MH4311 Cryptography

Lecture 9

Stream Cipher

Wu Hongjun

Lecture Outline

- Classical ciphers
- Symmetric key encryption
  - One-time pad & information theory
  - Block cipher
  - **Stream cipher**
    - **Block cipher based stream cipher**
    - **Dedicated stream cipher**
- Hash function and Message Authentication Code
- Public key encryption
- Digital signature
- Key establishment and management
- Introduction to other cryptographic topics

Recommended Reading

- HAC Chapter 6
- Wikipedia
  - Stream cipher
    http://en.wikipedia.org/wiki/Stream_cipher
  - A5/1
    http://en.wikipedia.org/wiki/A5/1
  - RC4
    http://en.wikipedia.org/wiki/RC4
  - eSTREAM
    http://en.wikipedia.org/wiki/ESTREAM

One-time pad→more practical ciphers

- The cost of generating/distributing key material of one-time pad is high
  - Two approaches to improve one-time pad:

Shift cipher → Vigenere cipher → One-time pad

One-time pad → Stream cipher

One-time pad → Quantum encryption scheme

One-time pad:

key length = message length

Stream cipher:

generate a long keystream from a short key and IV
use this long keystream to encrypt/decrypt message

Quantum encryption:

generate a long keystream with quantum uncertainty principle, a short key is required for ensuring the authenticity of keystream

Stream Cipher

- **Advantages of stream ciphers**
  - **Keystream can be precomputed for most of the stream ciphers (such as block cipher in CTR and OFB modes)**
    - **Encryption/decryption can be extremely fast when plaintext or ciphertext arrived (only XOR)**
      - **Suitable for real-time applications**
    - **Keystream is generated at a secure place, and keystream is used at a less secure place for encryption/decryption**
      - **Suitable for some military applications**

Stream Cipher

- **Advantages of stream ciphers (cont.)**
  - **No partial block problem**
  - **Dedicated stream cipher can be much more efficient than block cipher for the same security level**

Construction of Stream Cipher

Constructions of stream cipher

- Block cipher based stream ciphers
- Dedicated stream ciphers

Block cipher based stream cipher

- **The following block cipher modes are stream ciphers:**
  - **CFB**
  - **OFB**
  - **CTR**

The above stream ciphers are only as efficient as block cipher

Dedicated stream ciphers*

- Two Examples
  - RC4 *
    - Internet communication, ….
  - A5/1 *
    - Mobile GSM communication

Stream Cipher A5/1*

- **Linear feedback shift register (LFSR)**
  - **Normally defined by a primitive polynomial over GF(2), called feedback polynomial or characteristics polynomial**
    - **A polynomial over GF(2) (with degree *n*) is primitive if it has order 2^n-1.**
      - **The order of a polynomial $f(x)$ for which $f(0)$ is not 0 is the smallest integer *e* for which $f(x)$ divides x^e+1.**

      Example: x^2+x+1 has order 3 since $(x^2+x+1)(x+1)=x^3+1$.

      $2^2-1 = 3$, so x^2+x+1 is primitive
    - **A primitive polynomial is also irreducible**

