文章编号:1671-4598(2023)06-0183-08 DOI:10.16526/j.cnki.11-4762/tp.2023.06.028 中图分类号:TP309.2;TP368.1 文献标识码:A

基于轻量型 AES 加密算法的浮空器平台数据传输方案

张馨方[1,2], 周江华[1,2]
(1. 中国科学院 空天信息创新研究院, 北京 100094;
2. 中国科学院大学 航空宇航学院, 北京 100094)

摘要: 针对浮空器平台在数据传输过程中受到自身处理器性能限制的问题, 提出了一种基于轻量型 AES 加密算法的浮空器平台数据传输方案; 首先, 方案以 AES 加密算法为基础, 通过寻找轮函数循环的局部最优次数和将状态矩阵行移位变换改为列移位变换实现轻量型 AES 加密算法; 其次, 通过字节代换、列移位变换、列混合和轮密钥加 4 个步骤, 设计以七次轮函数循环为核心的轻量型 AES 加密算法; 最后, 通过字节填充和矩阵旋转两个操作对过往不同类型的浮空器平台飞行数据进行预处理, 并将预处理后的数据作为明文数据源输入对传输方案进行测试和分析, 验证了轻量型 AES 加密算法的安全性和有效性; 实验结果表明, 该算法与 AES 加密算法相比, 在保证数据安全传输的同时提高了算法运行速度, 可以较好地应用于浮空器平台。

关键词: 平流层浮空器; AES 加密算法; 轻量型 AES 加密算法; 数据传输; 嵌入式系统

Data Security Transmission Scheme of Aerostat Platform Based on Lightweight AES Encryption Algorithm

ZHANG Xinfang[1,2] ZHOU Jianghua[1,2]
(1. Institute of Aerospace Information Innovation, Chinese Academy of Sciences, Beijing 100094, China;
2. School of Aeronautics and Astronautics, University of Chinese Academy of Sciences, Beijing 100094, China)

Abstract: Aiming at the problem that an aerostat platform is limited by its own processor performance in the process of data transmission, a data transmission scheme of the aerostat platform based on lightweight advanced encryption standard (AES) encryption algorithm is proposed. Firstly, based on the AES encryption algorithm, the scheme realizes the lightweight AES encryption algorithm by finding the local optimal times of the loop function and changing the row shift transform of the state matrix to the column shift transform. Secondly, through four steps of byte substitution, column shift transformation, column mixing and round key addition, the lightweight AES encryption algorithm based on seven round function cycle is designed. Finally, the flight data of different types of aerostat platform are preprocessed by the byte filling and matrix rotation, the pre-processed data are used as the data source to test and analyze the transmission scheme, which verifies the security and effectiveness of the lightweight AES encryption algorithm. Experimental results show that compared with AES encryption algorithm, the proposed algorithm can not only ensure the safe transmission of the data, but also improve the running speed of the algorithm, and can be applied to the aerostat platform better.

Keywords: stratospheric aerostat; AES encryption algorithm; lightweight AES encryption algorithm; data transmission; embedded systems

0 引言

近年来, 在大气环境实时监测、自然灾害监视和区域侦察监视等方面, 距地面 18~35 km 高的平流层浮空器得到了广泛的应用[1-2]。国内外在长航时平流层浮空器方面取得较大突破, 谷歌公司的 Loon 气球最大飞行航时已超过 300 天[3]。平流层浮空器在远离本土执行长航时任务时, 其数据安全传输问题也随之显现。一方面, 常规的浮空器平台的数据传输方式为明文无线传输, 安全性较更低[4], 若其敏感数据被泄露将会留下严重的安全隐患。另一方面, 平流层浮空器平台的处理器大多采用微型嵌入式系统, 由于微型嵌入式系统计算能力弱且资源性能有限[5], 使其所需的安全技术方法也更苛刻。因此, 需要一种不但能够实现平台数据安全传输, 而且速度快、效率高且实时性强的数据传输方法。

加密算法是目前实现数据安全传输主流的技术之一,

收稿日期:2022-11-01; 修回日期:2022-12-07。
作者简介:张馨方(1998-),女,山西临汾人,硕士研究生,主要从事嵌入式系统开发方向的研究。
周江华(1973-),男,江西鹰潭人,研究员,博士生导师,主要从事浮空器飞行控制、飞行力学方向的研究。
引用格式:张馨方,周江华.基于轻量型 AES 加密算法的浮空器平台数据传输方案[J].计算机测量与控制,2023,31(6):183-190.

根据算法使用密钥的不同，通常将加密算法分为对称加密算法和非对称加密算法[6]。与非对称加密算法相比，对称加密算法运算量更小、性能更强且加密速度更快。AES（高级加密标准）是目前世界上最常见且使用最广泛的对称分组密码算法之一[7]，经过 AES 加密算法处理后的数据保留了明文的数据格式，其数据长度也不会发生改变，更不会破坏浮空器平台数据的传输结构，在通信容量有限的系统中具有不可替代的优势。

AES 加密算法是一类安全性能强、移植操作简单并且应用前景广阔的加密方法，因此，对该算法的研究与改进是近年来的研究热点之一。N. Gangurde[8] 等人将 AES 加密算法与 QR 码结合，设计了一套安全可靠的票务系统，防止了用户隐私的泄露；Y. W. Ke[9] 等人提出了一种基于时间自动机的嵌入式实时操作系统中密码可行性的定量判断方法，验证了 AES 加密算法运行在嵌入式实时操作系统上的可行性；M. Gupta[10] 等人提出通过将算法中的 S 盒拆分为上下两部分，利用多线程任务并行在双 S 盒中查找代换字节，但由于引入了双 S 盒，该方法在提高运行速度的同时扩大了对处理器的内存占用；Z. Rahman[11] 等人提出了一种基于混沌映射和三维 S 盒的密钥轮换技术来提高 AES 加密算法的安全性，确保物联网下智能家居数据传输系统的可靠性，但是提高算法安全性的同时由于添加了新的算法并且提高了 S 盒维度，使得算法计算量增大且运行时间增长。

