

基于 AES 轮函数认证加密算法研究与设计

高国强, 李子臣

(北京印刷学院数字版权保护技术研究中心, 北京 102600)

摘要: 认证加密算法同时保证信息的机密性和完整性, 在信息安全领域具有广泛的应用前景。利用混合整数线性规划方法, 搜索高效且最小活跃 S 盒较多的迭代结构, 基于 AES 轮函数和广义 Feistel 结构设计底层的轮函数, 实现了一个基于 AES 轮函数的认证加密算法。该认证加密算法具有抵抗碰撞攻击、差分攻击、线性攻击等攻击的能力, 且效率是原有认证加密算法 AES-GCM 的两倍。

关键词: 认证加密算法; AES 算法; 混合整数线性规划

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020028

Research and design of authenticated encryption algorithm based on AES round function

GAO Guoqiang, LI Zichen

Research Center for digital copyright protection technology, Beijing Institute of Graphic Communication, Beijing 102600, China

Abstract: The authenticated encryption algorithm guarantees the confidentiality and integrity of the information at the same time, and has extensive research and application prospects in the field of information security. With the mixed integer linear programming (MILP) method, the iterative structure with high efficiency and more active S boxes is searched. Based on this new round iterative function and the generalized Feistel structure, an authenticated encryption algorithm with the ability to resist collision attack, differential attack, linear attack and other attacks was designed and implemented, and the efficiency was twice that of AES-GCM.

Key words: authenticated encryption algorithm, AES algorithm, MILP

收稿日期: 2019-11-15; 修回日期: 2020-02-03

通信作者: 李子臣, lizc2020@163.com

基金项目: 国家自然科学基金资助项目 (No.61370188); 北京市教委科研计划一般基金资助项目 (No.KM201610015002, No.KM201510015009); 北京市教委科研计划重点基金资助项目 (No.KZ201510015015, No.KZ201710015010); 科技创新服务能力建设-科研水平提高定额基金资助项目 (No.PXM2017_014223_000063); 北京印刷学院校级基金资助项目 (No.Ec201803, No.Ed201802, No. Ea201806)

Foundation Items: The National Natural Science Foundation of China (No.61370188), The Scientific Research Common Program of Beijing Municipal Commission of Education (No.KM201610015002, No.KM201510015009), The Beijing City Board of Education Science and Technology Key Project (No.KZ201510015015, No.KZ201710015010), Project of Beijing Municipal College Improvement Plan (No.PXM2017_014223_000063), BIGC Project (No.Ec201803, No.Ed201802, No.Ea201806)

论文引用格式: 高国强, 李子臣. 基于 AES 轮函数认证加密算法研究与设计[J]. 网络与信息安全学报, 2020, 6(2): 106-115.
GAO G Q, LI Z C. Research and design of authenticated encryption algorithm based on AES round function[J]. Chinese Journal of Network and Information Security, 2020, 6(2): 106-115.

1 引言

认证加密算法能够同时保证数据的机密性和完整性，在信息安全领域有着极其重要的作用。早期认证加密算法多采用加密和消息认证组合的方式，分为先加密后认证、先认证后加密、加密认证同时进行，认证加密一直是密码学科研究的热点^[1-2]。2000年，Bellare 和 Rogaway 等^[3-4]密码学者正式提出认证加密(AE, authenticated encryption)的概念，认证和加密在一个算法中同时完成，提高了效率。目前，认证加密算法的设计主要有两种模式：一种为基于分组密码的认证加密模式，另一种为直接设计的认证加密算法。分组密码认证加密模式是以 S 盒调用分组密码方式，具有方便替换底层分组加密算法的优势，具有代表性的认证加密模式有 OCB2.0^[5]、CCM^[6]、EAX^[7]、GCM^[8]等。直接设计的认证加密算法从密码部件开始，具有速度快、代价低，以及灵活性高等优点，其安全性的理论证明仍是研究热点。其主要设计思想是借助已有的分组密码算法或加密标准设计的设计方法，并依靠各种安全性的分析方法^[9-13]进行安全性证明。

2019年2月完成的 CAESAR (competition for authenticated encryption: security, applicability, and robustness) 算法遴选，使认证加密算法的研究与设计进入了高潮。基于 AES 加密标准设计了如 AEGIS、TiaoXin、ALE 等诸多认证加密算法^[14-17]。

本文以抵抗碰撞攻击为目标，基于 AES 轮函数的多轮组合设计一种认证加密算法的通用轮函数结构 $R(t, s, l)$ ，其中 t 表示执行 AES 轮函数的次数， s 表示通用轮函数打入 128 bit 消息的块数， l 表示每个 AES 轮函数处理的 32 bit 消息的个数。利用混合整数线性规划 (MILP, mixed integer linear programming) 进行搜索实验，得到轮函数结构为 $R(4, 4, 1)$ 时满足抵抗碰撞攻击的安全目标和四轮传播定理，并且十轮最小活跃 S 盒数达到 80 个。利用上述安全结构直接设计一种新的认证加密算法 AMRAE (AES-based multi-round combination authen-

ticated encryption)，并进行了安全性分析和效率评估。实验结果表明，在并行优化下，AMRAE 认证加密算法的速度是原有 AES-GCM^[8]的两倍。

2 基础知识

2.1 AES 轮函数

本文使用的 AES 组长度为 128 bit。 X 是 4×4 的矩阵，具体如下。

$$X = \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}$$

其中， x_j 为 1 byte。用 X 表示 AES 加密算法的明文、中间状态矩阵和密文，用 K_i 表示 AES 算法第 i 轮的轮密钥。

AES 轮函数包含密钥加 (AddRoundKey)、字节替换 (SubBytes)、行移位 (ShiftRows)、列混合 (MixColumns) 等密码构件。