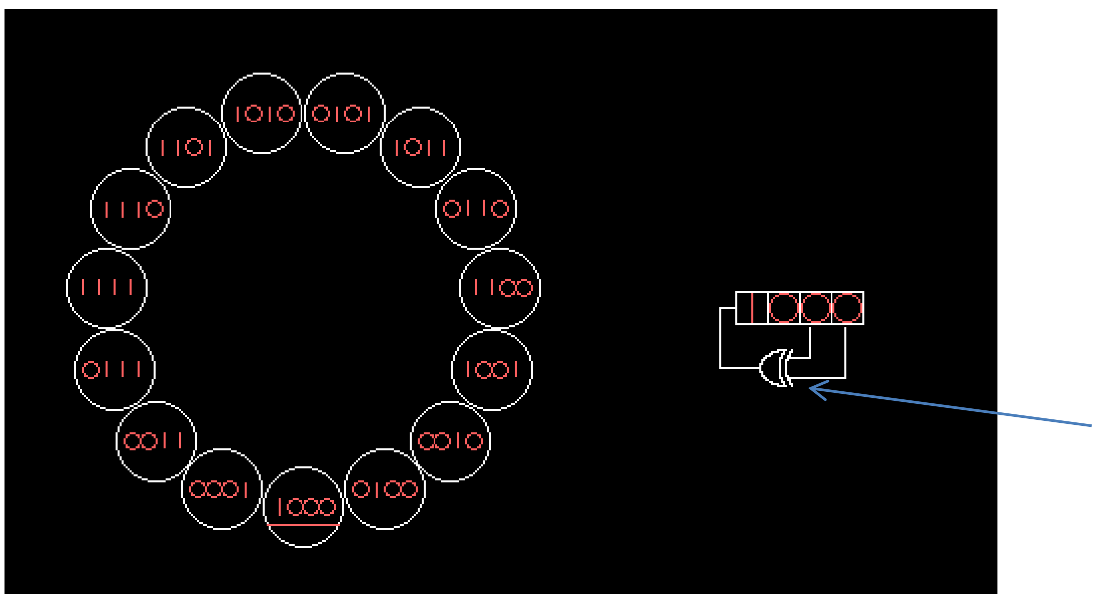
Stream Cipher A5/1*

- **Linear feedback shift register (LFSR)**
  - **Example: LFSR with primitive polynomial**

