现代密码学

祖冲之序列密码算法

信息与软件工程学院

祖冲之算法（ZUC）（GM/T 0001.1-2012）

- ZUC 算法最初是面向4G LTE 空口加密设计的序列密码算法
- 2011年9月被3GPP LTE 采纳为国际加密标准（3GPP TS 33.401）
- 2012年3月，发布为国家密码行业标准GM/T0001-2012
- 2016年10月，发布为国家标准GB/T 33133-2016
- ZUC 算法目前主要用于通信领域
- ZUC算法是一个基于字设计的同步序列密码算法
  - 种子密钥SK和初始向量IV的长度均为128比特
  - 在SK和IV的控制下，每拍输出一个32比特字
- 标准起草人：冯登国、林东岱、冯秀涛、周春芳

祖冲之序列密码算法

- 算法中的符号及含义
- 祖冲之密码的算法结构
- 祖冲之密码的运行
- 基于祖冲之密码的机密性算法128-EEA3

算法中的符号及含义

数制表示

文中整数如果不加特殊说明都为十进制，如果有前缀“0x”则表示十六进制，如果有下标“2”则表示二进制。

例　整数 a 可以有以下不同数制表示形式。

a =1234567890　　十进制表示

=0x499602D2　　十六进制表示

$=1001001100101100000001011010010_2$　　二进制表示

数据位序

文中所有数据的最高位（或字节）在左边，最低位（或字节）在右边。如 a=1001001100101100000001011010010,a 的最高位为其最左边一位1,a 的最低位为其最右边一位0。

算法中的符号及含义

运算符号表示

+	两个整数加
ab	两个整数a和b相乘
=	赋值运算
mod	整数取模
$\oplus$	整数间逐比特异或（模2加）
$\boxplus$	模 2^{32} 加
$a \| b$	串 a和 b级联
a_H	整数 a 的高（最左）16位
a_L	整数 a 的低（最右）16位
$a <<< k$	a 循环左移k位
$a >> 1$	a 右移一位
$(a_1, a_2, \cdots, a_n) \rightarrow (b_1, b_2, \cdots, b_n)$	a_i 到 b_i的并行赋值

算法中的符号及含义

例 $a=0\text{x}1234, b=0\text{x}5678$ ， $c=a\,||\,b=0\text{x}12345678$ 。

例 $a=10010011001011000000001011010010_2$ ，则

$a_H=1001001100101100_2$，$a_L=0000001011010010_2$ 。

例 $a=11001001100101100000001011010010_2$ ，则

$a>>1=11001001100101100000000101101001_2$ 。

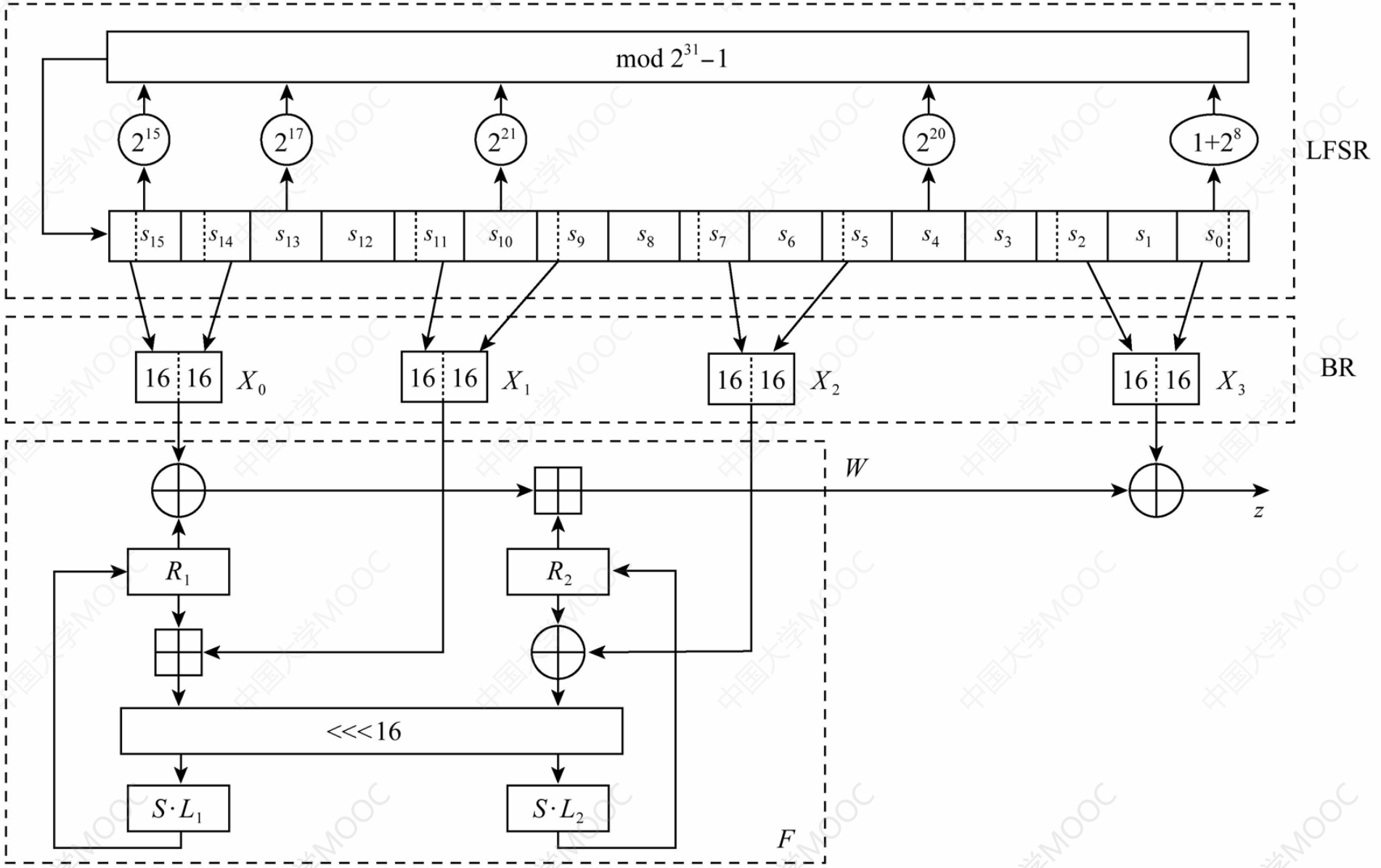
例 设 $a_1,a_2,\cdots,a_{15},b_1,b_2,\cdots,b_{15}$ 都是整数，$(a_1,a_2,\cdots,a_{15})\to(b_1,b_2,\cdots,b_{15})$

