软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(6):1324−1333 [doi: 10.3724/SP.J.1001.2013.04287]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

SNOW3G 与 ZUC 流密码的猜测决定攻击*

关 杰, 丁 林, 刘树凯

(信息工程大学 电子技术学院,河南 郑州 450004)

通讯作者: 关杰, E-mail: guanjie007@163.com

摘 要: SNOW3G 流密码算法是 3G Partnership Project(3GPP)中实现数据保密性和数据完整性的标准算法 UEA2&UIA2 的核心,ZUC 是 3GPP 中加密算法 128-EEA3 和完整性保护算法 128-EIA3 的核心.至今还没有针对 SNOW3G 进行猜测决定攻击的研究结果出现.对 SNOW3G 进行了猜测决定攻击,其计算复杂度为 2^{320},所需数据量为 9 个 32 比特密钥字.通过对 ZUC 算法设计的分析,将 ZUC 算法中基于 32 比特字的非线性函数转化为基于 16 比特半字的非线性函数,提出了基于 16 比特半字的猜测决定攻击,其计算复杂度为 2^{392},所需数据量为 9 个 32 比特密钥字,该结果优于已有的针对 ZUC 的猜测决定攻击.分析结果表明,尽管 ZUC 算法的内部状态规模小于 SNOW3G,在抵抗猜测决定攻击方面,ZUC 明显优于 SNOW3G.

关键词: 密码分析;猜测决定攻击;SNOW3G;ZUC;3GPP

中图法分类号: TP309 **文献标识码**: A

中文引用格式: 关杰,丁林,刘树凯.SNOW3G 与 ZUC 流密码的猜测决定攻击.软件学报,2013,24(6):1324−1333. http://www.jos.org.cn/1000-9825/4287.htm

英文引用格式: Guan J, Ding L, Liu SK. Guess and determine attack on SNOW3G and ZUC. Ruan Jian Xue Bao/Journal of Software, 2013,24(6):1324−1333 (in Chinese). http://www.jos.org.cn/1000-9825/4287.htm

Guess and Determine Attack on SNOW3G and ZUC

GUAN Jie, DING Lin, LIU Shu-Kai

(Institute of Electronic Technology, Information Engineering University, Zhengzhou 450004, China)

Corresponding author: GUAN Jie, E-mail: guanjie007@163.com

Abstract: SNOW3G stream cipher is the core of the standardized 3G Partnership Project (3GPP) confidentiality and integrity algorithms UEA2 & UIA2 while ZUC stream cipher is the core of the standardized 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. So far, there have been no Guess and Determine attacks applied to SNOW3G. In this paper, a Guess and Determine attack on SNOW3G is proposed with a computational complexity of 2^{320}, requiring 9 keystream words (each word consists of 32 bits). After analyzing the design of ZUC, a half-word-based Guess and Determine attack on ZUC is introduced, based on transforming the word-based nonlinear function of ZUC into a half-word-based nonlinear function. The attack on ZUC has a computational complexity of 2^{392} and requires 9 keystream words, which is better than the previous Guess and Determine attack on ZUC. These results show that ZUC has much better resistance against Guess and Determine attack than SNOW 3G, though the internal state size of ZUC is smaller than SNOW 3G.

Key words: cryptanalysis; Guess and Determine attack; SNOW3G; ZUC; 3GPP

3GPP(3G Partnership Project,第三代合作伙伴计划)是领先的 3G 技术规范机构,由欧洲的 ETSI、日本的 ARIB 和 TTC、韩国的 TTA 以及美国的 T1 在 1998 年底发起成立,旨在研究制定并推广基于演进的 GSM 核心网络的 3G 标准,即 WCDMA,TD-SCDMA,EDGE 等.CWTS(China Wireless Telecommunication Standard Group,

* 基金项目: 国家自然科学基金(61202491, 60272041, 61272488); 全军军事学研究生课题(2010JY0263-149)
收稿时间: 2011-11-19; 定稿时间: 2012-07-16

© 中国科学院软件研究所 http://www.jos.org.cn

中国无线通信标准组)于1999年加入3GPP.3GPP的目标是实现由2G网络到3G网络的平滑过渡,保证未来技术的后向兼容性,支持轻松建网及系统间的漫游和兼容性.

SNOW3G[1]流密码算法是3GPP中实现数据保密性和数据完整性的标准算法——UEA2 & UIA2的核心.SNOW3G是在SNOW2.0[2]的基础上发展而来,是一个面向32比特字实现的流密码算法.迄今为止,针对SNOW3G的分析结果主要有线性区分攻击和Multiset碰撞攻击.2005年,Kaisa和Johan[3]针对SNOW3G利用线性逼近技术构造出了一个简单的区分器,其线性逼近的偏差为$2^{-137.01}$,根据区分攻击理论[4],利用此偏差进行有效的区分攻击所需的数据量和计算复杂度皆约为2^{274}.由于实际应用中一次加密的数据量都是十分有限的,因而如此大的数据量使得该攻击的实际可行性面临质疑.2010年,Alex等人[5]针对初始化轮数为13的简化版SNOW3G构造了一个Multiset区分器,计算复杂度为2^8,而完整版SNOW3G的初始化轮数为33;因此,Multiset碰撞攻击不能对SNOW3G的安全性构成威胁.另外,Blandine和Irene[6]对SNOW3G进行了故障攻击,该攻击方法属于针对密码硬件实现进行攻击的一种.在SNOW3G的设计报告中,设计者声称SNOW3G能够抵抗猜测决定攻击,但并未给出具体的分析结果.

