6. 3DES, Mode of Operations, and RC4

Fuyou Miao，Wenchao Huang

Web：http://staff.ustc.edu.cn/~huangwc/crypto.html

Email: mfy@ustc.edu.cn，huangwc@ustc.edu.cn

Key Points

- **Multiple encryption** is a technique in which an encryption algorithm is used multiple times.
    - In the first instance, plaintext is converted to ciphertext using the encryption algorithm.
    - This ciphertext is then used as input and the algorithm is applied again.
    - This process may be repeated through any number of stages.
- **Triple DES** makes use of three stages of the DES algorithm, using a total of two or three distinct keys.

Key Points

- **A mode of operation** is a technique for enhancing the effect of a crypto-graphic algorithm or adapting the algorithm for an application, such as applying a block cipher to a sequence of data blocks or a data stream.

- **Five modes of operation** have been **standardized** by NIST for use with symmetric block ciphers such as DES and AES

  - (1) electronic codebook mode (2) cipher block chaining mode (3) cipher feedback mode (4) output feed-back modecounter mode (5)

- A **stream cipher** is a symmetric encryption algorithm in which ciphertext output is produced bit-by-bit or byte-by-byte from a stream of plaintext input. The most widely used such cipher is **RC4**.

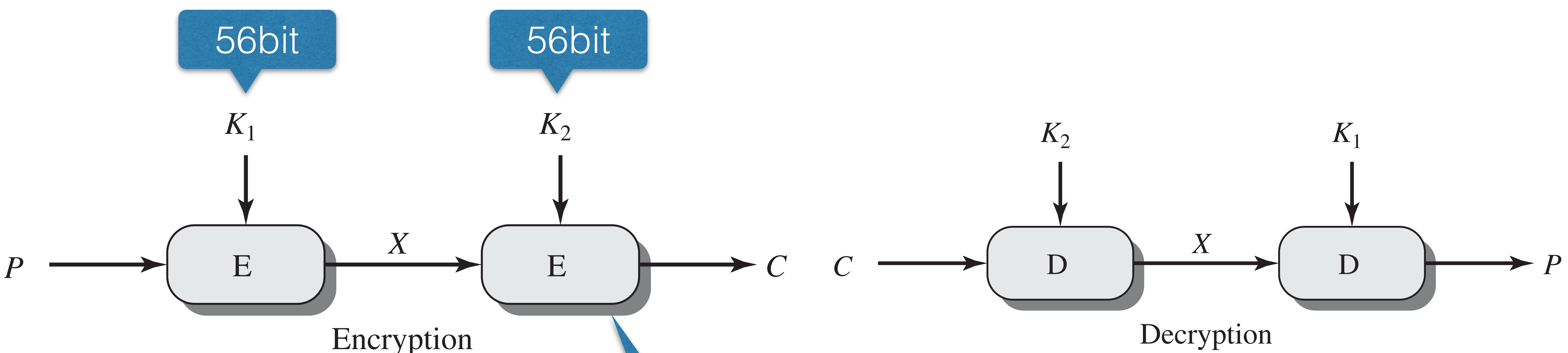
Contents

1. Multiple Encryption and Triple DES Motivation

- The cons of DES
    - Brute-force attacks
- Approaches
    - AES, or
    - use multiple encryption with DES and multiple keys
        - Question: How many encryption stages?

1. Multiple Encryption and Triple DES
(1) Double DES

- The **simplest** form of multiple encryption has **two** encryption stages and two keys
- Encryption： $C = E(K_2, E(K_1, P))$
- Decryption： $P = D(K_1, D(K_2, C))$



Complexity of attacks： 2^{112}???

1. Multiple Encryption and Triple DES
(1) Double DES

- Meet-in-the-middle attack (中间相遇攻击)

$$\text{2DES}(K_1\|K_2, M) \ = \ \text{DES}(K_2, \text{DES}(K_1, M))$$

$$\text{2DES}^{-1}(K_1\|K_2, C) \ = \ \text{DES}^{-1}(K_1, \text{DES}^{-1}(K_2, C))$$

$$\text{DES}^{-1}(K_2, C_1) = \text{DES}(K_1, M_1)$$

1. Multiple Encryption and Triple DES
(1) Double DES

- Given a known pair (M, C)
  - Encrypt M for all 2^{56} possible values of K_1
    - Store these results in a table and then sort the table by the values of DES(K_1, M)
  - Decrypt C using all 2^{56} possible values of K_2
    - As each decryption is produced, check the result against the table for a match
    - If a match occurs, then test the two resulting keys against a new known plaintext–ciphertext pair
- The complexity of the above operation: 2^{57}

1. Multiple Encryption and Triple DES
(1) Double DES

- Meet-in-the-middle attack (中间相遇攻击)

$MinM_{\mathsf{2DES}}(M_1, C_1)$
$\quad$ **for** $i = 1, \ldots, 2^{56}$ **do** $L[i] \leftarrow \mathrm{DES}(T_i, M_1)$
$\quad$ **for** $j = 1, \ldots, 2^{56}$ **do** $R[j] \leftarrow \mathrm{DES}^{-1}(T_j, C_1)$
$\quad$ $S \leftarrow \{\ (i, j)\ :\ L[i] = R[j]\ \}$
$\quad$ Pick some $(l, r) \in S$ and **return** $T_l \| T_r$

For any $(i, j) \in S$ we have

$$\mathrm{DES}(T_i, M_1) = L[i] = R[j] = \mathrm{DES}^{-1}(T_j, C_1)$$

- Question: Is the attack correct?

1. Multiple Encryption and Triple DES
(1) Double DES

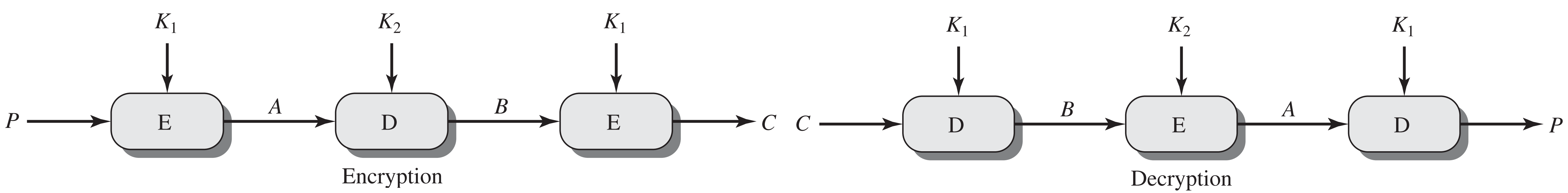
- Analysis
  - For any given plaintext M
    - 2^{64} possible ciphertext values, 2^{112} possible keys
    - How many keys can produce a given ciphertext C?
      - $2^{112}/2^{64} = 2^{48}$
      - i.e., false alarm rate: $1 - 2^{-48}$
  - For two blocks of known plaintext–ciphertext
    - 2^{128} ciphertext values, 2^{112} possible keys
    - How many possible ciphertexts correspond to a key on average?
      - $2^{128}/2^{112} = 2^{16}$
      - i.e., false alarm rate: 2^{-16}

1. Multiple Encryption and Triple DES
(2) Triple DES with Two Keys

- Triple DES with Two Keys

$$C = E(K_1, D(K_2, E(K_1, P)))$$

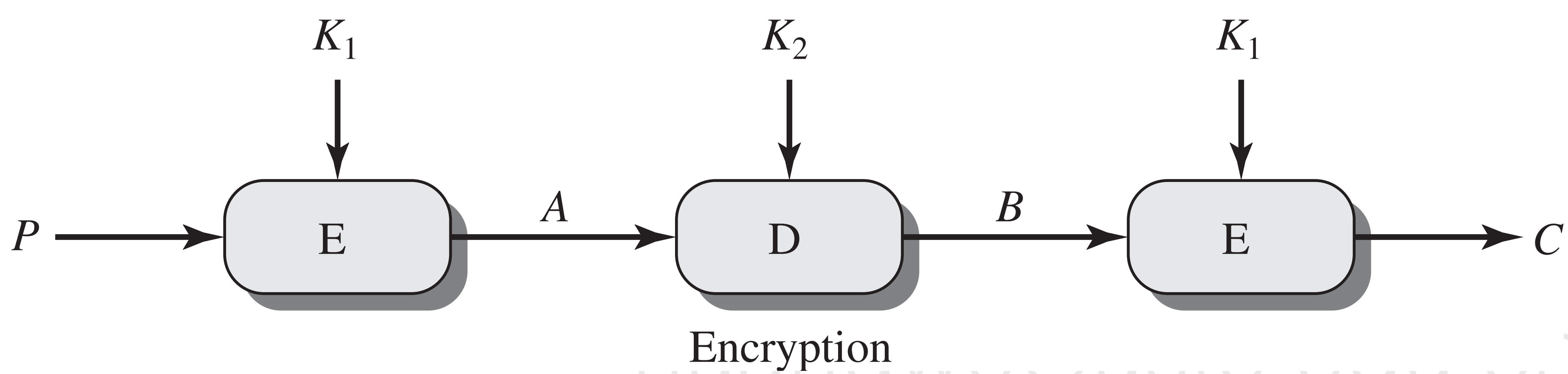
$$P = D(K_1, E(K_2, D(K_1, C)))$$



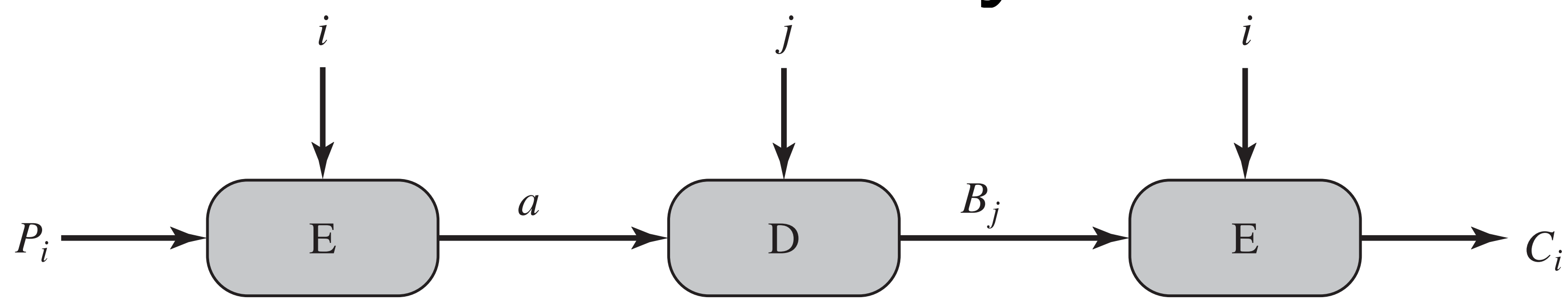
- Key management standards ANS X9.17 and ISO 8732

1. Multiple Encryption and Triple DES
(2) Triple DES with Two Keys

- Proposed attack 1 (**Impractical attack?**)
  - 2^{56} chosen plaintext-ciphertext (2^{56} is impractically large)
    - For $K_1=0\sim2^{56}$
      - Set A=0, compute P using K_1
      - **Choose** (P, C), then compute **B**
    - For $K_2=0\sim2^{56}$
      - Since A=0, compute **B** using K_2



