AEUR：基于 uBlock 轮函数的认证加密算法设计

杨亚涛[1,2]，董辉[1]，刘建韬[1]，张艳硕[1]

（1. 北京电子科技学院电子与通信工程系，北京 100070；2. 西安电子科技大学通信工程学院，陕西 西安 710071）

摘 要：为了提升认证加密算法的实现效率，同时不降低算法的安全性，基于 uBlock 算法设计了一种新型认证加密算法 AEUR。首先，在分组密码算法 uBlock 轮函数的基础上，将抵抗内部碰撞攻击作为安全性目标，利用混合整数线性规划方法，搜索设计符合安全性目标的通用迭代算法结构 $R(t,s)$。其次，利用该结构设计了认证加密算法 AEUR，AEUR 由认证加密和解密验证两部分构成，两部分执行过程相同，不需要额外设计操作环节，从而减少算法的资源消耗。再次，通过对比轮数状态值来验证算法的正确性，采用线性攻击、滑动攻击等多种方法分析了算法的安全性。最后，采用 C 语言对算法进行了软件实现，证明所提算法具有良好的软件实现性能。结果表明，以软件运行时间计算，所提算法相比 AEGIS 和 ALE，效率分别提升了 3%和 46%；相比 AES-GCM 和 ACORN，效率分别提升了 74%和 92%，具有较好的综合性能。

关键词：认证加密；分组密码 uBlock；安全性分析；软件实现

中图分类号：TN92

文献标志码：A

DOI: 10. 11959/j. issn. 1000−436x.2023159

AEUR: authenticated encryption algorithm design based on uBlock round function

YANG Yatao[1,2], DONG Hui[1], LIU Jiantao[1], ZHANG Yanshuo[1]

1. Department of Electronic and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China
2. School of Telecommunication Engineering, Xidian University, Xi'an 710071, China

Abstract: In order to improve the efficiency of the implementation of the authenticated encryption algorithm without compromising the security of the algorithm, a new authenticated encryption algorithm AEUR was designed. Firstly, based on the uBlock round function, with resistance to internal collision attacks as the security objective, a mixed integer linear programming approach was used to search for generic iterative component $R(t,s)$ to meet the security objective. Secondly, the authenticated encryption algorithm AEUR was designed by using this component. AEUR consisted of two parts: authenticated encryption and decrypted verification, both of which performed the same process without the need to design additional operational sessions, reducing the algorithm's resource consumption. In addition, the correctness of the algorithm was verified by comparing the corresponding round state values, and the security of the algorithm was analyzed using various analysis methods such as linear attacks and sliding attacks. Finally, the algorithm was implemented in C language to prove the AEUR has good performance. The results show that the proposed algorithm has a better overall performance in terms of software runtime, with efficiency improvements of 3% and 46% compared to AEGIS and ALE, and 74% and 92% compared to AES-GCM and ACORN, respectively.

Keywords: authenticated encryption, block cipher uBlock, security analysis, software implementation

收稿日期：2023−04−26；**修回日期：**2023−08−14

基金项目：北京市自然科学基金资助项目（No.4232034）；中央高校基本科研业务费专项资金资助项目（No.328202222）；“通信工程”“电子信息工程”国家级一流本科专业建设点基金资助项目

Foundation Items: Beijing Natural Science Foundation (No.4232034), The Fundamental Research Funds for the Central Universities (No.328202222), National First-Class Under Graduate Dicipline Construction of “Communication Engineering” and “Electronic Information Engineering”

0　引言

认证加密（AE, authenticated encryption）算法在信息保护领域作用重大，能够兼顾数据的机密性和完整性。通过将消息认证码和加解密算法结合，可以实现认证标签的单向计算过程，且在加解密环节拥有足够的安全强度，以抵抗常见的密码分析。

目前，主流的基于分组密码的认证加密算法主要有 2 种设计方式：一是基于分组密码的认证加密方式，二是在现有的分组密码的基础上进行设计。前者依靠设计可靠模式结构并在底层使用分组密码算法为相关的加解密和认证验证提供服务，但提供服务的过程是不可见的，底层的算法可以相对简单地进行替换，这类典型的认证加密算法有 OCB[1]（offset codebook）、EAX[2]（encryption with authentication for transfer）、GCM[3]（galois/counter mode）。

随着物联网等信息产业的迅速发展，认证加密算法的需求环境也发生了很大的变化，之前的通用算法和基于分组密码的工作方式类在现有的资源受限环境下的表现不尽如人意，如 RFID（radio frequency identification）环境。直接设计类认证加密算法是指采用已验证安全性和其他性能指标的、成型的分组密码部件来进行认证加密算法的设计。这种设计方式致力于降低算法的实现代价和提升软硬件表现，因其采用消息来更新算法的状态，用于消息认证的实现代价较低。但这些算法的安全性不能通过可证明安全理论来证明，需依靠各种安全性分析方法，常见的此类型的认证加密算法有 AEGIS[4]、ALE（authenticated lightweight encryption）[5]、AEZ（advanced encryption standard-easy）[6]等。为获得更好的认证加密效率与安全性，本文提出了新的方案。

本文的贡献如下。

1) 本文提出了一种认证加密算法通用结构 $R(t,s)$。不同于已有采用 AES（advanced encryption standard）轮函数的直接设计类认证加密算法，该结构基于轻量级分组密码算法 uBlock[7]设计，以抵抗内部碰撞攻击为安全性目标，使用了混合整数线性规划（MILP, mixed integer linear programming）[8]方法，搜索确定结构中关键参数 t、s 和算法所使用的置换 P，保证该结构在 10 轮后有不少于 66 个活跃 S 盒。

2) 基于通用结构 $R(t,s)$设计了认证加密算法 AEUR（authenticated encryption algorithm based on uBlock round function）。AEUR 算法的设计基于 uBlock 轮函数和广义 Feistel 结构，以 4 个 uBlock 轮函数和置换 P 组成了算法的轮函数，轮函数用来更新状态值，状态值保证算法的正确性。算法处理数据的过程简洁有效，从而优化了算法运行过程，减少了资源消耗。AEUR 算法对多种安全性分析方法具有充分的抵抗能力，为国产密码算法应用提供了一条新的参考途径。

3) 应用 SSE（streaming SIMD extension）指令集[9]对 AEUR 算法进行了软件实现。对 AEUR 算法进行实现速率测试，以软件运行时间计算，AEUR 算法相比其他同类算法在运算速度上均有不同程度的提升，具有较好的综合性能。

