MORUS-1280-128 算法的区分分析

郑秀林[1,2] 宋海燕[2] 付伊鹏[2]

(北京电子科技学院信息安全系 北京 100070)[1] (西安电子科技大学通信工程学院 西安 710071)[2]

摘 要 MORUS算法是被提交到CAESAR竞赛中的一种认证加密算法,已经进入第三轮安全评估。对算法进行区分分析对于其安全性评估具有很重要的意义。以MORUS-1280-128为例,在nonce重用的情况下,对算法进行区分分析能够区分出密文的绝大部分比特,并通过寻找内部状态碰撞对算法进行标签伪造攻击。该研究结果对MORUS算法的安全性分析有很重要的意义。

关键词 认证加密,MORUS算法,区分分析,伪造攻击

中图法分类号 TN918.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.04.025

Distinguishing Attack of MORUS-1280-128

ZHENG Xiu-lin[1,2] SONG Hai-yan[2] FU Yi-peng[2]

(Department of Information Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China)[1]

(School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)[2]

Abstract MORUS is an authenticated cipher, which is submitted to CAESAR competition and has been selected into the third-round security evaluation stage. To study the distinguishing attack of MORUS is significant for its security evaluation. This paper studied the distinguishing attack of MORUS-1280-128 in a nonce-resuse scenario. By using this method, the majority ciphertext can be distinguished, and a collision in internal state can be found for a tag forgery attack. The paper's research results are of great significance for the safety analysis of MORUS.

Keywords Authenticated encryption, MORUS algorithm, Distinguishing attack, Forgery attack

1 引言

2013年1月,在美国国家标准技术研究院NIST的支持下,Bernstein等人[1]发起了CAESAR(Competition for Authenticated Encryption:Security,Applicability,Robustness)竞赛,面向全球征集认证加密算法。该竞赛共征集到57种算法,经过3轮筛选,预计在2017年12月底公布最终的胜选算法,目前正处于第三轮安全性评估阶段。

认证加密算法可同时保证信息的机密性[2]和完整性[3]。在被提交到CAESAR竞赛的算法中,MORUS算法[4]是一种基于序列密码设计的认证加密算法,由南洋理工大学的伍宏军和黄涛设计,目前已经进入第三轮安全评估阶段。MORUS算法是一个软硬件非常高效的认证加密算法族,根据其内部状态和密钥长度的不同,可将MORUS算法分为3个子算法:MORUS-640-128,MORUS-1280-128和MORUS-1280-256。MORUS的内部状态有5个寄存器,通过5轮相似的操作对状态进行一次更新。Dwivedi等人[5]对MORUS进行了基于SAT的状态恢复攻击,Zhang P等人[6]研究了MORUS算法初始化过程的混乱和扩散性质,Shi T等人[7]则通过理论推导证明了MORUS的认证安全性。到目前为止,对MORUS算法的公开研究很少。本文以MORUS-1280-128为例,首次对其进行了区分分析。首先在nonce重用的情况下,构造了一个区分器[8],它能够区分出两段密文的绝大部分比特;然后通过寻找内部状态碰撞对算法进行标签伪造攻击。实验结果表明内部碰撞只在理论上成立,MORUS算法能够抵抗伪造攻击,这对其安全性评估具有很重要的意义。

2 区分分析

区分分析[8]是Coppersmith等人于2002年提出的一种有效的密码分析方法,其目的是通过观察输入与输出之间的关系来判定密钥流是否是真随机序列。区分分析在攻击结果方面不如密钥恢复的攻击性强,但区分分析依然能证明密码算法存在的某些缺陷,许多序列密码都受到了区分分析的威胁,如Shannon[9],RC4[10],SNOW[11]等。

区分分析的关键是构造区分器,根据密钥流的某些弱点,区分器能够将一串密钥流和一串真随机序列区分开。假设有一个密钥流生成器X和一串给定的字符串z,区分器D将字符串z作为输入,输出密钥流序列或真随机序列。定义概率:

$P_0=P(D(z)=X|z$由密钥流生成器生成$)$

$P_1=P(D(z)=X|z$由真随机生成器生成$)$

到稿日期:2016-10-09 返修日期:2017-02-19

郑秀林(1956-),男,博士,教授,主要研究方向为序列密码设计与分析;**宋海燕**(1991-),女,硕士生,主要研究方向为基于序列密码的认证加密算法的设计与分析,E-mail:891581395@qq.com(通信作者);**付伊鹏**(1990-),男,硕士生,主要研究方向为基于分组密码的认证加密算法。

区分器 D 的区分优势 Adv_D 是将给定字符串判定为密钥流序列的概率与将真随机序列判定为密钥流序列的概率之差的绝对值，即 $Adv_D=|P_0-P_1|=|\varepsilon|$，其中 ε 为偏差。若将逼近 D 成立的概率记为 Pr，则 $\varepsilon=2Pr-1$，因此 $Pr=\frac{(1+\varepsilon)}{2}$。当 ε 不接近 0 时，$D(z)$ 是 X 的一个区分器，可以对 X 进行区分分析。

