(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 109962766 A
(43)申请公布日 2019.07.02

(21)申请号 201711404595.7

(22)申请日 2017.12.22

(71)申请人 中国科学院沈阳自动化研究所
地址 110016 辽宁省沈阳市东陵区南塔街114号

(72)发明人 董策 于海斌 杨志家 谢闯 王剑 段茂强 张志鹏 张超

(74)专利代理机构 沈阳科苑专利商标代理有限公司 21002
代理人 王倩

(51)Int.Cl.
H04L 9/06(2006.01)
H04L 29/06(2006.01)

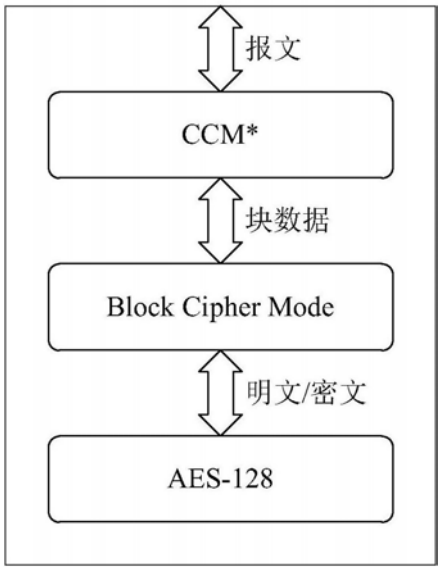
权利要求书2页 说明书6页 附图3页

(54)发明名称

基于IEEE802-15-4标准的安全服务协处理器的实现方法

(57)摘要

本发明涉及基于IEEE802-15-4标准的安全服务协处理器的实现方法，通过硬件自动地对报文进行加解密和认证操作，实现数据保密性和数据真实性安全服务，包括以下步骤：将安全服务协处理器自顶向下分为CCM*层、Block Cipher Mode层和AES-128层；所述CCM*层用于实现CCM*模式；对原始报文进行加密及认证，或者对接收到的报文进行解密及校验；所述Block Cipher Mode层用于实现分组密码工作模式；将CCM*层输出的数据块经加密或解密操作后返回至CCM*层；所述AES-128层用于实现AES加密和解密；将Block Cipher Mode层输出的128-bit数据经加密或解密操作后返回至Block Cipher Mode层。本发明既能单独实现多种分组密法算法工作模式，还能够实现IEEE 802.15.4标准中CCM*模式的加解密功能和认证功能，能够满足无线传感器网络的信息安全功能和需求。解决了传统上控制过程复杂、处理器工作负载繁重的弊端。



CN 109962766 A

1.基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于通过硬件自动地对报文进行加解密和认证操作,实现数据保密性和数据真实性安全服务,包括以下步骤:

将安全服务协处理器自顶向下分为CCM*层、Block Cipher Mode层和AES-128层;

所述CCM*层用于实现CCM*模式;通过对Block Cipher Mode层的功能调用,实现对原始报文进行加密及认证,或者对接收到的报文进行解密及校验;

所述Block Cipher Mode层用于实现分组密码工作模式;通过对AES-128层的功能调用,将CCM*层输出的数据块经加密或解密操作后返回至CCM*层;

所述AES-128层用于实现AES加密和解密;将Block Cipher Mode层输出的128-bit数据经加密或解密操作后返回至Block Cipher Mode层。

2.根据权利要求1所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述CCM*层为CCM*模块,其实现包括以下步骤:

根据需求设置CCM*模块的操作模式;

根据CCM*模块的操作模式,进入工作状态。

3.根据权利要求1所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述进入工作状态包括以下步骤:

当操作模式设定为加密及认证模式时,报文依次进入产生MIC、加密MIC和加密数据块状态;而每个状态都自动进入Block Cipher Mode模块,最终得到加密后的报文输出;

当操作模式设定为解密及认证校验时,报文依次进入解密MIC、解密数据块和校验MIC状态,而每个状态都自动进入Block Cipher Mode模块,最终得到解密后的报文输出。

4.根据权利要求1所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述Block Cipher Mode模块的实现包括以下步骤:

1)设置加密或解密操作类型;

