Cryptography - Secrets Behind a Letter

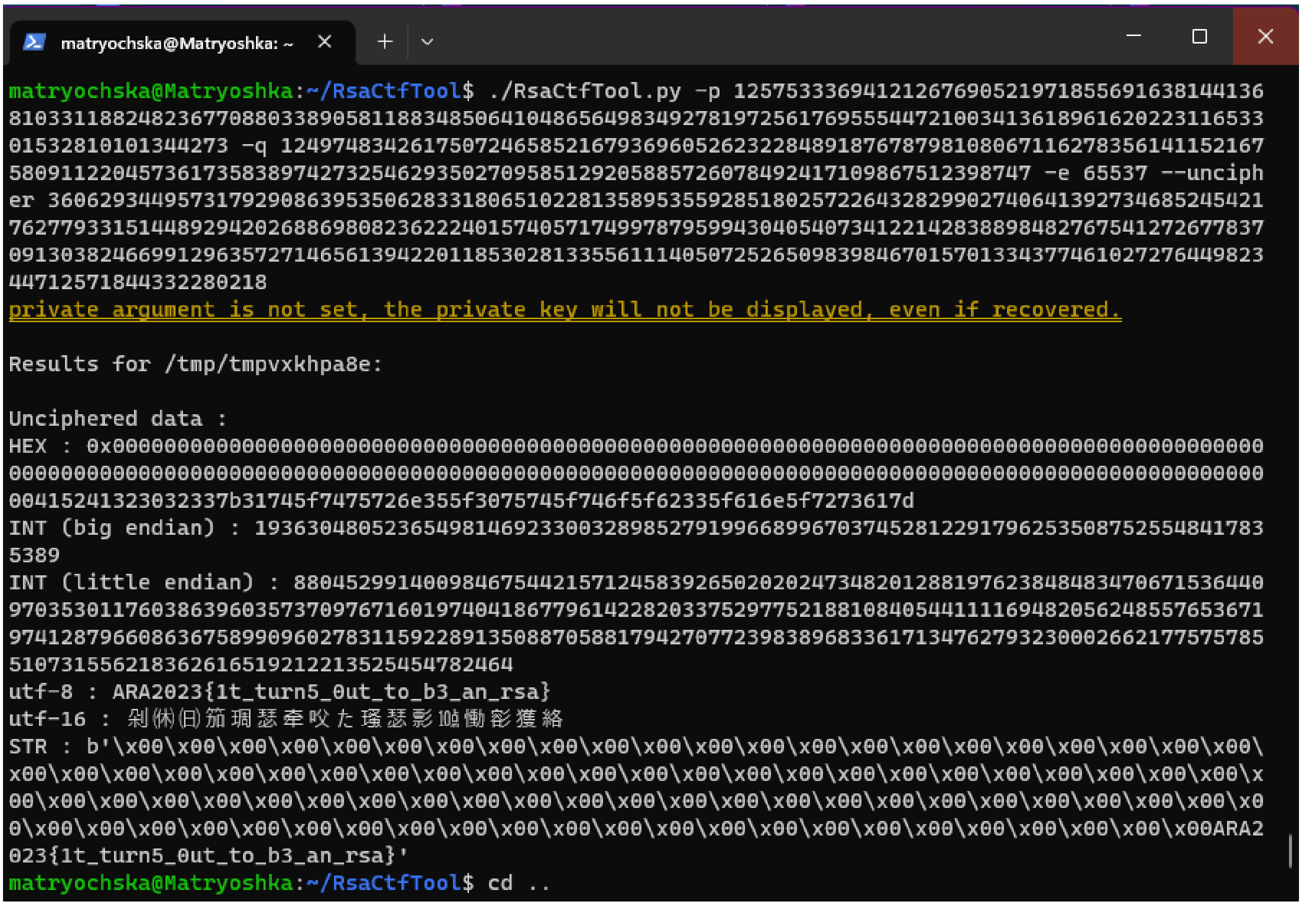
In this challenge, we were given a Letter attached below:

```
p:
12575333694121267690521971855691638144136810331188248236770880338905811883485064104865649834927
q:
12497483426175072465852167936960526232284891876787981080671162783561411521675809112204573617358
c:
36062934495731792908639535062833180651022813589535592851802572264328299027406413927346852454217

e = 65537
```

After doing some research, I can confirmed that this challenge is about RSA so I do searching with a powerful keyword and it brought me to this stack-overflow discussion.

Based on that discussion, the top answer mentioned a tools named `RsaCtfTool` in GitHub so I clone the repository and use it to solve this challenge.

