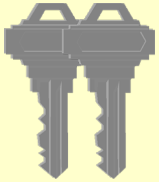
Information Security

2025 Project 1

Decrypt the ciphertext!



Prof. Junbeom Hur
TA. Donghee Kim, Juyeon Yi

Department of Computer Science and Engineering
Korea University

1. Project Overview

Given Information and Requirements

- You are given
  - A pair of plaintext $\boldsymbol{P1}$ and ciphertext $\boldsymbol{C1.}$
  - A ciphertext $\boldsymbol{C2.}$
  - The code for the encryption algorithm “enc.py”.
  - Graphical ciphertext “clue.png”.

- Goal is to recover the key $\boldsymbol{K}$ and decrypt ciphertext $\boldsymbol{C2.}$

2. Main Task

What are the key K and the plaintext $P2$?

- Executing the provided encryption algorithm **enc.py** produces the following result:

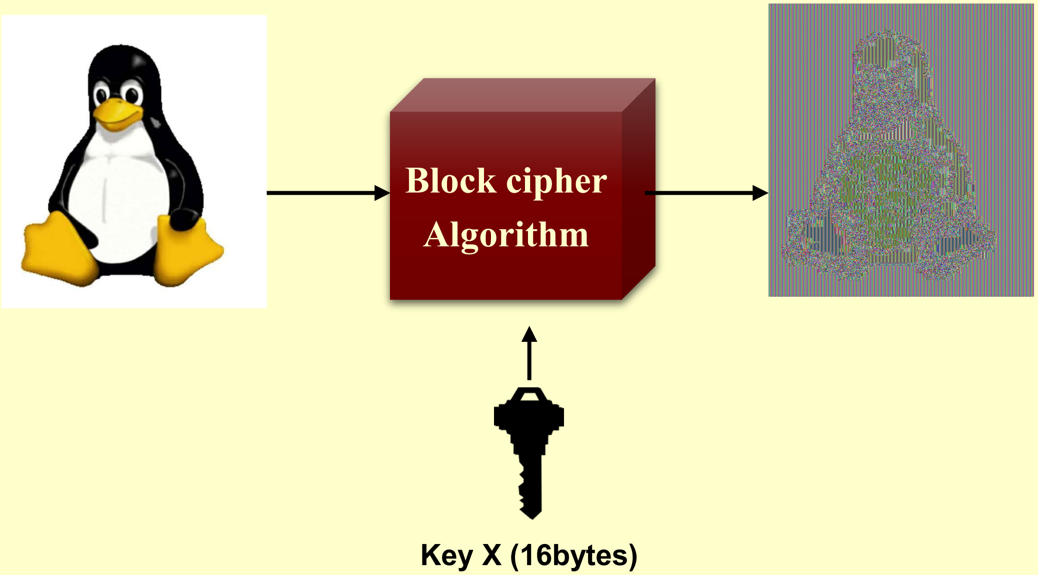
Plain Text (string)	$P2$ ↓
Cipher Text $C2$ (hex)	f0f1f84d807d9bfdf416a18ac5ab9c3b1a7a06e7b69e020d4 35ac230c6f1695e50dc5a139d217332f270363bdccffe1b

- You should
  - Recover the key K using python code.
  - Decrypt the $C2$ using python code to find $P2$.

3. Background

Electronic Codebook (ECB) mode encryption

- The simplest way to encrypt with a block cipher.
- Advantage: parallelizable.