1　相关研究

Bellare 等[10]在 ASIACRYPT 2000 上提出了认证加密这一全新概念，将认证和加解密在同一算法中完成，通过共享一个密钥来实现数据的机密性与完整性。近年来，部分学者拓宽了认证加密算法的应用领域，并基于不同的经典密码算法实现了新的认证加密方式。2002 年，Rogaway[11]提出了基于相关数据的认证加密（ADAE, associated-data authenticated encryption）算法，结合 OCB 和伪随机函数 PMAC（parallelizable MAC）构造了 ADAE 算法，这种认证方式在保证了明文信息的保密性和完整性同时也保证了相关数据的完整性。2008 年，Iwata[12]提出了一种针对区块密码的认证加密模式，其中加密部分由一个 CENC（common encryption）的密钥流生成，并结合了基于分组密码的哈希函数。2010 年，Sarkar[13]提出了并行认证加密模式，并与 IPMAC（improved parallelizable MAC）结合，获得新的认证加密方案。

随着对认证加密领域以及应用环境的不断研究和探索，原有的认证加密方案设计思路不能很好地适应新的应用环境。同时，基于新结构和新思想的方案不断被提出。CAESAR（competition for authenticated encryption: security, applicability, and robustness）竞赛的举办大大推动了认证加密算法设计的发展，其第三轮算法的特点介绍[1,4,14-16]如表 1 所示。

2018 年，张建等[17]基于 SM4 轮函数设计了认证加密算法 SMAE，该算法利用 MILP 方法构

表 1 CAESAR 竞赛第三轮算法特点

算法	算法类型	算法特点
AEGIS[4]	直接设计	整体采用序列密码框架，由初始化、加密、标签生成过程构成，底层算法采用 AES 算法的轮函数，能够使用 AES 指令集
DEOXYS[14]	直接设计	利用可调分组密码算法 Deoxys-BC 作为底层算法，以 AES 算法为基础来设计可调分组密码，进一步结合工作方式来设计认证加密算法
ASCON[15]	直接设计	采用 Sponge 结构中的 Duplex 结构来设计，置换为 SP 结构中的迭代函数，轮函数包含常数加、替换层和扩散层
ACORN[16]	直接设计	采用序列密码来设计，面向比特，包含 LFSR（linear feedback shift register），具有轻量级硬件实现优势，软件实现性能较好
OCB[1]	分组密码工作模式	OCB 算法并行运算性好，安全性可以得到证明；但 OCB 不能抵抗初始向量值（Nonce）重用，没有超越生日攻击的安全界
COLM/AES-COPA/ELMD[14]	分组密码工作模式	利用 AES 分组密码算法来构造，可以通过 AES 指令集实现软件优化

造了消息认证码和认证加密算法。2020 年，高国强等[18]基于 AES 算法底层的轮函数，实现了一种基于 AES 轮函数的认证加密算法 AMRAE。这 2 种算法都是基于现有的分组密码轮函数来设计的，但安全性分析与实现效率尚有不足。本文综合前述经验，分析了现有方案的不足，在上述学者的研究思路上进行了扩充改进，采用 2019 年我国第一届密码算法设计竞赛中获得一等奖的分组密码算法 uBlock 设计认证加密算法 AEUR，其在安全性和实现速率方面较 SMAE 和 AMRAE 均有不同程度的提升，为国产认证加密算法提供了一种新的参考。

2 基础知识

2.1 uBlock 轮函数

uBlock 算法[7]的分组和密钥长度为 128 bit 和 256 bit，记为 uBlock-128/128、uBlock-128/256、uBlock-256/256。本文采用 uBlock-128/128。

uBlock 算法对差分分析、线性分析、积分分析、不可能差分分析等密码分析方法都有相应的安全冗余。uBlock 算法的密钥扩展算法可以通过随用随生成的方法得到轮密钥，这样能够减少算法需要的存储空间。在本文中，$<<<b$ 表示循环左移 b bit，$\underset{32}{<<<}b$ 表示以 32 bit 为单位循环左移 b bit。

uBlock 算法整体采用 SPN（substitution permutation network）结构，轮函数如图 1 所示。输入 n bit 明文 x 和轮密钥 R_k^i，经异或（⊕）及 S 盒等操作，分别进入向量置换 PL_n 和 PR_n，与 R_k^i 异或后，输出 n bit 密文 Y。uBlock 函数由 $\frac{n}{4}$ 个相同的 4 bit S 盒构成，如表 2 所示。

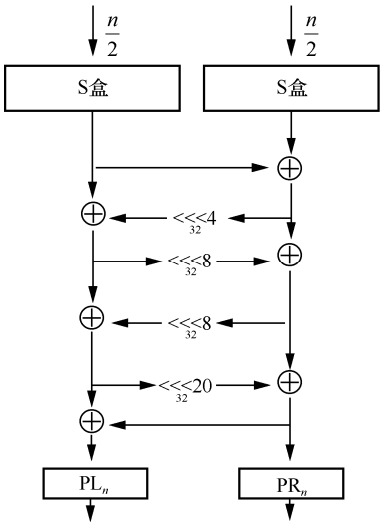

图 1 uBlock 算法轮函数

表 2 uBlock 算法 S 盒

x	$S(x)$	x	$S(x)$	x	$S(x)$	x	$S(x)$
0	7	4	b	8	f	c	0
1	4	5	a	9	e	d	3
2	9	6	d	a	1	e	2
3	c	7	8	b	6	f	5

PL_n 和 PR_n 都是 $\frac{n}{16}$ 个字节的向量置换，变换参数如表 3 所示。AEUR 采用 PR_{128}，其表达式为

$$\mathrm{PL}_{128}:\left(\{0,1\}^8\right)^8\to\left(\{0,1\}^8\right)^8$$

$$(y_0,y_1,y_2,\cdots,y_6,y_7)\to(z_0,z_1,z_2,\cdots,z_6,z_7)$$

表 3 PL_{128} 和 PR_{128} 变换参数

向量置换	变换参数
PL_{128}	{1,3,4,6,0,2,7,5}
PR_{128}	{2,7,5,0,1,6,4,3}

2.2　混合整数线性规划

MILP[8]是在一定约束的条件下对目标函数进行优化，遵循线性规划的优化问题方法。用 MILP 方法对密码算法进行差分分析和线性分析，就是通过设定相应的约束条件，将活跃 S 盒的个数之和最小设定为目标函数，并对其求解。

依据 S 盒的相关特性结合约束条件，再将最小活跃 S 盒的数量作为目标函数的构造条件，设计目标函数的数值即所求的最小值。完成上述构造后，就可以用求解器对此 MILP 问题进行求解。此外，当输入阶段差分和临时变量都被设为 0 或 1 时，对于求解器的求解速率更加友好[19]。

3　基于 uBlock 轮函数的通用结构设计

$R(t,s)$结构以 uBlock 算法轮函数和广义 Feistel 结构为基础，其安全性目标为在密钥长度为 128 bit 时，可以抵抗内部碰撞攻击。其构建过程采用了 MILP 方法，搜索符合安全性目标的结构参数 t、s 和置换 P，其中，t 为 uBlock 算法的执行次数，s 为轮函数输入信息的长度，长度表示消息的块数。