ZUC[7]是由中国科学院数据保护和通信安全研究中心(Date Assurance and Communication Security Research Center,简称DACAS)研制的流密码算法,经中国通信标准化协会、工业和信息化部电信研究院推荐给3GPP申请国际标准.ZUC密码算法的名字源于我国古代数学家祖冲之,它包括加密算法128-EEA3和完整性保护算法128-EIA3.经过历时两年的努力,2011年9月19日~21日,在日本福岗召开的第53次第三代合作伙伴计划(3GPP)系统框架组(SA)会议上,我国自主设计的ZUC流密码算法被批准为新一代无线移动通信系统(long term evolution,简称LTE)国际标准.这是我国商用密码首次走出国门参与国际标准竞争,并取得了重大突破.ZUC成为国际移动通信标准,提高了我国在在移动通信领域的地位和影响力,对我国移动通信产业和商用密码产业发展均具有重大而深远的意义[8].

在ZUC通过了算法标准组ETSI SAGE的内部评估和两个专业团队的外部评估后,ETSI SAGE认为ZUC算法强壮并推荐在LTE标准中使用.随后,DACAS将ZUC(版本v1.4)公开出来,ZUC算法进入到公开评估阶段.在公开评估阶段,孙兵等人[9]和吴宏军等人[10]分别发现了ZUC v1.4初始化过程存在的安全性漏洞.DACAS针对这些分析结果对ZUC算法进行了改进,并于2011年1月发布了最新版的ZUC算法,即ZUC v1.5[11],该版本最终在9月被批准为新一代无线移动通信系统国际标准.在第1届ZUC国际研讨会上,丁林等人[12]基于求解特殊的非线性方程提出了针对ZUC v1.4的猜测决定攻击,计算复杂度为2^{403},需要9个密钥流字.由于ZUC v1.4与ZUC v1.5的区别仅在于密码算法的初始化过程,密钥流生成过程完全一样,因而该结果也同样适用于ZUC v1.5.迄今为止,针对ZUC v1.5仅有的分析结果是周春芳等人[13]构造了一条24轮的选择IV差分传递链.由于完整版SNOW3G的初始化轮数为33;因此,选择IV差分攻击不能对ZUC的安全性构成威胁.

猜测决定攻击(guess and determine attack)是一种针对面向字的流密码的有效攻击方法,其基本思想是,在分析加密过程的内部状态之间的关系以及内部状态和密钥流之间的关系的基础上,猜测一部分内部状态,以此来决定其他的内部状态.猜测决定攻击的计算复杂度主要由猜测量决定,而攻击所需的数据量是指攻击所需的密钥流字的数量.自从Hawkes和Rose提出针对SNOW1.0的猜测决定攻击[14]后,猜测决定攻击逐渐成为一种针对面向字的流密码的有效攻击方法,近期出现了一些很有代表性的研究成果,如对Sober-t32,Polar Bear和SOSEMANUK的猜测决定攻击[15−17].

针对SNOW3G,至今还没有对其进行猜测决定攻击的研究结果出现.本文对SNOW3G进行了猜测决定攻击,其计算复杂度为2^{320},所需数据量为9个32比特密钥字.通过对ZUC算法设计的分析,本文将ZUC算法中基于32比特字的非线性函数转化为基于16比特半字的非线性函数,提出了基于16比特半字的猜测决定攻击,其计算复杂度为2^{392},所需数据量为9个32比特密钥字,该结果优于已有的针对ZUC的猜测决定攻击.分析结果表明,尽管ZUC算法的内部状态规模小于SNOW3G,在抵抗猜测决定攻击方面,ZUC明显优于SNOW3G.

© 中国科学院软件研究所 http://www.jos.org.cn

1 SNOW3G 和 ZUC 算法描述

1.1 SNOW3G流密码

SNOW3G 算法是面向字(32 比特)实现的流密码,密钥规模为 128 比特.密钥流序列的产生包含初始化过程和密钥流生成过程,因为前者对本文攻击没有影响,所以这里仅介绍密钥流生成过程.SNOW3G 算法的密钥流生成器包括一个 $F_{2^{32}}$ 上的16级线性反馈移位寄存器(linear feedback shift register,简称LFSR)和一个有限状态机(finite state machine,简称 FSM),如图 1 所示.其中,⊕表示逐比特异或,乘法是域上的乘法,⊞表示模 2^{32} 加运算,S_1,S_2 均表示 32×32 的 S 盒变换.

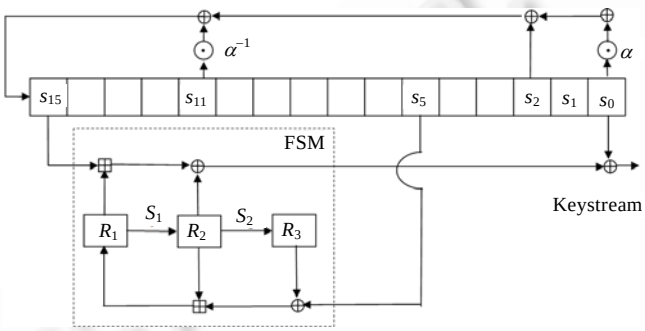

Fig.1 Structure of SNOW3G

图 1 SNOW3G 流密码算法的结构

SNOW3G 中 LFSR 的反馈多项式是域 $GF(2^{32})$上的本原多项式 $\pi(x)=\alpha x^{16}+x^{14}+\alpha^{-1}x^5+1\in F_{2^{32}}[x]$,其中,$GF(2^{32})$是由以下 F_2 上的不可约多项式$\pi(x)=x^{32}+x^{29}+x^{20}+x^{15}+x^{10}+x+1$ 产生的,其中,α是 $x^4+\beta^{23}x^3+\beta^{245}x^2+\beta^{48}x+\beta^{239}\in F_{2^8}[x]$ 的一个根,β是 $x^8+x^7+x^5+x^3+1\in F_2[x]$的一个根.记 $(s_{t+15},s_{t+14},...,s_t)\in F_{2^{32}}$ 为 LFSR 在 t 时刻的内部状态,则 t+1 时刻的内部状态为$(s_{t+16},s_{t+15},\ldots,s_{t+1})$,其中,$s_{t+16}=\alpha^{-1}s_{t+11}\oplus s_{t+2}\oplus\alpha s_t,t\geqslant 0$.

