Informe Laboratorio 4

Sección 2

Alumno Matias Herrera
e-mail: matias.herrera2@mail.udp.cl

Mayo de 2024

Índice

1. Descripción de actividades **2**

2. Desarrollo (Parte 1) **3**
2.1. Detecta el cifrado utilizado por el informante 3
2.2. Logra que el script solo se gatille en el sitio usado por el informante 4
2.3. Define función que obtiene automáticamente el password del documento . . . 4
2.4. Muestra la llave por consola 5

3. Desarrollo (Parte 2) **6**
3.1. Reconoce automáticamente la cantidad de mensajes cifrados 6
3.2. Muestra la cantidad de mensajes por consola 7

4. Desarrollo (Parte 3) **7**
4.1. Importa la librería cryptoJS 7
4.2. Utiliza SRI en la librería CryptoJS 7
4.3. Repercusiones de SRI inválido 8
4.4. Logra decifrar uno de los mensajes 8
4.5. Imprime todos los mensajes por consola 9
4.6. Muestra los mensajes en texto plano en el sitio web 9
4.7. El script logra funcionar con otro texto y otra cantidad de mensajes 10
4.8. Indica url al código .js implementado para su validación 11

1. Descripción de actividades

Para este laboratorio, deberá utilizar Tampermonkey y la librería CryptoJS (con SRI) para lograr obtener los mensajes que le está comunicando su informante. En esta ocasión, su informante fue más osado y se comunicó con usted a través de un sitio web abierto a todo el público https://cripto.tiiny.site/.

Sólo un ojo entrenado como el suyo logrará descifrar cuál es el algoritmo de cifrado utilizado y cuál es la contraseña utilizada para lograr obtener la información que está oculta.

1. Desarrolle un plugin para tampermonkey que permita obtener la llave para el descifrado de los mensajes ocultos en la página web. La llave debe ser impresa por la consola de su navegador al momento de cargar el sitio web. Utilizar la siguiente estructura:

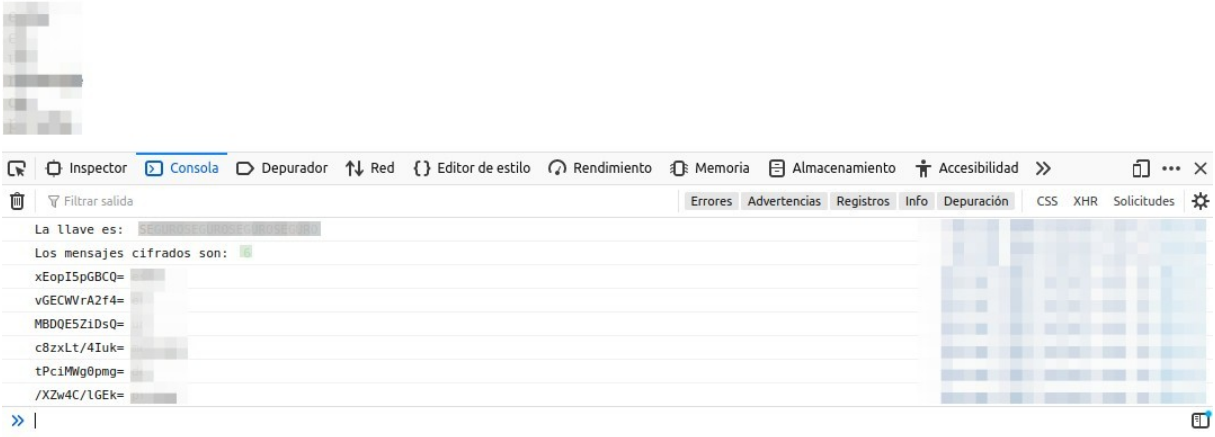
   - La llave es: KEY

2. En el mismo plugin, se debe detectar el patrón que permite identificar la cantidad de mensajes cifrados. Debe imprimir por la consola la cantidad de mensajes cifrados. Utilizar la siguiente estructura: Los mensajes cifrados son: NUMBER

3. En el mismo plugin debe obtener cada mensaje cifrado y descifrarlo. Ambos mensajes deben ser informados por la consola (cifrado espacio descifrado) y además cada mensaje en texto plano debe ser impreso en la página web.

