ISFCR CTF WORKSHOP

CRYPTOGRAPHY & STEGANOGRAPHY

How to get started with hacking ciphers and images

I: Steganography

- What is steganography?
- What is a file header?
- What are magic bytes?
- Other methods of steg
- How can we store and extract data in and from images?
- How to solve steg challenges?
- Useful Tools
- Some example CTFs

I: WHAT IS STEG?

The process of hiding secret data in images that is not visible by just opening the image. It requires the use of certain techniques of modifying the hex contents of the file to read the secret data

- In steganography, the most important part is the file headers.
- Not all files may run the way they seem. In many files the file extension may not even be the correct indicator.
- Some files may not open because the headers are not formatted right.
- Some files may have another file hidden in the hex code of the file or even a link to another file!

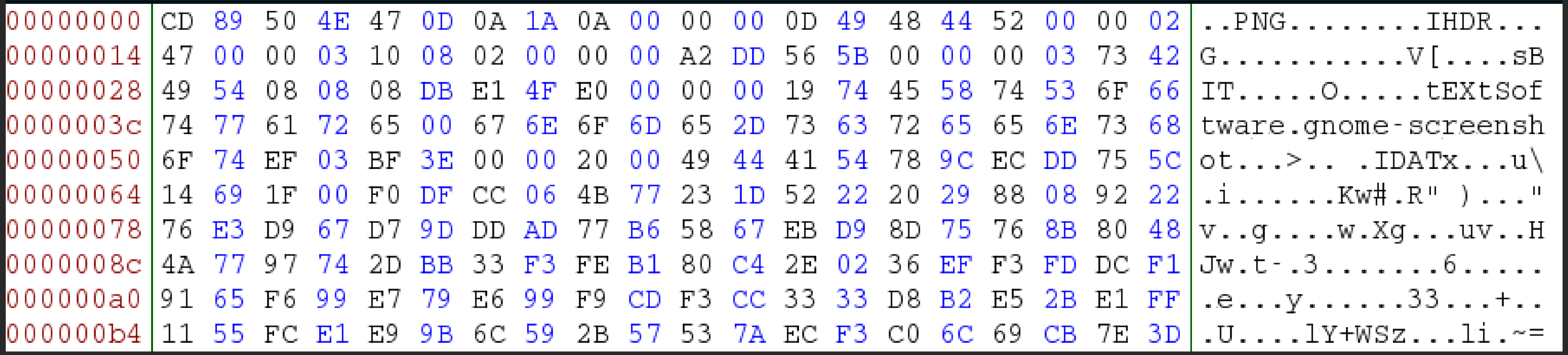
Note: any type of file can be manipulated from png to pdf. So make sure to check all files you get in a challenge!

WHAT IS A FILE HEADER?

How does a computer identify what type the file is of?

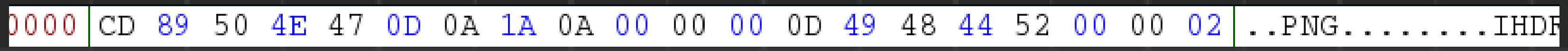
Most people would think that a file is identified by its extension. This is the case for most normal files, but sometimes the file may not even have an extension. A file is actually identified by its hex headers. These headers are a small block of data containing a set of magic bytes that tell the computer that the file needs to be executed in a certain way, The magic bytes are set in stone for specific file formats. These can be added or removed by opening the file in a hex editor

HEADER OF A PNG



```
00000000 CD 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 02 ..PNG........IHDR...
00000014 47 00 00 03 10 08 02 00 00 00 A2 DD 56 5B 00 00 00 03 73 42 G...........V[....sB
00000028 49 54 08 08 08 DB E1 4F E0 00 00 00 19 74 45 58 74 53 6F 66 IT.....O.....tEXtSof
0000003c 74 77 61 72 65 00 67 6E 6F 6D 65 2D 73 63 72 65 65 6E 73 68 tware.gnome-screensh
00000050 6F 74 EF 03 BF 3E 00 00 20 00 49 44 41 54 78 9C EC DD 75 5C ot...>.. .IDATx...u\
00000064 14 69 1F 00 F0 DF CC 06 4B 77 23 1D 52 22 20 29 88 08 92 22 .i......Kw#.R" )..."
00000078 76 E3 D9 67 D7 9D DD AD 77 B6 58 67 EB D9 8D 75 76 8B 80 48 v..g....w.Xg...uv..H
0000008c 4A 77 97 74 2D BB 33 F3 FE B1 80 C4 2E 02 36 EF F3 FD DC F1 Jw.t-.3.......6.....
000000a0 91 65 F6 99 E7 79 E6 99 F9 CD F3 CC 33 33 D8 B2 E5 2B E1 FF .e...y......33...+..
000000b4 11 55 FC E1 E9 9B 6C 59 2B 57 53 7A EC F3 C0 6C 69 CB 7E 3D .U....lY+WSz...li.~=
```