FSM 包括 3 个 32 比特的寄存器 $R1$,$R2$ 和 $R3$.记 FSM 的输出为 f_t,则 $f_t=(s_{t+15}\boxplus R1_t)\oplus R2_t$.密钥流 z_t 是由 f_t 与 s_t 异或形成,记为 $z_t=f_t\oplus s_t$.在 FSM 中,$R1$,$R2$ 和 $R3$ 的刷新变换描述如下:

$$R1_{t+1}=(s_{t+5}\oplus R3_t)\boxplus R2_t,$$
$$R2_{t+1}=S_1(R1_t),R3_{t+1}=S_2(R2_t).$$

1.2 ZUC流密码

ZUC 也是一个面向字实现的流密码算法,其密钥规模为 128 比特.ZUC 算法包含 3 个部分:$GF(2^{31}-1)$环上的 16 级反馈移位寄存器(feedback shift register,简称 FSR)、比特重组和非线性函数,如图 2 所示.

反馈移位寄存器是定义在 $GF(2^{31}-1)$环上的,级数为 16,每个寄存器包含 31 比特,记其在 t 时刻的内部状态为$(s_t,s_{t+1},\ldots,s_{t+15})$,则 t+1 时刻的内部状态为$(s_{t+16},s_{t+15},\ldots,s_{t+1})$,其中,$s_{t+16}$ 的更新方式如下:

{

1. $s_{t+16}=2^{15}s_{t+15}+2^{17}s_{t+13}+2^{21}s_{t+10}+2^{20}s_{t+4}+(1+2^8)s_t \bmod (2^{31}-1)$; (1)
2. 若 $s_{t+16}=0$,则 set $s_{t+16}=2^{31}-1$;

}

比特重组是指从 FSR 中抽出 128 比特组成 4 个中间状态字 X_0,X_1,X_2,X_3,其重组方式如下:

{

1. $X_0=s_{15H}\|s_{14L}$;
2. $X_1=s_{11L}\|s_{9H}$;

© 中国科学院软件研究所 http://www.jos.org.cn

3. $X_2=s_{7L}||s_{5H}$;

4. $X_3=s_{2L}||s_{0H}$;

}

其中,a_H 和 a_L 分别表示 a 的高 16 比特和低 16 比特.

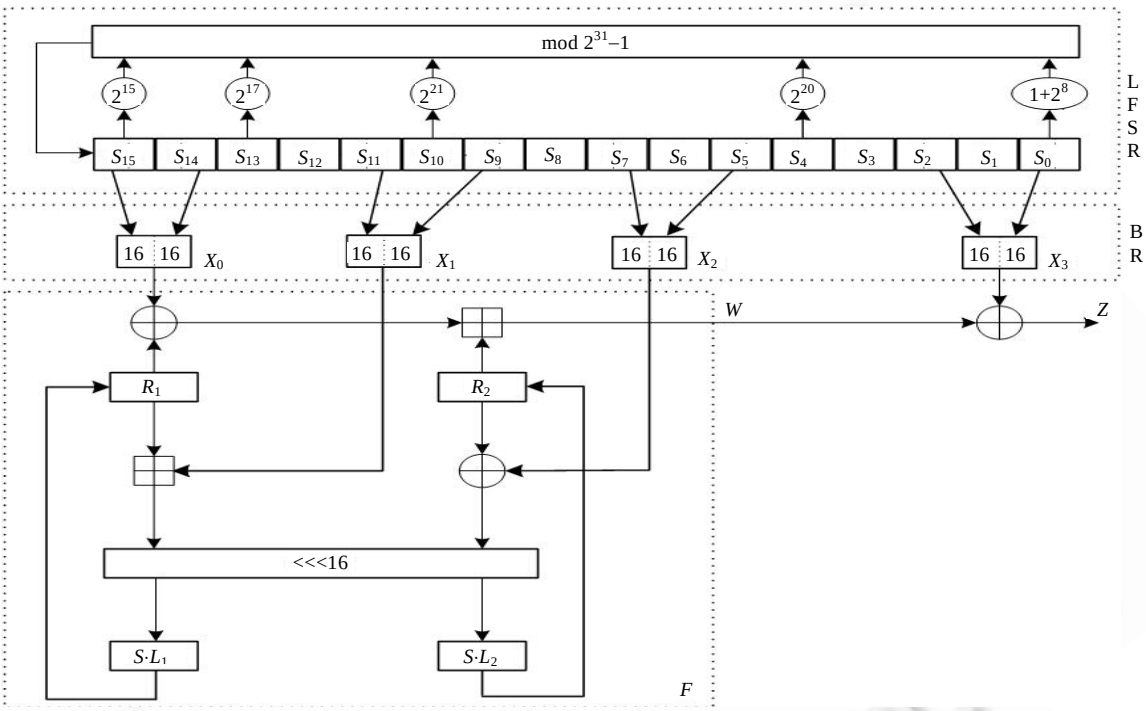

Fig.2 Structure of ZUC

图 2 ZUC 流密码的结构

ZUC 的非线性函数 F 包含两个寄存器单元 $R1$ 和 $R2$,F 的输入为 X_0,X_1 和 X_2,输出为一个 32 比特字 W,F 的更新方式定义如下:

$F(X_0,X_1,X_2)$

{

1. $W=(X_0 \oplus R1_t)\boxplus R2_t$;

2. $W1_t=R1_t\boxplus X_1$;

3. $W2_t=R2_t\oplus X_2$;

4. $R1_{t+1}=S(L_1(W_{1L}||W_{2H}))$;

5. $R2_{t+1}=S(L_2(W_{2L}||W_{1H}))$;

}

其中,S 是一个 32×32 的 S 盒变换,L_1 和 L_2 是两个定义在 32 比特字上的线性变换:

$L1(X)=X\oplus(X<<<2)\oplus(X<<<10)\oplus(X<<<18)\oplus(X<<<24)$;

$L2(X)=X\oplus(X<<<8)\oplus(X<<<14)\oplus(X<<<22)\oplus(X<<<30)$.

