计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(10):289−294 [doi: 10.15888/j.cnki.csa.005888]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

改进的 AES 算法在智慧住区门户中的应用与实现①

谢秀颖[1,2], 王　敏[2], 王少林[1,2], 唐　威[2]

[1](山东建筑大学 信息与电气工程学院, 济南 250101)
[2](山东建筑大学 山东省智能建筑技术重点实验室, 济南 250101)

摘　要: 智慧住区信息门户系统中包含着大量及涉及居民生命财产安全的敏感数据, 为了保证这些数据的保密性, 采用优化的 AES 加密算法对这些数据进行加密, 在保证数据安全的同时, 减少了加密时间, 从而减少了通信延时, 提高了系统的性能. 分析了高级加密标准 AES 的原理和加解密流程, 针对 AES 算法加解密过程耗时相差较大的问题, 在列混合和逆列混合运算时采用有限域 GF(2^8) 上最简形式的矩阵, 减少了解密过程的运算量, 使加解密过程耗时差减少了. 在此基础上对加解密过程进行了合并优化, 在保证加密速度的同时, 减少了算法所占用的存储空间. 在 Visual Studio 2010 平台上, 使用 C 语言实现了几种 AES 优化算法在智能家居中的应用, 结果显示, 所提的优化算法有较高的执行效率, 并占较少的存储空间.

关键词: 智慧住区; AES 算法; 加解密时差; 最简矩阵; 合并优化

引用格式: 谢秀颖,王敏,王少林,唐威.改进的 AES 算法在智慧住区门户中的应用与实现.计算机系统应用,2017,26(10):289–294. http://www.c-s-a.org.cn/1003-3254/5888.html

Application and Implementation of Improved AES Algorithm in Smart Residential Areas Web Portal

XIE Xiu-Ying[1,2], WANG Min[2], WANG Shao-Lin[1,2], TANG Wei[2]

[1](School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China)
[2](Shandong Provincial Key Laboratory of Intelligent Buildings Technology, Shandong Jianzhu University, Jinan 250101, China)

Abstract: Smart residential areas information portal system contains a large number of sensitive data related to the residential life and property security. In order to ensure the confidentiality of the data, optimized AES encryption algorithm is used to encrypt the data to reduce the encryption time, thereby reducing the communication time delay, which improves the performance of the system a lot. The principle, as well as the flow of encryption and decryption of AES algorithm, is analyzed in the following sections. In order to overcome the problem that the time used in decryption is longer than in encryption, a simplest matrix in the Galois field GF(2^8) is going to be adopted in MixColumn and inverse MixColumn operation to reduce the computation and time consumption in decryption. And then based on this, the steps in encryption and decryption process is merged and optimized in some degree, reducing the storage space, at the same time it can keep the execution efficiency. In the Visual Studio 2010 platform, several optimized AES algorithms are implemented in intelligentHome by adopting C language. It shows that the proposed optimized algorithm has higher efficiency and occupies less storage space.

Key words: smart residential areas; advanced encryption standerd(AES); time difference; simplest matrix; merge and optimization

① 基金项目: 山东省墙材革新与建筑节能科研开发项目 (QG021); 2016 年度山东省省级建筑节能与绿色建筑示范项目 (第二批)(鲁建节科函[2016]22 号)
收稿时间: 2016-12-06; 采用时间: 2016-12-26

随着人们物质生活水平的提高, 物联网、云计算、移动互联网等新一代信息技术的发展, 智慧地球、智慧城市和智慧住区等概念先后被提出, 旨在为人民提供更加安全、舒适、便捷的服务[1]. 智慧住区是智慧城市和智慧地球建设的基础, 是集城市管理、公共服务、社会服务、居民自治和互助服务于一体的新技术的应用[2]. 智慧住区是新形势下住区服务管理创新的一种新模式, 为住区居民提供现代化、智慧化的生活环境, 从而形成基于智能化、信息化社会服务与管理的一种全新的住区[3].