2)设置工作模式及分组输入的长度;

3)加载密钥和初始化向量并设定反馈缓存变量;

4)加载数据输入变量并生成AES-128加密或解密操作的输入x;

5)将待加密或解密的输入x进行AES-128加密或解密操作,生成输出y;

6)更新反馈缓存变量值;当反馈循环计数值小于设定次数时,返回步骤4),否则执行下一步骤;

7)根据操作类型和工作模式生成数据输出变量text_o。

5.根据权利要求4所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述生成输出y后,当工作模式为CFB、OFB和CTR时,生成位选取变量j_z和生成密/明文分组长度变量j_blk_o。

6.根据权利要求4所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述生成输出y后,当工作模式为CBC时,生成密/明文分组变量blk_o。

7.根据权利要求4所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述更新反馈缓存变量值后,当工作模式为CFB、OFB和CTR时,生成分组输出变量blk_o_r。

8.根据权利要求1所述的基于IEEE802-15-4标准的安全服务协处理器的实现方法,其特征在于所述CCM*层、Block Cipher Mode层和AES-128层均可以通过安全服务协处理器分

别与外部处理器建立数据通路,实现各层所得结果的数据传输。

基于IEEE802-15-4标准的安全服务协处理器的实现方法

技术领域

[0001] 本发明属于加解密技术领域，具体说是一种基于IEEE 802.15.4标准的安全服务协处理器的实现方法。

背景技术

[0002] IEEE 802.15.4标准是物联网中最受欢迎、应用最广泛、最核心的技术。

[0003] IEEE 802.15.4标准的媒体访问控制子层(MAC sublayer)提供了安全服务和安全模式。在安全模式下，设备可能会根据不同的安全级别提供两种安全服务：数据保密性和数据真实性。

[0004] 数据保密性是使用对称密码算法进行加密的，使用相同的密钥在报文源加密明文和在报文目的地解密密文。

[0005] 数据真实性，也被称为数据的完整性，该服务能使接收设备通过附加到报文中的消息完整性代码(MIC)来检测该报文是否受到了没有正确加密密钥的某方的篡改。

[0006] 安全级别1～3级提供了数据真实性服务和长度分别为32、64或128位的消息完整性代码。安全级别4级提供了数据保密性服务。安全级别5～7级提供了数据保密性和数据真实性服务(消息完整性代码分别为32、64或128位)。

[0007] IEEE 802.15.4安全服务是基于CCM*模式生成一系列的安全机制。CCM*模式是CCM模式的扩展，而CCM模式是结合了分组密码算法工作模式(CTR和CBC-MAC)而衍生出来的安全模式，这套安全机制的主要优点是所有安全级别都只采用了一种加密算法，即AES-128加密算法(128位数据分组长度和128位密钥长度的AES加密算法)。特别是，通过巧妙地重复利用AES算法，CCM*模式能使一个简单的算法在一个很小的实现中提供更高的安全服务。

[0008] 分组密码又称块密码算法，是一种对称密码算法，将明文划分成固定长度的分组进行加密。分组密码算法工作模式是分组密码算法的使用方式，主要包括电码本模式(ECB)、密码分组链接模式(CBC)、密码反馈模式(CFB)、输出反馈模式(OFB)、计数器模式(CTR)等。

[0009] AES(高级加密标准)，是由NIST(美国国家标准与技术研究院)于2001年11月26日发布于FIPS PUB 197，并在2002年5月26日成为有效的标准。AES加密算法，又称为Rijndael加密算法，该算法为比利时密码学家Joan Daemen和Vincent Rijmen所设计，这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。AES是一个迭代的、对称密钥分组的密码，它可以使用128、192和256位密钥，并且用128位(16字节)分组加密和解密数据。IEEE 802.15.4采用固定的128位密钥，记为AES-128。不论对于AES加密算法还是解密算法，都是使用轮变换的操作。轮变换操作次数与密钥的位数有关，AES-128轮数为10轮。IEEE 802.15.4协议中只用到了AES-128加密算法，明文首先进行一个密钥加的操作，然后进行10次轮变换操作。轮变换包括4个操作：字节置换、行换位、列混合和密钥加。