其中,<<<表示循环移位.

经过 33 轮的初始化过程后,ZUC 算法进入到密钥流生成过程,每个时刻产生一个密钥流字,记为 Z,其生成方式如下:

© 中国科学院软件研究所 http://www.jos.org.cn

密钥流生成过程

{

1. 比特重组生成 X_0,X_1,X_2,X_3;
2. $Z=W\oplus X_3$;
3. 刷新 FSR.

}

2 SNOW3G 的猜测决定攻击

2.1 攻击过程

在本节中,我们以“猜测一个、决定多个”为原则,利用所猜测的部分内部状态和 SNOW3G 中密钥流生成器的各种变换来决定其他的内部状态,攻击过程可以分为以下 3 个阶段:

- 阶段 1. 攻击者猜测 $R1_2,R2_2,R3_2,s_2,s_3,s_6,s_7,s_8,R1_5,R1_7$ 共 10 个字(320 比特);
- 阶段 2. 利用阶段一中所猜测的内部状态决定 LFSR 的 16 个连续状态$(s_{15},\ldots,s_0)$和 $R1_0,R2_0,R3_0$;
- 阶段 3. 攻击者运用阶段二中得到的$(s_{15},\ldots,s_0)$和 FSM 的状态 $R1_0,R2_0,R3_0$,利用这些初态产生密钥流,将其与观察到的密钥流进行对比以验证攻击结果的正确性.

为描述方便,我们先定义以下关系:“*FSM*”:表示关系式 $f_t=(s_{t+15}\boxplus R1_t)\oplus R2_t$;“*M*”:表示关系式 $R1_{t+1}=(s_{t+5}\oplus R3_t)\boxplus R2_t$;“$S_1$”:表示关系式 $R2_{t+1}=S_1(R1_t)$;“S_2”:表示关系式 $R3_{t+1}=S_2(R2_t)$;“*LFSR*”:表示关系式 $s_{t+16}=\alpha^{-1}s_{t+11}\oplus s_{t+2}\oplus\alpha s_t$.阶段 2 的具体决定过程描述见表 1.

Table 1 Determination process of our guess and determine attack on SNOW3G

表 1 SNOW 3G 的猜测决定攻击的决定过程

步骤	已知内部状态	变换	决定状态	步骤	已知内部状态	变换	决定状态
1	s_2, $R1_2$, $R2_2$, z_2	*FSM*	s_{17}	21	s_8, z_8, s_{23}, $R2_8$	*FSM*	$R1_8$
2	$R2_2$, s_7, $R3_2$	*M*	$R1_3$	22	$R2_6$	S_2	$R3_7$
3	$R1_2$	S_1	$R2_3$	23	$R2_7$, $R3_7$, $R1_8$	*M*	s_{12}
4	$R2_2$	S_2	$R3_3$	24	s_{17}, s_{12}, s_3	*LFSR*	s_1
5	s_3, $R1_3$, $R2_3$, z_3	*FSM*	s_{18}	25	$R2_2$	S_1	$R1_1$
6	$R2_3$, s_8, $R3_3$	*M*	$R1_4$	26	$R3_2$	S_2	$R2_1$
7	$R1_3$	S_1	$R2_4$	27	$R1_2$, $R2_1$, s_6	*M*	$R3_1$
8	$R2_3$	S_2	$R3_4$	28	z_1, $R1_1$, $R2_1$, s_1	*FSM*	s_{16}
9	$R2_4$, $R3_4$, $R1_5$	*M*	s_9	29	z_6, $R1_6$, $R2_6$, s_6	*FSM*	s_{21}
10	$R1_4$	S_1	$R2_5$	30	s_{21}, s_{16}, s_7	*LFSR*	s_5
11	$R2_4$	S_2	$R3_5$	31	z_5, $R1_5$, $R2_5$, s_5	*FSM*	s_{20}
12	$R1_5$	S_1	$R2_6$	32	s_{16}, s_{11}, s_2	*LFSR*	S_0
13	$R2_5$	S_2	$R3_6$	33	$R2_1$	S_1	$R1_0$
14	$R2_6$, $R3_6$, $R1_7$	*M*	s_{11}	34	$R3_1$	S_2	$R2_0$
15	s_{17}, s_8, s_6	*LFSR*	s_{22}	35	z_0, $R1_0$, $R2_0$, s_0	*FSM*	s_{15}
16	s_7, $R1_7$, z_7, s_{22}	*FSM*	$R2_7$	36	s_{20}, s_{15}, s_6	*LFSR*	s_4
17	$R2_7$	S_1	$R1_6$	37	z_4, $R1_4$, $R2_4$, s_4	*FSM*	s_{19}
18	$R2_5$, $R3_5$, $R1_6$	*M*	s_{10}	38	s_{18}, s_4, s_2	*LFSR*	s_{13}
19	$R1_7$	S_1	$R2_8$	39	s_{19}, s_5, s_3	*LFSR*	s_{14}
20	s_{18}, s_9, s_7	*LFSR*	s_{23}				

至此,我们就得到了 LFSR 的 16 个连续状态$(s_{15},\ldots,s_0)$和 $R1_0,R2_0,R3_0$,再结合阶段 3,便可以验证每种猜测的正确与否.