3　MORUS 算法描述

MORUS 算法分为 5 个阶段：初始化、关联数据处理、加密、标签生成过程和解密校验过程。本文只考虑前 3 个阶段，并将 MORUS-1280-128 作为研究对象，若无特别声明，文中的“MORUS”均指 MORUS-1280-128 算法。

3.1　符号说明

$\oplus$：按位异或运算；

$\&$：按位与运算；

$\|$：连接符号；

$\lll$：循环左移；

$\ggg$：循环右移；

$b^{(n)}$：长度为 n 的比特串，$b\in\{0,1\}$；

$|X|$：比特串 X 的比特长度；

$Rotl\_xxx\_yy(x,b)$：将 xxx 位的 x 分成 4 个 yy 位的字，对每个字向左循环移 b 位；

$Rotr\_xxx\_yy(x,b)$：将 xxx 位的 x 分成 4 个 yy 位的字，对每个字向右循环移 b 位；

LSB，MSB：分别代表最低比特位和最高比特位；

IV：128 比特初始向量(nonce)；

K：128 比特密钥；

$const_0$，$const_1$：两个不同的 128 比特常数；

S^i：第 i 步的内部状态，$0\leqslant i\leqslant 16$；

S_k^i：第 i 步第 k 轮的内部状态，$0\leqslant k\leqslant 4$；

$S_{k,j}^i$：S_k^i 的第 j 块 256 比特分组，$0\leqslant j\leqslant 4$。

3.2　状态更新函数

MORUS 算法的状态更新函数为 $StateUpdate(S,M)$，其中 S 是状态，M 是消息块，$|S|=|M|$。状态更新函数只使用了 3 个按位操作：循环左移($\lll$)、逻辑与($\&$)和异或($\oplus$)；每步用 5 轮相似的操作更新状态 S，每轮只更新两个状态寄存器：一个循环左移 w_i 位，另一个通过操作 $Rotl\_xxx\_yy(x,b_i)$ 更新，其中 $i\in\{0,1,2,3,4\}$，w_i 和 b_i 的取值如表 1 所列。MORUS 算法的状态更新函数 $StateUpdate(S,M)$ 如下。

$S^{i+1}=StateUpdate(S^i,m_i)$：

第一轮：

$S_{1,0}^i=Rotl\_256\_64(S_{0,0}^i\oplus(S_{0,1}^i\,\&\,S_{0,2}^i)\oplus S_{0,3}^i,b_0)$

$S_{1,3}^i=S_{0,3}^i\lll w_0$

$S_{1,1}^i=S_{0,1}^i$

$S_{1,2}^i=S_{0,2}^i$

$S_{1,4}^i=S_{0,4}^i$

第二轮到第四轮[6]：

For $k=1$ to 3

$S_{(k+1)\bmod 5,k}^i=Rotl\_256\_64(S_{k,k}^i\oplus(S_{k,(k+1)\bmod 5}^i\,\&\,S_{k,(k+2)\bmod 5}^i)\oplus S_{k,(k+3)\bmod 5}^i\oplus m_i,b_k)$

$S_{(k+1)\bmod 5,(k+3)\bmod 5}^i=S_{k,(k+3)\bmod 5}^i\lll w_k$

$S_{(k+1)\bmod 5,(k+1)\bmod 5}^i=S_{k,(k+1)\bmod 5}^i$

$S_{(k+1)\bmod 5,(k+2)\bmod 5}^i=S_{k,(k+2)\bmod 5}^i$

$S_{(k+1)\bmod 5,(k+4)\bmod 5}^i=S_{k,(k+4)\bmod 5}^i$

第五轮：

$S_{0,4}^{i+1}=Rotl\_256\_64(S_{4,4}^i\oplus(S_{4,0}^i\,\&\,S_{4,1}^i)\oplus S_{4,2}^i\oplus m_i,b_4)$

$S_{0,2}^{i+1}=S_{4,2}^i\lll w_4$

$S_{0,0}^{i+1}=S_{4,0}^i$

$S_{0,1}^{i+1}=S_{4,1}^i$

$S_{0,3}^{i+1}=S_{4,3}^i$

表 1　w_i 和 b_i 的取值

Table 1　Values of w_i and b_i

i	w_i	b_i
0	64	13
1	128	46
2	192	38
3	128	7
4	64	4

3.3　初始化阶段

在 MORUS 算法的初始化阶段将密钥 K 和初始向量 IV 装载到内部状态中，然后运行 16 步状态更新函数 $S^{i+1}=StateUpdate(S^i,m_i)$。密钥和初始向量的装载方式如下：