综上所述，AES 加密算法在不同领域得到了广泛应用，然而，若直接将该算法应用到浮空器平台这一微型嵌入式系统时，受到嵌入式系统处理器计算速度和性能资源的限制，将会导致算法运行速度降低。为了保障浮空器飞行时敏感数据得到安全高效传输，本文对 AES 加密算法进行改进与优化，提出了基于轻量型 AES 加密算法的浮空器平台数据传输方案，通过减少轮函数的循环次数和将行移位变换改为列移位变换以实现轻量型 AES 加密算法，减少数据加解密过程中消耗的时间。经过实验验证，该方案在保证尽可能高的算法安全性的同时提高了数据加解密速度。

1 典型浮空器平台数据安全传输需求

平流层浮空器平台的数据传输系统以无线传输方式作为其通信媒介，实现浮空器平台与地面监控中心的点对点双向通信，传输模型如图 1 所示。其中，浮空器作为被监控对象，由处理器与各类传感器节点和执行机构节点共同组成平台嵌入式系统，完成平台下行数据的采集与处理；地面监控中心由地面上位机软件组成，负责指令的发送以及实时显示各类飞行数据，完成人机交互；无线传输模块由浮空器平台传输终端和地面传输终端两部分组成，二者通过特定的 ID 进行身份识别，完成平台与地面之间的数据传输。

本文提出的数据传输方案通过在平台嵌入式系统和上位机软件中内置轻量型 AES 加密算法，实现平台下行数据和上行指令的加密和解密。以保证即使关键数据在远距离无线传输过程中被恶意截取，对方在没有密钥的情况下也难以破解密文获取到有用的信息，从而实现浮空器平台的安全性，方案整体设计如图 2 所示。

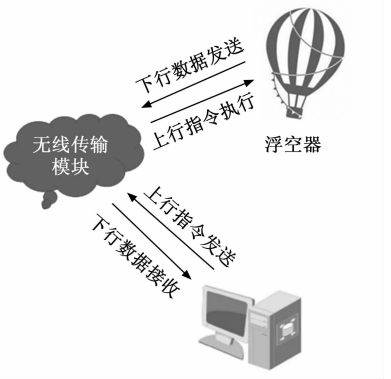


图 1 浮空器平台数据传输模型

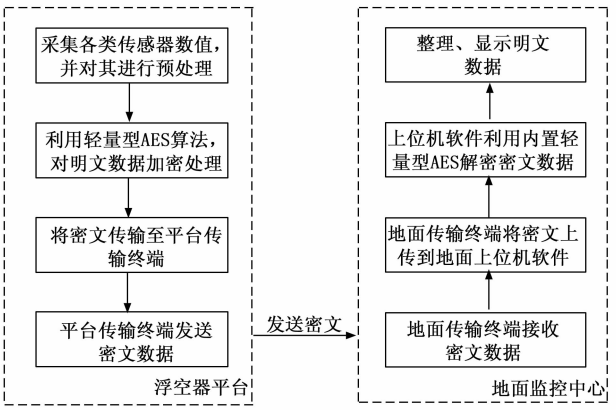


图 2 方案整体设计图

一方面，对于浮空器下行数据而言：首先，平台在自身嵌入式系统中内置轻量型 AES 加密模块，对采集到的各项下行数据进行加密处理，通过串口传输将密文数据发送到平台传输终端；其次，平台传输终端按照既定的通信协议将密文数据进行打包发送到地面传输终端；最后，地面传输终端通过串口将接收到的密文发送到上位机软件，利用上位机软件内置的轻量型 AES 加密算法完成数据解密处理，并实现各项下行数据的实时显示以及在线监测平台状态等功能。

另一方面，对于地面监控中心上行指令而言：首先，通过上位机软件内置的轻量型 AES 模块对上行指令进行加密处理，并以密文形式传输至地面无线传输终端；其次，地面无线传输终端按照既定的通信协议将密文指令进行打包发送到平台无线传输终端；最后，平台无线传输终端通过串口将密文指令发送到平台处理器，处理器根据内置的轻量型 AES 加密算法解密出原有指令内容，并实施相应动作，从而实现地面远距离对高空平台的控制。

2 轻量型 AES 加密算法

2.1 轻量型 AES 加密算法原理

2.1.1 原理介绍

高级加密标准（AES，advanced encryption standard）

由美国国家标准与技术研究院在 2001 年发布，是目前对称加密算法中研究热点之一。AES 加密算法采用分组加密的方式，分组长度固定为 128 位，即 16 字节[12-13]。算法通常有 3 种密钥长度，即 128 比特、192 比特和 256 比特，不同的密钥长度对应不同的加密轮数，如表 1 所示。

表 1　AES 加密算法参数

AES 加密算法	密钥长度(32 bit)	分组长度(32 bit)	加密轮数
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES 加密算法利用混淆和代换的工作原理，将明文数据元素代换成为另一个不相关的元素，并对明文数据元素进行重新排列和组合，从而生成密文。以 AES-128 加密算法为例：首先，在初始阶段利用混淆的原理，对 128 位初始密钥经过十轮密钥扩展操作以生成后续需要的轮密钥；其次，进行十轮算法迭代，前九轮的迭代依次按照字节代换、行移位、列混淆以及轮密钥加操作进行，第十的轮迭代只进行字节代换、行移位以及轮密钥加操作，不进行列混合操作；最后，输出大小同样为 128 位的密文。算法在每轮迭代过程中进行的 4 个可逆变换统称为轮函数，它是 AES 加密算法实现高安全度的核心部分。

本文提出的轻量型 AES 加密算法在 AES-128 算法的基础上做出改进。在保证算法安全性的前提下，通过寻找轮函数循环的局部最优次数和将状态矩阵行移位变换改为列移位变换，即通过轮函数优化和行移位优化两个操作对 AES 加密算法进行改进，从而在保证算法安全性的同时提高算法加解密速度。

2.1.2　轮函数优化

轮函数迭代是保证加密算法安全性最重要的步骤，随着轮函数循环次数的增加，算法抵御网络攻击的能力也随之增强[14]。然而，随着轮函数循环次数增加的同时，算法的计算量也会随之变大，从而导致算法的运行速度降低。因此，轻量型 AES 加密算法旨在寻找既能够最大限度保证算法安全性，又可以提升算法运行速度的轮函数循环的局部最优次数，以达到平衡算法的安全性与运行速度的目的。