2.2 复杂度分析

在对 SNOW3G 进行猜测决定攻击时,攻击者需要猜测 $R1_2,R2_2,R3_2,s_2,s_3,s_6,s_7,s_8,R1_5,R1_7$ 共 10 个字,因而其计算复杂度为 2^{320},所需数据量为 $z_0,z_1,\ldots,z_8$ 共 9 个 32 比特密钥字.

© 中国科学院软件研究所 http://www.jos.org.cn

3 ZUC的猜测决定攻击

3.1 攻击思想

与 SNOW3G 相比,ZUC 的设计有两个特色:一是选用了 $GF(2^{31}-1)$环上的反馈移位寄存器作为驱动部件;二是比特重组的引入和非线性函数的设计.根据对 ZUC 算法设计特点的分析我们发现,比特重组作为反馈移位寄存器和非线性函数之间的中间环节,对猜测决定攻击的结果有显著影响.因此,为充分利用非线性函数中内部状态之间的关系,减少猜测量和简化决定过程,本文将 ZUC 算法中基于 32 比特字的非线性函数转化为基于 16 比特半字的非线性函数,提出了基于 16 比特半字的猜测决定攻击.具体的转化过程描述如下:

- 密钥流生成变换

$$Z_t=[(s_{t+15,H}\|s_{t+14,L})\oplus R1_t\boxplus R2_t]\oplus(s_{t+2,L}\|s_{t,H}) \tag{2}$$

可以转化为

$$Z_{t,L}=[s_{t+14,L}\oplus R1_{t,L}\boxplus R2_{t,L}]\oplus s_{t,H} \tag{3}$$

$$Z_{t,H}=[s_{t+15,H}\oplus R1_{t,H}\boxplus R2_{t,H}\boxplus c_t^1]\oplus s_{t+2,L} \tag{4}$$

其中, c_t^1 表示 1 比特进位,满足如下关系:

$$c_t^1=\begin{cases}1, & 若(s_{t+14,L}\oplus R1_{t,L})+R2_{t,L}\geqslant 2^{16}\\ 0, & 若(s_{t+14,L}\oplus R1_{t,L})+R2_{t,L}<2^{16}\end{cases}$$

- 状态更新变换

$$W1_t=R1_t\boxplus(s_{t+11,L}\|s_{t+9,H}) \tag{5}$$

可以转化为

$$W1_{t,L}=R1_{t,L}\boxplus s_{t+9,H} \tag{6}$$

$$W1_{t,H}=R1_{t,H}\boxplus s_{t+11,L}\boxplus c_t^2 \tag{7}$$

其中, c_t^2 表示 1 比特进位,满足如下关系:

$$c_t^2=\begin{cases}1, & 若R1_{t,L}+s_{t+9,H}\geqslant 2^{16}\\ 0, & 若R1_{t,L}+s_{t+9,H}<2^{16}\end{cases}$$

- 状态更新变换

$$W2_t=R2_t\oplus(s_{t+7,L}\|s_{t+5,H}) \tag{8}$$

可以直接转化为

$$W2_{t,L}=R2_{t,L}\oplus s_{t+5,H} \tag{9}$$

$$W2_{t,H}=R2_{t,H}\oplus s_{t+7,L} \tag{10}$$

3.2 攻击思想

在完成转化过程后,我们需要选择合适的猜测量,并利用所猜测的内部状态决定其他的内部状态,恢复出 LFSR 的 16 个连续状态$(s_{15},\ldots,s_0)$和 $R1_0,R2_0,R3_0$,进而利用这些内部状态产生密钥流,将其与观察到的密钥流进行对比以验证攻击结果的正确性.

为描述方便,我们先标记以下关系:

$$R1_{t+1}=S(L_1(W1_{t,L}\|W2_{t,H})) \tag{11}$$

$$R2_{t+1}=S(L_2(W2_{t,L}\|W1_{t,H})) \tag{12}$$

针对 ZUC 的猜测决定攻击的过程描述如下:

首先,猜测内部状态 $s_5,s_6,s_7,s_9,s_{10},s_{13,L},s_{15},s_{16},s_{18},s_{19},s_{20},R1_5,c_4^1,c_4^2,c_5^2$ (共 361 比特),决定过程见表 2.

其次,验证步骤 8 中得到的 $s_{11,L}$ 的最高比特与步骤 19 中得到的 $s_{11,H}$ 的最低比特是否相等:对于正确的猜测,该验证式一定成立,而对于错误的猜测,该验证式成立的概率为 0.5.因此,通过该验证式可将猜测量降低一半,即

© 中国科学院软件研究所 http://www.jos.org.cn

由 2^{361} 降为 2^{360}.

随后,猜测内部状态 $s_{13,H^*}, s_{12,H}, c_3^1$ (共 32 比特),其中, s_{13,H^*} 表示 $s_{13,H}$ 的除最低比特之外的高 15 比特.剩余的决定过程见表 2.