智慧住区信息门户是基于 Web 的一种应用程序或信息平台, 可以整合区域人、地、物、情、事、组织和房屋等信息, 统筹公共管理、公共服务和商业服务等资源[4], 通过单一的访问入口, 能够为门户内部和外部的用户提供个性化服务, 方便用户在不同时间地点获得其所需的信息和服务, 从而提升住区治理和管理现代化, 促进公共服务和便民利民服务智能化. 智慧住区信息门户提供智能家居、智慧物业、智慧医疗等应用, 涉及有关居民的各种敏感私密信息, 一旦泄露, 将严重威胁居民的生命财产安全. 如何保证这些信息在信息门户系统中安全, 是智慧住区信息建设中不能避免的问题. 而数据加密就是这样一种技术, 对敏感信息进行加密, 实现信息隐藏, 为敏感信息提供了从发送者到接收者之间最安全的传输方式, 保证了这些敏感信息对接收者以外的人是不可读的[5].

数据加密又分为对称加密和不对称加密, 由于对称加密技术加解密速度快而得到广泛应用. 传统的对称加密技术 DES 密钥长度较短, 其安全性已不能满足分布式开放网络对信息安全的要求, 于是美国国家标准技术研究所 (National institute of standards and technology, 简称 ANSI) 在 1997 年发起了征集 AES (Advanced encryption standard, 简称 AES) 的活动, 寻找一种新的对称加密算法替代当前使用的 DES 加密算法. AES 要求候选加密算法的明文分组长度为 128 bits, 密钥长度为 128、192 或者 256 bits[6]. 经过 3 年多的讨论, Rijndae 脱颖而出, NIST 于 2000 年宣布 Rijndael 将作为新的 AES, AES 的相关参数如表 1 所示, Nk 的值为密钥的位数除以 32.

表 1 AES 参数表

参数名	参数值		
密钥长度Nk	4	6	8
轮数Nr	10	12	14

1 AES 算法研究现状

AES(Advanced encryption standerd, 高级加密标准) 是美国国家标准技术研究所在 2001 年发布的一种迭代对称分组密码算法, 旨在取代 DES 成为广泛使用的标准. 从此, AES 的软硬件实现受到了前所未有的关注[7]. 不同于它的前任标准 DES, Rijndael 使用的是代换-置换网络, 而非 Feistel 架构, 具有高安全性. 在软件实现中, 算法的执行速度和占用内存的大小是是衡量 AES 软件实现方式优劣的重要因素. 因此, 很多人提出了对 AES 不同方面的优化. 如 Santa Clara 大学的 Edward Schaefer 教授和他的学生开发出来的 S-AES[8], 该算法使用明文、密钥和密文均为 16 bits, 加密轮数为 2, 简化了 AES 算法. 文献[9]针对 AES 算法的解密过程比加解密过程耗时长的问题, 分析了加解密过程耗时不对等的原因, 对列混合矩阵进行了改进, 减少了解密运算的时间. 文献[10]对 AES 算法在 matlab 中进行了编程实现, 采用 2 张 256 Bytes 的表格分别存储 S 盒和逆 S 盒的数据, 虽然该方法占用的内存小, 但是实现过程复杂, 加密速度慢. 文献[11]提出采用事先计算好的 8 张 256 个 4 Bytes 表, 通过简单的查表操作和异或运算实现轮变换过程, 提高了加解密的速度. 但是这种方法需要 8 KB 的内存空间, 在一些资源受限的情况下, 这种方法并不实用. 文献[12]又提出了使用 8 张 256 Byets 的表, 利用表格复用技术实现 AES 算法, 使用的存储空间减少了 6 KB, 当分组为 128 bits 时, 执行效率相差不大. 文献[13]对加解密过程进行了合并优化, 并在文献[12]的基础上做了轻量化, 采用 6 张 256 Bytes 的表 (加密过程 2 张, 解密过程使用 4 张), 在保证算法执行速度的同时, 一定程度上减少了所需内存, 但是其解密运算的操做量比加密运算的操作量大了很多, 算法的性能仍有待提高, 不适用于某些要求较高的场合. 本文将在文献[13]的基础上, 采用文献[9]所提出的矩阵对 AES-128 进行优化, 对于矩阵变换过程, 采用有限域上的最简矩阵代替原始矩阵, 减少了解密过程的运算量, 解决了加密耗时不对称的问题, 提高了算法的运行速度, 然后对算法进行轻量化, 减少了算法实现所占的内存, 提高了算法的性能, 使之适用于某些要求较高的场合. 最后以智能家居中数据的加密为例进行说明, 采用优化的加密算法对数据加密, 与文献[13]所采用的方法相比, 算法所占的内存减少了 37.5%, 加密速度快了 14.5%, 解密速度快了 11.3%.

