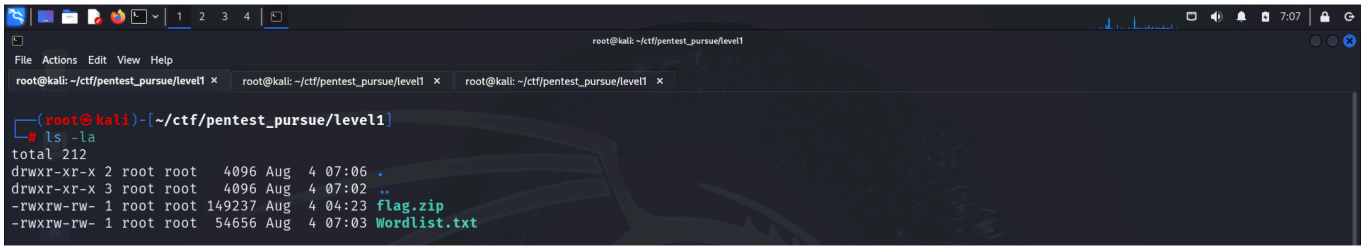
PENTEST PURSUE CTF

> To access the files used in this ctf, click here

We have 2 files:

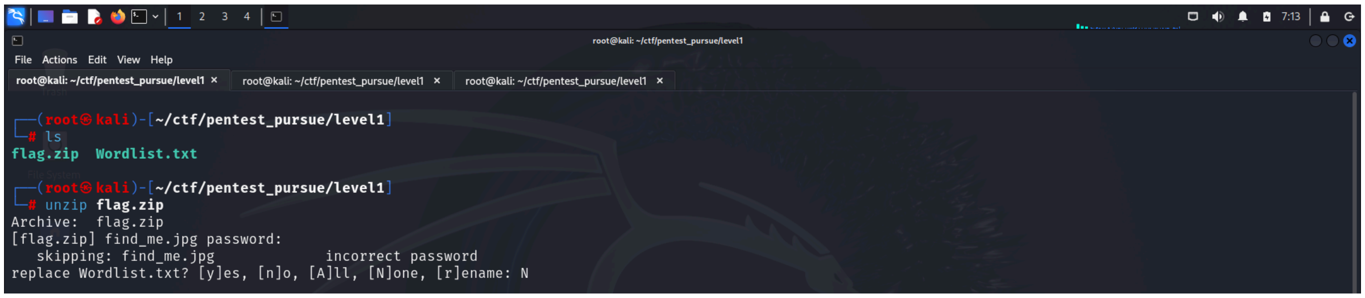
- flag.zip
- wordlist.txt

I downloaded both the files onto my system



```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# ls -la
total 212
drwxr-xr-x 2 root root   4096 Aug  4 07:06 .
drwxr-xr-x 3 root root   4096 Aug  4 07:02 ..
-rwxrw-rw- 1 root root 149237 Aug  4 04:23 flag.zip
-rwxrw-rw- 1 root root  54656 Aug  4 07:03 Wordlist.txt
```

