(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111064562 A
(43)申请公布日 2020.04.24

(21)申请号 201911271815.2

(22)申请日 2019.12.12

(71)申请人 北京计算机技术及应用研究所
地址 100854 北京市海淀区永定路51号

(72)发明人 冯志华 李艳婷 费生波 裴可 罗重 安东博 万星 梁书铭

(74)专利代理机构 中国兵器工业集团公司专利中心 11011
代理人 张然

(51)Int.Cl.
H04L 9/06(2006.01)

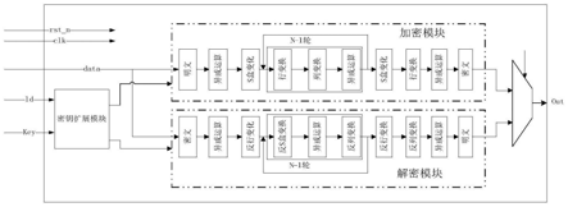
权利要求书3页 说明书8页 附图4页

(54)发明名称

一种FPGA上的AES算法的实现方法

(57)摘要

本发明涉及一种FPGA上的AES算法的实现方法，其中，包括：包括AES密钥扩展模块，AES加密模块；AES解密模块；AES密钥扩展模块主要是为轮变换提供轮密钥并将轮密钥派送至指定轮变换，包括由循环移位模块，S盒变换模块，轮常数赋值模块，扩展密钥数组赋值模块组成；AES加密模块实现明文数据和密钥的加密操作并输出密文，包括由异或运算模块，S盒变换模块，行变化模块，列变化模块组成；AES解密模块实现密文数据和密钥的解密操作并输出明文，包括由反S盒变换模块，反行变化模块，反列变化模块组成。本发明逻辑资源占有量小，性能稳定可靠，模块本身的扩展性高，安全方便的升级方式等优点。



CN 111064562 A

1.一种FPGA上的AES算法的实现方法,其特征在于,包括:

AES加密模块包括:

步骤1:将输入的密文以8bit为一个单位,组成4x4矩阵,

步骤2:将输入的密钥a按从高到底顺序分成4个32bit,分别为WN、WN+1、WN+2以及WN+3,将N、WN+1、WN+2以及WN+3组成的列矩阵与4x4矩阵做异或操作,设N=0;

步骤3:对步骤2得到的结果做S盒替换操作;

步骤4:对步骤3得到的结果做行变换操作,其中第一行不变,第二行循环左移1个字节,第3行循环左移2个字节,第4行循环左移3个字节;

步骤5:对步骤4得到结果做列变换;

步骤6:将步骤5得到的结果与WN+4、WN+5、WN+6以及WN+7异或;

步骤7:将N=N+1,若N=9则继续,否则转回步骤3;

步骤8:将步骤7得到结果经过S盒变化;

步骤9:将步骤8得到结果经过行变化;

步骤10:将W40、W41、W42以及W43异或,得到密文输出;

(2)AES解密模块,包括:

步骤11:将输入的密文以8bit为一个单位,组成4x4矩阵,

步骤12:将W40、W41、W42以及W43组成的列矩阵与步骤11得到的4x4矩阵做异或操作;

步骤13:对步骤12得到的结果做逆行变换操作,即向右偏移,第一行偏移量为0,第二行偏移量为1,第三行偏移量为2,第四行偏移量3;

步骤14:对步骤13得到的结果做逆S盒替换操作;

步骤15:将步骤14得到的结果与W40-4K、W40-4K+1、W40-4K+2以及40-4K+3异或,K=1;

步骤16:对步骤15得到结果做逆列变换;

步骤17:将K=K+1,若K=10则继续,否则转回步骤13;

步骤18:对步骤17得到结果做逆行变换;

步骤19:对步骤18得到结果做逆列变换;

步骤20:将W0、W1、W2以及W3异或,得到明文输出。

2.如权利要求1的FPGA上的AES算法的实现方法,其特征在于,在求取Wi的时候,如果i/4没有余数,Wi的求取过程如下:

a:将Wi-1循环左移一个字节;

b:将步骤a得到的结果经过S盒变换;

c:用i/4得到的商j作为轮常数f(j)的输入,通过查找得出轮常数f(j)的结果,最后与b步骤得到的结果异或;

d:将步骤c得到的结果与Wi-4异或,作为Wi的值;

步骤3:如果k/4有余数,Wk的求解过程如下:

Wk=Wk-1^Wk-4。

3.如权利要求1的FPGA上的AES算法的实现方法,其特征在于,将初始输入的密钥按从高到底顺序分成4个32bit,为[W0、W1、W2、W3]经过密钥扩展模块后,得到44个长度为32bit的子密钥,分别是:[W4、W5、W6、W7]、.....至[W40、W41、W42、W43],参与到后续的加解密步骤中。

4.如权利要求1的FPGA上的AES算法的实现方法，其特征在于，

步骤骤5对步骤4得到结果做列变换包括：

$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$ 为变化前任意一列；$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix}$ 为变化后；对 $\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$ 其进行列变化可以用公式1-1完成描述；

$$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 02 & 02 & 01 & 01 \\ 01 & 02 & 02 & 01 \\ 01 & 01 & 02 & 02 \\ 02 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} + \begin{bmatrix} 00 & 01 & 01 & 01 \\ 01 & 00 & 01 & 01 \\ 01 & 01 & 00 & 01 \\ 01 & 01 & 01 & 00 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad (1\text{-}1);$$