1. Multiple Encryption and Triple DES
(2) Triple DES with Two Keys



- Proposed attack 2
  - Known-plaintext attack

1. Obtain (P, C) pairs, and Create table 1
2. Pick an arbitrary value a
   - Repeat:
     - For i=0~2^{56}
       - Compute P_i from a and i
       - If (P_i, C_i) is in table 1
         - Compute B from i and C_i
         - Create (B, i) in table 2
3. For j=0~2^{56}, compute B_j from j and a
4. If B_j is in table 2, output result $(i||j)$

Table 1

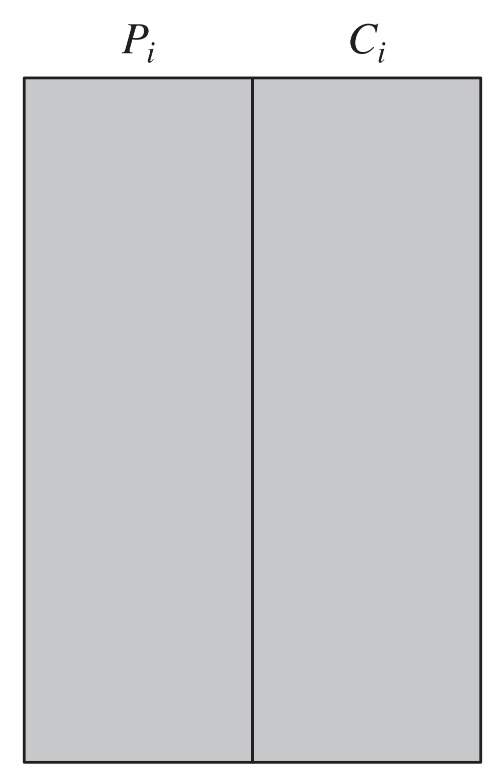
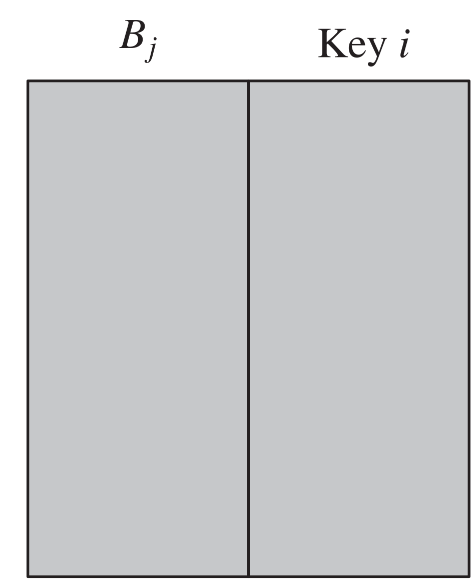


Table 2



复杂度: $\left(2^{56}\right)\dfrac{2^{64}}{n} = 2^{120-\log_2 n}$

1. Multiple Encryption and Triple DES (3) Triple DES with Three Keys

- Although the attacks just described appear impractical, anyone using two-key 3DES may feel some concern
  - Another solution: Triple DES with Three Keys

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$

- PGP
- S/MIME

2. Modes of operation

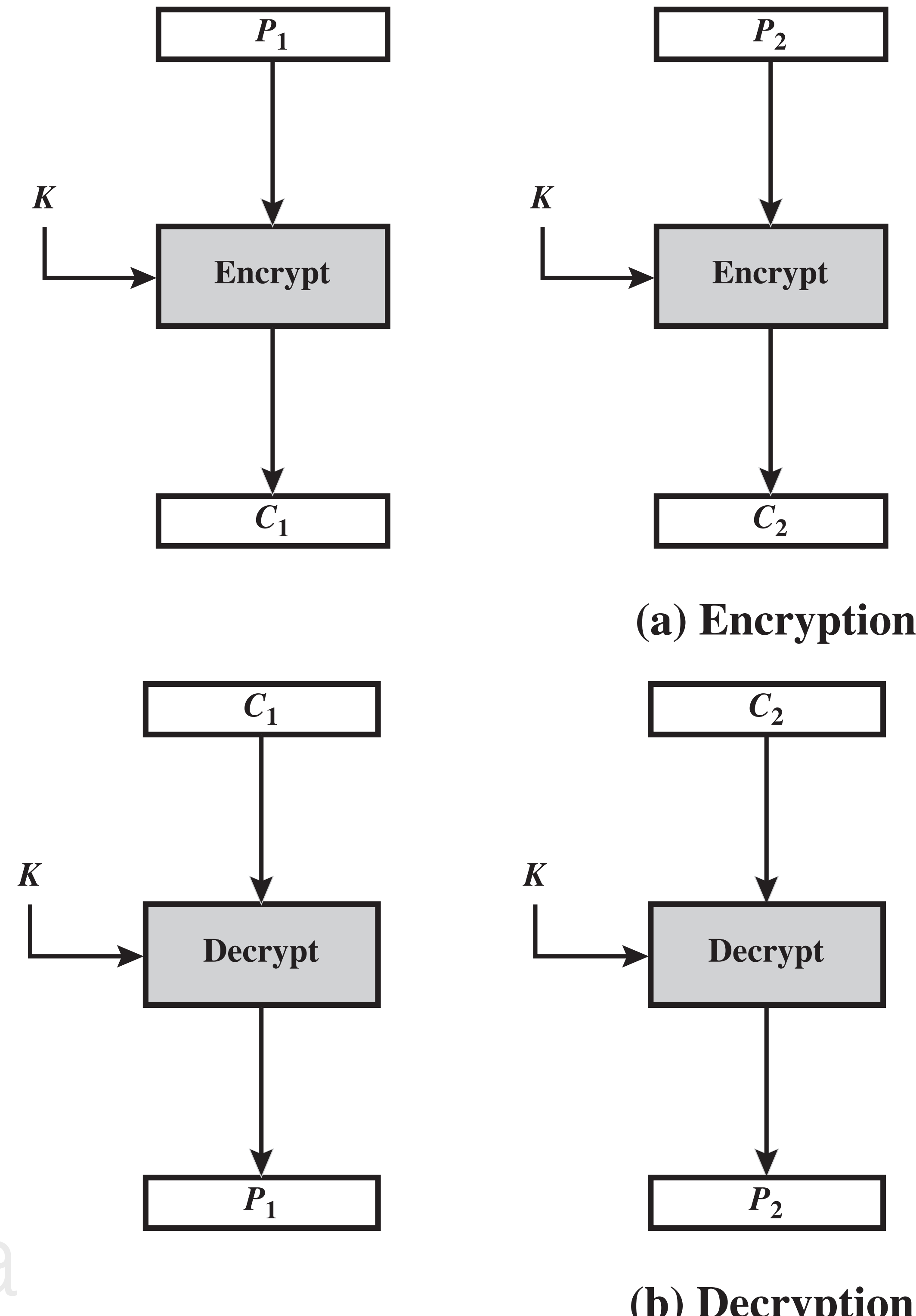
Mode		Description	Application
电码本（ECB） Electronic CodeBook		$\text{cipher} = \text{out}$ $\text{in} = \text{plain}$	Encryption of single values e.g., a key
密文分组链接 (CBC) Cipher Block Chaining		$\text{cipher} = \text{out}$ $\text{in} = \text{plain} \oplus \text{out}_{\text{prev}}$	General-purpose block-oriented transmission; Authentication
密文反馈 (CFB) Cipher FeedBack	Select s bits	$\text{cipher} = \text{out} \oplus \text{plain}$ $\text{in} = \text{cipher}_{\text{prev}} \oplus \text{shift}$	General-purpose block-oriented transmission; Authentication
输出反馈(OFB) Output FeedBack		$\text{cipher} = \text{out} \oplus \text{plain}$ $\text{in} = \text{out}_{\text{prev}}$	Stream-oriented transmission over noisy channel
计数器 (CTR) Counter		$\text{cipher} = \text{out} \oplus \text{plain}$ $\text{in} = \text{counter}$	General-purpose block-oriented transmission; High speed

2. Modes of operation
(1) ECB (electronic codebook)

- Process

$$\text{cipher} = \text{out} \qquad \text{in} = \text{plain}$$

- Observation
  - Fit for encryption of single values
    - E.g., keys
  - Possibly insecure for lengthy messages
    - Repetitive elements

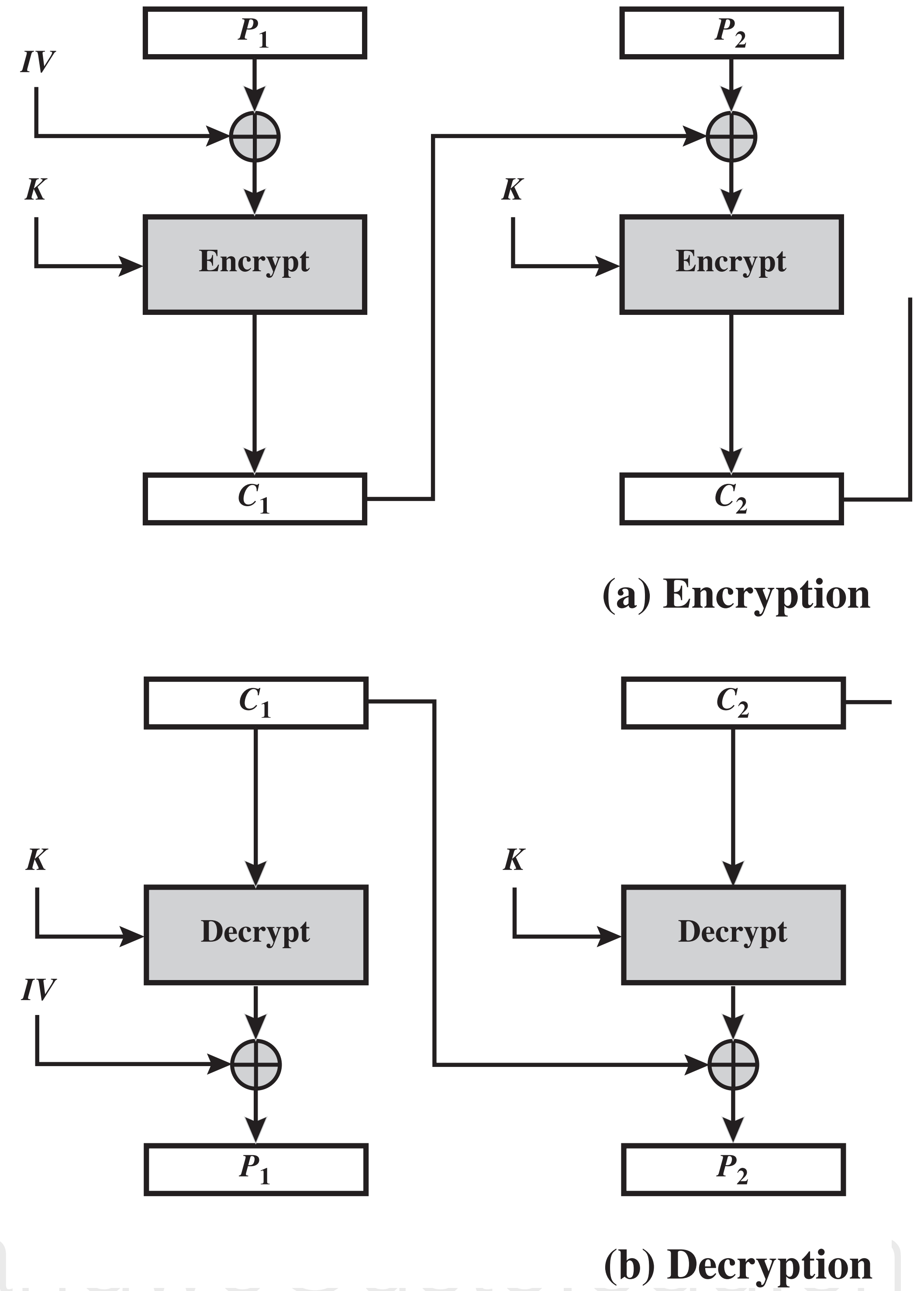


2. Modes of operation
(2) CBC (cipher block chaining)

- Process

$$\text{cipher} = \text{out} \qquad \text{in} = \text{plain} \oplus \text{out}_{\text{prev}}$$

