Blox

Abhishek Jamhoriya

Project Link: Visual_Cryptography
Code Link: github

01. **Question) Elaborate what your internship or academic projects were?**
    a. **What did the system do?**
    The system provided a platform where users could encrypt and decrypt images securely using a **security key**. Users uploaded their images, which were encrypted using the AES (Advanced Encryption Standard) algorithm through the **CryptoJS library**, and the encrypted data was stored securely. Similarly, users could decrypt and view their encrypted images by providing the correct security key. The system used Firebase for **user authentication**, **data storage**, and **hosting**, ensuring seamless and secure operations.

    b. **What other systems have you seen in the wild like that?**
        i. **File Encryption Tools**: Services like **VeraCrypt** and **AxCrypt** offer file and image encryption functionalities with user keys
        ii. **Cloud Storage Services**: Platforms like **Dropbox**, **Google Drive**, and **iCloud** implement encryption for user-uploaded files, often combining authentication with encryption keys.
        iii. **Visual Cryptography Platforms**: Academic and demo tools used to demonstrate image-based cryptographic techniques for secure sharing.

    c. **How do you approach the development problem?**
        i. **Requirements Analysis:**
            1. Identified key functionalities like image upload, encryption/decryption, and secure storage.
            2. Ensured the system had a simple yet secure authentication mechanism.
        ii. **Technology Selection:**
            1. CryptoJS for AES-based encryption/decryption.
            2. Firebase for its comprehensive suite (Authentication, Storage, and Hosting).
        iii. **System Design:**

1. Modularized the app into components: front-end UI for user interaction, backend Firebase integration for storage/authentication, and CryptoJS library for cryptographic operations.

iv. **Development Process:**

1. Iterative implementation: starting with the encryption/decryption functionality, integrating Firebase storage, then adding authentication and hosting.
2. Ensured encryption/decryption occurs client-side to maintain data confidentiality.

v. **Testing and Debugging:**

1. Tested edge cases like incorrect security keys, large file uploads, and multi-user operations.
2. Validated security to prevent unauthorized access.

d. What were interesting aspects where you copied code from Stack Overflow?

Some specific areas where Stack Overflow snippets were used:

i. **CryptoJS AES Implementation:**

1. Found snippets for AES encryption and decryption syntax using CryptoJS.

ii. **Firebase Integration:**

1. Borrowed sample code for setting up Firebase Authentication and Storage, ensuring quick onboarding and reducing initial setup errors.

iii. **Error Handling:**

1. Learned effective ways to handle promises and asynchronous calls, especially during Firebase storage operations.

e. **What did you learn from some very specific copy paste? Mention explicitly some of them.**

i. **AES Encryption:**

1. From a CryptoJS example, copied an implementation for encrypting data and secret key:

Stack Overflow :

```
var text = "My Secret text";
var key = CryptoJS.enc.Base64.parse("253D3FB468A0E24677C28A624BE0F939");
var iv  = CryptoJS.enc.Base64.parse("                ");
var encrypted = CryptoJS.AES.encrypt(text, key, {iv: iv});
console.log(encrypted.toString());

var decrypted = CryptoJS.AES.decrypt(encrypted, key, {iv: iv});
console.log(decrypted.toString(CryptoJS.enc.Utf8));
```

My Code:

```
var CryptoJS = require("crypto-js");
// Decrypt
var bytes = CryptoJS.AES.decrypt(`${fileBase64String}`, `${key}`);
// console.log("Bytes"+bytes.toString(CryptoJS.enc.Utf8));
  // console.log("bytes"+bytes.toString(CryptoJS.enc.Utf8))
  var originalText = bytes.toString(CryptoJS.enc.Utf8);
```

2. **Learning**: Understood how AES encryption works using only a secret key. Learned that CryptoJS internally handles aspects like IV generation when it isn't explicitly specified.

ii. **Firebase Authentication:**

1. Copied a Firebase sign-in snippet:

   My Code:

```
function signInWithEmailPassword(address) {

  var email = address.email;
  var password = address.password;

  firebase.auth().signInWithEmailAndPassword(email, password)
    .then((userCredential) => {
      // Signed in
      var user = userCredential.user;
      ReactDOM.render(<Router><Home/></Router>, document.getElementById('root'));
      console.log(user)
      alerttoast("Logged-in successfully");


    })
    .catch((error) => {
      var errorCode = error.code;
      var errorMessage = error.message;
      alerttoast(errorMessage)
    });

}
```

2. **Learning:** Gained insight into managing user authentication flows and handling common errors like invalid credentials or network issues.