$S_{0,0}^{-16}=IV\,\|\,0^{(128)}$

$S_{0,1}^{-16}=K\,\|\,K$

$S_{0,2}^{-16}=1^{(256)}$

$S_{0,3}^{-16}=0^{(256)}$

$S_{0,4}^{-16}=const_0\,\|\,const_1$

在装载密钥和初始向量后，采用状态更新函数对内部状态进行 16 步更新：

For $i=-16$ to -1

$S^{i+1}=StateUpdate(S^i,m_i)$

然后将密钥与第二个状态分组做异或运算：

$S_{0,1}^0=S_{0,1}^0\oplus(K\,\|\,K)$

3.4　关联数据处理和加密阶段

在关联数据处理阶段，将 32 字节长的关联数据分组 AD_i(若最后一块关联数据的长度不足 32 字节，则在其后面填充若干个 0，使之达到 32 字节)用于更新内部状态，设关联数据的长度为 $adlen$，令 $u=\lceil adlen/256\rceil$。

For $i=0$ to $u-1$，进行关联数据处理：

$S^{i+1}=StateUpdate(S^i,AD_i)$

处理完关联数据后，对 P_i 和密钥流进行异或运算后输出密文 C_i，然后用 32 字节的明文分组 P_i(若最后一块明文的长度不足 32 字节，则在其后填充若干个 0，使之达到 32 字节)更新内部状态。设明文长度为 $msglen$，令 $v=\lceil msglen/256\rceil$。For $i=0$ to $v-1$，加密和状态更新过程如下：

$C_i=P_i\oplus S_0^{u+i}\oplus(S_1^{u+i}\lll 192)\oplus(S_2^{u+i}\,\&\,S_3^{u+i})$

$S^{u+i+1}=StateUpdate(S^{u+i},P_i)$

4　MORUS 算法的区分分析

本节在 nonce 重用的情况下对 MORUS 算法构造了一个区分器。设关联数据长度 $|AD|=256$，明文消息长度 $|P|=$

256，$S^0=(s_0,s_1,s_2,s_3,s_4)$为初始化结束后的状态，那么处理关联数据后的状态为：

$$S^1=(x_0,x_1,x_2,x_3,x_4)=StateUpdate(S^0,AD) \quad (1)$$

加密阶段生成的密文 C 和更新后的状态 S^2 为：

$$C=P\oplus x_0\oplus(x_1 <<<192)\oplus(x_2 \& x_3) \quad (2)$$

$$S^2=(z_0,z_1,z_2,z_3,z_4)=StateUpdate(S^1,P) \quad (3)$$

定理 1 设 MORUS 算法经过密钥 K 和初始向量 IV（$|K|=|IV|=128$）初始化后的状态为 $S^0=(s_0,s_1,s_2,s_3,s_4)$，附加数据 $AD_1=0^{(256)}$，$AD_2=1\|0^{(255)}$，X 和 Y 分别是处理完 AD_1 和 AD_2 后的内部状态：

$$X=(x_0,x_1,x_2,x_3,x_4)=StateUpdate(S^0,AD_1) \quad (4)$$

$$Y=(y_0,y_1,y_2,y_3,y_4)=StateUpdate(S^0,AD_2) \quad (5)$$

那么，

1）$x_0=y_0$；

2）x_1 和 y_1 的第 147 位不同；

3）x_2 和 y_2 的第 219 位不同；

4）x_3 和 y_3 的第 12 位和第 58 位不同；

5）x_4 和 y_4 之和的第 23 位和第 61 位不同，第 143 位不同的概率为 1/2。

注：记 256 比特字 x_i 和 y_i 的最高位为第 1 位，最低位为第 256 位（$0\leqslant i\leqslant 4$）。

证明：设使用 AD_1 更新后的状态为

$$(x_0,x_1,x_2,x_3,x_4)=StateUpdate((s_0,s_1,s_2,s_3,s_4),AD_1) \quad (6)$$

其中：

$$x_0=Rotl\_256\_64(s_0\oplus(s_1\&s_2)\oplus s_3,13)<<<192 \quad (7)$$

$$x_1=Rotl\_256\_64(s_1\oplus(s_2\&(s_3<<<64))\oplus s_4\oplus AD_1,46)<<<128 \quad (8)$$

$$x_2=Rotl\_256\_64(s_2\oplus((s_3<<<64)\&(s_4<<<128))\oplus(x_0>>>192)\oplus AD_1,38)<<<64 \quad (9)$$

$$x_3=Rotl\_256\_64((s_3<<<64)\oplus((s_4<<<128)\&x_0)\oplus(x_1>>>128)\oplus AD_1,7) \quad (10)$$

$$x_4=Rotl\_256\_64((s_4<<<128)\oplus(x_0\&x_1)\oplus(x_2>>>64)\oplus AD_1,4) \quad (11)$$