意味着 $b_i=a_i(1\le i\le 15)$ 。

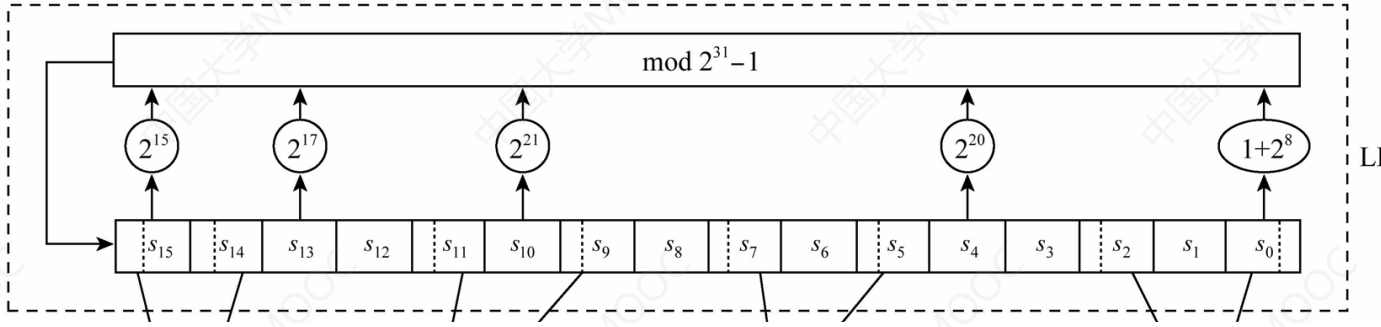
祖冲之序列密码算法

- 算法中的符号及含义
- 祖冲之密码的算法结构
- 祖冲之密码的运行
- 基于祖冲之密码的机密性算法128-EEA3

祖冲之密码的算法结构


mod $2^{31}-1$
2^{15}
2^{17}
2^{21}
2^{20}
$1+2^{8}$
LFSR
s_{15} s_{14} s_{13} s_{12} s_{11} s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0
16 16 X_0
16 16 X_1
16 16 X_2
16 16 X_3
BR
W
z
R_1
R_2
<<<16
$S \cdot L_1$
$S \cdot L_2$
F

祖冲之密码的算法结构



线性反馈移位寄存器

线性反馈移位寄存器（LFSR）由16个31比特寄存器单元 $s_0, s_1, \cdots, s_{15}$ 组成，每个单元在集合

$$\{1,\ 2,\ 3,\ \cdots,\ 2^{31}-1\}$$

中取值。

线性反馈移位寄存器的特征多项式是有限域 $GF(2^{31}-1)$ 上的16次本原多项式

$$p(x)=x^{16}-2^{15}x^{15}-2^{17}x^{13}-2^{21}x^{10}-2^{20}x^{4}-(2^{8}+1)$$

其输出为有限域 $GF(2^{31}-1)$ 上的 m 序列，具有良好的随机性。

$$s_{16+t}=2^{15}s_{15+t}+2^{17}s_{13+t}+2^{21}s_{10+t}+2^{20}s_{4+t}+(2^8+1)s_t(\mathrm{mod}2^{31}-1)$$

祖冲之密码的算法结构

线性反馈移位寄存器的运行模式有两种：初始化模式和工作模式。

（1）初始化模式

在初始化模式中，LFSR接收一个31比特字，是由非线性函数 F 的32比特输出 W 通过舍弃最低位比特得到，即 $u=W>>1$ 。计算过程如下：

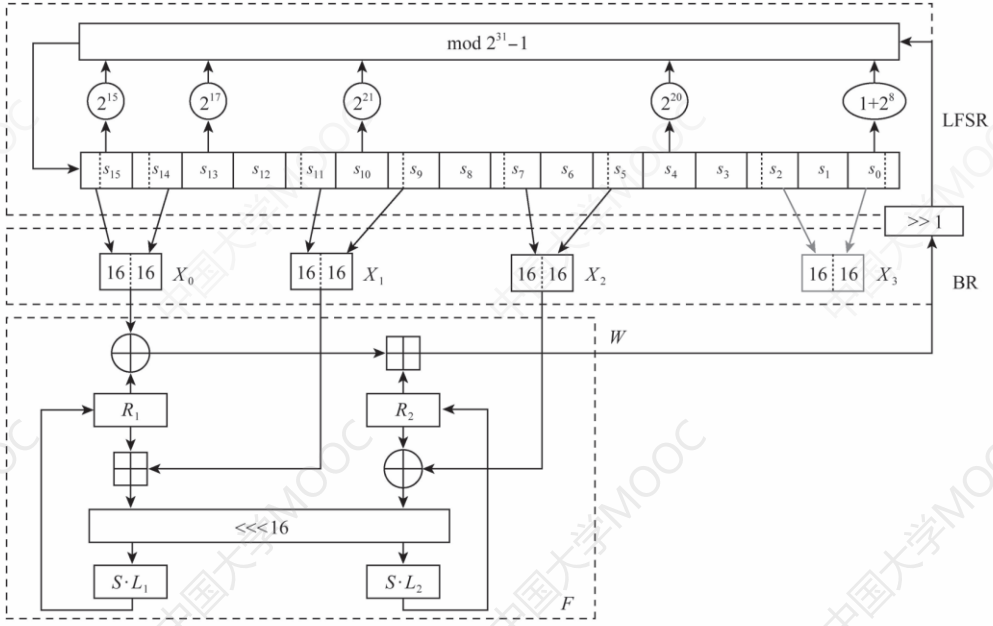
$\mathrm{LFSRWithInitialisationMode}(u)$