2 AES 算法的理论背景

AES 是标准的分组加密算法, 应用范围广泛. 算法主要由密钥扩展、数据加密和数据解密三大块构成. 算法的加密过程由字节替代、行移位、列混淆和轮密钥加四种变换构成, 解密过程由相应的逆变换构成. 这些所有的操作都是在一个 4×4 的二维字节矩阵上以字节位最小单位进行的, 这个矩阵又称为状态矩阵 (State), 矩阵中的每一个元素是一个 1 字节的十六进制数.

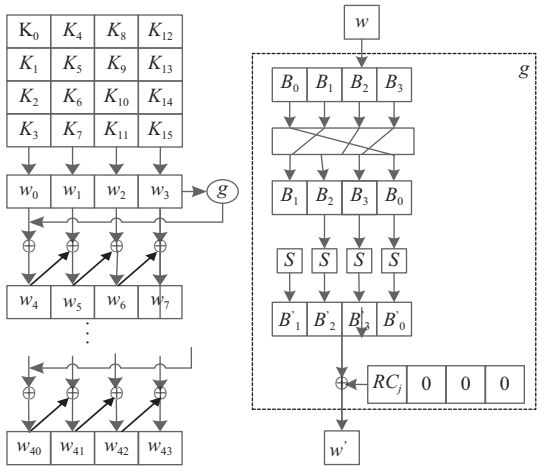
2.1 密钥扩展

密钥扩展过程如图 1(a) 所示, 将输入的一个 128 bits 的密钥扩展成 44 个字组成的扩展密钥数组 $w[i]$ (一维线型数组), 为加解密过程提供轮密钥. 输入密钥直接构成了扩展密钥数组的前 16 个字节, 然后每次使用 4 个新的字填充余下部分, 每一个新增的字由它前面一个字和前面第四个字进行异或操作得到, 当 w 数组中的元素的下表为四的倍数时, 它由前面一字节经过 g 变换然后和前面第四个字进行异或操作得到. g 变换的过程如图 1(b) 所示, 包括一下三种变换:

(1) 循环移位: 将输入的 4 个字节 (一个字) 向右循环移动移位.

(2) 字节替代: 对对步骤 (1) 处理过的每个字节进行 S 盒字节替代.

(3) 将步骤 (2) 的结果与常量 $RC[j]$ 相异或, RC={0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1b, 0x36}, 下标 j 对应相应的加密轮数.



(a) 总体算法 (b) 函数 g

图 1 密钥扩展示意图

2.2 数据加解密

加解密过程如图 2 所示, 由字节代替、行移位、列混淆和轮密钥加四种基本变换构成; 对于加密过程, 算法由轮密钥加变换开始, 经过 10 轮迭代运算, 生成生成相应的密文, 前 9 轮运算由四种基本变换构成, 最后一轮仅进行字节替代、行移位和轮密钥行移位和轮密钥加三种操作. 解密过程和加密过程相似, 只是将加密过程中的操作用相应的逆运算代替.

