

Informe Laboratorio 4

Sección 1

Kevin Muñoz

kevin.munoz_a@mail.udp.cl

Mayo de 2023

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	6
2.3. Define función que obtiene automáticamente el password del documento	7
2.4. Muestra la llave por consola	7
3. Desarrollo (Parte 2)	8
3.1. reconoce automáticamente la cantidad de mensajes cifrados	8
3.2. muestra la cantidad de mensajes por consola	8
4. Desarrollo (Parte 3)	9
4.1. Importa la librería cryptoJS	9
4.2. Utiliza SRI en la librería CryptoJS	9
4.3. Logra decifrar uno de los mensajes	9
4.4. Imprime todos los mensajes por consola	9
4.5. Muestra los mensajes en texto plano en el sitio web	9
4.6. El script logra funcionar con otro texto y otra cantidad de mensajes	10
4.7. 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.

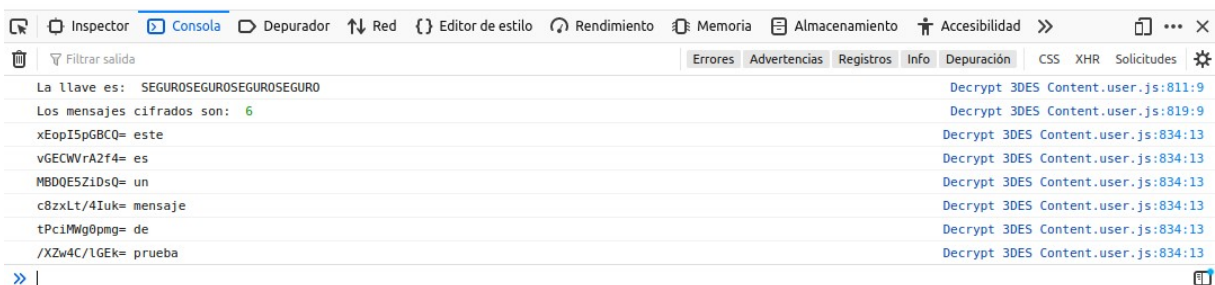
1. Desarrolle un plugin para tampermonkey que permita obtener la llave para el descifrado de los mensajes ocultos en la página web. La llave debe ser impresa por la consola de su navegador al momento de cargar el sitio web. Utilizar la siguiente estructura:
 - La llave es: KEY
2. En el mismo plugin,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

este
es
un
mensaje
de
prueba



2. Desarrollo (Parte 1)

2.1. Detecta el cifrado utilizado por el informante

Para llevar a cabo la detección del método de cifrado utilizado, fue necesario primero localizar la clave de descifrado. Esta tarea se inició con un análisis detallado del texto en la pagina. Durante este proceso, se descubrió que la clave correspondía a las letras mayúsculas presentes en el texto. Se observó que este patrón de letras mayúsculas se repetía cuatro veces a lo largo del texto, lo cual fue el hallazgo de la clave que corresponde a 'SEGUROSEGUROSEGUROSEGURO'.

Sin el conocimiento de informaci3n secreta, el criptoan3lisis se dedica al estudio de sistemas criptogr3ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoan3lisis es un componente importante del proceso de creaci3n de criptosistemas s3lidos. Gracias al criptoan3lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos d3biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un c3digo secreto m3s seguro y protegido. Resultado del criptoan3lisis es la protecci3n de la informaci3n cr3tica 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 criptogr3fica. Sin el conocimiento de informaci3n secreta, el criptoan3lisis se dedica al estudio de sistemas criptogr3ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoan3lisis es un componente importante del proceso de creaci3n de criptosistemas s3lidos. Gracias al criptoan3lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos d3biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un c3digo secreto m3s seguro y protegido. Resultado del criptoan3lisis es la protecci3n de la informaci3n cr3tica 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 criptogr3fica. Sin el conocimiento de informaci3n secreta, el criptoan3lisis se dedica al estudio de sistemas criptogr3ficos con el fin de encontrar debilidades en los sistemas y romper su seguridad. El criptoan3lisis es un componente importante del proceso de creaci3n de criptosistemas s3lidos. Gracias al criptoan3lisis, podemos comprender los criptosistemas y mejorarlos identificando los puntos d3biles. Un criptoanalista puede ayudarnos a trabajar en el algoritmo para crear un c3digo secreto m3s seguro y protegido. Resultado del criptoan3lisis es la protecci3n de la informaci3n cr3tica 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 criptogr3fica.