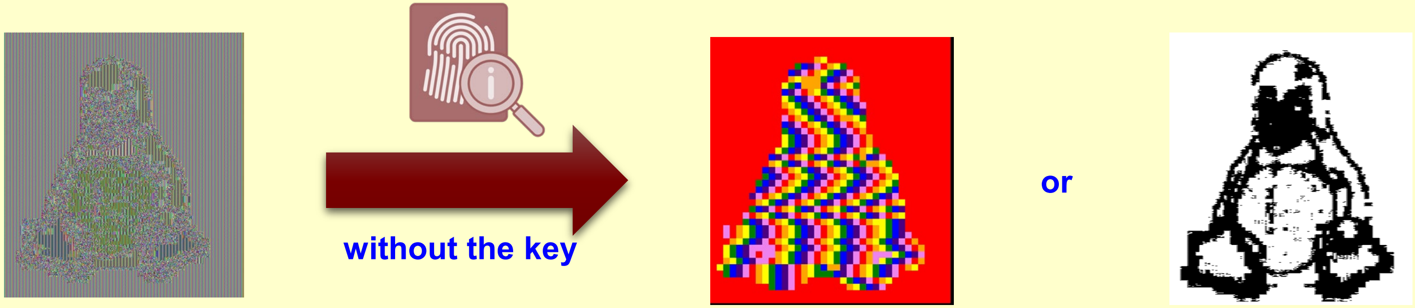
Block cipher Algorithm

Key X (16bytes)

3. Background

Electronic Codebook (ECB) mode encryption

- Limitation: equal plaintext blocks → equal ciphertext blocks.



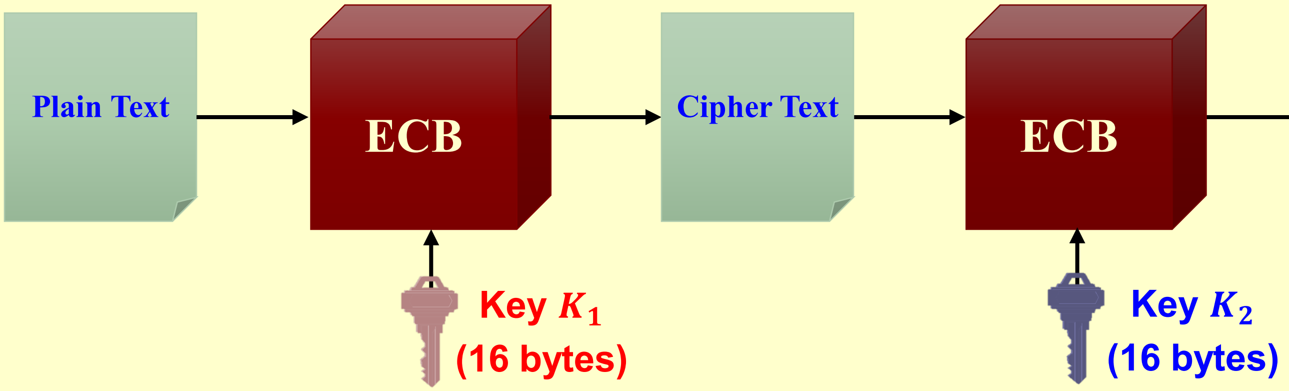
without the key

or

4. Given Encryption Algorithm

in enc.py

- Algorithm workflow (Double AES-128)



4. Given Encryption Algorithm

in **enc.py**

- When the encryption script **enc.py** is executed, the following plaintext–ciphertext pair is produced:

Plain Text $P1$ **(string)** **“This is a top secret message. Do not share it with anyone!”**

↓

Cipher Text $C1$ **(hex)**

```
3e40001d1bc6d179551288606d9404914c002383a158dbc4574
8957a845b3195eaf9ac3f1e34dc2ef8888c70399ec0acbed366b
8e1fcc8b501f5763fe91862a3
```

4. Given Encryption Algorithm

in enc.py

- Key K (32 bytes)
  - $K = K_1 + K_2$
    - $K_1 = a_1 + a_2$
    - $K_2 = a_3 + a_4$
- Find a_2 and a_4

a_1 (hex)	a3f19c8d4e6b72f0 (8 bytes)	a_2 (hex)	???????????????? (8 bytes)
a_3 (hex)	5e8b41c2d9f07a36 (8 bytes)	a_4 (hex)	**************** (8 bytes)