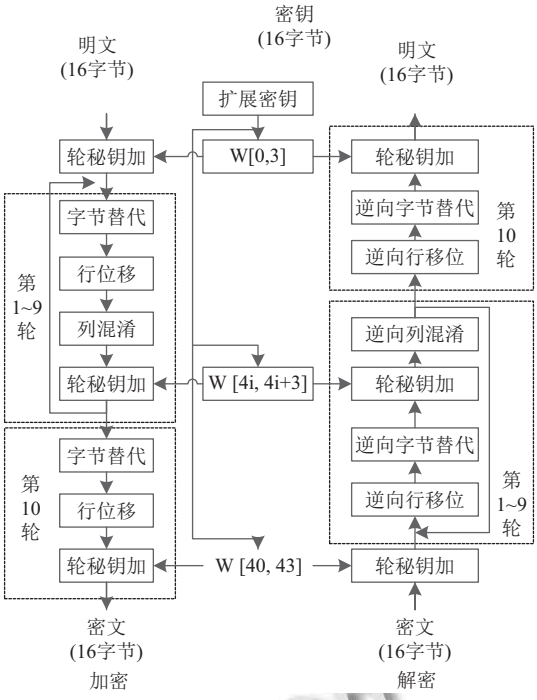


图 2 AES 加密流程图

(1) 字节代替: 分正向字节替代 (SB) 和逆向字节替代 (ISB) 是一个简单的查表操作, 加密时用 S 盒, 解密时使用逆 S 盒, 用 $s'_{i,j}=S\left[s_{i,j}\right]$ 表示正向字节替代变换, 用 $s_{i,j}=S^{-1}\left[s'_{i,j}\right]$ 表示逆向字节代替, S 盒和逆 S 盒的生成见文献[11].

(2) 行移位: 分正向行移位和逆向行移位, 两种移位中状态矩阵的第一行均保持不变, 对于正向行移位, 第二行向左循环移动一位, 第三行向左循环移动两位, 第四行向左循环移动三位, 对于逆向行移位每一行移动的方向与正向行移位相反, 移动位数一样. 对于加密过程正向行移位 (SR) 用式 (1) 表示, 解密过程逆向行移位 (ISR) 用式 (2) 表示, 列下表需要模 4.

$$\begin{pmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{pmatrix} = \begin{pmatrix} s_{0,j} \\ s_{1,j+1} \\ s_{2,j+2} \\ s_{3,j+3} \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} = \begin{pmatrix} s'_{0,j} \\ s'_{1,j-1} \\ s'_{2,j-2} \\ s'_{3,j-3} \end{pmatrix} \tag{2}$$

(3) 列混淆: 分正向列混淆和逆列混淆, 均是对状态矩阵的各列单独操作. 正向列混淆用式 (3) 表示, 逆向列混淆用式 (4) 表示, 两种变换中使用的矩阵是互逆的.

$$\begin{pmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \otimes \begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0E & 0D & 09 & 0E \end{pmatrix} \otimes \begin{pmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{pmatrix} \tag{4}$$

(4) 轮密钥加变换: 不管是正向轮密钥加 (ADK) 变换还是逆向轮密钥加 (IADK) 变换, 都是将状态矩阵和相应的轮密钥进行异或, 其原理是一个数与其本身异或为 0.

3 AES 算法的优化

智慧住区应用服务涉及到居民生活的方方面面, 比如医疗、安防和智能家居等, 信息的泄露将给人们的生活造成极大的困扰. 特别是在智能家居的应用中, 智能家居系统包括医疗保健系统、家庭安防系统和家电控制系统等, 这些都是事关居民生命财产安全的系统, 对门户系统的通信安全和通信延时有非常高的要求. 下文将对 AES 加密算法进行优化, 减少算法加解密的阶执行时间, 同时减少算法所占用的存储空间, 保证算法在条件恶劣的情况下也能执行.

3.1 列混淆矩阵的优化

由第二部分的算法分析可以看出, 正向列混淆变换比逆向列混淆变换所使用的矩阵简单得多, 在做正向列混合变换时需进行 2 次 Xtime 乘法运算和 4 次 Xor 加法运算, 而进行逆向列混合运算时需进行 12 次 Xtime 乘法运算和 9 次 Xor 加法运算[14], 这将导致 AES 算法加密运算与解密运算耗时不对等的问题[9], 文献[9]找出了具有最简形式的列混淆运算和逆列混淆运算, 对应的矩阵如下:

$$B = \begin{pmatrix} 02 & 01 & 03 & 01 \\ 01 & 02 & 01 & 03 \\ 03 & 01 & 02 & 01 \\ 01 & 03 & 01 & 02 \end{pmatrix}$$

可以算出, 矩阵 B 在有限域 GF(2^8) 上的逆矩阵就是它本身, 这样在进行列混合运算时, 加解密运算的简单程度是一样的, 都将只需进行 2 次 Xtime 乘法运算和 4 次 Xor 加法运算.