- Observation
  - Achieves confidentiality
  - Authentication



2. Modes of operation
(3) CFB (Cipher Feedback)

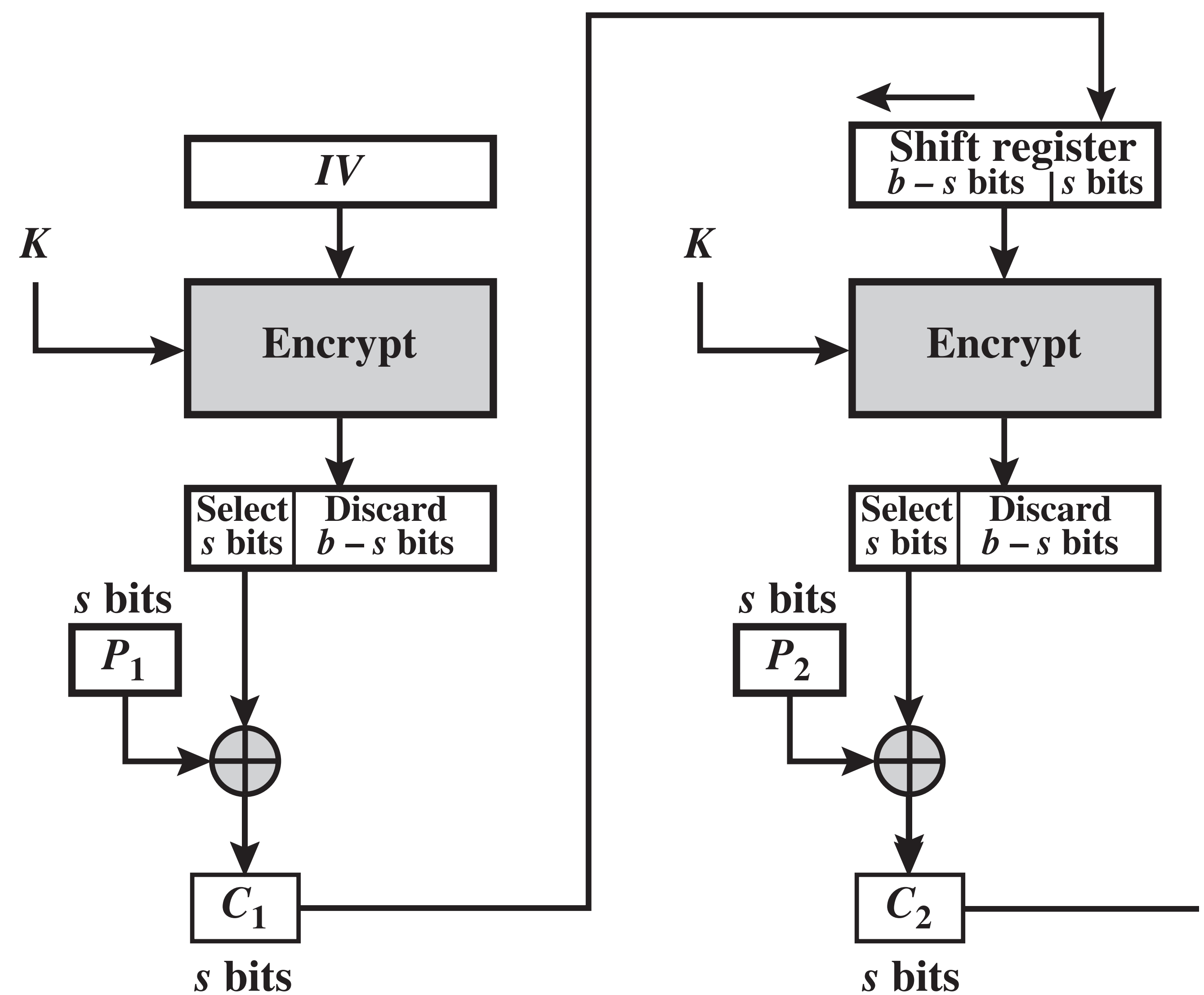
- Process

Select
s bits

$$\text{cipher} = \text{out} \oplus \text{plain}$$

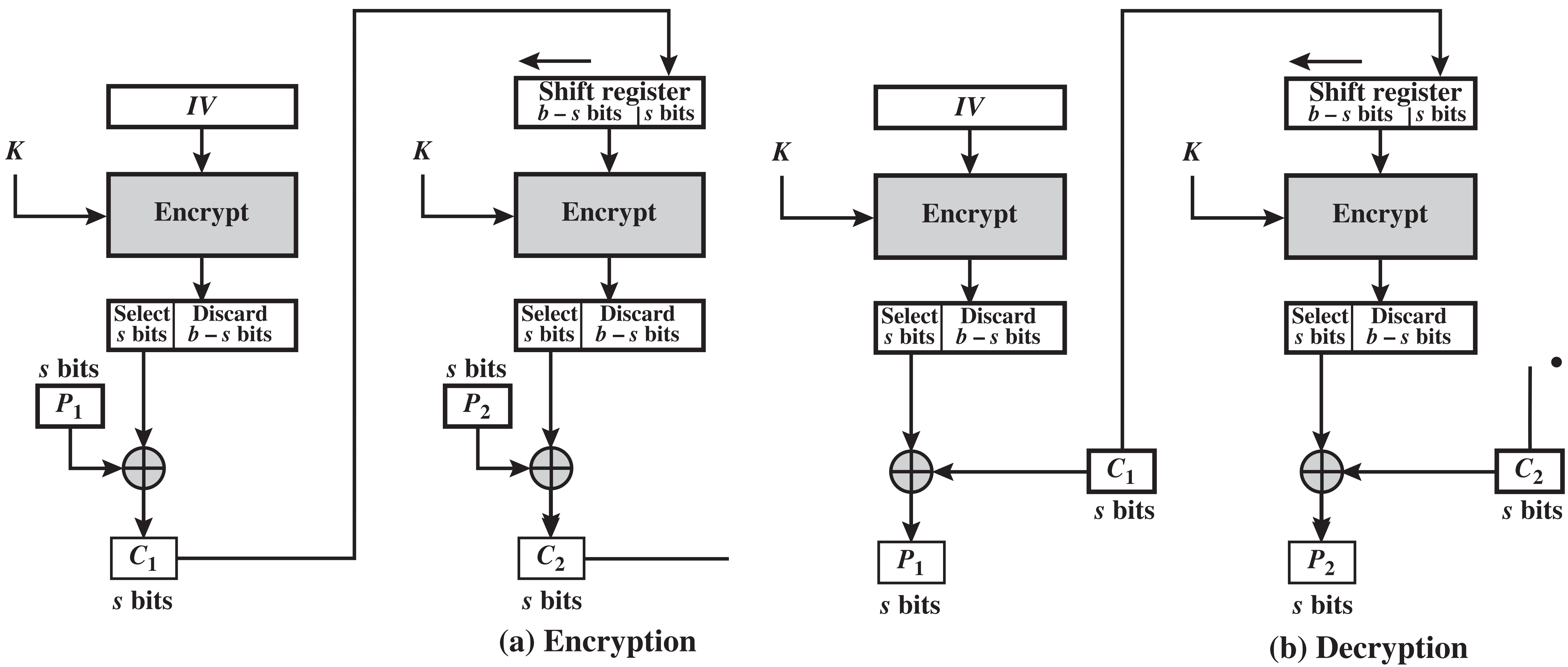
$$\text{in} = \text{cipher}_{\text{prev}} \oplus \text{shift}$$

- Observation
    - Can be viewed as stream cipher



(a) Encryption

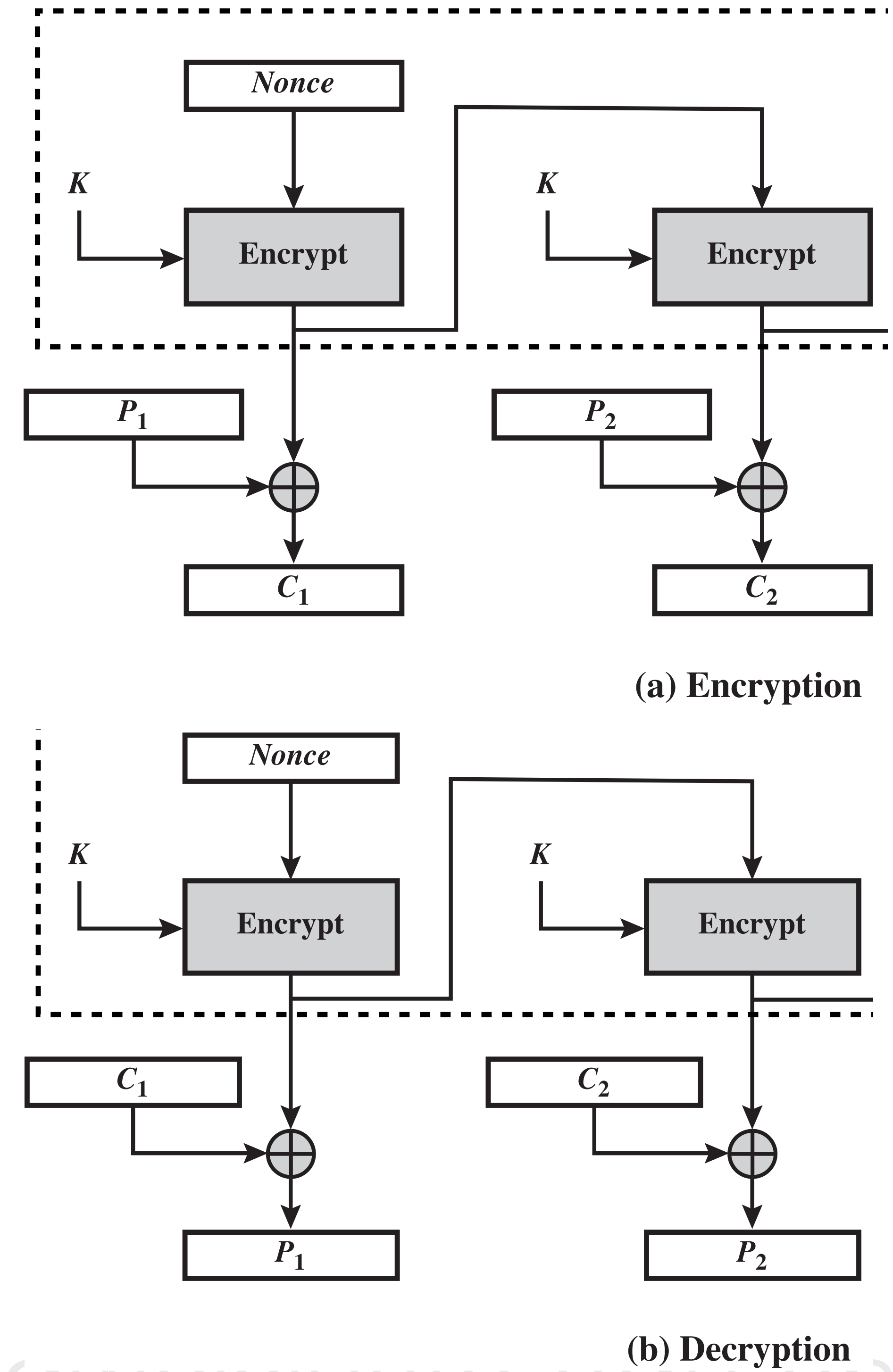
2. Modes of operation
(3) CFB (Cipher Feedback)