WHAT ARE MAGIC BYTES?



```
0000 CD 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 02 ..PNG........IHDR
```

Format

They are a set of hex digits that are inserted at the beginning of a file

What do they mean?

The hex bytes translate to <file_type> when converted to plaintext. For example, the magic bytes of a png spells out PNG when converted from hex to plaintext

How to know which magic bytes to use

In Linux, when running the file command, we can figure out what kind of file it is. From there we can insert the magic bytes to make the file openable

Other methods of steg

LSB Steg

This is the method of hiding data in the color information of an image file

Image manipulation

Sometimes data can be hidden in an image by simply changing the colors of it to make the data unreadable

HOW TO STORE AND EXTRACT DATA FROM AN IMAGE?

Linux!

Storing an zip file in a png

Zip files can be stored inside a png file using simple linux commands. They can also be extracted by just running the unzip command on the image. The command to embed is:
cat image.png file.zip >> out.png

Using StegHide

Steghide is a tool that automatically finds hidden files inside other files. A simple command would be:
steghide extract -sf <file>

Using a hex editor

A hex editor is simple. It shows the entire hex dump of the file which contains information like the headers, the color profiles and more. It can be used to modify the file to execute it if it is not executable

HOW TO SOLVE STEG CHALLENGES?

Common methods and challenge types

1

Step 1

Checking the file

First step is to always try opening the image file or any other file as if it were normally executable. Even if it is, some data might be hidden in it. If the file executes properly, we can search for clues in the image itself before looking at its hex

2

Step 2

Check for any hidden files

In Linux, a simple command called "strings" will allow us to read all the plaintext data that is present in a file. With strings we can find out if any other files have been embedded into the source file

3

Step 3

Hex editing

if we find anything in strings, we can open the file in a hex editor like bless and try to fix the image or search for other files or links

1

Step 4

Image manipulation using GIMP

GIMP is a free image editing software present in Linux. Whenever we encounter an image that is off coloured or has weird artifacts in it, we can check the image to see if t has hidden messages. This can be done by playing aroun with the color settings of the image and viewing the bit planes using GIMP.

2

Step 5

LSB

In a specific type of steganography, we can store data into the first bit of every pixel. This is a harder type of steg to detect so we will have to use python to write a payload to extract the secret from the image programmatically

3

Step 6

Tools!

Our last step is to wrap everything up with a quick run through of our tools. The main tool used in steg is called steghide. Steghide provides a simple CLI interface which allows us to hide or extract data from an image with one command. We can also try using unzip on images to see if they have any other hidden files in them.

USEFUL TOOLS

Some handy tools to pick apart image files or audio files

Python

There are various libraries to write payloads for steg in python

GIMP

https://gimp.org

John

https://github.com/openwall/john

USEFUL TOOLS

Bless Hex editor

https://github.com/afrantzis/bless

Steghide

https://www.kali.org/tools/steghide/

EXAMPLE CTFS

- **Header manipulation**

  Using magic bytes to make png files executable

- **Two Png challenge**

  Learning how to use a hex editor

- **Embedded files**

  Learning how to find out embedded files inside PNGs

2: Cryptography

- What is cryptography?
- How does a crypto algorithm work?
- What's the difference between encryption and encoding?
- What is symmetric and asymmetric encryption?
- What are the types of algorithms?
- How to solve crypto ciphers?
- Useful tools
- Some example CTFs

We will also look at how to identify ciphers

I: WHAT IS CRYPTO?

The process of converting a human-readable message into an unreadable format with the help of an algorithm is called cryptography

- In cryptography, some algorithms require keys to encrypt and decrypt.
- This is more secure as it allows for a better obfuscation of the input plaintext.
- Another method is the process of encoding a plaintext with an algorithm which can be reversed using the same algorithm without a key.