轮密钥扩展的发散程度对选取可行的轮函数循环次数具有一定的参考意义[15]。根据 AES 加密算法的原理可知每轮密钥扩展后的比特差（bit difference，不同数据相同位相异的总和）越大，安全性越高。为了进一步明确轮函数循环的局部最优次数，该实验利用线性同余法随机生成了 100 个初始密钥及 100 个与初始密钥相差一位比特的对比密钥，共同构成 100 组密钥对；通过计算 100 组密钥对经过十轮密钥扩展后每轮比特差的变化情况，以选取轮函数循环的局部最优次数具体流程如图 3 所示。

1）构建初始密钥矩阵：利用线性同余法随机生成 100×16 的二维初始密钥矩阵 $\boldsymbol{A}$，记作 $\boldsymbol{A}=[100][16]$，矩阵的各行代表初始密钥，各列代表密钥中的各个元素。线性同余方法是目前应用广泛的伪随机数生成算法之一[16]。一般形式为：对任意初始值 X_0，满足 $0\leqslant X_0<M$，随机数序列由如下递推公式（1）确定：

$$X[i]=(A^{*}X[i-1]+C)\bmod(M) \tag{1}$$

式中，M 为模数并且 $M>0$；A 为乘子且 $0<A<M$；C 为增量且 $0\leqslant C<M$。取 $M=2^{32}$，$A=2\ 140\ 13$，$C=2\ 531\ 011$，X_0 取系统当前时间值。

2）生成对比密钥矩阵：通过判断初始密钥矩阵 $\boldsymbol{A}$ 中某列元素的奇偶性，若该元素为偶数，对其进行加一操作；反之，对其进行减一操作，以确定与初始密钥相差一位比特的二维对比密钥矩阵 $\boldsymbol{B}$，记作 $\boldsymbol{B}=[100][16]$。

3）执行密钥扩展：对初始密钥矩阵 $\boldsymbol{A}$ 和对比密钥矩阵 $\boldsymbol{B}$ 分别进行十轮密钥扩展，并将结果存储在四维矩阵 $\boldsymbol{W}[100][11][4][4]$ 和 $\boldsymbol{V}[100][11][4][4]$ 之中。密钥扩展将 128 位的原密钥转换为 4 * 4 矩阵形式，并将矩阵中的列向量存储在 $\boldsymbol{W}[0]-\boldsymbol{W}[3]$ 中；基于 $\boldsymbol{W}[0]-\boldsymbol{W}[3]$ 通过公式（2）和（3）依次求解出 $\boldsymbol{W}[j]$，生成十个子密钥。

$$\boldsymbol{W}[4i]=\boldsymbol{W}[4(i-1)]+G(\boldsymbol{W}[4i-1]),1\leqslant i\leqslant 10 \tag{2}$$

$$\boldsymbol{W}[4i+j]=\boldsymbol{W}[4(i-1)+j]+\boldsymbol{W}[4i-1+j]$$
$$1\leqslant i\leqslant 10,0\leqslant j\leqslant 3 \tag{3}$$

公式（3）中引入的 G 函数对列向量 $\boldsymbol{W}$ 重新排列后执行 S 盒代换，并用第一个字节与轮系数进行异或运算，生成新

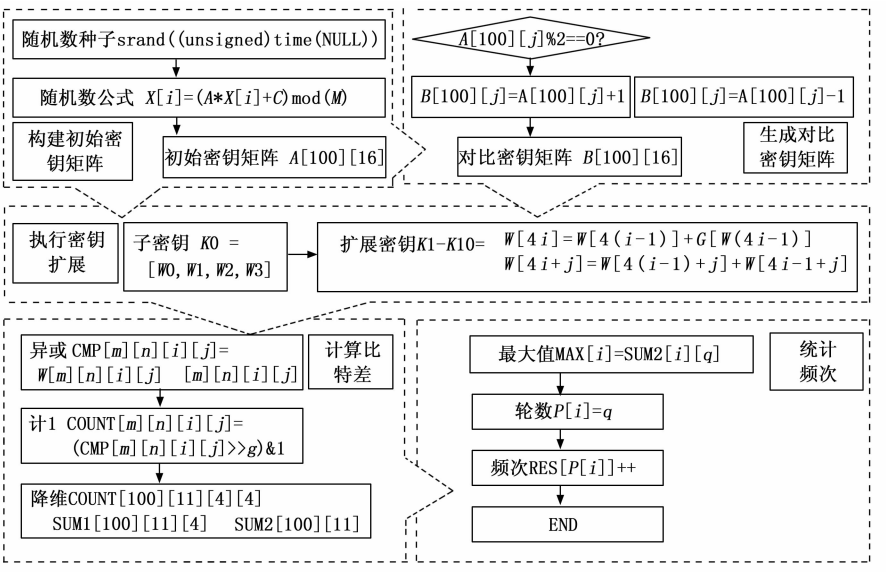


图 3　轮函数优化流程

的列向量 W'。一方面可以增强扩展密钥的非线性，提高密钥安全性；另一方面，可以消除算法中的对称性，使得密文数据更不容易被破解。

4）计算比特差：首先，将 W 和 V 矩阵中对应元素进行异或得到矩阵 CMP [100] [11] [4] [4]，该步骤的目的是得到两个矩阵中每个对应元素之间相差的比特数；其次，统计 CMP [100] [11] [4] [4] 各个元素中为“1”的位数得到矩阵 COUNT [100] [11] [4] [4]，即 W 和 V 矩阵中对应元素的比特差；最后，通过对四维矩阵 $COUNT$ [100] [11] [4] [4] 进行降维操作，最终得到 SUM2 [100] [11] 矩阵，该矩阵行值代表 100 组密钥，列值代表每组密钥经过密钥扩展后的比特差。

5）统计频次：得到的二维矩阵 $SUM2$ [100][11]后，找到矩阵每行元素中的最大值及其所在列数，即可得到 100 组密钥经过密钥扩展后各轮的比特差峰值情况。