密钥加将轮密钥与状态矩阵进行异或， $\text{AddRoundKey}_{K_i}(X) = X \oplus K_i$ 。

字节替换是 AES 唯一的非线性变换，每个字节 x_{ij} 按照规则映射到另一个字节 $y_{ij} = S(x_{ij})$ 。

行移位将状态矩阵的第 i 行 $X(i)$ 循环左移 i 个字节，其中 $i = 0, 1, 2, 3$ 。

列混合用 $F_{2^8}^{4 \times 4}$ 中的矩阵左乘状态矩阵的每一列，元素之间的乘法运算定义在有限域 F_{2^8} 上。其目的是将状态矩阵每一列的元素进行混合，如式(1)所示， $a_{i,j}$ 和 $b_{i,j}$ 为状态矩阵的值。

$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix} \quad (1)$$

2.2 差分概念

差分密码分析是分组密码算法分析方法之一，也是认证加密算法安全性分析的主要方法。其思想是通过分析明文对的差分值对密文对的差分值的影响来恢复某些密钥比特。通常情况，差分分析是比较两个明文的异或与相应的两个

密文的异或。差分的定义如下。

定义 1 对分组长度为 n 的比特串 Y_i 和 Y_i^* 的差分定义为

$$\Delta Y_i = Y_i \oplus Y_i^* \quad (2)$$

其中, $\oplus$ 表示 n 比特串集上的一个异或运算。

2.3 混合整数线性规划

混合线性整数规划 MILP 用于求目标函数在一些线性约束条件下的最大或最小值。2011 年 Mouha 等^[11]将该方法应用于密码算法的安全性证明,通过计算加密算法活跃 S 盒的个数来验证其安全性。在分组密码、Hash 函数以及认证加密算法的设计方面得到了大量应用^[18-22]。

定义 2 设 Δ 为 n 字节的字符串 $\Delta = (\Delta_0, \dots, \Delta_{n-1})$, 称 $x = (x_0, \dots, x_{n-1})$ 为 Δ 的差分变量, 其中

$$x_i = \begin{cases} 0, & \Delta_i = 0 \\ 1, & \text{其他} \end{cases}, \quad i = 0, 1, \dots, n-1 \quad (3)$$

异或运算约束方程: 令输入差分变量为 $(x_{in1}^\oplus, x_{in2}^\oplus)$, 输出差分变量为 $x_{out}^\oplus$, 异或运算的差分分支数为 2, 需引入新的二元临时变量 $d^\oplus$ 。当且仅当变量 $x_{in1}^\oplus, x_{in2}^\oplus, x_{out}^\oplus$ 都为零时, $d^\oplus$ 为零。异或运算线性约束方程如下。

$$\begin{cases} x_{in1}^\oplus + x_{in2}^\oplus + x_{out}^\oplus \geq 2d^\oplus \\ 0 \geq v - d^\oplus, v \in \{x_{in1}^\oplus, x_{in2}^\oplus, x_{out}^\oplus\} \end{cases} \quad (4)$$

线性变换约束方程: 列混合变换 MixColumns 的输入差分变量为 $(x_{in1}^{MC}, x_{in2}^{MC}, x_{in3}^{MC}, x_{in4}^{MC})$, 输出差分变量为 $(x_{out1}^{MC}, x_{out2}^{MC}, x_{out3}^{MC}, x_{out4}^{MC})$ 。已知列混合的线性分支数和差分分支数都是 5^[22], 引入二元临时变量 d^{MC} , 当且仅当所有变量为零时, d^{MC} 为零; 否则为 1。线性变换约束方程如下。

$$\begin{cases} x_{in1}^{MC} + x_{in2}^{MC} + x_{in3}^{MC} + x_{in4}^{MC} + x_{out1}^{MC} + x_{out2}^{MC} + x_{out3}^{MC} + x_{out4}^{MC} \geq 5d^{MC} \\ 0 \geq v - d^{MC}, v \in \{x_{in1}^{MC}, x_{in2}^{MC}, x_{in3}^{MC}, x_{in4}^{MC}, x_{out1}^{MC}, x_{out2}^{MC}, x_{out3}^{MC}, x_{out4}^{MC}\} \end{cases} \quad (5)$$

目标函数: S 盒的输入差分变量为 x_i^s , 目标函数等于所有 S 盒的输入差分变量总和的最小值。假设算法中 S 盒数量为 n , 称 $\text{Min} \sum_{i=0}^{n-1} x_i^s$ 为目标函数。

在上述约束方程下, 利用优化模型的求解工具, 如 CPLEX、Gurobi 等^[11], 对目标函数求解。若只要求输入差分变量和所有 d 变量为二元变量, 则上述问题变为混合线性整数规划问题。利用 MILP 的求解方法, 可以提高上述目标函数的求解速度。

3 基于 AES 认证加密算法轮函数的设计

3.1 通用轮函数结构

利用 AES 的轮函数结合广义 Feistel 结构, 定义一种通用迭代型认证加密算法的轮函数结构 $R(t, s, l)$ ^[21], 其中, t 表示执行 AES 轮函数的次数 (状态大小), s 表示轮函数输入消息的块数 (规定从左到右依次打入消息块), l 表示每个 AES 轮函数处理的 32 bit 消息的个数, $l \in \{1, 2, 4\}$ (规定从矩阵左侧第一列依次计算)。 $R(t, s, l)$ 轮函数结构如图 1 所示, 其中, $M_i \in Z_2^{32 \times l}$ ($i = 1, 2, \dots, s$) 表示消息, P 是向量置换, 虚线表示消息的输入, 是一个可选项。其中 AddRoundMes 为消息 M_i 与状态矩阵按列依次进行异或运算。