   El script desarrollado debe ser capaz de obtener toda la información del sitio web (llave, cantidad de mensajes, mensajes cifrados) sin ningún valor forzado. Para verificar el correcto funcionamiento de su script se utilizará un sitio web con otro texto y una cantidad distinta de mensajes cifrados. Deberá indicar la url donde se podrá descargar su script.

   Un ejemplo de lo que se debe visualizar en la consola, al ejecutar automáticamente el script, es lo siguiente:



Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica.

[redacted]

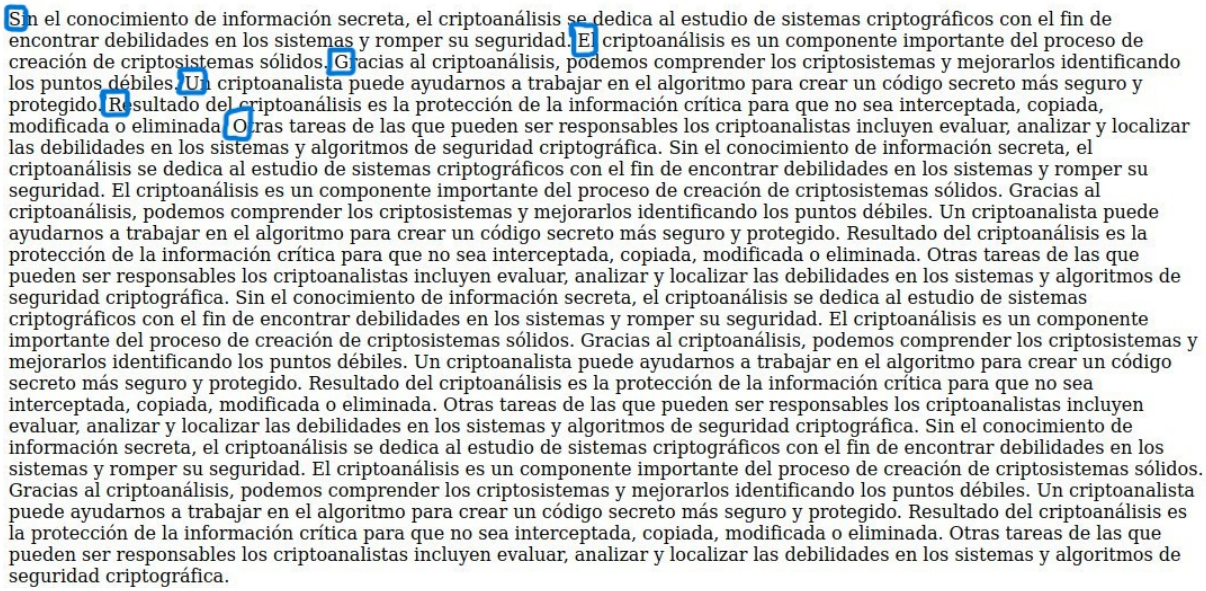
```
La llave es: [redacted]
Los mensajes cifrados son: [redacted]
xEopI5pGBCQ= [redacted]
vGECWVrA2f4= [redacted]
MBDQE5ZiDsQ= [redacted]
c8zxLt/4Iuk= [redacted]
tPciMWg0pmg= [redacted]
/XZw4C/lGEk= [redacted]
```

[redacted]

2. Desarrollo (Parte 1)

2.1. Detecta el cifrado utilizado por el informante

El cifrado que usa el informante es a través de oraciones separadas por punto en texto plano donde cada letra mayúscula al inicio de la oración es una letra de la clave secreta que nos entrego el informante. Como se puede observar en la figura 1



Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica. Sin el conocimiento de información secreta, el criptoanálisis se dedica al estudio de sistemas criptográficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanálisis es un componente importante del proceso de creación de criptosistemas sólidos. Gracias al criptoanálisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos débiles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un código secreto más seguro y protegido. Resultado del criptoanálisis es la protección de la información crítica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptográfica.

