第 3 章：流密码

赵俊舟

junzhou.zhao@xjtu.edu.cn

2025 年 3 月 14 日

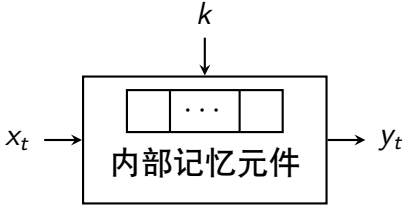
目录

目录

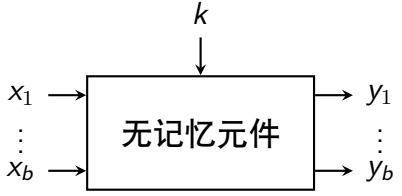
目录

流密码

- 在电话语音通信、TCP/UDP 通信等场景下，会产生持续的语音或数据等流式数据。
- 对这类数据流进行加密需要设计流密码（Stream Cipher），或序列密码（Sequential Cipher）。
- 流密码对明文消息按比特或字节逐位加密，不同于分组密码按定长数据块进行加密。



流密码

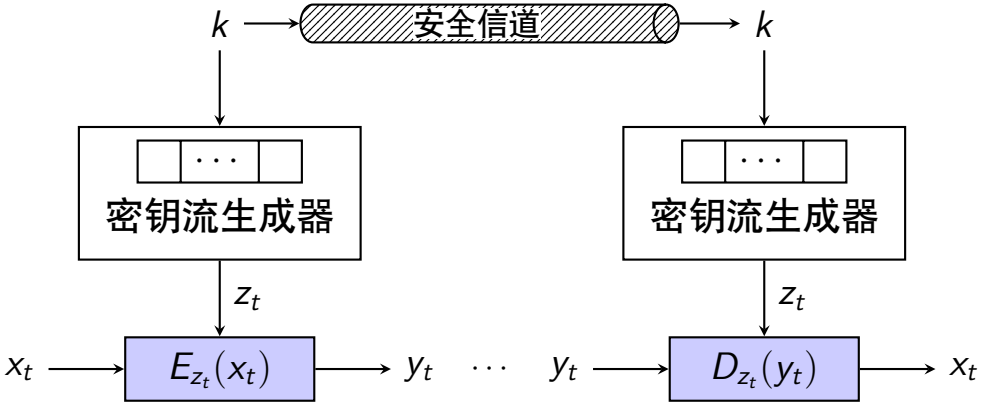


分组密码

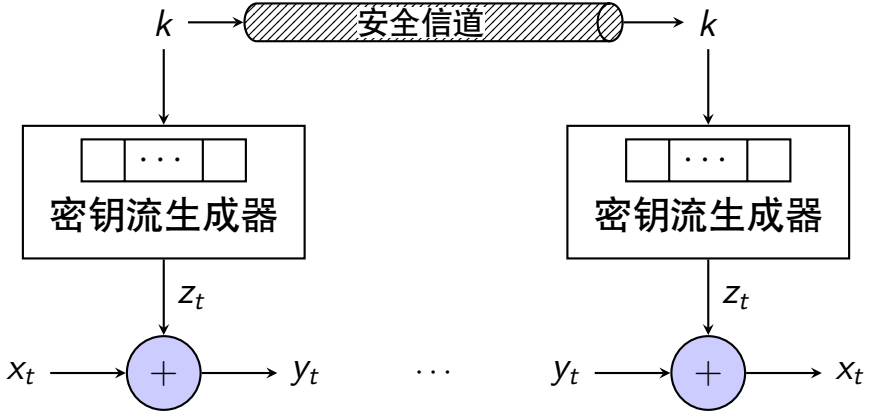
流密码的基本思想

- 利用密钥 k 产生密钥流 $z = z_0 z_1 z_2 \cdots$
- 对明文串 $x = x_0 x_1 x_2 \cdots$ 加密得到密文串 $y = y_0 y_1 y_2 \cdots$，其中
$$y_t = E_{z_t}(x_t)$$
- 密钥流 z 由密钥流发生器 f 产生：
$$z_t = f(k, \sigma_t)$$
其中 σ_t 是内部记忆元件在时刻 t 时的状态。
- 流密码具有记忆性，其内部记忆元件由一组移位寄存器构成。
- 同步流密码：内部状态与明文无关，否则为自同步流密码。
- 目前对同步流密码的研究比较深入。

