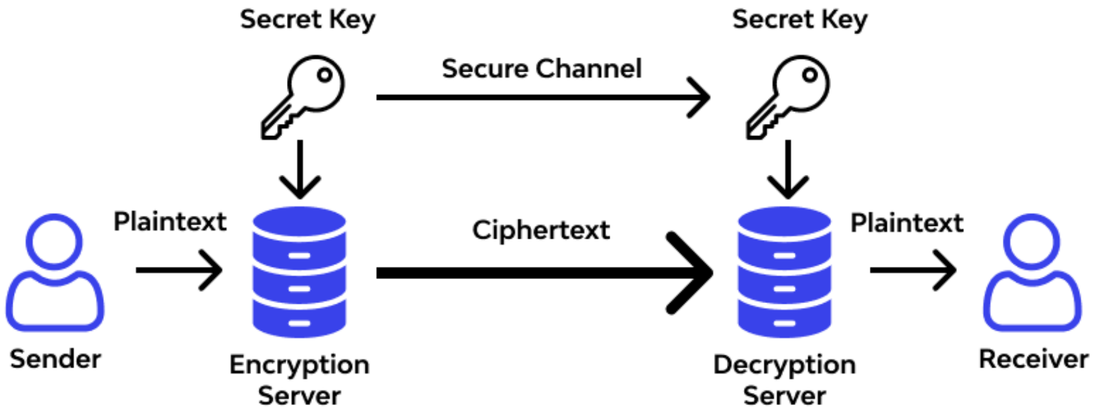
Back-end

- CryptoJS library : JavaScript library that provides a collection of cryptographic algorithms, including AES encryption and decryption, SHA hashing, and HMAC authentication.
- Built-in browser APIs
  - FileReader : is a built-in web API that provides methods for reading data from files stored on the user's computer.
  - Blob : is a type of object in JavaScript that represents a raw data in a format that can be easily converted to other types of data, such as text or audio file.
  - Uint8Array : is a typed array in JavaScript that represents an array of 8-bit unsigned integers. It is often used to manipulate binary data.

AES
(Advanced Encryption Standard)

- AES used symmetric encryption algorithm for secure sensitive data.
- It uses a secret key to encrypt and decrypt the data.
- Use in application such as online banking, data storage, and communication protocols.

AES Algorithm Working



Encryption and Decryption

Encryption

For the Encryption part, it uses FileReader API to read an audio file, split it into two equal halves, encrypt each half using the AES encryption algorithm from the CryptoJS library, and then concatenate the encrypted halves into a single encrypted string. The encrypted string is then used to create a new Blob object and a new File object with the .encrypted extension. The new File object is saved using setEncryptedFile function and the original file name is saved using setOriginalFileName function.

Decryption

For the Decryption part, it also uses the FileReader API to read an encrypted audio file. The encrypted string is split into two equal halves, each half is decrypted using the same AES encryption algorithm and key used in the encryption process. The decrypted halves are then concatenated into a single string, which is then converted to a binary format using the atob() function. The binary data is then used to create a new Blob object, which is used to create a new File object with the original file name and extension. The new File object is saved using setDecryptedFile.