Based of their documentation, I used a command `--uncipher` , here is the result:



```
matryochska@Matryoshka:~/RsaCtfTool$ ./RsaCtfTool.py -p 12575333694121267690521971855691638144136
8103311882482367708803389058118834850641048656498349278197256176955544721003413618961620223116533
01532810101344273 -q 1249748342617507246585216793696052623228489187678798108067116278356141152167
5809112204573617358389742732546293502709585129205885726078492417109867512398747 -e 65537 --unciph
er 3606293449573179290863953506283318065102281358953559285180257226432829902740641392734685245421
7627793315144892942026886980823622240157405717499787959943040540734122142838898482767541272677837
0913038246699129635727146561394220118530281335561114050725265098398467015701334377461027276449823
44712571844332280218
private argument is not set, the private key will not be displayed, even if recovered.

Results for /tmp/tmpvxkhpa8e:

Unciphered data :
HEX : 0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00415241323032337b31745f7475726e355f3075745f746f5f62335f616e5f7273617d
INT (big endian) : 193630480523654981469233003289852791996689967037452812291796253508752554841783
5389
INT (little endian) : 880452991400984675442157124583926502020247348201288197623848483470671536440
9703530117603863960357370976716019740418677961422820337529775218810840544111169482056248557653671
9741287966086367589909602783115922891350887058817942707723983896833617134762793230002662177575785
510731556218362616519212213525454782464
utf-8 : ARA2023{1t_turn5_0ut_to_b3_an_rsa}
utf-16 : 剁㈱㈰筲瑟獵湲張畴瑟彯㍢慟彮獲絡
STR : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00ARA2
023{1t_turn5_0ut_to_b3_an_rsa}'
matryochska@Matryoshka:~/RsaCtfTool$ cd ..
```

So based on this unciphered data, I found that the flag was:

```
ARA2023{it_turn5_0ut_to_b3_an_rsa}
```

Give feedback

main ctf-writeup / 2023-ARA / Cryptography-Help /

elshiraphine style: add line break … 2 minutes ago History

..

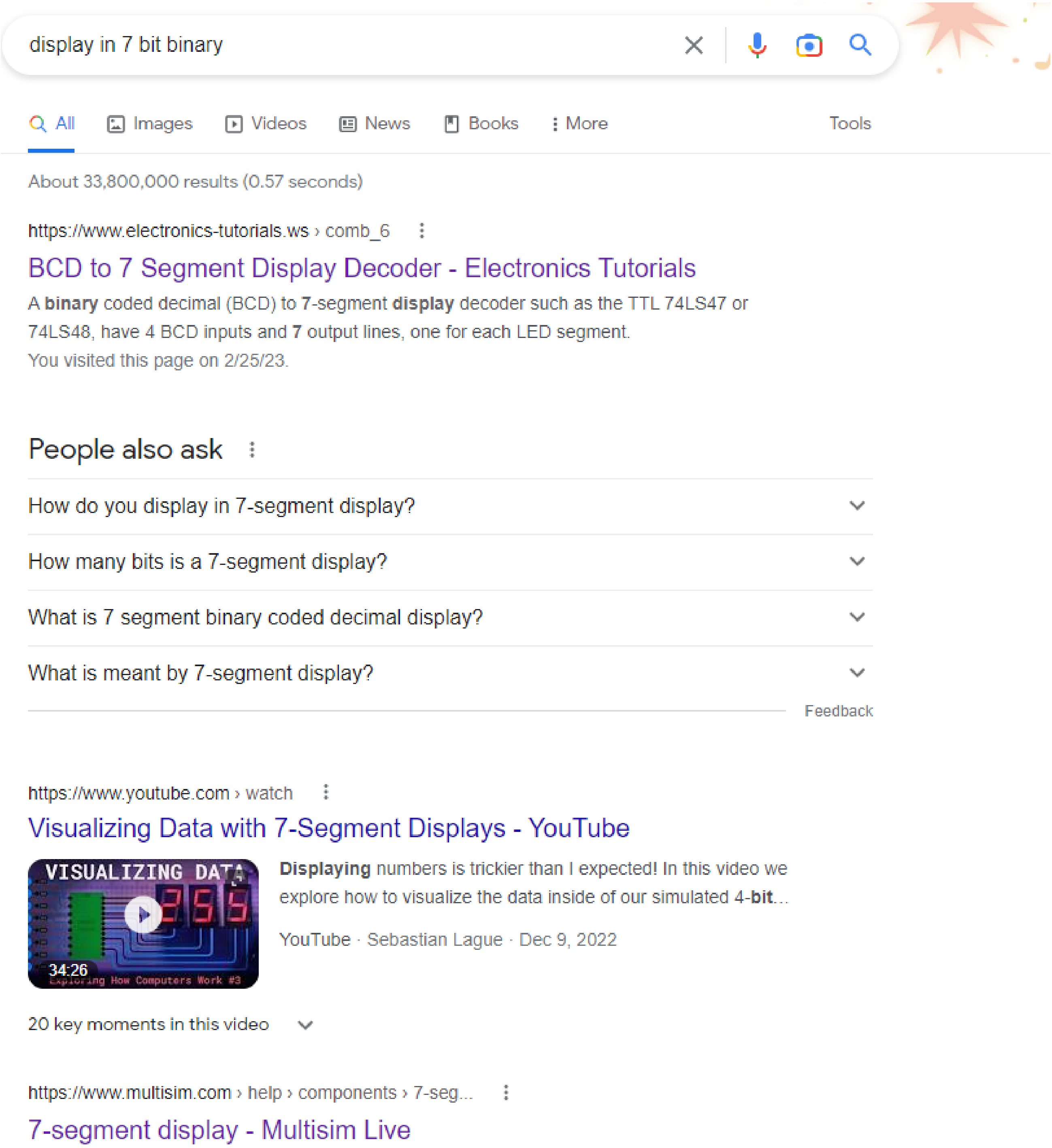
assets	7 minutes ago
README.md	2 minutes ago

README.md

Cryptography - Help

In this challenge we were given 32 number of 7-bit binary number. I was confused because there is no clue about this challenge. But after and hours, I found something interesting from the challenge's statement which is `display in the office`.