3.1　安全目标

因为 $R(t,s)$是基于分组密码算法轮函数和分组密码算法结构所设计的通用结构，不具备算法的完整性，从算法分析的角度来看，可以从某些分析方法的角度来设定其安全性目标。在 AEUR 算法通用结构的设计中，主要考虑的是对内部碰撞攻击的防御。碰撞攻击[20]利用生日悖论，分析算法本身或其等效结构，结合与轮密钥之间的对应关系来获得区分属性，继而分析得到密钥信息。碰撞攻击大大减少了原始穷举攻击的计算量。对于 $R(t,s)$结构，当存在 2 个不同消息的差分时，引入初始差分，经过多轮迭代，内部状态差分为 0。$R(t,s)$的密钥为 128 bit，当攻击复杂度大于 128 bit 时，对该结构的攻击是无效的。uBlock 算法使用的 4 bit S 盒的最大差分概率[7]为 2^{-2}，为满足上述安全性条件，在进行碰撞攻击时活跃 S 盒的个数不能少于 66 个。

3.2　通用结构 $R(t,s)$

$R(t,s)$结构是一种通用的、可迭代的认证加密算法轮函数结构，采用 uBlock 算法的轮函数，结合广义 Feistel 结构，如图 2 所示。置换 P 是区别于 uBlock 轮函数的向量置换 PL_n 和 PR_n 的新的置换，$F_i \in Z_2^{32}$ 表示消息，S_0～S_7 表示状态值。MesHandl 为消息 F_i 与状态值分别进行异或运算。

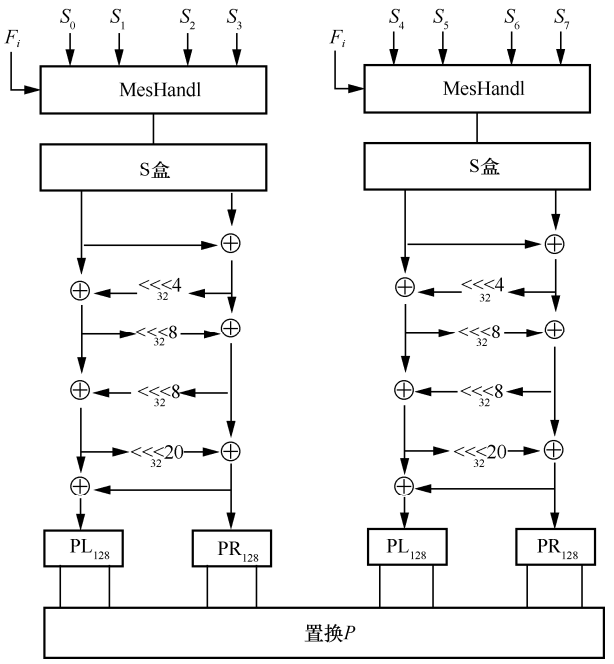

图 2　$R(t,s)$轮函数结构

定义 1　要对结构 $R(t,s)$的效率进行计量，需要确定一个指标 $\mathrm{rate}=\dfrac{t}{s}$，即处理 32 bit 的消息块需要的 uBlock 轮函数的个数。

$R(t,s)$的效率与 rate 相关性强，需要通过对结构的设计细节进行研究，以选取更高效率的结构。

3.3　MILP 模型搜索实验结果

假设置换 P 基于 32 bit，给出 t、s 和相应约束，结合 MILP 方法验证是否找到符合条件的置换 P。

当 rate=1 时，存在只有一个活跃 S 盒的差分路径使结构发生内部碰撞[17]。下面证明 rate 至少为 2 时结构 $R(t,s)$才可能避免内部碰撞。在搜索了 t=1 和 t=2 的 4!和 8!个置换后，发现并没有符合安全目标的结构；当 t=3 时，开始出现满足结构安全的轮函数。对于 $R(3,1)$，找到了多个可用的 P，如 P=(1,2,3,4,5,6,7,8,9,10,11,0），此时 P 被设为一个循环移位，进行 10 轮后，活跃 S 盒的个数达到 70，满足安全性目标，但其 rate=3，需要寻找效率值更高的 P。在对 $R(3,2)$、$R(3,3)$、$R(4,1)$、$R(4,2)$、$R(4,3)$、$R(4,4)$进行搜索后发现，以上结构均存在满足安全性目标的 P，其中 $R(3,3)$、$R(4,4)$的 rate=1，效率较高。但是 $R(3,3)$的算法全扩散轮数表现弱于 $R(4,4)$，故选择 $R(4,4)$[18]。

表 4 列出了部分使 $R(4,4)$符合安全性目标的置换 P 及其活跃 S 盒个数。在满足安全性目标的前提下，综合各个结构的实现效率，P_5 的实现效率和安全性均能够达到通用结构对认证加密算法实现的

要求，其扩散和混淆特性也满足作为认证加密算法部件的要求，经验证 P_5 的综合性能优于其他置换 P，故本文方案选取 P_5 作为通用结构的置换 P。

表 4 部分使 $R(4,4)$ 符合安全性目标的置换 P 及其活跃 S 盒个数

序号	置换 P	活跃 S 盒个数
P_0	11,5,7,9,14,8,12,1,15,0,3,2,10,6,13,4	67
P_1	6,0,8,13,1,15,5,10,4,9,12,2,11,3,7,14	67
P_2	9,7,4,11,6,0,3,8,14,5,2,13,1,15,10,12	66
P_3	0,15,11,8,12,9,6,3,13,1,2,4,10,7,5,14	66
P_4	7,2,12,5,8,4,6,11,14,9,1,15,13,3,10,0	68
P_5	6,11,12,9,10,2,15,3,0,13,1,4,7,14,8,5	66

3.4 通用结构 $R(4,4)$ 差分安全性分析

$R(4,4)$ 结构的安全目标是抵抗内部碰撞攻击。考虑到该结构作为认证加密算法的主要部件，其安全性需要从更多方面进行检验，可以通过差分分析来实现。

差分分析针对明文差分对和相应的密文差分，尽可能地获得更多的密钥信息。对于 $R(4,4)$ 结构，建立 MILP 差分模型，搜索该结构的高概率差分路径，该结构中存在异或和移位 2 种线性操作。在构建差分模型过程中，移位只是单纯地改变了差分值的位置，所以不需要进行多余限制。对于比特异或 $a \oplus b = c$，可以采用等式 $\triangle a + \triangle b + \triangle c = 2d$ [21] 结合 $R(4,4)$ 结构的差分性质来求解，过程中还需利用 MILP 进行约束，并用 Gurobi 求解器求解。