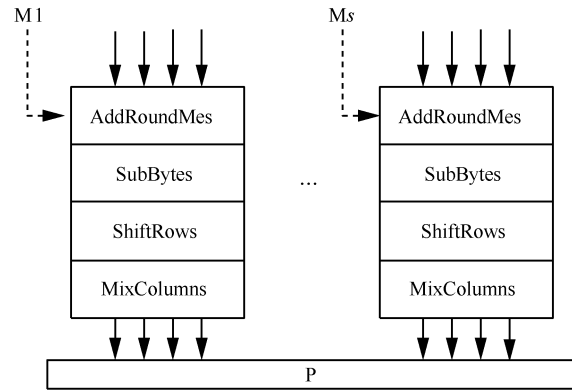


图 1 认证加密算法通用 $R(t, s, l)$ 轮函数结构

Figure 1 Authentication encryption algorithm general round function structure $R(t, s, l)$

比如 $l=1$ 时, AddRoundMes 进行如下的运算。

$$\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \oplus \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & m_1 \\ m_2 & m_2 & m_2 & m_2 \\ m_3 & m_3 & m_3 & m_3 \end{bmatrix} =$$

$$\begin{bmatrix} x_{16} & x_{20} & x_{24} & x_{28} \\ x_{17} & x_{21} & x_{25} & x_{29} \\ x_{18} & x_{22} & x_{26} & x_{30} \\ x_{19} & x_{23} & x_{27} & x_{31} \end{bmatrix} \quad (6)$$

比如 $l=2$ 时, AddRoundMes 进行如下的运算。

$$\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \oplus \begin{bmatrix} m_0 & m_4 & m_0 & m_4 \\ m_1 & m_5 & m_1 & m_5 \\ m_2 & m_6 & m_2 & m_6 \\ m_3 & m_7 & m_3 & m_7 \end{bmatrix} = \begin{bmatrix} x_{16} & x_{20} & x_{24} & x_{28} \\ x_{17} & x_{21} & x_{25} & x_{29} \\ x_{18} & x_{22} & x_{26} & x_{30} \\ x_{19} & x_{23} & x_{27} & x_{31} \end{bmatrix} \quad (7)$$

定义 3 $R(t,s,l)$ 效率: 称每一个 32 bit 消息块所需要的 AES 轮函数的个数为 $R(t,s,l)$ 的效率, 即 $rate = t / (s \times l)$ 。

基于结构 $R(t,s,l)$, 定义衡量其效率的指标 $rate$ 。若状态大小相同, 每轮处理的 32 bit 消息块越多, 即 $rate$ 越小, 所对应结构的效率越高。

3.2 安全性目标

所设计的 $R(t,s,l)$ 是一个通用结构, 区别于具体的算法, 因此只能分析其抵抗部分攻击的安全性。本文主要考虑该结构抵抗内部碰撞攻击的能力, 内部碰撞攻击是利用两次不同消息的差分值, 通过引入初始差分, 在随后的差分值中使内部发生碰撞, 即内部截断差分为零。该攻击是一种分析认证加密算法的常用分析方法。

AES 的四轮传播定理^[21]证明了四轮 AES 的差分或线性特征中活跃的 S 盒数至少为 25 个。但考虑密钥加运算后, 即在相关密钥差分分析下, AES 的四轮活跃 S 盒数降为 5。

本文要求在密钥加参与的情况下, 碰撞攻击

的差分传播过程中活跃 S 盒的个数不小于 25。以此为安全目标, 对结构 $R(t,s,l)$ 的参数 t 、 s 、 l 和置换 P 进行搜索。

3.3 MILP 搜索实验结果

本文的目标是在通用的 $R(t,s,l)$ 轮函数结构进行实验搜索, 得到一些安全且高效的轮函数结构, 为设计安全的认证加密算法奠定基础。

假设 P 是基于字 (32 bit, 即矩阵的一列) 的向量置换, 提前给定参数 t 、 s 、 l 和置换 P , 利用 MILP 方法验证其四轮活跃 S 盒个数下界是否大于等于 25, 以满足安全性目标抵抗内部碰撞攻击。实验过程中每轮使用相同的消息差分变量。

首先, 假设 $l=1$, 通过实验可得到如下结果。

当参数 $t < 3$ 时, 也就是 $t=1$ 和 $t=2$, 本文分别穷尽搜索了所有的 8! 和 4! 种置换, 没有找到满足安全目标的结构。然而, 当参数 $t \geq 3$ 时, 总能搜索到结构安全的轮函数。

因此, 搜索分组最小的结构函数 $R(3,1,1)$, 可找到诸多满足安全目标的置换。

例如, 当 $P=(11,9,1,6,4,8,10,3,5,7,0,2)$ 的向量置换, 其四轮最小活跃 S 盒达到 25, 满足四轮传播定理。当十轮时, 最小活跃 S 盒能达到 80。但是, 结构函数的效率较低, 其指标 $rate=3$ 。

为得到效率更高的结构函数, 分别对 $R(3,2,1)$ 、 $R(3,3,1)$ 、 $R(4,1,1)$ 、 $R(4,2,1)$ 、 $R(4,3,1)$ 及 $R(4,4,1)$ 结构进行了搜索。实验结果证明, 各结构均存在 P 置换可使其满足安全目标。明显地, 结构函数 $R(4,4,1)$ 的 $rate=1$, 效率较高。

表 1 列出了部分 P 置换及其活跃 S 盒个数的下界, 以及达到全状态混淆所需的最低轮次。考虑达到全混淆的速度, 在达到安全性目标的情况下, 应选择达到全状态混淆轮次较低的置换 P 。由表可知, 可选择置换 P_2 或 P_5 。本文使用十轮活跃 S 盒数较高