So, I started googling about `display in 7 bit binary` and the result was suprisingly good.



About 33,800,000 results (0.57 seconds)

https://www.electronics-tutorials.ws › comb_6

BCD to 7 Segment Display Decoder - Electronics Tutorials

A **binary** coded decimal (BCD) to **7**-segment **display** decoder such as the TTL 74LS47 or 74LS48, have 4 BCD inputs and **7** output lines, one for each LED segment.

You visited this page on 2/25/23.

People also ask

How do you display in 7-segment display?

How many bits is a 7-segment display?

What is 7 segment binary coded decimal display?

What is meant by 7-segment display?

Feedback

https://www.youtube.com › watch

Visualizing Data with 7-Segment Displays - YouTube

Displaying numbers is trickier than I expected! In this video we explore how to visualize the data inside of our simulated 4-**bit**...

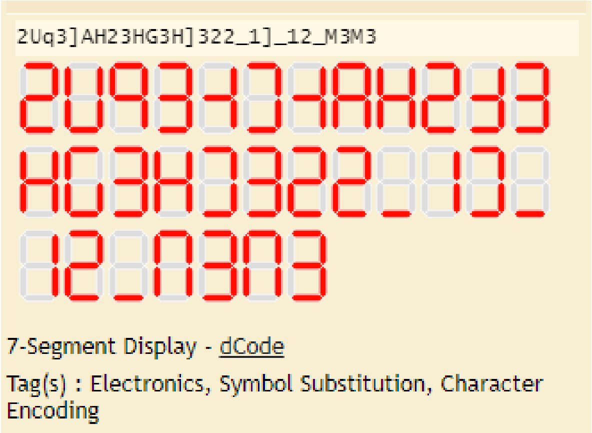
YouTube · Sebastian Lague · Dec 9, 2022

20 key moments in this video

https://www.multisim.com › help › components › 7-seg...

7-segment display - Multisim Live

I tried to find a tools to help me mapping this binary number into seven segment display and brought me to this site. The result is not really good because there is some inconsistency between the number and there are some symbols in the result.



```
2Uq3]AH23HG3H]322_1]_12_M3M3
```