4. Given Encryption Algorithm

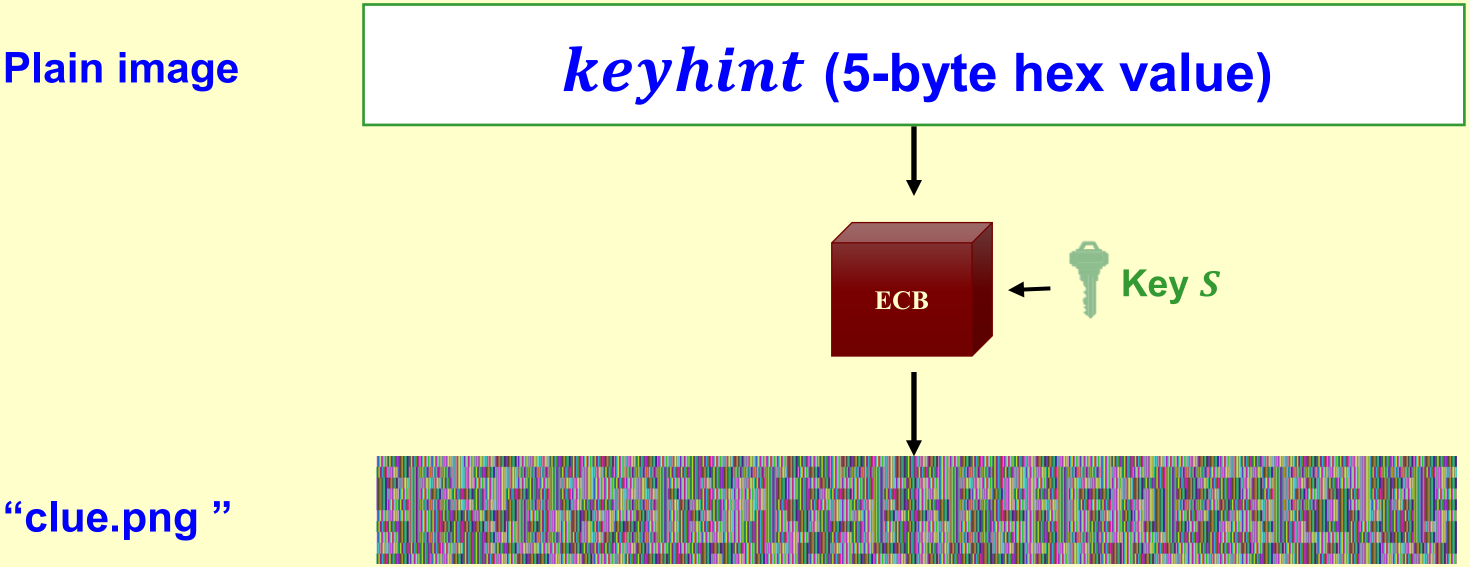
in enc.py

- a_2 **(hex) = *keyhint* (hex, 5 bytes)+ @@@@@@ (hex, 3 bytes)**
- a_4 **(hex) = *keyhint* (hex, 5 bytes)+ ###### (hex , 3 bytes)**

5. Hints on the key *k*

Encrypted image “clue.png” with embedded *keyhint*

- Encryption algorithm for “**clue.png** ”: AES in ECB mode.



Plain image

keyhint (5-byte hex value)

ECB

Key *S*

“clue.png ”

6. Submission Guideline

- Source code (python) for
  1. **finding the *keyhint*** (filename: keyhint_[your student number].py).
  2. **recovering** the key ***K*** (filename: recover_[your student number].py).
  3. **decrypting ciphertext c2** (filename: dec_[your student number].py).

- Report (filename: report_[your student number].pdf)
  - **Approach for finding the *keyhint*** (*Do not use external tools*) .
  - **Approach for recovering the key *K*** (*Do not use external tools*) .
  - **Comments in your source codes.**
  - **Explain the functions, variables, and other elements used in your codes.**
  - Answer:
    1. Key ***K***
    2. plaintext ***P2***

6. Submission Guideline

- Submit your final deliverable as a single ZIP archive (filename: 2025project1_[your student number]_[your name].zip).

1. keyhint_[your student number].py
2. recover_[your student number].py
3. dec_[your student number].py
4. report_[your student number].pdf

7. Grading Criteria

- 30pt in total
  - 20pt for answers (10pt for each answer)
  - 10pt for the others

- 0pt if any of the four files (i.e., source codes, report) is not submitted.

- Late submission is not allowed (0pt will be given for any reason).

8. Submission

- Due date
  - 2025. 10. 5, 23:59.
- Upload the solutions into LMS system.