3.2 加解密过程的合并优化

在加密过程中, 对每一轮运算的过程进行和并, 前 9 轮运算过程的合并如式 5 所示, 这样可以省去 AES 算法软件实现很多的中间环节, 使每一轮的每个输出字节可以一步得到, 由于最后一轮加密没有列混淆, 在合并中我们直接去掉前面的列混淆矩阵即可.

$$\begin{pmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 01 & 03 & 01 \\ 01 & 02 & 01 & 03 \\ 03 & 01 & 02 & 01 \\ 01 & 03 & 01 & 02 \end{pmatrix} \otimes \begin{pmatrix} \mathrm{S}[s_{0,j}] \\ S[s_{1,j+1}] \\ S[s_{2,j+2}] \\ S[s_{3,j+3}] \end{pmatrix} \oplus \begin{pmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{pmatrix} \tag{5}$$

在解密过程中, 对每一轮中逆向行移位、逆向字节替代和轮密钥加三步操作进行和并, 如式 (6) 所示, 在前 9 轮解密中, 再将所得的结果做逆列混淆运算, 在最后 1 轮解密中则无需再做逆列混淆运算.

$$\begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 01 & 03 & 01 \\ 01 & 02 & 01 & 03 \\ 03 & 01 & 02 & 01 \\ 01 & 03 & 01 & 02 \end{pmatrix} \otimes \left(\begin{pmatrix} S^{-1}[s'_{0,j}] \\ S^{-1}[s'_{1,j-1}] \\ S^{-1}[s'_{2,j-2}] \\ S^{-1}[s'_{3,j-3}] \end{pmatrix} \oplus \begin{pmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{pmatrix} \right) \tag{6}$$

3.3 AES 算法的运算优化及轻量化

由式 (5) 可得:

$$\begin{aligned} \begin{pmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{pmatrix} = & 02 \otimes \begin{pmatrix} \mathrm{S}[s_{0,j}] \\ S[s_{1,j+1}] \\ S[s_{2,j+2}] \\ S[s_{3,j+3}] \end{pmatrix} \oplus 01 \otimes \begin{pmatrix} S[s_{1,j+1}] \\ S[s_{2,j+2}] \\ S[s_{3,j+3}] \\ \mathrm{S}[s_{0,j}] \end{pmatrix} \oplus 03 \\ & \otimes \begin{pmatrix} S[s_{2,j+2}] \\ S[s_{3,j+3}] \\ \mathrm{S}[s_{0,j}] \\ S[s_{1,j+1}] \end{pmatrix} \oplus 01 \otimes \begin{pmatrix} S[s_{3,j+3}] \\ \mathrm{S}[s_{0,j}] \\ S[s_{1,j+1}] \\ S[s_{2,j+2}] \end{pmatrix} \oplus \begin{pmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{pmatrix} \end{aligned} \tag{7}$$

对于加密过程定义 2 张表 $T_1[x] = S[x]$ 和 $T_2[x] = 2S[x]$, $T_1[x]$是 S 盒所构成表, $T_2[x]$是 S 盒里面的每个元素在有限域 GF(2^8) 上乘以二所得到的表, 则 $T_3[x] = 3S[x] = T_1[x] \oplus T_1[x]$, 由式 (7) 得, 加密轮函数可以写成:

$$\begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} = 02 \otimes \begin{pmatrix} \mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j} \\ S^{-1}[s'_{1,j-1}] \oplus k_{1,j} \\ S^{-1}[s'_{2,j-2}] \oplus k_{2,j} \\ S^{-1}[s'_{3,j-3}] \oplus k_{3,j} \end{pmatrix} \oplus 01 \otimes \begin{pmatrix} S^{-1}[s'_{1,j-1}] \oplus k_{1,j} \\ S^{-1}[s'_{2,j-2}] \oplus k_{2,j} \\ S^{-1}[s'_{3,j-3}] \oplus k_{3,j} \\ \mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j} \end{pmatrix}$$
$$\oplus 03 \otimes \begin{pmatrix} S^{-1}[s'_{2,j-2}] \oplus k_{2,j} \\ S^{-1}[s'_{3,j-3}] \oplus k_{3,j} \\ \mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j} \\ S^{-1}[s'_{1,j-1}] \oplus k_{1,j} \end{pmatrix} \oplus 01 \otimes \begin{pmatrix} S^{-1}[s'_{3,j-3}] \oplus k_{3,j} \\ \mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j} \\ S^{-1}[s'_{1,j-1}] \oplus k_{1,j} \\ S^{-1}[s'_{2,j-2}] \oplus k_{2,j} \end{pmatrix} \tag{8}$$

