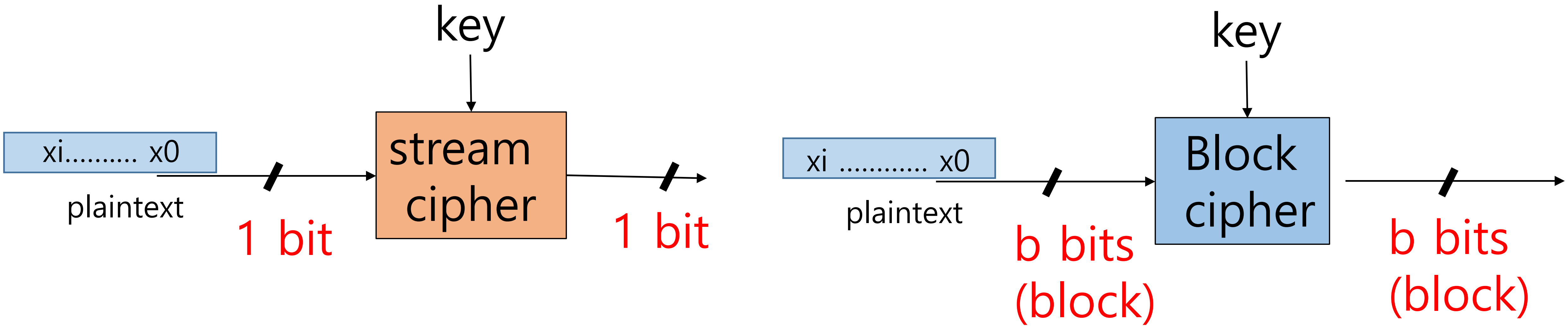
Symmetric Crypto: Stream Ciphers

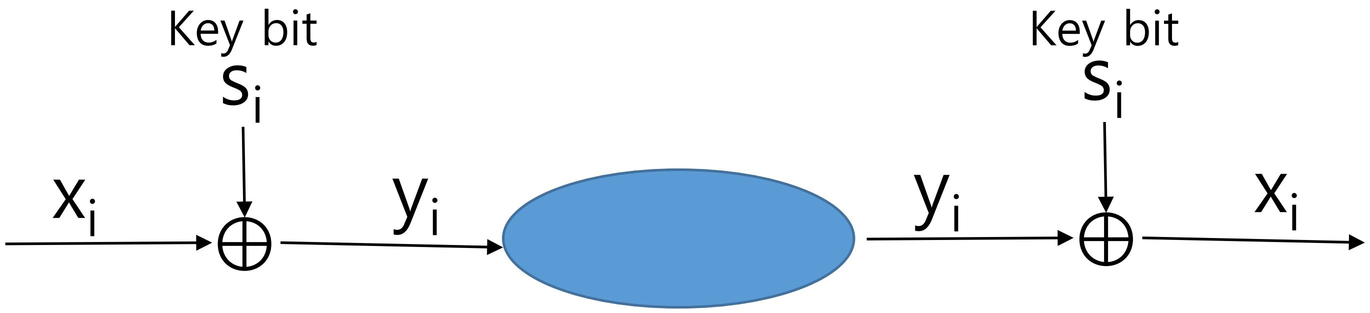
2019. 3. 11

Contents

Stream Cipher vs. Block Cipher



Encryption/Decryption of Stream cipher



$x_i, y_i, s_i \in \{0,1\}$
encryption : $y_i = E_{si}(x_i) \equiv x_i + s_i \bmod 2$
dncryption : $x_i = D_{si}(y_i) \equiv y_i + s_i \bmod 2$

Modulo 2 addition

- Modulo 2 addition is equivalent to the XOR operation.

x_i	s_i	$y_i \equiv x_i + s_i \bmod 2$	$y_i \equiv x_i \oplus s_i$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

- If the key bit s_i behaves perfectly randomly, y_i is unpredictable with a 50% chance of being 0 or 1.
- If we have perfect random numbers, the stream cipher can achieve the unconditional security.

Key Stream

- The security of the stream cipher completely depends on the key stream.
- Then, the question is how or whether we can generate the random key stream.

True Random Number Generators

- A sequence of bits cannot be reproduced.
- The true RNG is based on physical processes such as coin flipping, dice rolling, semiconductor noise, radioactive decay, and so on.

Pseudorandom Number Generators(PRNG)

- The bits are generated recursively from an initial seed value.

$$s_0 = \text{seed}$$
$$s_{i+1} = f(s_i),\ i=0,1,\ldots$$

- Popular example: the linear congruential generator

$$s_0 = \text{seed}$$
$$s_{i+1} = as_i+b \bmod m,\ \ i=0,1,\ldots$$

Cryptographically Secure PRNG(CSPRNG)

- CSPRNG is PRNG which is unpredictable.
  - Given n output bits of the key stream $s_i, s_{i+1}, \ldots, s_{i+n-1}$, it is computationally infeasible to compute the sequence bits s_{i+n}, $s_{i+n+1}, \ldots$
  - In other words, given n consecutive bits of the key stream, there is no polynomial time algorithm that can predict the next bit s_{n+1} with the better than 50% chance of success.

