Designing and Realization for AES-CCM Encryption Protocol in WiMAX

XIA Jinwei, REN Shijin

School of Computer Science & Technology, Xuzhou Normal University, Xuzhou, China

e-mail: *xjinw@126.com*

Abstract: With the characteristics of higher transmission rate and longer transmission distance, more and more attention has been attached to WiMAX technology based on IEEE 802.16 standards. IEEE 802.16e standard adopts AES-CCM encryption protocol, which solves the deficiency of IEEE 802.16d—the lack of anti-replay protection on security mechanisms and insecurity of encryption algorithm itself. In this paper, principles of AES-CCM data encryption protocol have been analyzed in detail, which is carried out on design and hardware making use of FPGA. Analysis and simulation show that the design can occupy much less resources but high throughput.

Keywords: AES; encryption algorithm; FPGA; CCM

AES-CCM 数据加密协议在 WiMAX 中的设计和实现

夏劲伟，任世锦

徐州师范大学计算机科学与技术学院，徐州，中国，221116

e-mail: Xjinw@126.com

【摘　要】基于 IEEE 802.16 标准的 WiMAX 技术，以其传输速率高、传输距离远的特点越来越受到人们的重视。IEEE 802.16e 标准采用的是 AES-CCM 数据加密协议，该协议解决了 IEEE 802.16d 在安全机制上缺乏抗重放保护和加密算法本身不安全的问题。文中详细分析了 AES-CCM 数据加密协议的原理，并利用 FPGA 对其进行了设计和硬件实现。对设计实现进行的分析和仿真表明，该设计具有占用资源少，吞吐率高的特点。

【关键词】AES；加密算法；FPGA；CCM

1 引言

基于 IEEE 802.16 标准的 WiMAX 技术，以其传输速率高、传输距离远的特点越来越受到人们的重视。预计到 2010 年年底，WiMAX 网络覆盖的人数将达到 8 亿人。随着 WiMAX 无线网络的广泛应用，其安全问题也受到人们的关注。

IEEE 802.16e 标准采用的是 AES-CCM 数据加密协议，该协议解决了 IEEE 802.16d 在安全机制上缺乏抗重放保护和加密算法本身不安全的问题。AES-CCM 数据加密协议即以 AES 为加密核心算法，采用 CCM 认证方式，具有分组序号的初始向量[1]。

AES 加解密算法是一种具有对称性的分组密码。分组密码在应用中有多种工作模式，如电子密码本模式（ECB）、密码分组链接模式（CBC）或计数模式（CTR）等[2]。与 ECB 、CBC、 CTR 这几种模式只能提供加密服务不同， CCM 是一种同时能提供加密和鉴别服务的操作模式。它综合了计数模式 CTR 和鉴别服务 CBC 模式的优点，具有更高的安全性和可靠性[3]。采用 AES-CCM 加密模式，能够为系统提供数据保密、数据完整性保护、身份认证和反重放保护等功能，从而能够达到保护网络安全通信的目的。

AES 算法其本身有一定的计算量，再加上 802.16 支持最大为 75Mbps 的传输速度，客观上要求要有 75Mbps 的加/解密速度与之匹配，因此在实现 AES 运算将需要较大的资源。在网络产品中，如果用软件方式实现 AES-CCM 的加密解密，就要耗费大量的嵌入式微处理器资源，产品其它功能的应用将受到限制。另外，数字信号处理器（DSP）由于不含对 AES 优化的指令，所以采用通常的编程方法也无法满足 AES 加解密 75Mbps 速度的要求。因此，在 WiMAX 产品中

基金项目：徐州师范大学自然科学基金 07XLB16

要想达到 75Mbps 的 AES 加解密速度，必须开发专用芯片。

现场可编程门阵列（FPGA）是一类适于开发 AES 芯片的硬件设备，因为 AES 中含有大量可并行处理的运算步骤，而并行处理正是 FPGA 相对于微处理器最大的优势之一。

本文介绍了 AES-CCM 的原理，以及在 FPGA 中实现的方法，并验证了该方法的正确性。通过分析比较，该方法具有占用空间小，高吞吐量，安全高效的特点。

2 AES 算法设计

AES 算法是把要加密的数据分为每组 128bits 的数据块(block)，然后用长为 128、196 或者 256bits 的密钥对每组数据进行加密，最后得到密文。AES 算法的核心有：AddRoundKey（密钥扩展）、SubTytes（字节变换）、ShiftRows（行位移变换）和 MixColumns（列混合变换）这 4 种操作[3]。