2. Modes of operation
(4) OFB (Output Feedback)

- Process

$$\text{cipher} = \text{out} \oplus \text{plain}$$
$$\text{in} = \text{out}_{\text{prev}}$$



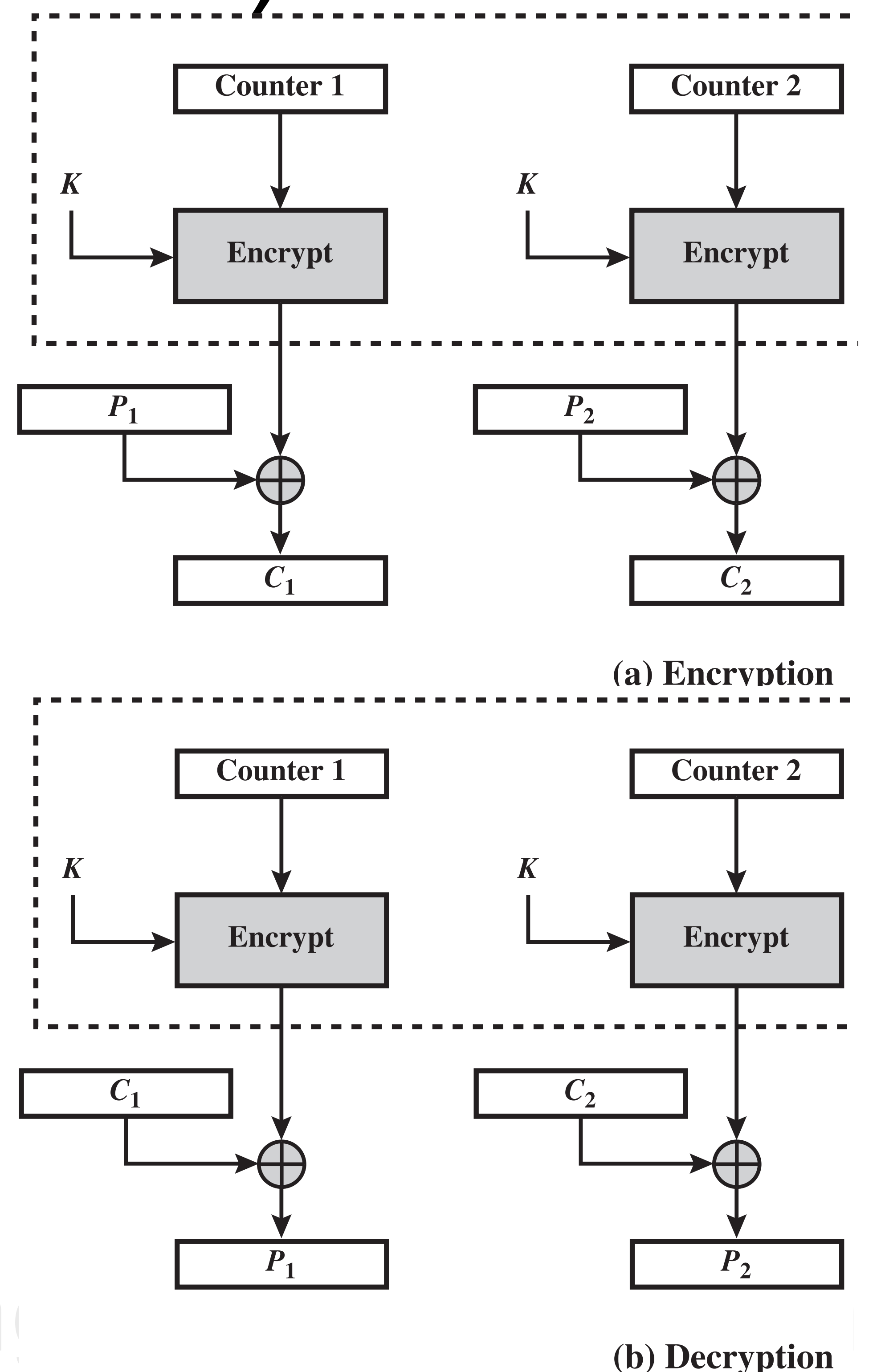
- Observation
  - Similar to CFB
  - bit errors in transmission do not propagate
  - OFB is more vulnerable to a message **stream modification attack** than is CFB
  - .

2. Modes of operation
(5) CTR (Counter)

- Process

$$\text{cipher} = \text{out} \oplus \text{plain}$$
$$\text{in} = \text{counter}$$

- Observation
  - Efficiency、Preprocessing
  - Provable security
  - Simplicity



2. Modes of operation
(6) Mode in Disk Encryption

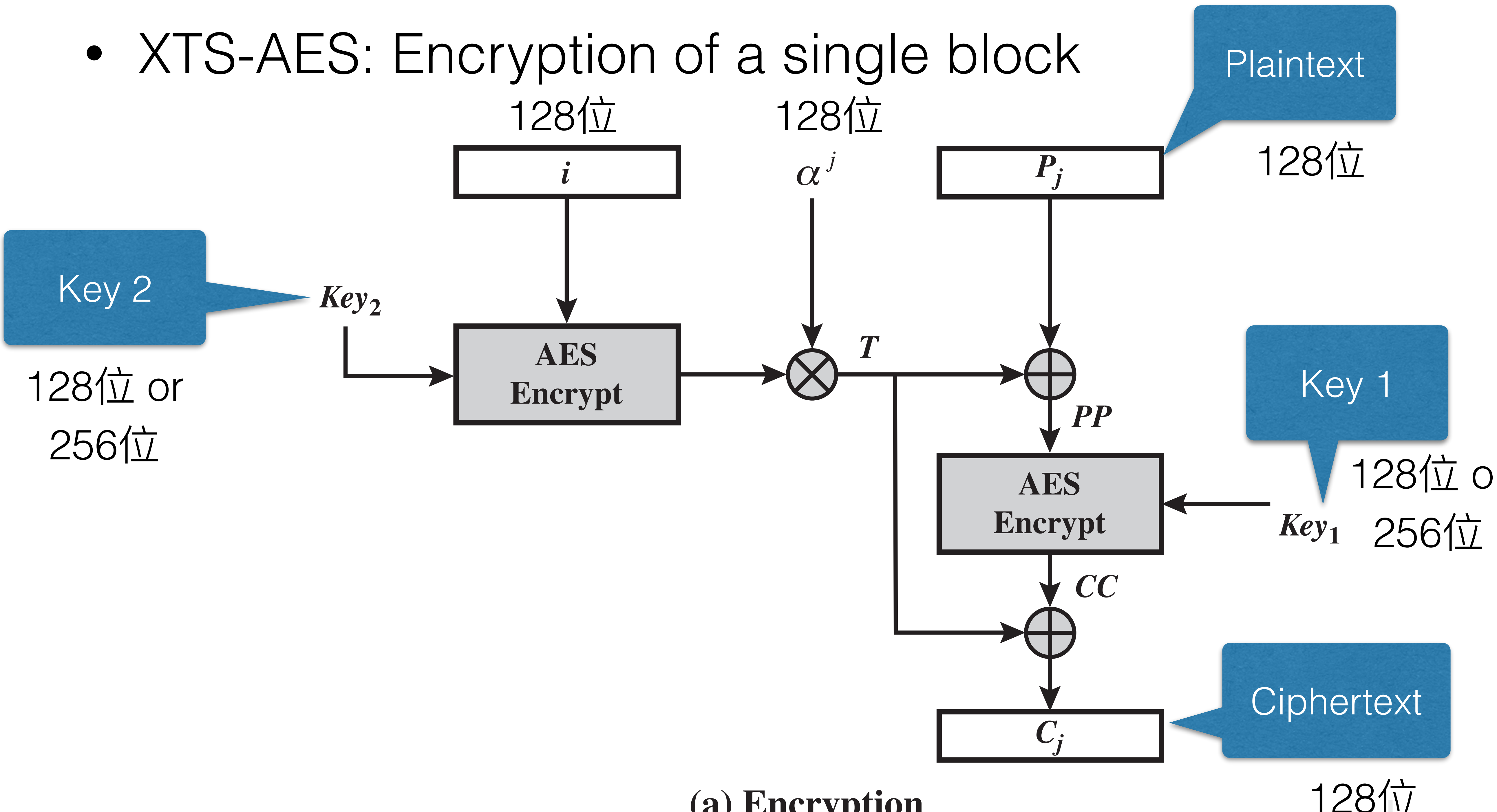
- Storage Encryption Requirements (**P1619 standard, XTS-AES**)
  - The ciphertext is freely available for an attacker
  - The data layout is not changed on the storage medium and in transit.
  - Data are accessed in fixed sized blocks, independently from each other
  - Encryption is performed in 16-byte blocks, independently from other blocks
  - There are no other metadata used, except the **location of the data blocks** within the whole data set
  - The same plaintext is encrypted to different ciphertexts at different locations, but always to the same ciphertext when written to the same location again
  - A standard conformant device can be constructed for decryption of data encrypted by another standard conformant device.

2. Modes of operation
(6) Mode in Disk Encryption

- XTS-AES
  - XEX-based tweaked-codebook mode with ciphertext stealing
  - Applications
    - Mac OS X Lion's FileVault 2
    - Windows 10's BitLocker, etc.

