密码学实验综合实验一

18374480-黄翔

2021 年 6 月 9 日

1 实验目的

1. 实现 AES 算法的快速实现

2 实验环境

1. C

3 实验内容——AES 算法加速

3.1 算法原理

主要加速实现通过转换为位级表示以及查表法，以及通过宏替换函数加速。

位级表示 通过使用位运算代替 $GF(2^8)$ 域上的运算，提高运算速度，同时较小内存需要

宏替换函数 宏实际为代码块，用之替换函数，减少函数调用与返回的开销

查表法 AES 包含四个对字节矩阵的基本操作：字节替换、行移位、列混淆、轮密钥加。这些运算带来的开销是 AES 运算开销的主要来源。在软件级实现中，我们可以将字节替换、行移位、列混淆统一到一个查找表中，这样每轮只需要差表与轮密钥加操作即可。具体实现如下：
交换字节代换与行移位，我们定义行移位、字节代换、列混合、轮密钥加为

一轮，每轮输入表示为 A，列混合输入为 B，输出为 C，轮输出为 D。对列混淆层，有如下矩阵公式

$$\begin{pmatrix} C_0 & C_4 & C_8 & C_{12} \\ C_1 & C_5 & C_9 & C_{13} \\ C_2 & C_6 & C_{10} & C_{14} \\ C_3 & C_7 & C_{11} & C_{15} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 & B_4 & B_8 & B_{12} \\ B_1 & B_5 & B_9 & B_{13} \\ B_2 & B_6 & B_{10} & B_{14} \\ B_3 & B_7 & B_{11} & B_{15} \end{pmatrix}$$

以第一列为例，有:

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 \\ 01 \\ 01 \\ 02 \end{pmatrix} B_0 + \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix} B_1 + \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix} B_2 + \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix} B_3$$

因此，一轮的运算可以表示为（u 以第一列为例）:

$$\begin{pmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{pmatrix} = \begin{pmatrix} 02 \\ 01 \\ 01 \\ 03 \end{pmatrix} S(A_0) + \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix} S(A_5) + \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix} S(A_{10}) + \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix} S(A_{15}) + W_{k0}$$

我们由此定义 Te 表如下:

$$Te_0(A_x) = \begin{pmatrix} 02 \\ 01 \\ 01 \\ 03 \end{pmatrix} S(A_x) \quad Te_1(A_x) = \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix} S(A_x)$$

$$Te_2(A_x) = \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix} S(A_x) \quad Te_3(A_x) = \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix} S(A_x)$$

由此我们看到，由查找表存储 32 位列混淆输出的所有可能值是可行的，查找表元素个数为 $2^8 = 256$，即每个查找表存储 256 个 32 位字即可每轮操作如下:

$$\begin{pmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{pmatrix} = Te_0(A_0) + Te_1(A_5) + Te_2(A_{10}) + Te_3(A_{15}) + W_{k0}$$

$$\begin{pmatrix} D_4 \\ D_5 \\ D_6 \\ D_7 \end{pmatrix} = Te_0(A_4) + Te_1(A_9) + Te_2(A_{14}) + Te_3(A_3) + W_{k1}$$

$$\begin{pmatrix} D_8 \\ D_9 \\ D_{10} \\ D_{11} \end{pmatrix} = Te_0(A_8) + Te_1(A_{13}) + Te_2(A_2) + Te_3(A_7) + W_{k2}$$

$$\begin{pmatrix} D_{12} \\ D_{13} \\ D_{14} \\ D_{15} \end{pmatrix} = Te_0(A_{12}) + Te_1(A_1) + Te_2(A_6) + Te_3(A_{11}) + W_{k3}$$

同样，我们可以改变解密中密钥加与逆向列混合的顺序，同样可以构成与加法类似的结构（需要对密钥也进行逆向列混合），同理，我们定义：

$$Td_0(A_x) = \begin{pmatrix} 0E \\ 09 \\ 0D \\ 0B \end{pmatrix} S^{-1}(D_x) \quad Td_1(A_x) = \begin{pmatrix} 0B \\ 0E \\ 09 \\ 0D \end{pmatrix} S^{-1}(D_x)$$

$$Td_2(A_x) = \begin{pmatrix} 0D \\ 0B \\ 0E \\ 09 \end{pmatrix} S^{-1}(D_x) \quad Td_3(A_x) = \begin{pmatrix} 09 \\ 0D \\ 0B \\ 0E \end{pmatrix} S^{-1}(D_x)$$

