基于REESSE3+算法的改进算法

董大强 殷新春
(扬州大学信息工程学院 扬州 225100)

摘 要 REESSE3+算法是苏盛辉教授于2014年提出的一个8轮迭代的分组密码算法。本文在REESSE3+算法的基础上做出了一些改进,提出了一种新的改进算法。由于REESSE3+算法受到了来学嘉教授提出的IDEA算法的启发,采用了3个不相容的群运算来保证其安全性,因此采用来学嘉教授提出的马尔科夫密码模型来对REESSE3+(16)算法和16位输入的改进算法进行比较。通过实验发现,在面对差分攻击时,16位输入的改进算法比原REESSE3+(16)算法更加安全。

关键词 REESSE3+算法,改进算法,马尔科夫密码,分组密码

中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.024

New Improved Algorithm Based on REESSE3+

DONG Da-qiang YIN Xin-chun
(School of Information Engineering, Yangzhou University, Yangzhou 225100, China)

Abstract REESSE3+ is an 8 rounds block cipher algorithm proposed by Professor Su in 2014. Based on the REESSE3+, this paper made some improvements. Since REESSE3+ is inspired by IDEA which was proposed by Professor Lai, three incompatible group operations were used to ensure their security, and we used the Markov model proposed by Professor Lai to make a comparison between REESSE3+(16) and the 16 bit input of improved algorithm. Through experiments, we found that in the face of differential crypt analysis, the 16 bit input of improved algorithm is more secure than the original REESSE3+(16).

Keywords REESSE3+, Improved algorithm, Markov cipher, Block cipher

1 引言

REESSE3+算法[1]是由苏盛辉教授在REESSE对称密钥密码体制[2]的基础上提出的。本文在REESSE3+算法的基础上做出了一些改进,以提高其面对差分攻击[3-6]时的安全性。

由于REESSE3+算法受到了IDEA算法[7-8]的启发,采用3个不同的群运算来构成轮函数[9],因此本文采用来学嘉教授提出的马尔科夫密码模型[7]对REESSE3+(16)算法和16位输入的改进算法的安全性进行对比。

本文第2节介绍了REESSE3+算法;第3节介绍了改进算法;第4节介绍了马尔科夫密码和其抵抗差分攻击[10-11]的证明方法;第5节对比了REESSE3+(16)算法和16位输入的改进算法的安全性;最后给出了结论和未来的研究方向。

2 REESSE3+算法

REESSE3+算法是一个分组密码[12-13]算法,是对IDEA算法的一个扩展,其分组长度由IDEA的64位扩展到128位,密钥长度由128位扩展到256位,并且轮函数也做出了大量的改变。本节简单介绍REESSE3+算法。

2.1 REESSE3+算法的描述

REESSE3+算法由8轮迭代和1轮输出变换组成,该算法的分组长度为128位,用X表示一个分组,并且X被分成8个子块,即$X_1,X_2,X_3,X_4,X_5,X_6,X_7$和$X_8$,$X_i$表示16位的输入;输出为128位,用$Y$表示,同样$Y$也被分成8个子块,即$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7$和$Y_8$,$Y_i$表示16位的输出;用$Z$表示256位的会话密钥,$Z_i^{(j)}$表示16位的子密钥,其中$1\leqslant i\leqslant 8, 1\leqslant j\leqslant 9$,并且$i$表示子块数,$j$表示轮数。在后文图1中,$a\oplus b$表示两个16位的子块$a$和$b$按位异或,$a[+]b$表示两个16位子块$a$和$b$相加后模$2^{16}$,$a\odot b$表示两个16位

到稿日期:2016-11-30 返修日期:2017-03-21 本文受国家自然科学基金项目(61472343)资助。

董大强(1991-),男,硕士,主要研究方向为分组密码,E-mail:1666913344@qq.com;殷新春(1962-),男,博士,教授,博士生导师,主要研究方向为密码学、软件质量保障、高性能计算等,E-mail:xcyin@yzu.edu.cn(通信作者)。

子块相乘后模$(2^{16}+1)$，在乘法模中 0 代表2^{16}。

2.2 REESSE3＋算法的子密钥

按顺序将密钥 Z 的最左边的 128 位分成 8 个子块，每个子块 16 位，分别用 $Z_1^{(1)}, Z_2^{(1)}, Z_3^{(1)}, Z_4^{(1)}, Z_5^{(1)}, Z_6^{(1)}, Z_7^{(1)}$ 和 $Z_8^{(1)}$ 表示，用作第一轮加密。然后将密钥 Z 循环左移 25 位，并按顺序将最左边的 128 位分成 8 个子块，每个子块 16 位，分别用 $Z_1^{(2)}, Z_2^{(2)}, Z_3^{(2)}, Z_4^{(2)}, Z_5^{(2)}, Z_6^{(2)}, Z_7^{(2)}$ 和$Z_8^{(2)}$表示，用作第二轮加密。其他轮的密钥重复上述过程。

解密子密钥的使用由加密子密钥得到，详细过程请参考文献[1]。

2.3 REESSE3＋算法的加密和解密过程

REESSE3＋算法的加密过程如图 1 所示，输入明文和加密子密钥，经过 8 轮迭代和 1 轮输出变换后得到密文。REESSE3＋算法的解密过程与加密过程相同，只是输入部分变成密文和解密子密钥，输出解密后的明文。