从式1-1中可以得出式1-2：

$$\begin{aligned} s_0' &= 02 \bullet (s_0 \oplus s_1) \oplus s_1 \oplus s_2 \oplus s_3 \\ s_1' &= 02 \bullet (s_1 \oplus s_2) \oplus s_0 \oplus s_2 \oplus s_3 \\ s_2' &= 02 \bullet (s_2 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_3 \\ s_3' &= 02 \bullet (s_0 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_2 \end{aligned} \quad (1\text{-}2);$$

对式(1-2)中的•做出说明：

02•a＝xtime(a)＝{a[6:0],1'b0}^(8'h1b&{8{a[7]}})；

以此将4列全部实现列变化。

5.如权利要求1的FPGA上的AES算法的实现方法，其特征在于，步骤6对步骤5得到结果做逆列变换，包括：

首先，设S0、S1、S2以及S3为状态矩阵的某一列，在解密过程中式(2-1)：

$$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 00 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (2\text{-}1)$$

其可拆分为式(2-2)：

$$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \\ 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

$$=\begin{bmatrix}02&02&01&01\\01&02&02&01\\01&01&02&02\\02&01&01&02\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\\S_3\end{bmatrix}+\begin{bmatrix}00&01&00&00\\00&00&01&00\\00&00&00&01\\01&00&00&00\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\\S_3\end{bmatrix}$$
$$+\begin{bmatrix}08&08&08&08\\08&08&08&08\\08&08&08&08\\08&08&08&08\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\\S_3\end{bmatrix}+\begin{bmatrix}04&00&04&00\\00&04&00&04\\04&00&04&00\\00&04&00&04\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\\S_3\end{bmatrix} \quad (2\text{-}2)$$

其中S′0、S′1、S’2、S′3的表示分别如式(2-3)、(2-4)、(2-5)、(2-6)

$$S_0^{'}=(\{02\}\bullet(S_0\oplus S_1))\oplus(S_1\oplus S_2)\oplus S_3\oplus(\{08\}\bullet(S_0\oplus S_1\oplus S_2\oplus S_3))\oplus(\{04\}\bullet(S_0\oplus S_2)) \quad (2\text{-}3)$$

$$S_1^{'}=S_0\oplus(\{02\}\bullet(S_1\oplus S_2))\oplus(S_3\oplus S_2)\oplus(\{08\}\bullet(S_0\oplus S_1\oplus S_2\oplus S_3))\oplus(\{04\}\bullet(S_1\oplus S_3)) \quad (2\text{-}4)$$

$$S_2^{'}=(S_0\oplus S_1)\oplus(\{02\}\bullet(S_2\oplus S_3))\oplus S_3\oplus(\{08\}\bullet(S_0\oplus S_1\oplus S_2\oplus S_3))\oplus(\{04\}\bullet(S_0\oplus S_2)) \quad (2\text{-}5)$$

$$S_3^{'}=(\{02\}\bullet(S_0\oplus S_3))\oplus(S_1\oplus S_2)\oplus S_0\oplus(\{08\}\bullet(S_0\oplus S_1\oplus S_2\oplus S_3))\oplus(\{04\}\bullet(S_1\oplus S_3)) \quad (2\text{-}6)$$

上式涉及到域GF(2^8)上的常数{02}运算,可以用下述xtime函数来实现,对于任一字节a,有式(2-7):

xtime(a)＝{02}・a＝{a[6:0],1’b0}∧(8’h1b&{8{a[7]}}) (2-7)

式(2-7)xtime(a)中的a为一个8bit的数,由公式可以看出,如果a的最高bit数字是0,其操作过程仅是将a向左移动一个bit,最低位补0,如果最高bit是1,操作过程是将a左移一个bit之后在与十六进制数字1b异或;

式中常数{04}运算可以表达为2个常数乘{02}运算的级联,记常数乘{04}运算为x2time,对于任一字节a,有式(2-8)。同样常数{08}运算可以表达为3个常数乘{02}运算的级联,记常数乘{08}运算为x3time,有式(2-9);

{04}・a＝x2time(a)＝xtime(xtime(a)) (2-8)

{08}・a＝{02・02・02}・a＝x3time(a)＝xtime(xtime(xtime(a))) (2-9)。

一种FPGA上的AES算法的实现方法

技术领域

[0001] 本发明涉及一种AES加解密算法技术，特别涉及一种FPGA上的AES算法的实现方法。

背景技术

[0002] AES算法属于分组密码算法，它的输入分组、输出分组以及K为128、192、256比特。用Nk＝4、6、8代表密钥串的字数(1字＝32比特)用Nr表示对一个数据分组加密的论数，每一轮读需要一个和输入分组具有相同长度的扩展密钥Ke的参与。由于外部输入的加密密钥K长度有限，所以在AES中要用一个密钥扩展程序把外部密钥K在扩展成更长的比特串，以生成各轮的加密密钥。

[0003] 密钥(Key)是密码算法中参与运算的数值(或者数值集)。对报文进行加密，我们需要一个加密算法、一个加密密钥以及明文，并由此产生密文。对报文进行解密，我们需要一个解密算法、一个解密密钥以及密文，并由此复原原始的明文。