为了选取更多的实验样本，该实验将上述步骤重复进行 10 次，即通过总计 1 000 组密钥经过各轮密钥扩展后比特差的变化情况以确定轮函数循环的局部最优次数，最终实验结果如图 4 所示。

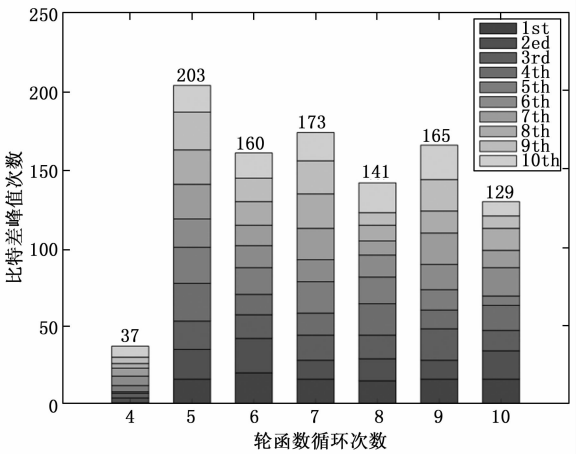


图 4 各轮密钥扩展后比特差峰值图

从图 4 中可知，1 000 组密钥在最初只相差一位比特差的情况下，在第四轮密钥扩展后开始出现比特差峰值，即此时密钥的混淆程度最大，说明经过前三轮密钥扩展后密钥各位已基本得到完全扩散；从第五轮密钥扩散之后，各轮的比特差峰值次数基本维持在 200 次左右，并且在第五轮密钥扩展后比特差峰值次数达到最大，占整体试验次数的 20.3%，并在第七轮密钥扩展后比特差峰值次数达到了第二次小高峰，占整体实验次数的 17.3%。

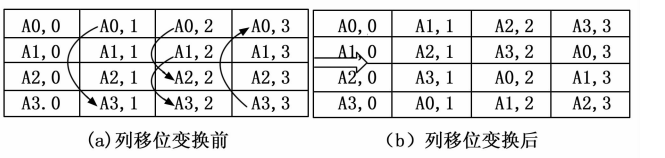
J. Daemen[17] 指出可以从加密算法抗击捷径攻击的能力出发考虑算法的轮函数的循环次数，并且证实对于 AES-128 加密算法而言，目前尚未发现能够对具有轮函数循环六次以上的简化版本实施的捷径攻击方法。综上分析，本文将该理论研究与实验结果相结合，选取前七轮作为轻量型 AES 加密算法的轮函数循环的局部最优次数。

2.1.3 行移位优化

轻量型 AES 加密算法将 AES 加密算法中的行移位变换改为列移位变换，其执行方式与行移位基本相同，目的在于打破 AES 加密算法常规的执行方式以优化算法结构并提升算法安全性，从而使密文数据更难被破解。列移位变换是一种线性变换，通过公式（4）有规则地对状态矩阵进行字节移动实现：

$$\begin{bmatrix} B_{j,0} \\ B_{j,1} \\ B_{j,2} \\ B_{j,3} \end{bmatrix} = \begin{bmatrix} A_{j,0} \\ A_{(j+1)\bmod Nb,1} \\ A_{(j+2)\bmod Nb,2} \\ A_{(j+3)\bmod Nb,3} \end{bmatrix} \tag{4}$$

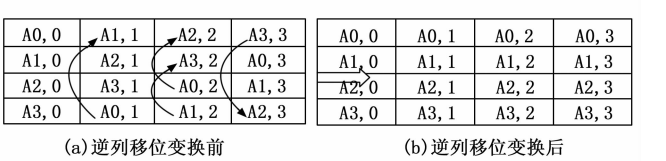
式中，Nb 取 4。状态矩阵由 4 * 4 个字节构成，在加密过程中，保持矩阵的第一列不变，将第二行向下移动一个字节、第三行向下移动二个字节、第四行向上移动一个字节，如图 3 所示。列移位处理后，得到的状态矩阵如图 5 所示。



(a)列移位变换前 (b）列移位变换后

图 5 状态矩阵图

解密时：保持矩阵的第一列不变，第二行向上移动一个字节、第三行向上移动二个字节、第四行向下移动一个字节，如 5 五所示。列移位处理后，得到的状态矩阵如图 6 所示。



(a)逆列移位变换前 (b)逆列移位变换后

图 6 状态矩阵图

2.2 轻量型 AES 加密算法设计

轻量型 AES 加密算法在 AES-128 算法的基础上，对 AES-128 的处理速度和效率进行优化提高。该算法采用 7 次轮函数循环，除最后一轮外，前六轮次都包含字节代换（SubByte）、列位移变换（ColumnRow）、列混合（MixColumn）、轮密钥加（AddRoundKey）4 个处理步骤，另外，最后一轮不包含列混合操作，算法具体实现流程如图 7 所示。

2.2.1 字节代换

字节代换作为轻量型 AES 加密算法的第一步，是整个算法中唯一的非线性变换，关键在于 S 盒的构造。S 盒是一个具有很强代数结构的 16×16 矩阵[19,18]，设计中包含 3 个步骤：首先，初始化矩阵的第一行是 {00}， {01} … {0F}，第二行是 {10}，{11} … {1F}，其次，将矩阵中每个元素的值代换为其在有限域 GF（2^8）上的逆乘法；最后，利用公式（5）对每个元素执行仿射变换：