Figura 1: Texto y mayúsculas

2.2. Logra que el script solo se gatille en el sitio usado por el informante

Para lograr este objetivo solo debemos definir a travez de tampermonkey que el script solo se ejecutara cuando la url del sitio sea **https://cripto.tiiny.site/** en caso contrario no se ejecutara se usa la exprecion @match para definir lo mencionado anteriormente como se puede observar en la figura .

2.3. Define función que obtiene automáticamente el password del documento

La funcion a utilizar es el siguiente:

```
var parrafoDiv = document.querySelector('p');
     if (!parrafoDiv) return;

     var textoCompleto = parrafoDiv.innerText;
     var oraciones = textoCompleto.split('. ');
     var contraseña = "";
     for (var i = 0; i < oraciones.length; i++) {
         var primeraLetra = oraciones[i].charAt(0);
         contraseña += primeraLetra;
     }
```

```
if (contraseña.length > 24) {
    contraseña = contraseña.substring(0, 24);
}

console.log("La llave es:", contraseña);
```

- '**var parrafoDiv = document.querySelector('p');**': Busca un elemento HTML <p> en el documento y lo guarda en la variable **parrafoDiv**. Si no se encuentra ningún elemento <p>, el script termina aquí.
- '**if (!parrafoDiv) return;**': Si no se encuentra ningún párrafo, el script se detiene y no se ejecuta más.
- '**var textoCompleto = parrafoDiv.innerText;**': Obtiene el texto completo dentro del párrafo seleccionado y lo guarda en la variable **textoCompleto**.
- '**var oraciones = textoCompleto.split('. ');**': Divide el texto completo en oraciones usando el punto y espacio como separador y guarda cada oración en un array llamado **oraciones**.
- '**var contraseña = ;**': Inicializa una variable llamada **contraseña** para almacenar la contraseña generada.
- '**for (var i = 0; i ¡oraciones.length; i++) ...** ': Itera sobre cada oración en el array **oraciones**.
- '**var primeraLetra = oraciones[i].charAt(0);**': Obtiene la primera letra de cada oración y la guarda en la variable **primeraLetra**.
- '**contraseña += primeraLetra;**': Agrega la primera letra de cada oración a la variable **contraseña**.
- '**if (contraseña.length ¿24) contraseña = contraseña.substring(0, 24);**': Si la longitud de la contraseña es mayor que 24 caracteres, se corta para asegurarse de que no exceda ese límite.
- '**console.log("La llave es:", contraseña);**': Imprime la contraseña generada en la consola del navegador, a traves de la estructura "La llave es:".

2.4. Muestra la llave por consola

Luego de tener la función para obtener la clave accedemos a la pagina objetivo para obtener la llave mediante consola como se puede observar en la figura 2