按照 $R(4,4)$ 结构进行差分路径分析，在每一轮都选取差分值与上一轮差分值相同的差分，构成差分路径，最终求得一条长度和精确度符合要求的差分路径概率。搜索结果如表 5 所示。

表 5 $R(4,4)$ 结构差分路径概率

轮数	差分路径概率
6	2^{-72}
7	2^{-97}
8	2^{-123}
9	2^{-152}

从上述结果可知，当 $R(4,4)$ 结构执行 9 轮时，差分路径概率为 2^{-152}，满足设定的差分复杂度安全目标 128 bit，因此 $R(4,4)$ 结构从理论上可以抵抗差分分析，且不存在可行的差分路径。

4 认证加密算法 AEUR 设计

AEUR 算法主要考虑底层算法的选取原则、通用结构设计和整体工作流程与安全性 3 个方面。

首先，对于底层算法 uBlock，主要关注其轻量级的设计思路和随用随生成的密钥扩展算法，且能够抵抗内部碰撞攻击。其次，对于通用结构 $R(t,s)$ 的设计，主要关注其安全性目标和使用 MILP 方法搜索时的参数结果，且 $R(t,s)$ 结构可以根据底层算法的安全性需求和运行特点，选择不同的置换 P 和底层轮函数个数，适用性广泛。最后，对于整体认证加密算法工作流程，主要考虑其认证加密过程和解密过程中对消息数据处理的运算效率和正确性。AEUR 算法的设计思路如图 3 所示。

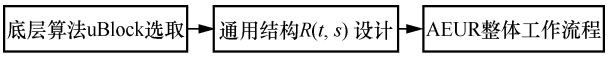

图 3 AEUR 算法的设计思路

AEUR 算法采用 $R(4,4)$ 作为轮函数，密钥、初始向量和随机常数作为状态值的初值，利用轮函数进行状态值的更新，利用状态值对数据进行加解密和生成标签操作。

4.1 AEUR 算法认证加密过程

AEUR 的输入包括 128 bit 的密钥 $k \in \{0,1\}^{128}$、初始向量 $\boldsymbol{N} \in \{0,1\}^{128}$、相关数据 $A = (A_0, \cdots, A_i, \cdots, A_{4d-1})$、$M = (M_0, \cdots, M_j, \cdots, M_{4q-1})$，$A_i, M_j \in \{0,1\}^{32}$，其中 M 为明文，d 和 q 为相关数据和明文填充对应的轮函数轮数。本文中 $\oplus$ 表示异或运算、$\wedge$ 表示与运算。

相关数据包含发送者、发送时间等信息。AEUR 算法的输出为密文 C，C 的长度与填充后明文 M 的长度相同；128 bit 长度的标签 tag，$\text{tag} \in \{0,1\}^{128}$。AEUR 的每一轮都处理 4 个 32 bit 的数据块，每一轮状态更新函数记为 $\text{Round}(S^i, Y^i)$，其中 $Y^i = \left(Y_{4i}^i, Y_{4i+1}^i, Y_{4i+2}^i, Y_{4i+3}^i\right)$，$Y$ 代表相关数据 A 和 M 等信息，按照算法不同阶段进行赋值。状态值集合 S^i 是由 16 个 32 bit 的字构成的数据集合，记为 $S^i = \left(S_0^i, S_1^i, \cdots, S_{15}^i\right)$，算法的轮函数结构如图 4 所示。

1) 初始化

首先进行相关数据与明文的填充。本文使用 PKCS5 算法进行数据填充。PKCS7 算法是目前常用加密算法都遵循的数据填充标准，PKCS5 算法作为 PKCS7 算法的子集算法，在数据块大小 blockSize 上固定为 128 bit，即数据始终会被切割为 128 bit 的数据块。然后计算需要填充的长度，由需要填充的字节数目来决定要填充的内容。此外，为了解决加解

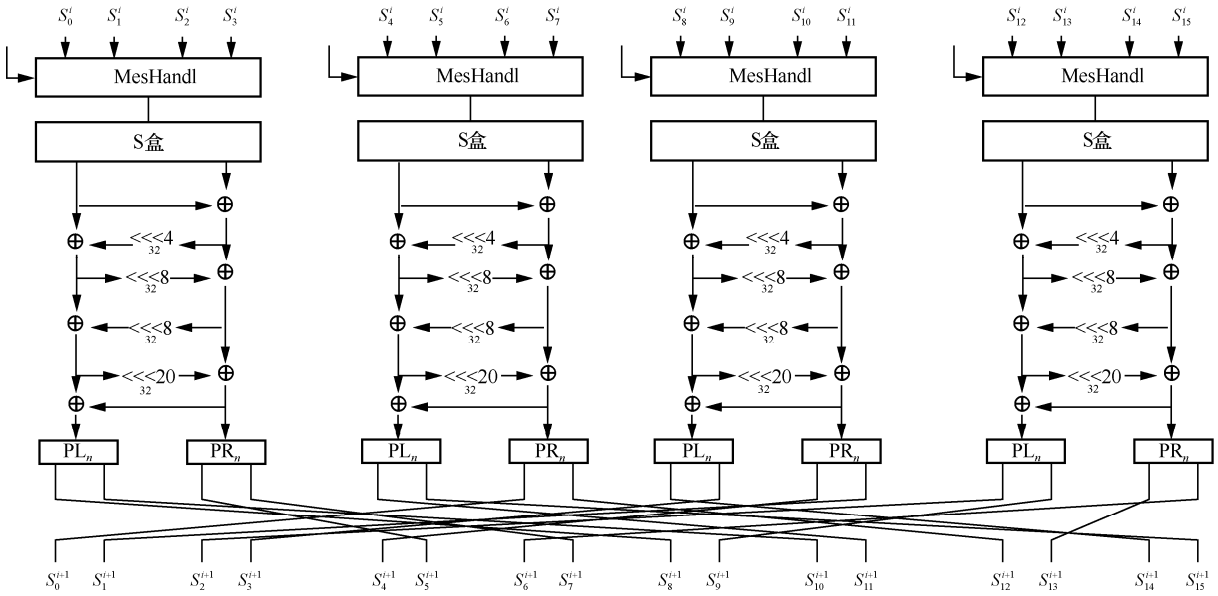

图 4　AEUR 算法的轮函数结构

密时对于数据填充的处理问题，当数据本身长度满足 128n bit 时，依据 PKCS5 的规则，在数据的尾部仍需要填充 8 个字节的内容，此时填充内容为 0x08。

将 k 和 N 赋值到状态值中，并利用轮函数更新 16 轮状态值，再对相关数据进行初始化，如算法 1 所示。

算法 1　AEUR 算法初始化

输入　k，N，相关数据 $A=(A_0,\cdots,A_i,\cdots,A_{4d-1})$

RC_1=(0x31415824000000000000000000000000)