Table 2 Determination process of our Guess and Determine attack on ZUC

表 2 ZUC 的猜测决定攻击的决定过程

步骤	已知内部状态	变换	决定状态
1	$s_5, s_9, s_{15}, s_{18}, s_{20}$	(1)	s_{21}
2	$s_6, s_{10}, s_{16}, s_{19}, s_{21}$	(1)	s_{22}
3	$Z_5, (s_{20,H}\|\|s_{19,L}), R1_5, (s_{7,L}\|\|s_{5,H})$	(2)	$R2_5$
4	$R1_5, R2_5$	(11,12)	$W1_4, W2_4$
5	$W2_{4,L}, s_{9,H}$	(9)	$R2_{4,L}$
6	$W1_{4,H}, s_{15,L}, c_4^2$	(7)	$R1_{4,H}$
7	$Z_{4,H}, s_{19,H}, R1_{4,H}, c_4^1, s_{6,L}$	(4)	$R2_{4,H}$
8	$W2_{4,H}, R2_{4,H}$	(9)	$s_{11,L}$
10	$R1_{5,H}, s_{16,L}, c_5^2$	(7)	$W1_{5,H}$
11	$R2_{5,L}, s_{10,H}$	(9)	$W2_{5,L}$
12	$W2_{5,L}, W1_{5,H}$	(12)	$R2_6$
13	$Z_{6,L}, s_{20,L}, R2_{6,L}, s_{6,H}$	(3)	$R1_{6,L}, c_6^1$
14	$R1_{6,L}, s_{15,H}$	(6)	$W1_{6,L}, c_6^2$
15	$R2_{6,H}, s_{13,L}$	(10)	$W2_{6,H}$
16	$W1_{6,L}, W2_{6,H}$	(11)	$R1_7$
17	$Z_7, (s_{22,H}\|\|s_{21,L}), R1_t, (s_{9,L}\|\|s_{7,H})$	(2)	$R2_7$
18	$R2_7$	(12)	$W1_{6,H}, W2_{6,L}$
19	$W2_{6,L}, R2_{6,L}$	(9)	$s_{11,H}$
20	$W1_{4,L}, s_{13,H}$	(6)	$R1_{4,L}$
21	$Z_{4,L}, s_{15,L}, R1_{4,L}, R2_{4,L}$	(3)	$s_{4,H}$
22	$s_7, s_{13}, s_{16}, s_{18}, s_{19}$	(1)	s_3
23	$R1_4, R2_4$	(11,12)	$W1_3, W2_3$
24	$W2_{3,H}, s_{10,L}$	(10)	$R2_{3,H}$
25	$W1_{3,L}, s_{12,H}$	(6)	$R1_{3,L}, c_3^2$
26	$Z_{3,H}, s_{18,H}, c_3^1, s_{5,L}, R2_{3,H}$	(4)	$R1_{3,H}$
27	$W1_{3,H}, R1_{3,H}, c_3^2$	(7)	$s_{14,L}$
28	$R1_7, R2_7, (s_{18,L}\|\|s_{16,H}), (s_{14,L}\|\|s_{12,H})$	(5,8,11,12)	$R1_8, R2_8$
29	$Z_{8,L}, s_{22,L}, R1_{8,L}, R2_{8,L}$	(3)	$s_{8,H}, c_8^1$
30	$Z_{8,H}, R1_{8,H}, R2_{8,H}, c_8^1, s_{10,L}$	(4)	$s_{23,H}$
31	$W2_{3,L}, s_{8,H}$	(9)	$R2_{3,L}$
32	$Z_{3,L}, R1_{3,L}, R2_{3,L}, s_{3,H}$	(3)	$s_{17,L}$
33	$W1_{6,H}, s_{17,L}, c_6^2$	(7)	$R1_{6,H}$
34	$Z_{6,H}, s_{21,H}, R1_{6,H}, R2_{6,H}, c_6^1$	(4)	$s_{8,L}$
35	$R1_6, R2_6$	(11,12)	$W1_{5,L}, W2_{5,H}$
36	$W1_{5,L}, R1_{5,L}$	(6)	$s_{14,H}$
37	$W2_{5,H}, R2_{5,H}$	(10)	$s_{12,L}$
38	$R1_3, R2_3, (s_{14,L}\|\|s_{12,H}), (s_{10,L}\|\|s_{8,H})$	(5,8,11,12)	$R1_2, R2_2$
39	$Z_{2,L}, s_{16,L}, R1_{2,L}, R2_{2,L}$	(3)	$s_{2,H}$
40	$R1_2, R2_2, (s_{13,L}\|\|s_{11,H}), (s_{9,L}\|\|s_{7,H})$	(5,8,11,12)	$R1_1, R2_1$
41	$R1_1, R2_1, (s_{12,L}\|\|s_{10,H}), (s_{8,L}\|\|s_{6,H})$	(5,8,11,12)	$R1_0, R2_0$
42	$Z_0, (s_{15,H}\|\|s_{14,L}), R1_0, R2_0$	(2)	$S_{2,L}, s_{0,H}$
43	$s_2, s_6, s_{12}, s_{15}, s_{18}$	(1)	s_{17}
44	$s_8, s_{14}, s_{17}, s_{19}, s_{20}$	(1)	s_4
45	$s_5, s_{11}, s_{14}, s_{16}, s_{17}$	(1)	s_1
46	$s_4, s_{10}, s_{13}, s_{15}, s_{16}$	(1)	s_0

至此,我们就得到了 LFSR 的 16 个连续状态$(s_{15},\ldots,s_0)$和 $R1_0$,$R2_0$.

3.3 攻击分析

下面对本攻击的复杂度进行分析.

© 中国科学院软件研究所 http://www.jos.org.cn

在对 ZUC 进行猜测决定攻击时,攻击者需要先猜测 $s_5,s_6,s_7,s_9,s_{10},s_{13,L},s_{15},s_{16},s_{18},s_{19},s_{20},R1_5,c_4^1,c_4^2,c_5^2$(共 361 比特),执行表 2 中步骤 1~步骤 19,此时的猜测量为 2^{361}.通过验证步骤 8 中得到的 $s_{11,L}$ 的最高比特与步骤 19 中得到的 $s_{11,H}$ 的最低比特是否相等,将猜测量降低一半,即由 2^{361} 降为 2^{360}.随后再猜测 $s_{13,H^*},s_{12,H},c_3^1$(共 32 比特),此时的猜测量上升为 $2^{360}\cdot 2^{32}=2^{392}$,执行表 3 中步骤 20~步骤 46,完成整个决定过程.因此,该攻击的计算复杂度为 $2^{360}+2^{392}\approx 2^{392}$,攻击所需的数据量为 $z_0,z_1,\ldots,z_8$ 共 9 个 32 比特密钥字.