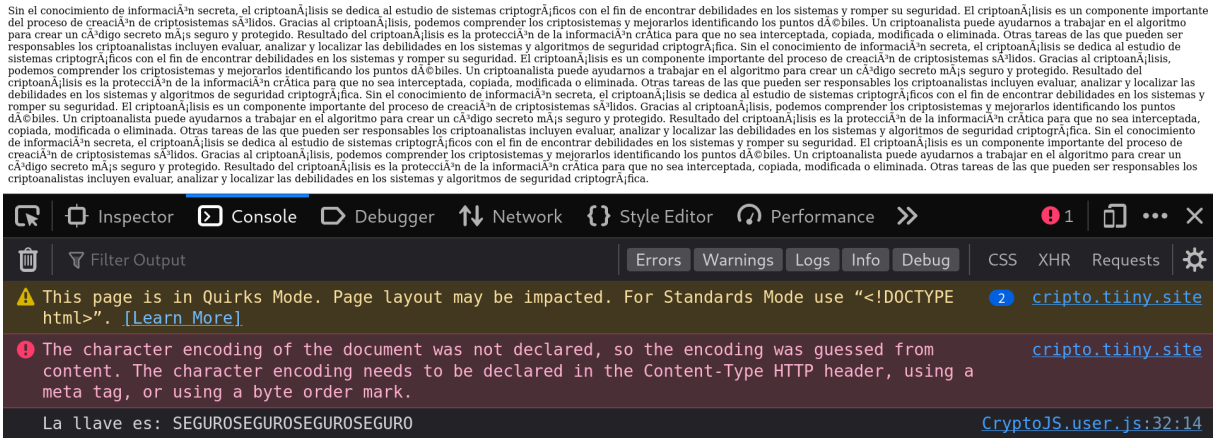


Figura 2: Resultado del script

3. Desarrollo (Parte 2)

3.1. Reconoce automáticamente la cantidad de mensajes cifrados

Cuando revisamos los elementos de la pagina notamos que tenemos diferentes **div** con mensajes cifrados y con la clase **Mx** donde x es un numero, debemos realizar un script que identifique ese patrón de de letra en la clase de los **div** para así determinar la cantidad de mensajes cifrados que tenemos. El codigo es el siguiente

```
 var elementos = document.querySelectorAll('div[class^="M"]');
  var ids = [];
  elementos.forEach(function(elemento) {
    ids.push(elemento.id);
});
  var repeticiones = {};
  var patron = /\d+/;

  for (i = 0; i < elementos.length; i++) {
      var clases = elementos[i].classList;
      for (var j = 0; j < clases.length; j++) {
          var clase = clases[j];
          if (patron.test(clase)) {
              if (repeticiones[clase]) {
                  repeticiones[clase]++;
              } else {
                  repeticiones[clase] = 1;
              }
          }
```

```
    }
}
}
var mensajeCifrado = "Los mensajes cifrados son: " + Object.keys(repeticiones).le
console.log(mensajeCifrado);
```

Como se puede observar en el codigo anterior tenemos una variable llamada elementos, el cual lee el codigo fuente y selecciona todos los div que contengan la clase 'M' y tomamos un variable que es el ID que es lo que acompaña a la clase M, realizamos el conteo de las clases y retornamos por consola el resultado con la estructura solicitada.

3.2. Muestra la cantidad de mensajes por consola

Como resultado de este Script obtenemos la cantidad de 6 como se puede observa en la figura 3