表 1 $R(4,4,1)$ 的部分安全置换 P
Table 1 Partial safe permutation P of $R(4,4,1)$

No.	置换 P	四轮活跃 S 盒数	十轮活跃 S 盒数	全状态混淆轮次
P_1	9,6,12,15,5,2,8,11,1,14,4,7,13,10,0,3	25	78	5
P_2	7,13,10,0,6,3,9,12,2,15,5,8,14,11,1,4	25	80	3
P_3	13,10,0,6,3,9,12,2,15,5,8,14,11,1,4,7	25	88	4
P_4	1,7,13,10,0,3,9,6,12,15,5,2,8,11,14,4	25	78	5
P_5	2,15,5,11,8,14,1,7,4,10,13,3,0,6,9,12	25	78	3

的 P_2 来构造认证加密算法。

本文对 $l=2$ 和 $l=4$ 的结构进行了同样的实验,并未搜索到安全的结构,因此推测它是不安全的。本文中 MILP 的约束条件为基本约束,得到的最小活跃 S 盒个数可能不紧,可参考文献[18],通过添加约束条件或更换 P 的置换方式得到 $rate$ 值更小的安全轮函数结构。

4 认证加密算法 AMRAE

根据搜索实验结果,得到轮函数结构 $R(4,4,1)$ 安全且效率较高。

接下来,利用该轮函数结构和表 1 中的向量置换 P_2 设计认证加密算法 AMRAE,并进行安全性分析和效率评估。

AMRAE 认证加密算法输入包含:长度为 128 bit 的密钥 $K \in \{0,1\}^{128}$, 初始向量(称为 nonce) $N \in \{0,1\}^{128}$, 关联数据 A 和明文 M , 其中关联数据为与信息的发送者、发送时间等相关信息。

输出包含:密文 C 和长度为 128 bit 的标签 $T \in \{0,1\}^{128}$, 其中密文 C 与填充后的明文 M 长度相同。

4.1 关联数据与明文的填充

本文使用斐波那契数列(Fibonacci sequence)进行数据填充,首先,将前 16 个斐波那契数列的数模 256,分别用 8 bit 二进制序列表示,如表 2 所示。

表 2 斐波那契数列的 8 bit 二进制表示
Table 2 The 8-bit binary representation of a Fibonacci sequence

斐波那契数列	模 256 后二进制序列
1、1、2、3	00000001、00000001、00000010、 00000011
5、8、13、21	00000101、00001000、00001101、 00010101
34、55、89、144	00100010、00110111、01011001、 10010000
233、377、610、987	11101001、01111001、01100010、 11011011

然后,按从左到右的顺序对关联数据和明文进行填充,使其长度为 128 bit 的整数倍。即使关联数据和明文信息的长度恰好为 128 bit 的

整数倍,同样需要进行填充。填充后关联数据和明文分别记为 $A=(A_0, A_1, \dots, A_{4p-1})$, $M=(M_0, M_1, \dots, M_{4t-1})$, 其中 $A_i, M_j \in \{0,1\}^{32}$, $0 \leq i \leq 4p-1$, $0 \leq j \leq 4t-1$ 。其中, p 、 t 分别表示填充后关联数据与明文对应 4×32 bit 字的个数。

比如,明文为 $0x37abc=(001101111010101111100)_2$, 使用斐波那契二进制序列按从左到顺序进行填充,使其长度为 128 bit 整数倍,填充结果如下。

$(0011011110101011111000000000100000001000000100000001100000010100001000000011010001010100100010001101110101100110010000111010010111)_2$

4.2 算法轮函数

为简单起见,AMRAE 算法的轮函数用 $Round$ 表示,那么 $Round=R(4,4,1)$ 。第 i 轮状态矩阵由 16 个 32 bit 的字构成,记为 $S^i=(S_0^i, \dots, S_{15}^i) \in (Z_2^{32})^{16}$ 。

$Round$ 的输入为状态矩阵 S^i 和 4 个 32 bit 的字 $\bar{M}$, $\bar{M}=(\bar{M}_0, \bar{M}_1, \bar{M}_2, \bar{M}_3) \in (Z_2^{32})^4$, $\bar{M}$ 由填充后关联数据 A 或填充后的明文 M 产生。输出为新的状态矩阵 S^{i+1} 。则第 i 轮函数记为 $S^{i+1}=Round(\bar{M}_0, \bar{M}_1, \bar{M}_2, \bar{M}_3, S^i)$, 用图 2 所示。

4.3 AMRAE 认证加密的初始化

初始化过程主要完成密钥 K 和初始化变量 N 的混合,过程如下:

(1) 初始状态装载;