[0010] 传统无线网络节点所采用的安全机制存在一个弊端，在硬件上没有针对无线网络特性的安全服务协处理器，对于硬件资源和计算能力有限的无线网络节点，只能依靠运行

在通用的嵌入式处理器上的软件程序,无法实现计算复杂度高、计算资源消耗较大的安全协议。即在一次加解密或身份认证流程中,处理器必须进行多次干预,既使控制过程复杂化,也增加了处理器的工作负载。

发明内容

[0011] 本发明提出了一种基于IEEE 802.15.4标准的安全服务协处理器的实现方法,以克服上述技术不足。

[0012] 本发明采用如下技术方案:基于IEEE 802.15.4标准的安全服务协处理器的实现方法,通过硬件自动地对报文进行加解密和认证操作,实现数据保密性和数据真实性安全服务,包括以下步骤:

[0013] 将安全服务协处理器自顶向下分为CCM*层、Block Cipher Mode层和AES-128层;

[0014] 所述CCM*层用于实现CCM*模式;对原始报文进行加密及认证,或者对接收到的报文进行解密及校验;通过对Block Cipher Mode层的功能调用,完成本层的功能;

[0015] 所述Block Cipher Mode层用于实现分组密码工作模式;将CCM*层输出的数据块经加密或解密操作后返回至CCM*层;通过对AES-128层的功能调用,完成本层的功能;

[0016] 所述AES-128层用于实现AES加密和解密;将Block Cipher Mode层输出的128-bit数据经加密或解密操作后返回至Block Cipher Mode层。

[0017] 所述CCM*层、Block Cipher Mode层和AES-128层均可以通过安全服务协处理器的控制,独立地与外部处理器建立相应的数据通路,并实现相应的层次功能。

[0018] 所述CCM*层为CCM*模块,其实现包括以下步骤:

[0019] 根据需求设置CCM*模块的操作模式;

[0020] 根据CCM*模块的操作模式,进入工作状态。

[0021] 所述进入工作状态包括以下步骤:

[0022] 当操作模式设定为加密及认证模式时,报文依次进入产生MIC、加密MIC和加密数据块状态;而每个状态都自动进入Block Cipher Mode模块,最终得到加密后的报文输出;

[0023] 当操作模式设定为解密及认证校验时,报文依次进入解密MIC、解密数据块和校验MIC状态,而每个状态都自动进入Block Cipher Mode模块,最终得到解密后的报文输出。

[0024] 所述Block Cipher Mode模块的实现包括以下步骤:

[0025] 1)设置加密或解密操作类型;

[0026] 2)设置工作模式及分组输入的长度;

[0027] 3)加载密钥和初始化向量并设定反馈缓存变量;

[0028] 4)加载数据输入变量并生成AES-128加密或解密操作的输入x;

[0029] 5)将待加密或解密的输入x进行AES-128加密或解密操作,生成输出y;

[0030] 6)更新反馈缓存变量值;当反馈循环计数值小于设定次数时,返回步骤4),否则执行下一步骤;