[0004] AES加密变换，设X是AES的明文输入，Y是密文输出，则AES的密文Y可以用下面的复合变换表示：$Y = A_{k(r+1)} \circ R \circ S \circ A_{kr} \circ C \circ R \circ S \circ A_{k(r-1)} \circ \ldots\ldots \circ C \circ R \circ S \circ A_{k1}(X)$其中“∘”表示复合运算。这里$A_{ki}$:表示对X的一个变换$A_{ki}(X) = X \oplus Ki$(ki为第i轮的子密钥，为比特串的异或运算)。S:S盒置换。即对每一个字节用S-Box做一个置换。S-Box是一个给定的转换表。R:行置换。C:列置换。$S'(x) = a(x) \otimes s(x)$，这里⊗是特殊的乘法运算。

[0005] AES算法的解密是加密的逆过程，由于AES算法的内部函数都是可逆的，因此解密过程仅仅是将密文作为初始化输入，按照轮子密钥相反的方向对输入的密文再进行加密的过程，该过程加密的最终结果就可以恢复出相应的明文。

发明内容

[0006] 本发明涉及一种FPGA上的AES算法的实现方法，用于解决上述现有技术的问题。

[0007] 本发明一种FPGA上的AES算法的实现方法，其中，包括:AES加密模块包括:步骤1:将输入的密文以8bit为一个单位，组成4x4矩阵，步骤2:将输入的密钥a按从高到底顺序分成4个32bit，分别为WN、WN+1、WN+2以及WN+3，将N、WN+1、WN+2以及WN+3组成的列矩阵与4x4矩阵做异或操作，设N＝0;步骤3:对步骤2得到的结果做S盒替换操作;步骤4:对步骤3得到的结果做行变换操作，其中第一行不变，第二行循环左移1个字节，第3行循环左移2个字节，第4行循环左移3个字节;步骤5:对步骤4得到结果做列变换;步骤6:将步骤5得到的结果与WN+4、WN+5、WN+6以及WN+7异或;步骤7:将N＝N+1，若N＝9则继续，否则转回步骤3;步骤8:将步骤7得到结果经过S盒变化;步骤9:将步骤8得到结果经过行变化;步骤10:将W40、W41、W42以及W43异或，得到密文输出;(2)AES解密模块，包括:步骤11:将输入的密文以8bit为一个单位，组成4x4矩阵;步骤12:将W40、W41、W42以及W43组成的列矩阵与步骤11得到的4x4矩阵做异或操作;步骤13:对步骤12得到的结果做逆行变换操作，即向右偏移，第一行偏移量为

0,第二行偏移量为1,第三行偏移量为2,第四行偏移量3;步骤14:对步骤13得到的结果做逆S盒替换操作;步骤15:将步骤14得到的结果与W40-4K、W40-4K+1、W40-4K+2以及40-4K+3异或,K＝1;步骤16:对步骤15得到结果做逆列变换;步骤17:将K＝K+1,若K＝10则继续,否则转回步骤13;步骤18:对步骤17得到结果做逆行变换;步骤19:对步骤18得到结果做逆列变换;步骤20:将W0、W1、W2以及W3异或,得到明文输出。

[0008] 根据本发明的FPGA上的AES算法的实现方法的一实施例,其中,在求取Wi的时候,如果i/4没有余数,Wi的求取过程如下:a:将Wi-1循环左移一个字节;b:将步骤a得到的结果经过S盒变换;c:用i/4得到的商j作为轮常数f(j)的输入,通过查找得出轮常数f(j)的结果,最后与b步骤得到的结果异或;d:将步骤c得到的结果与Wi-4异或,作为Wi的值;步骤3:如果k/4有余数,Wk的求解过程如下:Wk＝Wk-1^Wk-4。

[0009] 根据本发明的FPGA上的AES算法的实现方法的一实施例,其中,将初始输入的密钥a按从高到底顺序分成4个32bit,为[W0、W1、W2、W3]经过密钥扩展模块后,得到40个长度为32bit的子密钥,分别是:[W4、W5、W6、W7]、.....至[W40、W41、W42、W43],参与到后续的加解密步骤中。

[0010] 根据本发明的FPGA上的AES算法的实现方法的一实施例,其中,步骤骤5对步骤4得到结果做列变换包括:

[0011] $\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$ 为变化前任意一列;$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix}$ 为变化后;对 $\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$ 其进行列变化可以用公式1-1完成描述;

[0012] $$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 02 & 02 & 01 & 01 \\ 01 & 02 & 02 & 01 \\ 01 & 01 & 02 & 02 \\ 02 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} + \begin{bmatrix} 00 & 01 & 01 & 01 \\ 01 & 00 & 01 & 01 \\ 01 & 01 & 00 & 01 \\ 01 & 01 & 01 & 00 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad (1\text{-}1);$$

[0013] 从式1-1中可以得出式1-2:

[0014] $$\begin{aligned} s_0' &= 02 \bullet (s_0 \oplus s_1) \oplus s_1 \oplus s_2 \oplus s_3 \\ s_1' &= 02 \bullet (s_1 \oplus s_2) \oplus s_0 \oplus s_2 \oplus s_3 \\ s_2' &= 02 \bullet (s_2 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_3 \\ s_3' &= 02 \bullet (s_0 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_2 \end{aligned} \quad (1\text{-}2);$$