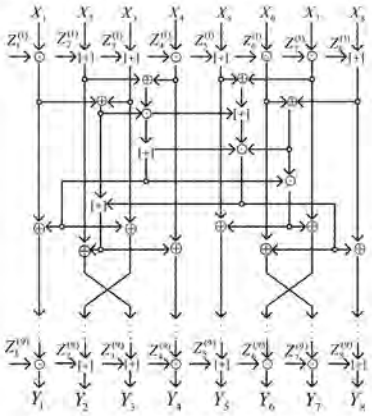

图 1　REESSE3＋算法的结构

3 改进算法

本节主要介绍改进算法与原 REESSE3＋算法的不同点。两者基本相同，只是改进算法在混合密钥和交换过程上做出了一些改变。

3.1 改进算法的描述

改进算法与原 REESSE3＋算法类似，其混合密钥的过程由原来的⊙[＋][＋]⊙[＋]⊙⊙[＋]变成了[＋]⊙⊙[＋][＋]⊙⊙[＋]，同时每轮最后多了一个 X_4 和 X_5 的交换。其中，增加交换可以提高算法扩散的速度，混合密钥的修改是为了保证加解密算法的一致性。其加密过程的密钥调度与原 REESSE3＋算法相同，不过解密过程的密钥调度与原算法有所区别。其他过程与原 REESSE3＋算法基本相同。

3.2 改进算法的密钥调度

改进算法的加密密钥与原算法的相同，详细可以参考第 2.2 节；解密密钥与原算法的有所不同，改进算法的解密密钥由加密密钥得到，首先由 Z 得到 8 * 9 个加密子密钥 $Z_1^{(1)}, Z_2^{(1)}, Z_3^{(1)}, Z_4^{(1)}, Z_5^{(1)}, Z_6^{(1)}, Z_7^{(1)}, Z_8^{(1)}, \cdots, Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}, Z_5^{(9)}, Z_6^{(9)}, Z_7^{(9)}, Z_8^{(9)}$，然后按照表 1 所列规则使用子密钥。

表 1　改进算法的解密密钥

轮数	解密子密钥							
1	$-Z_1^{(9)}$	$Z_2^{(9)-1}$	$Z_3^{(9)-1}$	$-Z_4^{(9)}$	$-Z_5^{(9)}$	$Z_6^{(9)-1}$	$Z_7^{(9)-1}$	$-Z_8^{(9)}$
2	$-Z_1^{(8)}$	$Z_3^{(8)-1}$	$Z_2^{(8)-1}$	$-Z_5^{(8)}$	$-Z_4^{(8)}$	$Z_7^{(8)-1}$	$Z_6^{(8)-1}$	$-Z_8^{(8)}$
3	$-Z_1^{(7)}$	$Z_3^{(7)-1}$	$Z_2^{(7)-1}$	$-Z_5^{(7)}$	$-Z_4^{(7)}$	$Z_7^{(7)-1}$	$Z_6^{(7)-1}$	$-Z_8^{(7)}$
4	$-Z_1^{(6)}$	$Z_3^{(6)-1}$	$Z_2^{(6)-1}$	$-Z_5^{(6)}$	$-Z_4^{(6)}$	$Z_7^{(6)-1}$	$Z_6^{(6)-1}$	$-Z_8^{(6)}$
5	$-Z_1^{(5)}$	$Z_3^{(5)-1}$	$Z_2^{(5)-1}$	$-Z_5^{(5)}$	$-Z_4^{(5)}$	$Z_7^{(5)-1}$	$Z_6^{(5)-1}$	$-Z_8^{(5)}$
6	$-Z_1^{(4)}$	$Z_3^{(4)-1}$	$Z_2^{(4)-1}$	$-Z_5^{(4)}$	$-Z_4^{(4)}$	$Z_7^{(4)-1}$	$Z_6^{(4)-1}$	$-Z_8^{(4)}$
7	$-Z_1^{(3)}$	$Z_3^{(3)-1}$	$Z_2^{(3)-1}$	$-Z_5^{(3)}$	$-Z_4^{(3)}$	$Z_7^{(3)-1}$	$Z_6^{(3)-1}$	$-Z_8^{(3)}$
8	$-Z_1^{(2)}$	$Z_3^{(2)-1}$	$Z_2^{(2)-1}$	$-Z_5^{(2)}$	$-Z_4^{(2)}$	$Z_7^{(2)-1}$	$Z_6^{(2)-1}$	$-Z_8^{(2)}$
输出变换	$-Z_1^{(1)}$	$Z_2^{(1)-1}$	$Z_3^{(1)-1}$	$-Z_4^{(1)}$	$-Z_5^{(1)}$	$Z_6^{(1)-1}$	$Z_7^{(1)-1}$	$-Z_8^{(1)}$

注：$Z_i^{(j)-1}$ 表示 $Z_i^{(j)}$ 的乘法逆元；$-Z_i^{(j)}$ 表示 $Z_i^{(j)}$ 的加法逆元

3.3 改进算法的加密和解密过程

改进算法的加密过程如图 2 所示，输入明文和加密子密钥，经过 8 轮迭代和 1 轮输出变换后得到密文。改进算法的解密过程与加密过程相同，只是输入部分变成密文和解密子密钥，输出解密后的明文。