(2) 状态进行十轮迭代更新。

$$(S_0^0, S_1^0, S_2^0, S_3^0) = K;$$

$$(S_4^0, S_5^0, S_6^0, S_7^0) = N;$$

$$\lim_{x \rightarrow \infty} (S_8^0, S_9^0, S_{10}^0, S_{11}^0) =$$

$$0x27082818459045235360287471352662;$$

$$(S_{12}^0, S_{13}^0, S_{14}^0, S_{15}^0) =$$

$$0x49775724709369995957496696762772$$

For $i = 0$ to 9

$$S^{i+1} = Round(0, 0, 0, 0, S^i)$$

状态更新后生成状态 $S^{10}=(S_0^{10}, S_1^{10}, \dots, S_{15}^{10})$ 。

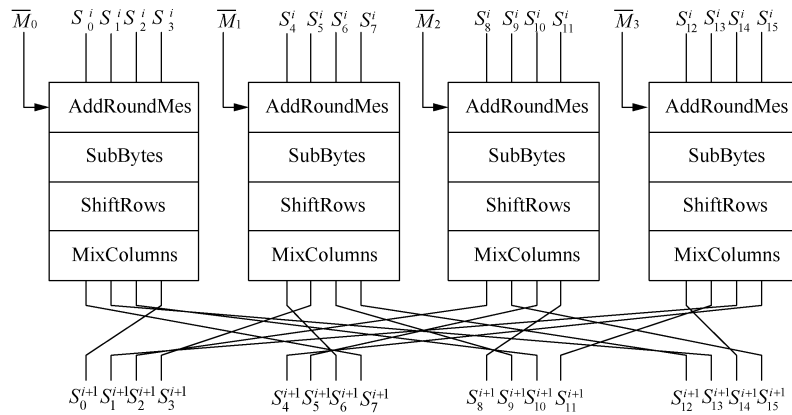


图 2 Round 轮函数结构

Figure 2 Structure diagram of round function Round

4.4 关联数据处理

对填充后的关联数据逐块处理，这里的块是 4 个 32 bit 的字，具体如下。

For $i = 0$ to $p-1$

For $j = 0$ to 3

$$S^{4i+j+11} = \text{Round}(A_{4i}, A_{4i+1}, A_{4i+2}, A_{4i+3}, S^{4i+j+10})$$

生成状态 $S^{4p+10} = (S_0^{4p+10}, S_1^{4p+10}, \dots, S_{15}^{4p+10})$ 。

4.5 明文加密过程

填充后的信息逐块进行处理，通过状态和明文生成密文 C ，随后进行状态更新，具体如下。

For $i = 0$ to $t-1$

$$C_{4i} = M_{4i} \oplus S_0^{4i+4p+10} \oplus S_4^{4i+4p+10} \oplus (S_8^{4i+4p+10} \wedge S_{12}^{4i+4p+10})$$

$$C_{4i+1} = M_{4i+1} \oplus S_1^{4i+4p+10} \oplus S_5^{4i+4p+10} \oplus (S_9^{4i+4p+10} \wedge S_{13}^{4i+4p+10})$$

$$C_{4i+2} = M_{4i+2} \oplus S_2^{4i+4p+10} \oplus S_6^{4i+4p+10} \oplus (S_{10}^{4i+4p+10} \wedge S_{14}^{4i+4p+10})$$

$$C_{4i+3} = M_{4i+3} \oplus S_3^{4i+4p+10} \oplus S_7^{4i+4p+10} \oplus (S_{11}^{4i+4p+10} \wedge S_{15}^{4i+4p+10})$$

For $j = 0$ to 3

$$S^{4i+j+4p+11} =$$

$$\text{Round}(M_{4i}, M_{4i+1}, M_{4i+2}, M_{4i+3}, S^{4i+j+4p+10})$$

得到密文 $C = (C_0, C_1, \dots, C_{4t-1})$ ，同时状态更新为 $S^{4t+4p+10}$ 。

4.6 认证标签生成

标签的生成分为如下 3 步。

(1) 依据关联数据和明文长度的更新状态。

首先，用 $adlen$ 、 $msglen$ 分别表示关联数据和明文的长度， $(M'_1, M'_2) = |adlen|_{64}$ ， $(M'_3, M'_4) = |msglen|_{64}$ ，其中， $|X|_{64}$ 表示 X 的 64 bit 二进制表示， $(M'_0, M'_1, M'_2, M'_3) \in (\mathbb{Z}_2^{32})^4$ 。

$$S^{4p+4t+11} = \text{Round}(M'_1, M'_2, M'_3, M'_4, S^{4p+4t+10})$$

(2) 以 $(0, 0, 0, 0) \in (\mathbb{Z}_2^{32})^4$ 为输入进行十轮迭代。

For $i = 0$ to 9

$$S^{i+4p+4t+12} = \text{Round}(0, 0, 0, 0, S^{i+4p+4t+11})$$

状态更新为 $S^{4t+4p+21}$ 。

(3) 由最后一轮状态生成标签。

$$T = (S_0^{4p+4t+21} \oplus S_4^{4p+4t+21} \oplus S_8^{4p+4t+21} \oplus S_{12}^{4p+4t+21}) \parallel (S_1^{4p+4t+21} \oplus S_5^{4p+4t+21} \oplus S_9^{4p+4t+21} \oplus S_{13}^{4p+4t+21}) \parallel (S_2^{4p+4t+21} \oplus S_6^{4p+4t+21} \oplus S_{10}^{4p+4t+21} \oplus S_{14}^{4p+4t+21}) \parallel (S_3^{4p+4t+21} \oplus S_7^{4p+4t+21} \oplus S_{11}^{4p+4t+21} \oplus S_{15}^{4p+4t+21})$$

最后，认证加密算法 AMRAE 输出密文 $C = (C_0, \dots, C_{4t-1})$ 和标签 T 。

4.7 解密认证过程

AMRAE 的解密认证算法输入为密钥 K 、初始向量 N 、关联数据 A 和密文 C ，输出为明文 M 或 $\perp$ ，包括初始化、关联数据处理、解密和认证过程。