[0015] 对式(1-2)中的•做出说明:

[0016] 02•a＝xtime(a)＝{a[6:0],1'b0}^(8'h1b&{8{a[7]}});

[0017] 以此将4列全部实现列变化。

[0018] 根据本发明的FPGA上的AES算法的实现方法的一实施例,其中,步骤6对步骤5得到结果做逆列变换,包括:

[0019] 首先,设S0、S1、S2以及S3为状态矩阵的某一列,在解密过程中式(2-1):

[0020] $$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 00 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (2\text{-}1)$$

[0021] 其可拆分为式(2-2)：

$$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \\ 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

[0022] $$= \begin{bmatrix} 02 & 02 & 01 & 01 \\ 01 & 02 & 02 & 01 \\ 01 & 01 & 02 & 02 \\ 02 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 01 & 00 & 00 & 00 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}$$
$$+ \begin{bmatrix} 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 04 & 00 & 04 & 00 \\ 00 & 04 & 00 & 04 \\ 04 & 00 & 04 & 00 \\ 00 & 04 & 00 & 04 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (2\text{-}2)$$

[0023] 其中S'_0、S'_1、S'_2、S'_3的表示分别如式(2-3)、(2-4)、(2-5)、(2-6)

[0024] $$S_0' = (\{02\} \bullet (S_0 \oplus S_1)) \oplus (S_1 \oplus S_2) \oplus S_3 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_0 \oplus S_2)) \quad (2\text{-}3)$$

[0025] $$S_1' = S_0 \oplus (\{02\} \bullet (S_1 \oplus S_2)) \oplus (S_3 \oplus S_2) \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_1 \oplus S_3)) \quad (2\text{-}4)$$

[0026] $$S_2' = (S_0 \oplus S_1) \oplus (\{02\} \bullet (S_2 \oplus S_3)) \oplus S_3 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_0 \oplus S_2)) \quad (2\text{-}5)$$

[0027] $$S_3' = (\{02\} \bullet (S_0 \oplus S_3)) \oplus (S_1 \oplus S_2) \oplus S_0 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_1 \oplus S_3)) \quad (2\text{-}6)$$

[0028] 上式涉及到域GF(2^8)上的常数{02}运算，可以用下述xtime函数来实现，对于任一字节a，有式(2-7)：

[0029] xtime(a)＝{02}•a＝{a[6:0],1'b0}∧(8'h1b&{8{a[7]}}) (2-7)

[0030] 式(2-7)xtime(a)中的a为一个8bit的数，由公式可以看出，如果a的最高bit数字是0，其操作过程仅是将a向左移动一个bit，最低位补0，如果最高bit是1，操作过程是将a左移一个bit之后在与十六进制数字1b异或；

[0031] 式中常数{04}运算可以表达为2个常数乘{02}运算的级联，记常数乘{04}运算为x2time，对于任一字节a，有式(2-8)。同样常数{08}运算可以表达为3个常数乘{02}运算的级联，记常数乘{08}运算为x3time，有式(2-9)；

[0032] {04}·a＝x2time(a)＝xtime(xtime(a)) (2-8)

[0033] {08}·a＝{02·02·02}·a＝x3time(a)＝xtime(xtime(xtime(a))) (2-9)。

[0034] 本发明涉及FPGA上一种AES算法的实现方法,可实现对称加密算法的加密速率及硬件逻辑资源的平衡;AES算法是一个典型的迭代型分组密码,其分组长度和密钥长度都可变,分组长度和密钥长度可以独立地指定为128位、192位和256位;被采用的AES算法加密轮数依赖于所选择的子密钥长度;选择128位的密钥长度,加密轮数为10轮,选择192位的密钥长度,加密轮数为12轮,选择256位的密钥长度,加密轮数为14轮;AES算法模块包含密钥扩展模块、加密模块、解密模块;密钥扩展模块实现将初始密钥进行密钥扩展后生成轮密钥,并传给加、解密算法模块;加密模块实现明文数据和密钥的加密操作并输出密文;解密模块实现密文数据和密钥的解密操作并输出明文。

[0035] 本发明的FPGA上一种AES加解密算法的快速实现方式,该实现方式可根据AES加解密算法优化其相关步骤,达到合理清晰的模块化设计,逻辑资源占有量小,性能稳定可靠,模块本身的扩展性高,安全方便的升级方式等优点。

附图说明

[0036] 图1为图1为AES算法整体框图;

[0037] 图2为AES密钥扩展流程示意图;

[0038] 图3为AES加密算法实现流程示意图;

[0039] 图4为结果做行变换操作后的示意图;

[0040] 图5为AES加密算法实现流程示意图。

具体实施方式

[0041] 为使本发明的目的、内容、和优点更加清楚,下面结合附图和实施例,对本发明的具体实施方式作进一步详细描述。

[0042] 本发明的目的是提供FPGA上一种AES加解密算法,该实现方式可根据AES加解密算法优化其相关步骤,达到合理清晰的模块化设计,逻辑资源占有量小,性能稳定可靠,模块本身的扩展性高,安全方便的升级方式等优点。

[0043] FPGA上一种AES算法的实现方法,其特征在于:

[0044] (1)AES加密模块,其步骤如下:

[0045] 步骤1:将输入的明文以8bit为一个单位,组成4x4矩阵,

[0046] 步骤2:将W0、W1、W2、W3组成的列矩阵与步骤1得到的矩阵做异或操作。

[0047] 步骤3:对步骤2得到的结果做S盒替换操作。

[0048] 步骤4:对步骤3得到的结果做行变换操作,其中第一行不变,第二行循环左移1个字节,第3行循环左移2个字节,第4行循环左移3个字节。

[0049] 步骤5:对步骤4得到结果做列变换。

[0050] 步骤6:将步骤5得到的结果与[W4、W5、W6、W7]第1次迭代)异或。(当第2此迭代时,将步骤5得到结果与[W8、W9、W10、W11]异或,以此类推迭代第3、4、5、6、7、8次,当第9次迭代时,将步骤5得到结果与[W36、W37、W38、W39]异或)。

[0051] 步骤7:重复步骤3、4、5、6,迭代9次。

[0052] 步骤8:将步骤7得到结果经过S盒变化。

[0053] 步骤9:将步骤8得到结果经过行变化。

[0054] 步骤10:将步骤9得到结果与[W40、W41、W42、W43]异或,得到密文输出。

[0055] (2)AES解密模块,其步骤如下:

[0056] 步骤1:将输入的密文以8bit为一个单位,组成4x4矩阵,

[0057] 步骤2:将W40、W41、W42、W43组成的列矩阵与步骤1得到的矩阵做异或操作。

[0058] 步骤3:对步骤2得到的结果做逆行变换操作,即向右偏移,第一行偏移量为0,第二行偏移量为1,第三行偏移量为2,第四行偏移量3。

[0059] 步骤4:对步骤3得到的结果做逆S盒替换操作。

[0060] 步骤5:将步骤4得到的结果与([W36、W37、W38、W39]第1次迭代时)异或。(当第2次迭代时,将步骤4的得到结果与[W32、W33、W34、W35]异或,以此类推迭代第3、4、5、6、7、8次,当第9次迭代时,将步骤5得到结果与[W4、W5、W6、W7]异或)。

[0061] 步骤6:对步骤5得到结果做逆列变换。

[0062] 步骤7:重复步骤3、4、5、6,迭代9次。

[0063] 步骤8:对步骤7得到结果做逆行变换。

[0064] 步骤9:对步骤8得到结果做逆列变换。

[0065] 步骤10:将步骤9得到结果[W0、W1、W2、W3]异或,得到明文输出。

[0066] (3)对于AES加解密算法,在进行FPGA逻辑设计时为了减少逻辑资源和提高算法速率,将加密和解密步骤中的步骤4、步骤5、步骤6合并在一起实现,可以有效减少逻辑资源并提升算法速率。

[0067] FPGA上一种高效AES算法的实现方法,图1所示是本发明的整体框图。该算法包括AES密钥扩展模块,AES加密模块;AES解密模块。

[0068] 如图2所示,AES密钥扩展步骤如下:

[0069] 步骤1:将输入的密钥a按从高到底顺序分成4个32bit(字),分别为W0、W1、W2、W3,(W0=a[127:96],以此类推)。

[0070] 步骤2:在求取Wi的时候,如果i/4没有余数,Wi的求取过程如下:

[0071] a:将Wi-1循环左移一个字节。

[0072] b:将步骤a得到的结果经过S盒变换。

[0073] c:用i/4得到的商j,作为轮常数f(j)的输入,通过查找得出轮常数f(j)的结果,最后与b步骤得到的结果异或。

[0074] d:将步骤c得到的结果与Wi-4异或,作为Wi的值。

[0075] 步骤3:如果k/4有余数,Wk的求解过程如下:

[0076] Wk=Wk-1^Wk-4。

[0077] 通过上述的分析得到在已知W0、W1、W2、W3的情况下,如果要求取Wi,i=4,5,.....,43时都需要得出上一个Wj的具体数值,所以在用FPGA实现密钥扩展时可以顺序的实现Wj数值的计算。

[0078] 如图3所示,AES加密算法步骤如下:

[0079] 加密过程可以分解为一下步骤:

[0080] 步骤1:将输入的明文以8bit为一个单位,组成4x4矩阵,(其中a00=a[127:120],

a33＝a[7:0])。

[0081] 步骤2:将W0、W1、W2、W3组成的列矩阵与步骤1得到的矩阵做亦或操作。

[0082] 步骤3:对步骤2得到的结果做S盒替换操作。

[0083] 步骤4:对步骤3得到的结果做行变换操作,其中第一行不变,第二行循环左移1个字节,第3行循环左移2个字节,第4行循环左移3个字节。图4为结果做行变换操作后的示意图。

[0084] 步骤5:对步骤4得到结果做列变换。其具体过程如下:

[0085] $\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$为变化前任意一列。$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix}$为变化后。对$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$其进行列变化可以用公式1-1完成描述。

[0086] $$\begin{bmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 02 & 02 & 01 & 01 \\ 01 & 02 & 02 & 01 \\ 01 & 01 & 02 & 02 \\ 02 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} + \begin{bmatrix} 00 & 01 & 01 & 01 \\ 01 & 00 & 01 & 01 \\ 01 & 01 & 00 & 01 \\ 01 & 01 & 01 & 00 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad (1\text{-}1)$$

[0087] 从式1-1中可以得出式1-2:

[0088] $$\begin{aligned} s_0' &= 02 \bullet (s_0 \oplus s_1) \oplus s_1 \oplus s_2 \oplus s_3 \\ s_1' &= 02 \bullet (s_1 \oplus s_2) \oplus s_0 \oplus s_2 \oplus s_3 \\ s_2' &= 02 \bullet (s_2 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_3 \\ s_3' &= 02 \bullet (s_0 \oplus s_3) \oplus s_0 \oplus s_1 \oplus s_2 \end{aligned} \quad (1\text{-}2)$$

[0089] 对式1.2中的·做出说明:

[0090] 02·a＝xtime(a)＝{a[6:0],1'b0}^(8'h1b&{8{a[7]}})。

[0091] 以此将4列全部实现列变化。

[0092] 步骤5:将步骤4得到的结果与(W4、W5、W6、W7)异或。(当第二轮迭代时,步骤4的得到结果与(W8、W9、W10、W11)异或以此类推)。

[0093] 步骤6:重复步骤3、4、5,迭代9次。

[0094] 步骤7:将步骤6得到结果经过S盒变化。

[0095] 步骤8:将步骤7得到结果经过行变化。

[0096] 步骤9:将步骤8得到结果(W40、W41、W42、W43)异或,得到密文输出。

[0097] 如图5所示,AES解密算法步骤如下:

[0098] 步骤1:将输入的密文以8bit为一个单位,组成4x4矩阵,

[0099] 步骤2:将W40、W41、W42、W43组成的列矩阵与步骤1得到的矩阵做异或操作。

[0100] 步骤3:对步骤2得到的结果做逆行变换操作,即向右偏移,第一行偏移量为0,第二行偏移量为1,第三行偏移量为2,第四行偏移量3。

[0101] 步骤4:对步骤3得到的结果做逆S盒替换操作,逆S盒变换与加密过程的S盒变换一样,也是查表,查表的方式也一样,只不过查的是另外一个置换表(S-Box的逆表)。

[0102] 步骤5:将步骤4得到的结果与(W4、W5、W6、W7)异或。

[0103] 步骤6:对步骤5得到结果做逆列变换。

[0104] 首先,设S0、S1、S2、S3为状态矩阵的某一列,在解密过程中式(2-1):

[0105]
$$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 00 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \tag{2-1}$$

[0106] 其可拆分为式(2-2):

[0107]
$$\begin{bmatrix} S_0' \\ S_1' \\ S_2' \\ S_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \\ 0C & 08 & 0C & 08 \\ 08 & 0C & 08 & 0C \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

[0108]
$$= \begin{bmatrix} 02 & 02 & 01 & 01 \\ 01 & 02 & 02 & 01 \\ 01 & 01 & 02 & 02 \\ 02 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 01 & 00 & 00 & 00 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix}$$
$$+ \begin{bmatrix} 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \\ 08 & 08 & 08 & 08 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} + \begin{bmatrix} 04 & 00 & 04 & 00 \\ 00 & 04 & 00 & 04 \\ 04 & 00 & 04 & 00 \\ 00 & 04 & 00 & 04 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \tag{2-2}$$

[0109] 其中S'0、S'1、S'2、S'3的表示分别如式(2-3)、(2-4)、(2-5)、(2-6)

[0110]
$$S_0' = (\{02\} \bullet (S_0 \oplus S_1)) \oplus (S_1 \oplus S_2) \oplus S_3 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_0 \oplus S_2)) \tag{2-3}$$

[0111]
$$S_1' = S_0 \oplus (\{02\} \bullet (S_1 \oplus S_2)) \oplus (S_3 \oplus S_2) \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_1 \oplus S_3)) \tag{2-4}$$

[0112]
$$S_2' = (S_0 \oplus S_1) \oplus (\{02\} \bullet (S_2 \oplus S_3)) \oplus S_3 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_0 \oplus S_2)) \tag{2-5}$$

[0113]
$$S_3' = (\{02\} \bullet (S_0 \oplus S_3)) \oplus (S_1 \oplus S_2) \oplus S_0 \oplus (\{08\} \bullet (S_0 \oplus S_1 \oplus S_2 \oplus S_3)) \oplus (\{04\} \bullet (S_1 \oplus S_3)) \tag{2-6}$$

[0114] 上式涉及到域GF(2^8)上的常数{02}运算,可以用下述xtime函数来实现,对于任一字节a,有式(2-7):

[0115] xtime(a)=={02}•a={a[6:0],1'b0}∧(8'h1b&{8{a[7]}}) (2-7)

[0116] 说明:式(2-7)xtime(a)中的a为一个8bit的数,由公式可以看出,如果a的最高bit数字是0,其操作过程仅是将a向左移动一个bit,最低位补0,如果最高bit是1,操作过程是将a左移一个bit之后在与十六进制数字1b异或。

[0117] 式中常数{04}运算可以表达为2个常数乘{02}运算的级联,记常数乘{04}运算为x2time,对于任一字节a,有式(2-8)。同样常数{08}运算可以表达为3个常数乘{02}运算的级联,记常数乘{08}运算为x3time,有式(2-9)。

[0118] {04}·a=x2time(a)=xtime(xtime(a))

[0119] (2-8)

[0120] {08}·a={02·02·02}·a=x3time(a)=xtime(xtime(xtime(a)))