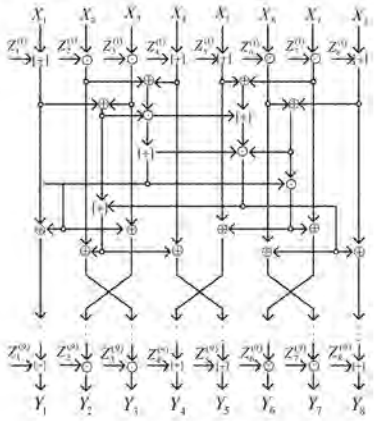

图 2　改进算法的结构

3.4 改进算法的符号说明

为了便于对分析过程进行描述，在加密的中间过程加入相应的符号。一轮迭代过程如下：

(1)$A=X_1[+]Z_1^{(1)}$　(2)$B=X_2\odot Z_2^{(1)}$

(3)$C=X_3\odot Z_3^{(1)}$　(4)$D=X_4[+]Z_4^{(1)}$

(5)$E=X_5[+]Z_5^{(1)}$　(6)$F=X_6\odot Z_6^{(1)}$

(7)$G=X_7\odot Z_7^{(1)}$　(8)$H=X_8[+]Z_8^{(1)}$

(9)$I=A\oplus C$　(10)$J=B\oplus D$

(11)$K=E\oplus G$　(12)$L=F\oplus H$

(13)$M=I\odot J$　(14)$N=K[+]M$

(15)$\Gamma=L\odot N$　(16)$P=M[+]\Gamma$

(17)$\Phi=I[+]\Gamma$　(18)$\Omega=L\odot P$

(19)$Q=A\oplus P$　(20)$R=C\oplus P$

(21)$S=E\oplus\Omega$　(22)$T=G\oplus\Omega$

(23)$U=B\oplus\Phi$　(24)$V=D\oplus\Phi$

(25)$W=F\oplus\Gamma$　(26)$\Lambda=H\oplus\Gamma$

输出变换去掉了最后一轮的交换影响。输出变换过程如下：

(1)$Y_1=Q[+]Z_1^{(9)}$　(2)$Y_2=U\odot Z_2^{(9)}$

(3)$Y_3=R\odot Z_3^{(9)}$　(4)$Y_4=V[+]Z_4^{(9)}$

(5)$Y_5=S[+]Z_5^{(9)}$ (6)$Y_6=W\odot Z_6^{(9)}$

(7)$Y_7=T\odot Z_7^{(9)}$ (8)$Y_8=\Delta[+]Z_8^{(9)}$

4 马尔科夫密码的安全性

马尔科夫密码首先由来学嘉教授提出[7,10]。文献[7]详细介绍了马尔科夫密码及其安全性的证明。马尔科夫密码在面对差分攻击时有固定的证明方法,通过该方法我们能够知道该马尔科夫密码是否存在一个达到安全的最低轮数。本节主要介绍马尔科夫密码的概念以及如何证明其安全性,不涉及这种证明方法的正确性,因为在文献[7]中已有详细的证明过程。

4.1 马尔科夫密码

在介绍马尔科夫密码之前,需给出一些相关的概念和符号说明。首先,本文中的安全性指的是算法面对差分攻击时的安全性。其次,马尔科夫密码是针对迭代型密码[12-13]定义的(例如 DES,IDEA 和 REESSE3+都是迭代型密码)。在 REESSE3+算法中,定义运算符号$\otimes$,$X\otimes X^*=(X_1\odot X_1^*,X_2[+]X_2^*,X_3[+]X_3^*,X_4\odot X_4^*,X_5[+]X_5^*,X_6\odot X_6^*,X_7\odot X_7^*,X_8[+]X_8^*)$;$X$ 的逆元为 $X^{-1}=(X_1^{-1},-X_2,-X_3,X_4^{-1},-X_5,X_6^{-1},X_7^{-1},-X_8)$,其中,$X_1^{-1}$ 为 X_1 的乘法逆元,$-X_2$ 为 X_2 的加法逆元;中性元为 $e=(X_1=1,X_2=0,X_3=0,X_4=1,X_5=0,X_6=1,X_7=1,X_8=0)$,简记为 $e=(1,0,0,1,0,1,1,0)$或者 $e=10010110$;定义差分为 $\Delta X=X\otimes X^{*-1}$,第 1 轮输入的差分定义为 $\Delta Y(0)=\Delta X$,第 1 轮输出的差分定义为 $\Delta Y(1)$,相应的第 i 轮输出的差分定义为 $\Delta Y(i)$。同样地,在改进算法中定义运算符号$\otimes$,$X\otimes X^*=(X_1[+]X_1^*,X_2\odot X_2^*,X_3\odot X_3^*,X_4[+]X_4^*,X_5[+]X_5^*,X_6\odot X_6^*,X_7\odot X_7^*,X_8[+]X_8^*)$;$X$ 的逆元为 $X^{-1}=(-X_1,X_2^{-1},X_3^{-1},-X_4,-X_5,X_6^{-1},X_7^{-1},-X_8)$,其中,$-X_1$ 为 X_1 的加法逆元,X_2^{-1} 为 X_2 的乘法逆元;中性元为 $e=(X_1=0,X_2=1,X_3=1,X_4=0,X_5=0,X_6=1,X_7=1,X_8=0)$,简记为 $e=(0,1,1,0,0,1,1,0)$或者 $e=01100110$;定义差分为 $\Delta X=X\otimes X^{*-1}$,第 1 轮输入的差分定义为 $\Delta Y(0)=\Delta X$,第 1 轮输出的差分定义为 $\Delta Y(1)$,相应的第 i 轮输出的差分定义为 $\Delta Y(i)$。需要说明的是,在这两个算法中 X^* 是一个与 X 不同的明文,因为当 $X=X^*$ 时,进行差分攻击没有作用[7]。