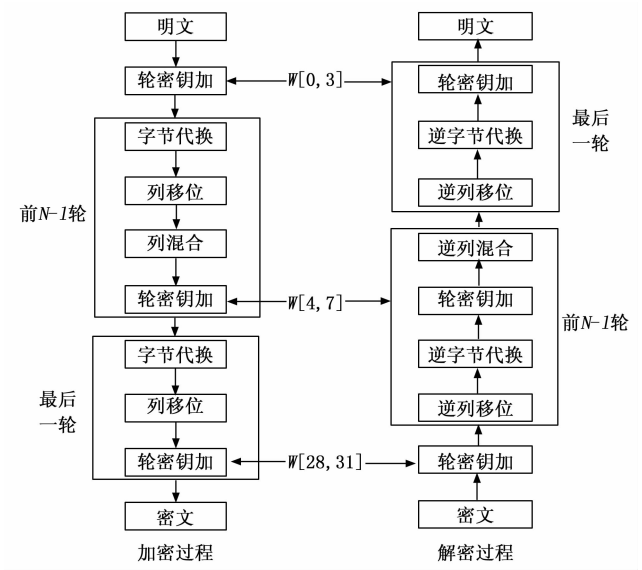


图 7　轻量型 AES 加密算法流程图

$$\begin{bmatrix}C_7\\C_6\\C_5\\C_4\\C_3\\C_2\\C_1\\C_0\end{bmatrix}=\begin{bmatrix}1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\\1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\end{bmatrix}\begin{bmatrix}B_7\\B_6\\B_5\\B_4\\B_3\\B_2\\B_1\\B_0\end{bmatrix}\oplus\begin{bmatrix}0\\1\\1\\0\\0\\0\\1\\1\end{bmatrix}\tag{5}$$

在加密过程中，字节代换是将状态矩阵中的每个字节代换为 S 盒矩阵中的另一个字节。取状态矩阵中每个字节的高四位作为 S 盒行值，低四位作为 S 盒列值，根据行值和列值确定该字节所对应的 S 盒中的字节，并用 S 盒中该位置的字节代换状态矩阵中的元素，作为新的字节输出；在解密过程中，字节代换的逆运算为逆字节代换。逆 S 盒的生成与 S 盒生成步骤相反，初始化 S 盒进行逆仿射变换后，对每个字节在有限域上取乘法逆运算，完成逆字节代换步骤并输出结果。

2.2.2　列移位

列移位操作是对状态矩阵中各个元素进行重新排列以打破原状态矩阵中的元素之间的统计关系。列移位是整个轻量型 AES 加密算法中独具特色的操作，利用列移位变换来代替原 AES 加密算法中的行移位变化，使得黑客攻击难以预测算法正在执行的操作方式。列移位变换的执行方式与行移位相同，并在此基础上做了一些改进，具体操作与 2.1.2 行移位优化章节一致。

2.2.3　列混合

列混合变换是数据混淆的关键步骤，该步骤将列移位后的状态矩阵与常数矩阵相乘，得到混淆后新的状态矩阵，运算过程如公式（6）所示：

$$\begin{bmatrix}S'_{0,0}&S'_{0,1}&S'_{0,2}&S'_{0,3}\\S'_{1,0}&S'_{1,1}&S'_{1,2}&S'_{1,3}\\S'_{2,0}&S'_{2,1}&S'_{2,2}&S'_{2,3}\\S'_{3,0}&S'_{3,1}&S'_{3,2}&S'_{3,3}\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&02\end{bmatrix}\begin{bmatrix}S_{0,0}&S_{0,1}&S_{0,2}&S_{0,3}\\S_{1,0}&S_{1,1}&S_{1,2}&S_{1,3}\\S_{2,0}&S_{2,1}&S_{2,2}&S_{2,3}\\S_{3,0}&S_{3,1}&S_{3,2}&S_{3,3}\end{bmatrix}\tag{6}$$

在具体计算中，列混合变换可视为将原状态矩阵中的每一列与常数矩阵相乘，得到与原状态矩阵对应的新状态矩阵，即将原状态矩阵中每列所有元素的加权和赋值给新状态矩阵中对应的元素，从而输入的每个字节都会影响到输出的 4 个字节，第 j（$0\leqslant j<3$）列可表示为：

$$\begin{aligned}S'_{0,j}&=(2*S_{0,j})\oplus(3*S_{1,j})\oplus S_{2,j}\oplus S_{3,j}\\S'_{1,j}&=S_{0,j}\oplus(2*S_{1,j})\oplus(3*S_{2,j})\oplus S_{3,j}\\S'_{2,j}&=S_{0,j}\oplus S_{1,j}\oplus(2*S_{2,j})\oplus(3*S_{3,j})\\S'_{3,j}&=(3*S_{0,j})\oplus S_{1,j}\oplus S_{2,j}\oplus(2*S_{3,j})\end{aligned}\tag{7}$$

列混合变换的逆变换为列混合变换的逆。逆变换是将矩阵中的每一列与常数矩阵相乘，该常数矩阵与列混合变换中的常数矩阵互为逆矩阵，从而便可恢复出原状态矩阵中的元素值，逆列混合变换的计算过程如公式（8）所示：

$$\begin{bmatrix}S'_{0,0}&S'_{0,1}&S'_{0,2}&S'_{0,3}\\S'_{1,0}&S'_{1,1}&S'_{1,2}&S'_{1,3}\\S'_{2,0}&S'_{2,1}&S'_{2,2}&S'_{2,3}\\S'_{3,0}&S'_{3,1}&S'_{3,2}&S'_{3,3}\end{bmatrix}=\begin{bmatrix}0E&0B&0D&09\\09&0E&0B&0D\\0D&09&0E&0B\\0B&0D&09&0E\end{bmatrix}\begin{bmatrix}S_{0,0}&S_{0,1}&S_{0,2}&S_{0,3}\\S_{1,0}&S_{1,1}&S_{1,2}&S_{1,3}\\S_{2,0}&S_{2,1}&S_{2,2}&S_{2,3}\\S_{3,0}&S_{3,1}&S_{3,2}&S_{3,3}\end{bmatrix}\tag{8}$$

2.2.4　轮密钥加

在轮函数转换中，每轮将生成一个通过密钥扩展获得的新密钥。轮密钥加是将 128 位新生成的轮密钥与状态矩阵中的数据进行逐位异或操作，轮密钥加操作的计算如公式（9）所示：

$$\begin{bmatrix}E_{0,j}\\E_{1,j}\\E_{2,j}\\E_{3,j}\end{bmatrix}=\begin{bmatrix}S_{0,j}\\S_{1,j}\\S_{2,j}\\S_{3,j}\end{bmatrix}\oplus\begin{bmatrix}K_{0,j}\\K_{1,j}\\K_{2,j}\\K_{3,j}\end{bmatrix}\tag{9}$$