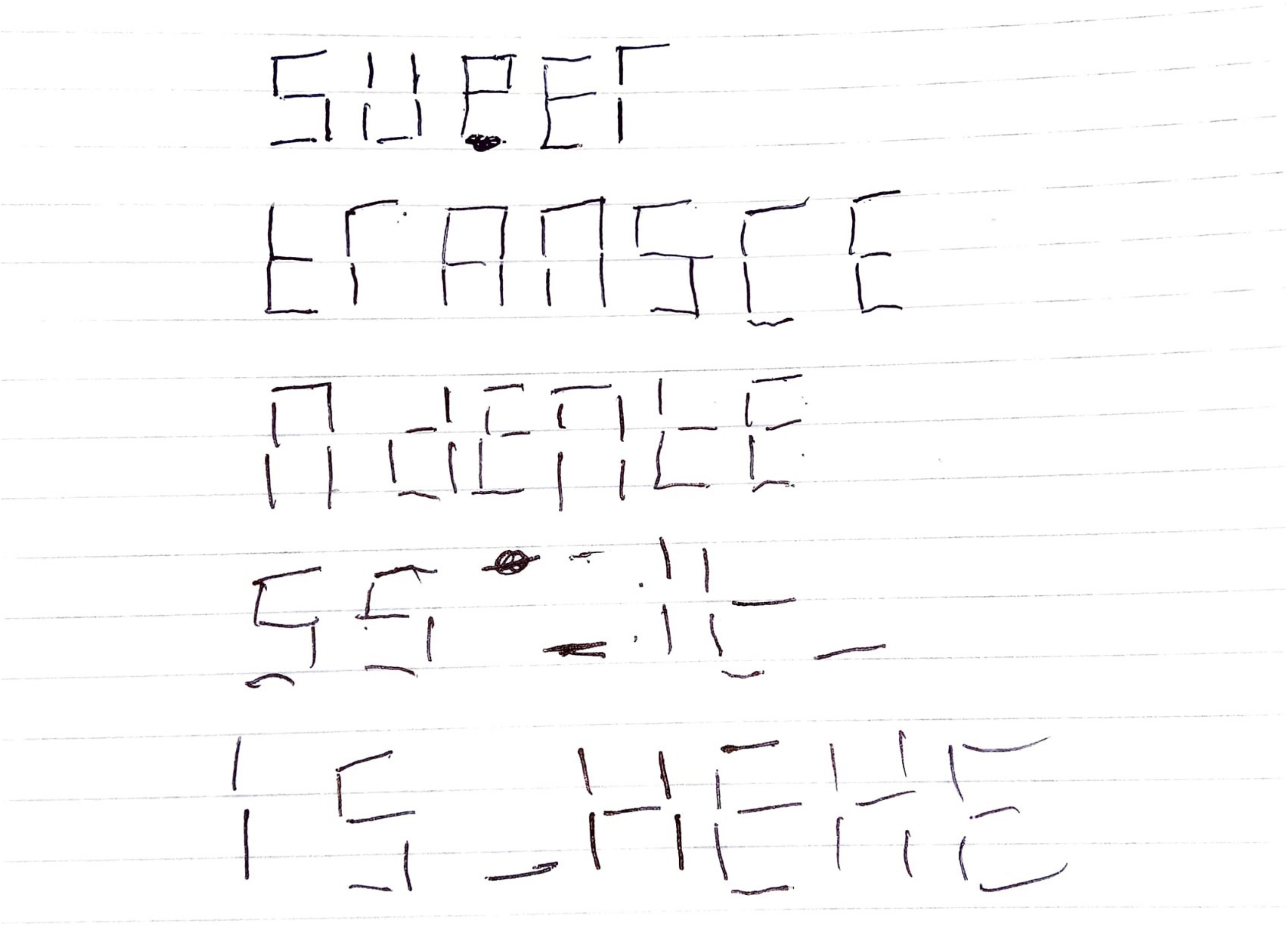
7-Segment Display - dCode

Tag(s) : Electronics, Symbol Substitution, Character Encoding

.

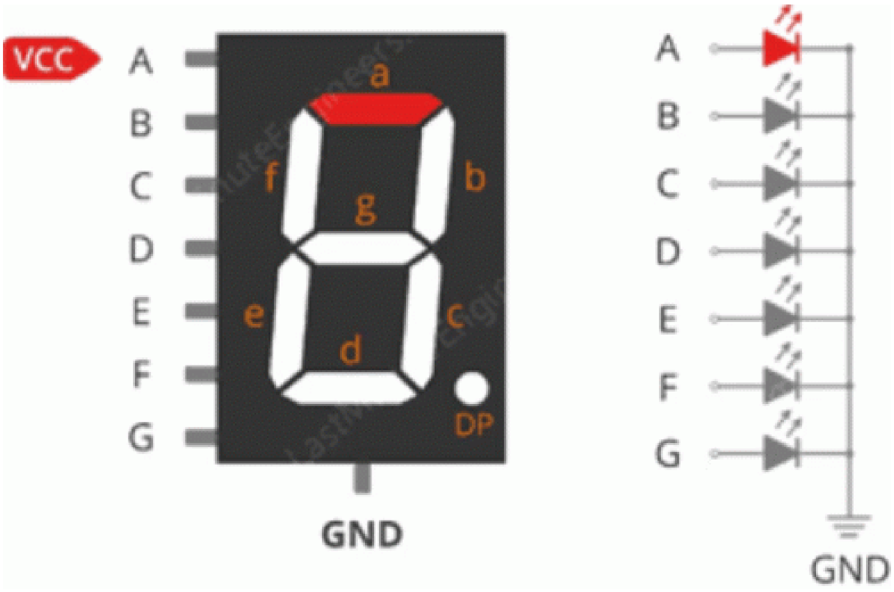
I tried to find another possibilities such as lower-case display in seven segment but confusing because

it is not common.
After quite tired of searching, I tried manually by writing in my book and it is mindblowing as attached below:



SO THE FLAG SHOULD BE

```
ARA2023{supertranscendentess_it_is_hehe}
```



elshiraphine / ctf-writeup Private

Code Issues Pull requests Actions Projects Wiki Security Insights Se

main ctf-writeup / 2023-ARA / Cryptography-babychall /

elshiraphine feat: add Cryptography-babychall write up ... 7 minutes ago History

..