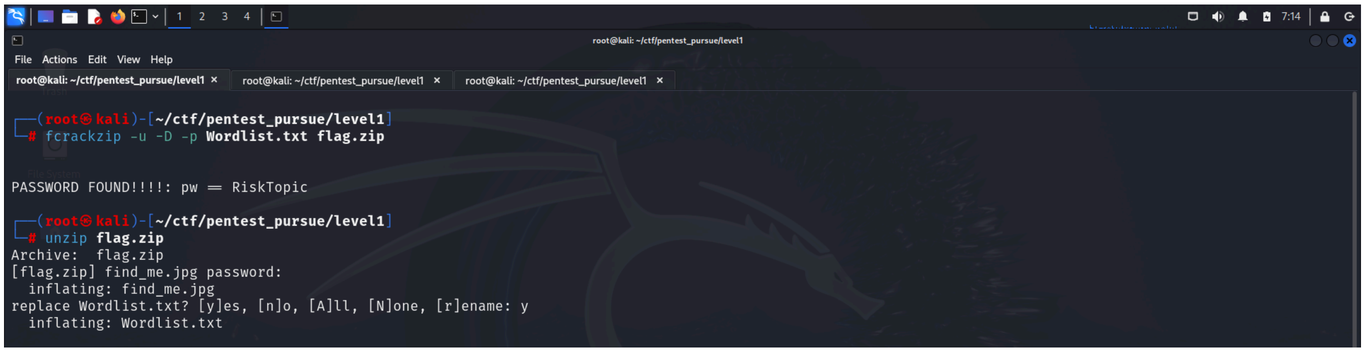
I then tried extracting the contents from the zip.



```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# ls
flag.zip  Wordlist.txt

┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# unzip flag.zip
Archive:  flag.zip
[flag.zip] find_me.jpg password:
   skipping: find_me.jpg             incorrect password
replace Wordlist.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: N
```

However, it required a password. So I used **fcrackzip** to brute force the password

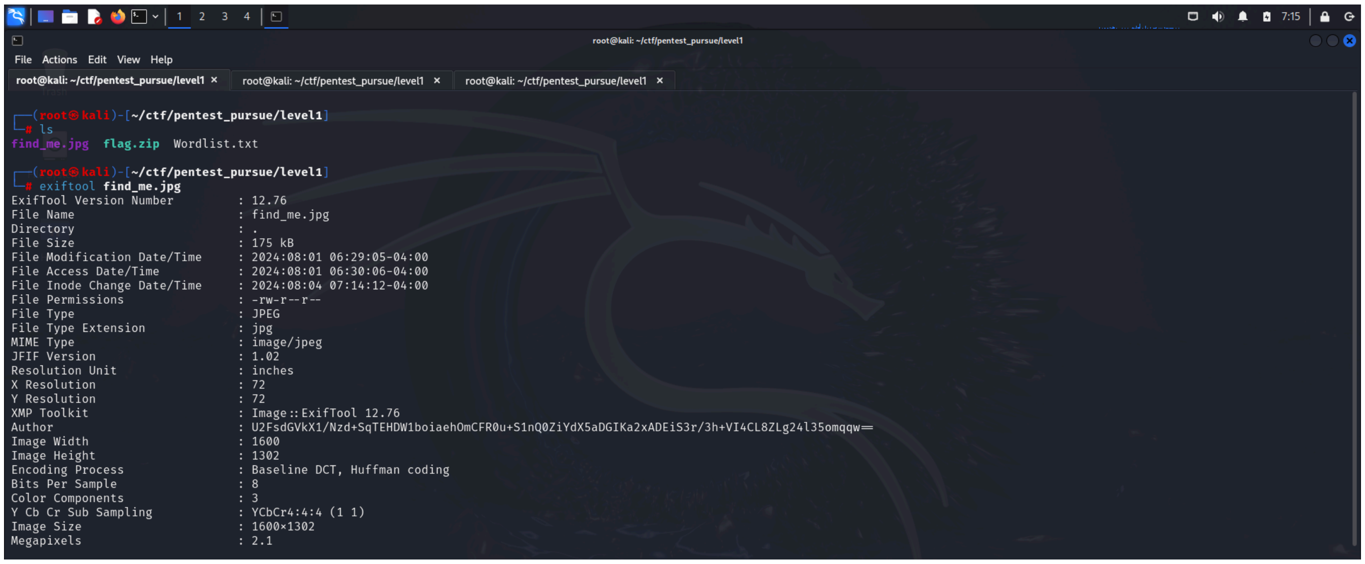


```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# fcrackzip -u -D -p Wordlist.txt flag.zip


PASSWORD FOUND!!!!: pw == RiskTopic

┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# unzip flag.zip
Archive:  flag.zip
[flag.zip] find_me.jpg password:
  inflating: find_me.jpg
replace Wordlist.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: Wordlist.txt
```