由于异或运算的逆操作是其自身，因此轮密钥加的逆运算与轮密钥加的运算过程一致。

3　实验验证与分析

为了验证所提出的轻量型 AES 加密算法应用于浮空器平台数据传输的安全性和高效性，实验使用 C 语言在 KEIL5 软件中编写 AES 加密算法和轻量型 AES 加密算法，并在主频为 168 MHz 、FLASH 内存为 1 M、基于 Cortex-M4 内核的 STM32F407 处理器上实现算法的模拟和仿真。实验选取 2021 年浮空器平台的飞行数据作为加密算法的明

文测试数据源。首先，对输入的浮空器平台明文数据进行字节填充和矩阵旋转两步预处理操作；其次，基于 STM32 嵌入式处理器利用轻量型 AES 加密算法对不同大小明文数据源进行加解密实验并得到各个数据源的传输结果；最后，从算法安全性和运行速度两方面对 AES 加密算法与轻量型 AES 加密算法进行对比分析。

3.1 数据预处理

在执行轻量型 AES 加密算法之前，需对明文数据进行预处理操作以减少明文数据之间的统计关系，增强明文数据的扩散程度。预处理的流程包括字节填充和明文矩阵旋转。

Step1：字节填充

字节填充是对明文数据处理的第一个步骤，填充的目的是确保明文数据为分组长度 16 字节的整数倍，以保证明文数据可全部转换为 4 * 4 字节的矩阵。字节填充的具体步骤如图 8 所示。

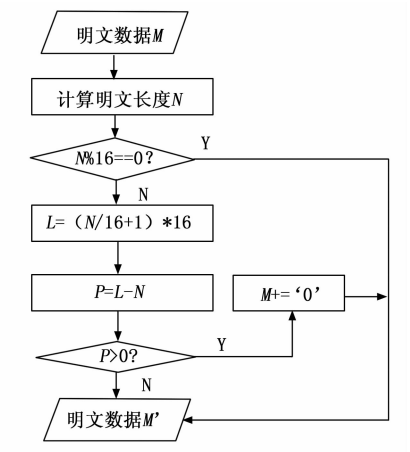


图 8 字节填充流程图

在进行字节填充流程中，首先计算得到待处理的明文数据的长度；其次，判断明文数据长度是否为 16 字节的整数倍，如若明文数据长度不足为 16 字节的整数倍，则用 0 将数据长度补齐为 16 字节整数倍；最后，输出完成字节填充后的明文数据。

Step2：明文矩阵旋转

预处理步骤的第二步是将完成字节填充流程后的明文数据分为若干 4 * 4 字节大小的矩阵，并依次对各个矩阵逆时针旋转 90°以进一步增强明文数据内部的混淆程度。该步骤的输入为若干 4 * 4 字节大小的明文矩阵。矩阵旋转是对明文数据内部的字节进行重新排列，不仅可以打破原有明文数据间的统计关系，还能够补偿后续在减少轮函数循环次数过程中所削弱的算法安全性。明文矩阵逆时针旋转 90° 的操作如图 9 所示。

3.2 实验结果

在实验过程中，选取 10 kB、20 kB、30 kB、40 kB、50 kB、60 kB 六种不同大小的数据源利用轻量型 AES 加密算法分别对各个数据源进行十次处理后，取 10 次结果的平均值作为实验结果。具体实验结果详见表 2。

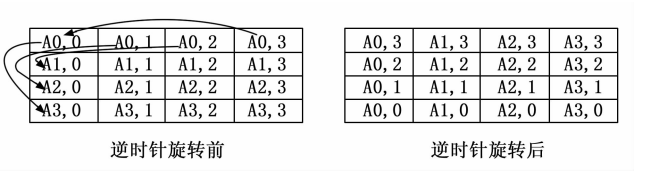


图 9 逆时针旋转操作图

表 2 加密算法实验测试结果

数据大小/kB	轻量型 AES 加密算法加密完整率/%
10	100
20	100
30	100
40	99.4
50	99.0
60	98.3

由表 2 可知，经过轻量型 AES 加密算法处理后的数据与明文相比，其准确率随着明文数据源的增大逐渐减小：对 30 kB 以内的小数据量明文数据源进行传输时完整实现了加解密操作，成功保证了数据准确无误传输；由于受到嵌入式系统处理器性能的限制，在对数据量为 40～60 kB 大小的数据源传输时存在误差，数据加密完整率平均达到 99% 左右。综上分析，本文提出的轻量型 AES 加密算法能够保证各个大小的数据源稳定完整传输，适合应用于浮空器平台数据传输。

3.3 性能分析

3.3.1 统计分析

在明文数据源中存在的重复字节可能会泄露某些重要信息，因此密文数据必须具有均匀分布的字节以保证数据的安全性。为了测试轻量型 AES 加密算法的加密效果，统计了各个数据源在加密前后数据内容的变化，以描述明文和密文中的字节分布情况[20]，如图 10 和图 11 所示，图 10 中明文数据的字节分布出现了明显的峰值，这些峰值反映了明文数据中重要数据信息的细节；而经过加密处理后的