同步流密码体制模型



加法同步流密码体制模型



加密变换为 $y_t = z_t \oplus x_t$，其原型为一次一密。

目录

有限状态自动机模型（Finite State Automata）

定义 (有限状态自动机)

具有有限离散输入、输出，包含以下部分：

1. 有限状态集 $S = \{s_i : i = 1, 2, \ldots, r\}$；
2. 有限输入字符集 $I = \{\alpha_j : j = 1, 2, \ldots, m\}$ 和有限输出字符集 $O = \{o_k : k = 1, 2, \ldots, n\}$；
3. 输出函数

$$o = \psi(s, \alpha)$$

即在状态 s 输入为 α 时，输出 o。

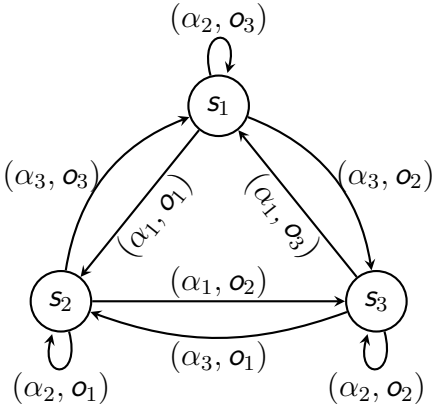
4. 状态转移函数

$$s' = \phi(s, \alpha)$$

即在状态 s 输入为 α 时，状态转移到 s'。

有限状态自动机的有向图表示

- 有限状态自动机可用有向图表示，称为转移图。
- 转移图的顶点对应于有限状态自动机的状态。
- 转移图的边上标有输入输出字符。



有限状态自动机的矩阵表示

设 $S = \{s_1, s_2, s_3\}$, $I = \{\alpha_1, \alpha_2, \alpha_3\}$, $O = \{o_1, o_2, o_3\}$

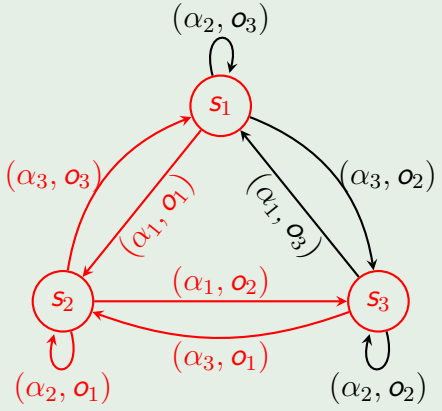
ψ	α_1	α_2	α_3
s_1	o_1	o_3	o_2
s_2	o_2	o_1	o_3
s_3	o_3	o_2	o_1

ϕ	α_1	α_2	α_3
s_1	s_2	s_1	s_3
s_2	s_3	s_2	s_1
s_3	s_1	s_3	s_2

举例：有限状态自动机

例 (有限状态自动机)

- 输入序列：$\alpha_1\alpha_2\alpha_1\alpha_3\alpha_3\alpha_1$
- 状态序列为：$s_1s_2s_2s_3s_2s_1s_2$
- 输出字符序列：$o_1o_1o_2o_1o_3o_1$



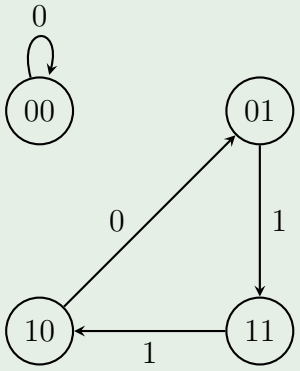
举例：有限状态自动机

例 (周期序列)

设 $S = \{00, 01, 10, 11\}$，$I = \emptyset$，$O = \{0, 1\}$，转移图如图所示。

- 若初始状态为 00
- 则状态序列为：
  $00, 00, 00, 00, \ldots$
- 输出字符序列为：$0000\cdots$

- 若初始状态为 01
- 则状态序列为：
  $01, 11, 10, 01, \ldots$
- 输出字符序列为：$1101\cdots$



两种情况下输出序列都为周期序列，周期分别为 1 和 3。

目录

周期序列的伪随机性

- 流密码的安全性依赖于密钥流的随机性，随机性越好流密码越安全。
- 如果密钥流是周期的，那么必然不能做到完全随机。
- 密码分析者只要获得一个周期内的密钥流后，那么整个密钥流就被暴露。
- 伪随机序列：要求截获比周期短的一段序列时，仍然不会泄露序列在周期内更多的信息，这样的序列为伪随机序列。
- 问题：如何评价一个周期序列伪随机性的好坏程度？