One-Time Pad (OTP)

- OTP
  - The key stream is generated by a true random number generator,
  - The key stream is only known to the legitimate communicating parties,
  - Every key stream bit s_i is only used once.
- Provably secure
- Unconditional security

OTP requirements

- OTP requires True RNG, so it needs a device that can generate true random number.
- The sender have a mean to deliver the bits to the receiver.
- Key stream cannot be reused. OTP need one key bit for every bit of plaintext.
- Because of these requirements, OTP is rarely used in practice.
- However, it gives us a design idea for secure cipher.

Stream cipher

- Plaintext is XORed with keystream generated from secret key and initialization vector (IV)
  - Vernam cipher (one-time pad)
  - RC4, Seal
  - *linear feedback shift registers* (LFSR)
- Here we explain the algorithm using A5/1.
  - Based on shift registers
  - Used in GSM mobile phone system

A5/1: Shift Register

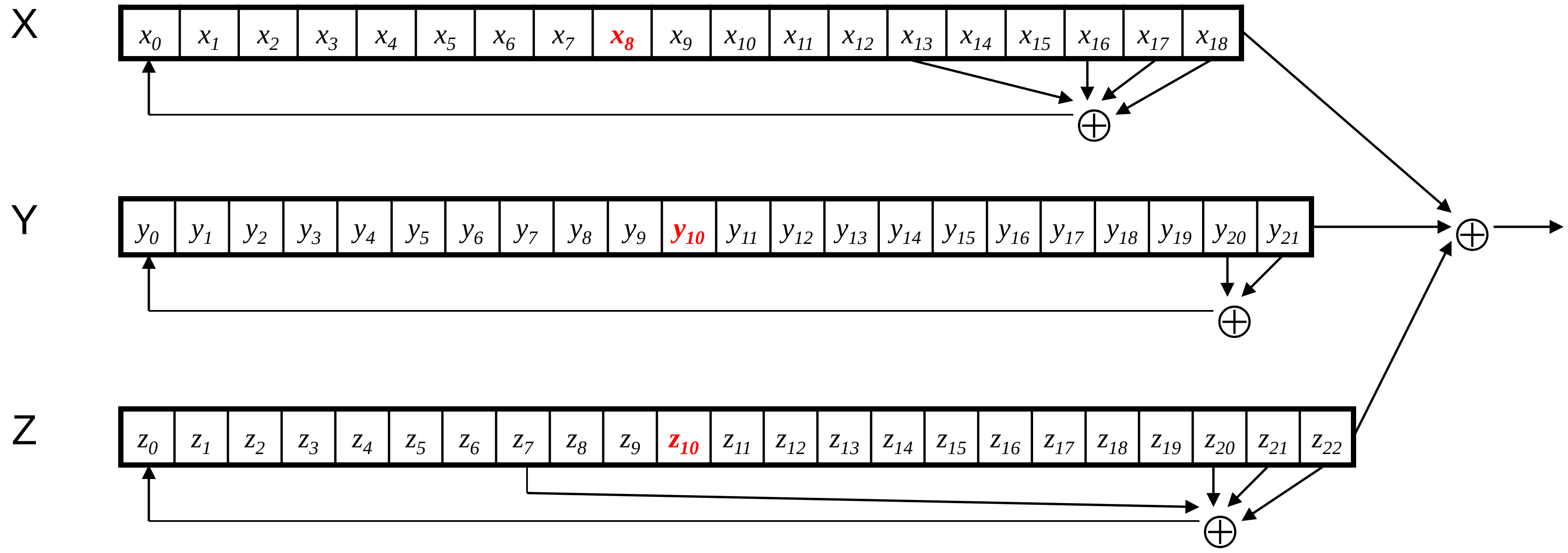
- A5/1 uses 3 *linear feedback shift registers*
  - X: 19 bits $(x_0, x_1, x_2, \ldots, x_{18})$
  - Y: 22 bits $(y_0, y_1, y_2, \ldots, y_{21})$
  - Z: 23 bits $(z_0, z_1, z_2, \ldots, z_{22})$
  - X+Y+Z = 64 bits