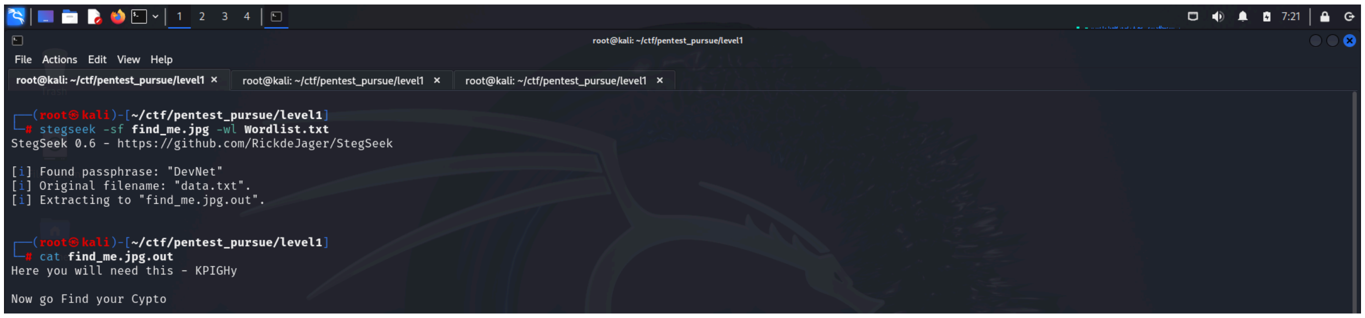
I then viewed the image meta data and found an encrypted author name.



```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# ls
find_me.jpg  flag.zip  Wordlist.txt

┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# exiftool find_me.jpg
ExifTool Version Number         : 12.76
File Name                       : find_me.jpg
Directory                       : .
File Size                       : 175 kB
File Modification Date/Time     : 2024:08:01 06:29:05-04:00
File Access Date/Time           : 2024:08:01 06:30:06-04:00
File Inode Change Date/Time     : 2024:08:04 07:14:12-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
XMP Toolkit                     : Image::ExifTool 12.76
Author                          : U2FsdGVkX1/Nzd+SqTEHDW1boiaehOmCFR0u+S1nQ0ZiYdX5aDGIKa2xADEiS3r/3h+VI4CL8ZLg24l35omqqw==
Image Width                     : 1600
Image Height                    : 1302
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 1600×1302
Megapixels                      : 2.1
```

The image itself could also contain some data so I used **stegseek** to crack the password and extract any data present inside.



```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# stegseek -sf find_me.jpg -wl Wordlist.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "DevNet"
[i] Original filename: "data.txt".
[i] Extracting to "find_me.jpg.out".


┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# cat find_me.jpg.out
Here you will need this - KPIGHy

Now go Find your Cypto
```

The `Now go Find your Crypto` suggested that there was some sort of cryptography for which, the given key could be used.

I quickly copied the `AUTHOR` information from the **exiftool** output and asked **chatgpt** about it. Through it I found out that the text was encrpyted using **openssl**'s **aes-256-cbc** algorithm.

Chatgpt's response

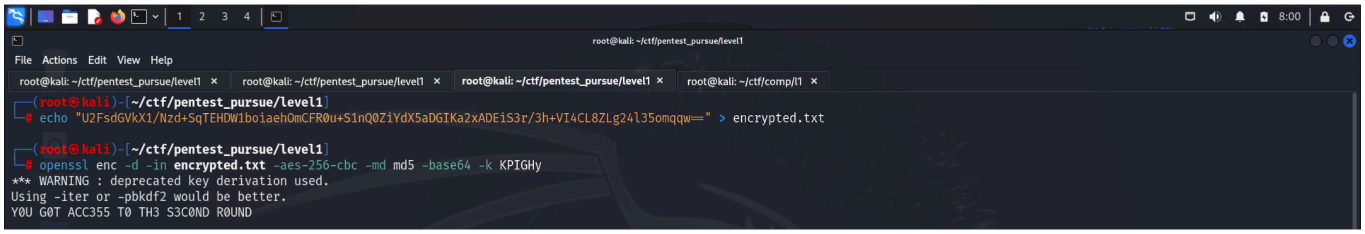
This appears to be a string that has been encrypted using a method that outputs Base64-encoded ciphertext. The string `U2FsdGVkX1/` suggests that it might have been encrypted using OpenSSL with the AES (Advanced Encryption Standard) algorithm, as OpenSSL uses this prefix for encrypted data.

