Stream Cipher

- Introduction
  - Same P ≠ Same C, memory, state cipher
  - bit-by-bit Exor with pt and key stream
    \( c_i = m_i \oplus z_i \)
  - Encryption = Decryption → Symmetric
  - Use LFSR (Linear Feedback Shift Register)
  - (external) Synchronous or self-synchronous
  - Security measure: Period of key stream, LC (Linear Complexity)

Sequence

- Def)
  - \( s = s_0, s_1, \ldots \): infinite seq.,
  - \( n \) term of \( s \): \( s^n = s_0, s_1, \ldots, s_{n-1} \)
  - if \( s_i = s_{i+n} \) for all \( i \geq 0 \), \( s \) is periodic seq. having period \( n \).
  - run: subsequence of consecutive ‘0’ (gap) or consecutive ‘1’ (block)
Golomb’s postulates(I)

$s^N$: periodic seq. of N

1. for a cycle of $s^N$, 0-1 balanceness, i.e.,
   $$\#\{s_i=1\} - \#\{s_i=0\} = 1$$
   for all $i \neq j < N$.

2. for a cycle of $s^N$, half the runs have length 1, 1/4 have the length 2, etc.

3. Autocorrelation* function has 2 values.
   $$N \cdot C(t) = \sum_{i=0}^{N-1} (2s_i - 1)(2s_{i+t} - 1) = \begin{cases} N, \text{ if } t = 0 \\ K, \text{ if } 1 \leq t \leq N - 1 \end{cases}$$

* Measuring similarity between original and t-shifted sequences
** If satisfies all, called as Pseudo-Noise(PN) sequence.

Golomb’s postulates(II)

(Ex) $s^{15} = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1$

1. $\#\{0\} = 7$, $\#\{1\} = 8$

2. 8 runs, 4 runs with length 1 (2 gaps, 2 blocks), 2 runs with length 2 (1 gap, 1 block), 1 run with length 3 (1 gap), 1 run with length 4 (1 block)

3. Autocorrelation function, $C(0) = 1$, $C(t) = -\frac{1}{15}$

Thus, PN-seq.
LFSR(I)

- **Notation:** \(< L, C[D] >\) where connection poly.
  \(C[D] = 1 + c_1D + c_2D^2 + \ldots + c_LD^L \in \mathbb{Z}_2[D]\)
- If \(c_1 = 1\), (i.e., \(\text{deg}(C[D]) = L\)), singular polynomial.
- If initial stage is \([s_{L-1}, \ldots, s_1, s_0]\), output seq. \(s_0, s_1, \ldots s_j = (c_1s_{j-1} + c_2s_{j-2} + \ldots + c_Ls_{j-L}) \mod 2\), \(j \geq L\)
- (Ex) \(<4, 1 + D + D^4>, \sigma_0 = [0, 1, 1, 0]\)

Finite State Machine

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Stage 0</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

LFSR(II)

<table>
<thead>
<tr>
<th>(m)</th>
<th>(k(k_1,k_2,k_3))</th>
<th>(m)</th>
<th>(k(k_1,k_2,k_3))</th>
<th>(m)</th>
<th>(k(k_1,k_2,k_3))</th>
<th>(m)</th>
<th>(k(k_1,k_2,k_3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>12</td>
<td>7,4,3</td>
<td>22</td>
<td>1</td>
<td>32</td>
<td>28,27,1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>13</td>
<td>4,3,1</td>
<td>23</td>
<td>5</td>
<td>33</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>14</td>
<td>12,11,1</td>
<td>24</td>
<td>4,3,1</td>
<td>34</td>
<td>15,14,1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>15</td>
<td>1</td>
<td>25</td>
<td>3</td>
<td>35</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>16</td>
<td>5,3,2</td>
<td>26</td>
<td>8,7,1</td>
<td>36</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>17</td>
<td>3</td>
<td>27</td>
<td>8,7,1</td>
<td>37</td>
<td>12,10,2</td>
</tr>
<tr>
<td>8</td>
<td>6,5,1</td>
<td>18</td>
<td>7</td>
<td>28</td>
<td>3</td>
<td>38</td>
<td>6,5,1</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>19</td>
<td>6,5,1</td>
<td>29</td>
<td>2</td>
<td>39</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>20</td>
<td>3</td>
<td>30</td>
<td>16,15,1</td>
<td>40</td>
<td>21,19,2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>21</td>
<td>2</td>
<td>31</td>
<td>3</td>
<td>41</td>
<td>3</td>
</tr>
</tbody>
</table>

* Primitive polynomial over \(\mathbb{Z}_2: x^m + x^k + 1\) (trinomial) or \(x^m + x^{k_1} + x^{k_2} + 1\) (pentanomial) called as Maximum-Length Shift Register Seq., \(M - \text{seq.}\)

* # of monic primitive poly \(\equiv (2^m-1)/m \mod \mathbb{Z}_2[x]\) where \(\phi\) is Euler-phi ft.
LFSR(III)

- Well suited for H/W implementation
- Produce seq. of large period
- Good statistical properties
- Readily analyzed by algebraic structure
- Breakable by consecutive $2 \times L$ sequence:
  depends on computing an inverse matrix
  whose complexity is $O(L^3)$, $L$ : length of LFSR.
  -> one LFSR is useless.