Some algorithms work on complex mathematical concepts. If you want to know more, feel free to message us on discord!

ENCRYPTION VS ENCODING

What's the difference and how to identify them

Encoding

- Encoding is the process of taking a simple plaintext and running it through and algorithm to obfuscate it.
- It is helpful for assigning a value with a specific format for applications.
- An example of an encoding algorithm is base64.
- A string is in base64 if: usually end with one or two "=" signs. They also only contain characters from a-z and 0-9. The length of a base64 string is always a multiple of 4.

Encryption

- Encryption is thee process of applying an algorithm to a simple plaintext and then converting it into a seemingly random ciphertext which can be decrypted only with a key
- It is used for storing sensitive information like passwords and encrypting data in transit through the internet
- Examples of encryption algorithms are AES, RSA and CURVE-25519

SYMMETRIC VS ASYMMETRIC

Encryption Key

In symmetric encryption there exists only one key whereas in asymmetric encryption there exists a private and a public key which are both different. In asymmetric, when a specific private key is used for encryption, the data can be decrypted only with its associated public key

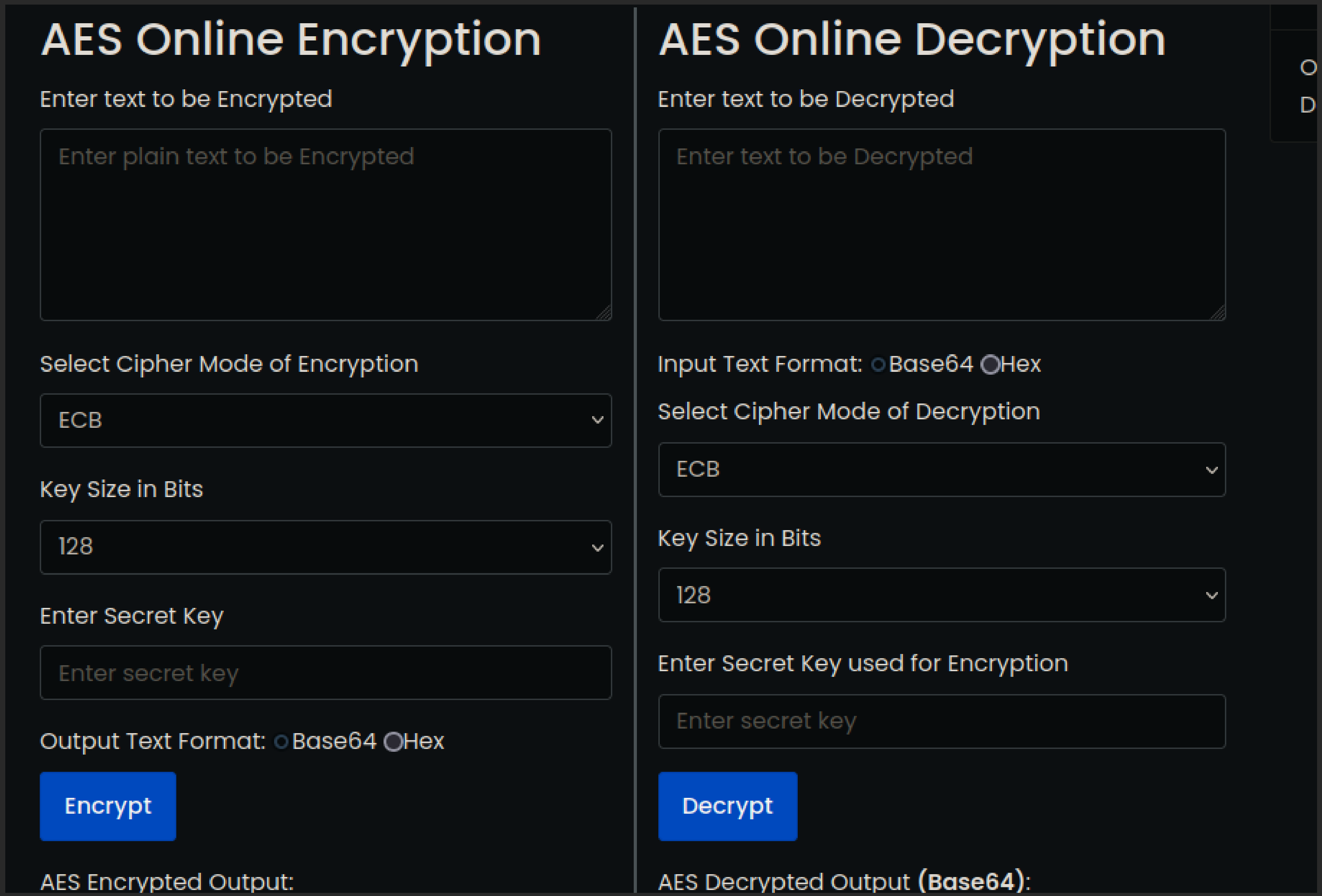
AES

AES has 2 main modes: Electronic Code Book (ECB) and Cipher Block Chaining (CBC). To decrypt AES, we require a secret key. In CBC mode we require an initialization vector that we can obtain during encryption. This is absent in ECB which can lead to leakage of pattern info when large amounts of data is encrypted

RSA

RSA has a rather complex working. It uses the concept of prime numbers, generating primes and Carmichael's totient. Generally when trying to decrypt RSA, we are met with a few values:

C - Cipher message
E - Pub Key
D - Private Key
P - Factor 1 (prime num)
Q - Factor 2 (prime num)
Phi - Intermediate Value



AES Online Encryption

Enter text to be Encrypted

Enter plain text to be Encrypted

Select Cipher Mode of Encryption

ECB

Key Size in Bits

128

Enter Secret Key

Enter secret key

Output Text Format: ○ Base64 ○ Hex

AES Encrypted Output:

AES Online Decryption

Enter text to be Decrypted

Enter text to be Decrypted

Input Text Format: ○ Base64 ○ Hex

Select Cipher Mode of Decryption

ECB

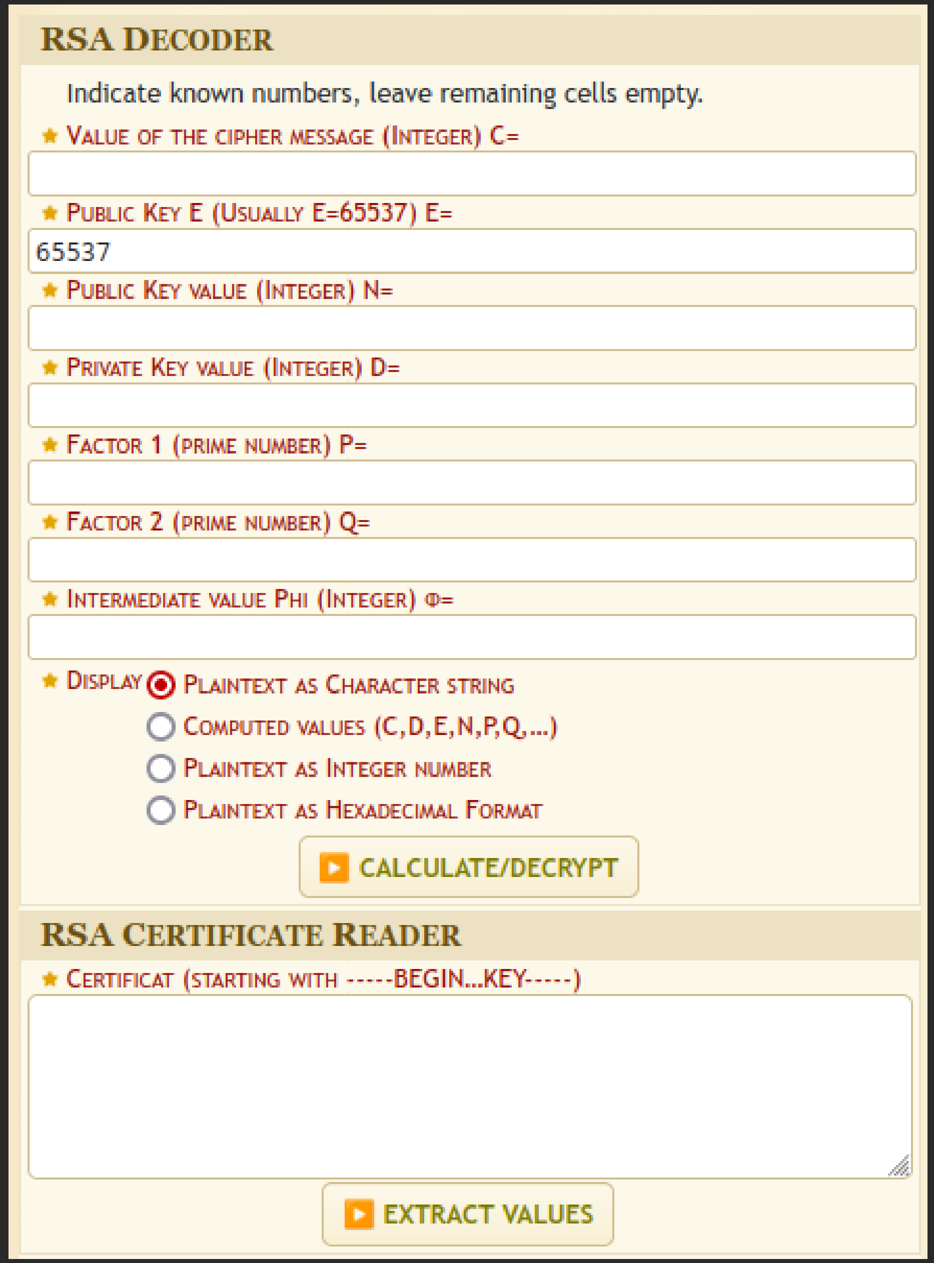
Key Size in Bits

128

Enter Secret Key used for Encryption

Enter secret key

AES Decrypted Output **(Base64)**:



RSA DECODER

Indicate known numbers, leave remaining cells empty.

★ VALUE OF THE CIPHER MESSAGE (INTEGER) C=

★ PUBLIC KEY E (USUALLY E=65537) E=
65537

★ PUBLIC KEY VALUE (INTEGER) N=

★ PRIVATE KEY VALUE (INTEGER) D=

★ FACTOR 1 (PRIME NUMBER) P=

★ FACTOR 2 (PRIME NUMBER) Q=

★ INTERMEDIATE VALUE PHI (INTEGER) Φ=

★ DISPLAY ● PLAINTEXT AS CHARACTER STRING
○ COMPUTED VALUES (C,D,E,N,P,Q,...)
○ PLAINTEXT AS INTEGER NUMBER
○ PLAINTEXT AS HEXADECIMAL FORMAT

RSA CERTIFICATE READER

★ CERTIFICAT (STARTING WITH -----BEGIN...KEY-----)

Other types of ciphers and encoding

Vigenere

Vigenere is based on the relative position of the letters of the ciphertext and the key. This cipher can be decrypted with a key. The algorithm involves subtracting the positions of the letters of the cipher and key

ROT-13

ROT-13 stands for rotational 13 which is an algorithm that involves incrementing the current alphabet by 13 places in the English alphabet. The increment is circular

Diffie-Helman

Diffie-Helman is a key exchange cryptosystem which facilitates the exchange of a secret message with the help of just one commonly agreed upon number

