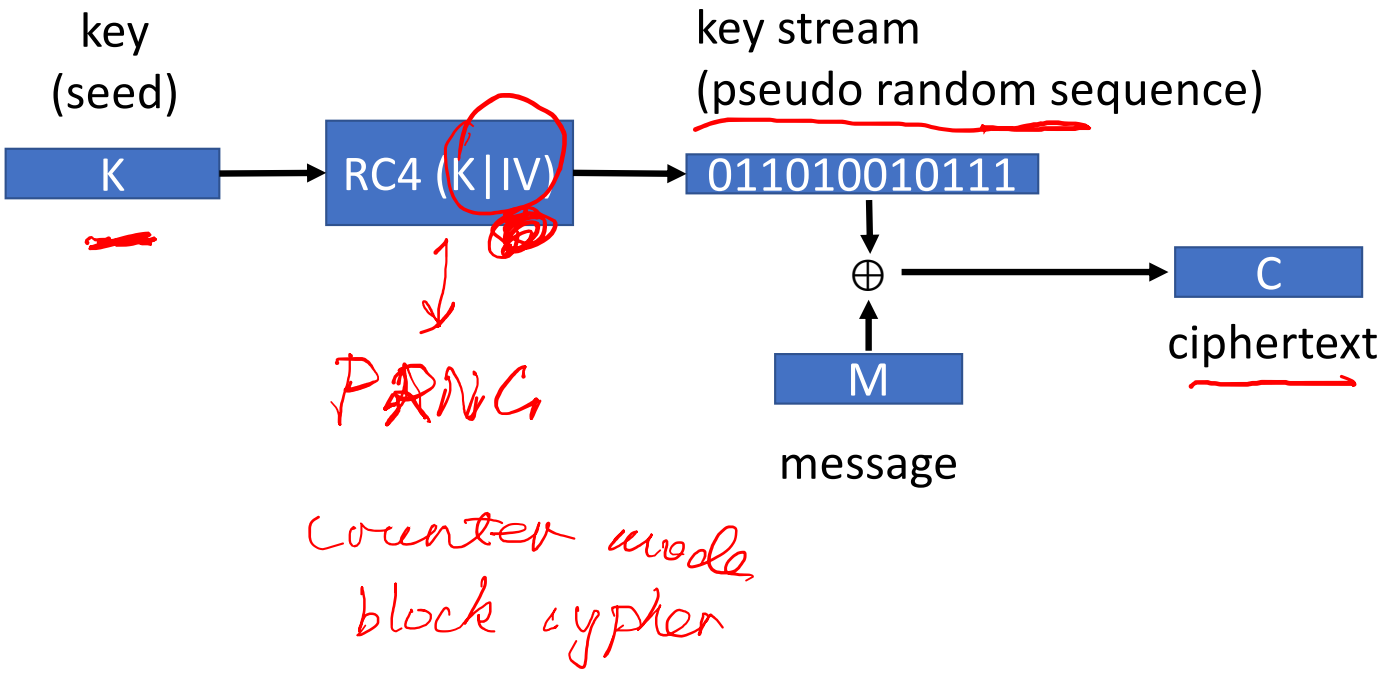
RC4 Stream Cipher



RC4 Key Schedule

- starts with an array S of numbers: 0…255
- use key to well and truly shuffle
- S forms internal state of the cipher
- given a key k of length l bytes

```
/* 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]);
```

Throw away T & K, retain S

$T[i] = K[i \bmod keylen]$ $K = [1, 2, 3, 4]$ $keylen = 4$

If $keylen > i$; $T[i] = [1, 2, 3, 4]$ $0 \bmod 4 = 0 \Rightarrow K[0]$

$1 \bmod 4 = 1 \Rightarrow K[1]$

$T[i] = K[i]$

If $keylen \leq i$ $T[4] = K[4 \bmod 4] = K[0] = 1$

$T[5] = K[5 \bmod 4] = K[1] = 2$

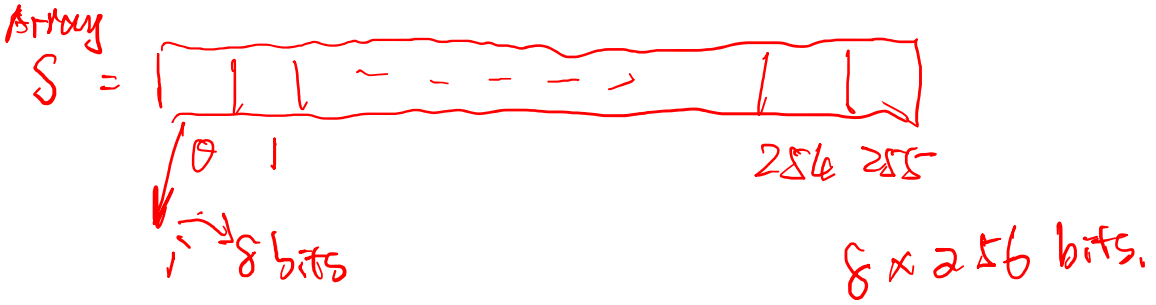
interesting $[0, 255]$ $T = [1, 2, 3, 4, 1, 2]$ 3 4, 12 > 4