A5/1: Keystream

- At each step: $m = \mathrm{maj}(x_8, y_{10}, z_{10})$
  - Examples: $\mathrm{maj}(0,1,0) = 0$ and $\mathrm{maj}(1,1,0) = 1$
- If $x_8 = m$ then X *steps*
  - $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
  - $x_i = x_{i-1}$ for $i = 18,17,\ldots,1$ and $x_0 = t$
- If $y_{10} = m$ then Y *steps*
  - $t = y_{20} \oplus y_{21}$
  - $y_i = y_{i-1}$ for $i = 21,20,\ldots,1$ and $y_0 = t$
- If $z_{10} = m$ then Z *steps*
  - $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
  - $z_i = z_{i-1}$ for $i = 22,21,\ldots,1$ and $z_0 = t$
- Keystream bit is $x_{18} \oplus y_{21} \oplus z_{22}$

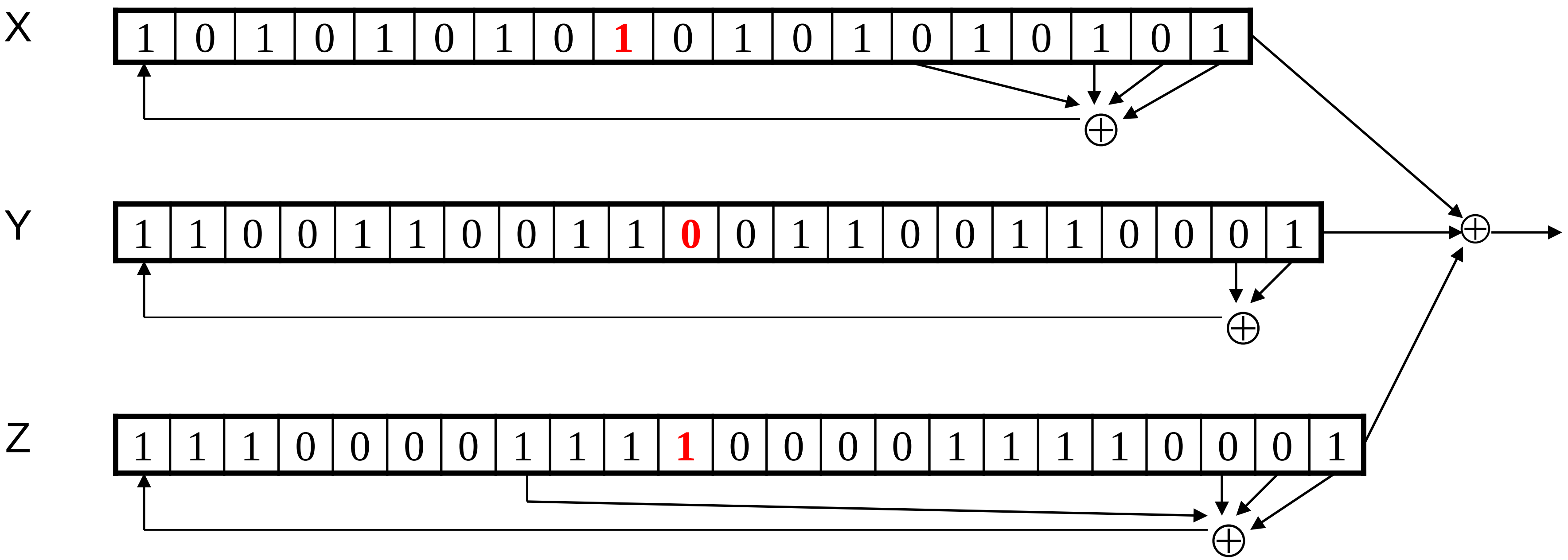
A5/1

- Each variable here is a single bit
- Key is used as **initial fill** of registers
- Each register steps (or not) based on $\mathrm{maj}(x_8, y_{10}, z_{10})$
- Keystream bit is XOR of rightmost bits of registers



A5/1

- In this example, $m = \mathrm{maj}(x_8, y_{10}, z_{10}) = \mathrm{maj}(1,0,1) = 1$
- Register X steps, no Y steps, and Z steps
- Keystream bit is XOR of right bits of registers
- Here, keystream bit will be 0 ⊕ 1 ⊕ 0 = 1



Shift Register Crypto

- Shift register crypto efficient in hardware
- Often, slow if implement in software
- In the past, very popular
- Today, more is done in software due to fast processors
- Shift register crypto still used some
    - Resource-constrained devices

Looking back on Stream Ciphers

- Stream ciphers tends to be small and fast, so it is beneficial for applications with little computational resources.
- LFSR-based algorithms are hardware-oriented ciphers.
  - A5/1 was used in GSM mobile networks for voice encryption between cell phones and base station.
  - A5/1 and A5/2 were broken.
- Once, it was considered to be more efficient than block ciphers. But these days block ciphers are as efficient as stream ciphers for either software or hardware-optimized implementation.
- Then, is the stream cipher dead?