assets	7 minutes ago
README.md	7 minutes ago

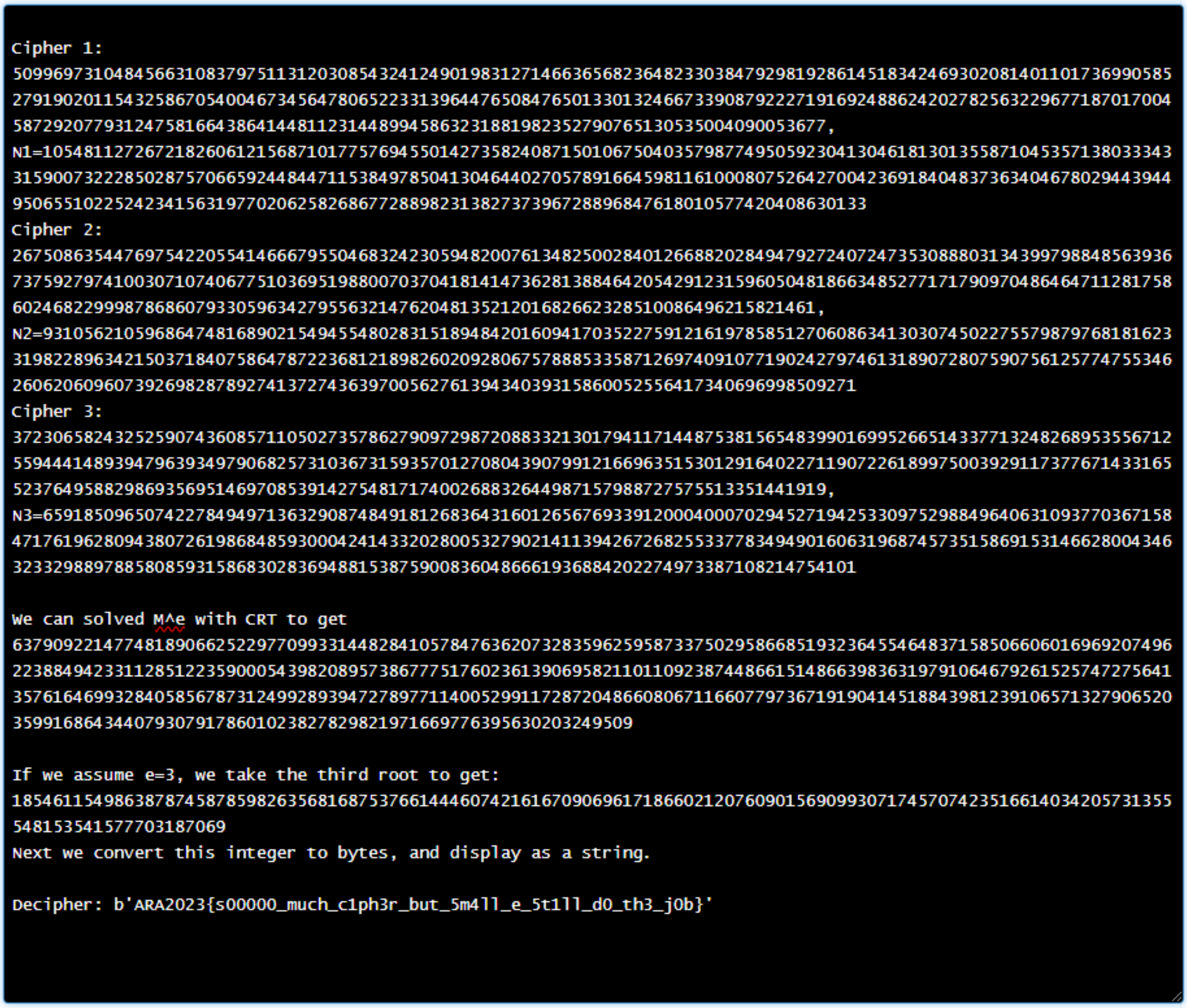
README.md

Cryptography - babychall

Babychall challenge is similar to the previous crypto challenge (Secret Behind a Letter) but in this challenge we were not given any information about the key (the `e` value).
We were only given pairs of number:

```
c1=50996973104845663108379751131203085432412490198312714663656823648233038479298192861451834246
c2=26750863544769754220554146667955046832423059482007613482500284012668820284947927240724735308
c3=37230658243252590743608571105027357862790972987208833213017941171448753815654839901699526651

n1=10548112726721826061215687101775769455014273582408715010675040357987749505923041304618130135
n2=93105621059686474816890215494554802831518948420160941703522759121619785851270608634130307450
n3=65918509650742278494971363290874849181268364316012656769339120004000702945271942533097529884
```

So I expected that the key maybe a common key (it should be `e = 3`). To solve this challenge, I used `Chinese Remainder Theory` and found a tools on the internet to solve the Cipher.
And here is the result:



```
Cipher 1:
50996973104845663108379751131203085432412490198312714663656823648233038479298192861451834246930208140110173699058527919020115432586705400467345647806522331396447650847650133013246673390879222719169248862420278256322967718701700458729207793124758166438641448112314489945863231881982352790765130535004090053677,
N1=105481127267218260612156871017757694550142735824087150106750403579877495059230413046181301355871045357138033343315900732228502875706659244844711538497850413046440270578916645981161000807526427004236918404837363404678029443944950655102252423415631977020625826867728898231382737396728896847618010577420408630133
Cipher 2:
2675086354476975422055414666795504683242305948200761348250028401266882028494792724072473530888031343997988485639367375927974100307107406775103695198800703704181414736281388464205429123159605048186634852771717909704864647112817586024682299987868607933059634279556321476204813521201682662328510086496215821461,
N2=93105621059686474816890215494554802831518948420160941703522759121619785851270608634130307450227557987976818162331982289634215037184075864787223681218982602092806757888533587126974091077190242797461318907280759075612577475534626062060960739269828789274137274363970056276139434039315860052556417340696998509271
Cipher 3:
37230658243252590743608571105027357862790972987208833213017941171448753815654839901699526651433771324826895355671255944414893947963934979068257310367315935701270804390799121669635153012916402271190722618997500392911737767143316552376495882986935695146970853914275481717400268832644987157988727575513351441919,
N3=65918509650742278494971363290874849181268364316012656769339120004000702945271942533097529884964063109377036715847176196280943807261986848593000424143320280053279021411394267268255337783494901606319687457351586915314662800434632332988978858085931586830283694881538759008360486661936884202274973387108214754101

We can solved M^e with CRT to get
6379092214774818906625229770993314482841057847636207328359625958733750295866851932364554648371585066060169692074962238849423311285122359000543982089573867775176023613906958211011092387448661514866398363197910646792615257472756413576164699328405856787312499289394727897711400529911728720486608067116607797367191904145188439812391065713279065203599168643440793079178601023827829821971669776395630203249509

If we assume e=3, we take the third root to get:
185461154986387874587859826356816875376614446074216167090696171866021207609015690993071745707423516614034205731355548153541577703187069
Next we convert this integer to bytes, and display as a string.

Decipher: b'ARA2023{s00000_much_c1ph3r_but_5m4ll_e_5t1ll_d0_th3_j0b}'
```

Or you can access on this link.