Hence these are the things I knew about the data:

- It could be decrypted using **openssl**
- It was encrypted using **aes-256-cbc**
- It was encoded using **base64**

- The key could be made using the password found inside **data.txt**

Now the only thing I had to figure out is the digest used for the key for decrypting it. The key derivation process converts a password into an encryption key suitable for the specified encryption algorithm. This process ensures that the same password consistently produces the same key, which is necessary for successful encryption and decryption. Hence for this, I try different digests like **md5** (most common), **sha1**, **sha256**, **sha512**

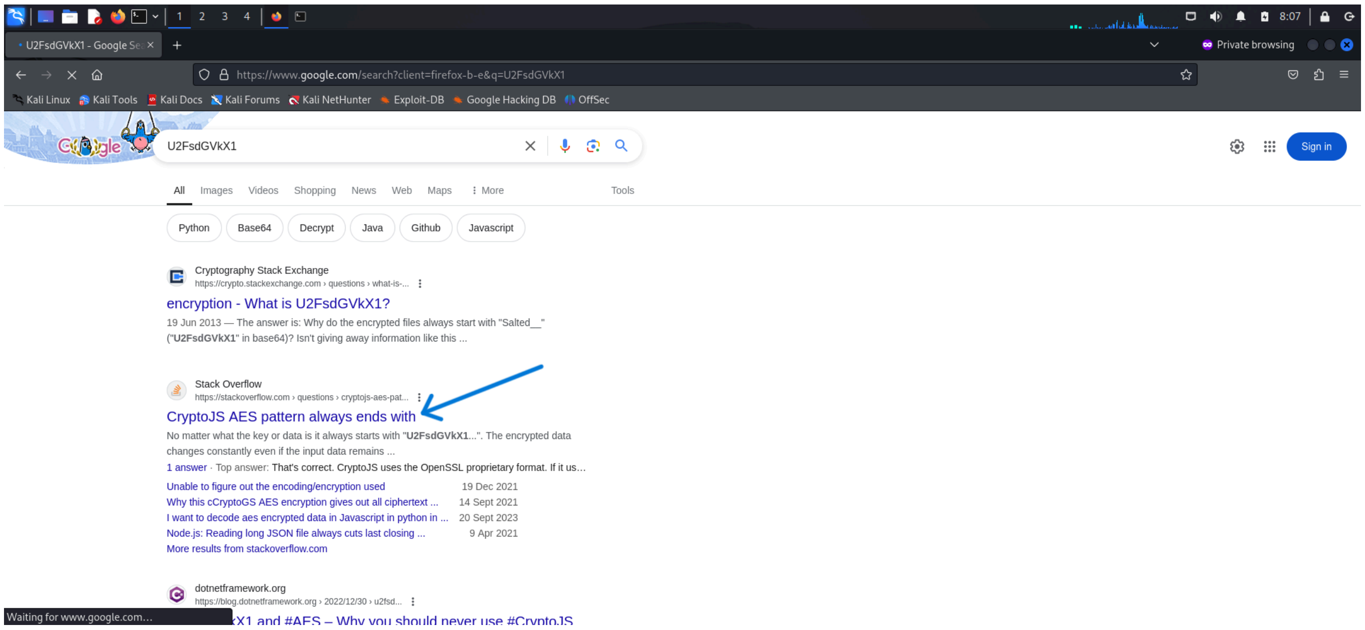


```
┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# echo "U2FsdGVkX1/Nzd+SqTEHDW1boiaehOmCFR0u+S1nQ0ZiYdX5aDGIKa2xADEiS3r/3h+VI4CL8ZLg24l35omqqw==" > encrypted.txt

┌──(root㉿kali)-[~/ctf/pentest_pursue/level1]
└─# openssl enc -d -in encrypted.txt -aes-256-cbc -md md5 -base64 -k KPIGHy
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Y0U G0T ACC355 T0 TH3 S3C0ND R0UND
```