Figura 1: Texto de la pagina

Al continuar indigando en la pagina, se observo los elementos de la pagina viendo asi que, los 'div' en c3digo HTML, cada uno con una clase que indica un n3mero de mensaje secuencial (como "M1" para el primer mensaje, "M2" para el segundo, etc.) y un atributo ID que contiene un mensaje cifrado en lo que parece ser base64. La estructura sugiere que los mensajes est3n ordenados y que el contenido cifrado se aloja dentro de los atributos ID.

Para descifrar estos mensajes, se identific3 que solo se proporciona una clave de cifrado y no se necesitan elementos adicionales como un Vector de Inicializaci3n o Nonce, lo que indica que el modo de cifrado probablemente sea ECB (Electronic Codebook), un modo que no requiere de estos elementos. Dado este modo, se consult3 la biblioteca CryptoJS que soporta varios algoritmos de cifrado como AES, DES, Triple DES y RC4. La tarea entonces consisti3 en probar cada uno de estos algoritmos con la clave dada, descifrando las cadenas base64 hasta que los mensajes fueran legibles, lo que indicaría que se ha encontrado el algoritmo correcto.

```
<div class="M1" id="xEopI5pGBCQ="> </div>
<div class="M2" id="vGECWVrA2f4="> </div>
<div class="M3" id="MBDQE5ZiDsQ="> </div>
<div class="M4" id="c8zxLt/4Iuk="> </div>
<div class="M5" id="tPciMWg0pmg="> </div>
<div class="M6" id="/XZw4C/lGEk="> </div>
```

Figura 2: Inspeccionar elementos de la pagina

El proceso de descifrado implica un m3todo sistem3tico de prueba y error utilizando CryptoJS, hasta que los datos cifrados revelen informaci3n coherente y legible, lo que confirmaría que la clave y el algoritmo seleccionado son los adecuados.

```
Elementos con clase 'M' encontrados: 6
ID: xEopI5pGBCQ=, el mensaje Descifrado es:
ID: vGECWVrA2f4=, el mensaje Descifrado es:
ID: MBDQE5ZiDsQ=, el mensaje Descifrado es:
ID: c8zxLt/4Iuk=, el mensaje Descifrado es:
ID: tPciMWg0pmg=, el mensaje Descifrado es:
ID: /XZw4C/lGEk=, el mensaje Descifrado es:
```

Figura 3: Prueba de AES

```
Elementos con clase 'M' encontrados: 6
ID: xEopI5pGBCQ=, el mensaje Descifrado es:
ID: vGECWVrA2f4=, el mensaje Descifrado es:
ID: MBDQE5ZiDsQ=, el mensaje Descifrado es:
ID: c8zxLt/4Iuk=, el mensaje Descifrado es:
ID: tPciMWg0pmg=, el mensaje Descifrado es:
ID: /XZw4C/lGEk=, el mensaje Descifrado es:
```

Figura 4: Prueba de DES

```
Elementos con clase 'M' encontrados: 6
ID: xEopI5pGBCQ=, el mensaje Descifrado es:
ID: vGECWVrA2f4=, el mensaje Descifrado es:
ID: MBDQE5ZiDsQ=, el mensaje Descifrado es:
ID: c8zxLt/4Iuk=, el mensaje Descifrado es:
ID: tPciMWg0pmg=, el mensaje Descifrado es:
ID: /XZw4C/lGEk=, el mensaje Descifrado es:
```

Figura 5: Prueba de RC4

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

Como se puede apreciar en todos estas pruebas con AES, DES Y RC4 no se obtuvieron resultados.

```
Elementos con clase 'M' encontrados: 6
ID: xEopI5pGBCQ=,el mensaje Descifrado es: este
ID: vGECWVrA2f4=,el mensaje Descifrado es: es
ID: MBDQE5ZiDsQ=,el mensaje Descifrado es: un
ID: c8zxLt/4Iuk=,el mensaje Descifrado es: mensaje
ID: tPciMWg0pmg=,el mensaje Descifrado es: de
ID: /XZw4C/lGEk=,el mensaje Descifrado es: prueba
```