So based on the result, I found that the flag was:

```
ARA2023{s00000_much_c1ph3r_but_5m4ll_e_5t1ll_d0_th3_j0b}
```

Give feedback

elshiraphine / ctf-writeup Private

<> Code ⊙ Issues Pull requests Actions Projects Wiki Security Insights Se

main ctf-writeup / 2023-ARA / OSINT-Hey-detective-can-you-help-me / ...

elshiraphine style: fix line break, add assets folder, and fix typo ... 2 hours ago History

..

assets	2 hours ago
README.md	2 hours ago

README.md

OSINT - Hey detective, can you help me

In this challenge we were given some informations to find a cosplayer from China who like to post their photos on Facebook and Instagram. There were several instructions given, first I tried to find cosplayer who had collaborated with Sakura (also from china).

On instagram I found a cosplayer account named Sakura with the username sakura.gun. Here is the profile sakura.gun on instagram and looked for accounts that had collaborated with her.

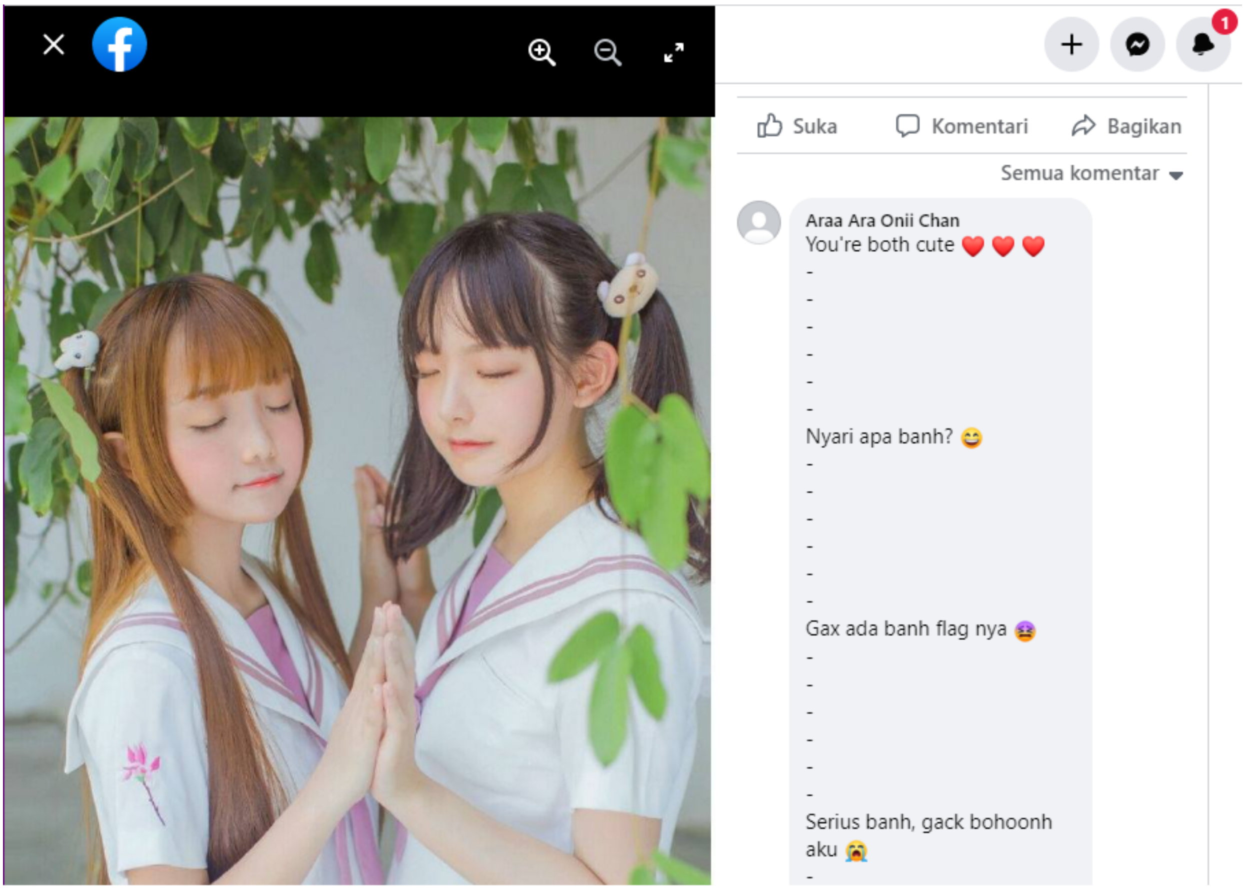
1. First, I found a cosplayer named rakuko but it didn't match the next clue which is **studying in top university in China** because rakuko herself was studying abroad in America.
2. Second, I found another collaboration with account named skylaryuuu, but I thought it was not her that I tried to search because skylaryuuu currently move to Canada.
3. Third, I found an ordinary photo (not doing cosplay) with an account named Yanzikenko. I am interested with this account so I tried to find her page in Facebook.