加密是先将 128 位的密钥进行计算扩展成为 128 位×11 的密钥序列，然后通过对待加密数据反复进行以下几个数据变换算法，来得到最终的密文。对这几个变换算法描述如下：

（1）密钥扩展（addroundkey）

该模块的作用是将输入的 128 位密钥扩展为 128 位×11 的密钥序列。密钥扩展是一个关于 32 位一个数据单元的递归的算法，中间牵涉到 subword、rotword、rcon 三种运算以及对数组 rk[44]的指针运算。密钥扩展实际上也是一系列子算法循环得到一系列数值。运算流程比较繁琐，在此不作叙述。

整个密钥扩展完毕后，得到的是一个 44 个每单元 32 位的数组。将其顺序排列，每 4 个单元作为 roundkey，构成一个新的数组 roundkey[11]，并将其存在寄存器中，供运算时顺序取用。这样，就得到了 addroundkey 运算中所需要的 roundkey。

（2）字节变换（subbytes）

字节变换可以分为两步实现，先进行 GF(2^8)域的求逆运算，然后进行在 GF(2^8)域上定义的一个特定的乘法。由于求逆运算很复杂，在硬件中难以实现，而在 FPGA 中可以用 ROM 很方便的实现表。故可采用查表的方式设计实现字节变换，这样可以提高速度并省略了烦琐的推算。

实现方法如下：将 128 位的数据视为 16 个字节，将各字节对应的变换结果列为 16×16 大小的表格。

x\y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 1. S-box

图 1. S-box

S-box (substitution table)，查表对每个字节进行一个替换，从而得到一组新的 128 位数据。

以上即是 s-box，因一个 8 位字节在 16 进制表示下可写为 xy 的形式，将 x 和 y 的具体值分别对应到上表的行和列即可查得所需值。

（3）行位移变换（shiftrows）

行位移是在状态矩阵中，以行为单位进行横向位移，第一行保持原状。先把 128 位数据表示为 16 个 8 位字节的形式，并把这 16 个字节按列下标排为一个 4×4 的矩阵。shiftrows 就是针对这个矩阵做如下变换：

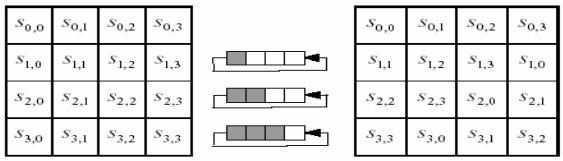

Figure 2. Shift rows

图 2. 行位移变换

（4）列混合变换（mixcolumns）

列混合变换是对状态矩阵中的各列进行如下图所示的 GF(2^8)矩阵乘法。先将输入的 128 位数据视为一个 4×4 的字节矩阵。针对矩阵的每一列，有如下图的矩阵乘法：c 从 0 取到 3。这里的乘法为点乘，当 2 去乘一个字节的时候，相当于将该字节左移 1 位而末尾补 0；由于 AES 算法中所有涉及到的加法都是模 2 加也即异或，因此当用 3 去乘一个字节 Sij 的时候，相当于 (2+1)• Sij=2• Sij ⊕ Sij，且矩阵乘法中的加法也是异或。

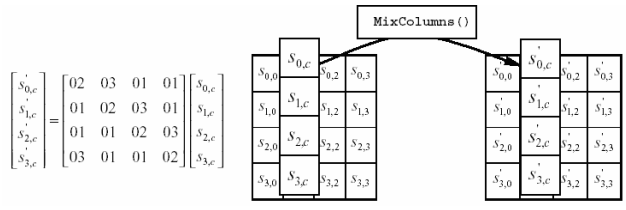

Figure 3. Mix columns

图 3. 列混合变换

978-1-935068-06-8 © 2009 SciRes.