2. Modes of operation
(6) Mode in Disk Encryption

- XTS-AES: Encryption of a single block

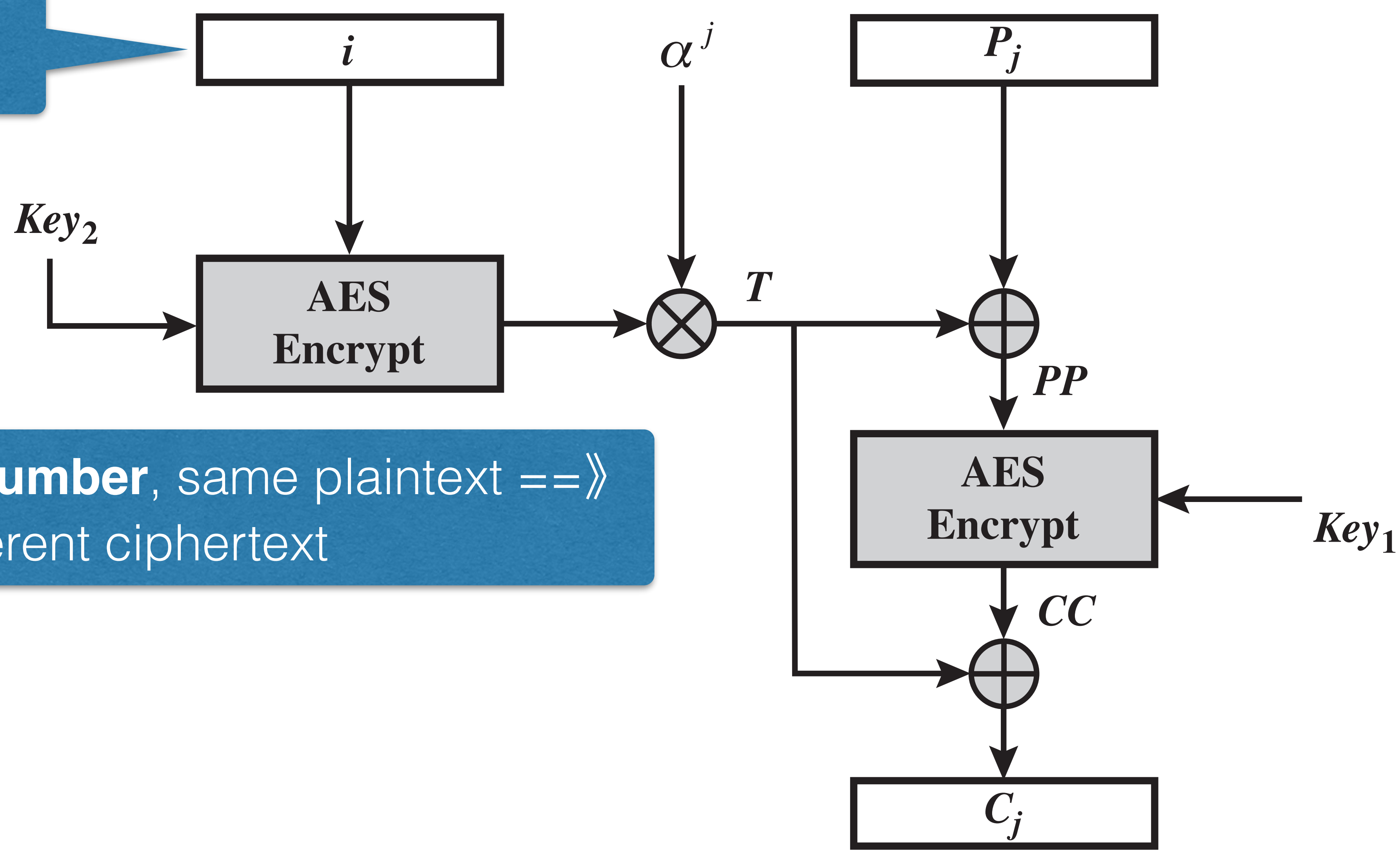


(a) Encryption

2. Modes of operation
(6) Mode in Disk Encryption

- XTS-AES: Encryption of a single block

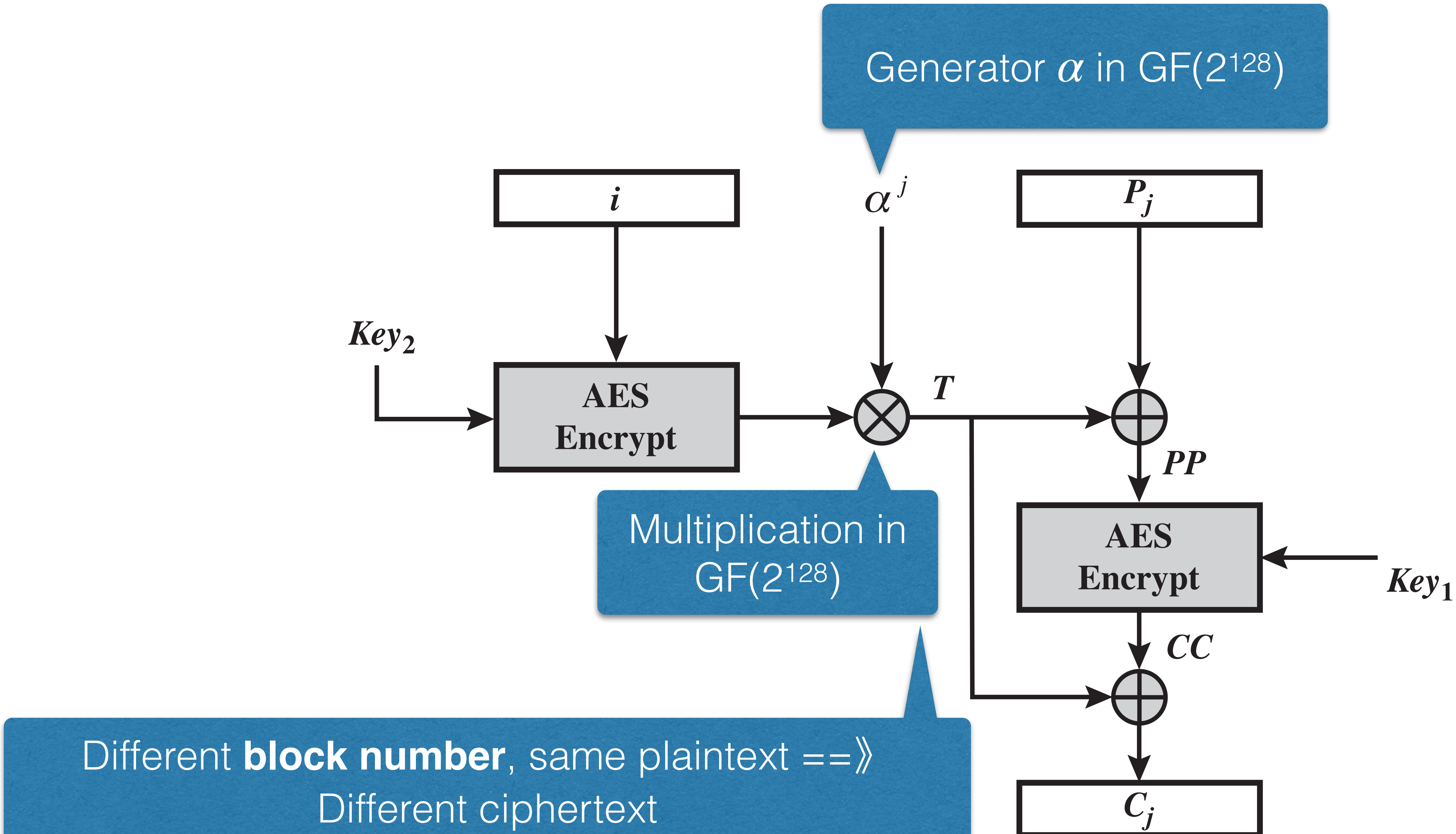
Sector of disk



Different **sector number**, same plaintext ==》 Different ciphertext

(a) Encryption

2. Modes of operation
(6) Mode in Disk Encryption



(a) Encryption

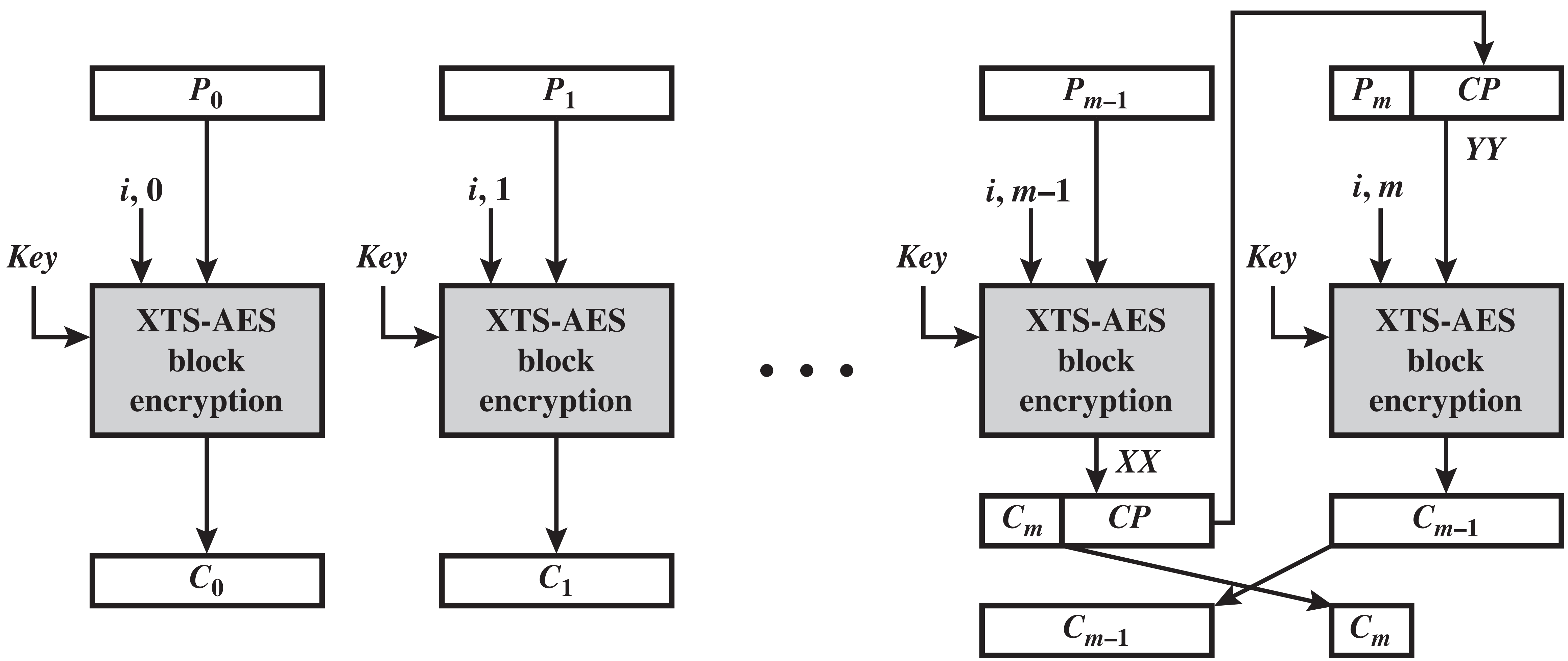
2. Modes of operation (6) Mode in Disk Encryption