repeat pattern, ⇒ reuse key

∴ add IV to be large enough

RC4 Encryption

- encryption continues shuffling array values
- sum of shuffled pair selects "stream key" value
- XOR with next byte of message to en/decrypt




```
i = j = 0
for each message byte Mi
    i = (i + 1) (mod 256)
    j = (j + S[i]) (mod 256)
    swap(S[i], S[j])
    t = (S[i] + S[j]) (mod 256)
    Ci = Mi' XOR S[t]
```

relates

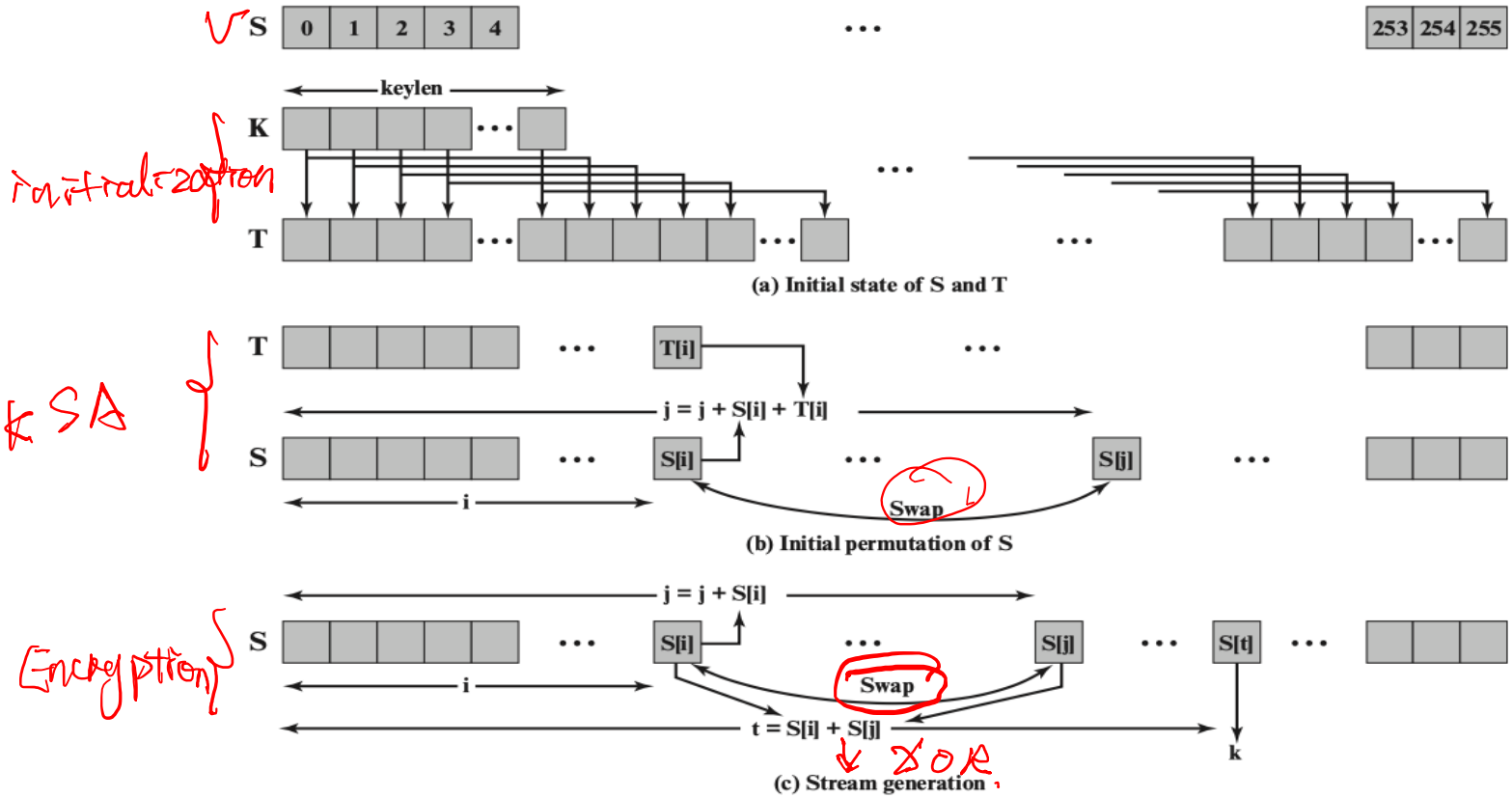
diffusion

8 bit byte

new key byte

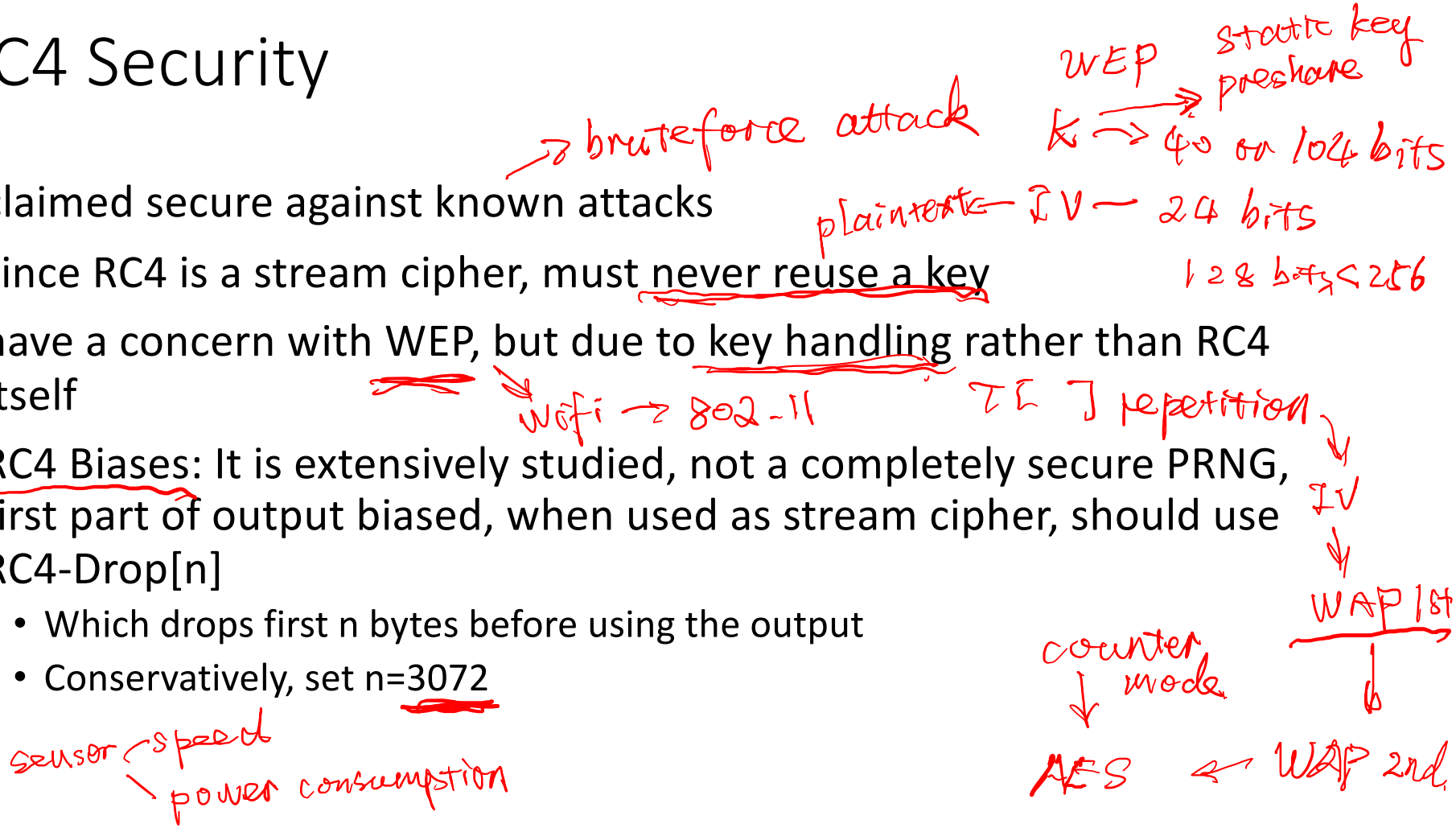
XOR

RC4


S
0
1
2
3
4
253
254
255
keylen
K
T
(a) Initial state of S and T
T
T[i]
j = j + S[i] + T[i]
S
S[i]
S[j]
i
Swap
(b) Initial permutation of S
j = j + S[i]
S
S[i]
S[j]
S[t]
i
Swap
t = S[i] + S[j]
k
(c) Stream generation

RC4 Security

- claimed secure against known attacks
- since RC4 is a stream cipher, must never reuse a key
- have a concern with WEP, but due to key handling rather than RC4 itself
- RC4 Biases: It is extensively studied, not a completely secure PRNG, first part of output biased, when used as stream cipher, should use RC4-Drop[n]
  - Which drops first n bytes before using the output
  - Conservatively, set n=3072