在 IDEA 算法中,作者假设子密钥是独立且均匀随机分布的[7]。可以这样理解该假设条件:若是在这样的条件下,算法都不能抵抗差分攻击,那么这个算法在子密钥不是独立且均匀随机分布时一定是不安全的。我们也假设在REESSE3+算法和改进算法中,子密钥是独立且均匀随机分布的。接下来给出马尔科夫密码的相关定义。

定义 1[7] 一个 i 轮差分是一个(α,β)元组,其中 α 是一对不同的明文 X 和 X^* 的差分,β 是相应的 i 轮输出的结果 $Y(i)$和 $Y(i)^*$ 的差分。i 轮差分概率是一个条件概率,假设每轮加密使用的子密钥 $Z^{(1)},\cdots,Z^{(r)}$ 是独立且均匀随机分布的,我们用 $P(\Delta Y(i)=\beta|\Delta X=\alpha)$表示这个概率。

定义 2[7] 拥有轮函数 f 的迭代型密码是马尔科夫密码,当存在群操作$\otimes$,使得对于所有的差分 $\alpha(\alpha\neq e)$和差分 $\beta(\beta\neq e)$,$P(\Delta Y=\beta|\Delta X=\alpha,X=\gamma)$独立于明文 γ,其中,$Y=f(X,Z)$,$Y^*=f(X^*,Z)$,且子密钥 Z 是均匀随机分布的;即当子密钥均匀随机分布时,对于所有的 γ,$P(\Delta Y=\beta|\Delta X=\alpha,X=\gamma)=P(\Delta Y=\beta|\Delta X=\alpha)$都成立。

4.2 马尔科夫密码的差分攻击

本节介绍马尔科夫密码差分攻击用到的引理和性质,相关的证明可以参考文献[7]。

引理 1[7] 如果 r 轮迭代密码是一个马尔科夫密码,且 r 轮子密钥独立且均匀随机分布,那么差分序列 $\Delta X=Y(0),\Delta Y(1),\cdots,\Delta Y(r)$是一个齐次马尔科夫链。

差分序列 $\Delta X=Y(0),\Delta Y(1),\cdots,\Delta Y(r)$是一个马尔科夫链,意味着 $P(\Delta Y(i+1)=\beta_{i+1}\mid\Delta Y(i)=\beta_i,\Delta Y(i-1)=\beta_{i-1},\cdots,\Delta Y(1)=\beta_1,\Delta Y(0)=\beta_0)=P(\Delta Y(i+1)=\beta_{i+1}\mid\Delta Y(i)=\beta_i)$,其中 $0\leqslant i<r$,也就是说它只与上一轮的差分有关,与之前的其他输入无关。而齐次马尔科夫链意味着$P(\Delta Y(i+1)=\beta|\Delta Y(i)=\alpha)$与 i 的具体取值无关。要注意,我们已经假设子密钥是独立且均匀随机分布的,因此一个马尔科夫密码对应一个齐次马尔科夫链。

定义 3[7] 在马尔科夫密码中,一个齐次马尔科夫链 $\Delta X=\Delta Y(0),\Delta Y(1),\cdots,\Delta Y(r)$所对应的概率转移矩阵用 Π 表示,在 Π 中第 i 行、第 j 列的元素用$P(\Delta Y(1)=\alpha_j\mid\Delta X=\alpha_i)$表示,其中 $\alpha_1,\alpha_2,\cdots,\alpha_M$ 是 ΔX 的 M 种可能,对于 m 位的密码来说,$M=2^m-1(\Delta X\neq e)$,因此对于所有的 $r\geqslant 1$,有:

$$\Pi^r=[p_{ij}^{(r)}]=[P(\Delta Y(r)=\alpha_j\mid\Delta X=\alpha_i)] \tag{1}$$

式(1)给出了计算 $P(\Delta Y(r)=\alpha_j\mid\Delta X=\alpha_i)$的方法,我们只需要计算齐次马尔科夫链所对应的概率转移矩阵的 r 次幂即可。由于一个马尔科夫密码对应一个齐次马尔科夫链,这意味着一个马尔科夫密码对应一个概率转移矩阵。

随机相等假设[7]:对于几乎所有的最大可能的$(r-1)$轮差分(α,β)的概率,$P(\Delta Y(r-1)=\beta|\Delta X=\alpha)\approx P(\Delta Y(r-1)=\beta|\Delta X=\alpha,Z^{(1)}=z_1,\cdots,Z^{(r-1)}=z_{r-1})$对于大部分子密钥$(z_1,z_2,\cdots,z_{r-1})$成立。

在 IDEA 算法中作者假设随机相等假设成立,因此在本文中我们也假设随机相等假设在 REESSE3+算法和改进算法中成立。

引理 2[7] 假设随机相等假设是正确的,那么在对一个 r 轮迭代型密码进行差分攻击时,

$$C_d(r)\geqslant 2/(P_{\max}^{(r-1)}-\frac{1}{2^m-1}) \tag{2}$$

其中,$P_{\max}^{(r-1)}=\max_{\alpha}\max_{\beta}P(\Delta Y(r-1)=\beta|\Delta X=\alpha)$,且 m 是分组密码的分组长度,$C_d(r)$表示需要加密的明文对数。

引理 2 给出了一个 r 轮迭代密码何时能够抵抗差分攻击,特别地,当 $P_{\max}^{(r-1)}\leqslant 3/2^m$ 时,则 $C_d(r)\geqslant 2^{m-1}$,这意味着至少需要 2^m 次加密[7],也就是说此时该密码是能够抵抗差分攻击的。