[0031] 7)根据操作类型和工作模式生成数据输出变量text_o。

[0032] 所述生成输出y后,当工作模式为CFB、OFB和CTR时,生成位选取变量j_z和生成密/明文分组长度变量j_blk_o。

[0033] 所述生成输出y后,当工作模式为CBC时,生成密/明文分组变量blk_o。

[0034] 所述更新反馈缓存变量值后，当工作模式为CFB、OFB和CTR时，生成分组输出变量blk_o_r。

[0035] 本发明具有以下有益效果及优点：

[0036] 1、本发明能够完全由硬件自动实现以下功能：

[0037] 1)独立实现IEEE 802.15.4标准中要求的数据保密性和数据真实性安全服务；

[0038] 2)独立实现分组密法算法的多种工作模式；

[0039] 3)独立实现AES加密和解密；

[0040] 2、本发明解决了传统上控制过程复杂、处理器工作负载繁重的弊端。

附图说明

[0041] 图1为安全服务协处理器的设计层次示意图；

[0042] 图2为CCM*模块工作状态示意图；

[0043] 图3为Block Cipher Mode模块的数据流程示意图；

[0044] 图4为AES-128加密操作的结构示意图；

[0045] 图5为AES-128解密操作的结构示意图。

具体实施方式

[0046] 下面结合附图对本发明做进一步的详细说明。

[0047] 安全服务协处理器的设计自顶向下分为3个层次，如图1所示：

[0048] 1、CCM*层：即CCM*模块，实现CCM*模式；

[0049] 1)实现IEEE Std 802.15.4标准中要求的数据保密性和数据真实性安全服务；

[0050] 2)提供数据链路层及应用层的MIC(消息完整性代码)生成和检查，数据负载的加密和解密功能。

[0051] 2、Block Cipher Mode层：即Block Cipher Mode模块，实现分组密码工作模式；

[0052] 3、AES-128层：即AES-128模块，实现AES加密和解密，分组长度为128-bit，密钥长度为128-bit。

[0053] 本实施采用硬件描述语言Verilog编写RTL代码，使用逻辑综合工具DesignCompiler生成Verilog网表，形成安全服务协处理器电路。安全服务协处理器电路包括CCM*模块、Block Cipher Mode模块和AES-128模块。

[0054] 本发明中的CCM*模块实现方法如下：

[0055] 步骤1：根据需求设置安全级别，安全级别如下表所示：

[0056]

安全级别	安全级别	数据保密性	数据真实性
0	None	否	否
1	MIC-32	否	是(4字节MIC)
2	MIC-64	否	是(8字节MIC)
3	MIC-128	否	是(16字节MIC)
4	ENC	是	否
5	ENC-MIC-32	是	是(4字节MIC)

6	ENC-MIC-64	是	是(8字节MIC)
7	ENC-MIC-128	是	是(16字节MIC)

[0057] 步骤2:根据需求设置模块的操作模式。操作模式有加密及认证模式和解密及认证校验模式。

[0058] 步骤3:根据需求设置模块中与IEEE 802.15.4标准安全服务相关的参数,例如密钥(key)、源地址(source address)和帧计数器(frame counter)等。

[0059] 步骤4:根据模块的操作模式,依次进入以下工作状态,如图2所示。

[0060] ①当操作模式设定为加密及认证模式时,依次进入产生MIC、加密MIC和加密数据块状态,而每个状态都自动进入Block Cipher Mode模块,Block Cipher Mode模块的所有实现步骤完全由硬件自动实现,无需软件介入,当Block Cipher Mode完成操作时,返回到当前状态:

[0061] 产生MIC状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0062] 输入是以多个128-bit数据块形式存在的原始报文(加密前);

[0063] 输出是包含MIC(加密前)的128-bit数据块;

[0064] 设置工作模式为CBC。

[0065] 加密MIC状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0066] 输入是包含MIC(加密前)的128-bit数据块;

[0067] 输出是包含MIC(加密后)的128-bit数据块;

[0068] 设置工作模式为CTR。

[0069] 加密数据块状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0070] 输入是以多个128-bit数据块形式存在的原始报文(加密前);

[0071] 输出是以多个128-bit数据块形式存在的加密报文;

[0072] 设置工作模式为CTR。

[0073] ②当操作模式设定为解密及认证校验时,依次进入解密MIC、解密数据块和校验MIC状态,而每个状态都自动进入Block Cipher Mode模块,Block Cipher Mode模块的所有实现步骤完全由硬件自动实现,无需软件介入,当Block Cipher Mode完成操作时,返回到当前状态。

[0074] 解密MIC状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0075] 输入是包含MIC(解密前)的128-bit数据块;

[0076] 输出是包含MIC(解密后)的128-bit数据块;

[0077] 设置工作模式为CTR。

[0078] 解密数据块状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0079] 输入是以多个128-bit数据块形式存在的原始报文(解密前);

[0080] 输入是以多个128-bit数据块形式存在的解密报文;

[0081] 设置工作模式为CTR。

[0082] 校验MIC状态时,Block Cipher Mode模块的输入、输出和设置如下:

[0083] 输入是以多个128-bit数据块形式存在的解密报文;