It turned out that what I did was right because I found several matching photos with the clues given in the challenge. Here is her facebook page.

Based on her photo in Facebook, I tried to follow the instructions as below:

1. Social Media ID
   Based on the Flag format, it should be an Instagram ID so I use this Find Instagram User ID Tools.
   And the value was `44793134117`
2. Her university but in abbreviation.
   So, based on this photo, she graduated from Beijing Normal University. So the second flag pattern should be `BNU` .
3. Mascot name where she was in the doll shop.
   There is a post that should include her photo with a mascot in the doll store She took a picture with a mascot named `Molly` from PopMart.

4. Date and Time when she posted a photo in the bookstore.
   Previously, an instruction mentioned that she was photographed sitting in a bookstore. So, there are two different post which have a photo that it is could be taken in the bookstore.
   First post and the second post.
   So it could be `3Juni2019-10:25` or `14Februari2019-14:59` .
5. The last part, which is the most challenging part is to find redacted flag. There are so many photo which have Kenko and Sakura collaborated. So after an hours read, I found a powerful comments attached below:

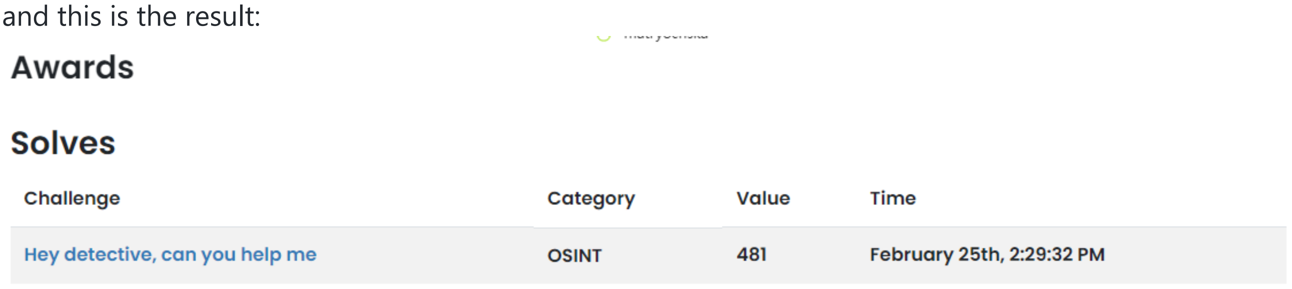


Araa Ara Onii Chan
You're both cute ❤️❤️❤️
-
-
-
-
-
-
Nyari apa banh? 😄
-
-
-
-
-
-
Gax ada banh flag nya 😫
-
-
-
-
-
-
Serius banh, gack bohoonh aku 😭
-

AND FINALLY AFTER TRYING SEVERAL POSSIBILITIES, I FOUND THAT THE FLAG WAS:

```
ARA2023{44793134117_BNU_Molly_3Juni2019-10:25_Y0u4r3ThE0s1nTm45t3R}
```

and this is the result:



Awards

Solves

Challenge	Category	Value	Time
Hey detective, can you help me	OSINT	481	February 25th, 2:29:32 PM

Give feedback

elshiraphine style: add line break ... 1 minute ago History

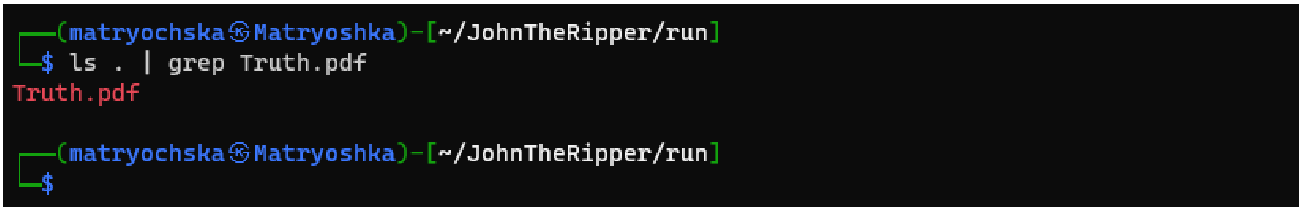
..

assets	6 minutes ago
README.md	1 minute ago
isUpper.py	6 minutes ago

README.md

Misc - Truth

We were given a locked PDF File but there is no information about the password. So I search using `brute force PDF Password` keyword on google and found this website.
Next, I followed the instruction to clone, make and configure the tools on my Kali Linux WSL. Then, I copied the PDF file to `run` directory.