MD5

Generally has no decryption method but it is possible to brute-force. MD5 also has the ability to oroduce a hash collision which can be exploited

XOR-Cipher

Converts plain text into binary, XORs each binary digit and returns an XORed output

HOW TO SOLVE CRYPTO CIPHERS?

Common methods and challenge types

1

Step 1

Identifying the cipher

When doing crypto challenges, it generally is mentioned what cryptosystem the cipher is using. But in some cases, it can be identified from looking at patterns in the string

2

Step 2

Find the key

In most challenges that require a key to crack the cipher, the challenge description or some included files will contain a peculiar word that will be the key used to decrypt the message

3

Step 3

Write the payload

Languages like python that contain many libraries for cryptosystems can be used to write a custom payload to decrypt the given ciphertext of a challenge

USEFUL TOOLS

Some handy tools to crack and identify ciphers and write payloads in Python

Pwntools

https://github.com/Gallopsled/pwntools

Dcode.fr

https://dcode.fr/en

John

https://github.com/openwall/john

EXAMPLE CTFS

RSA Challenge

Understanding what each letter means in the RSA cryptosystem and how to decrypt a message that was encrypted using RSA

Vigenere Cipher

Decrypting a message in vigenere cipher using a special key

XOR-Cipher

Understanding how an XOR-Cipher works and decrypting it