状态 $X=(x_0,x_1,x_2,x_3,x_4)$ 和 $Y=(y_0,y_1,y_2,y_3,y_4)$ 中的元素均由 4 个 64 比特字组成，记为 $x_j=(x_{j0},x_{j1},x_{j2},x_{j3})$，$y_j=(y_{j0},y_{j1},y_{j2},y_{j3})$，其中，$0\leqslant j\leqslant 4$，$0\leqslant i\leqslant 3$。

1）由式(7)可知 x_0 与 AD_1 无关，同理，y_0 与 AD_2 无关，故 $x_0=y_0$。

2）设：

$$a_1=s_1\oplus(s_2\&(s_3<<<64))\oplus s_4 \quad (12)$$

$$b_1=(b_{10},b_{11},b_{12},b_{13})=Rotl\_256\_64(a_1\oplus AD_1,46)=Rotl\_256\_64(a_1,46) \quad (13)$$

循环左移 128 位后，

$$x_1=(x_{10},x_{11},x_{12},x_{13})=(b_{12},b_{13},b_{10},b_{11}) \quad (14)$$

当附加数据为 $AD_2=1\|0^{(255)}$ 时，设：

$$c_1=(c_{10},c_{11},c_{12},c_{13})=Rotl\_256\_64(a_1\oplus AD_2,46)=Rotl\_256\_64(a_1\oplus(1\|0^{(255)}),46) \quad (15)$$

其中，

$$c_{10}=b_{10}\oplus(0^{(18)}\|1\|0^{(45)})=x_{12}\oplus(0^{(18)}\|1\|0^{(45)}) \quad (16)$$

$$c_{11}=b_{11}=x_{13} \quad (17)$$

$$c_{12}=b_{12}=x_{10} \quad (18)$$

$$c_{13}=b_{13}=x_{11} \quad (19)$$

将 c_1 循环左移 128 位后，得到：

$$y_1=(y_{10},y_{11},y_{12},y_{13})=(c_{12},c_{13},c_{10},c_{11})=(x_{10},x_{11},x_{12}\oplus(0^{(18)}\|1\|0^{(45)}),x_{13}) \quad (20)$$

由式(20)可知，x_1 和 y_1 的第 147 位不同。

3）设：

$$a_2=s_2\oplus((s_3<<<64)\&(s_4<<<128))\oplus(x_0>>>192) \quad (21)$$

$$b_2=(b_{20},b_{21},b_{22},b_{23})=Rotl\_256\_64(a_2\oplus AD_1,38)=Rotl\_256\_64(a_2,38) \quad (22)$$

循环左移 64 位后，

$$x_2=(x_{20},x_{21},x_{22},x_{23})=(b_{21},b_{22},b_{23},b_{20}) \quad (23)$$

当附加数据为 $AD_2=1\|0^{(255)}$ 时，设：

$$c_2=(c_{20},c_{21},c_{22},c_{23})=Rotl\_256\_64(a_2\oplus AD_2,38)=Rotl\_256\_64(a_2\oplus(1\|0^{(255)}),38) \quad (24)$$

其中，

$$c_{20}=b_{20}\oplus(0^{(26)}\|1\|0^{(37)})=x_{23}\oplus(0^{(26)}\|1\|0^{(37)}) \quad (25)$$

$$c_{21}=b_{21}=x_{20} \quad (26)$$

$$c_{22}=b_{22}=x_{21} \quad (27)$$

$$c_{23}=b_{23}=x_{22} \quad (28)$$

将 c_2 循环左移 64 位后，得到：

$$y_2=(y_{20},y_{21},y_{22},y_{23})=(c_{21},c_{22},c_{23},c_{20})=(x_{20},x_{21},x_{22},x_{23}\oplus(0^{(26)}\|1\|0^{(37)})) \quad (29)$$

由式(29)知，x_2 和 y_2 的第 219 位不同。

4）设：

$$a_3=(s_3<<<64)\oplus((s_4<<<128)\&x_0) \quad (30)$$

$$x_3=Rotl\_256\_64(a_3\oplus(x_1>>>128)\oplus AD_1,7)=Rotl\_256\_64(a_3\oplus(x_{12},x_{13},x_{10},x_{11}),7)=((a_{30}\oplus x_{12})<<<7,(a_{31}\oplus x_{13})<<<7,(a_{32}\oplus x_{10})<<<7,(a_{33}\oplus x_{11})<<<7)=(x_{30},x_{31},x_{32},x_{33}) \quad (31)$$