Details of the Design

Secure Audio Player

3 Put the key in the box

Enter your Secret Key: poUwt{!gImK:ut0s}*AEDfb[Show

You can click "Show" or "Hide" button

Secret Key length must be 16, 24, or 32 characters

poUwt{!gImK:ut0s}*AEDfb[

2 Generate Random Key

1 Select Key length: 24 characters

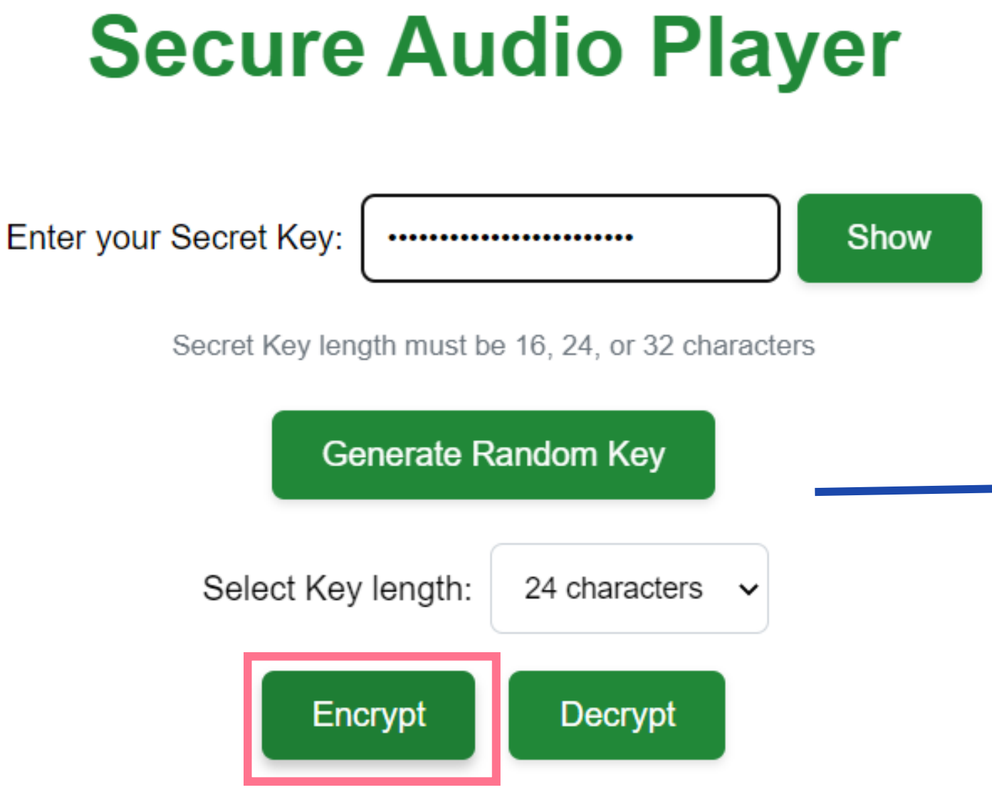
Select Key length: 16 characters

16 characters
24 characters
32 characters

Encrypt Decrypt

Encrypt

Details of the Design



Click "Encrypt"

This page will show up

Click " Encrypt" button

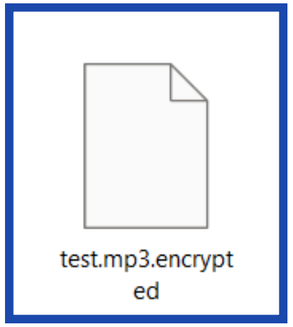
Details of the Design

Secure Audio Player

Click "Download"

Download

Click to download the encrypted file



Then, will show the main page automatically

Secure Audio Player

Enter your Secret Key: ••••••••••••••••••••••• Show

Secret Key length must be 16, 24, or 32 characters

Generate Random Key

Select Key length: 24 characters

Encrypt Decrypt

Click "Decrypt"

Import file that you recieved

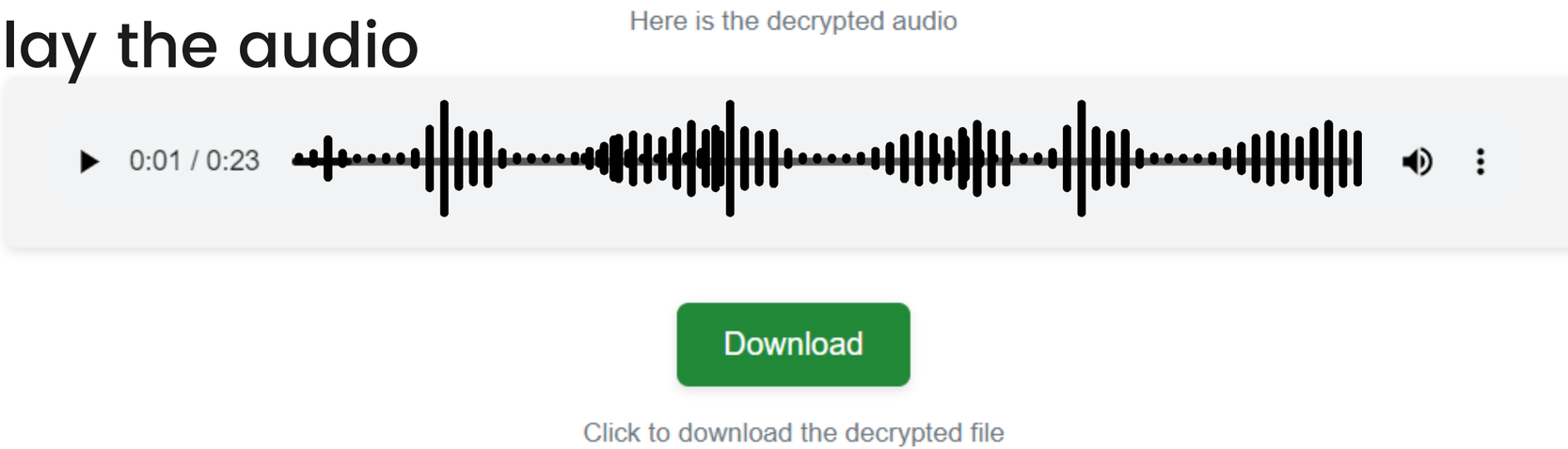
Choose File test.mp3.encrypted

Decrypt

Support only .encrypted file type encrypted by this tool

Details of the Design

You can play the audio



Click "Dowload" , You will get the file that decrypted.

Demonstration of the implementation

Link Demo: https://audioplayer-49127.web.app/

Thank you

Q & A session

Team

6388014 Waris Damkham

6388087 Chanisara Kotrachai