定义 2 张表 $T_1'[x]$ 和 $T_2'[x]$, $T_1'[x]$ 是有限域 GF(2^8) 上的所有元素对应的 16 进制数按从小到大排列所构成的表, $T_2'[x]$ 是 $T_1'[x]$ 里面的每个元素在有限域 GF(2^8) 上乘以二所得到的表, 则 $T_3'[x] = 3T_1'[x] = T_1'[x] \oplus T_2'[x]$, 由式 (8) 得, 解密密轮函数可以写成:

$$\begin{pmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{pmatrix} = \begin{pmatrix} T_2'[\mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j}] \\ T_2'[S^{-1}[s'_{1,j-1}] \oplus k_{1,j}] \\ T_2'[S^{-1}[s'_{2,j-2}] \oplus k_{2,j}] \\ T_2'[S^{-1}[s'_{3,j-3}] \oplus k_{3,j}] \end{pmatrix} \oplus \begin{pmatrix} T_1'[S^{-1}[s'_{1,j-1}] \oplus k_{1,j}] \\ T_1'[S^{-1}[s'_{2,j-2}] \oplus k_{2,j}] \\ T_1'[S^{-1}[s'_{3,j-3}] \oplus k_{3,j}] \\ T_1'[S^{-1}[s'_{0,j}] \oplus k_{0,j}] \end{pmatrix}$$
$$\oplus \begin{pmatrix} T_1'[S^{-1}[s'_{2,j-2}] \oplus k_{2,j}] \\ T_1'[S^{-1}[s'_{3,j-3}] \oplus k_{3,j}] \\ T_1'[S^{-1}[s'_{0,j}] \oplus k_{0,j}] \\ T_1'[S^{-1}[s'_{1,j-1}] \oplus k_{1,j}] \end{pmatrix} \oplus \begin{pmatrix} T_2'[S^{-1}[s'_{2,j-2}] \oplus k_{2,j}] \\ T_2'[S^{-1}[s'_{3,j-3}] \oplus k_{3,j}] \\ T_2'[\mathrm{S}^{-1}[s'_{0,j}] \oplus k_{0,j}] \\ T_2'[S^{-1}[s'_{1,j-1}] \oplus k_{1,j}] \end{pmatrix}$$
$$\oplus \begin{pmatrix} T_1'[S^{-1}[s'_{3,j-3}] \oplus k_{3,j}] \\ T_1'[S^{-1}[s'_{0,j}] \oplus k_{0,j}] \\ T_1'[S^{-1}[s'_{1,j-1}] \oplus k_{1,j}] \\ T_1'[S^{-1}[s'_{2,j-2}] \oplus k_{2,j}] \end{pmatrix}$$

经过上面的优化, 在加解密过程中共需要 5 张 256 字节的表 (还有一张存储逆 S 盒的表), 将算法中原来需要在有限域中做乘法运算的列混淆操作变成了简单的查表操作, 大大节省的算法的执行时间.

4 数据加密的实现

选择 Intel i3 3.30 GHz, 2.00 GB RAM PC 机, 对采集到的智能家居中的部分数据进行测试, 以 Windows 7 系统为平台, 使用 Visual Studio 2010 实现几种 AES 加密算法, 测试结果如表 2 所示. 方法一代表文献[10]中才用两张 256 字节的表实现 AES 加密的方法, 方法二表示文献[11]中用 8 张 1kb 的表实现 AES 的方法, 方法三表示文献[12]用八张 256 字节的表实现 AES 的方法, 方法四是本文所采用的方法.