初始化和关联数据处理与 AMRAE 认证加密过程中的初始化和关联数据处理相同。同时，生

成状态 $S^{4p+10} = (S_0^{4p+10}, S_1^{4p+10}, \dots, S_{15}^{4p+10})$ 。

解密过程：通过状态 S^{4p+10} 和密文 $C = (C_0, \dots, C_{4t-1})$ 生成明文 M ，同时更新状态。具体如下。

$$\begin{aligned}
 &\text{For } i = 0 \text{ to } t-1, \\
 &M_{4i} = C_{4i} \oplus S_0^{4i+4p+10} \oplus S_4^{4i+4p+10} \oplus \\
 &\quad (S_8^{4i+4p+10} \wedge S_{12}^{4i+4p+10}) \\
 &M_{4i+1} = C_{4i+1} \oplus S_1^{4i+4p+10} \oplus S_5^{4i+4p+10} \oplus \\
 &\quad (S_9^{4i+4p+10} \wedge S_{13}^{4i+4p+10}) \\
 &M_{4i+2} = C_{4i+2} \oplus S_2^{4i+4p+10} \oplus S_6^{4i+4p+10} \oplus \\
 &\quad (S_{10}^{4i+4p+10} \wedge S_{14}^{4i+4p+10}) \\
 &M_{4i+3} = C_{4i+3} \oplus S_3^{4i+4p+10} \oplus S_7^{4i+4p+10} \oplus \\
 &\quad (S_{11}^{4i+4p+10} \wedge S_{15}^{4i+4p+10}) \\
 &\text{For } j = 0 \text{ to } 3 \\
 &S^{4i+j+4p+11} = \text{Round} \cdot \\
 &\quad (M_{4i}, M_{4i+1}, M_{4i+2}, M_{4i+3}, S^{4i+j+4p+10})
 \end{aligned}$$

认证过程：利用 AMRAE 认证加密过程中的标签生成算法，得到新的标签 T' 。如果 T' 和 T 两个标签相等，解密成功且通过认证，输出明文 M ；否则认证解密失败，输出 $\perp$ 。

5 AMRAE 正确性分析

AMRAE 算法的加密生成标签过程和认证过程相似，有相同的初始化过程、关联数据处理过程和标签生成过程。

假设第 i 轮函数进行四轮迭代后产生新状态矩阵简写为 $S^{i+3} = \text{Round}^4(M_{4i}, M_{4i+1}, M_{4i+2}, M_{4i+3}, S^i)$ ，这里的 Round^4 表示四轮迭代函数。

加密过程或解密过程状态矩阵计算的简写为 $F(j, i) = S_{j+0}^i \oplus S_{j+4}^i \oplus (S_{j+8}^i \wedge S_{j+12}^i)$ ， $0 \leq j \leq 3$ 。

加密过程为

$$\begin{aligned}
 &\text{For } i = 0 \text{ to } t-1 \\
 &C_{4i} = M_{4i} \oplus F(0, 4i+4p+10) \\
 &C_{4i+1} = M_{4i+1} \oplus F(1, 4i+4p+10) \\
 &C_{4i+2} = M_{4i+2} \oplus F(2, 4i+4p+10) \\
 &C_{4i+3} = M_{4i+3} \oplus F(3, 4i+4p+10) \\
 &S^{4i+4p+14} = \text{Round}^4(M_{4i}, M_{4i+1}, M_{4i+2}, M_{4i+3}, S^{4i+4p+10})
 \end{aligned}$$

解密过程为

$$\begin{aligned}
 &\text{For } i = 0 \text{ to } t-1 \\
 &M_{4i} = C_{4i} \oplus F(0, 4i+4p+10) \\
 &M_{4i+1} = C_{4i+1} \oplus F(1, 4i+4p+10) \\
 &M_{4i+2} = C_{4i+2} \oplus F(2, 4i+4p+10) \\
 &M_{4i+3} = C_{4i+3} \oplus F(3, 4i+4p+10) \\
 &S^{4i+4p+14} = \text{Round}^4(M_{4i}, M_{4i+1}, M_{4i+2}, M_{4i+3}, S^{4i+4p+10})
 \end{aligned}$$

明显地，若加密过程和认证过程中对应轮次使用的状态矩阵 S^i 相同，可保证加解密的正确性。由 AMRAE 认证加密算法的加密算法和解密算法，可知初始化过程和关联数据处理过程相同，因此输出同样的状态矩阵为 $S^{4i+4p+10}$ ，所以加密过程和认证过程使用了相同的状态矩阵，保证了 AMRAE 算法加解密过程的正确性。

下面证明认证的正确性。

填充后的明文按 4 个 32 bit 的字分为一组，共分为 t 组。第 1 组明文加密过程所使用的状态矩阵为 S^{4p+10} ，第 2 组明文加密过程中使用进行四轮迭代输出的状态矩阵 S^{4p+14} 。以此类推，加密结束后得到状态矩阵为 $S^{4t+4p+10}$ 。同样密文按 4 个 32 bit 的字分为一组，共分为 t 组。第 1 组密文解密过程所使用的状态矩阵为 S^{4p+10} ，第 2 组密文解密过程中使用进行四轮迭代输出的状态矩阵同样为 S^{4p+14} 。以此类推，解密结束后得到明文 M 和状态矩阵为 $S^{4t+4p+10}$ 。

加密算法和解密算法使用相同的标签生成方法，由于输入状态矩阵、关联数据相同，生成的标签相同，保证了 AMRAE 算法认证的正确性。

6 AMRAE 安全性分析

直接设计的认证加密算法与基于工作模式的认证加密算法不同，不能进行安全性证明，只能通过不同的密码分析角度对其安全性进行评估。

首先，为了保证认证加密算法 AMRAE 抵抗密钥恢复攻击和伪造攻击能力达到 128 bit，且具有抵抗文献[12]中攻击方法的能力，提出如下规定。

(1) 初始向量是随机且带状态的比特串，即每次加密进行更新以不重复使用。

(2) 同一密钥下能够加密的关联数据和消息

的数量总和小于 2^{128} bit。

(3) 认证过程中, 验证成功, 则输出明文, 验证失败, 则输出 $\perp$ 。

根据本文设计的安全性目标, 在引入消息差分, 至少需要 25 个活跃 S 盒构造内部碰撞, 本文的 AMRAE 算法选择了表 1 中的 P_2 置换, 满足安全性目标, 因此 AMRAE 算法能够抵抗内部碰撞攻击。本文将 AMRAE 所使用的轮函数结构 $R(4, 4, 1)$ 与认证加密算法 ALE 所使用的 AES 进行了对比, 分别计算了两算法在密钥加运算约束参与和未参与情况下的活跃 S 盒下界, 如表 3 所示。