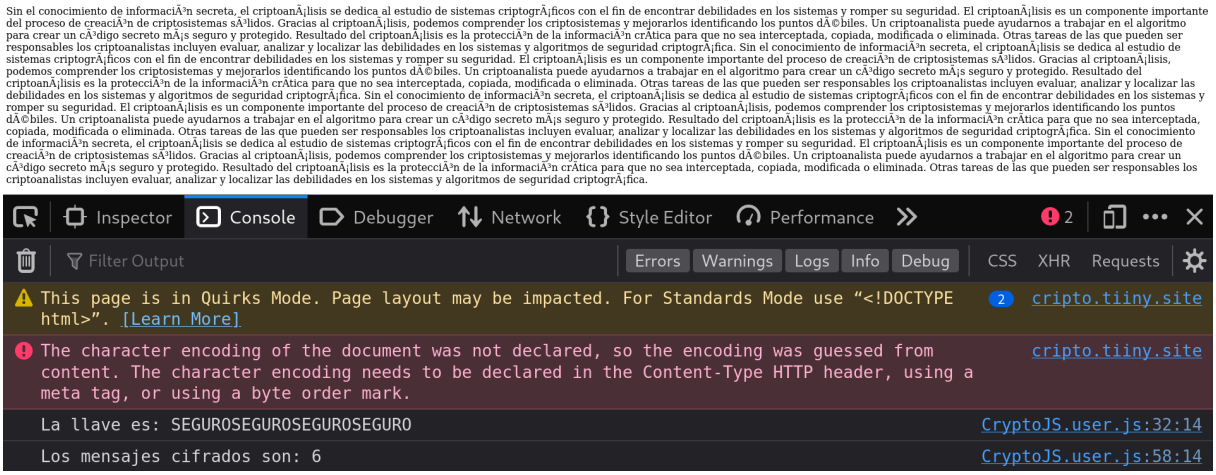


Figura 3: Cantidad de mensajes cifrados

4. Desarrollo (Parte 3)

4.1. Importa la librería cryptoJS

Para lograr importar la librería de cryptoJS se necesita una url del recurso, esto se legra mencionando en el script con un **@require** y es de la siguiente forma:

```
// @require      https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.2.0/crypto-js.min.
```

4.2. Utiliza SRI en la librería CryptoJS

Para utilizar SRI en la libreria debemos agregar informacion en la URL de CryptoJS quedando de la siguiente manera:

```
// @require      https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.2.0
/crypto-js.min.js#sha512-a+SUDuwNzXDvz4XrIcXHuCf089
/iJAoN4lmrXJg18XnduKK6YlDHNRalv4yd1N40OKI80tFidF+rqTFKGPoWFQ==
```

Este hash SHA-512 garantiza la integridad del archivo descargado. Cuando el navegador intenta cargar el script desde esta URL, primero calculará el hash del archivo descargado y lo comparará con el hash proporcionado en la URL. Si los hashes coinciden, el navegador cargará el script. Si no coinciden, es posible que el navegador muestre un mensaje de error o simplemente no cargue el script.

4.3. Repercusiones de SRI inválido

Las repercusiones que podría causar un SRI (Subresource Integrity) es que el archivo ha sido manipulado y no cumple con los requerimientos de seguridad de subrecursos por lo que podría traer componentes maliciosos que podrían comprometer la información que intentamos obtener de forma anónima.

4.4. Logra decifrar uno de los mensajes

El codigo para decifrar los mensajes en 3DES es el siguiente:

```
var divs = document.getElementsByTagName('div');
 var contenidoDesencriptado = '';
 for (i = 0; i < divs.length; i++) {
     var div = divs[i];
     var id = div.id;
     var ciphertextBytes = CryptoJS.enc.Base64.parse(id);
     var decryptedBytes = CryptoJS.TripleDES.decrypt({ ciphertext: ciphertextBytes
         mode: CryptoJS.mode.ECB,
         padding: CryptoJS.pad.Pkcs7
     });
     var decryptedText = decryptedBytes.toString(CryptoJS.enc.Utf8);
     console.log(id + ": " + decryptedText);
```

Este código de JavaScript busca todos los elementos ¡div¿en la página, asumiendo que sus IDs contienen texto cifrado en formato Base64. Luego, desencripta este texto utilizando el algoritmo TripleDES con una contraseña proporcionada, mostrando el contenido desencriptado en la consola. El resultado del primer mensaje es el que se puede observar en la siguiente figura 4