当附加数据为 $AD_2=1\|0^{(255)}$ 时，

$$y_3=Rotl\_256\_64(a_3\oplus(y_1>>>128)\oplus AD_2,7)=Rotl\_256\_64(a_3\oplus(x_{12}\oplus(0^{(18)}\|1\|0^{(45)}),x_{13},x_{10},x_{11})\oplus(1\|0^{(255)}),7)=Rotl\_256\_64(a_3\oplus(x_{12}\oplus(1\|0^{(17)}\|1\|0^{(45)}),x_{13},x_{10},x_{11}),7)=((a_{30}\oplus x_{12}\oplus(1\|0^{(17)}\|1\|0^{(45)}))<<<7,(a_{31}\oplus x_{13})<<<7,(a_{32}\oplus x_{10})<<<7,(a_{33}\oplus x_{11})<<<7)=(x_{30}\oplus(0^{(11)}\|1\|0^{(45)}\|1\|0^{(6)}),x_{31},x_{32},x_{33}) \quad (32)$$

由式(32)可知，x_3 和 y_3 的第 12 位和第 58 位不同。

5）设：

$$a_4=s_4<<<128 \quad (33)$$

$$x_4 = Rotl\_256\_64(a_4 \oplus (x_0 \& x_1) \oplus (x_2 >>> 64) \oplus AD_1, 4)$$
$$= Rotl\_256\_64(a_4 \oplus (x_0 \& x_1) \oplus (x_{23}, x_{20}, x_{21}, x_{22}), 4)$$
$$= ((a_{40} \oplus (x_{00} \& x_{10}) \oplus x_{23}) <<< 4, (a_{41} \oplus (x_{01} \& x_{101}) \oplus x_{20}) <<< 4, (a_{42} \oplus (x_{02} \& x_{12}) \oplus x_{21}) <<< 4, (a_{43} \oplus (x_{03} \& x_{13}) \oplus x_{22}) <<< 4)$$
$$= (x_{40}, x_{41}, x_{42}, x_{43}) \quad (34)$$

当附加数据为 $AD_2 = 1 \| 0^{(255)}$ 时,

$$y_4 = Rotl\_256\_64(a_4 \oplus (y_0 \& y_1) \oplus (y_2 >>> 64) \oplus AD_2, 4)$$
$$= Rotl\_256\_64(a_4 \oplus (x_0 \& y_1) \oplus (x_{23} \oplus (0^{(26)} \| 1 \| 0^{(37)}), x_{20}, x_{21}, x_{22}) \oplus (1 \| 0^{(255)}), 4)$$
$$= ((a_{40} \oplus (x_{00} \& x_{10}) \oplus x_{23} \oplus (1 \| 0^{(25)} \| 1 \| 0^{(37)})) <<< 4, (a_{41} \oplus (x_{01} \& x_{11}) \oplus x_{20}) <<< 4, (a_{42} \oplus (x_{02} \& (x_{12} \oplus (0^{(18)} \| 1 \| 0^{(45)}))) \oplus x_{21}) <<< 4, (a_{43} \oplus (x_{03} \& x_{13}) \oplus x_{22}) <<< 4)$$
$$= ((a_{40} \oplus (x_{00} \& x_{10}) \oplus x_{23}) <<< 4 \oplus (0^{(22)} \| 1 \| 0^{(37)} \| 1 \| 0^{(3)}), (a_{41} \oplus (x_{01} \& x_{11}) \oplus x_{20}) <<< 4, (a_{42} \oplus (x_{02} \& (x_{12} \oplus (0^{(18)} \| 1 \| 0^{(45)}))) \oplus x_{21}) <<< 4, (a_{43} \oplus (x_{03} \& x_{13}) \oplus x_{22}) <<< 4) (a_{43} \oplus (x_{03} \& x_{13}) \oplus x_{22}) <<< 4)$$
$$= (x_{40} \oplus (0^{(22)} \| 1 \| 0^{(37)} \| 1 \| 0^{(3)}), x_{41}, x'_{42}, x_{43}) \quad (35)$$

由式(35)可知,x_4 和 y_4 的第 23 位和第 61 位不同,第 143 位不同的概率为 1/2。证毕。

接下来构造一个区分器。令 $AD_1, AD_2, X = (x_1, x_2, x_3, x_4)$ 和 $Y = (y_0, y_1, y_2, y_3, y_4)$ 的定义同定理 1,其中 $x_j = (x_{j0}, x_{j1}, x_{j2}, x_{j3})$, $y_j = (y_{j0}, y_{j1}, y_{j2}, y_{j3})$, $|x_{ji}| = |y_{ji}| = 64$, $0 \leqslant j \leqslant 4$, $0 \leqslant i \leqslant 3$。$M$ 是一个 256 比特明文分组,在初始化相同、附加数据不同的情况下加密明文 M:

$$C_1 = M \oplus x_0 \oplus (x_1 <<< 192) \oplus (x_2 \& x_3) \quad (36)$$
$$C_2 = M \oplus y_0 \oplus (y_1 <<< 192) \oplus (y_2 \& y_3) \quad (37)$$

注意到:

$$(x_1 <<< 192) = ((x_{10}, x_{11}, x_{12}, x_{13}) <<< 192) = (x_{13}, x_{10}, x_{11}, x_{12}) \quad (38)$$