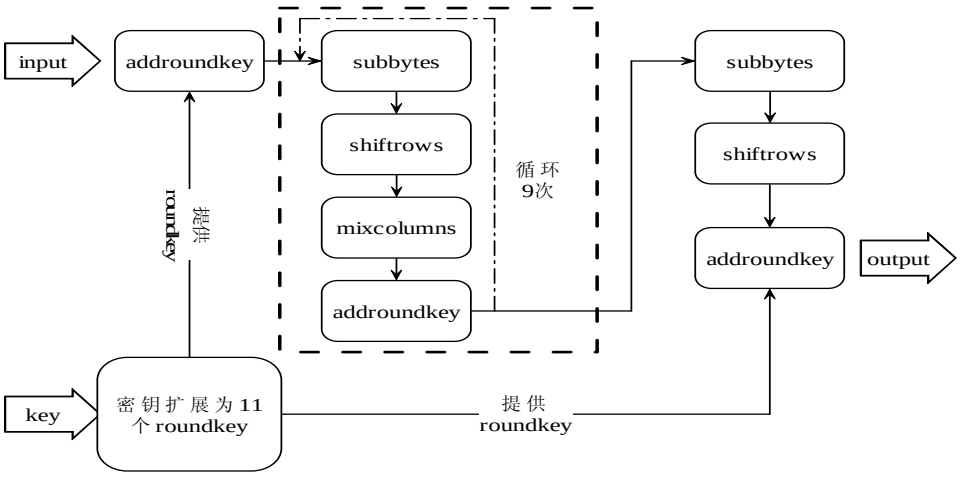

Figure 4. AES flow diagram

图 4. AES 流程图

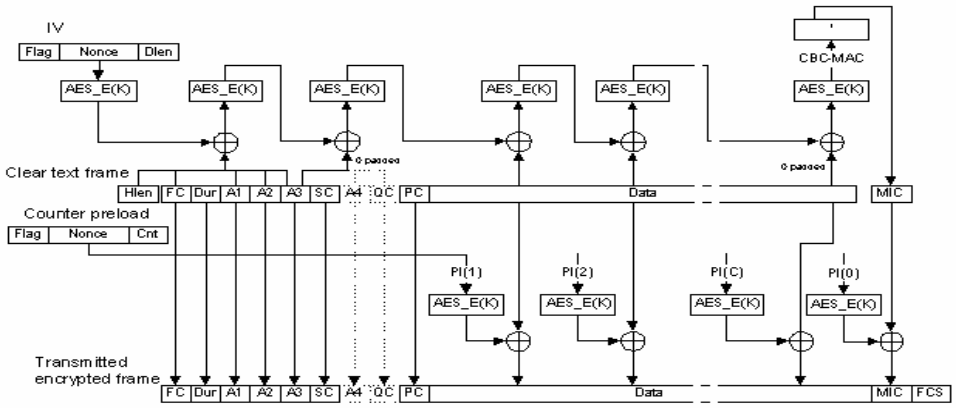

Figure 5. AES-CCM principium

图 5. AES-CCM 原理

对一个 128 位数据块完成一次 AES 的加密流程如 Figure 4，整个流程中一共需要进行 11 次 addroundkey，每次使用的 roundkey 由密钥扩展部分提供，使用的顺序是指针由 0 到 10 依次加 1。

3 AES-CCM 原理简介

CCM 模式有认证域大小 M 和长度域的大小 L 这两个选择参量[4]。M 是信息量与攻击者模仿信息的能力之间所做的权衡值，可取 4、6、8、10、12 和 16 等字节；L 是信息量最大值和 Nounce 域大小之间的权衡值。不同的应用环境可取不同的 M、L 参量值。在 Nounce 域中各有 3bit 描述（M-2）/2 和(L-1)的值。在本设计中，M、L 分别取默认的值 8 字节和 2 字节。

如图 5 所示，上半部分为 CBC-MAC 校验部分，它对帧头也进行了分组，并用 AES 算法与数据一起进行了加密，最后生成了一个 8 个字节的 MIC 校验码。然后将这个校验码存放在最后加密得到的密文数据的尾部，以供解密时的校验使用。与顺序循环方式下的 CBC 模式加密不同，CCM 不是对数据分组直接进行加密和异或。而是对初始向量 IV（Flag、Nonce、Dlen）进行 AES 加密运算，然后与帧头做异或运算，取运算所得的结果再进行 AES 加密。然后再继续同下一个分组做异或运算，如此循环下去，直到所有的分组运算结束。这样最终所得到的是一个 16 个字节的向量 CBC-MAC。

图 5 下半部分为数据加密部分，与 CBC-MAC 的计算类似，明文分组与初始向量的 AES 加密结果进行异或。不同的是初始向量（Flag、Nonce、Cnt）是随着分组不同而不同的。这些向量在进行 AES 算法加密后产生的伪随机数也是不同的。另外，计算密文时的