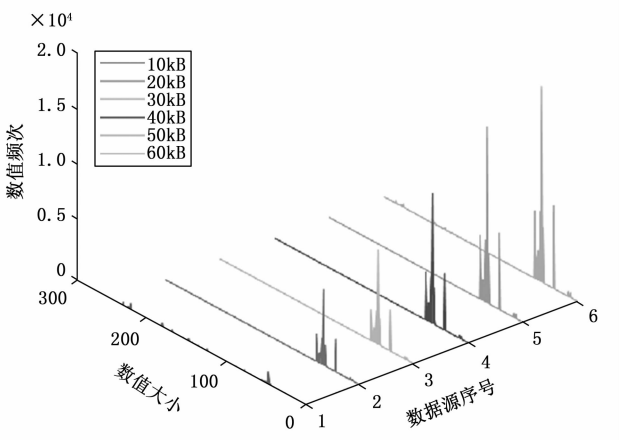


图 10 明文数据分布直方图

密文数据的字节分布图 11 呈现无规律的均匀分布，有效地隐藏了明文数据源中的统计信息，说明轻量型 AES 加密算法可以有效地抵御基于数据统计的攻击。

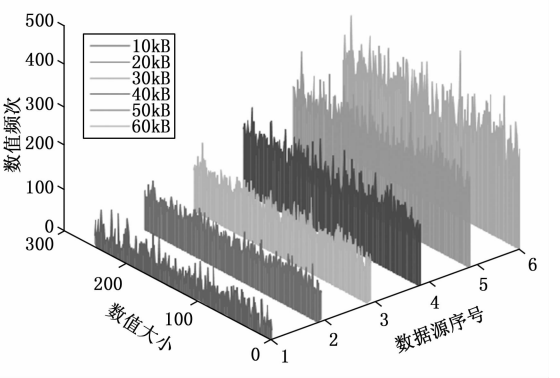


图 11　密文数据分布直方图

3.3.2　信息熵

信息熵一种衡量数据随机性水平的指标，用来表示数据的混乱程度。熵值越大表明数据越混乱所包含的有效信息越少，信息熵计算公式如下[21]：

$$H(m) = -\sum_{i=0}^{2^n-1} p(m_i)\log p(m_i) \tag{10}$$

式中，$p(m_i)$ 表示数值为 m_i 出现的概率。由于密文数据值以字节为单位进行表示，所以其取值范围为 [0，255]，对应的理想信息熵值为 8。表 3 列出了对不同大小数据源使用 AES 加密算法和轻量型 AES 加密算法加密后的密文信息熵及其二者之间的熵差（信息熵差值的绝对值）。从表中可以看出，AES 加密算法和轻量型 AES 加密算法加密后二者的密文信息熵相近皆与理想信息熵值接近，并且最大熵差小于 0.01，说明本文提出的加密算法能够有效防止明文数据源的信息泄露，可以有效地抵抗基于信息熵的攻击。

3.3.3　运行时间分析

算法加解密运行时间是衡量加密算法性能的重要参数[22]。本文的主要目标是提供一种用于嵌入式微型系统上的加密算法，并且该算法在对浮空器平台上数据进行加密和解密时处理速度更为高效快捷。通过输入不同大小的明文数据源，对 AES 加密算法和轻量型 AES 加密算法各自的加密和解密的时间进行测试与统计，测试结果如表 4 所示。

实验结果表明：在对各个不同大小数据源加解密过程中，轻量型 AES 加密算法的运行时间都低于 AES 加密算法，并且对各个数据源多次测试取平均值后，轻量型 AES 加密算法的加密时间比 AES 加密算法少 46.33 ms，而解密时间则少 50.83 ms，新算法的运行速度更快。

为了进一步分析明文数据源大小对算法运行时间的影响，该实验以数据源的大小（kB）为横坐标，算法运行时间（ms）为纵坐标，绘制了利用轻量型 AES 加密算法和 AES 加密算法对不同大小的数据源执行加密和解密过程中算法运行时间的对比图，如图 12 所示。

表 3　密文信息熵

数据大小/kB	信息熵		
	AES 加密算法	轻量型 AES 加密算法	熵差
10	7.970	7.975	0.005
20	7.979	7.982	0.003
30	7.980	7.972	0.008
40	7.981	7.975	0.006
50	7.985	7.980	0.005
60	7.971	7.976	0.005

表 4　算法运行时间

数据大小/kB	加密时间/ms			解密时间/ms		
	AES 加密算法	轻量型 AES 加密算法	时间差	AES 加密算法	轻量型 AES 加密算法	时间差
10	59	47	12	53	37	16
20	104	78	26	123	95	28
30	149	108	41	178	137	41
40	193	139	54	232	169	63
50	238	172	66	312	237	75
60	426	347	79	454	372	82

其中，轻量型 AES 加密算法和 AES 加密算法在执行加密和解密过程中算法的运行时间皆随数据源大小的增加而上升，但由于轻量型 AES 加密算法减少了轮函数的循环次数，使得每个数据源的加解密时间都有所缩短，并且在数据源小于 40 kB 时，轻量型 AES 加密算法和 AES 加密算法执行加解密过程中所需要的运行时间与数据源大小基本呈线性增长，在数据源大于 40 kB 时，受到处理器性能和资源的限制，轻量型 AES 加密算法和 AES 加密算法执行加解密过程所需要的运行时间皆有所增加。

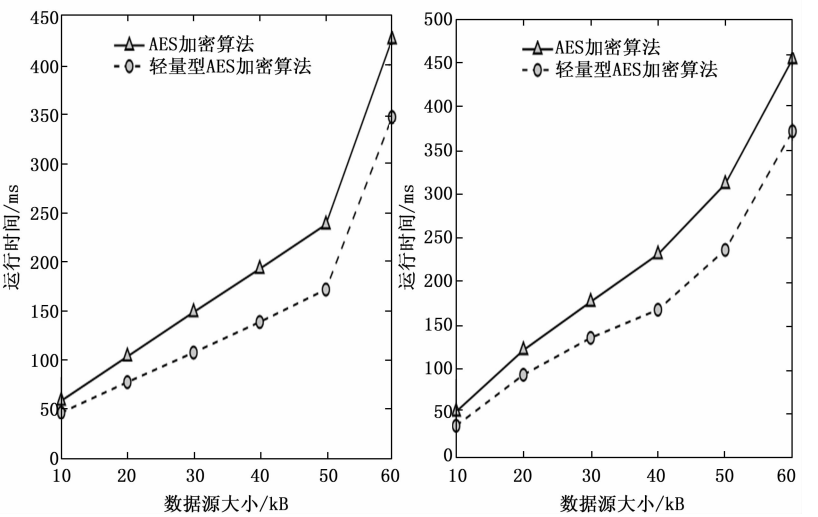


图 12　算法加密过程运行时间

如图 13 所示，以数据源的大小（kB）为横坐标，算法运行时间差（ms）为纵坐标，绘制了轻量型 AES 加密算法和 AES 加密算法在执行加密过程和解密过程中算法运行时间差。当输入的数据源越大时，AES 加密算法比轻量 6 型 AES 加密算法的运行时间越长，二者的运行时间差随着数据源增大而加大。因此，与 AES 加密算法相比，轻量型 AES 加密算法运行速度有所提升。