与其它方法相比, 通过优化之后本文所采用的加密方法性能有很大的提高, 相对于方法三, 所占的内存减少了 37.5%, 加密速度快了 14.5%, 解密速度快了 11.3%. 本文在 AES 算法的实现时不但对轮变换过程行了合并, 减少了轮函数中的一些中间转换步骤, 同时将列混淆变换的乘法运算转换成简单的查表运算, 提高了加解密的效率; 而且对列混淆矩阵做了变化, 采用有限域 GF(2^8) 上最简形式的矩阵作为列混淆矩阵, 在简化了逆列混淆矩阵的同时, 简化了逆列混淆运算, 列混合和逆列混合运算耗时相差较大的问题.

表 2 实验结果

实验方法	表占内存(Bytes)	加密速度(Mbps)	解密速度(Mbps)
方法一	512	4.78	3.03
方法二	8192	62.43	61.19
方法三	2048	61.89	61.54
方法四	1280	70.01	68.73

5 总结

在在智慧住区信息门户系统中, 网络安全是一个很重要的问题, 使用高级加密算法加密智慧住区信息门户中的敏感信息, 大大提高了门户系统通信的安全性. 由于在门户系统中部分信息需要快速安全的传输, 本文对 AES 加密算法进行了改进, 采用有限域 GF(2^8) 上最简形式的列混合矩阵, 减少了逆混淆运算的运算量, 在此基础上对 AES 加解密过程进行了合并, 提高了算法的执行速度, 从而减少了通信时传输时数据的处理时间, 减少了延时, 并进行了算法进行了轻量化, 减少了算法所占用的内存空间, 保证了算法在内存环境恶劣的情况下也能应用.

参考文献

1 李宇翔, 费世英, 李端明. 智慧社区系统架构研究. 图书情报工作网刊, 2012, (12): 39–44.

2 张彭, 王轶斌, 沈玉梅, 等. 基于城乡统筹综合信息服务平台构建智慧社区的研究. 中国管理信息化, 2012, 15(6): 83–84.

3 中国城市科学研究会数字城市工程研究中心. 管理创新: 构建智慧社区平台. 建设科技, 2014, (17): 36–40.

4 住房城乡建设部印发《智慧社区建设指南 (试行)》. 城市规划通讯, 2014, (10): 6.

5 Pendli V, Pathuri M, Yandrathi S, *et al*. Improvising performance of advanced encryption standard algorithm. Proc. of the 2016 Second International Conference on Mobile and Secure Services (MobiSecServ). Gainesville, FL, USA. 2016. 1–5.

6 Daemen J, Rijmen V. The block cipher rijndael. Lecture Notes in Computer Science. Berlin, Heidelberg. 2000, 1820: 277–284.

7 Rachh RR, Mohan PVA, Anami BS. Efficient implementations for AES encryption and decryption. Circuits, Systems, and Signal Processing, 2012, 31(5): 1765–1785. [doi: 10.1007/s00034-012-9395-0]

8 Musa MA, Schaefer EF, Wedig S. A simplified aes algorithm and its linear and differential cryptanalyses. Cryptologia, 2003, 27(2): 148–177. [doi: 10.1080/0161-110391891838]

9 肖振久, 胡驰, 姜正涛, 等. AES 与 RSA 算法优化及其混合加密体制. 计算机应用研究, 2014, 31(4): 1189–1194, 1198.

10 Buchholz J J. Matlab implementation of the advanced encryption standard. buchholz.hs-bremen.de/aes/aes.htm, 2001.

11 威廉•斯托林斯. 密码编码学与网络安全: 原理与实践. 6 版. 北京: 电子工业出版社, 2015: 96–130.

12 崔国华, 唐国富, 洪帆. AES 算法的实现研究. 计算机应用研究, 2004, 21(8): 99–101.

13 赵跃华, 马林林. AES 算法的轻量化实现研究. 计算机工程与应用, 2015, 51(6): 79–83.

14 刘文浩, 许春香. 无双线性配对的无证书签密方案. 软件学报, 2011, 22(8): 1918–1926.