Figura 6: Prueba de Triple DES

Al aplicar Triple DES se puede ver como se obtivo el descifrado de los mensaje.

```
function descifrarContenido(claveCifrado) {
  const elementosClaseM = document.querySelectorAll('[class*="M"]');
  console.log('Elementos con clase 'M' encontrados: ${elementosClaseM.length}');

  elementosClaseM.forEach(elemento => {
    const idCifrado = elemento.id;
    const claveEncriptacion = CryptoJS.enc.Utf8.parse(claveCifrado);

    // Configuración para el descifrado
    const configuracion = {
      mode: CryptoJS.mode.ECB
    };

    // Descifrar el ID
    const idDescifrado = CryptoJS.TripleDES.decrypt(idCifrado, claveEncriptacion, configuracion);
    console.log(`ID: ${elemento.id},el mensaje Descifrado es: ${idDescifrado.toString(CryptoJS.enc.Utf8)}`);
    const textoDescifrado = idDescifrado.toString(CryptoJS.enc.Utf8);
    const elementoH2 = document.createElement('h2');
    elementoH2.textContent = textoDescifrado;
    document.body.appendChild(elementoH2);
  });
}
```

Figura 7: script de Triple DES

Para realizar las pruebas se utilizo la libreria de CryptoJS, donde se vario el algoritmo de descifrado donde se encuentra destacado en la figura anterior.

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

La directiva @match en l script de Tampermonkey especifica qué páginas web deben coincidir para que el script se ejecute. En este caso, <https://cripto.tiiny.site/> indica que el

2.3 Define función que obtiene automáticamente el password del documento (PARTE 1)

script solo se ejecutará cuando la URL de la página que estás visitando sea exactamente esa.

```
// ==UserScript==  
// @name      Script de Descifrado Avanzado  
// @namespace  http://tampermonkey.net/  
// @version   0.1  
// @description ¡Descifra mensajes ocultos!  
// @author    Tú  
// @match     https://cripto.tiiny.site/  
// @icon      https://www.google.com/s2/favicons?sz=64&domain=tiiny.site  
// @grant     none
```

Figura 8: Comando para ejecutar solo en la pagina

2.3. Define función que obtiene automáticamente el password del documento

La función obtenerClaveCifrado() busca todas las letras mayúsculas en el texto del cuerpo del documento HTML, las concatena y devuelve esta cadena de texto como la clave de cifrado. Si no se encuentran letras mayúsculas, devuelve la cadena "NoDisponible".

```
// Extraer la clave de cifrado de las letras mayúsculas en el texto  
function obtenerClaveCifrado() {  
    const texto = document.body.textContent;  
    const letrasMayusculas = texto.match(/[A-Z]/g) || [];  
    const clave = letrasMayusculas.join('');  
    console.log('Clave del Cifrado corresponde a :', clave);  
    return clave || "NoDisponible";  
}
```

Figura 9: Obtener clave del texto

2.4. Muestra la llave por consola

En la función obtenerClaveCifrado() de la figura 9, tras extraer las letras mayúsculas del texto de la página y formar la clave de cifrado, la función imprime dicha clave en la consola del navegador utilizando console.log.

```
Clave del Cifrado corresponde a : SEGUROSEGUROSEGUROSEGURO
```

Figura 10: Mostrar Clave

3. Desarrollo (Parte 2)

3.1. reconoce automáticamente la cantidad de mensajes cifrados

Una de las funciones de la función `descifrarContenido` es reconocer la cantidad de mensajes, esto lo hace de la siguiente manera. Selecciona todos los elementos del DOM que tienen una clase que contiene la letra "M", muestra la cantidad encontrada en la consola.

```
function descifrarContenido(claveCifrada) {
  const elementosClaseM = document.querySelectorAll('[class*="M"]');
  console.log(`Elementos con clase 'M' encontrados: ${elementosClaseM.length}`);
```