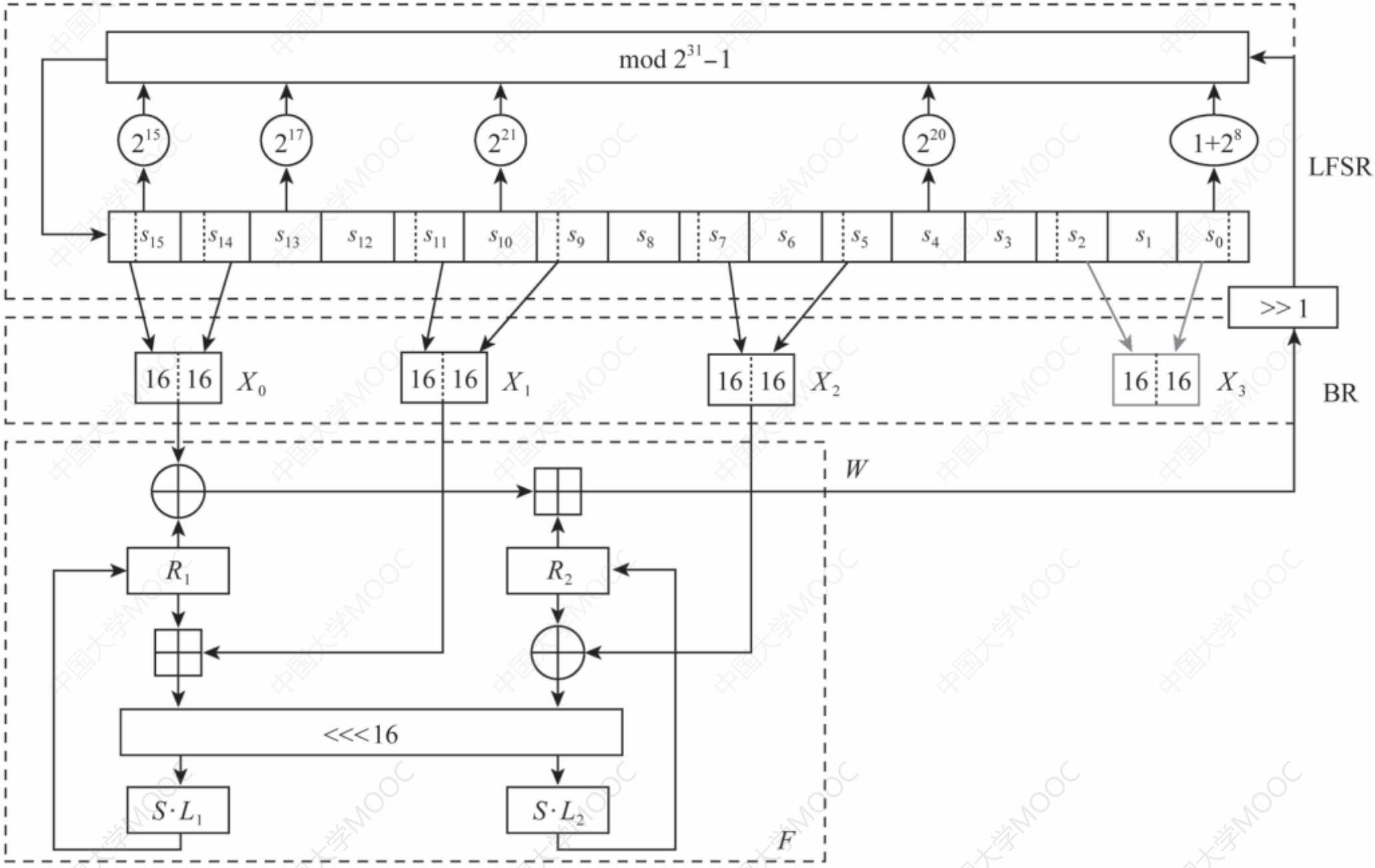
① $v=2^{15}s_{15}+2^{17}s_{13}+2^{21}s_{10}+2^{20}s_4+(1+2^8)s_0 \bmod (2^{31}-1)$

② $s_{16}=(v+u)\bmod(2^{31}-1)$

③ 如果 $s_{16}=0$ ，则置 $s_{16}=2^{31}-1$

④ $(s_1, s_2, \cdots, s_{15}, s_{16})\rightarrow(s_0, s_1, \cdots, s_{14}, s_{15})$




mod $2^{31}-1$
2^{15}
2^{17}
2^{21}
2^{20}
$1+2^8$
LFSR
s_{15} s_{14} s_{13} s_{12} s_{11} s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0
>> 1
16 16 X_0
16 16 X_1
16 16 X_2
16 16 X_3
BR
W
R_1
R_2
<<<16
$S \cdot L_1$
$S \cdot L_2$
F

祖冲之密码的算法结构

（2）工作模式

在工作模式下，LFSR没有输入。其计算过程如下：

LFSRWithWorkMode( )

① $s_{16}=2^{15}s_{15}+2^{17}s_{13}+2^{21}s_{10}+2^{20}s_4+\left(1+2^8\right)s_0 \bmod\left(2^{31}-1\right)$ ；

②如果 $s_{16}=0$，则置 $s_{16}=2^{31}-1$；

③ $\left(s_1, s_2, \cdots, s_{15}, s_{16}\right)\rightarrow\left(s_0, s_1, \cdots, s_{14}, s_{15}\right)$。