4 分析比较

4.1 与已有攻击结果的比较分析

SNOW3G 与 ZUC 自公布以来,因其重要性和广泛的应用前景,对它们的安全性分析一直是个热点问题.迄今为止,针对 SNOW3G 的分析结果主要有线性区分攻击和 Multiset 碰撞攻击.遗憾的是,这两个分析结果都是以构造区分器为目的,不能恢复内部状态或者密钥.下面就本文的分析结果与已有的分析结果进行比较,见表 3.

Table 3 Comparisons with previous attacks on SNOW3G

表 3 与已有的针对 SNOW3G 的分析结果的比较

攻击	算法	攻击类型	计算复杂度	数据量
线性区分攻击[3]	SNOW3G(完整版)	区分攻击	2^{274}	2^{274}
Multiset 碰撞攻击[5]	13 轮的 SNOW3G(简化版)	区分攻击	2^8	2^8
猜测决定攻击(本文)	SNOW3G(完整版)	状态恢复攻击	2^{320}	9

从攻击的算法来看,本文能够攻击完整版的 SNOW3G 算法,而 Multiset 碰撞攻击只能攻击初始化轮数为 13 的简化版 SNOW3G,因而本文的猜测决定攻击要优于 Multiset 碰撞攻击.从攻击类型上讲,本文针对 SNOW3G 的猜测决定攻击以恢复内部状态为目的,这与线性区分攻击和 Multiset 碰撞攻击有本质的区别.就分析结果而言,虽然猜测决定的计算复杂度较高,但所需数据量非常小.鉴于实际应用中一次加密的数据量都是十分有限的,因此从实际攻击时所能获得的数据量来看,猜测决定攻击比线性区分攻击更有威胁.综合以上分析,与线性区分攻击和 Multiset 碰撞攻击相比,本文的猜测决定攻击对 SNOW3G 的安全性具有更大的威胁.

目前,针对 ZUC v1.5 的分析结果主要是猜测决定攻击和选择 IV 差分攻击.下面就本文的分析结果与已有的分析结果进行比较,见表 4.由于选择 IV 差分攻击仅是构造了一条 24 轮的选择 IV 差分传递链,并未形成具体的攻击结果,因而在此不作比较.

Table 4 Comparisons with previous attacks on ZUC

表 4 与已有的针对 ZUC 的分析结果的比较

攻击	算法	攻击类型	计算复杂度	数据量
猜测决定攻击[12]	ZUC	状态恢复攻击	2^{403}	9
猜测决定攻击(本文)	ZUC	状态恢复攻击	2^{392}	9

从表 4 可以看出,本文的猜测决定攻击结果明显优于已有的猜测决定攻击.

4.2 SNOW3G与ZUC的比较分析

从算法的结构上讲,SNOW3G 与 ZUC 是比较相似的,均是采用了反馈移位寄存器作为驱动部分,使用有记忆变换构造非线性函数.下面就两个流密码算法进行比较,见表 5.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 5 Comparisons with SNOW3G and ZUC

表 5 SNOW3G 与 ZUC 的比较

参数		SNOW 3G	ZUC
密钥规模		128 比特	128 比特
内部状态规模		576 比特	560 比特
反馈移位寄存器的抽头个数		3	5
非线性函数的输入规模		32 比特	96 比特
猜测决定攻击	计算复杂度	2^{320}	2^{392}
	数据量	9 个密钥流字	9 个密钥流字

从表 5 可以看出,尽管 ZUC 的内部状态规模小于 SNOW3G,但在抵抗猜测决定攻击方面,ZUC 明显优于 SNOW3G,其原因在于如下两方面:

- 与 SNOW3G 相比,ZUC 中非线性函数的输入规模更大,且非线性函数的混乱和扩散效果更好.非线性函数输入规模的增大,使得攻击者需要猜测更多的内部状态才能决定出下一时刻非线性函数的内部状态,这增加了攻击成功所需的猜测量,从而增大了攻击的计算复杂度;在 SNOW3G 中,$R1_t$ 和 $R2_t$ 可以通过 $R2_{t+1}=S_1(R1_t)$ 和 $R3_{t+1}=S_2(R2_t)$ 分别决定出 $R2_{t+1}$ 和 $R3_{t+1}$.即当前时刻非线性函数的若干内部状态可以直接决定下一时刻非线性函数的若干内部状态;而在 ZUC 中,下一时刻非线性函数的任何内部状态都不能仅由当前时刻非线性函数的若干内部状态直接决定,采用的刷新变换公式(11)和公式(12)使得 ZUC 的非线性函数达到了更好的混乱和扩散效果;
- 与 SNOW3G 相比,ZUC 中反馈移位寄存器的抽头个数更多.反馈移位寄存器抽头个数的增多,意味着攻击者需要猜测或者决定更多的内部状态,才能使用移位寄存器的反馈多项式决定未知的内部状态,这增加了攻击成功所需的猜测量,从而增大了攻击的计算复杂度.

5 结束语

SNOW3G 和 ZUC 作为 3GPP 中所使用的面向字的标准流密码算法,全面考察其抵抗各种攻击的能力是非常必要的.猜测决定攻击作为一种针对面向字的流密码的有效攻击方法,在对 SNOW3G 和 ZUC 进行安全性分析时,应对这种攻击方法进行充分的研究,以评估 SNOW3G 和 ZUC 抵抗该攻击的能力.