And with this I was successfully able to clear the first round ;)

PS: If you are not comfortable with using command line tools to decrypt the key, you can use the encryption identifier from the encrypted text to google relevant sites.



https://www.google.com/search?client=firefox-b-e&q=U2FsdGVkX1

U2FsdGVkX1

Cryptography Stack Exchange
https://crypto.stackexchange.com › questions › what-is-...
encryption - What is U2FsdGVkX1?
19 Jun 2013 — The answer is: Why do the encrypted files always start with "Salted__" ("**U2FsdGVkX1**" in base64)? Isn't giving away information like this ...

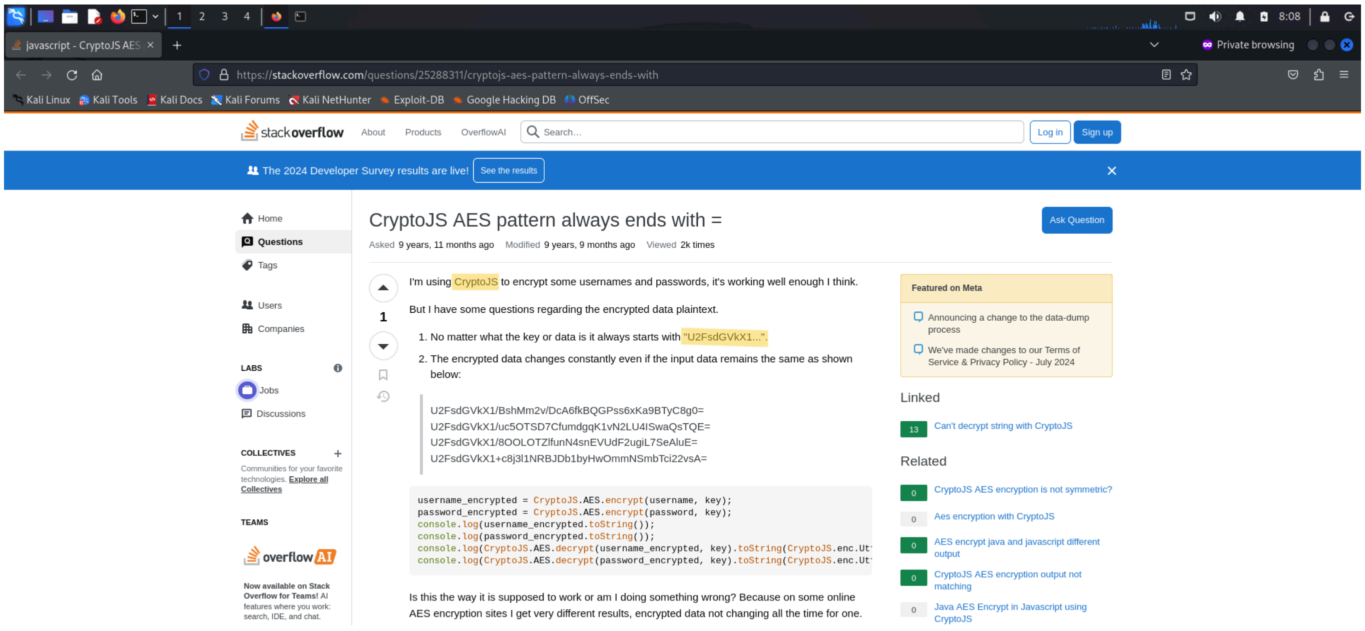
Stack Overflow
https://stackoverflow.com › questions › cryptojs-aes-pat...
CryptoJS AES pattern always ends with
No matter what the key or data is it always starts with "**U2FsdGVkX1**...". The encrypted data changes constantly even if the input data remains ...
1 answer · Top answer: That's correct. CryptoJS uses the OpenSSL proprietary format. If it us...