由式(20)可知:

$$y_1 = (y_{10}, y_{11}, y_{12}, y_{13}) = (x_{10}, x_{11}, x_{12} \oplus (0^{(18)} \| 1 \| 0^{(45)}), x_{13}) \quad (39)$$

则

$$(y_1 <<< 192) = ((x_{10}, x_{11}, x_{12} \oplus (0^{(18)} \| 1 \| 0^{(45)}), x_{13}) <<< 192) = (x_{13}, x_{10}, x_{11}, x_{12} \oplus (0^{(18)} \| 1 \| 0^{(45)})) \quad (40)$$

故 $(x_1 <<< 192)$ 和 $(y_1 <<< 192)$ 的第 211 位不同。

由式(36)可知:

$$(x_2 \& x_3) = ((x_{20} \& x_{30}), (x_{21} \& x_{31}), (x_{22} \& x_{32}), (x_{23} \& x_{33})) \quad (41)$$

由式(29)和(32)可知:

$$y_2 = (y_{20}, y_{21}, y_{22}, y_{23}) = (x_{20}, x_{21}, x_{22}, x_{23} \oplus (0^{(26)} \| 1 \| 0^{(37)})) \quad (42)$$
$$y_3 = (y_{30}, y_{31}, y_{32}, y_{33}) = (x_{30} \oplus (0^{(11)} \| 1 \| 0^{(45)} \| 1 \| 0^{(6)}), x_{31}, x_{32}, x_{33}) \quad (43)$$

故

$$(y_2 \& y_3) = ((x_{20} \& (x_{30} \oplus (0^{(11)} \| 1 \| 0^{(45)} \| 1 \| 0^{(6)}))), (x_{21} \& x_{31}), (x_{22} \& x_{32}), ((x_{23} \oplus (0^{(26)} \| 1 \| 0^{(37)})) \& x_{33})) \quad (44)$$

根据式(41)和式(44)可知,$(x_2 \& x_3)$ 和 $(y_2 \& y_3)$ 的第 12 位、第 58 位和第 219 位不同的概率均为 1/2。

由以上分析可知,C_1 和 C_2 只有 4 个位置不同:第 211 位不同的概率为 1,第 12 位、第 58 位和第 219 位不同的概率均为 1/2。因此,若给定用 (AD_1, M) 加密的密文 C_1,攻击者可以确定加密得到的密文 C_2 的 253 比特。

从式(36)和式(37)可以看出,该方法同样适用于两个不同的明文 M_1 和 M_2,即对于上面定义的 AD_1 和 AD_2,明文 M_1 和 M_2 在除了第 12 位、第 58 位、第 211 位和第 219 位之外的 253 比特中有 k 个位置不同,那么其经过加密后,密文 C_1 和 C_2 对应的 k 个位置不同。

5　内部状态碰撞和伪造攻击

本节尝试找到状态更新函数 $StateUpdate(S, M)$ 的一个碰撞,利用该碰撞构造一个伪造的标签,使其能够通过认证算法,以达到伪造攻击的目的。

定义 1[12]　设 $M, x_i \in Z_2^{256}, i \geqslant 0, w_i \leqslant 256, b_i \leqslant 64$ 是一些轮常数,那么函数 $F_M: (Z_2^{256})^5 \to (Z_2^{256})^5$ 是 $(Z_2^{256})^5$ 上的一个置换:

$$F_M(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}) = (Rotl\_256\_64(x_i \oplus (x_{i+1} \& x_{i+2}) \oplus x_{i+3} \oplus M, b_i), x_{i+1}, x_{i+2}, (x_{i+3} <<< w_i), x_{i+4}) \quad (45)$$

根据函数 F_M 的定义,状态更新函数 $StateUpdate(S, M)$ 相当于连续应用 5 次 F_{m_i},其中,当 $i \in \{1,2,3,4\}$ 时,$m_i = M$;当 $i = 0$ 时,$m_i = 0^{(256)}$。

$$(s_i, s_{(i+1)}, s_{(i+2)}, s_{(i+3)}, s_{(i+4)}) = F_{m_i}(s_i, s_{(i+1)}, s_{(i+2)}, s_{(i+3)}, s_{(i+4)}) \quad (46)$$

其下标 $(i+1), \cdots, (i+4)$ 均取模 5 后的值。对于固定的 $M \in Z_2^{256}$,状态更新函数 $StateUpdate(S, M)$ 是 $(Z_2^{256})^5$ 上的一些置换的复合。接下来构造函数 F_M 的碰撞。