如下性质可以在文献[7]中找到，也可以在与有限马尔科夫链[14-15]和非负矩阵[16]相关的参考文献中找到。

性质 1[7]　一个拥有概率转移矩阵 $\Pi=[p_{ij}]$ 的有限马尔科夫链有一个稳定的(steady-state)分布，当且仅当存在一个 r，使得 Π^r 中没有 0 元素。这种情况下，这样的矩阵是本原的(primitive)。矩阵 Π 是本原的还有一个等价条件，即当且仅当存在一个 r_0，使得 Π^{r_0} 中有一列不含 0 元素。

引理 3[7]　对于一个拥有分组长度为 m，并拥有独立且均匀随机分布子密钥的马尔科夫密码，如果半无限的(semi-infinite)马尔科夫链 $\Delta X=\Delta Y(0),\Delta Y(1),\cdots,\Delta Y(r)$ 拥有一个稳定的(steady-state)分布，那么这个稳定的分布一定是均匀的分布，也就是说：

$$\lim_{r\to\infty}P(\Delta Y(r)=\beta|\Delta X=\alpha)=\frac{1}{2^m-1}$$

对于所有的非中性元差分(α,β)都成立。如果我们假设随机相等假设对这个马尔科夫密码成立，那么这个密码在足够的轮数之后就能够抵抗差分攻击。

性质 1、引理 2 和引理 3 说明：对于一个马尔科夫密码，如果随机相等假设成立，那么在这个马尔科夫密码所对应的概率转移矩阵中，若能找到 Π^{r_0} 有一列不含 0 元素，则这个密码在足够的轮数之后就一定能够抵抗差分攻击。

5　REESSE3+(16)算法和 16 位输入的改进算法的安全性

本节介绍了 REESSE3+(16)算法和 16 位输入的改进算法的概念及其安全性证明的方法。REESSE3+(16)算法和相应的 16 位输入的改进算法的定义是根据 IDEA(m)算法提出的。来学嘉教授[7]提出了 IDEA(m)算法及其安全性证明[17-19]的方法。本节参考 IDEA(m)算法的证明过程来证明 REESSE3+(16)算法和 16 位输入的改进算法的安全性。

5.1　REESSE3+(16)算法和 16 位输入的改进算法的概率转移矩阵

在 REESSE3+算法中，分组长度为 $m=128$ 位，每个子块位数为 $n=16$，共 8 个子块，即 $16*8=128$ 位。而在 REESSE3+(m)算法中，算法的结构图与 REESSE3+算法的结构图是一样的，不同的是其分组长度由 128 位变成了 m 位，m 可以取的值为 16、32、64 和 128。当 m 分别取 16、32、64 和 128 时，每个子块长度 n 分别为 2、4、8 和 16。在REESSE3+(m)算法中，$a\oplus b$ 表示两个 n 位的子块 a 和 b 按位异或，$a[+]b$表示两个 n 位子块 a 和 b 相加后模 2^n，$a\odot b$ 表示两个 n 位子块相乘后模(2^n+1)，其中在乘法模中 0 表示 2^n。来学嘉教授证明[7]，当 $n=2,4,8$ 和 16 时，3 种群运算是不相容的[20]。同理，16 位输入的改进算法的定义与 REESSE3+(16)的定义是一样的，都是将输入位数变成 16，即 m 取值 16。

算法 1　REESSE3+(16)算法的概率转移矩阵生成算法

Step1　设置子密钥：$Z^{(1)}=e=10010110$；

Step2　对于每个 $\Delta X(\Delta X\neq e)$，遍历 X，计算相应的 X^* $(\Delta X=X\otimes X^{*-1})$；

Step3　在密钥 $Z^{(1)}=e=10010110$ 的情况下，计算一轮加密后相应的 Y 和 Y^*；

Step4　计算 $\Delta Y=Y\otimes Y^{*-1}$，将转移矩阵中第 ΔX 行、第 ΔY 列对应的元素加 1；

Step5　遍历完 ΔX 后，将矩阵中的每个元素除以 2^{16} 来得到概率转移矩阵。

算法 2　16 位输入的改进算法的概率转移矩阵生成算法

Step1　设置子密钥：$Z^{(1)}=e=01100110$；

Step2　对于每个 $\Delta X(\Delta X\neq e)$，遍历 X，计算相应的 X^* $(\Delta X=X\otimes X^{*-1})$；

Step3　在密钥 $Z^{(1)}=e=01100110$ 的情况下，计算一轮加密后相应的 Y 和 Y^*；

Step4　计算 $\Delta Y=Y\otimes Y^{*-1}$，将转移矩阵中第 ΔX 行、第 ΔY 列对应的元素加 1；

Step5　遍历完 ΔX 后，将矩阵中的每个元素除以 2^{16} 来得到概率转移矩阵。

对于这两个算法要特别注意数值转换，如：在算法 1 中，$e=10010110$转换成 16 比特二进制时为：0100000100010100，e 对应的十进制为 16660，也就是说 ΔX 要遍历 $0\sim2^{16}-1$(除了 16660 外)。矩阵中第一行对应$\Delta X=0$，第 16660 行对应 $\Delta X=16659$，第 16661 行对应 $\Delta X=16661$，第 $2^{16}-1$ 行对应 $\Delta X=2^{16}-1$。矩阵的列同样如此。得到的矩阵中共有$(2^{16}-1)*(2^{16}-1)$个元素。每个 ΔX 对应 2^{16}对(X,X^*)。