游程

定义 (游程)

设序列 $\{z_i\}$ 是 $\mathrm{GF}(2)$ 上周期为 T 的周期序列，将序列的一个周期

$$(z_1, z_2, \ldots, z_T)$$

依次排列在一个圆周上，使 z_T 与 z_1 相连，称这个圆周上形如

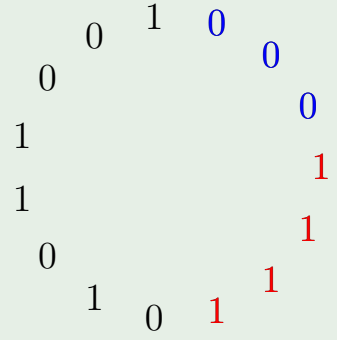
$$0\underbrace{11\cdots11}_{\text{全为 1}}0 \quad \text{或} \quad 1\underbrace{00\cdots00}_{\text{全为 0}}1$$

的一连串相邻的项分别称为序列 $\{z_i\}$ 的一个周期中的一个 1 游程或一个 0 游程。

游程举例

例 (游程)

- 考虑一个周期为 15 的序列: 100010011010111
- 011110 为 1 的 4 游程;
- 10001 为 0 的 3 游程。



周期序列的自相关函数

定义 (自相关函数)

设序列 $\{z_i\}$ 是 $\mathrm{GF}(2)$ 上周期为 T 的周期序列，称

$$R(t) \triangleq \frac{1}{T}\sum_{i=1}^{T}(-1)^{z_i}(-1)^{z_{i+t}}, \quad 0 \le t < T$$

为序列的自相关函数。

- 表示序列 $\{z_i\}$ 与序列 $\{z_{i+t}\}$（平移 t 个单位）在一个周期内对应位相同位数与不同位数的差。
- 当 $t = 0$ 时，$R(t) = 1$；当 $t \neq 0$ 时，称 $R(t)$ 为异自相关函数。

Golomb 伪随机公设

Golomb 伪随机公设

一个好的伪随机周期序列应满足以下三个条件：

1. 在序列的一个周期内，0 与 1 的个数相差至多为 1。
2. 在序列的一个周期内，长为 i 的游程占游程总数的 $1/2^i$，$i = 1, 2, \ldots$，且在等长的游程中 0 的游程个数和 1 的游程个数相等。
3. 异自相关函数是一个常数。

- 公设 1 说明序列中 0 和 1 出现的概率基本相同；
- 公设 2 说明 0 和 1 在序列中每个位置上出现的概率相同；
- 公设 3 说明通过平移序列，计算自相关函数进行比较，不能得到有用信息。

伪随机序列还应满足的条件

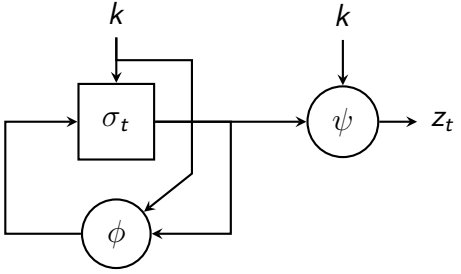
1. 周期 T 要足够大，如大于 10^{50}；
2. 序列 $\{z_i\}$ 产生易于高速生成；
3. 由密文及相应明文的部分信息不能确定整个序列 $\{z_i\}$。