通过表 3 可知, 密钥加约束未参与运算下, 两个算法都满足四轮传播定理, 且 AMRAE 的 $R(4, 4, 1)$ 结构在十轮传播后可达到 100 个活跃 S 盒下界。但密钥加参与运算情况下, 即相关密钥差分分析下的 AES 仅达到四轮 5 个和十轮 17 个活跃 S 盒下界, 而轮函数结构 $R(4, 4, 1)$ 依然具有四轮 25 个和十轮 80 个活跃 S 盒下界。AMRAE 算法的初始化过程和生成标签过程进行了十轮轮函数迭代, 关联数据处理过程和加解密过程进行了四轮轮函数迭代, 说明 AMRAE 算法可以抵抗差分攻击, 且相对于相关密钥差分分析下的 AES 具有更高的安全性。

线性攻击^[23]是利用密文之间高概率的线性关系, 与随机函数进行区分。本文同样通过文献[11]中的方法对 AMRAE 算法进行评估。通过 MILP 方法构建其目标函数与约束条件得到其四轮和十轮活跃 S 盒下界分别为 25 和 100, 不仅满足线

性特征中活跃 S 盒下界 25 的四轮传播定理, 且提高了十轮活跃 S 盒的下界。因此 AMRAE 算法具有抵抗线性攻击的能力。

滑动攻击^[24]是通过找到滑动对的方法, 使中间的加密过程变得毫不相关, 以达到减少所要分析的密码体制的轮数的目的。为了避免滑动攻击, 应该使用不同的函数进行初始化和加密/身份验证, 即使很小的差别也是足够的。考虑各过程的抵抗差分/线性攻击能力和运行效率, 使消息处理和关联数据处理过程最小活跃 S 盒为 25 个。因此, AMRAE 进行四轮加密和身份验证过程, 以及十轮初始化过程和生成标签过程。本算法具有抵抗滑动攻击的能力。

猜测确定攻击^[25]主要通过猜测一些变量利用轮函数的传播得到一些新的变量, 从而恢复出所有状态变量。由于 AMRAE 中线性变换 MixColumn 的分支数是 5, 那么至少已知 4 byte 才能得到新的字节。其中向量置换和异或运算都是基于 32 bit 的字运算。所以整个猜测过程是以 32 bit 的字进行的, 因此猜测至多 3 个变量恢复所有状态是不可能的, 本文算法抵抗猜测确定攻击。

7 AMRAE 实现效率

本文在个人计算机上使用 Golang 语言编码实现了 AMRAE 算法, 采用了 MacBook Pro 2015 Intel Core i5 处理器。在实现过程中, 使 AMRAE 轮函数结构中的 4 个 AES 轮函数进行并行化处理。

表 3 活跃 S 盒下界比较
Table 3 Comparison of lower bounds of active S-boxes

Rounds	$R(4, 4, 1)$ 密钥加未参与	$R(4, 4, 1)$ 密钥加参与	AES 密钥加未参与	AES 密钥加参与
1	1	1	1	1
2	5	4	5	1
3	9	9	9	3
4	25	25	25	5
5	41	38	26	7
6	60	42	30	9
7	64	55	34	11
8	80	71	50	13
9	84	76	51	15
10	100	80	55	17

同时，在相同条件下编码实现了 AEGIS 和 AES-GCM 算法，并在表 4 和图 3 给出 3 种算法加密长度为 128 bit 到 40 kB 消息的时间消耗对比。

表 4 算法消耗时间
Table 4 Time consumed by the algorithm

消息长度	AMRAE/ μ s	AEGIS/ μ s	AES-GCM/ μ s
128 bit	126.73	120.07	106.03
1 kB	158.49	137.51	118.21
2 kB	180.28	158.333	156.37
4 kB	167.13	163.34	223.55
40 kB	625.35	603.25	1 122.47

通过表 4 和图 3 可以看出，AMRAE 的时间复杂度与 AEGIS 算法相比，具有相同的效率。当消息长度小于 4 kB 时，3 个算法的时间复杂度相似，但当消息长度大于 4 kB 时，AES-GCM 算法的时间复杂度明显偏高，在消息长度为 40 kB 时，接近 AMRAE 算法的两倍。即消息长度越长，AMRAE 算法的加密速度越快。总体来说，AMRAE 算法的效率较高。

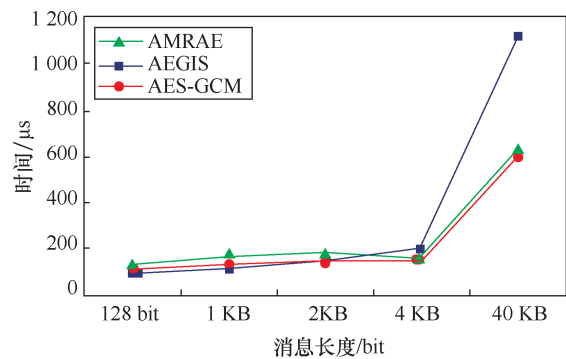


图 3 算法效率对比折线图
Figure 3 Algorithm efficiency comparison line chart

8 结束语

本文设计了一个基于 AES 轮函数的认证加密算法的通用迭代结构 $R(t,s,l)$ ，然后利用 MILP 方法搜索得到多个抵抗内部碰撞攻击的结构，如 $R(3,1,1)$ 、 $R(4,4,1)$ 等，结合 $R(4,4,1)$ 设计了高效率的认证加密算法 AMRAE，并用多种方法，分析了算法的安全性。利用 Golang 语言编码实现了新的 AMRAE 算法，与已有的 AES-GCM 相比，新算法具有更高的效率。下一步将继续对