RC_2=(0x53589793000000000000000000000000)

输出　算法状态值 $S^i=\left(S_0^i,S_1^i,\cdots,S_j^i\right)$

$\left(S_0^0,S_1^0,S_2^0,S_3^0\right)=k$；

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

$\left(S_8^0,S_9^0,S_{10}^0,S_{11}^0\right)=\mathrm{RC}_1$；

$\left(S_{12}^0,S_{13}^0,S_{14}^0,S_{15}^0\right)=RC_2$；

for $i=0$ to 16

Round(S^i,0)

end for

for $i=0$ to $d-1$，

Round$\left(S^{16+i},A_i\right)$

$A_i=\left(A_i^0,A_i^1,A_i^2,A_i^3\right)$

end for

2) 明文信息的处理

将明文信息填充后按 32 bit 分块，利用状态值与明文异或来生成密文，之后明文进行状态值更新，重复 $q-1$ 轮。具体算法过程如算法 2 所示。

算法 2　AEUR 算法明文信息处理

输入　明文 M

输出　算法状态值 S^{16+d+i}

for $i=0$ to $q-1$

$C_{4i}=M_{4i}\oplus S_0^i\oplus S_4^i\oplus\left(S_8^i\wedge S_{12}^i\right)$；

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

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

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

Round$\left(S^{16+d+i},M^i\right)$；

$M^i=\left(M_{4i}^i,M_{4i+1}^i,M_{4i+2}^i,M_{4i+3}^i\right)$

end for

3) 标签的生成

首先利用相关数据和消息长度 adlen 和 msglen 更新一轮状态值；然后迭代更新 16 轮；最后利用状态值生成标签 tag。具体过程如算法 3 所示。

算法 3　AEUR 算法标签生成

输入　相关数据长度 adlen，明文信息长度 msglen，状态值 $S^{16+q+d+i}$

输出　标签值 tag

$\left(M_1',M_2'\right)=\left|\mathrm{adlen}\right|_{64}$；

$\left(M_3',M_4'\right)=\left|\mathrm{msglen}\right|_{64}$；

Round($S^{16+q+d+i}$,M')；

Round($S^{i+q+d+17}$,0_{128})；

$$T=\left(S_0^i\oplus S_4^i\oplus S_8^i\oplus S_{12}^i\right)\|\left(S_1^i\oplus S_5^i\oplus S_9^i\oplus S_{13}^i\right)\|\left(S_2^i\oplus S_6^i\oplus S_{10}^i\oplus S_{14}^i\right)\|\left(S_3^i\oplus S_7^i\oplus S_{11}^i\oplus S_{15}^i\right)$$

return $C=\left(C_0,\cdots,C_{4q-1}\right)$,tag

4.2 AEUR 解密认证过程

AEUR 的解密认证算法输入为密钥 k、初始向量 $\boldsymbol{N}$、相关数据 A 和密文 C；输出为明文 M 或停止符 $\perp$，认证解密过程由初始化、密文信息处理和标签生成过程构成。

解密认证过程的初始化、密文信息处理与加密认证过程的输入输出相同。最后生成状态 S^{16+d}。

解密过程。利用状态 S^{16+d} 和密文 $C=(C_0,\cdots,C_{4q-1})$ 生成 M，同时更新状态。具体算法如算法 4 所示。

算法 4 AEUR 算法密文信息处理

输入 状态值 S^{16+d}，密文 C

输出 状态值 S^{16+d+i}，明文信息 M

for $i=0$ to $q-1$

$M_{4i}=C_{4i}\oplus\left(S_8^i\wedge S_{12}^i\right)\oplus S_4^i\oplus S_0^i$;

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

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

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

Round$\left(S^{16+d+i},M^i\right)$;

$M^i=\left(M_{4i}^i,M_{4i+1}^i,M_{4i+2}^i,M_{4i+3}^i\right)$

end for

认证过程。在解密过程结束后可以得到状态值 S^{16+d+q}，之后利用加密认证过程中的标签生成算法，得到标签 tag′，如果标签值 tag′ 与之前加密认证过程中生成的标签值 tag 相同，那么解密成功且通过认证，即解密验证算法执行成功，输出明文 M；否则解密验证过程失败，输出停止符 $\perp$。

4.3 基于通用结构 $R(t,s)$的认证加密算法设计思路

$R(t,s)$结构是能够满足不同算法使用的通用型迭代结构，满足分组密码算法要求的混淆和扩散特性，且可以保障认证加密算法的安全性和正确性。采用这种结构生成的认证加密算法总体类似于大型序列密码算法。该结构通过并行操作可以大大提升运行效率，根据底层算法的安全性需求和运行特点，可以选择不同的置换 P 和底层轮函数个数，从而构造出满足安全性和实用性要求的结构，继而将其进行组合排列，满足算法的设计需求。

5 正确性证明

加密认证和解密验证使用了相同的初始化、明/密文信息处理和标签生成过程，如算法 5 和算法 6 所示。

算法 5 AEUR 算法加密过程

输入 明文 M

输出 算法状态值 S^{16+d+i}

for $i=0$ to $q-1$

$C_{4i}=M_{4i}\oplus S_0^i\oplus S_4^i\oplus\left(S_8^i\wedge S_{12}^i\right)$;

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

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

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

Round(S^{16+d+i},M^i);

$M^i=\left(M_{4i}^i,M_{4i+1}^i,M_{4i+2}^i,M_{4i+3}^i\right)$

end for

算法 6 AEUR 算法解密过程

输入 状态值 S^{16+d}，密文 C

输出 状态值 S^{16+d+i}，明文信息 M

for $i=0$ to $q-1$

$M_{4i}=C_{4i}\oplus\left(S_8^i\wedge S_{12}^i\right)\oplus S_4^i\oplus S_0^i$;

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

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

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

Round(S^{16+d+i},M^i);

$M^i=\left(M_{4i}^i,M_{4i+1}^i,M_{4i+2}^i,M_{4i+3}^i\right)$

end for

由于算法的加解密与状态值紧密相关，如果在任意某一轮数时，算法在加解密过程中产生的状态值 S^i 相同，则算法满足正确性。执行 AEUR 算法时，其加密与解密的初始化过程、相关数据处理过程完全一致，每一轮产生的状态值都相同，因此 AEUR 算法满足正确性。

6 安全性分析

现有的可证明安全理论虽然可以对基于分组密码的认证加密模式进行安全性分析，但对于直接设计的认证加密算法，不能给出适合的安全假设。目前，直接设计的认证加密算法普遍使用针对密码算法的安全性分析方法[22]。为了让 AEUR 算法具备能够抵抗密钥恢复攻击和伪造攻击的能力，保证攻