目录

目录

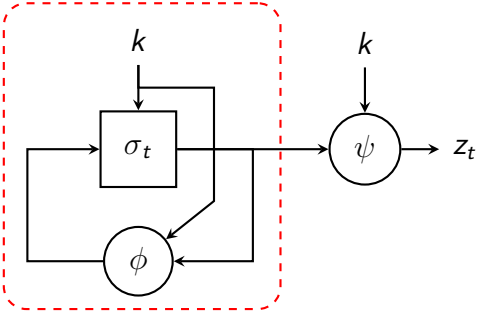
密钥流生成器的一般模型

- 可以看作是一个参数为密钥 k，没有输入字符集的有限状态自动机。
- 包含输出符号集 Z，状态集 Σ，两个函数 ϕ、ψ，以及初始状态 σ_0。
  - 状态转移函数 $\phi\colon \Sigma \mapsto \Sigma$ 将当前状态 σ_t 变为新状态 σ_{t+1}
  - 输出函数 $\psi\colon \Sigma \mapsto Z$ 将当前状态 σ_t 变为输出符号 z_t



密钥流生成器设计

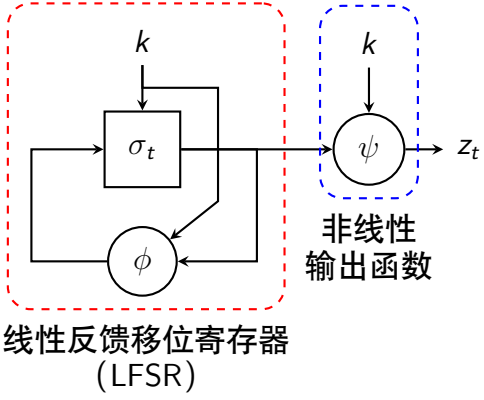
- 目标：设计状态转移函数 ϕ 和输出函数 ψ，使得输出序列 z 满足随机性要求，并且易于分析和实现。
- 方式一：非线性反馈移位寄存器 + 简单的输出函数



非线性反馈移位寄存器

密钥流生成器设计

- 方式二：线性反馈移位寄存器 + 非线性输出函数
- 方式三：多个线性反馈移位寄存器 + 非线性输出函数

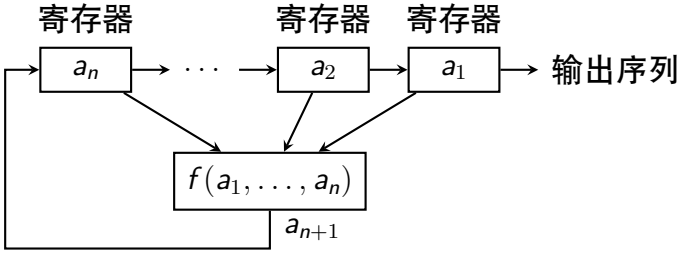


- 方式二和方式三比方式一更易于分析和实现，是目前最为流行和实用的密钥流生成器工作方式。

目录

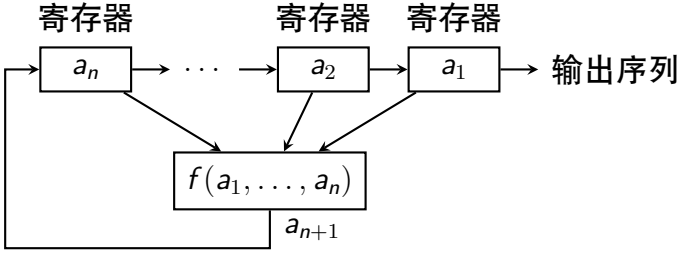
反馈移位寄存器（Feedback Shift Register, FSR）

- n 级反馈移位寄存器由 n 个寄存器与一个反馈函数 f 组成。
- 第 i 个寄存器存储的值为 $a_i \in \{0, 1\}$，反馈函数根据寄存器中的值计算出一个结果，作为 a_{n+1}。
- 每来一个时钟，a_1 作为当前时刻的输出，所有寄存器的值右移一位，a_{n+1} 赋给 a_n。
- 从而产生无限长输出序列 $a_1 a_2 a_3 \cdots$



反馈移位寄存器的状态

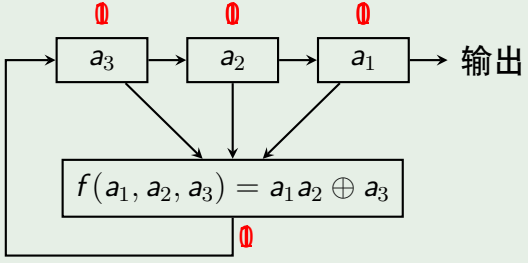
- 在任一时刻，n 个寄存器的值构成反馈移位寄存器的状态。
- 每一个状态对应 $\mathrm{GF}(2)$ 上的一个 n 维向量 $(a_1, \ldots, a_n)$。
- 共有 2^n 种可能状态。