- XTS-AES

i: sector# （Tweaked）， j： block#

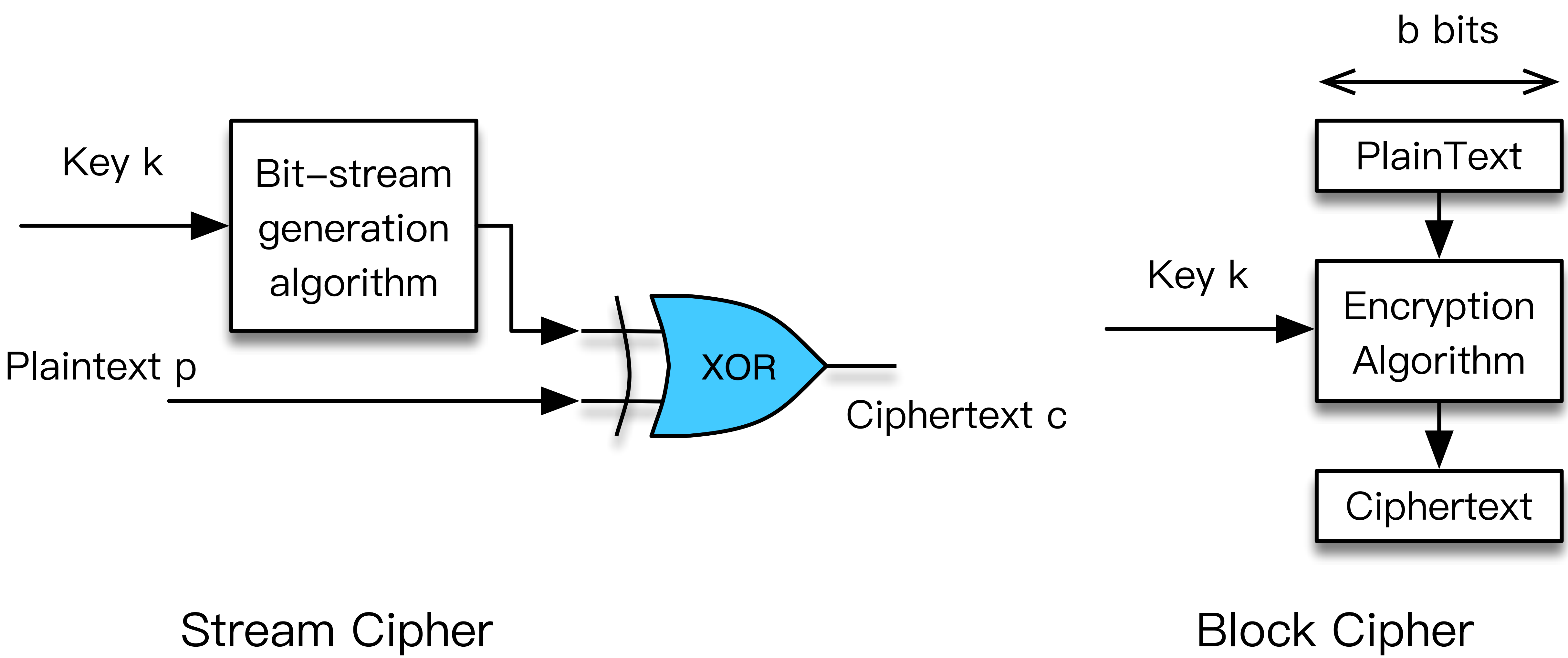
Block encryption: XTS-AES-blockEnc(K, P_j, i, j)

Block decryption: XTS-AES-blockDec(K, C_j, i, j)

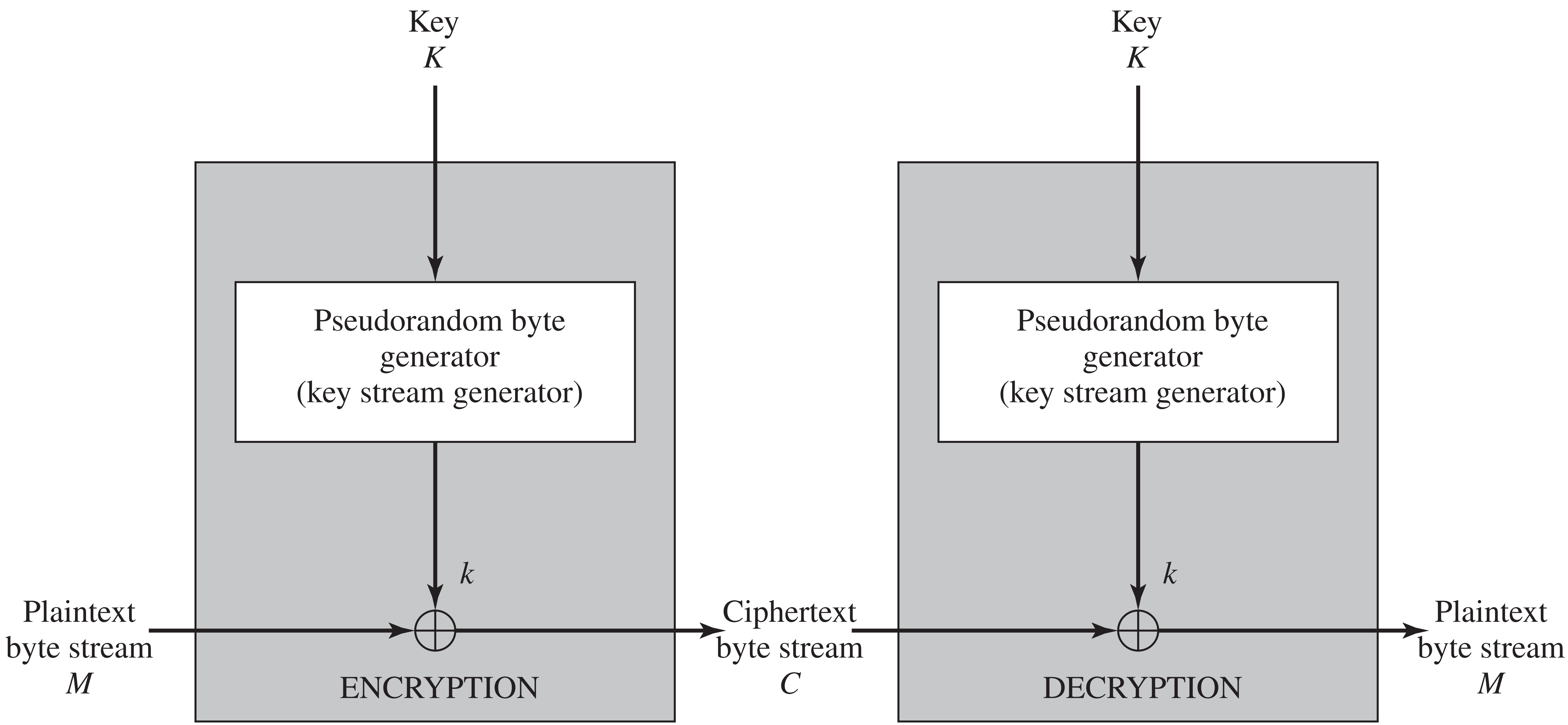


(a) Encryption

3. Stream cipher and RC4
Recall: stream cipher



3. Stream cipher and RC4
Stream cipher



3. Stream cipher and RC4
Stream cipher

- Important design considerations
  - The encryption sequence should have a large period
  - The keystream should approximate the properties of a true random number stream as close as possible
  - The output of the pseudorandom number generator is conditioned on the value of the input key

3. Stream cipher and RC4
RC4

- In 1987，by Ron Rivest
- Period: greater than 10^{100}
- Eight to sixteen machine operations are required per output byte
- Used in web SSL/TLS, wireless WEP，WPA)

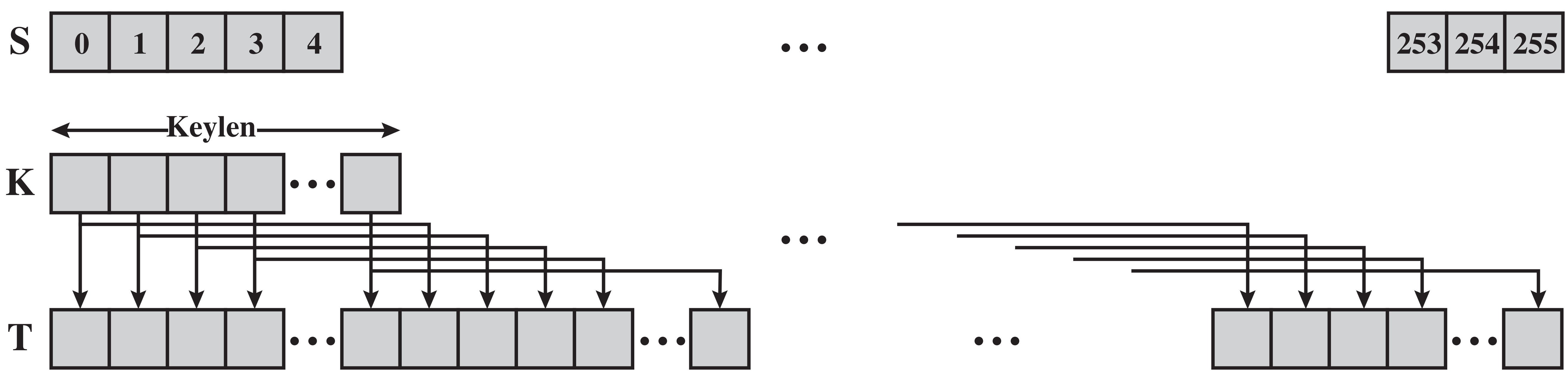
3. Stream cipher and RC4
RC4

- RC4 Algorithm
  - Input： **variable**-length key **K** of from 1 to 256 **bytes**
  - Process
    1. Initialization of S
       - S[0]=0, S[1]=1, … S[255]=255
       - Initial permutation (置换)
    2. Stream Generation
       - Output a byte per Permutation

3. Stream cipher and RC4
RC4

1. Initialization of S

```
/* Initialization */
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

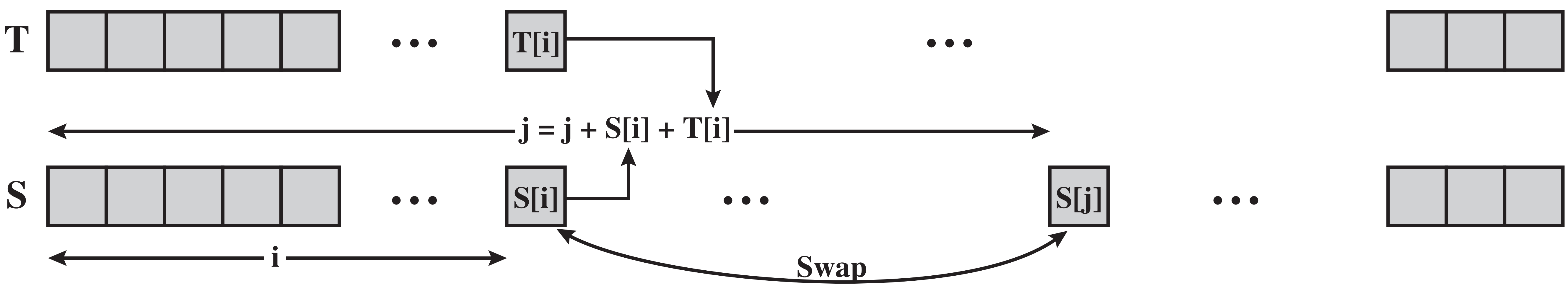


3. Stream cipher and RC4
RC4

1. Initialization of S

```
/* Initialization */
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

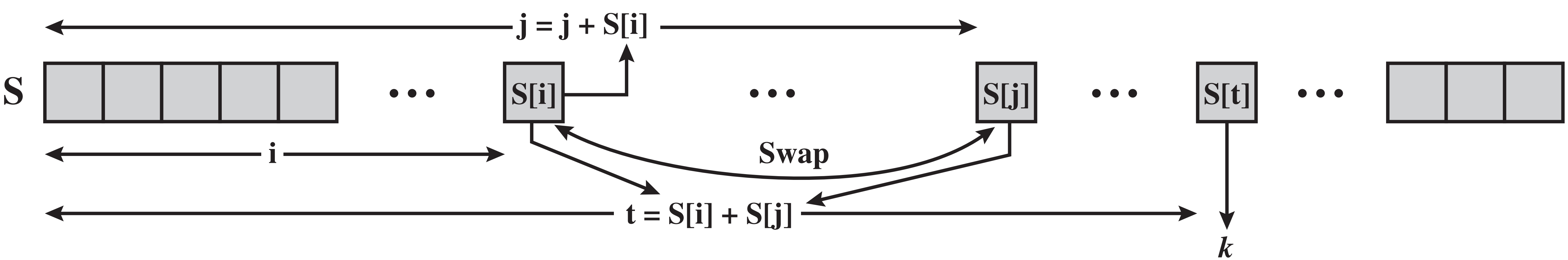
```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
    Swap (S[i], S[j]);
```