[0121] (2-9)

[0122] 步骤7:重复步骤3、4、5、6,迭代9次。

[0123] 步骤8:对步骤7得到结果做逆行变换。

[0124] 步骤9:对步骤8得到结果做逆列变换。

[0125] 步骤10:将步骤9得到结果(W0、W1、W2、W3)异或,得到明文输出。

[0126] 本发明涉及FPGA上一种AES算法的实现方法,可实现对称加密算法的加密速率及硬件逻辑资源的平衡;AES算法是一个典型的迭代型分组密码,其分组长度和密钥长度都可变,分组长度和密钥长度可以独立地指定为128位、192位和256位;被采用的AES算法加密轮数依赖于所选择的子密钥长度;选择128位的密钥长度,加密轮数为10轮,选择192位的密钥长度,加密轮数为12轮,选择256位的密钥长度,加密轮数为14轮;AES算法模块包含密钥扩展模块、加密模块、解密模块;密钥扩展模块实现将初始密钥进行密钥扩展后生成轮密钥,并传给加、解密算法模块;加密模块实现明文数据和密钥的加密操作并输出密文;解密模块实现密文数据和密钥的解密操作并输出明文。

[0127] 本发明的设计方法通过优化AES密码算法模块,合并相关加解密步骤,达到合理清晰的模块化设计,逻辑资源占有量小,性能稳定可靠,模块本身的扩展性高,安全方便的升级方式等优点。

[0128] 以上仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明技术原理的前提下,还可以做出若干改进和变形,这些改进和变形也应视为本发明的保护范围。

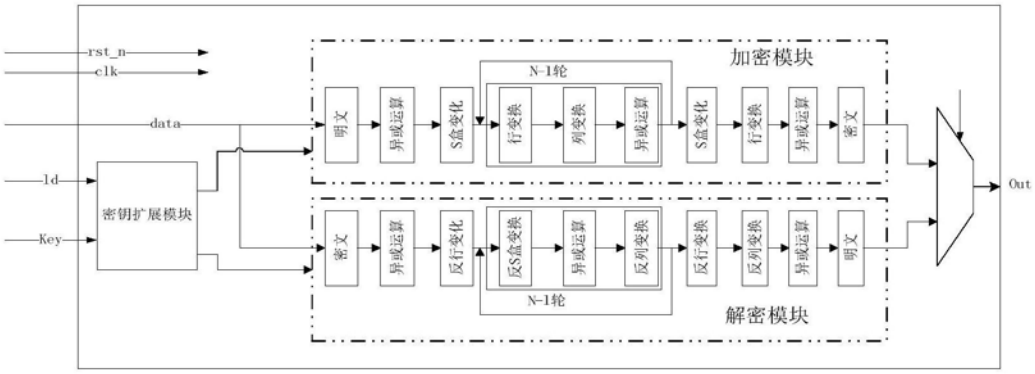

rst_n
clk
data
ld
Key
密钥扩展模块
加密模块
N-1轮
明文
异或运算
S盒变化
行变换
列变换
异或运算
S盒变化
行变换
异或运算
密文
Out
密文
异或运算
反行变化
反S盒变换
异或运算
反列变换
反行变换
反列变换
异或运算
明文
N-1轮
解密模块