对于所有的 $M_1, M_2 \in Z_2^{256}$,向量 $(x_0, x_1, x_2, x_3, x_4) \in (Z_2^{256})^5$,有:

$$F_{M_2}(M_1 \oplus M_2 \oplus x_0, x_1, x_2, x_3, x_4)$$
$$= (Rotl\_256\_64(M_1 \oplus M_2 \oplus x_0 \oplus (x_1 \& x_2) \oplus x_3 \oplus M_2, b_i), x_1, x_2, (x_3 <<< w_i), x_4)$$
$$= (Rotl\_256\_64(M_1 \oplus x_0 \oplus (x_1 \& x_2) \oplus x_3, b_i), x_1, x_2, (x_3 <<< w_i), x_4)$$
$$= F_{M_1}(x_0, x_1, x_2, x_3, x_4) \quad (47)$$

根据 F_M 和 $StateUpdate(S, M)$ 的关系,很容易得出如下结论:

$$StateUpdate((x_0, x_1, x_2, x_3, x_4), M_1) = StateUpdate((x_0, M_1 \oplus M_2 \oplus x_1, M_1 \oplus M_2 \oplus x_2, M_1 \oplus M_2 \oplus x_3, M_1 \oplus M_2 \oplus x_4), M_2) \quad (48)$$

接下来利用式(48)在 IV 重用的条件下构造一个标签伪造攻击。

设 AD_1 和 M_1 分别是 256 比特的附加数据和 256 比特的

消息块，$S^0=(s_0,s_1,s_2,s_3,s_4)$是由密钥 K 和初始向量 IV 经过初始化之后的状态，处理完附加数据 AD_1 后的状态：$S^1=(x_0,x_1,x_2,x_3,x_4)=StateUpdate(S^0,AD_1)$。本文试图找到一个 256 比特的数据块 ΔM 和一个 256 比特的附加数据 AD_2（$AD_2\neq AD_1$），使得 $StateUpdate(S^0,AD_1)$ 和 $StateUpdate(S^0,AD_2)$满足如下关系：

$$\begin{aligned}&StateUpdate(S^0,AD_2)\\&=(x_0,\Delta M\oplus x_1,\Delta M\oplus x_2,\Delta M\oplus x_3,\Delta M\oplus x_4)\\&=(x_0,x_1,x_2,x_3,x_4)\oplus(0^{(256)},\Delta M,\Delta M,\Delta M,\Delta M)\\&=StateUpdate(S^0,AD_1)\oplus(0^{(256)},\Delta M,\Delta M,\Delta M,\Delta M)\end{aligned}\tag{49}$$

如果 ΔM 和 AD_2 存在，那么根据式(48)可以构造消息块 $M_2=M_1\oplus\Delta M$，内部状态在加密完成后将会产生碰撞：

$$\begin{aligned}&StateUpdate(StateUpdate(S^0,AD_1),M_1)\\&=StateUpdate((x_0,x_1,x_2,x_3,x_4),M_1)\\&=StateUpdate((x_0,M_1\oplus M_2\oplus x_1,M_1\oplus M_2\oplus x_2,\\&\quad M_1\oplus M_2\oplus x_3,M_1\oplus M_2\oplus x_4),M_2)\\&=StateUpdate((x_0,\Delta M\oplus x_1,\Delta M\oplus x_2,\Delta M\oplus x_3,\\&\quad\Delta M\oplus x_4),M_2)\\&=StateUpdate(StateUpdate(S^0,AD_2),M_2)\end{aligned}\tag{50}$$

由此可见，在 nonce 重用的情况下，两个消息块 M_1 和 M_2（$M_2=M_1\oplus\Delta M$）在加密后产生相同的内部状态。此外，由于(AD_1,M_1)和(AD_2,M_2)有相同的 *adlen* 和 *msglen*，因此该碰撞最终会产生相同的标签。

下面寻找满足要求的 ΔM 和 AD_2。

令 $X=(x_0,x_1,x_2,x_3,x_4)=StateUpdate(S^0,AD_1)$，$Y=(y_0,y_1,y_2,y_3,y_4)=StateUpdate(S^0,AD_2)$，$x_1\oplus y_1=x_2\oplus y_2=x_3\oplus y_3=x_4\oplus y_4=\Delta M$，$AD_1\oplus AD_2=\Delta A$，显然 $x_0=y_0$。

由 $x_1\oplus y_1$ 得：

$$\begin{aligned}x_1\oplus y_1=&Rotl\_256\_64(s_1\oplus(s_2\&(s_3\lll w_0))\oplus s_4\oplus\\&AD_1,b_1)\lll w_3\oplus Rotl\_256\_64(s_1\oplus(s_2\&\\&(s_3\lll w_0))\oplus s_4\oplus AD_2,b_1)\lll w_3\end{aligned}$$

故

$$\begin{aligned}&Rotr\_256\_64((x_1\oplus y_1)\ggg w_3,b_1)\\&=s_1\oplus(s_2\&(s_3\lll w_0)\oplus s_4\oplus AD_1\oplus s_1\oplus(s_2\&\\&\quad(s_3\lll w_0)\oplus s_4\oplus AD_2\\&=AD_1\oplus AD_2\end{aligned}\tag{51}$$