本文对 SNOW3G 进行了猜测决定攻击,其计算复杂度为 2^{320},所需数据量为 9 个 32 比特密钥字.通过对 ZUC 算法设计的分析,本文将 ZUC 算法中基于 32 比特字的非线性函数转化为基于 16 比特半字的非线性函数,提出了基于 16 比特半字的猜测决定攻击,其计算复杂度为 2^{392},所需数据量为 9 个 32 比特密钥字,该结果优于已有的针对 ZUC 的猜测决定攻击.由于 SNOW3G 和 ZUC 的密钥规模都为 128 比特,本文的分析结果表明,这两个算法都能够抵抗猜测决定攻击.同时,本文的分析结果也表明,尽管 ZUC 算法的内部状态规模小于 SNOW3G,在抵抗猜测决定攻击方面,ZUC 明显优于 SNOW3G.针对 SNOW3G 和 ZUC 是否能够选取更优的猜测量对本文的攻击结果进行改进,是我们需要进一步研究的问题.

References:

[1] ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms UEA2&UIA2. Document 5: Design and evaluation report. Version 1.0, 2006. http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_42_Bangalore/Docs/S3060180.zip

[2] Ekdahl P, Johansson T. A new version of the stream cipher SNOW. In: Nyberg K, Heys H, eds. Proc. of the SAC 2002. LNCS 2595, Heidelberg: Springer-Verlag, 2002. 47−61. [doi: 10.1007/3-540-36492-7_5]

[3] Nyberg K, Wallen J. Improved linear distinguishers for SNOW 2.0. In: Matthew JB, ed. Proc. of the FSE 2006. LNCS 4047, Heidelberg: Springer-Verlag, 2006. 144−162. [doi: 10.1007/11799313_10]

[4] Baigneres T, Junod P, Vaudenay S. How far can we go beyond linear cryptanalysis? In: Pil JL, ed. Proc. of the ASIACRYPT 2004. LNCS 3329, Heidelberg: Springer-Verlag, 2004. 432−450. [doi: 10.1007/978-3-540-30539-2_31]

© 中国科学院软件研究所 http://www.jos.org.cn

[5] Biryukov A, Priemuth-Schmid D, Zhang B. Multiset collision attacks on reduced-round SNOW 3G and SNOW 3G⊕. In: Zhou JY, Moti Y, eds. Proc. of the ACNS 2010. LNCS 6123, Heidelberg: Springer-Verlag, 2010. 139−153. [doi: 10.1007/978-3-642-13708-2_9]

[6] Debraize B, Corbella IM. Fault analysis of the stream cipher snow 3G. In: Luca B, Israel K, David N, Elisabeth O, Seifert JP, eds. Proc. of the FDTC 2010. Lausanne: IEEE Computer Society, 2009. 103−110. http://doi.ieeecomputersociety.org/10.1109/FDTC.2009.33 [doi: 10.1109/FDTC.2009.33]

[7] ETSI/SAGE Specification. Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version 1.4, 2010.

[8] Feng XT. ZUC algorithm: 3GPP LTE international encryption standard. Information Security and Communications Privacy, 2011,19(12):45−46 (in Chinese with English abstract).

[9] Sun B, Tang XH, Li C. Preliminary cryptanalysis results of ZUC. In: Proc. of the Record of the 1st Int'l Workshop on ZUC Algorithm. Beijing, 2010. 18−19.

[10] Wu HJ. Cryptanalysis of the stream cipher ZUC in the 3GPP confidentiality & integrity algorithms 128-EEA3 & 128-EIA3. In: Proc. of the Record of the Sump Session in ASIACRYPT 2010. Singapore, 2010.

[11] ETSI/SAGE Specification. Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version 1.5, 2011.

[12] Ding L, Liu SK, Zhang ZY, Guan J. Guess and Determine attack on ZUC based on solving nonlinear equations. In: Proc. of the Record of the 1st Int'l Workshop on ZUC Algorithm. Beijing, 2010. 1−9.

[13] Zhou CF, Feng XT, Lin DD. The initialization stage analysis of ZUC v1.5. In: Lin DD, Tsudik G, Wang XY, eds. Proc. of the ACNS 2011. LNCS 7092, Heidelberg: Springer-Verlag, 2011. 40−53. [doi: 10.1007/978-3-642-25513-7_5]

[14] Hawkes P, Rose GG. Guess-and-Determine attacks on SNOW. In: Nyberg K, Heys H, eds. Proc. of the SAC 2002. LNCS 2595, Heidelberg: Springer-Verlag, 2002. 37−46. [doi: 10.1007/3-540-36492-7_4]

[15] Babbage S, De Canniere C, Lano J. Cryptanalysis of SOBER-t32. In: Thomas J, ed. Proc. of the FSE 2003. LNCS 2887, Heidelberg: Springer-Verlag, 2003. 111−128. [doi: 10.1007/978-3-540-39887-5_10]

[16] Mattsson J. A Guess-and-Determine attack on the stream cipher polar bear. In: Anne C, ed. Proc. of the SASC 2006. Leuven, 2006. 149−153.

[17] Feng XT, Liu J, Zhou ZC, Wu CK, Feng DG. A byte-based Guess and Determine attack on SOSEMANUK. In: Masayuki A, ed. Proc. of the ASIACRYPT 2010. LNCS 6477, Heidelberg: Springer-Verlag, 2010. 146−157. [doi: 10.1007/978-3-642-17373-8_9]

附中文参考文献:

[8] 冯秀涛.3GPP LTE 国际加密标准 ZUC 算法.信息安全与通信保密,2011,19(12):45−46.

关杰(1974－),女,河南郑州人,博士,副教授,主要研究领域为密码学,信息安全.
E-mail: guanjie007@163.com

刘树凯(1988－),男,硕士生,主要研究领域为流密码分析.
E-mail: wasd268@126.com

丁林(1987－),男,博士生,主要研究领域为流密码设计与分析.
E-mail: dinglin_cipher@163.com

© 中国科学院软件研究所 http://www.jos.org.cn