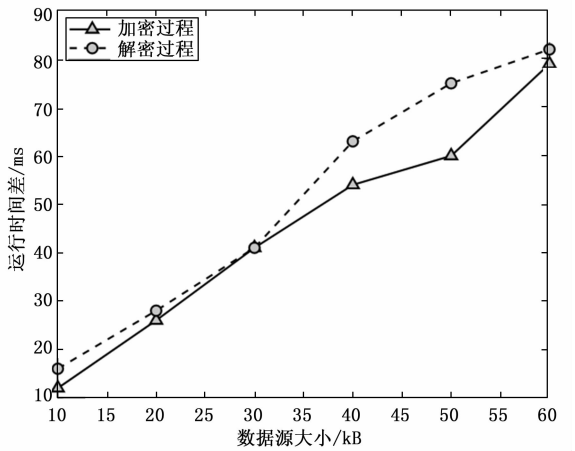


图 13 算法加解密过程运行时间差

4 结束语

针对 AES 加密算法研究应用到球载微型嵌入式系统中难以平衡算法安全性和速度相互制约而引起的无法保证浮空器平台数据实时回传到地面监控终端的问题，提出了一种基于轻量型 AES 加密算法的浮空器平台数据安全传输方案。该方案通过对 AES 加密算法进行改进和优化以保证能够在算法安全性和速度之间取得平衡。一方面，将在执行加密算法之前加入了字节填充和矩阵旋转操作并且行移位变换改为列移位变换以增强算法的安全性；另一方面，减少算法中轮函数的循环次数以提高算法运行速度，从而保证平台数据能够实现实时传输。实验结果表明，该算法实现了不同大小数据源的稳定可靠传输，数据传输完成率可达 98%以上；相比于 AES 加密算法，该加密算法在保障数据安全传输的同时，加密速度提高了 24.5%，解密速度提高了 24.2%，有效地实现了浮空器平台数据可靠、安全且快速传输。

参考文献：

[1] 黄宛宁，张晓军，李智斌，等．临近空间科学技术的发展现状及应用前景 [J]．科技导报，2019，37 (21)：46－62.

[2] 王彦广，王伟志，黄灿林．平流层飞行器技术的最新发展 [J]．航天返回与遥感，2019，40 (2)：1－13.

[3] 李智斌，黄宛宁，张 钊，等．2020 年临近空间科技热点回眸 [J]．科技导报，2021，39 (1)：54－68.

[4] 赵 达，刘东旭，孙康文，等．平流层飞艇研制现状、技术难点及发展趋势 [J]．航空学报，2016，37 (1)：45－56.

[5] 秦慧娴，苗景刚，郝 勇，等．基于物联网技术的浮空器远置终端设计 [J]．计算机测量与控制，2021，29 (12)：156－160.

[6] 闫乐乐，李 辉．基于复合混沌序列的动态密钥 AES 加密算法 [J]．计算机科学，2017，44 (6)：133－138.

[7] ALI H H，SHAKER S H. Modified advanced encryption standard algorithm for fast transmitted data protection [J]. IOP Conference Series：Materials Science and Engineering，2020，928 (3)：32－43.

[8] GANGURDE N，GHOSH S，GIRI A，et al. Ticketing system using AES encryption based QR code [C] //2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)，IEEE，2022：201－206.

[9] 张月华，张新贺，刘鸿雁．AES 算法优化及其在 ARM 上的实现 [J]．计算机应用，2011，31 (6)：1539－1542.

[10] GUPTA M，SINHA A. Enhanced-AES encryption mechanism with S-box splitting for wireless sensor networks [J]. International Journal of Information Technology (Singapore)，Springer Singapore，2021，13 (3)：933－941.

[11] RAHMAN Z，YI X，BILLAH M，et al. Enhancing AES using chaos and logistic map-based key generation technique for securing IoT-based smart home [J]. Electronics，2022，11 (7)：1083－1096.

[12] 南亚会，刘继华，薛艳锋．混沌参数调制下 RSA 数据加密算法研究 [J]．计算机测量与控制，2017，25 (6)：203－206.

[13] KAREEM S M，RAHMA A M S. New method for improving add round key in the advanced encryption standard algorithm [J]. Information Security Journal：A Global Perspective，Taylor & Francis，2021，30 (6)：371－383.

[14] 李升亮．基于 AES 算法的光盘库数据加密技术研究与实现 [D]．武汉：华中科技大学，2015.

[15] 张 治，魏嘉鑫，王 林．LoRa 数据传输网络混合加密设计 [J]．计算机工程与科学，2021，43 (12)：2177－2182.

[16] 王 侃，沈华韵，张 鹏．改进线性同余法随机数发生器 [J]．清华大学学报（自然科学版），2009，49 (2)：191－193.

[17] JOAN D，RIJMEN V. 高级加密标准（AES）算法—Rijndael 的设计 [M]．谷大武译．北京：清华大学出版社，2003.

[18] 谷聚辉，张学毅，周志伟，等．基于 AES 算法汽车门禁系统的设计与研究 [J]．计算机测量与控制，2014，22 (8)：2600－2602.

[19] ZHENG J，ZENG Q. An image encryption algorithm using a dynamic S-box and chaotic maps [J]. Applied Intelligence，Springer US，2022，52 (2)：15703－15717.

[20] PRONIKA P，TYAGIS. Performance analysis of encryption and decryptionalgorithm [J]. Indonesian Journal of Electrical Engineering and Computer Science，2021，23 (2)：10－30.

[21] PREMKUMAR R，ANAND S. Secured and compound 3-D chaos image encryption using hybrid mutation and crossover operator [J]. Multimedia Tools and Applications，2019，78 (8)：9577－9593.

[22] ARIFFIN MOHD N A，AHMED ASHAWESH A Y. Enhanced AES algorithm based on 14 rounds in securing data and minimizing processing time [J]. Journal of Physics：Conference Series，2021，1793 (1)：12－66.