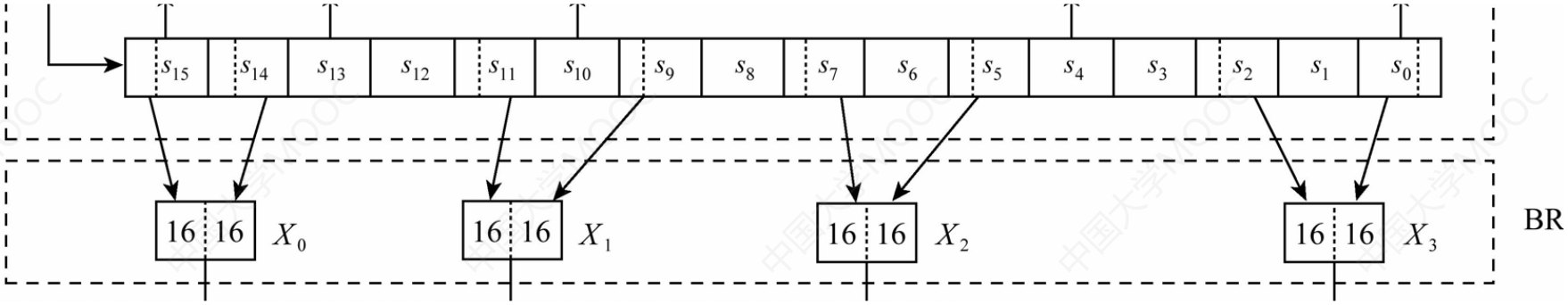
祖冲之密码的算法结构

比特重组

比特重组从LFSR的寄存器单元中抽取128比特组成4个32比特字 X_0, X_1, X_2, X_3，其中前3个字将用于下层的非线性函数 F，第4个字参与密钥流的计算。

BitReconstruction( )

1. $X_0 = s_{15H} \| s_{14L}$
2. $X_1 = s_{11L} \| s_{9H}$
3. $X_2 = s_{7L} \| s_{5H}$
4. $X_3 = s_{2L} \| s_{0H}$



祖冲之密码的算法结构

非线性函数 F

非线性函数 有2个32比特长的存储单元 R_1 和 R_2，其输入为来自上层比特重组的3个32比特字 X_0 、X_1 、X_2，输出为一个32比特字 W。因此，非线性函数 F 是一个把96比特压缩为32比特的一个非线性压缩函数。

具体计算过程如下：

$$F(X_0, X_1, X_2)$$

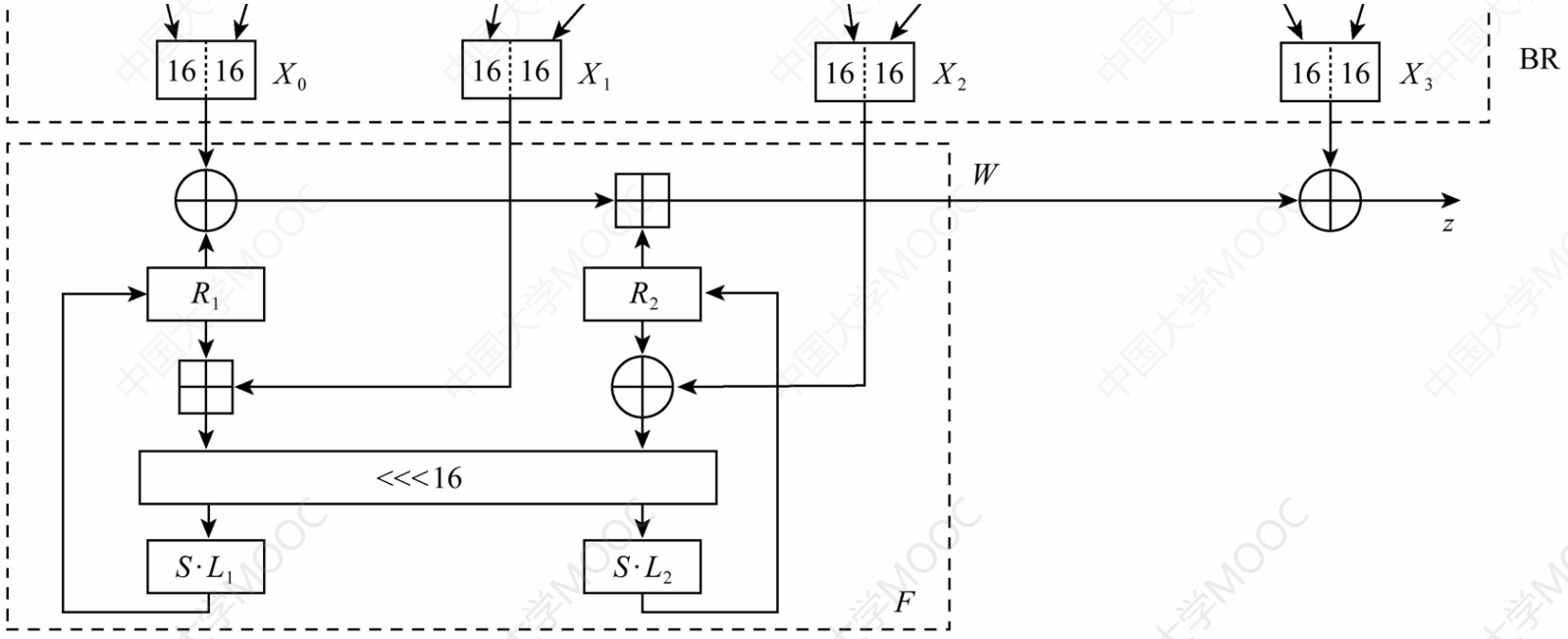
$$W = (X_0 \oplus R_1) \boxed{+} R_2$$

$$W_1 = R_1 \boxed{+} X_1$$

$$W_2 = R_2 \oplus X_2$$

$$R_1 = S(L_1(W_{1L} \| W_{2H}))$$

$$R_2 = S(L_2(W_{2L} \| W_{1H}))$$



祖冲之密码的算法结构

S盒：32×32(即输入长和输出长都为32比特）的S盒由4个并置的8×8 的S盒构成，即

$$S=(S_0, S_1, S_2, S_3)$$

其中$S_2=S_0$,$S_3=S_1$，于是有

$$S=(S_0, S_1, S_0, S_1)$$

例：设S 的输入、输出分别为 X（32比特长）和 Y（32比特长），将X和Y分别表示成4个字节 $X=x_0\|x_1\|x_2\|x_3, Y=y_0\|y_1\|y_2\|y_3$，那么 $y_i=S_i(x_i),(i=0,1,2,3)$.