Figura 11: Cantidad de mensajes

3.2. muestra la cantidad de mensajes por consola

la función `descifrarContenido` es la encargada de esto de la siguiente manera. Selecciona todos los elementos del DOM que tienen una clase que contiene la letra "M". Itera sobre cada uno de estos elementos y recupera su ID, que se asume que es el mensaje cifrado. Parsea la clave de cifrado a un formato entendible para la biblioteca CryptoJS. Establece la configuración para el algoritmo de descifrado, especificando el modo ECB (Electronic Codebook). Utiliza la función `decrypt` de CryptoJS con el algoritmo `TripleDES` para descifrar el ID utilizando la clave parseada y la configuración definida. Imprime el ID original y el mensaje descifrado en la consola.

```
// Función para descifrar el contenido
function descifrarContenido(claveCifrada) {
  const elementosClaseM = document.querySelectorAll('[class*="M"]');
  console.log(`Elementos con clase 'M' encontrados: ${elementosClaseM.length}`);

  elementosClaseM.forEach(elemento => {
    const idCifrado = elemento.id;
    const claveEncriptacion = CryptoJS.enc.Utf8.parse(claveCifrada);

    // Configuración para el descifrado
    const configuracion = {
      mode: CryptoJS.mode.ECB
    };

    // Descifrar el ID
    const idDescifrado = CryptoJS.TripleDES.decrypt(idCifrado, claveEncriptacion, configuracion);
    console.log(`ID: ${elemento.id}, el mensaje Descifrado es: ${idDescifrado.toString(CryptoJS.enc.Utf8)}`);
```

Figura 12: Funcion descifrarContenido

el mensaje que muestra por consola se puede apreciar en la figura 6.

4. Desarrollo (Parte 3)

4.1. Importa la librería cryptoJS

Para importar la librería se utilizó la directiva `@require` se utiliza para incluir bibliotecas externas o scripts adicionales en el script de usuario. La URL proporcionada después de `@require` es el enlace directo a la biblioteca CryptoJS.

```
@require https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.2.0/crypto-js.min.js#sha512-a+SU0wWzX0v24Xr1cXHUC689/LJA0N41mPX3g18X0dUKK6Y1DHRRA1v4yd1N40K189TFIdF+qTFKGP0wF0=
```

Figura 13: Importa la librería cryptoJS

4.2. Utiliza SRI en la librería CryptoJS

La parte que sigue al `,` conocida como "Subresource Integrity" (SRI) hash, es un mecanismo de seguridad que permite al navegador verificar que los recursos que se están siendo fetchados (en este caso, la biblioteca CryptoJS) no han sido manipulados.

```
@require https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.2.0/crypto-js.min.js#sha512-a+SU0wWzX0v24Xr1cXHUC689/LJA0N41mPX3g18X0dUKK6Y1DHRRA1v4yd1N40K189TFIdF+qTFKGP0wF0=
```

Figura 14: Subresource Integrity

4.3. Logra decifrar uno de los mensajes

En la Figura 6 se puede apreciar que la función `descifrarContenido` funciona ya que decifro todos los mensajes con Triple DES.

4.4. Imprime todos los mensajes por consola

En la Figura 6 se muestra claramente que todos los mensajes previamente cifrados ahora están en texto plano por la consola, lo que indica que el proceso de descifrado ha sido exitoso. Esto significa que se ha aplicado correctamente Triple DES y la clave de descifrado.

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

Como se puede apreciar la función `descifrarContenido` convierte el resultado del descifrado a una cadena de texto UTF-8 y la coloca dentro de un nuevo elemento HTML `'h2'`. Finalmente, añade el nuevo elemento `'h2'` con el texto descifrado al cuerpo del documento HTML para mostrar el mensaje descifrado en la página web.

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



Figura 15: Texto plano en pagina

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

Para probarla se utilizo la extensión Tampermonkey, se implementó un script con el objetivo de modificar el contenido de una página web con fines de prueba. El propósito de este script era cambiar la informacion de la pagina y así validar su funcionalidad del procedimiento de descifrado de la pagina.

4.7 Indica url al código .js implementado para su validación

```

4 // @version 0.1
5 // @description CryptoJS
6 // @author You
7 // @match https://cripto.tiiny.site
8 // @icon https://www.google.com/s2/favicons?sz=64&domain=tiiny.site
9 // @grant none
10 // @require https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.2.0/crypto-js.min.js#sha512-a+SUDuwNzXDvz4XrIcG4uGf089/1JAoN4lmrXJg18XnduK6YLDHMRlv4yd1N400KI80tFt
11
12 // ==UserScript==
13
14 (function() {
15     'use strict';
16     var nuevo = 'Karla encontró varias ilustraciones, conocidas e inusuales, narrando sagas antiguas. Entonces sin embargo, la joven nunca sabía que cada historia oculta
17     var ids_nuevos = ['M41mGsaFgok=', 'N6BBnoZu/3A=', 'FH4a+A7d+kE=', 'XyshFz99Wxg=', 'knQuopaN97k=', 'ZL8Cf8zF0g8=', 'PLpUajPXyw='];
18
19     // Actualización del párrafo
20     var parrafo = document.querySelector('p');
21     if (parrafo) {
22         parrafo.textContent = nuevo;
23     }
24
25     // Actualización de los IDs de los divs
26     document.querySelectorAll('div[class^="M"]').forEach((div, index) => {
27         if (index < ids_nuevos.length) {
28             div.id = ids_nuevos[index];
29         }
30     });
31
32     // Adición de un nuevo div si es necesario
33     var ultimoDiv = document.querySelector('div[class="M7"]');
34     if (!ultimoDiv) {
35         ultimoDiv = document.createElement('div');
36         ultimoDiv.className = 'M6';
37         ultimoDiv.id = ids_nuevos[ids_nuevos.length - 1];
38         document.body.appendChild(ultimoDiv);
39     }
40
41 })();