击的复杂度大于 2^{128}，做出如下约束[18]。

1) 初始向量值（Nonce）的取值要随机，且不能复用，即每次使用认证加密算法之前都要更换随机初始向量值。

2) 在解密验证的过程中，明文信息对用户不可见，只有在验证成功后才会输出明文；如果验证失败，则输出停止符 $\perp$。

3) 加密时，相同密钥对相关数据和明文信息加密的最大数据长度不超过 2^{128} bit。

6.1 AEUR 算法安全性分析

1) 线性攻击

线性攻击[23]是一种已知明文攻击，也就是攻击者能够获取当下使用密钥状态下的明密文对。在对明文、密文和密钥三者满足的某种线性关系的概率 p 进行研究后，得到一个统计特征，利用该特征能够将分组密码和随机置换区分开。对分组密码而言，期待找到一个线性区分器来恢复最后一轮密钥。采用 MILP 方法构建目标函数和约束条件来搜索其最小活跃 S 盒，设置活跃 S 盒的下界，从准确度考虑，将搜索过程中每一轮的 S 盒个数进行输出，结果如表 6 所示。在线性攻击的标准方法评估下，AEUR 算法的 10 轮活跃 S 盒下界为 71，大于安全性目标设定的 66 个。且 AEUR 算法在标签生成和初始化阶段均使用了 16 轮迭代，有足够的安全冗余，所以其具有抵抗线性攻击的能力。

表 6　AEUR 算法在线性攻击条件下的活跃 S 盒个数

轮数	S 盒个数
2	8
3	16
4	26
5	31
6	37
7	48
8	52
9	60
10	71

2) 滑动攻击

滑动攻击[24]是受相关密钥攻击的启发而提出的。当分组密码在迭代过程中出现一定的自相似性时，可以使用滑动攻击对算法进行分析。与线性或者差分攻击不同，迭代轮函数的性质和执行轮数对滑动攻击的影响很小，轮数对算法抵抗滑动攻击的能力几乎没有影响。在密码算法中，如果迭代函数可以被分解为若干小的轮函数，那么就可以利用这一缺点进行滑动攻击。为了避免滑动攻击，算法的轮函数，特别是迭代的轮函数应该避免在执行过程中形成周期而被分解，在综合考虑算法抵抗几种攻击方法的能力以及对运行效率的影响后，与文献[18]的分析相同，AEUR 算法的轮函数并未生成周期，因此 AEUR 算法可以抵抗滑动攻击。

3) 猜测确定攻击

猜测确定攻击[25]的原理是对密码算法实现过程中的一些变量进行猜测，猜测的变量在轮函数中迭代可得到新的变量，利用得到的变量恢复出算法实现过程中相应的状态值。AEUR 算法的扩散层来源于 uBlock 算法，其扩散层的二元域矩阵分支数为 8，攻击者需要已知 64 bit 才能得到下一个字节变量的值。AEUR 算法轮函数是基于状态值集合 S^i 构造的，有 16 个 32 bit 的状态值。那么对于 AEUR 算法的猜测确定攻击，则需要以 32 bit 的状态值进行攻击，为了满足算法的安全性目标，攻击复杂度最高临界值被设定为 2^{128}，显然想通过 4 个变量来恢复其他的状态值并不现实，所以 AEUR 算法可以抵抗猜测确定攻击。

4) 状态恢复攻击

Pelican MAC 攻击[26]主要利用小尺寸的内部状态来构造基于生日悖论的碰撞。AEGIS、ALE 算法都具有较大的内部状态，在 AEUR 算法中，状态值 S^i 的大小为 16×32=512 bit，足以抵抗生日攻击。当攻击者使用相同的密钥和随机数进行多次重复攻击时，可能会在标签生成过程中对状态值注入差分，特别是当标签的长度小于算法的密钥长度时，认证加密算法面对这种攻击会显得脆弱。在 AEUR 算法中，轮函数结构的安全性目标为抵抗内部碰撞攻击，因此针对状态恢复攻击是不现实的。在多次攻击之后，可能获得相对成功的伪造，但无法恢复整个状态值。因此 AEUR 算法可以抵抗状态恢复攻击。

5) 伪造攻击

伪造攻击是指攻击者伪造出通过验证的密文来进行攻击。$R(4,4)$在 9 轮迭代情况下的差分路径概率为 2^{-152}，远超过设定的 128 bit 安全性复杂度目标，因此，该结构能够抵抗差分分析，且无可用差分路径。AEUR 算法的结构与 $R(4,4)$相同，且迭代轮数超过 10 轮，拥有绝对安全冗余，无法实现伪造。因此，AEUR 可以抵抗伪造攻击。

AEUR 算法作为直接设计类认证加密算法，其

底层算法 uBlock 对差分分析和不可能差分分析等具有良好的抵抗性。$R(t,s)$结构可以抵抗差分分析和伪造攻击等，并以 128 bit 的复杂度为安全性目标，保障结构的安全性。因此可以认为 AEUR 算法对差分分析等方法也有足够的抵抗能力。

6.2 安全性对比分析

为了进一步说明 AEUR 算法的安全性，选用近年来最有代表性的具有同类结构的 AEGIS 算法、SMAE 算法和 Pyjamask 算法[27]作为对比对象。

AEGIS 算法[4]在拒绝初始向量重用的情况下，在标签生成过程中，恢复密钥攻击的速度要慢于穷举搜索，因此状态恢复攻击对于拒绝 Nonce 重用的 AEGIS 无法实施。在伪造攻击中，解密的明文若验证成功，当 t 为标签值 tag 的长度时，有 2^{-t} 的概率可以被攻击者破解。AEGIS 的标签长度为 128 bit，在恢复状态时至少需要 2^{128} 次伪造尝试，由计算复杂性理论可知，这一数值在计算上是不可接受的。因此，AEGIS 对状态恢复攻击和伪造攻击具有安全性。

SMAE 算法[17]就猜测确定攻击而言，因为其底层函数来源于 SM4 算法，SM4 算法中线性变换 L 的分支数为 5，即对于 SMAE 算法而言，要想通过猜测确定攻击来分析算法，至少需要得到 32 bit 的输出值，才能恢复新的字节值，同时考虑到算法的 128 bit 的计算安全性指标，超过 3×32 bit 的恢复状态攻击不可行，因此 SMAE 算法对于抵抗猜测确定攻击具有安全性。在线性攻击分析中，计算搜索线性掩码成立的最优偏差为 $2^{-92.43}$，区分攻击和明文恢复攻击的数据复杂度约为 2^{184}，因此 SAME 算法对于线性攻击具有安全性。