一轮的运算如下：

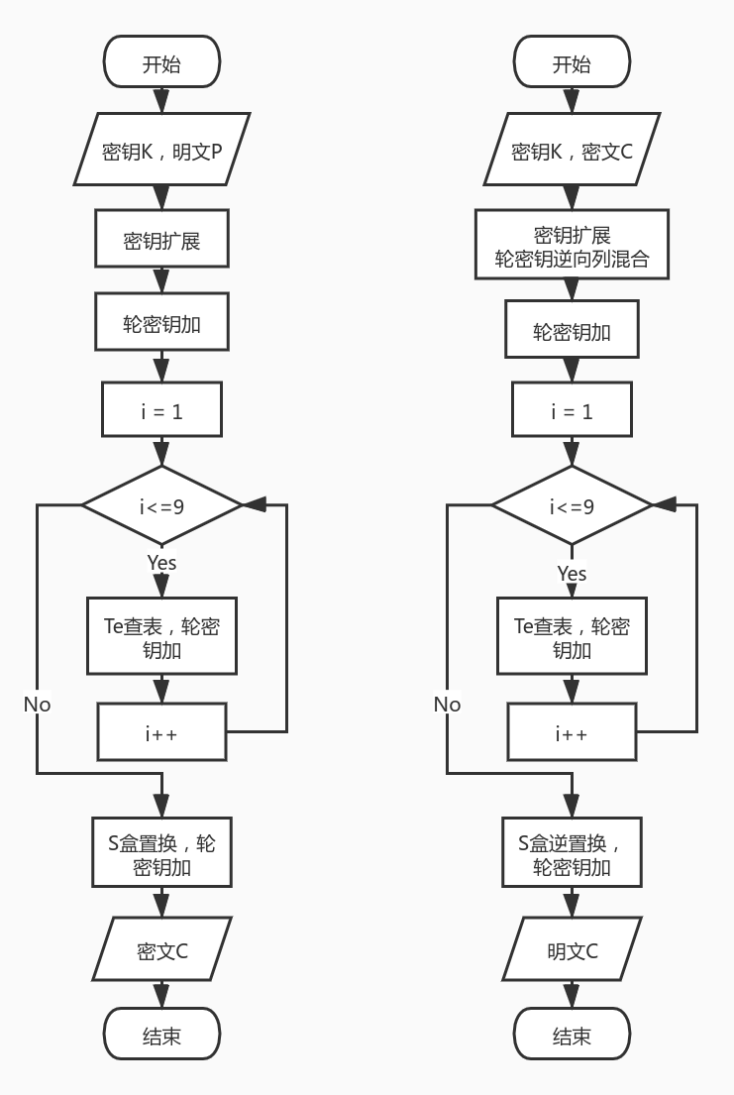
$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix} = Td_0(D_0) + Td_1(D_{13}) + Td_2(D_{10}) + Td_3(D_7) + W_{k0}$$

$$\begin{pmatrix} A_4 \\ A_5 \\ A_6 \\ A_7 \end{pmatrix} = Td_0(D_4) + Td_1(D_1) + Td_2(D_{14}) + Td_3(D_{11}) + W_{k1}$$

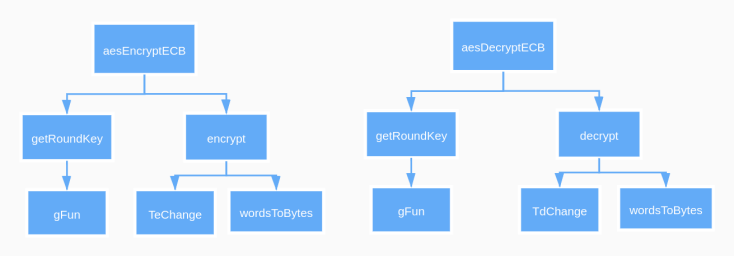
$$\begin{pmatrix} A_8 \\ A_9 \\ A_{10} \\ A_{11} \end{pmatrix} = Td_0(D_8) + Td_1(D_5) + Td_2(D_2) + Td_3(D_{15}) + W_{k2}$$

$$\begin{pmatrix} A_{12} \\ A_{13} \\ A_{14} \\ A_{15} \end{pmatrix} = Td_0(D_{12}) + Td_1(D_9) + Td_2(D_6) + Td_3(D_3) + W_{k3}$$

3.2 算法流程图


开始
密钥K，明文P
密钥扩展
轮密钥加
i = 1
i<=9
Yes
Te查表，轮密钥加
i++
No
S盒置换，轮密钥加
密文C
结束
开始
密钥K，密文C
密钥扩展
轮密钥逆向列混合
轮密钥加
i = 1
i<=9
Yes
Te查表，轮密钥加
i++
No
S盒逆置换，轮密钥加
明文C
结束