[0084] 输出是包含MIC的128-bit数据块及MIC的校验结果;

[0085] 设置工作模式为CBC。

[0086] 步骤5:当完成以上步骤后,产生相应的完成信号或中断信号。

[0087] 如图3所示,本发明中的Block Cipher Mode模块实现方法如下:

[0088] 步骤1:根据需求设置模块的操作类型。操作类型可分为加密和解密两种。

[0089] 步骤2:根据需求设置分组密码算法工作模式。本发明中设置了五种常用的分组密码算法工作模式:ECB、CBC、CFB、OFB和CTR。

[0090] ①ECB(电码本工作模式)是分组密码算法的一种工作模式,明文分组直接作为加密算法的输入,对应的输出作为密文分组。

[0091] ②CBC(密码分组链接工作模式)是分组密码算法的一种工作模式,当前的明文分组与前一密文分组进行异或运算后再进行加密得到当前的密文分组。

[0092] ③CFB(密码反馈工作模式)是分组密码算法用于构造序列密码的一种工作模式,用密文依次更新存储该密码算法启动变量的反馈缓冲器。

[0093] ④OFB(输出反馈工作模式)是分组密码算法用于构造序列密码的一种工作模式,用该算法当前时刻的输出作为下一时刻的输入。

[0094] ⑤CTR(计数器工作模式)是分组密码算法用于构造序列密码的一种工作模式,通过加密不断变化的计数器来产生密钥序列。

[0095] 步骤3:根据需求设置工作模式为CFB、OFB和CTR时分组输入的长度。分组输入的长度可以设置为8-bit、16-bit、32-bit、64-bit和128-bit。

[0096] 同时根据分组输入的长度,设置反馈循环变量。同时根据分组输入的长度,设置反馈循环计数值等于0。

[0097] 步骤4:加载key(密钥),密钥长度为128-bit。

[0098] 步骤5:加载iv(初始化向量),初始化向量长度为128-bit。

[0099] 当工作模式为CBC、CFB、OFB和CTR时,同时设定fb(反馈缓存变量)的初值。

[0100] 步骤6:加载text_i(数据输入变量),数据输入变量长度为128-bit。

[0101] 步骤7:生成AES-128模块的输入x(AES-128输入变量),AES-128输入变量长度为128-bit。

[0102] 当模块操作类型是加密且工作模式为ECB时,x=text_i;

[0103] 当模块操作类型是加密且工作模式为CBC时,x=text_i^fb;

[0104] 当模块操作类型是加密且工作模式为CFB、OFB和CTR时,x=fb;

[0105] 当模块操作类型是解密且工作模式为ECB和CBC时,x=text_i;

[0106] 当模块操作类型是解密且工作模式为CFB、OFB和CTR时,x=fb;

[0107] 步骤8:进入AES-128模块,AES-128模块的所有实现步骤完全由硬件自动实现,无需软件介入。将待加密或解密的x(AES-128输入变量)进行加密或解密操作,生成y(AES-128输出变量),然后返回到Block Cipher Mode模块。

[0108] 步骤9:当工作模式为CFB、OFB和CTR时,生成j_z(位选取变量)。位选取变量的长度由分组输入的长度决定。可参考GB/T 17964-2008。

[0109] 步骤10:当工作模式为CBC时,生成blk_o(密/明文分组变量)。密/明文分组变量的长度为128-bit。

[0110] 步骤11:当工作模式为CFB、OFB和CTR时,生成j_blk_o(密/明文分组长度变量)。密/明文分组长度变量的长度由分组输入的长度决定。可参考GB/T17964-2008。

[0111] 步骤12:更新反馈缓存变量fb的值。同时反馈循环计数值加1。

[0112] 步骤13:当工作模式为CFB、OFB和CTR时,生成blk_o_r(分组输出变量)。分组输出变量的长度为128-bit。

[0113] 步骤14:比较反馈循环计数值与反馈循环变量的大小。当反馈循环计数值小于反馈循环次数时,返回值步骤7;当反馈循环计数值等于反馈循环次数时,进入下一步骤。