```

Figura 16: Scrip para modificar la pagina

Como se puede apreciar funciona con otro texto y otra cantidad de mensajes.

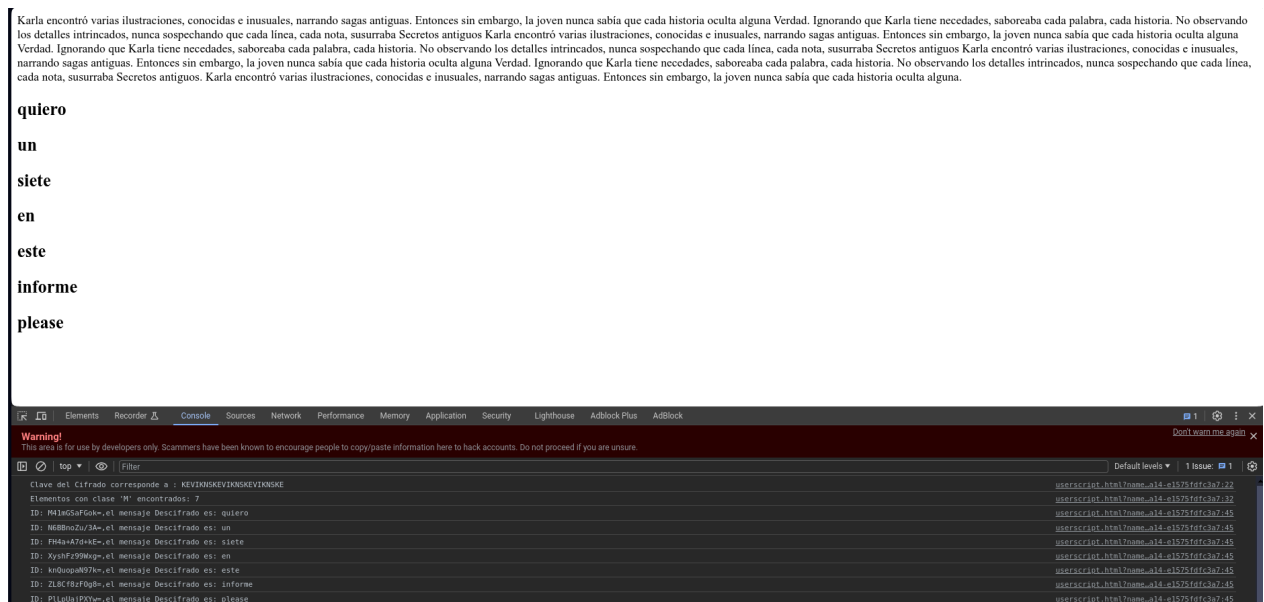


Figura 17: Pagina modificada

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

Toda la documentacion se encuentra en el link de github:https://github.com/IamNejoo/Laboratorio4_criptografia.git

Conclusiones y comentarios

Este informe se detalla con éxito el proceso de detección y descifrado de mensajes cifrados utilizando la biblioteca CryptoJS en un entorno controlado por Tampermonkey. La clave de cifrado se extrajo del texto de una página web y se utilizó para descifrar mensajes codificados en base64 ocultos en los atributos 'id' de elementos 'div'. A través de un enfoque de prueba y error, se estableció que el algoritmo TripleDES era el adecuado para el descifrado, mientras que otros algoritmos no resultaron efectivos. El script fue preciso al ejecutarse solo en la URL específica del informante y demostró su funcionalidad al descifrar y mostrar los mensajes en la página web.