$$x^4 + x^3 + 1$$

  - **The period of the above LFSR is maximal: 2^4-1**

… → 1000 → 0100 → 0010 → 1001 → 1100 → …



XOR operation

Stream Cipher A5/1*

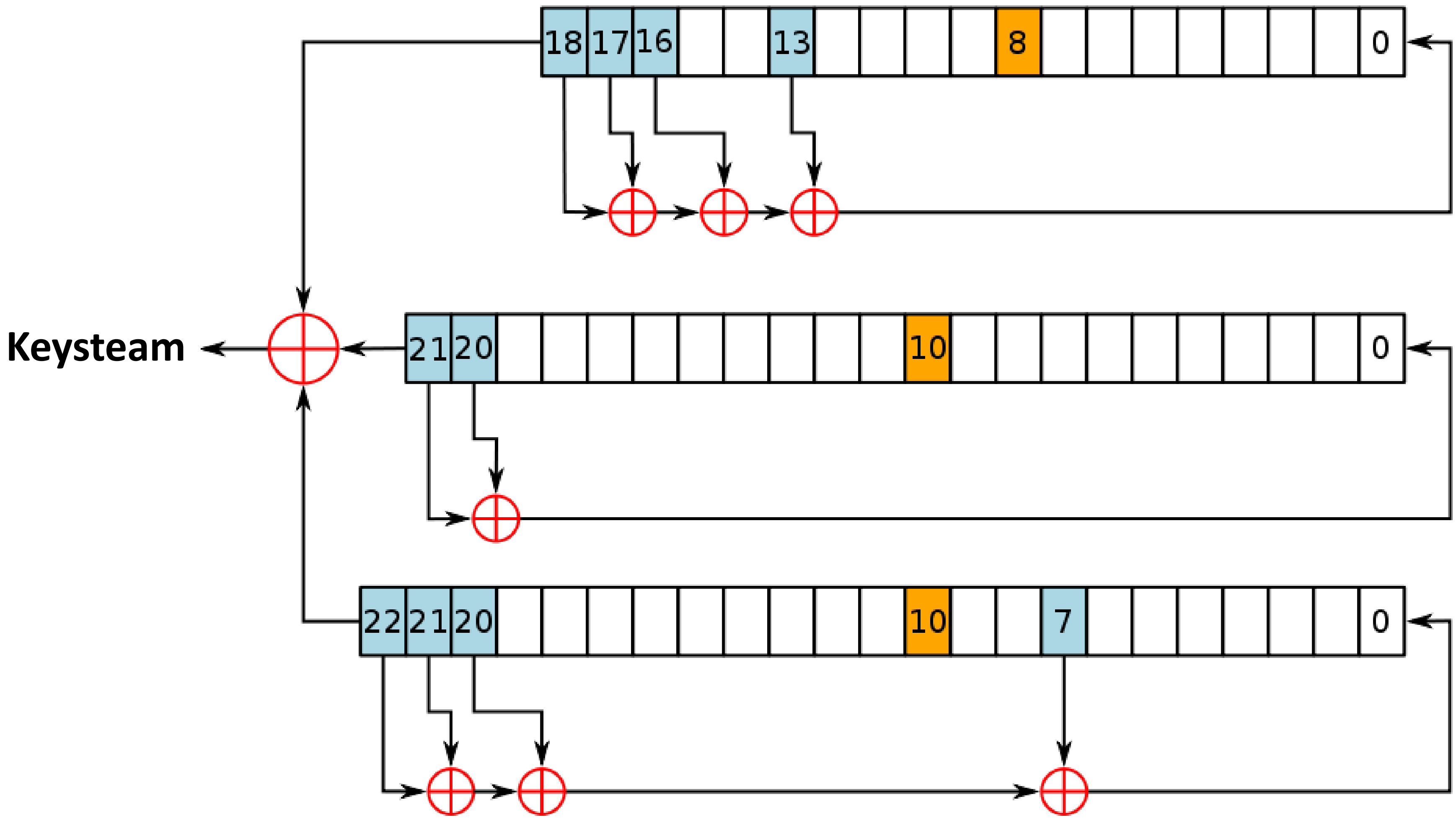
- **A5/1 used for GSM mobile network (1987 -- present)**
  - **Simple structure**
  - **To strengthen LFSR by clocking LFSR irregularly, so as to introduce nonlinearity into the LFSR**
  - **Reasonable security**
    - **Some western European countries wanted a strong cipher, but some did not want …**
    - **Key size reduced from 64 bits to 54 bits (10 key bits being set to 0)**
    - **But with only 64-bit state in order to reduce the hardware cost**

Stream cipher A5/1*

- Three irregularly clocked LFSRs
- At each step, each LFSR provides one clocking bit
  - Compute the majority of those three bits
  - If the clocking bit of an LFSR is the majority, clock that LFSR. (each step, at least two LFSRs get clocked).

LFSR number	Length in bits	Characteristic polynomial	Clocking bit	Tapped bits
1	19	$x^{19} + x^5 + x^2 + x + 1$	8	13, 16, 17, 18
2	22	$x^{22} + x + 1$	10	20, 21
3	23	$x^{23} + x^{15} + x^2 + x + 1$	10	7, 20, 21, 22

Stream cipher A5/1*



Stream cipher A5/1*

- **Initialization**
  - **Load the 64-bit key (10 zero bits) into the LFSRs**
    - **64 steps**
    - **At the *i*-th step, XOR the *i*-th bit of the key to the least significant bits of those three LFSRs**
  - **Load the 22-bit IV into the LFSRs**
    - **22 steps**
    - **Similar to the key loading**
  - **Run the cipher 100 steps to mix key and IV**
    - **no output for these 100 steps**

Stream cipher A5/1*

- Keystream generation
  - Update the state (clock those LFSRs according to the majority bit)
  - At each step, one keystream bit is generated
  - Only 228 bits are generated from each IV
    - First 114 bits for decrypting the received packet
    - Last 114 bits for encrypting the outgoing packet
    - About 217 packets/second

Stream cipher A5/1*

- **A5/1 is dedicated to mobile communication**
  - **The most widely used hardware stream cipher**
- **Not suitable for other applications**
  - **64-bit key size (small)**
  - **22-bit IV size (small)**
  - **64-bit state size (too small)**
  - **Small keystream period**
    - **since the small state is updated in an non-invertible way**