Unable to figure out the encoding/encryption used	19 Dec 2021
Why this cCryptoGS AES encryption gives out all ciphertext ...	14 Sept 2021
I want to decode aes encrypted data in Javascript in python in ...	20 Sept 2023
Node.js: Reading long JSON file always cuts last closing ...	9 Apr 2021

More results from stackoverflow.com

dotnetframework.org
https://blog.dotnetframework.org › 2022/12/30 › u2fsd...
kX1 and #AES – Why you should never use #CryptoJS

Waiting for www.google.com...


javascript - CryptoJS AES

https://stackoverflow.com/questions/25288311/cryptojs-aes-pattern-always-ends-with

Kali Linux · Kali Tools · Kali Docs · Kali Forums · Kali NetHunter · Exploit-DB · Google Hacking DB · OffSec

The 2024 Developer Survey results are live! See the results

CryptoJS AES pattern always ends with =

Asked 9 years, 11 months ago · Modified 9 years, 9 months ago · Viewed 2k times

I'm using CryptoJS to encrypt some usernames and passwords, it's working well enough I think.

But I have some questions regarding the encrypted data plaintext.

1. No matter what the key or data is it always starts with "U2FsdGVkX1...".
2. The encrypted data changes constantly even if the input data remains the same as shown below:

> U2FsdGVkX1/BshMm2v/DcA6fkBQGPss6xKa9BTyC8g0=
> U2FsdGVkX1/uc5OTSD7CfumdgqK1vN2LU4ISwaQsTQE=
> U2FsdGVkX1/8OOLOTZlfunN4snEVUdF2ugiL7SeAluE=
> U2FsdGVkX1+c8j3l1NRBJDb1byHwOmmNSmbTci22vsA=