3. Stream cipher and RC4
RC4

2. Stream Generation

```
/* Stream Generation */
i, j = 0;
while (true)
  i = (i + 1) mod 256;
  j = (j + S[i]) mod 256;
  Swap (S[i], S[j]);
  t = (S[i] + S[j]) mod 256;
  k = S[t];
```



3. Stream cipher and RC4
RC4

- Is RC4 secure? — See Usenix Security ’15
- Applicated
  - 1997 WEP
  - 2003/2004 WPA
  - 1995 SSL
  - 1999 TLS
- Deprecated
  - 2015 TLS

3. Stream cipher and RC4
RC4

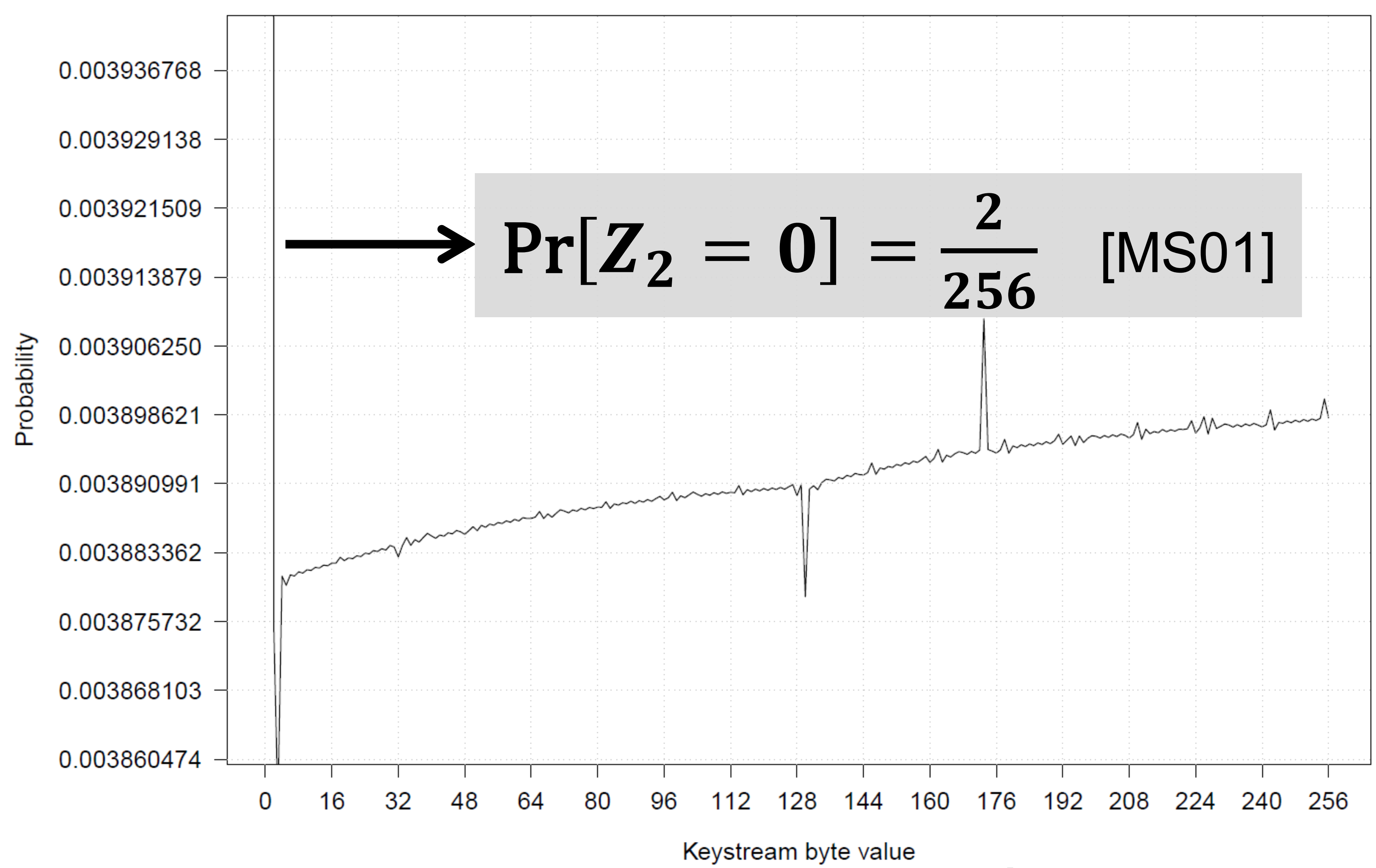
- Problem：The distribution is biased
  - Fluhrer-McGrew biases
    - Two consecutive bytes are biased towards certain values
  - Mantin's ABSAB biases

huangwc@ustc.edu.cn

3. Stream cipher and RC4
RC4

- Problem： Short-Term Biases

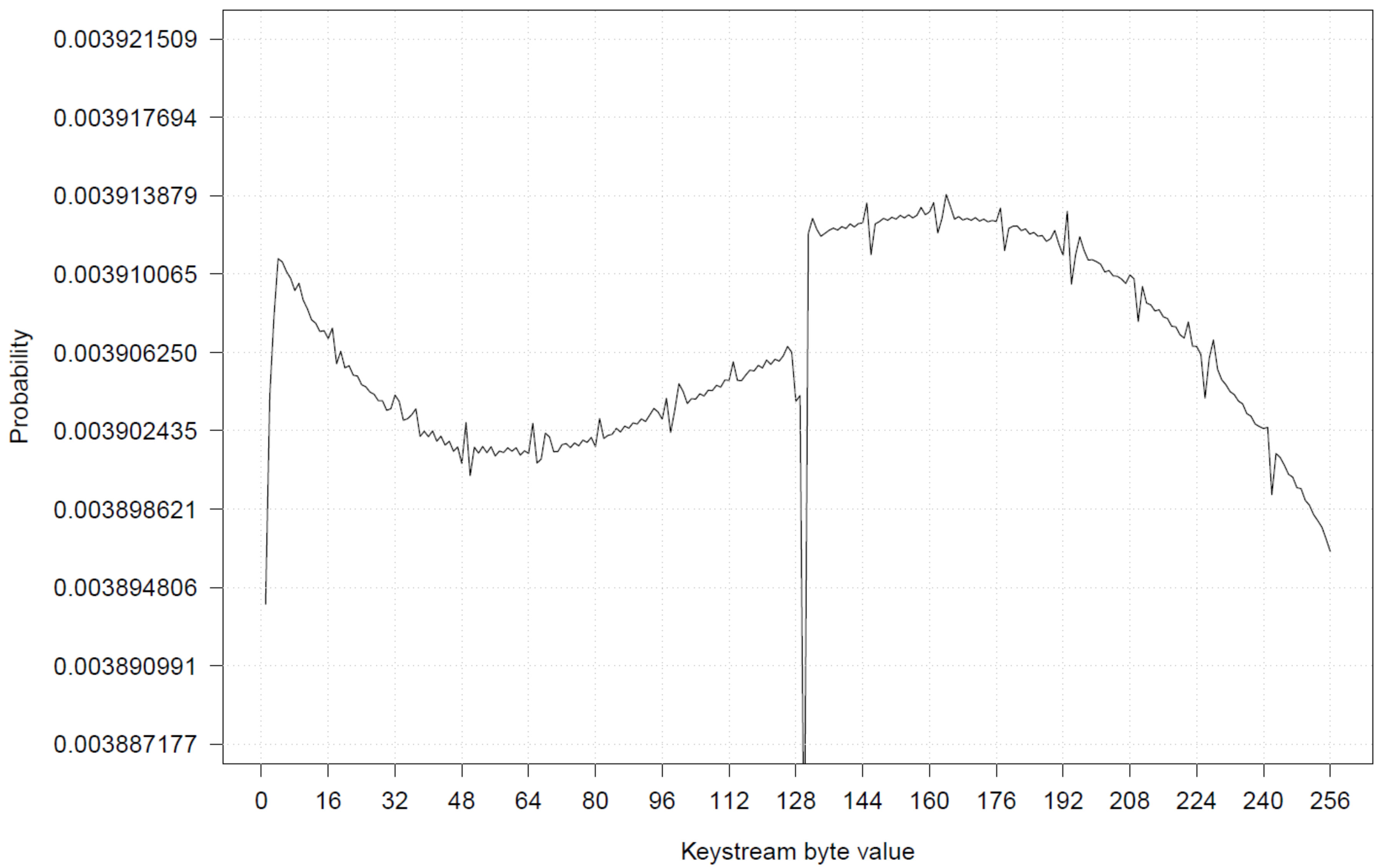
Distribution keystream byte 2



3. Stream cipher and RC4
RC4

- Problem： Short-Term Biases

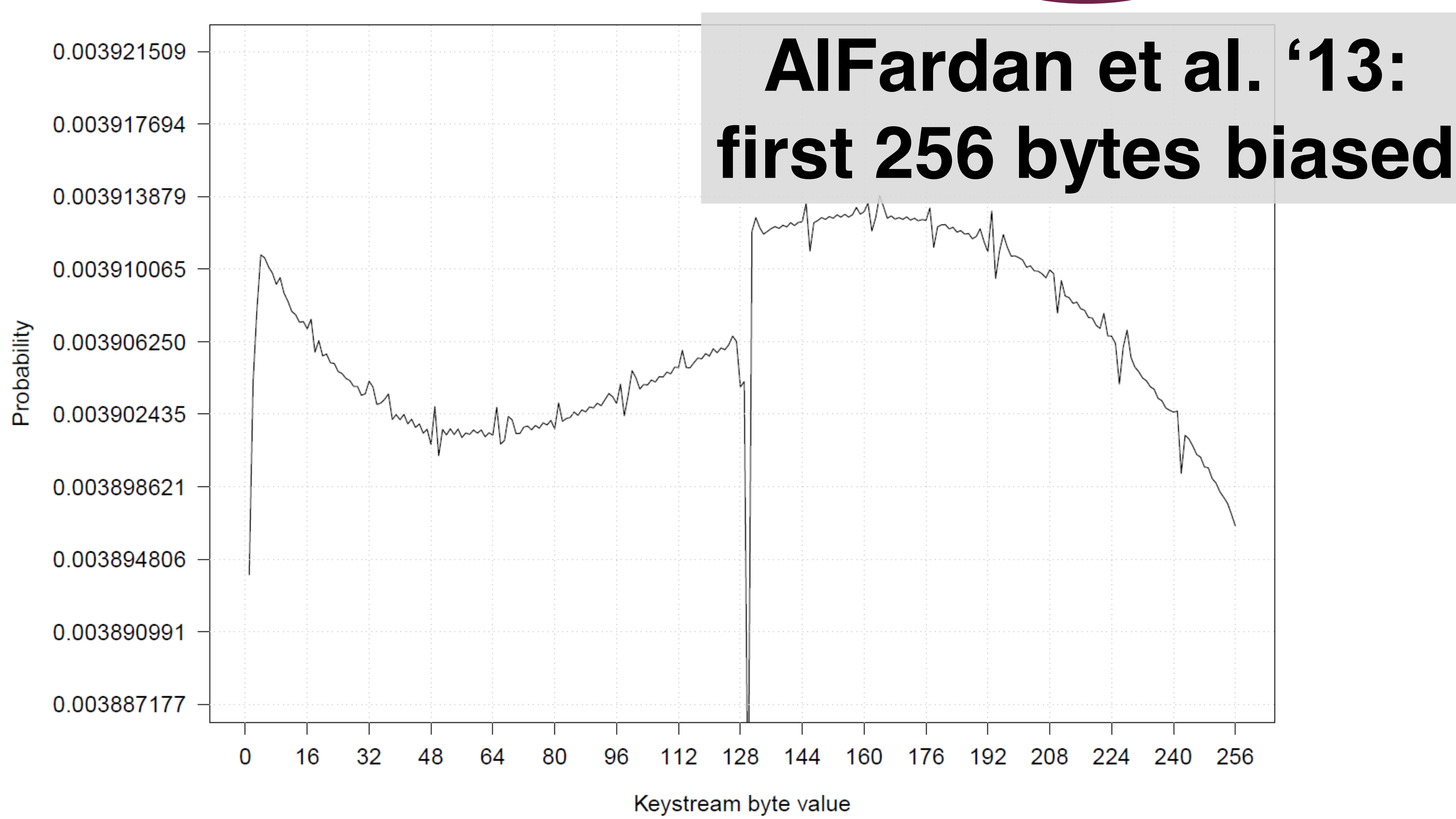
Distribution keystream byte 1



3. Stream cipher and RC4
RC4

- Problem： Short-Term Biases

Distribution keystream byte 1 (to 256)