AEUR 算法对于多种攻击具有安全性，较 SMAE[17]和 AEGIS[4]更全面，对比细节如表 7 所示。

表 7 AEUR 算法和其他算法的安全性对比

算法	线性攻击	伪造攻击	猜测确定攻击	状态恢复攻击	滑动攻击
SMAE	√	√	√	—	—
AEGIS	√	√	√	√	—
AEUR	√	√	√	√	√

2020 年，Goudarzi 等[27]提出了 Pyjamask 算法并给出其规范和 AEAD（authenticated encryption with associated data）建议。2022 年，贺水喻等[28]基于 Pyjamask 算法提出一种对明文进行伪造的方法，可以伪造出认证标签。在对 Pyjamask 和 AEUR 算法的 AEAD 安全性进行对比分析时，两者均具有良好的认证加密安全性。AEAD 安全性要求算法具有保密性、真实性、对称性和随机分配的特点。

1) 在实际认证加密执行过程中，AEUR 算法能够抵抗线性攻击等多种攻击方法，密文解密后对应的明文正确，标签值对应正确。除了明文长度之外，其他关于明文的内容是未知的，满足保密性。

2) AEUR 由于 $R(t,s)$结构，未经有效的密码分析检测，攻击者不可能更改密文的底层，满足真实性。

3) AEUR 算法对各类明文进行加密和对各类密文进行解密时使用的密钥相同，满足对称性。

4) AEUR 算法的加密过程是随机分配的，同一明文的 2 个消息会产生不同的密文，攻击者无法获知明文与密文的对应关系，满足随机分配性。

综上，AEUR 算法满足对称密码算法理想的 AEAD 安全性。

7 效率与实现分析

本节使用 C 语言结合 SSE 指令集实现 AEUR 算法。环境为 Intel 处理器，16 GB 内存，Visual Studio 2019。AEUR 算法中的 uBlock 轮函数采用 SSE 实现。SSE 指令集采用 SIMD（single instruction multiple data）来提升数据的并行操作性和算法的实现效率。

将 AEUR 算法的时间复杂度和空间复杂度与其他算法进行对比分析，结果如表 8 所示。

表 8 计算复杂度对比

算法	时间复杂度	空间复杂度
SMAE	$O(\log n)$	$O(n)$
AMRAE	$O(n)$	$O(\log n)$
AEUR	$O(\log n)$	$O(n)$

由表 8 可知，AEUR 算法的时间复杂度与 SAME 算法相同，优于 AMRAE 算法[18]；AEUR 算法的空间复杂度与 SMAE 算法相同，低于 AMRAE 算法。参考 AEUR 的存储开销和运行开销等，其综合性能良好。

AEUR 算法认证加密速率如图 5 所示。从图 5 可以看出，AEUR 算法认证最高为 5.41 Gbit/s，最低为 3.91 Gbit/s，平均为 4.63 Git/s，考虑到 AEUR 算法的运行环境，相比文献[17]中 SMAE 算法 3.8 Gbit/s 的加密速率，AEUR 算法认证加密速率对比 SMAE 算法仍有较大优势。

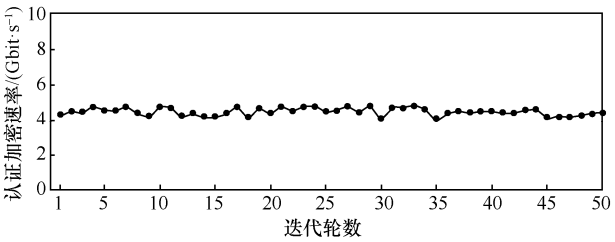

图 5　AEUR 算法认证加密速率

由表 9 可知，AEUR 的认证加密速率相比 AEGIS[4]、ALE[5]效率分别提升了 3%和 46%，相比 AES-GCM[3]、ACORN[29]算法优势较明显，加密速率分别提升了 74%和 92%。

表 9　　算法速度比较

算法	运行 1 MB 消息所需字节周期/（cycle·MB^{-1}）
AEGIS	0.80
AES-GCM	3.05
ALE	1.44
ACORN	9.70
AEUR	0.78

另外，通过测试不同数据长度的认证加密运算过程，可以得到如图 6 所示的状态曲线。随着待处理数据长度的增加，AEUR 算法执行的效率较 AEGIS 算法和 AES-GCM 算法更加稳定。

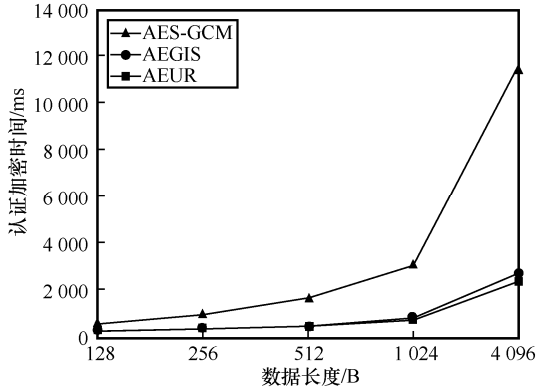

图 6　算法认证加密时间随数据长度变化

结合图 5、图 6 和表 9 可以得出结论，AEUR 算法与以 AES 算法为基础设计的 AEGIS 算法、ALE 算法、AES-GCM 算法和以序列密码为基础设计的 ACORN 算法相比，具有更优的运算性能。

通过上述对计算复杂度、认证加密速率和算法效率的对比分析可以看出，本文的 AEUR 算法较其他认证加密算法具有技术优势。其原因一方面是 AEUR 的底层结构是基于轻量级分组密码算法 uBlock，uBlock 算法的密钥扩展算法可以通过随用随生成的方法得到轮密钥，在一定程度上缩小了存储空间，并且能够提升运算效率。而且所使用的 MILP 方法可以根据所采用的分组密码算法 S 盒的特性来设置约束条件，使其适用性得到增强。

另一方面是 AEUR 算法的设计基于分组密码 uBlock 的轮函数和广义 Feistel 结构，以 4 个 uBlock 轮函数和置换 P 组成了算法的轮函数，状态值的更新通过轮函数来完成，因此算法处理数据的过程在一定程度上得到了优化，也减少了资源消耗，从而降低了运算复杂度，提升了运算速率。

8　结束语

本文采用国产分组密码算法 uBlock 轮函数结合混合整数线性规划方法，设计了一个适用于认证加密算法的通用迭代结构 $R(t,s)$，并基于 $R(t,s)$结构设计了认证加密算法 AEUR。AEUR 算法由初始化、明文信息处理和生成标签过程构成。对算法的正确性与安全性进行分析，并与同类算法进行对比，AEUR 表现较好。使用 C 语言结合 SSE 指令集编码实现了 AEUR 算法，测试了算法的软件实现效率。AEUR 算法与以 AES 算法为基础设计的 AEGIS 算法、ALE 算法、AES-GCM 算法和以 SM4 算法为基础构造的 SMAE 算法，以及以序列密码为基础设计的 ACORN 算法相比，具有更优的实现性能。

随着口令认证加密技术的发展，该方向得到了广泛研究[30-31]，未来 AEUR 算法可结合口令安全技术应用于物联网、隐私计算等领域。

参考文献：

[1] ROGAWAY P, BELLARE M, BLACK J. OCB[J]. ACM Transactions on Information and System Security, 2003, 6(3): 365-403.