RC4: Another widely used stream cipher*

- **RC4**
  - **The most widely used software stream cipher**
    - **Such as in SSL/TLS**
  - **Designed by Ron Rivest (1987)**
  - **Extremely simple**
  - **Generally strong**
    - **But weak when a key is used with different IVs**

Stream Cipher RC4*

- **The state**
  - **A secret table *S* with 256 elements + 2 indices**
  - **Each element is one-byte**

RC4: Initialization*

```
for i from 0 to 255
    S[i] = i
endfor
j = 0
for i from 0 to 255
    j = (j + S[i] + key[i mod keylength]) mod 256
    swap values of S[i] and S[j]
endfor
```

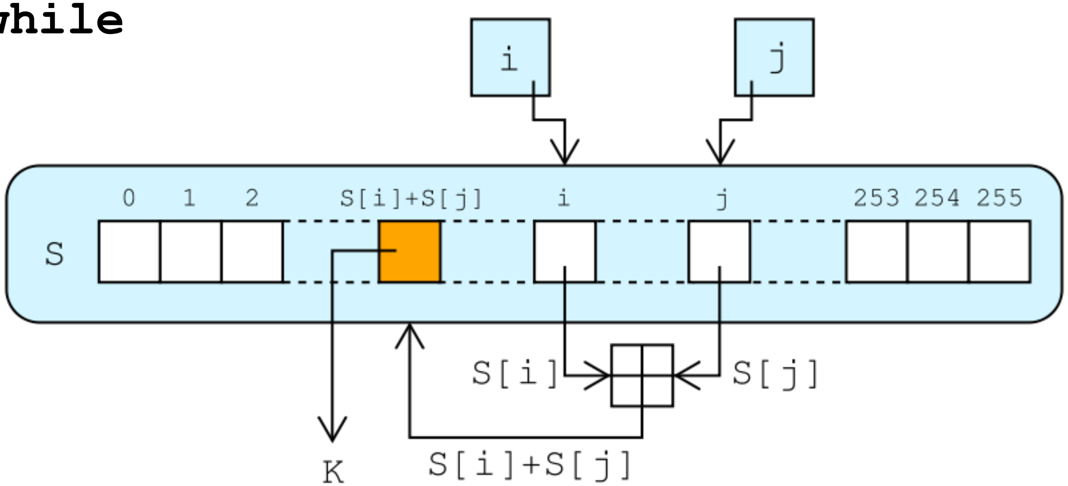
- **In the above initialization, there is no IV**
- **If an IV is used, it is considered as part of the key (with increased key length)**
  - **The above initialization is insufficient to mix well the key & IV**

RC4: Keystream generation*

```
i = 0
j = 0
while Generating Keystream:
      i := (i + 1) mod 256
      j := (j + S[i]) mod 256
      swap values of S[i] and S[j]
      Keystream[i] := S[(S[i] + S[j]) mod 256]
endwhile
```

Update the state

Generate keystream



Keystream[i]

Recent Developments on Stream Cipher

- **eSTREAM project (2004 -- 2008)**
  - **The stream cipher project of the European Network of Excellence for Cryptology (ECRYPT)**
  - **To identify secure & efficient stream ciphers**
  - **Around 35 submissions (around 50 ciphers)**
    - **7 winners**
    - **4 for software: HC-128, Rabbit, Salsa20/12, SOSEMANUK**
    - **3 for hardware: Grain, MICKEY, Trivium**
    - **Only SOSEMANUK is based on LFSR, the other winners are based on nonlinear state update functions**

Summary

- One-time pad → stream cipher
- Two main constructions
  - Block cipher based stream cipher
    - CFB, OFB, CTR
  - Dedicated stream cipher*
- Two dedicated stream cipher examples
  - A5/1*
  - RC4*