反馈移位寄存器举例

例 (3 级反馈移位寄存器)

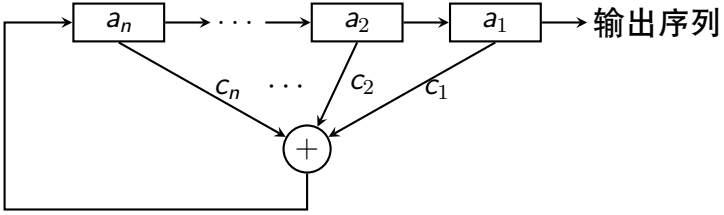
一个 3 级反馈移位寄存器的初始状态为 $(a_1, a_2, a_3) = 101$。



(a_3, a_2, a_1)	输出
1 0 1	1
1 1 0	0
1 1 1	1
0 1 1	1
1 0 1	1
1 1 0	0

- 输出序列为：101110111011...
- 周期为 4

线性反馈移位寄存器 (Linear Feedback Shift Register, LFSR)



- n 级线性反馈移位寄存器
- 反馈函数是 $a_1, a_2, \ldots, a_n$ 的线性函数：

$$f(a_1, \ldots, a_n) = c_1 a_1 \oplus c_2 a_2 \oplus \cdots \oplus c_n a_n$$

其中反馈系数 $c_i \in \{0, 1\}$。

- 输出序列满足递推关系

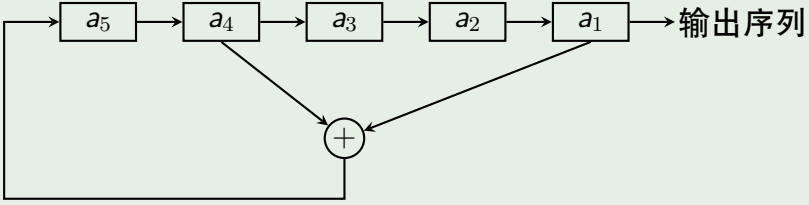
$$a_{n+t} = c_1 a_t \oplus c_2 a_{t+1} \oplus \cdots \oplus c_n a_{n+t-1}, \quad t = 1, 2, \ldots$$

- LFSR 实现简单、速度快、理论成熟。

LFSR 举例

例 (5 级线性反馈移位寄存器)

初始状态 $(a_1, a_2, a_3, a_4, a_5) = 10011$



- 反馈函数递推关系为 $a_{5+t} = a_t \oplus a_{t+3},\ t = 1, 2, \ldots$
- 输出序列为: 10011010010000101011101100011111100110...
- 周期为 31。

LFSR 的性质

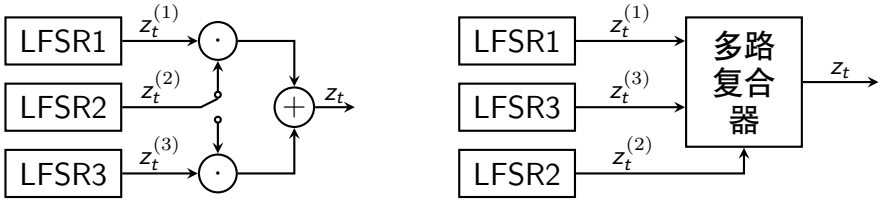
- 假定 $c_1, \ldots, c_n$ 中至少有一个不为 0，否则 $f(a_1, \ldots, a_n) \equiv 0$。
- LFSR 输出序列的性质完全由反馈函数 f 确定。
- 若初始状态为 0，则状态恒为 0。
- n 级 LFSR 的状态数最多为 2^n 个。
- n 级 LFSR 的状态周期小于等于 $2^n - 1$。
- 输出序列的周期等于状态周期，也小于等于 $2^n - 1$。
- LFSR 输出序列的周期由反馈系数 c_i 确定，选择合适的系数可使序列的周期达到最大值 $2^n - 1$。