祖冲之密码的算法结构

S_1

设 x 是 S_0 的8比特长输入，将 x 写成2个16进制数 $x=h\|\ell$，那么其输出是 S_0 盒的第 h 行和第 ℓ 列交叉位置的16进制数。

输入：10100110

即a6

输出：A3

即10100011

表1 S_0盒

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	3E	72	5B	47	CA	E0	00	33	04	D1	54	98	09	B9	6D	CB
1	7B	1B	F9	32	AF	9D	6A	A5	B8	2D	FC	1D	08	53	03	90
2	4D	4E	84	99	E4	CE	D9	91	DD	B6	85	48	8B	29	6E	AC
3	CD	C1	F8	1E	73	43	69	C6	B5	BD	FD	39	63	20	D4	38
4	76	7D	B2	A7	CF	ED	57	C5	F3	2C	BB	14	21	06	55	9B
5	E3	EF	5E	31	4F	7F	5A	A4	0D	82	51	49	5F	BA	58	1C
6	4A	16	D5	17	A8	92	24	1F	8C	FF	D8	AE	2E	01	D3	AD
7	3B	4B	DA	46	EB	C9	DE	9A	8F	87	D7	3A	80	6F	2F	C8
8	B1	B4	37	F7	0A	22	13	28	7C	CC	3C	89	C7	C3	96	56
9	07	BF	7E	F0	0B	2B	97	52	35	41	79	61	A6	4C	10	FE
A	BC	26	95	88	8A	B0	A3	FB	C0	18	94	F2	E1	E5	E9	5D
B	D0	DC	11	66	64	5C	EC	59	42	75	12	F5	74	9C	AA	23
C	0E	86	AB	BE	2A	02	E7	67	E6	44	A2	6C	C2	93	9F	F1
D	F6	FA	36	D2	50	68	9E	62	71	15	3D	D6	40	C4	E2	0F
E	8E	83	77	6B	25	05	3F	0C	30	EA	70	B7	A1	E8	A9	65
F	8D	27	1A	DB	81	B3	A0	F4	45	7A	19	DF	EE	78	34	60

祖冲之密码的算法结构

表2 S_1盒

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	55	C2	63	71	3B	C8	47	86	9F	3C	DA	5B	29	AA	FD	77
1	8C	C5	94	0C	A6	1A	13	00	E3	A8	16	72	40	F9	F8	42
2	44	26	68	96	81	D9	45	3E	10	76	C6	A7	8B	39	43	E1
3	3A	B5	56	2A	C0	6D	B3	05	22	66	BF	DC	0B	FA	62	48
4	DD	20	11	06	36	C9	C1	CF	F6	27	52	BB	69	F5	D4	87
5	7F	84	4C	D2	9C	57	A4	BC	4F	9A	DF	FE	D6	8D	7A	EB
6	2B	53	D8	5C	A1	14	17	FB	23	D5	7D	30	67	73	08	09
7	EE	B7	70	3F	61	B2	19	8E	4E	E5	4B	93	8F	5D	DB	A9
8	AD	F1	AE	2E	CB	0D	FC	F4	2D	46	6E	JD	97	E8	D1	E9
9	4D	37	A5	75	5E	83	9E	AB	82	9D	B9	1C	E0	CD	49	89
A	01	B6	BD	58	24	A2	5F	38	78	99	15	90	50	B8	95	E4
B	D0	91	C7	CE	ED	0F	B4	6F	A0	CC	F0	02	4A	79	C3	DE
C	A3	EF	EA	51	E6	6B	18	EC	1B	2C	80	F7	74	E7	FF	21
D	5A	6A	54	1E	41	31	92	35	C4	33	07	0A	BA	7E	0E	34
E	88	B1	98	7C	F3	3D	60	6C	7B	CA	D3	1F	32	65	04	28
F	64	BE	85	9B	2F	59	8A	D7	B0	25	AC	AF	12	03	E2	F2

祖冲之密码的算法结构

（2）线性变换 L_1 和 L_2：L_1 和 L_2 为32比特线性变换，定义如下：

$$\begin{cases} L_1(X) = X \oplus (X <<< 2) \oplus (X <<< 10) \oplus (X <<< 18) \oplus (X <<< 24) \\ L_2(X) = X \oplus (X <<< 8) \oplus (X <<< 14) \oplus (X <<< 22) \oplus (X <<< 30) \end{cases}$$

其中符号 $a <<< n$ 表示把 a 循环左移 n 位。

非线性函数F输出的W与比特重组（BR）输出的x_3异或，形成输出密钥序列Z。

祖冲之密码的算法结构

密钥载入

密钥载入过程将128比特的初始密钥k和128比特的初始向量IV扩展为16个31比特长的整数，作为LFSR寄存器单元$s_0, s_1, \cdots, s_{15}$的初始状态。

设k和IV分别为

$$k = k_0 \| k_1 \| \cdots \| k_{15} \quad \text{和} \quad \mathrm{IV} = iv_0 \| iv_1 \| \cdots\cdots \| iv_{15}$$

其中：k_i和iv_i均为8比特长字节，$0 \le i \le 15$。

密钥载入步骤

1. 设D为240比特的常量，可按如下方式分成16个15比特的子串：

$$D = d_0 \| d_1 \| \cdots \| d_{15}$$

2. 对$0 \le i \le 15$，取 $s_i = k_i \| d_i \| iv_i$

$d_0 = 100010011010111_2$
$d_1 = 010011010111100_2$
$d_2 = 110001001101011_2$
$d_3 = 001001101011110_2$
$d_4 = 101011110001001_2$
$d_5 = 011010111100010_2$
$d_6 = 111000100110101_2$
$d_7 = 000100110101111_2$
$d_8 = 100110101111000_2$
$d_9 = 010111100010011_2$
$d_{10} = 110101111000100_2$
$d_{11} = 001101011110001_2$
$d_{12} = 101111000100110_2$
$d_{13} = 011110001001101_2$
$d_{14} = 111100010011010_2$
$d_{15} = 100011110101100_2$