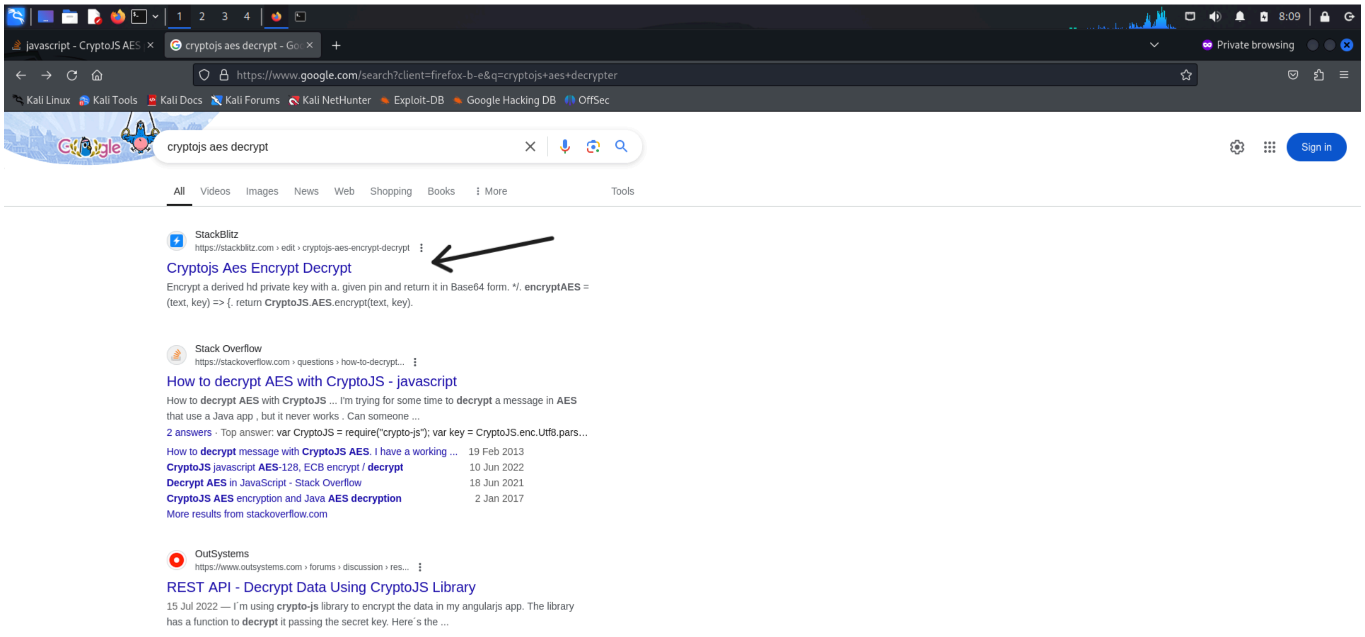
```
username_encrypted = CryptoJS.AES.encrypt(username, key);
password_encrypted = CryptoJS.AES.encrypt(password, key);
console.log(username_encrypted.toString());
console.log(password_encrypted.toString());
console.log(CryptoJS.AES.decrypt(username_encrypted, key).toString(CryptoJS.enc.Ut
console.log(CryptoJS.AES.decrypt(password_encrypted, key).toString(CryptoJS.enc.Ut
```

Is this the way it is supposed to work or am I doing something wrong? Because on some online AES encryption sites I get very different results, encrypted data not changing all the time for one.

Featured on Meta
- Announcing a change to the data-dump process
- We've made changes to our Terms of Service & Privacy Policy - July 2024

Linked
- 13 Can't decrypt string with CryptoJS

Related
- 0 CryptoJS AES encryption is not symmetric?
- 0 Aes encryption with CryptoJS
- 0 AES encrypt java and javascript different output
- 0 CryptoJS AES encryption output not matching
- 0 Java AES Encrypt in Javascript using CryptoJS