[0114] 步骤15:生成text_o(数据输出变量)。

[0115] 数据输出变量的长度为128-bit,并通过输出端口被链路下一单元读取。

[0116] 当工作模式为ECB和CBC时,text_o=y;

[0117] 当模块操作类型是加密且工作模式为ECB和CBC时,text_o=y;

[0118] 当模块操作类型是加密且工作模式为CFB、OFB和CTR时,text_o=blk_o_r;

[0119] 当模块操作类型是解密且工作模式为ECB时,text_o=y;

[0120] 当模块操作类型是解密且工作模式为CBC时,text_o=blk_o;

[0121] 当模块操作类型是解密且工作模式为CFB、OFB和CTR时,text_o=blk_o_r;

[0122] 本发明中的AES-128模块实现方法如下:

[0123] 加密或解密的操作基于现有AES-128算法,每一轮的密钥加等操作采用现有AES-128算法中的方法。

[0124] 图4可以视为一个AES-128加密操作的循环结构示意图(以2次轮变换为例)。

[0125] 当执行AES-128加密功能时,AES-128模块实现方法如下:

[0126] 步骤1:加载明文数据,以数据长度等于128-bit进行分组。

[0127] 步骤2:密钥扩展操作,生成用于10次轮变换操作的轮密钥k(0)～k(9)。

[0128] 步骤3:待加密的明文数据首先经过一个初始密钥加操作,然后进行10次轮变换,最后生成密文数据。

[0129] 轮变换由字节置换、行换位、列混合和密钥加一共4个操作构成。最后一次轮变换只有字节置换、行换位和密钥加一共3个操作。

[0130] 轮变换及其每一步操作都作用在中间结果上,将该中间结果成为状态。状态可以表示为一个矩形的字节数组,该数组共有4行4列。

[0131] 字节置换是作用在字节上的砖匠置换。

[0132] 行换位是一个字节换位,它将4字节的行按照不同的偏移量进行循环移位。

[0133] 列混合是作用在4字节列上的砖匠置换。

[0134] 密钥加是指中间结果(状态)与一个轮密钥逐位异或。

[0135] 图5可以视为一个AES-128解密操作的循环结构示意图(以2次轮变换为例)。

[0136] 当执行AES-128解密功能时,AES-128模块实现方法如下:

[0137] 步骤1:加载密文数据,以数据长度等于128-bit进行分组。

[0138] 步骤2:密钥扩展操作,生成用于10次轮变换求逆操作的轮密钥k(0)～k(9)。

[0139] 步骤3:待解密的密文数据首先进行10次轮变换,最后再使用一个密钥加操作,生成明文数据。

[0140] 轮变换由密钥加、列混合求逆、行换位求逆和字节置换求逆一共4个操作构成。第一次轮变换只有密钥加、行换位求逆和字节置换求逆一共3个操作。

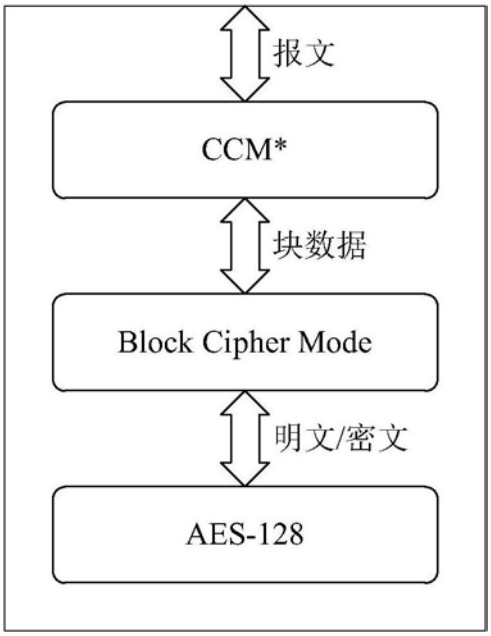

报文
CCM*
块数据
Block Cipher Mode
明文/密文
AES-128

图1

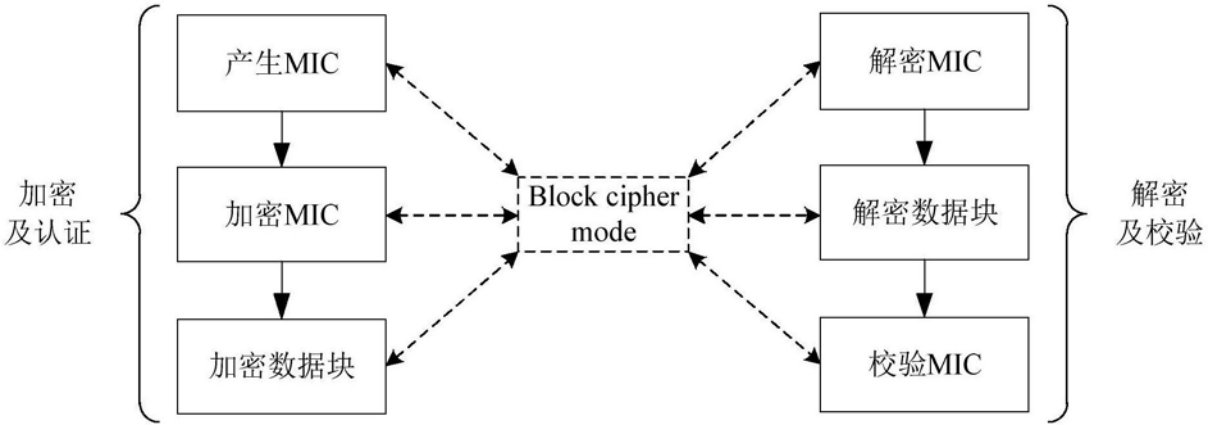

加密
及认证
产生MIC
加密MIC
加密数据块
Block cipher
mode
解密MIC
解密数据块
校验MIC
解密
及校验

图2