[2] BELLARE M, ROGAWAY P, WAGNER D. The EAX mode of operation[C]//International Workshop on Fast Software Encryption. Berlin: Springer, 2004: 389-407.

[3] MCGREW D A, VIEGA J. The security and performance of the galois/counter mode (GCM) of operation[C]//Proceedings of International Conference on Cryptology in India. Berlin: Springer, 2004: 343-355.

[4] WU H J, PRENEEL B. AEGIS: a fast authenticated encryption algorithm[C]//International Conference on Selected Areas in Cryptography. Berlin: Springer, 2014: 185-201.

[5] BOGDANOV A, MENDEL F, REGAZZONI F, et al. ALE: AES-based lightweight authenticated encryption[C]//International Workshop on Fast Software Encryption. Berlin: Springer, 2014: 447-466.

[6] HOANG V T, KROVETZ T, ROGAWAY P. Robust authenticated-encryption AEZ and the problem that it solves[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2015: 15-44.

[7] 吴文玲, 张蕾, 郑雅菲, 等. 分组密码 uBlock[J]. 密码学报, 2019, 6(6): 690-703.
WU W L, ZHANG L, ZHENG Y F, et al. The block cipher uBlock[J]. Journal of Cryptologic Research, 2019, 6(6): 690-703.

[8] MOUHA N, WANG Q J, GU D W, et al. Differential and linear cryptanalysis using mixed-integer linear programming[C]//International Conference on Information Security and Cryptology. Berlin: Springer, 2012: 57-76.

[9] ZABUNOV S. Digital signal processing in RadioSolariz project using SSE2[J]. Aerospace Research in Bulgaria, 2022, 34: 66-71.

[10] 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.

[11] ROGAWAY P. Authenticated-encryption with associated-data[C]//Proceedings of the 9th ACM Conference on Computer and Communications Security. New York: ACM Press, 2002: 98-107.

[12] IWATA T. Authenticated encryption mode for beyond the birthday bound security[C]//International Conference on Cryptology in Africa. Berlin: Springer, 2008: 125-142.

[13] SARKAR P. Pseudo-random functions and parallelizable modes of operations of a block cipher[J]. IEEE Transactions on Information Theory, 2010, 56(8): 4025-4037.

[14] GRUBER M, PROBST M, TEMPELMEIER M. Persistent fault analysis of OCB, DEOXYS and COLM[C]//Proceedings of 2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC). Piscataway: IEEE Press, 2019: 17-24.

[15] DOBRAUNIG C, EICHLSEDER M, MENDEL F, et al. ASCON v1.2(submission to the CAESAR competition)[EB].2016.

[16] WU H J. ACORN: a lightweight authenticated cipher (v3) [EB]. 2016.

[17] 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. Authenticated encryption based on SM4 round function[J]. Acta Electronica Sinica, 2018, 46(6): 1294-1299.

[18] 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.

[19] BORGHOFF J, KNUDSEN L R, STOLPE M. Bivium as a mixed-integer linear programming problem[C]//International Conference on Cryptography and Coding. Berlin: Springer, 2009: 133-152.

[20] TOLBA A M R. Trust-based distributed authentication method for collision attack avoidance in VANETs[J]. IEEE Access, 2018, 6: 62747-62755.

[21] 刘帅, 关杰, 胡斌, 等. 基于 MILP 的轻量级密码算法 ACE 的差分分析[J]. 通信学报, 2023, 44(1): 39-48.
LIU S, GUAN J, HU B, et al. Differential analysis of lightweight cipher algorithm ACE based on MILP[J]. Journal on Communications, 2023, 44(1): 39-48.

[22] 吴文玲. 认证加密算法研究进展[J]. 密码学报, 2018, 5(1): 70-82.
WU W L. Research advances on authenticated encryption algorithms[J]. Journal of Cryptologic Research, 2018, 5(1): 70-82.

[23] BEYNE T. A geometric approach to linear cryptanalysis[C]//International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2021: 36-66.

[24] DUNKELMAN O, KELLER N, LASRY N, et al. New slide attacks on almost self-similar ciphers[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2020: 250-279.

[25] MINAUD B. Linear biases in AEGIS keystream[C]//International Conference on Selected Areas in Cryptography. Berlin: Springer, 2014: 290-305.

[26] YUAN Z, WANG W, JIA K T, et al. New birthday attacks on some MACs based on block ciphers[C]//Annual International Cryptology Conference. Berlin: Springer, 2009: 209-230.

[27] GOUDARZI D, JEAN J, KÖLBL S, et al. Pyjamask: block cipher and authenticated encryption with highly efficient masked implementation[J]. IACR Transactions on Symmetric Cryptology, 2020, 2020: 31-59.

[28] 贺水喻, 魏悦川, 潘峰, 等. 对认证加密算法 Pyjamask 的伪造攻击[J]. 计算机工程与科学, 2022, 44(12): 2140-2145.
HE S Y, WEI Y C, PAN F, et al. Forgery attack on the authenticated encryption algorithm Pyjamask[J]. Computer Engineering and Science, 2022, 44(12): 2140-2145.

[29] 张国双, 陈晓, 王安, 等. 面向 ACORN v3 消息认证码的随机差分故障分析[J]. 密码学报, 2021, 8(3): 498-520.
ZHANG G S, CHEN X, WANG A, et al. Random differential fault attack for ACORN v3 message authentication code[J]. Journal of Cryptologic Research, 2021, 8(3): 498-520.

[30] WANG D, CHENG H B, WANG P, et al. Zipf's law in passwords[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11): 2776-2791.

[31] 王平, 汪定, 黄欣沂. 口令安全研究进展[J]. 计算机研究与发展, 2016, 53(10): 2173-2188.
WANG P, WANG D, HUANG X Y. Advances in password security[J]. Journal of Computer Research and Development, 2016, 53(10): 2173-2188.

[作者简介]

杨亚涛（1978− ），男，河南平顶山人，博士，北京电子科技学院教授、博士生导师，西安电子科技大学硕士生导师，主要研究方向为信息安全、抗量子密码、白盒密码、密码协议和算法。

董辉（1997− ），男，山东济南人，北京电子科技学院硕士生，主要研究方向为认证加密和白盒密码。

刘建韬（1998− ），男，山东潍坊人，北京电子科技学院硕士生，主要研究方向为认证加密和分组密码。

张艳硕（1979− ），男，陕西宝鸡人，博士，北京电子科技学院副教授、硕士生导师，主要研究方向为密码学理论及应用。