祖冲之序列密码算法

- 算法中的符号及含义
- 祖冲之密码的算法结构
- 祖冲之密码的运行
- 基于祖冲之密码的机密性算法128-EEA3

祖冲之密码的运行

算法的运行有两个阶段：初始化阶段和工作阶段

（1）初始化阶段

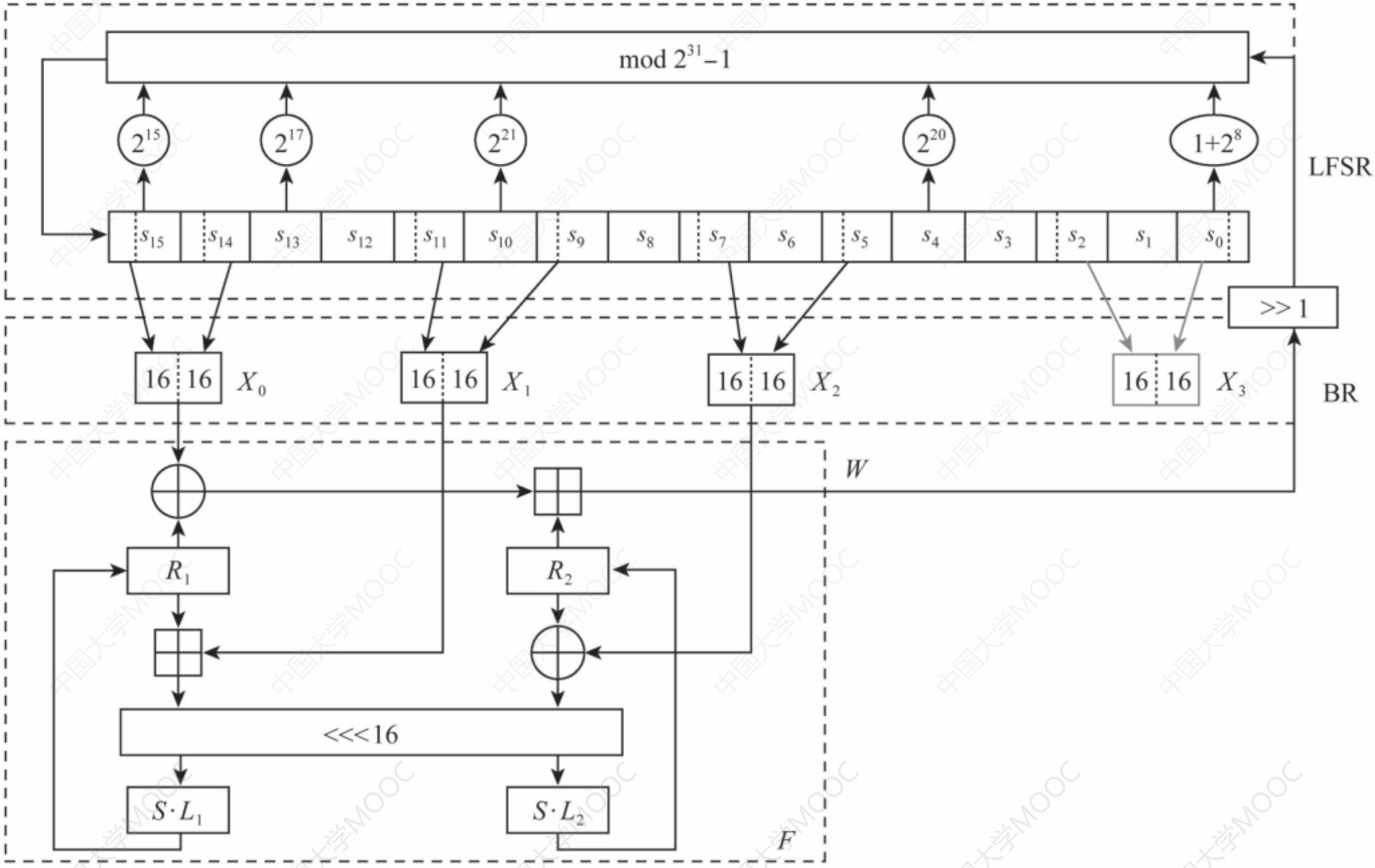
调用密钥装载过程，将128比特的初始密钥 k 和128比特的初始向量 IV 装入到LFSR的寄存器单元变量$s_0, s_1, \cdots, s_{15}$中，作为LFSR的初态，并置非线性函数 F 中的32比特存储单元R_1 和 R_2全为0。

然后重复执行以下过程32次：

BitReconstruction( )

$W = F(X_0, X_1, X_2)$

LFSRWithInitialisationMode (u)


mod $2^{31}-1$
2^{15}
2^{17}
2^{21}
2^{20}
$1+2^8$
LFSR
s_{15} s_{14} s_{13} s_{12} s_{11} s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0
>> 1
16 16 X_0
16 16 X_1
16 16 X_2
16 16 X_3
BR
W
R_1
R_2
<<<16
$S \cdot L_1$
$S \cdot L_2$
F

