CS2107 Assignment 1

Easy Challenges

Sanity Check

- We scroll down to the bottom of the pdf
- Copy the flag
- Submit flag: `CS2107{l3t_7He_j0uRn3Y_b3g1N}`

Something's Off

- The Characters `{`,`}`, _ seems to match the usual CTF format, so we assume those are not characters.
- Based on characters shown we only have `a-zA-Z0-9`.
- We get the ascii value with `ord()` in python, then use a char array to shift the characters.
- Retry with different combinations, e.g. `A-Z0-9a-z` if it fails.

flag: `CS2107{5h1ft_c1ph3rs_4r3_4_gR8_w4rMuP}`

HMAC

- run the command:

```
openssl
sha256 -hmac CS21072022 text.txt
```

flag: `CS2107{28025dd41abceecad6056fd5b99587feac67089e6cc874679ab75e0c335a9a67}`

Secret penguin

- run the command:

```
openssl
aes-128-cbc -iv abcdef1234567890abcdef1234567890 -K 1234567890abcdef1234567890abcdef -in
sha256 tux.out
```

flag: `CS2107{4851ed69abe9830dda4ecca87c4634aef98ef8c2f9d7060e8ec5aaedf787a262}`

Prime Time

- Use public database/solver. link

Medium

Insecure OTP

- Xor first 20 bytes of p and c to obtain key
- Use key

flag: `CS2107{OTPOTP_0tp0tp_R3p3at_k3y_15_vuln3rable}`

Public password

- Search twitter
- password on post it
- netcat with password

John the ripper

- Use John the ripper

flag: `CS2107{abcd1234}`

Birthday hash.

- Birthday paradox: It is extremely difficult to find a person with a birthday equals to a specific date, but it is surprisingly common for two people to have the same birthdays in a small crowd.
- We compute various possible strings until we find 2 strings that cause a hash collision.

flag: `CS2107{No_h@sh_can_esc4pe_b1rthd@y_p@rad0x}`

Perfect AES imperfect key

- There are only 3 bytes that are used in the sha512 key used for encryption
- brute force on compute cluster

Substitution cipher

Looking at the text we can infer:

`UT2107{`: "C" = "U", "S" = "T"

`(H), (HH), (HHH)...`: "I" = "H", "V" = "M",

`B., Q., U.`: "A" = "B", "B" = "Q", "C" = "U", "D" = "P", "E" = "Y", "F" = "O", "G" = "N", "H" = "F", "I" = "H"

- Run decoder once

`WHIS AGXEEKEDW DESCXIBES`: `THIS AGREEMENT DESCRIBES`, "T" = "W", "R" = "X", "M" = "K", "N" = "D", "T" = "W"

`DISCJAIKEX`, "L" = "J"

- Run decoder again

`RARRANTG`: "W" = "R", "Y" = "G"

`LIMITATILN`: "O" = "L"

`DEMICE MANAFACTARER`: "V" = "M", "U" = "A"

`ADOBE FLASH SLAYER LICENSE TERMS`: "P" = "S"

flag: `CS2107{SUBSITUTION_CIPHER_IS_OFTEN_SHOWN_IN_MOVIE_FOR_SOME_REASON}`

HARD

COPPER RSA

Reference

- Since we have `[c1,..., c5], [n1, ..., n5]`, we can use Chinese Remainder Theorem to find M using Hastad's Broadcast Attack:
- Using mathematica, we compute cube root of M, m. We then find the roots from the quadratic equation.

```
b = a^(1/3)
Solve[4*x^2 + 521  * x + 47829 == b, x]
```

- We get 1 negative root and 1 positive root. We can then use `long_to_bytes()` and `decode()` to get the text:

```
THIS FISH IS SO RAW CS2107{c0pP3r_br@s5_Br0nz3_m3tAl_s73el_1r0n_Go1d} HE'S STILL FINDING
```

RSA doors

Reference

- Since phi is provided, which is (p - 1)(q - 1), where p and q are primes, (p - 1) and (q - 1) will likely not be primes.
- We find all factors of phi {f1 ... fn} and brute force to get possible n's.
- `cs2107` and `CS2107` failed. trying `door` decrypts ciphertext to get password for docx
- trailing = indicates base64 like data. no lowercase spotted, decrypt with base32 online.
- use online decoders to get private key
- Decode `DATA`, verify `SIGNATURE` using public key, then for each `SEQ` take the 1st legitimate packet.

plaintext: `n0_noiS3_t00_d1fficult_7o_cLeAn`

try `cs2107{` and `CS2107{` to get full flag.