定理 1　算法 1 和算法 2 生成的矩阵与子密钥取值无关。

证明：假设 ΔX 为一个固定值，因为 $\Delta X=X\otimes X^{*-1}$，所以 $X^*=X\otimes\Delta X^{-1}$，且假设子密钥为 $Z^{(1)}$。定义 $Temp=X\otimes Z^{(1)}$，$Temp^*=X^*\otimes Z^{(1)}$(差分攻击中对不同明文加密要用同一个密钥)。由于 X 是 $0\sim2^{16}-1$ 的一个遍历，因此不论 $Z^{(1)}$ 的取值为多少，$Temp$ 都是 $0\sim2^{16}-1$ 的一个遍历。因为 $Temp\otimes Temp^{*-1}=(X\otimes Z^{(1)})\otimes(X^*\otimes Z^{(1)})^{-1}=X\otimes X^{*-1}=\Delta X$，所以 $Temp^*=Temp\otimes\Delta X^{-1}$。由于不论 $Z^{(1)}$ 的取值是多少，$Temp$ 都是 $0\sim2^{16}-1$ 的一个遍历，同时由于 $Temp^*=Temp\otimes\Delta X^{-1}$，因此算法 1 和算法 2 生成的矩阵与子密钥取值无关。注意，在算法 1 中 $X\otimes X^*=(X_1\odot X_1^*,X_2[+]X_2^*,X_3[+]X_3^*,X_4\odot X_4^*,X_5[+]X_5^*,X_6\odot X_6^*,X_7\odot X_7^*,X_8[+]X_8^*)$，$X^{-1}=(X_1^{-1},-X_2,-X_3,X_4^{-1},-X_5,X_6^{-1},X_7^{-1},-X_8)$；在算法 2 中 $X\otimes X^*=(X_1[+]X_1^*,X_2\odot X_2^*,X_3\odot X_3^*,X_4[+]X_4^*,X_5[+]X_5^*,X_6\odot X_6^*,X_7\odot X_7^*,X_8[+]X_8^*)$，$X^{-1}=(-X_1,X_2^{-1},X_3^{-1},-X_4,-X_5,X_6^{-1},X_7^{-1},-X_8)$。虽然符号定义有所不同，但是并不影响证明过程。

算法 1、算法 2 和定理 1 显然都可以扩展到 $m=32,64$ 和 128。同时，算法 1 和算法 2 显然也可以等价地转换成固定明文 X，遍历子密钥 $Z^{(1)}$，因为从 $Temp=X\otimes Z^{(1)}$ 来看，固定任何一个遍历另一个，结果都是相同的，而且本文已经假设子密钥独立且均匀随机分布，所以固定明文 X，遍历子密钥 $Z^{(1)}$ 更加合理。之所以不将算法 1 和算法 2 改写成固定明文 X，遍历子密钥 $Z^{(1)}$，是因为算法 1 和算法 2 更容易通过代码实现。

考虑到固定明文 X，遍历子密钥 $Z^{(1)}$ 的情况，可以得到

$P(\Delta Y(1)=\beta|\Delta X=\alpha,X=\gamma)=P(\Delta Y(1)=\beta|\Delta X=\alpha)$。我们已经假设子密钥是独立且均匀随机分布的，因此由定义 2 可以得到如下定理。

定理 2 对于 $m=16,32,64$ 和 128，在差分定义为 $\Delta X=X\otimes X^{*-1}$（其中$\otimes$在 3.1 节中定义，REESSE3+算法和改进算法中的$\otimes$定义不同）时，REESSE3+(m)是马尔科夫密码，m 位输入的改进算法也是马尔科夫密码。

在图 3 中，用 $In(\cdot)$表示退化算法[7]（需要特别注意的是，该退化算法的定义与文献[7]中的有所区别），即对于任意的 S 有 $S=In(In(S))$。用 $P_I(\cdot)$表示退化转换[7]，即对于任意的 T 有 $T=P_I(P_I(T))$。退化算法和退化转换的区别是：退化转换只是简单的交换操作，而退化算法则要经过一系列的变换。在 REESSE3+(m)算法中，$P_I(\cdot)$将子块 2 和子块 3 交换，子块 6 和子块 7 交换；在改进算法中，$P_I(\cdot)$将子块 2 和子块 3 交换，子块 4 和子块 5 交换，子块 6 和子块 7 交换。

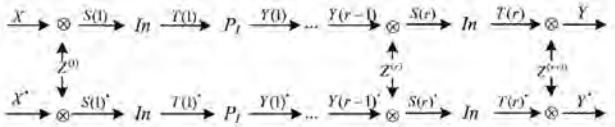

图 3 REESSE3+(m)算法和改进算法加密一对明文的 r 轮流程

定理 3 REESSE3+(m)算法和 m 位输入的改进算法具有性质：式(3)对于所有的 $i(1\leqslant i\leqslant r-1)$都成立。

$$P(\Delta Y(i)=\beta|\Delta X=\alpha)=P(\Delta Y(i)=P_I(\alpha)|\Delta X=P_I(\beta)) \tag{3}$$

证明：从图 3 可得 $\Delta X=\Delta S(1)$；因为 $P_I(\cdot)$是简单的交换操作，而且子块 2 和子块 3、子块 6 和子块 7 的逆元运算分别相同，所以 $\Delta Y(1)=P_I(T(1))\otimes(P_I(T(1)^*))^{-1}=P_I(T(1)\otimes(T(1)^*)^{-1})=P_I(\Delta T(1))$；因为 $In(\cdot)$为退化算法，所以当输入为一对(S,S^*)、输出为一对(T,T^*)时，若输入为一对(T,T^*)，则输出一定为一对(S,S^*)。因此得到 $P(\Delta T=\beta|\Delta S=\alpha)=P(\Delta T=\alpha|\Delta S=\beta)$，所以：

$$\begin{aligned}&P(\Delta Y(1)=P_I(\alpha)|\Delta X=P_I(\beta))\\&=P(\Delta T(1)=\alpha|\Delta S(1)=P_I(\beta))\\&=P(\Delta T(1)=P_I(\beta)|\Delta S(1)=\alpha)\\&=P(\Delta Y(1)=\beta|\Delta X=\alpha)\end{aligned}$$