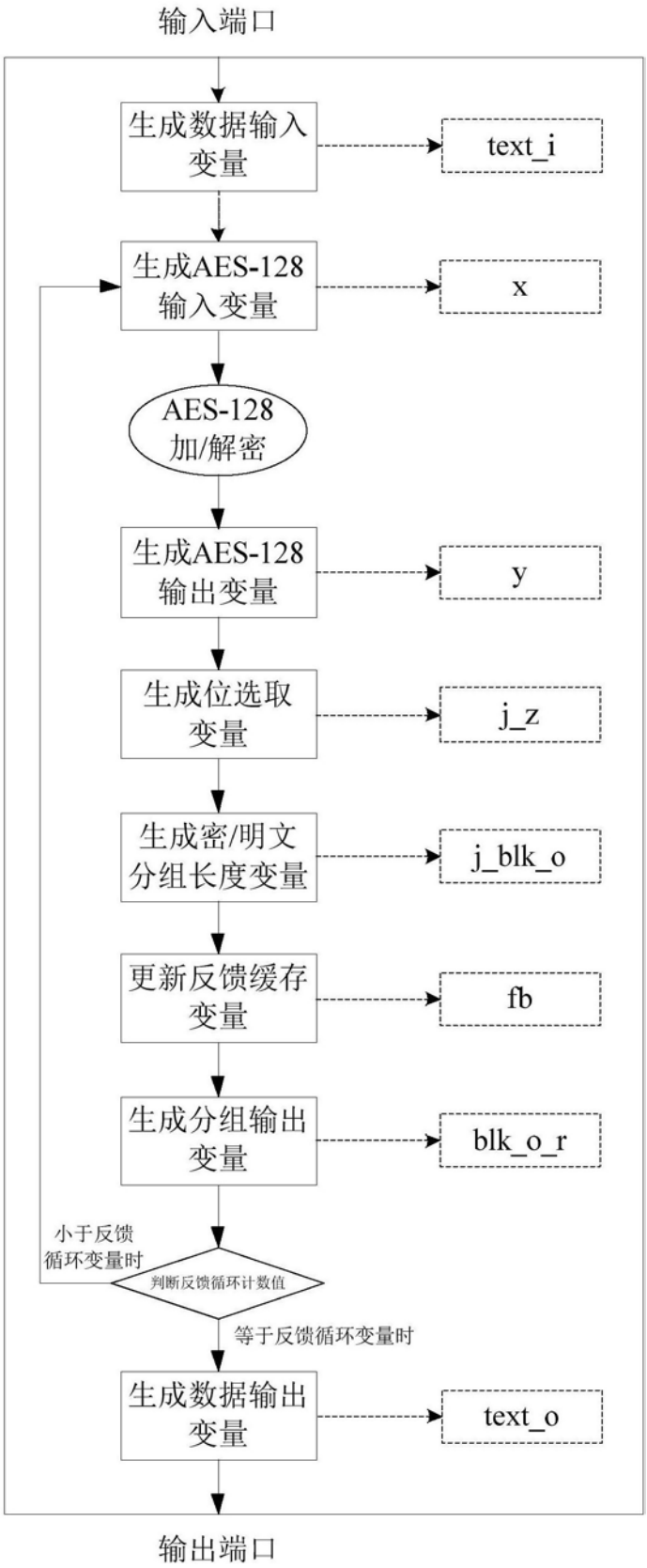

输入端口
生成数据输入变量
text_i
生成AES-128输入变量
x
AES-128加/解密
生成AES-128输出变量
y
生成位选取变量
j_z
生成密/明文分组长度变量
j_blk_o
更新反馈缓存变量
fb
生成分组输出变量
blk_o_r
小于反馈循环变量时
判断反馈循环计数值
等于反馈循环变量时
生成数据输出变量
text_o
输出端口

图3

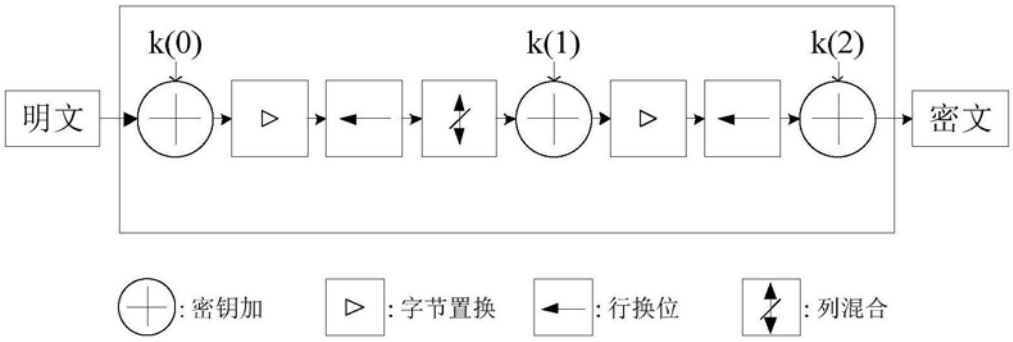

k(0)
k(1)
k(2)
明文
密文
: 密钥加
: 字节置换
: 行换位
: 列混合

图4

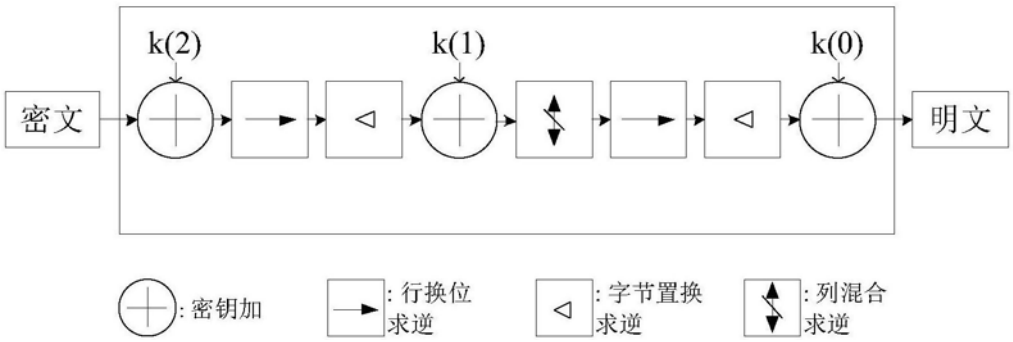

k(2)
k(1)
k(0)
密文
明文
: 密钥加
: 行换位
求逆
: 字节置换
求逆
: 列混合
求逆

图5