3.3 函数调用关系



3.4 算法伪代码

Algorithm 1 优化 AES 加密

Input: 明文 $P(128bits)$，密钥 $K(128bits)$

Output: 密文 $C(128bits)$

$k \leftarrow$ 轮密钥扩展 $getRoundKey(K)$
for $i \leftarrow 0$ to 3 **do**
    $s[4 \times i : 4 \times (i+1)] \leftarrow P[4 \times i : 4 \times (i+1)] \oplus k[i]$
end for
for $i \leftarrow 1$ to 9 **do**
    $r[0] \leftarrow Te_0[s_0] \oplus Te_1[s_5] \oplus Te_2[s_{10}] \oplus Te_3[s_{15}] \oplus k[4 \times i]$
    $r[1] \leftarrow Te_0[s_4] \oplus Te_1[s_9] \oplus Te_2[s_{14}] \oplus Te_3[s_3] \oplus k[4 \times i + 1]$
    $r[2] \leftarrow Te_0[s_8] \oplus Te_1[s_{13}] \oplus Te_2[s_2] \oplus Te_3[s_7] \oplus k[4 \times i + 2]$
    $r[3] \leftarrow Te_0[s_{12}] \oplus Te_1[s_1] \oplus Te_2[s_6] \oplus Te_3[s_{11}] \oplus k[4 \times i + 3]$
    $s \leftarrow r$
end for
$r_0 \leftarrow (sbox[s_0]||sbox[s_5]||sbox[s_{10}||sbox[s_{15}]) \oplus k[40]$
$r_1 \leftarrow (sbox[s_4]||sbox[s_9]||sbox[s_{14}||sbox[s_3]) \oplus k[41]$
$r_2 \leftarrow (sbox[s_8]||sbox[s_{13}]||sbox[s_2||sbox[s_7]) \oplus k[42]$
$r_3 \leftarrow (sbox[s_{12}]||sbox[s_1]||sbox[s_6||sbox[s_{11}]) \oplus k[43]$
$C \leftarrow r_0||r_1||r_2||r_3$
return C

Algorithm 2 优化 AES 解密

Input: 密文 $C(128bits)$，密钥 $K(128bits)$

Output: 明文 $P(128bits)$

$k \leftarrow$ 轮密钥扩展 $getRoundKey(K)$
for $i \leftarrow 4$ to 39 **do**
    $k[i] \leftarrow Td_0[sbox[(k[i] >> 24)\&ff]] \oplus Td_1[sbox[(k[i] >> 16)\&ff]] \oplus Td_2[sbox[(k[i] >> 8)\&ff]] \oplus Td_3[sbox[(k[i]\&ff)]]$
end for
for $i \leftarrow 0$ to 3 **do**
    $s[4 \times i : 4 \times (i+1)] \leftarrow P[4 \times i : 4 \times (i+1)] \oplus k[40+i]$
end for
for $i \leftarrow 9$ to 1 **do**
    $r_0 \leftarrow Td_0[s_0] \oplus Td_1[s_{13}] \oplus Td_2[s_{10} \oplus Td_3[s_7] \oplus k[4 \times i]]$
    $r_1 \leftarrow Td_0[s_4] \oplus Td_1[s_1] \oplus Td_2[s_{14} \oplus Td_3[s_{11}] \oplus k[4 \times i + 1]]$
    $r_2 \leftarrow Td_0[s_8] \oplus Td_1[s_5] \oplus Td_2[s_2 \oplus Td_3[s_{15}] \oplus k[4 \times i + 2]]$
    $r_3 \leftarrow Td_0[s_{12}] \oplus Td_1[s_9] \oplus Td_2[s_6 \oplus Td_3[s_3] \oplus k[4 \times i + 3]]$
end for
$r_0 \leftarrow (invsbox[s_0]||invsbox[s_{13}]||invsbox[s_{10}]||invsbox[s_7]) \oplus k[0]$
$r_1 \leftarrow (invsbox[s_4]||invsbox[s_1]||invsbox[s_{14}]||invsbox[s_{11}]) \oplus k[1]$
$r_2 \leftarrow (invsbox[s_8]||invsbox[s_5]||invsbox[s_2]||invsbox[s_{15}]) \oplus k[2]$
$r_3 \leftarrow (invsbox[s_{12}]||invsbox[s_9]||invsbox[s_6]||invsbox[s_3]) \oplus k[3]$
$P \leftarrow r_0||r_1||r_2||r_3$
return P

3.5 测试样例及结果截图

```
encrypt speed: 3658.724128 Mbits/s
decrypt speed: 2348.438437 Mbits/s
```

设备配置如图：

处理器:	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
已安装的内存(RAM):	16.0 GB (15.9 GB 可用)
系统类型:	64 位操作系统，基于 x64 的处理器

4 总结

本实验实现了 AES 查表法的快速实现，最终在处理器 Intel(R) Core(TM) i7-10750H,RAM 16GB 的机器下，加解密速度达到预约 2.5Gbits/s。查表法的核心思想是将字节代换层、行移位层和列混合层融合为查找表：每个表的

大小是 32 bits 乘 256 项，加密过程四个表 Te，解密过程四个表 Td，这简化了伽罗瓦域上的运算和矩阵乘法操作。当然，由于需要事先存储 10 个表（包含 S 盒与逆 S 盒），查表法的内存要求较大，对于嵌入式等内存有限的设备并不适用。