同理可得：

$$Rotr\_256\_64((x_2\oplus y_2)\ggg w_4,b_2)=AD_1\oplus AD_2\tag{52}$$

$$Rotr\_256\_64(x_3\oplus y_3,b_3)=(x_1\ggg w_3)\oplus(y_1\ggg w_3)\oplus AD_1\oplus AD_2\tag{53}$$

$$\begin{aligned}&Rotr\_256\_64(x_4\oplus y_4,b_4)\\&=(x_0\&x_1)\oplus(y_0\&y_1)\oplus(x_2\ggg w_4)\oplus(y_2\ggg\\&\quad w_4)\oplus AD_1\oplus AD_2\end{aligned}\tag{54}$$

根据式(51)－式(54)得到关于$(x_0,\Delta M,\Delta A)$的方程组：

$$\begin{cases}Rotr\_256\_64(\Delta M\ggg w_3,b_1)=\Delta A\\Rotr\_256\_64(\Delta M\ggg w_4,b_2)=\Delta A\\Rotr\_256\_64(\Delta M,b_3)=(\Delta M\ggg w_3)\oplus\Delta A\\Rotr\_256\_64(\Delta M,b_4)=(\Delta M\ggg w_4)\oplus\Delta A\oplus\\\qquad\qquad(x_0\&\Delta M)\end{cases}\tag{55}$$

将式(55)的前 3 个方程的变量和轮常数通过模 8 变成 8 比特长后再进行穷搜，得到一个平凡解：$(\Delta M,\Delta A)=(0,0)$。这说明内部状态在实际应用中不可能发生碰撞。

结束语 MORUS 算法是被提交到 CAESAR 竞赛中的一种认证加密算法，目前已经进入第三轮安全评估。本文在 nonce 重用的情况下构造了一个区分器，它能够区分密文的绝大部分比特，并在理论上找到了内部状态的一个碰撞，标签伪造攻击没有对 MORUS 的安全性产生威胁，这说明 MORUS 算法是一种设计良好的认证加密算法。

参 考 文 献

[1] CAESAR-Competition for Authenticated Encryption:Security, Applicability, and Robustness[OL]. http://competitions.cr.yp.to/caesar.html.

[2] DAEMEN J, RIJMEN V. AES and the wide trail design strategy[J]. Lecture Notes in Computer Science, 2002, 2260: 108-109.

[3] BERTONI G, DAEMEN J, PEETERS M, et al. Keccak[M]// Advances in Cryptology-EUROCRYPT 2013. 2013: 313-314.

[4] WU H J, HUANG T. The Authenticated Cipher MORUS[EB/OL]. http://competitions.cr.yp.to/caesar-submissions.html.

[5] DWIVEDI A D, KLOUCEK M, MORAW-IECKI P, et al. SAT-based Cryptanalysis of Authenticated Ciphers from the CAESAR Competition[C]// International Conference on Security & Cryptography. 2017.

[6] ZHANG P, GUAN J, LI J Z, et al. Research on the Confusion and Diffusion Properties of the Initialization of MORUS[J]. Journal of Cryptologic Research, 2015, 2(6): 536-548. (in Chinese)
张沛，关杰，李俊志，等. MORUS 算法初始化过程的混乱与扩散性质研究[J]. 密码学报，2015，2(6)：536-548.

[7] SHI T, GUAN J, LI J, et al. Improved Collision Cryptanalysis of Authenticated Cipher MORUS[C]// International Conference on Artificial Intelligence & Industrial Engineering. 2016.

[8] COPPERSMITH D, HALEVI S, JUTLA C. Cryptanalysis of Stream Ciphers with Linear Masking[C]// Advances in Cryptology-CRYPTO 2002. Springer Heidelberg, 2002: 515-532.

[9] CHANG Y Q, JIN C H. Linear Distinguishing Attack on Shannon Algorithm[J]. Journal of Electronics & Information Technology, 2011, 33(1): 190-193. (in Chinese)
常亚勤，金晨辉. 对 Shannon 算法的线性区分攻击[J]. 电子与信息学报，2011，33(1)：190-193.

[10] MAITRA S, PAUL G, GUPTA S S. Attack on broadcast RC4 revisited[M]// Fast Software Encryption. Springer Berlin Heidelberg, 2011: 199-217.

[11] WATANABE D, BIRYUKOV A, CANNIERE C D. A Distinguishing Attack of SNOW 2.0 with Linear Masking Method[C]// Selected Areas in Cryptography, International Workshop(SAC 2003). Ottawa, Canada, DBLP. 2003: 222-233.

[12] MILEVA A, DIMITROVA V, VELICHKO V V. Analysis of the Authenticated Cipher MORUS(v1)[M]// Cryptography and Information Security in the Balkans. Springer International Publishing, 2015.