LFSR 的安全性

- 达到最大周期的 LFSR 产生的序列具有很好的统计特性：输出序列中 0 和 1 的个数近似相等。
- 但是 LFSR 产生的序列不能作为密钥流使用，有安全问题。
- 如果 LFSR 的反馈系数公开（根据 Kerckhoffs 准则），那么 LFSR 输出的前 n 个比特其实是 LFSR 的初始状态。
- 密码分析者根据初始状态和反馈系数可以完全预测 LFSR 未来的输出序列，使密钥流完全可预测。
- 如果不公开反馈系数，密码分析者可以由接下来 n 个输出构建 n 个线性方程，由线性方程组计算出反馈系数。
- 为了抵抗以上攻击，需要使输出序列具有非线性。

目录

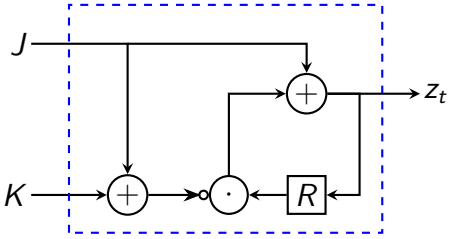
Geffe 序列生成器



- Geffe 序列生成器由 3 个 LFSR 组成，其中 LFSR2 作为控制生成器使用。
- 当 LFSR2 输出 1 时，LFSR2 与 LFSR1 相连接；
- 当 LFSR2 输出 0 时，LFSR2 与 LFSR3 相连接；
- 输出序列可以表示为

$$z_t = z_t^{(1)} z_t^{(2)} \oplus z_t^{(3)} \overline{z_t^{(2)}} = z_t^{(1)} z_t^{(2)} \oplus z_t^{(3)} z_t^{(2)} \oplus z_t^{(3)}$$

JK 触发器

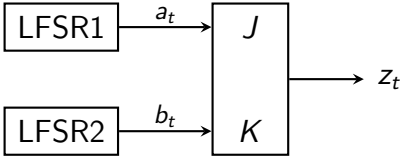


x_J	x_K	z_t
0	0	z_{t-1}
0	1	0
1	0	1
1	1	$\overline{z_{t-1}}$

- JK 触发器的输出 z_t 不仅依赖输入 x_J, x_K 还依赖于上一时刻的输出 z_{t-1}

$$z_t = \overline{(x_J \oplus x_K)} z_{t-1} \oplus x_J$$

利用 JK 触发器的非线性序列生成器



- 输出序列为

$$z_t = \overline{(a_t \oplus b_t)} z_{t-1} \oplus a_t = (a_t \oplus b_t \oplus 1) z_{t-1} \oplus a_t$$

弱点

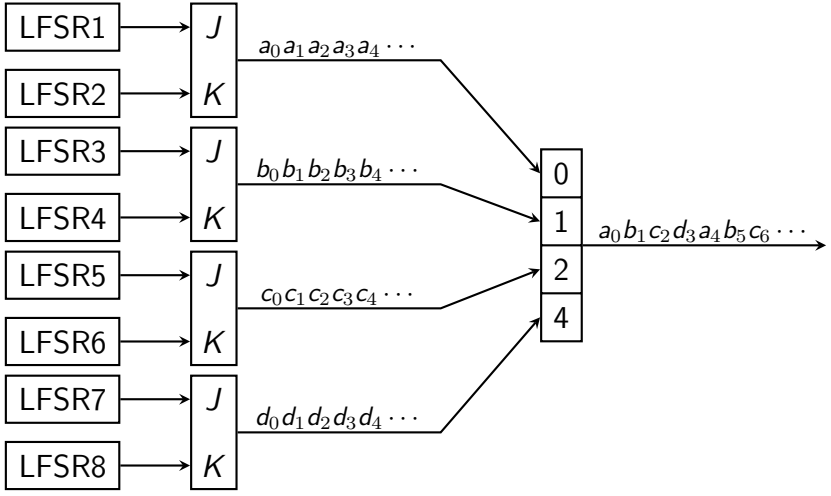
- 由 $z_t = \overline{(a_t \oplus b_t)} z_{t-1} \oplus a_t$ 得

$$z_t = \begin{cases} a_t, & z_{t-1} = 0 \\ \overline{b_t}, & z_{t-1} = 1 \end{cases}$$