Linear Complexity(I)

- (Def) LC of finite seq. $s^n$, $L(s)$ : length of shortest LFSR that generates a seq.
  having $s^n$ as its 1st $n$ terms.
  : Berlekamp-Massey algorithm
- (Properties of LC) $s,t$ : binary seq.
  - For any $n \geq 1$, $0 \leq L(s^n) \leq n$
  - $L(s^n) = 0$ iff $s^n$ is ‘0’ seq. of length $n$.
  - $L(s^n) = n$ iff $s^n=0,0,\ldots,0,1$.
  - If $s$ is periodic with period $N$, $L(s^n) \leq N$.
  - $L(s \oplus t) \leq L(s) + L(t)$
Linear Complexity(II)

- $s^n$: random seq. from all seq. of length $n$
- Expectation value of LC $E(L(s^n)) = \frac{n}{2} + \frac{4 + B(n)}{18} - \frac{1}{2^n} \left( \frac{n}{3} + \frac{2}{9} \right)$
  where $B(n) = 0$ if even $n$, otherwise 0
  for large $n$ $E(L(s^n)) \approx n/2 + 2/9$ and $\text{Var}(L(s^n)) \approx 86/81$
- (Def) LCP (Linear Complexity Profile)
  $L_N : \text{LC of } s^N=s_0,s_1,\ldots,s_{N-1}, L_1, L_2, \ldots, L_N \text{ is LCP}$

Synchronous Stream Cipher(I)

- $f$: next state ft, $\sigma_{i+1} = f(\sigma_i, k)$, $\sigma_0$: initial value
- $g$: keystream generating ft, $z_i = g(\sigma_i, k)$, $k$: key
- $h$: output ft, $c_i = h(z_i, m_i)$, $m_i$: pt, $z_i$: key stream, $c_i$: ct
Synchronous Stream Cipher(II)

1. Synchronization requirement: loss
2. No error propagation: gain
3. Active attack: (1) -> insertion, deletion or replay then lose synchronization

Need to consider other integrity check mechanisms

Self-Synchronous Stream Cipher(I)

- $s_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1})$, $s_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$: initial value
- $g$: keystream generating ft, $z_i = g(s_i, k)$, $k$: key
- $h$: output ft, $c_i = h(z_i, m_i)$, $m_i$: pt, $z_i$: keystream, $c_i$: ct

© ICU, Kwangjo Kim
Self-Synchronous Stream Cipher (II)

(1) self-synchronization: gain
(2) error propagation: loss
(3) active attack: (2) -> easy to find
    passive modification, (1) -> by active adv., difficult to find
⇒ other integrity check mechanism required.

Nonlinear Combiner (I)

LFSR 1 → LFSR 2 → ... → LFSR n → f → Keystream, z

Algebraic Normal Form (ANF): mod. 2 sum of distinct m-th order product of its variable, 0 ≤ m ≤ n
Ex) f(x_1, x_2, x_3, x_4, x_5) = 1 + x_2 + x_3 + x_4 + x_4x_5 + x_1x_2x_3x_4, deg(f) = 4
Nonlinear Combiner(II)

- **Geffe generator**

  \[ f(x_1, x_2, x_3) = x_1x_2 \oplus (1 + x_2)x_3 = x_1x_2 \oplus x_2x_3 \oplus x_3 \]

  \[ p(z) = (2^{L_1-1})(2^{L_2-1})(2^{L_3-1}) \]

  where \( L_1, L_2, \) and \( L_3 \) are relatively prime.

  \[ L(z) = L_1L_2 + L_2L_3 + L_3 : \text{Correlation attack is possible!} \]

Nonlinear Combiner(III)

- **Summation generator**

  \[ p(z) = \prod_{i=1}^{n} (2^{L_i-1}) \]

  \[ LC = p(z) \]

  But weak in 2-adic span.
Clock-controlled generator(I)

- Alternating step generator

R_1: de Bruijn seq. of period 2^{L_1}
p(z) = 2^{L_1} (2^{L_2-1}) (2^{L_3-1})
R_2, R_3: m-seq s.t., gcd(L_1, L_2) = 1

L(z): (L_2 + L_3) 2^{L_1-1} < L(z) <= (L_2 + L_3) 2^{L_1}

Clock-controlled generator(II)

- Shrinking generator

If gcd(L_1, L_2) = 1,
p(z) = (2^{L_2-1}) 2^{L_1-1}
L_2 2^{L_1-2} < L(z) < L_2 2^{L_1-1}
Other generators

- Cascade Generator
- CSPRBG (Cryptographically Secure Pseudo Random Bit Generator)
  - RSA LSB Generator
  - BBS Generator
- Pseudo-noise Generator
  - Noise Diode or Noise Transistor

Nonlinear FSR

\[ f(s_{j-1}, s_{j-2}, \ldots, s_{j-L}) \]

\[ f() : \text{nonlinear ft} \]
Security of Stream Cipher

- Period: Depends on req'd level of security
- Linear Complexity
  - shortest LFSR that generates a given seq.
- Measure against Correlation Attack
  - Correlation Immune ft
  - Nonlinear ft
- DC, LC, and DFA are applicable


Statistical Randomness

- Frequency Test
- Serial Test
- Run Test
- Poker Test
- Spectral Test
- Linear Complexity Profile
- Quadratic Complexity
Statistical Test by FIPS 140-1

For a given 20,000bit sample seq.
(I) monobit test:
The number of ‘1’ =n_1, 9,654 < n_1 < 10,346
(2) poker test:
m=4, 1.03 < X_3 < 57.4
(3) runs test: 1 ≤ i ≤ 6
(4) long run test: no run greater than 34