由于加密过程除了最后一轮每轮都是相同的，因此由定义 3 的式(1)可知，式(3)对于 $i(1\leqslant i\leqslant r-1)$成立。

在文献[7]中也有与定理 3 类似的定理，但是由于二者对应的退化算法有所不同，此处的退化算法此文献[7]中的退化算法要简单很多，因此本文借鉴了文献[7]中的证明过程，具体可以参考文献[7]。

引理 4[7] 对于任意满足式(3)的马尔科夫密码，若其对应的概率转移矩阵的 $r(r$ 可以取 $1,2,\cdots)$次幂中有一行不含 0 元素，则必有一列不含 0 元素。

推论 1[7] 引理 2－引理 4 和性质 1 共同说明：对于一个马尔科夫密码，若是随机相等假设成立，且这个马尔科夫密码所对应的概率转移矩阵的 $r(r$ 可以取 $1,2,\cdots)$次幂中有一行不含 0 元素，则这个密码在足够的迭代轮数之后一定能够抵抗差分攻击。

5.2 REESSE3+(16)算法和 16 位输入的改进算法的实验结论

由于 REESSE3+(16)算法和 16 位输入的改进算法只有 16 位输入，因此共有 $2^{16}-1$ 个 $\Delta X(\Delta X\neq e)$，每个 ΔX 对应 2^{16}对(X,X^*)，所以需要一轮加密的次数大约是 2^{32}次，在微型计算机的计算能力之内（这也是我们不计算更多位数的原因）。采用算法 1 和算法 2 分别得到 REESSE3+(16)算法和 16 位输入的改进算法的概率转移矩阵，然后分别计算概率转移矩阵的 $r(r$ 取 $2,3,\cdots)$次幂。

在 REESSE3+(16)所对应的概率转移矩阵的 3 次幂中，找到不含 0 元素的行，如第 43 行、47 行、59 行等，由推论 1 得到：REESSE3+(16)算法在足够的轮数之后一定能够抵抗差分攻击。

在 16 位输入的改进算法所对应的概率转移矩阵的 3 次幂中，也找到了不含 0 元素的行，如第 7 行、23 行、25 行等，由推论 1 得到：16 位输入的改进算法在足够的轮数之后一定能够抵抗差分攻击。

为了计算出确定的轮数，分别继续计算矩阵的 4 次幂、5 次幂、…，并找出每次幂中最大的概率值，在REESSE3+(16)所对应的概率转移矩阵的前 15 次幂中，每次幂中的最大的概率如表 2 所列。同样地，在 16 位输入的改进算法所对应的概率转移矩阵的前 7 次幂中，每次幂中最大的概率如表 3 所列。在两个表中矩阵的最大概率 $P_{\max}(\Delta Y(r)=\beta|\Delta X=\alpha)$出现不止一次，因此没有写出具体的$(\alpha,\beta)$。关于表 2 需要说明的是，表中缺少的矩阵的 8,9,10 和 11 次幂我们并没有计算，因为从 1～7 次幂中最大的概率总结出每次幂的最大值大约为前一次的一半，所以我们需要的概率在 14 次幂左右。

表 2 REESSE3+(16)所对应的概率矩阵的 r 次幂中最大的概率

r	$P_{\max}(\Delta Y=\beta\|\Delta X=\alpha)$
1	0.25
2	0.127044677734375
3	0.06461310386657715
4	0.03287025587633252
5	0.01673254927254675
6	0.00852087601631398
7	0.004344402646308193
12	1.6343392594562016E−4
13	9.065603467447555E−5
14	5.367425812604281E−5
15	3.483878560829191E−5

表 3 16 位输入改进算法所对应的概率矩阵的 r 次幂中最大的概率

r	$P_{\max}(\Delta Y=\beta\|\Delta X=\alpha)$
1	0.25
2	0.125
3	0.06256318092346191
4	0.00199487115605734 3
5	2.651491796825667E−4
6	7.873327137675190E−5
7	2.497408028320769E−5

由引理 2 可知，当 $P_{\max}^{(r-1)}\leqslant 3/2^m$ 时，算法是能够抵抗差分攻击的，而 $3/2^{16}\approx 4.57763671875\text{E}-5\geqslant P_{\max}(\Delta Y(15)=\beta|$

$\Delta X=\alpha)\approx 3.483878560829191E-5$，因此 REESSE3＋(16)算法在经过 16 轮迭代后就能抵抗差分攻击；同样地，$3/2^{16}\approx 4.57763671875E-5\geqslant P_{max}(\Delta Y(7)=\beta\mid\Delta X=\alpha)=2.497408028320769E-5$，因此 16 位输入的改进算法在经过 8 轮迭代后就能够抵抗差分攻击。

由以上分析可知，16 位输入的改进算法在经过 8 轮迭代后就能够抵抗差分攻击；而 REESSE3＋(16)算法要经过 16 轮迭代才能抵抗差分攻击。因此，在 16 位输入时，改进算法比 REESSE3＋算法更能抵抗差分攻击。

结束语　本文采用了马尔科夫密码模型，通过实验对比直接证明了：16 位输入的改进算法达到安全性要求所需要的轮数比原 REESSE3＋(16)算法所需轮数更少。由此可见，16 位输入时，改进算法比 REESSE3＋算法更能抵抗差分攻击。至于在输入位数为 128 时，改进算法是否更加安全，还有待进一步论证。

参 考 文 献

[1] SU S H, LÜ S W, DONG D Q. A 128-bit Block Cipher Based on Three Group Arithmetics[DB/OL]. http://eprint.iacr.org/2014/704.pdf.