图1

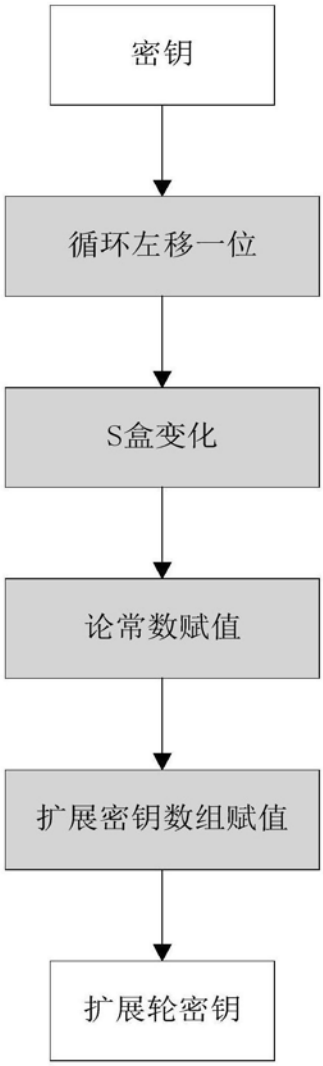

密钥
循环左移一位
S盒变化
论常数赋值
扩展密钥数组赋值
扩展轮密钥

图2

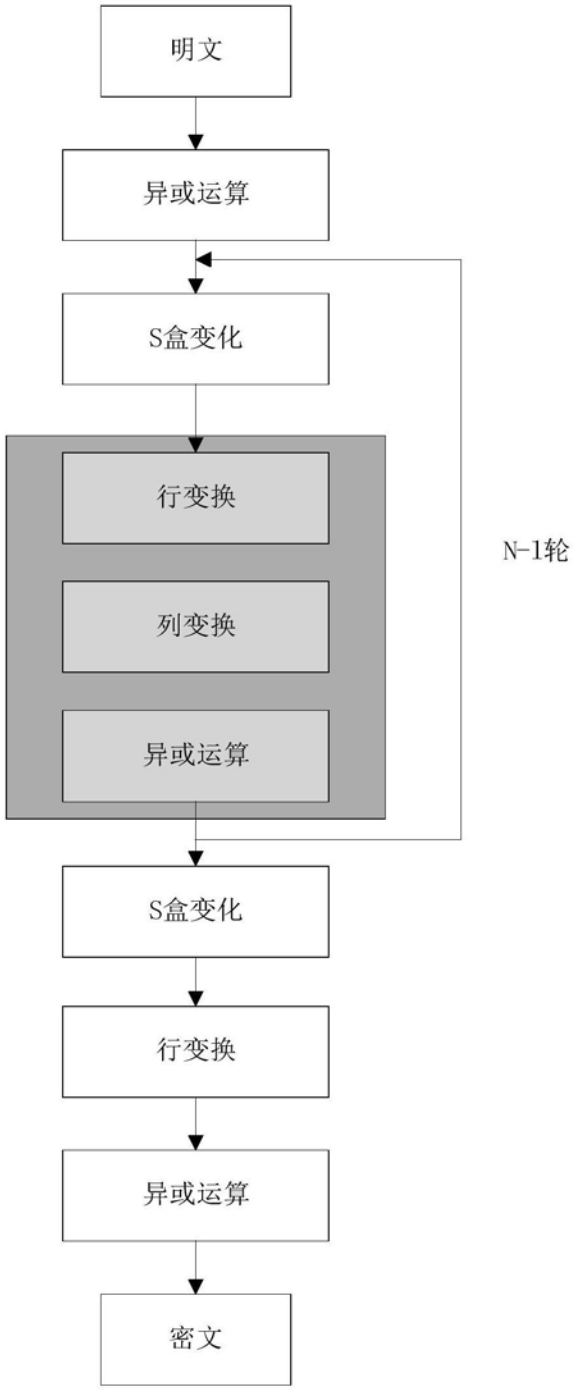


图3

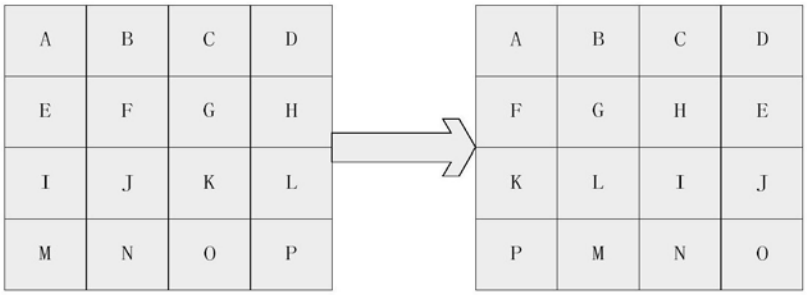


图4

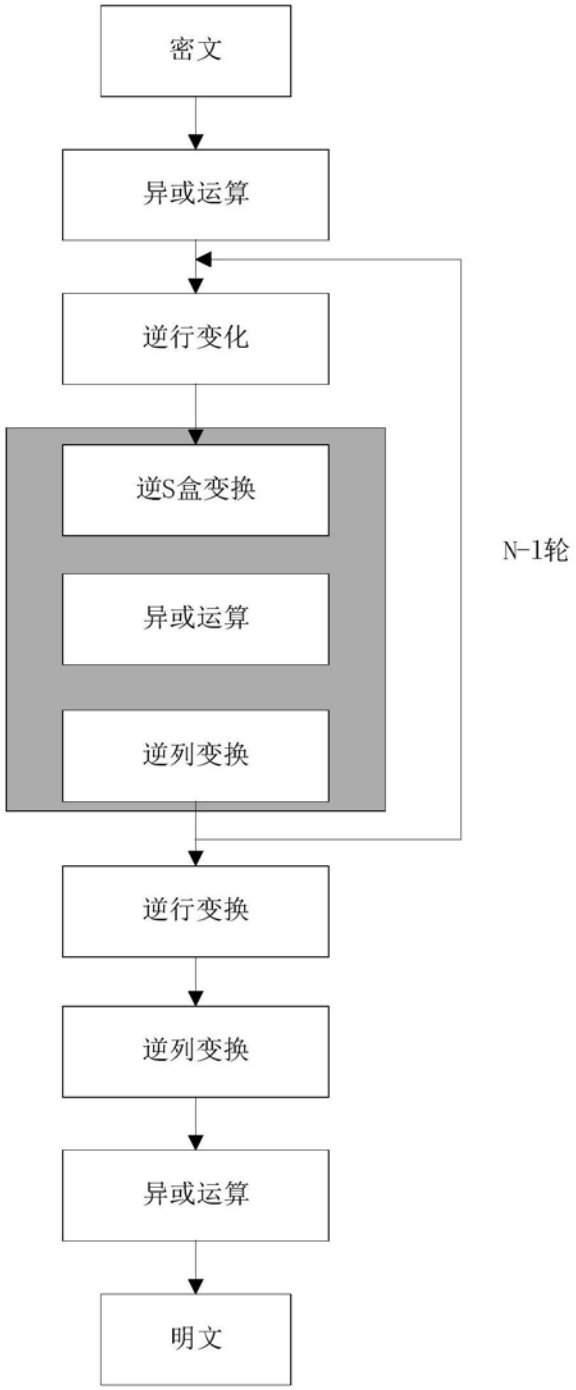


图5