祖冲之密码的运行

（2）工作阶段

初始化阶段以后，执行工作阶段。

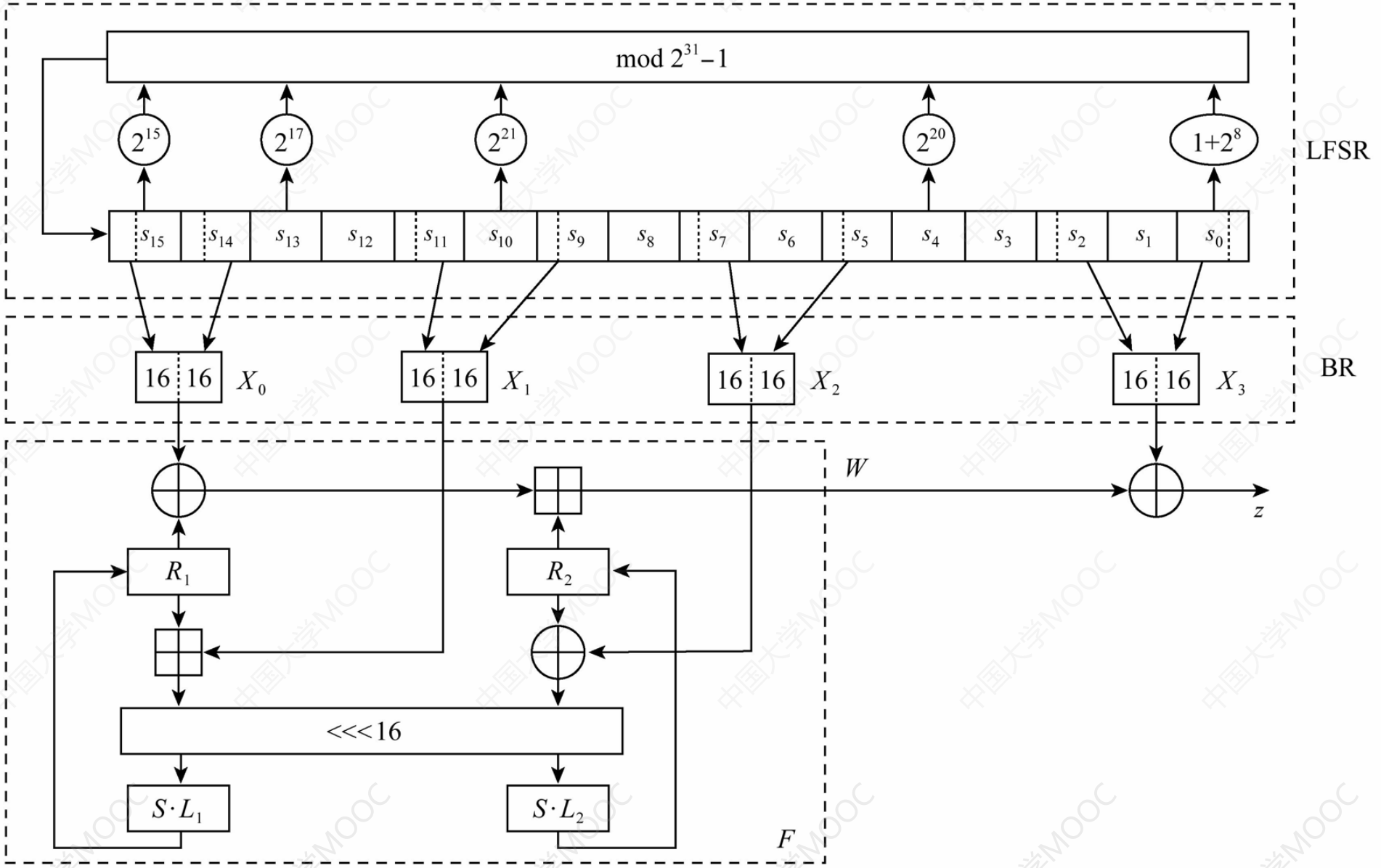
首先执行以下过程一次，并将F 的输出W丢弃：

1. BitReconstruction( )；
2. $F(X_0, X_1, X_2)$；
3. LFSRWithWorkMode ( )。

然后进入密钥输出阶段，其中每进行一次循环，执行以下过程一次，输出一个32比特的密钥字 ：

1. BitReconstruction( )；
2. $Z = F(X_0, X_1, X_2) \oplus X_3$；
3. LFSRWithWorkMode ( ) 。

祖冲之密码的算法结构


mod $2^{31}-1$
2^{15}
2^{17}
2^{21}
2^{20}
$1+2^{8}$
LFSR
s_{15} s_{14} s_{13} s_{12} s_{11} s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0
16 16 X_0
16 16 X_1
16 16 X_2
16 16 X_3
BR
W
z
R_1
R_2
<<<16
$S \cdot L_1$
$S \cdot L_2$
F