[2] SU S H. The REESSE Symmetric Key Cryptosystem[J]. Computer Engineering and Applications, 2004, 40(19): 84-86. (in Chinese)
苏盛辉. REESSE 对称密钥密码体制[J]. 计算机工程与应用, 2004, 40(19): 84-86.

[3] BIHAM E, SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[J]. Journal of Cryptology, 1991, 4(1): 3-72.

[4] BIHAM E, SHAMIR A. Differential Cryptanalysis of the Full 16-Round DES[C]// International Cryptology Conference on Advances in Cryptology. Springer-Verlag, 1992: 487-496.

[5] HEYS H M. A Tutorial on Linear and Differential Cryptanalysis [J]. Cryptologia, 2001, 26(3): 189-221.

[6] BIHAM E, SHAMIR A. Differential Cryptanalysis of the Data Encryption Standard[M]. Springer-Verlag, 1993.

[7] LAI X J. On the design and security of block ciphers[C]// Series in Information Processing. 1992.

[8] LAI X J. International Data Encryption Algorithm[J]. Hepatology, 2007, 60(6): 2125-2126.

[9] STINSON D R. Cryptography: Theory and Practice[M]. CRC Press, 1995.

[10] LAI X J, MASSEY J L, MURPHY S. Markov ciphers and differential cryptanalysis[C]// International Conference on Theory and Application of Cryptographic Techniques. Springer-Verlag, 1991: 17-38.

[11] O'CONNOR L, GOLIC J D. A Unified Markow Approach to Differential and Linear Cryptanalysis[C]// International Conference on the Theory and Applications of Cryptology: Advances in Cryptology. Springer-Verlag, 1994: 387-397.

[12] KATZ J, LINDE Y. Introduction to modern cryptography[OL]. http://www.pdfsdocuments.com/introductions-to-modern-crypotography-principles-and-protocols.pdf.

[13] SCHNEIER B, SUTHERLAND P. Applied Cryptography: Protocols, Algorithms, and Source Code in C[M]. John Wiley & Sons, 2015.

[14] BURGIN M. Theory of information: fundamentality, diversity and unification[M]. World Scientific, 2010.

[15] KEILSON J. Markov Chain Models — Rarity and Exponentiality[M]. Springer New York, 1979.

[16] MINC H. Nonnegative matrices[M]. John Wiley & Sons, 1988: 318-330.

[17] HARPES C, HARPES C. Cryptanalysis of iterated block ciphers [C]// ETH Series in Information Processing. Hartung-Gorre, 1996.

[18] BIHAM E, DUNKELMAN O, KELLER N. New cryptanalytic results on IDEA[C]// International Conference on Theory and Application of Cryptology and Information Security. Springer-Verlag, 2006: 412-427.

[19] BIRYUKOV A, NAKAHARA J, YILDIRIM H M. Differential entropy analysis of the IDEA block cipher[J]. Journal of Computational & Applied Mathematics, 2014, 259(6): 561-570.

[20] LAI X J, MASSEY J L. A Proposal for a New Block Encryption Standard[M]// Advances in Cryptology — EUROCRYPT '90. Springer Berlin Heidelberg, 1999: 389-404.

(上接第 113 页)

[11] JEBBAOUI H, MOURAD A, OTROK H, et al. Semantics-based approach for detecting flaws, conflicts and redundancies in XACML policies[J]. Computers & Electrical Engineering, 2015, 44(C): 91-103.

[12] MOURAD A, TOUT H, TAHLI C, et al. From model-driven specification to design-level set-based analysis of XACML policies[J]. Computers & Electrical Engineering, 2016, 52(C): 65-79.

[13] WANG Y Z, FENG D G. A Conflict and Redundancy Analysis Method for XACML Rules[J]. Journal of Computers, 2009, 32(3): 516-530. (in Chinese)
王雅哲，冯登国. 一种 XACML 规则冲突及冗余分析方法[J]. 计算机学报, 2009, 32(3): 516-530.

[14] CHEN W H, WANG N N. Research on XACML policy evaluation optimization technology[J]. Application Research of Computers, 2013, 30(3): 900-905. (in Chinese)
陈伟鹤，王娜娜. 基于 XACML 的策略评估优化技术的研究[J]. 计算机应用研究, 2013, 30(3): 900-905.

[15] QI Y, CHEN J, LI Q M. XACML policy evaluationoptimization method based on recording[J]. Journal of Nanjing University of Science and Technology, 2015, 39(2): 187-193. (in Chinese)
戚湧，陈俊，李千目. 一种基于重排序的 XACML 策略评估优化方法[J]. 南京理工大学学报, 2015, 39(2): 187-193.

[16] MAROUF S, SHEHAB M, SQUICCIARINI A, et al. Adaptive Reordering and Clustering-Based Framework for Efficient XACML Policy Evaluation[J]. IEEE Transactions on Services Computing, 2012, 4(4): 300-313.

[17] XACML 2.0 conformance test[OL]. http://www.oasis-open.org/committees/download.php/14846/xacml2.0-ct-v.0.4.zip.