- 如果知道 $\{z_t\}$ 中相邻位的值 z_{t-1} 和 z_t，就可以推断出 a_t 和 b_t 中的一个。而一旦知道足够多的这类信息，就可通过密码分析的方法得到序列 a_t 和 b_t。
- 为了克服上述缺点，Pless 提出了由多个 JK 触发器序列驱动的多路复合序列方案，称为 Pless 生成器。

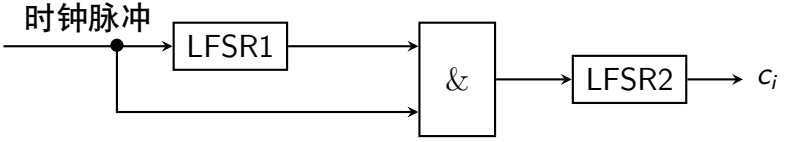
Pless 生成器

- 由 8 个 LFSR、4 个 JK 触发器和 1 个循环计数器构成。
- 由循环计数器选通控制，假定时刻 t 输出第 $t \bmod 4$ 个单元。



钟控序列生成器

- 钟控序列最基本的模型是用一个 LFSR 控制另外一个 LFSR 的移位时钟脉冲。



- 当 LFSR1 输出 1 时，移位时钟脉冲通过与门使 LFSR2 进行一次移位，从而生成下一位。
- 当 LFSR1 输出 0 时，移位时钟脉冲无法通过与门影响 LFSR2。因此 LFSR2 重复输出前一位。

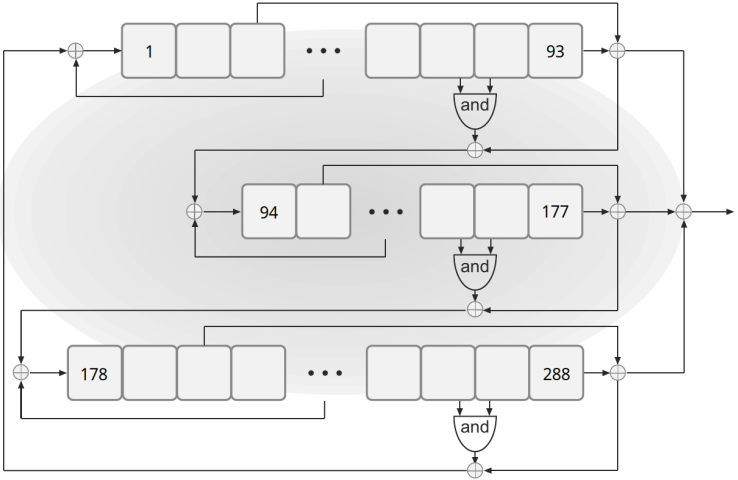
目录

目录

Trivium 流密码介绍

- Trivium 流密码由比利时密码学家于 2008 年提出。
- 是一种轻量级流密码，可以使用较少的门电路实现，需要较少的计算、存储等资源，适宜应用于嵌入式系统。
- 包含三组相互耦合的非线性反馈移位寄存器 A, B 和 C，级数分别为 $93, 84$ 和 111，所以 Trivium 的状态共包含 288 比特。
- 密钥长度为 80 比特，初始向量 IV 为 80 比特。
- 反馈函数：每组寄存器的部分寄存器值经过非线性运算，然后和下一组中某个寄存器值异或，作为下一组寄存器最左边的寄存器值。
- 输出函数：每组寄存器最右边的寄存器值和本组中某个寄存器值分别异或后，再异或，作为输出密钥流。
- 安全性：目前尚无比穷举攻击更好的攻击方法。

Trivium 流密码结构



- **初始化**：用 80 位密钥填充寄存器 A 的最左边 80 个寄存器，用 80 位 IV 填充寄存器 B 的最左边 80 个寄存器，寄存器 C 的最右边 3 个寄存器设为 1，其余寄存器都设为 0。

目录

其他流密码设计

- RC4 流密码
  - 易于软件实现，由 Ron Rivest 与 1987 年提出。
  - 曾普遍应用于 802.11 无线网保密通信中的 WEP 加密方案，但现在已经不安全。
- ChaCha20 流密码
  - 作为 RC4 的替代，目前仍可以安全使用。

小结