3. Stream cipher and RC4
RC4

- Problem： **Long**-Term Biases

Fluhrer-McGrew (2000):

- Some consecutive values are biased

Examples: $(0, 0)$ and $(0, 1)$

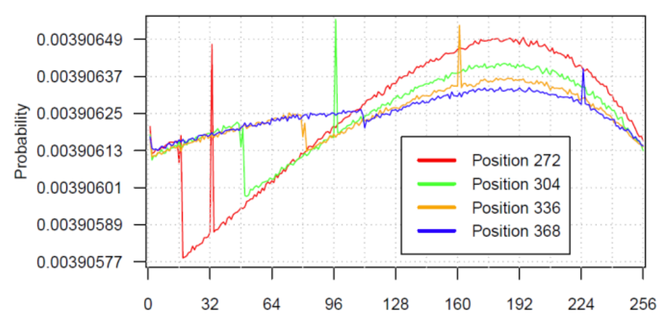
Mantin's ABSAB Bias (2005):

- A byte pair (A, B) likely reappears

3. Stream cipher and RC4

RC4

- RC4 NOMORE attack (Usenix Security '15)



New Biases

$$\lambda_{\widehat{\mu}} = (1-\alpha(g))^{|\mathcal{C}|-|\widehat{u}|} \cdot \alpha(g)^{|\widehat{\mu}|}$$

Plaintext Recovery

Break WPA-TKIP

Attack HTTPS

3. Stream cipher and RC4
RC4

- RC4 NOMORE attack (Usenix Security ’15)
  - Assuming there’s **surrounding** known plaintext
  - Cracking WPA-TKIP: an hour
  - HTTPS-cookie: 75 hours, $9*2^{27}$ request, 4450 r/s

3. Stream cipher and RC4
RC4

- RC4 NOMORE attack (Usenix Security ’15)

  - HTTPS attack (Idea： modifying HTTP request)

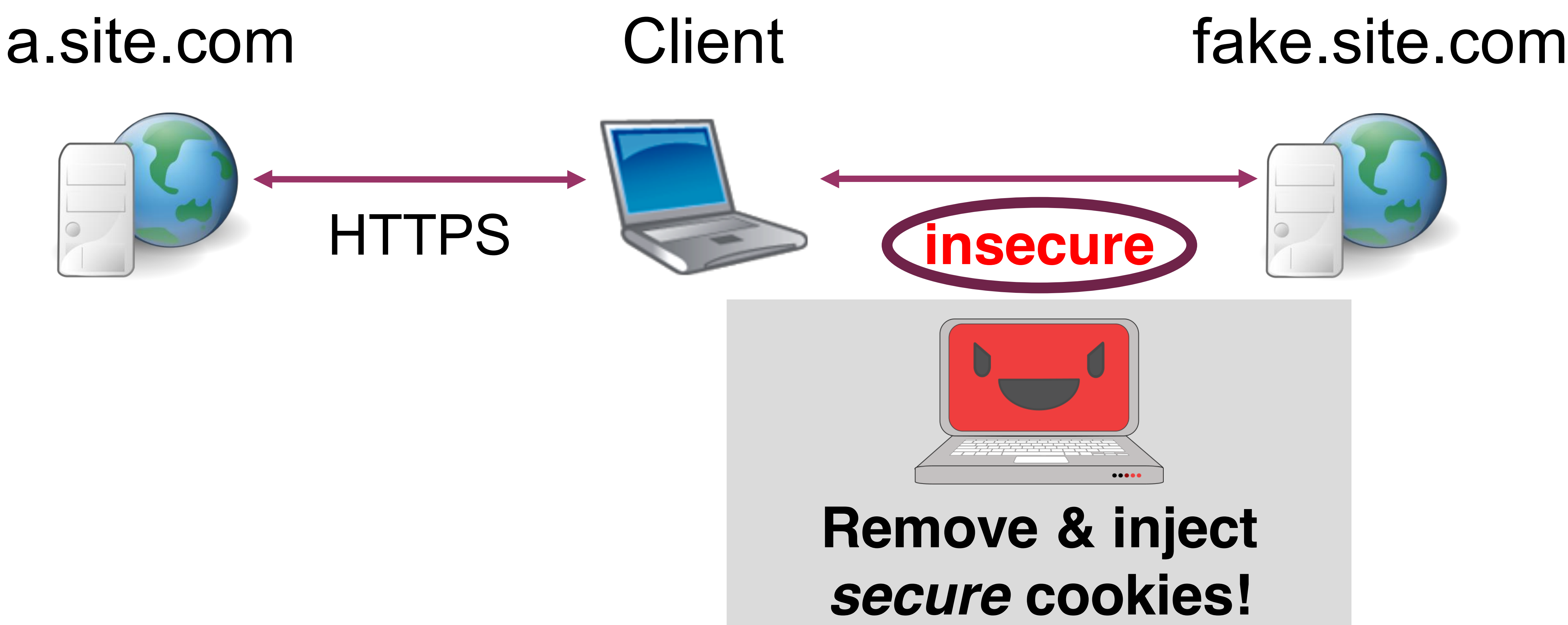
```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;
Trident/7.0; rv:11.0) like Gecko
Host: a.site.com
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: auth=????????????????; P=aaaaaaaaaaaaaaaaa
```

Headers are predictable

Surrounded by known plaintext at both sides

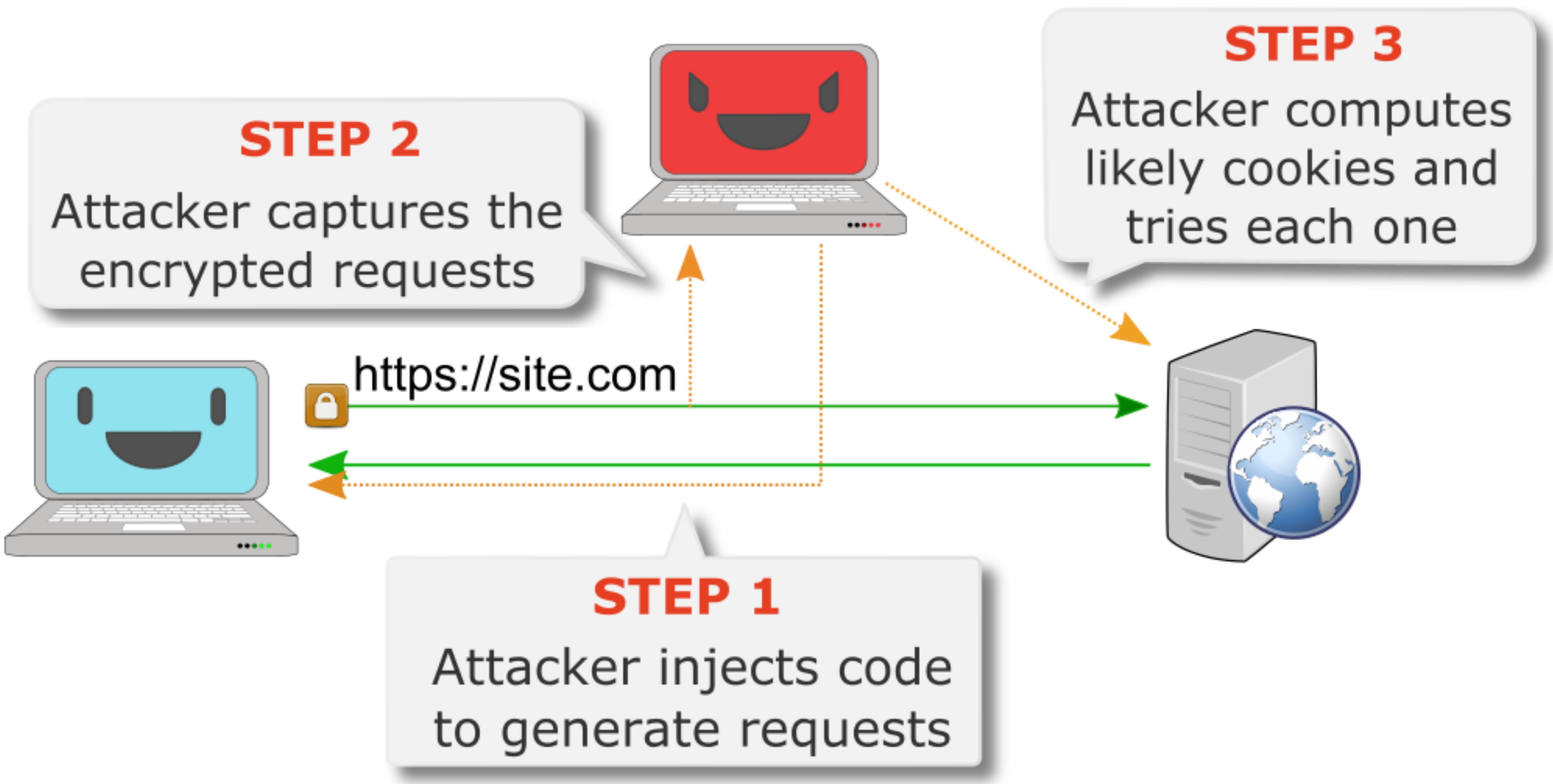
3. Stream cipher and RC4
RC4

- RC4 NOMORE attack (Usenix Security '15)
  - HTTPS attack



3. Stream cipher and RC4
RC4

- RC4 NOMORE attack (Usenix Security '15)
  - HTTPS attack



3. Stream cipher and RC4
RC4

- RC4 NOMORE attack (Usenix Security ’15)
  - http://www.rc4nomore.com
  - video

Homework

6.4 With the ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C_1 (Figure 6.4) obviously corrupts P_1 and P_2.

a. Are any blocks beyond P_2 affected?

b. Suppose that there is a bit error in the source version of P_1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?

6.8 If a bit error occurs in the transmission of a ciphertext character in 8-bit CFB mode, how far does the error propagate?

6.10 In discussing the CTR mode, it was mentioned that if any plaintext block that is encrypted using a given counter value is known, then the output of the encryption function can be determined easily from the associated ciphertext block. Show the calculation.

7.8 What RC4 key value will leave S unchanged during initialization? That is, after the initial permutation of S, the entries of S will be equal to the values from 0 through 255 in ascending order.