AMRAE 算法进行改进及优化，实现更高效率。

参考文献:

- [1] BELLARE M, KOHNO T, NAMPREMPRE C. Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol[C] //Proceedings of the 9th ACM Conference on Computer and Communications Security. 2002: 1-11.
- [2] KRAWCZYK H. The order of encryption and authentication for protecting communications (or: How secure is SSL) [C]//Advances in Cryptology-CRYPTO 2001. 2001: 310-331.
- [3] BELLARE M, NAMPREMPRE C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm[C]//Advances in Cryptology-ASIACRYPT 2000. Berlin:Springer. 2000: 531-545.
- [4] BELLARE M, ROGAWAY P. Encode-then-encipher encryption: how to exploit nonces or redundancy in plaintexts for efficient cryptography[C]//Advances in Cryptology—ASIACRYPT 2000. 2000: 317-330.
- [5] ROGAWAY P, BELLARE M, BLACK J. OCB: a block-cipher mode of operation for efficient authenticated encryption[J]. ACM Transactions on Information and System Security (TISSEC), 2003, 6(3): 365-403.
- [6] DWORKIN M J. National institute of standards and technology (NIST). Recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality[R]. 2004.
- [7] BELLARE M, ROGAWAY P, WAGNER D. The EAX mode of operation[M]//Fast Software Encryption. 2004.
- [8] MCGREW D A, VIEGA J. The Galois/counter mode of operation (GCM)[J]. Submission to NIST Modes of Operation Process, 2004, (20).
- [9] GOLDWASSER S, MICALI S. Probabilistic encryption[J]. Journal of Computer and System Sciences, 1984, 28(2): 270-299.
- [10] SASAKI Y, LEI W. Message extension attack against authenticated encryptions: application to PANDA[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2016, 99(1): 49-57.
- [11] MOUHA N, WANG Q, GU D, et al. Differential and linear cryptanalysis using mixed-integer linear programming[C]//International Conference on Information Security and Cryptology. 2011: 57-67.
- [12] 田玉丹, 韦永壮. 认证加密算法 SCREAM 及 iSCREAM 的新伪造攻击[J]. 网络与信息安全学报, 2016, 2(1): 60-64.
TIAN Y D, WEI Y Z. New forgery attack on the authenticated cipher SCREAM and iSCREAM[J]. Chinese Journal of Network and Information Security, 2016, 2(1): 60-64.
- [13] 田玉丹, 韦永壮. 认证加密模型 JAMBU 的新分析[J]. 网络与信息安全学报, 2017, 3(7): 53-57.
TIAN Y D, WEI Y Z. New cryptanalysis of the authenticated cipher model JAMBU[J]. Chinese Journal of Network and Information Security, 2017, 3(7): 53-57.
- [14] WU H, PRENEEL B. AEGIS: a fast authenticated encryption algorithm[C]//Selected Areas in Cryptography-SAC. 2014: 185-201.
- [15] BOGDANOV A, MENDEL F, REGAZZONI F, et al. ALE:

- AES-based lightweight authenticated encryption[C]//International Workshop on Fast Software Encryption. 2013: 447-466.
- [16] NIKOLIC I. TiaoXin-346 [EB].
- [17] HOANG V T, KROVETZ T. AEZ v5: authenticated encryption by enciphering[EB].
- [18] ZHANG J, WU W, ZHENG Y. Security of SM4 against (related-key) differential cryptanalysis[C]//Information Security Practice and Experience. 2016: 65-78.
- [19] BILGIN B, BOGDANOV A, KNEŽEVIĆ M, et al. Fides: lightweight authenticated cipher with side-channel resistance for constrained hardware[M]//Cryptographic Hardware and Embedded Systems - CHES 2013. 2013.
- [20] BEIERLE C, JÉRÉMY J, STEFAN K, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS[C]//Annual Cryptology Conference. Berlin: Springer, 2016.
- [21] 张建, 吴文玲. 基于 SM4 轮函数设计的认证加密算法[J]. 电子学报, 2018, 46(6): 1294-1299.
- ZHANG J, WU W L. Authentication encryption based on SM4 round function[J]. Chinese Journal of Electronics, 2018, 46(6):1294-1299.
- [22] DAEMEN J, RIJMEN V. The design of rijndael[M]. Berlin: Springer, 2002.
- [23] MATSUI M. Linear cryptanalysis method for DES cipher[C]//Advances in Cryptology -EUROCRYPT '93. 1994.
- [24] 尤加勇, 李超. 针对 LEX 算法的截断滑动攻击[J]. 信息安全与通信保密, 2007(9): 96-98.
- YOU J Y, LI C. Interruption slide attack against stream cipher lex algorithm[J]. Information Security and Communications Privacy, 2007(9): 96-98.
- [25] MINAUD B. Linear biases in AEGIS keystream[C]//Selected Areas in Cryptography. 2014: 290-305.

[作者简介]



高国强(1995—),男,内蒙古兴安盟人,北京印刷学院硕士生,主要研究方向为认证加密算法分析与设计。



李子臣(1962—),男,河南温县人,博士,北京印刷学院教授、博士生导师,主要研究方向为公钥密码、数字签名、后量子密码。