Summary – Chapter 2

- Symmetric block cipher
  - DES, 3DES
  - AES
- Random number
  - true random number
  - pseudorandom number
- Stream cipher
- The security of symmetric encryption depends on the secrecy of the key
- Symmetric encryption: pros and cons

Modular Arithmetic

- Definition (congruent modulo):
  - given b – a = km for some k ϵ Z, then a $\equiv b$ (mod m)
- Given a $\equiv b$ (mod m) and c $\equiv d$ (mod m), then
  - a + b ≡ c + d (mod m)
  - a - b ≡ c - d (mod m)
  - a + c ≡ b + d (mod m)
  - a × c ≡ b × d (mod m)
  - $a^k \equiv b^k$ (mod m)
  - ka = kb (mod m)
  - p(a) ≡ p(b) (mod m), any polynomial *p*(*x*) with integer coefficients
- $A \oplus B \oplus B$ = A

Thank you!

Network Security

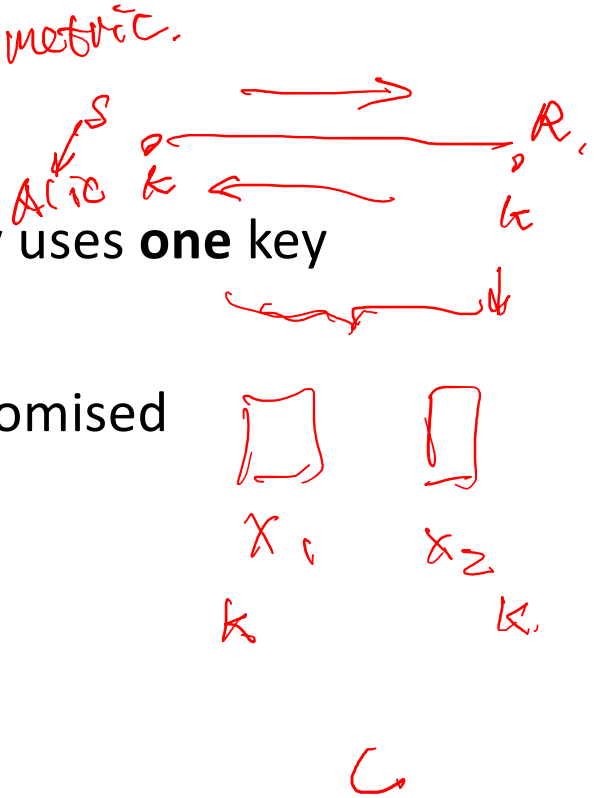
Chapter 3

Public-Key Cryptography and Message Authentication

Public-Key Cryptography

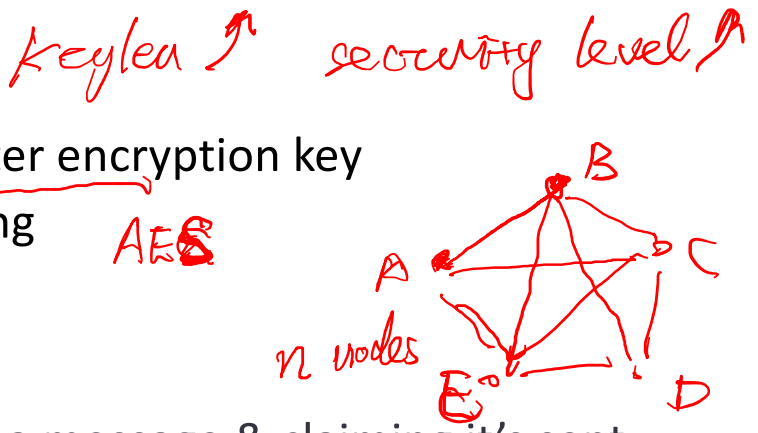
Conventional cryptography

- traditional **private/secret/single-key** cryptography uses **one** key
- shared by both sender and receiver
- if this key is disclosed, communications are compromised
- also is **symmetric**, parties are equal



Pros and cons

- Pros:
  - Encryption is fast for large amounts of data
  - Provide the same level of security with a shorter encryption key
  - By now, it's unbreakable to quantum computing
- Cons
  - Key distribution assumes a secure channel
  - Does not protect sender from receiver forging a message & claiming it's sent by sender
  - It does not scale well for large networks. It requires a separate key for each pair of communicating parties, which can result in a large number of keys to manage and protect.



Homework 1 - individual

- Chapter 1 & 2
- **Deadline**: Tuesday, October 8, 11:59 PM
- Submit your homework via the provided link.
- The Google submission timestamp will be considered final.
- A 10% penalty will be applied for each day of late submission.

Review & Quiz I

- Chapter 1 & 2
- Wednesday (Oct. 9, 2024), in class
- Please ensure your participation
- No make-up quiz