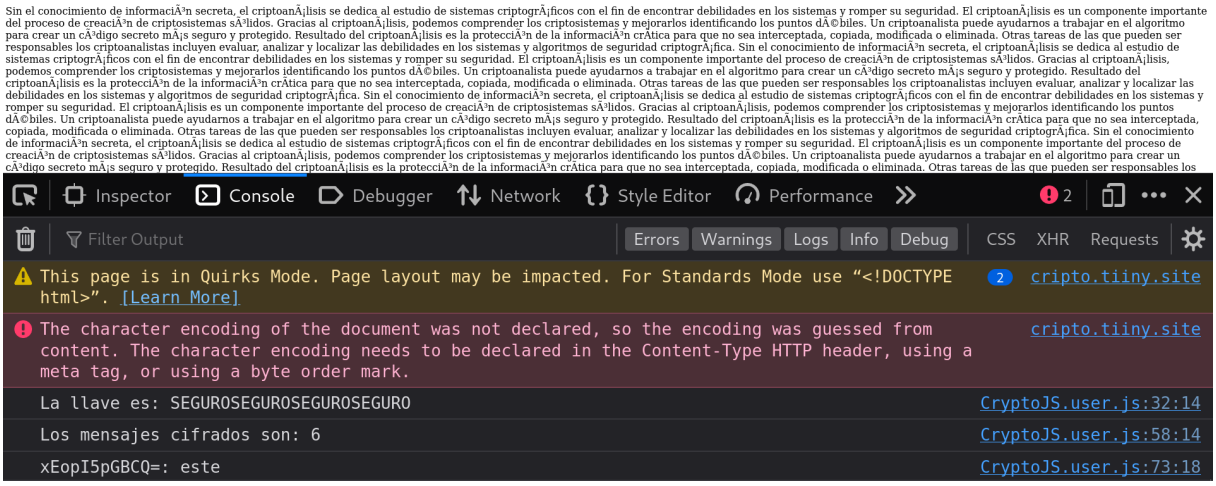


Figura 4: Mensaje descifrado por consola

4.5. Imprime todos los mensajes por consola

El mismo código se aplica para los demás mensajes cifrados, su resultado se puede observar en la figura 5

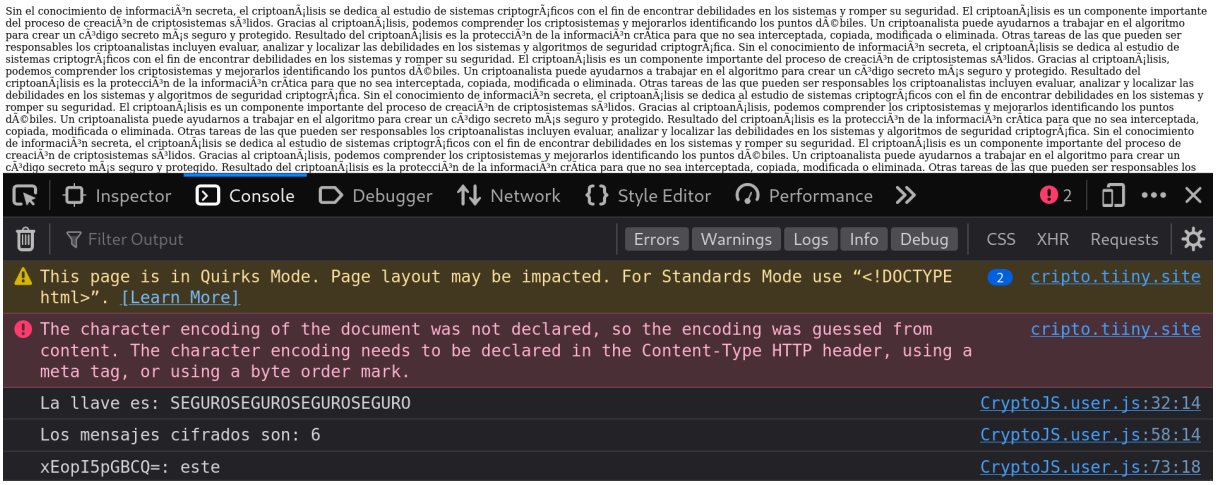


Figura 5: Mensaje descifrado por consola

4.6. Muestra los mensajes en texto plano en el sitio web

Para realizar que los mensajes cifrados se observen en texto plano se deben agregar algunas cosas al código visto anteriormente, las modificaciones serian las siguientes:

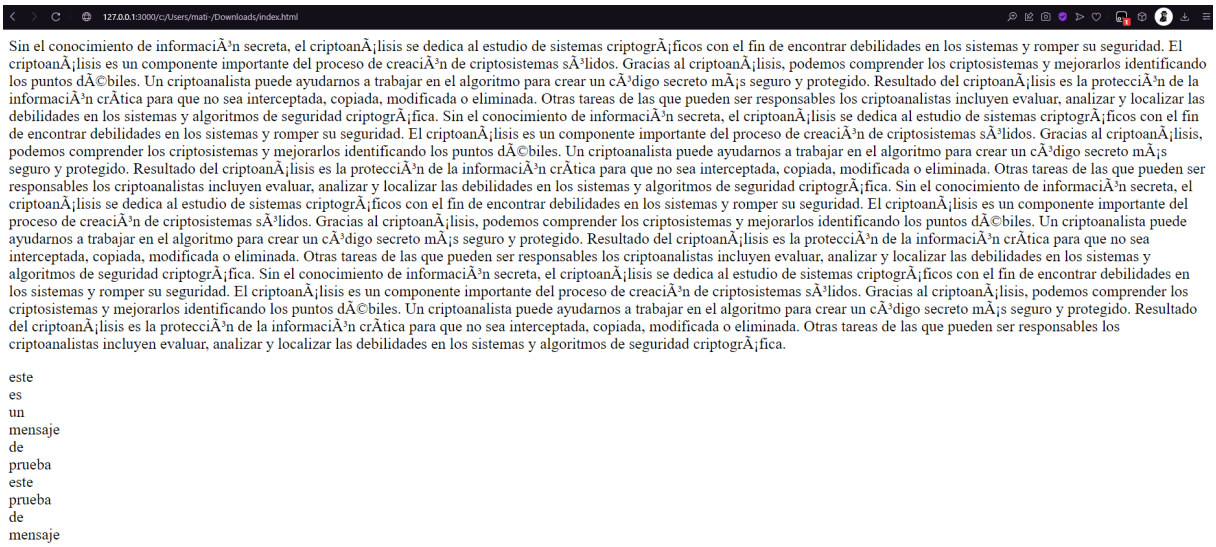
```
var divs = document.getElementsByTagName('div');
 var contenidoDesencriptado = '';
```

```
for (i = 0; i < divs.length; i++) {
    var div = divs[i];
    var id = div.id;
    var ciphertextBytes = CryptoJS.enc.Base64.parse(id);
    var decryptedBytes = CryptoJS.TripleDES.decrypt({ ciphertext: ciphertextBytes
        mode: CryptoJS.mode.ECB,
        padding: CryptoJS.pad.Pkcs7
    });
    var decryptedText = decryptedBytes.toString(CryptoJS.enc.Utf8);
    console.log(id + ": " + decryptedText);
    contenidoDesencriptado += decryptedText + ' ';
}

var palabrasDesencriptadas = contenidoDesencriptado.split(' ');
var mensajeDesencriptado = document.createElement('p');
for (var k = 0; k < palabrasDesencriptadas.length; k++) {
    mensajeDesencriptado.innerHTML += palabrasDesencriptadas[k] + '<br>';
}

document.body.appendChild(mensajeDesencriptado);
```

Lo que hace ahora el script ademas de entregar los mensajes desencriptado ademas agrega cada mensaje en la pagina web separado por saltos de linea y agregandolos a travez de elementos ¡p¿y ¡br¿.

4.7. El script logra funcionar con otro texto y otra cantidad de mensajes

El script para lograr esto es el mismo simplemente se han tenido que agregar mas mensajes cifrados con la misma nomenclatura, el resultado de esto es la siguiente:



Sin el conocimiento de informaciÃ³n secreta, el criptoanÃ¡lisis se dedica al estudio de sistemas criptogrÃ¡ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanÃ¡lisis es un componente importante del proceso de creaciÃ³n de criptosistemas sÃ³lidos. Gracias al criptoanÃ¡lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos dÃ©biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un cÃ³digo secreto mÃ¡s seguro y protegido. Resultado del criptoanÃ¡lisis es la protecciÃ³n de la informaciÃ³n crÃtica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptogrÃ¡fica. Sin el conocimiento de informaciÃ³n secreta, el criptoanÃ¡lisis se dedica al estudio de sistemas criptogrÃ¡ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanÃ¡lisis es un componente importante del proceso de creaciÃ³n de criptosistemas sÃ³lidos. Gracias al criptoanÃ¡lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos dÃ©biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un cÃ³digo secreto mÃ¡s seguro y protegido. Resultado del criptoanÃ¡lisis es la protecciÃ³n de la informaciÃ³n crÃtica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptogrÃ¡fica. Sin el conocimiento de informaciÃ³n secreta, el criptoanÃ¡lisis se dedica al estudio de sistemas criptogrÃ¡ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanÃ¡lisis es un componente importante del proceso de creaciÃ³n de criptosistemas sÃ³lidos. Gracias al criptoanÃ¡lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos dÃ©biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un cÃ³digo secreto mÃ¡s seguro y protegido. Resultado del criptoanÃ¡lisis es la protecciÃ³n de la informaciÃ³n crÃtica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptogrÃ¡fica. Sin el conocimiento de informaciÃ³n secreta, el criptoanÃ¡lisis se dedica al estudio de sistemas criptogrÃ¡ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoanÃ¡lisis es un componente importante del proceso de creaciÃ³n de criptosistemas sÃ³lidos. Gracias al criptoanÃ¡lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos dÃ©biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un cÃ³digo secreto mÃ¡s seguro y protegido. Resultado del criptoanÃ¡lisis es la protecciÃ³n de la informaciÃ³n crÃtica para que no sea interceptada, copiada, modificada o eliminada. Otras tareas de las que pueden ser responsables los criptoanalistas incluyen evaluar, analizar y localizar las debilidades en los sistemas y algoritmos de seguridad criptogrÃ¡fica.

este
es
un
mensaje
de
prueba
este
prueba
de
mensaje

Figura 6: Mensaje descifrado por consola

4.8. Indica url al código .js implementado para su validación

La URL de los codigos es la siguiente: `https://github.com/ELABUELO19/Lab_Crypto/tree/main/Lab4`

El Script para realizar las comprobaciones se llama 'Script_pagina_replica.js'

Conclusiones y comentarios

En conclusión podemos determinar que la experiencia de laboratorio para obtener la clave y los mensajes cifrados del informante han obtenido resultados positivos, con algunas dificultades a la hora de entender los procedimiento a realizar, pero no generando un estancamiento completo de la actividad, aprendimos el funcionamiento de como uno puedo entregar información sensible (si es el caso) a través de paginas web abiertas.

Issues

Las dificultades encontradas fueron las siguientes:

- Problema 1: Saber como importar cryptoJS de forma correcta, inicialmente la hora de comenzar el laboratorio se me generaron problemas al hora de importar la librería, la url no funcionaba o estaba mala y no entregaba nada, para solucionar esto tuve que realizar un importación de una librería precargada de cryptoJS para realizar la actividad.

- Problema 2: El reconocimiento del cifrado que utilizo fue complicado de entender a que se referian con ese objetivo debido a que nos entregaban la url del texto y su codigo fuente podiamos observar que el texto tenia mayusculas y cuando las juntaba cada una comenzaba a tener sentido, al igual que con los mensajes cifrados tuve que solicitar a chatgpt que tipo posible de cifrado era ya que al ir probando uno por uno de los vistos en clases no podia entender cual era.

- Problema 3: Para entender a que se referian con SRI tuve que investigar un poco sobre el tema y como comprobar la integridad de un archivo a traves de los hash y como posteriormente integrarlo en la url de la pagina web para obtener CryptoJS, entendiendo eso pude explicar el procedimiento y entregar la URL correctamente

- Problema 4: A la hora de solicitar una pagina en donde comprobar el funcionamiento del Script no sabia como importar una url utilizando el mismo Script para hacer las comprobaciones, en lo que resulto fue en copiar el codigo fuente de la pagina web para poder visualizarlo de manera local para realizar las comprobaciones.