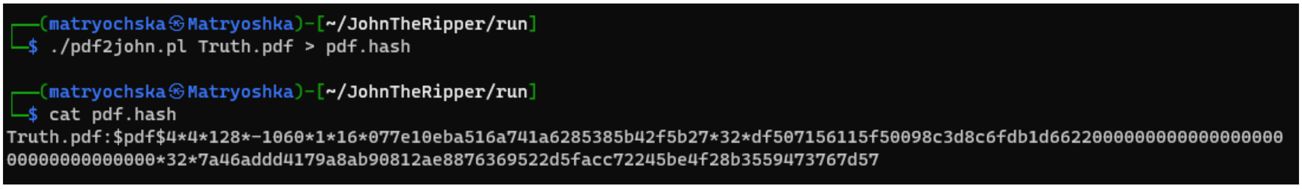


```
┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$ ls . | grep Truth.pdf
Truth.pdf

┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$
```

The next step is use this command to generate PDF hash file:

```
pdf2john.pl Truth.pdf > pdf.hash
```

The tools generated a PDF hash file below:



```
┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$ ./pdf2john.pl Truth.pdf > pdf.hash

┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$ cat pdf.hash
Truth.pdf:$pdf$4*4*128*-1060*1*16*077e10eba516a741a6285385b42f5b27*32*df507156115f50098c3d8c6fdb1d66220000000000000000000
0000000000000*32*7a46addd4179a8ab90812ae8876369522d5facc72245be4f28b3559473767d57
```

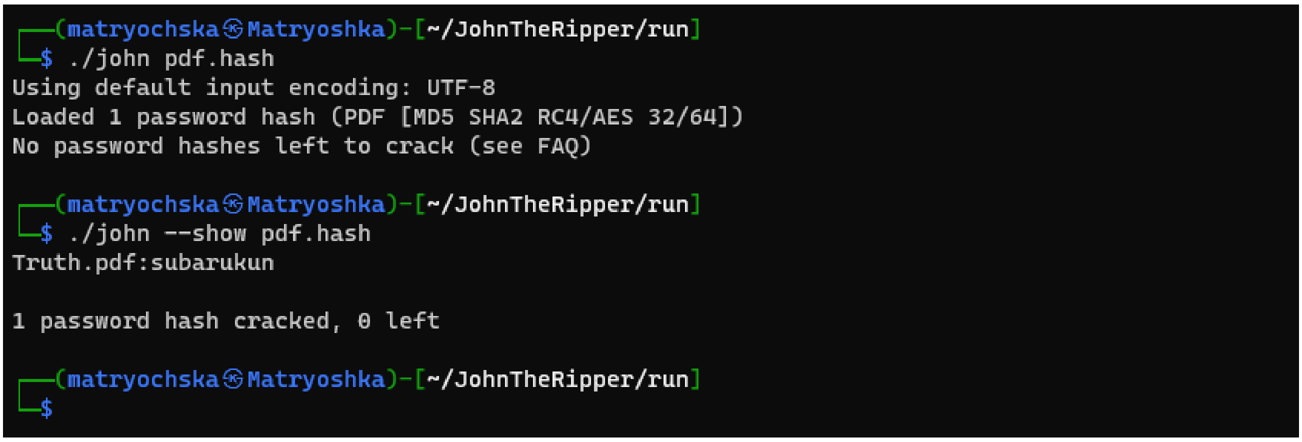
To did brute forcing password using this tools I use:

```
./john pdf.hash
```

And the tools will brute forcing the password for some moments using their wordlist. After the process was finished, we can show the password using

```
./john --show pdf.hash
```

command and here is the result:



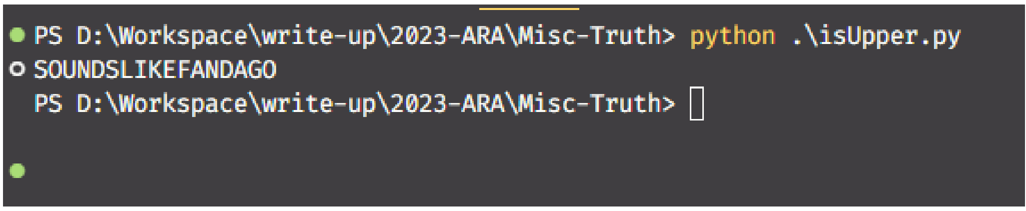
```
┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$ ./john pdf.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
No password hashes left to crack (see FAQ)

┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$ ./john --show pdf.hash
Truth.pdf:subarukun

1 password hash cracked, 0 left

┌──(matryochska㉿Matryoshka)-[~/JohnTheRipper/run]
└─$
```

Based on the result, the password was `subarukun` . After the password was succesfully leaked, I opened the file and read the instructions. Based on the instructions we should erase the title then find uppercase letter, so i used this python code:

```python
result = ''.join(c for c in text if c.isupper())
```

or you can run this code.

Based on that code, the result was:



```
PS D:\Workspace\write-up\2023-ARA\Misc-Truth> python .\isUpper.py
SOUNDSLIKEFANDAGO
PS D:\Workspace\write-up\2023-ARA\Misc-Truth>
```

.

Then, followed the challenge's format, the flag should be:

```
ARA2023{SOUNDS_LIKE_FANDAGO}
```

Give feedback