javascript - CryptoJS AES · cryptojs aes decrypt - Go

https://www.google.com/search?client=firefox-b-e&q=cryptojs+aes+decrypter

cryptojs aes decrypt

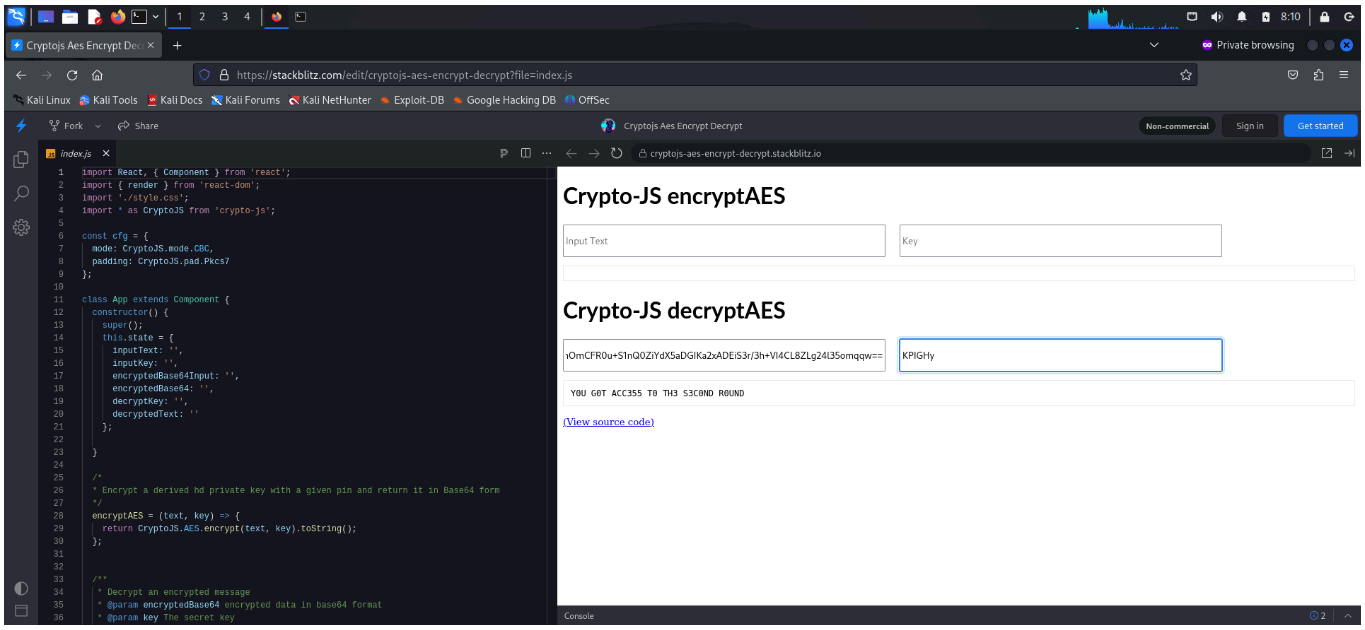
StackBlitz
https://stackblitz.com › edit › cryptojs-aes-encrypt-decrypt
Cryptojs Aes Encrypt Decrypt
Encrypt a derived hd private key with a. given pin and return it in Base64 form. */. encryptAES = (text, key) => {. return CryptoJS.AES.encrypt(text, key).

Stack Overflow
https://stackoverflow.com › questions › how-to-decrypt...
How to decrypt AES with CryptoJS - javascript
How to decrypt AES with CryptoJS ... I'm trying for some time to decrypt a message in AES that use a Java app , but it never works . Can someone ...
2 answers · Top answer: var CryptoJS = require("crypto-js"); var key = CryptoJS.enc.Utf8.pars...

How to decrypt message with CryptoJS AES. I have a working ...	19 Feb 2013
CryptoJS javascript AES-128, ECB encrypt / decrypt	10 Jun 2022
Decrypt AES in JavaScript - Stack Overflow	18 Jun 2021
CryptoJS AES encryption and Java AES decryption	2 Jan 2017

More results from stackoverflow.com

OutSystems
https://www.outsystems.com › forums › discussion › res...
REST API - Decrypt Data Using CryptoJS Library
15 Jul 2022 — I´m using crypto-js library to encrypt the data in my angularjs app. The library has a function to decrypt it passing the secret key. Here´s the ...



```
import React, { Component } from 'react';
import { render } from 'react-dom';
import './style.css';
import * as CryptoJS from 'crypto-js';

const cfg = {
  mode: CryptoJS.mode.CBC,
  padding: CryptoJS.pad.Pkcs7
};

class App extends Component {
  constructor() {
    super();
    this.state = {
      inputText: '',
      inputKey: '',
      encryptedBase64Input: '',
      encryptedBase64: '',
      decryptKey: '',
      decryptedText: ''
    };

  }

  /*
  * Encrypt a derived hd private key with a given pin and return it in Base64 form
  */
  encryptAES = (text, key) => {
    return CryptoJS.AES.encrypt(text, key).toString();
  };


  /**
   * Decrypt an encrypted message
   * @param encryptedBase64 encrypted data in base64 format
   * @param key The secret key
```

Crypto-JS encryptAES

Input Text | Key

Crypto-JS decryptAES

ıOmCFR0u+S1nQ0ZiYdX5aDGIKa2xADEiS3r/3h+VI4CL8ZLg24l35omqqw== | KPIGHy

```
Y0U G0T ACC355 T0 TH3 S3C0ND R0UND
```

(View source code)