祖冲之序列密码算法

- 算法中的符号及含义
- 祖冲之密码的算法结构
- 祖冲之密码的运行
- 基于祖冲之密码的机密性算法128-EEA3

祖冲之密码的机密性算法 128-EEA3

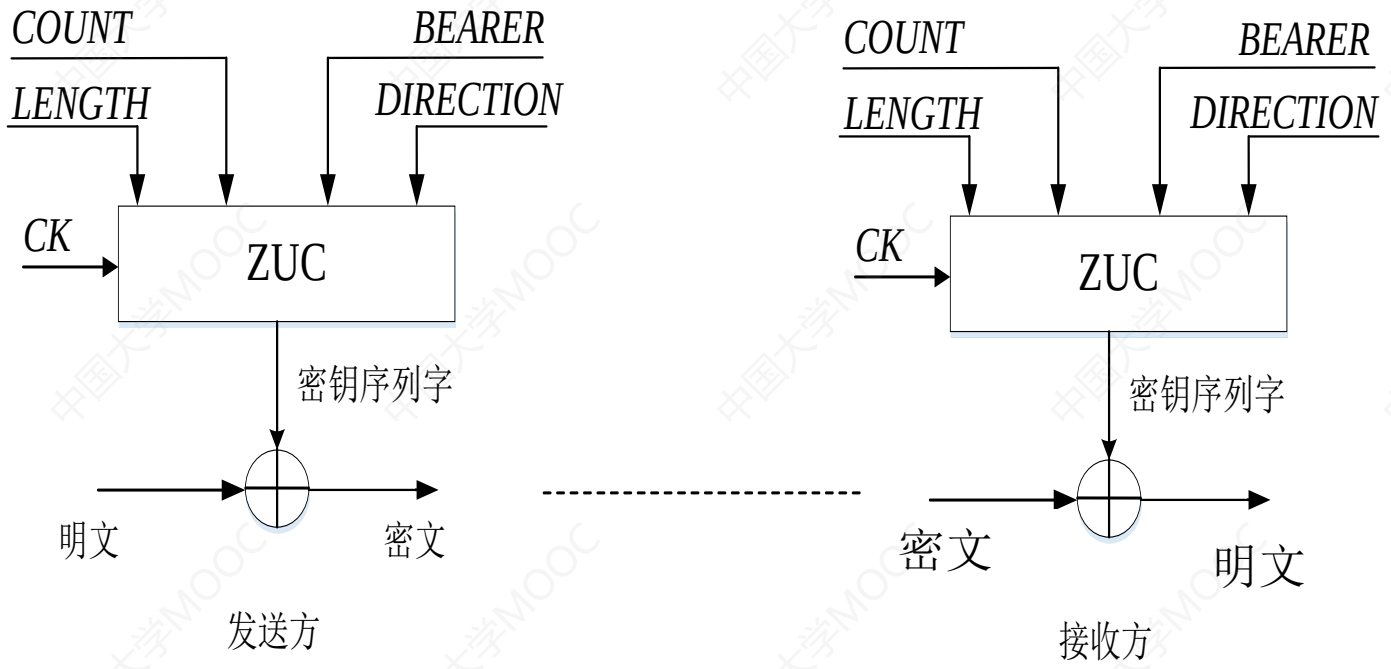
表3 ZUC机密性算法输入参数表

输入参数	比特长度	备注
COUNT	32	计数器
BEARER	5	承载层标识
DIRECTION	1	传输方向标识
CK	128	机密性密钥
LENGTH	32	明文消息的比特长度
M	*LENGTH*	明文消息的比特流

表4 ZUC机密性算法输出参数表

输出参数	比特长度	备注
C	*LENGTH*	输出比特流

祖冲之密码的机密性算法 128-EEA3



基于祖冲之密码的机密性算法128-EEA3

祖冲之密码的机密性算法 128-EEA3

算法工作流程

（1）初始化

初始化是指根据机密性密钥CK以及其他输入参数构造祖冲之算法的初始密钥k和初始向量IV。

把CK（128比特长）和 k（128比特长）分别表示为16个字节：

$$CK = CK[0] \| CK[1] \| CK[2] \| \cdots \| CK[15]$$

$$k = k[0] \| k[1] \| k[2] \| \cdots \| k[15]$$

$$k[i] = CK[i] \quad i = 0,1,2\cdots,15$$

祖冲之密码的机密性算法 128-EEA3

把计数器$COUNT$（32比特长）表示为4个字节：

$$COUNT = COUNT[0] \| COUNT[1] \| COUNT[2] \| COUNT[3]$$

把IV（128比特长）表示为16个字节：

令

$$\begin{cases} IV[0] = COUNT[0], IV[1] = COUNT[1], \\ IV[2] = COUNT[2], IV[3] = COUNT[3], \\ IV[4] = BEARER \| DIRECTION \| 00_2, \\ IV[5] = IV[6] = IV[7] = 00000000_2, \\ IV[8] = IV[0], IV[9] = IV[1], \\ IV[10] = IV[2], IV[11] = IV[3], \\ IV[12] = IV[4], IV[13] = IV[5], \\ IV[14] = IV[6], IV[15] = IV[7]. \end{cases}$$

祖冲之密码的机密性算法 128-EEA3

（2）密钥流的产生

设消息长为 $LENGTH$ 比特，由初始化算法得到的初始密钥 k 和初始向量 IV，调用ZUC密码产生 L 个字（每个32比特长）的密钥，其中 L 为

$$L = \lceil LENGTH / 32 \rceil$$

将生成的密钥流用比特串表示为 $z[0], z[1], \cdots, z[32 \times L - 1]$，其中 $z[0]$ 为ZUC算法生成的第一个密钥字的最高位比特，$z[31]$ 为最低位比特，其他以此类推。

祖冲之密码的机密性算法 128-EEA3

（3）加解密

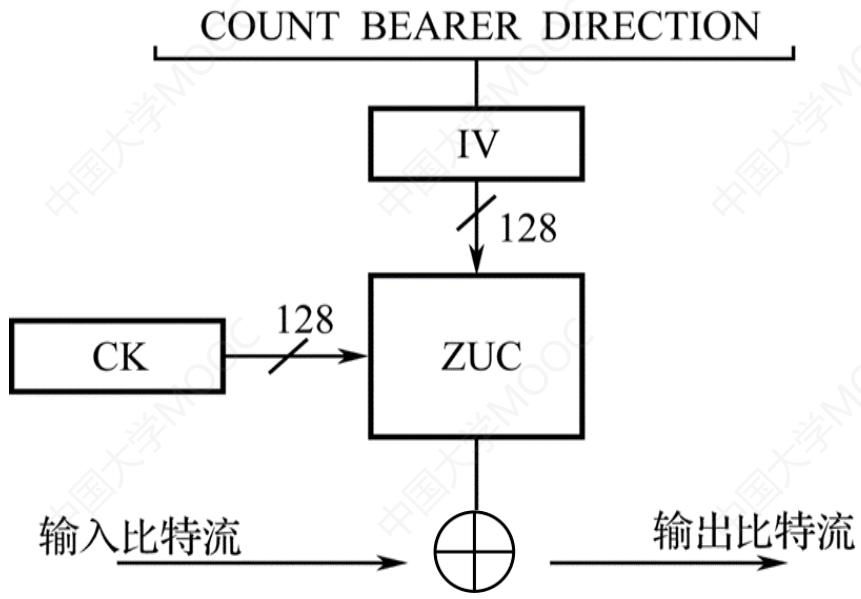
密钥流产生之后，数据的加解密就十分简单了。
设长度为 *LENGTH* 的输入消息的比特流为

$$M = M[0] \| M[1] \| M[2] \| \cdots \| M[LENGTH\text{-}1]$$

则输出的密文比特流为

$$C = C[0] \| C[1] \| C[2] \| \cdots \| C[LENGTH\text{-}1]$$

其中 $C[i] = M[i] \oplus z[i]$ ，$i = 0,1,2,\cdots,LENGTH\text{-}1$



感 谢 聆 听!

xynie@uestc.edu.cn