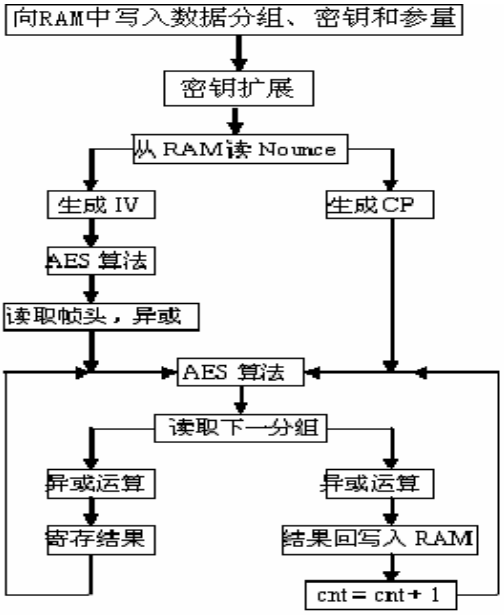

Figure 6. AES-CCM Encryption

图 6. AES-CCM 加密流程

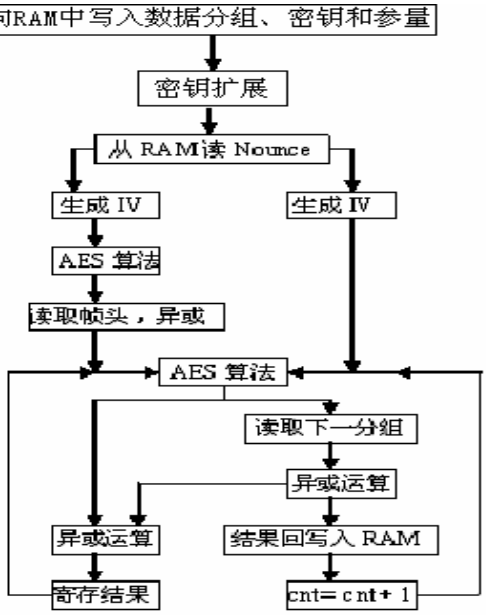

Figure 7. AES-CCM Decryption

图 7. AES-CCM 解密流程

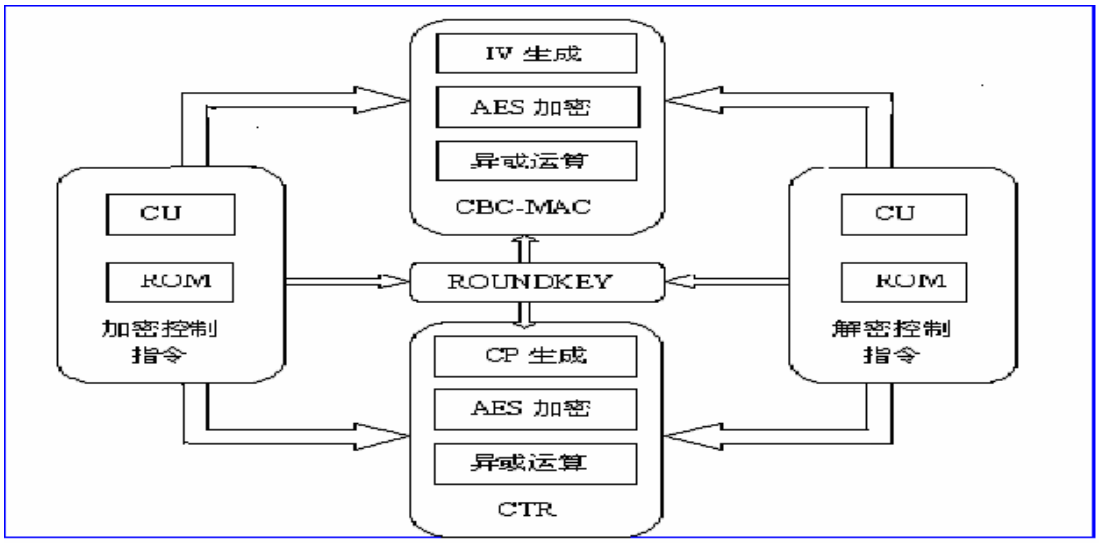

Figure 8. AES-CCM Encryption and Decryption

图 8. AES-CCM 加密与解密

分组不包括帧头的信息数据，而只是对帧内的数据进行加密运算形成密文，帧头则直接转给密文。最后将 Cnt 置零，经过 AES 算法加密后与上述产生的 CBC-MAC 向量异或，取前八字节即得最后的校验向量 MIC。

4 AES-CCM 的设计与实现

用原理图描述 AES-CCM 时，为 CBC-MAC 和 CTR 各建立一套基本 AES 循环模块。使用同一套控制模块，数据加密和校验码的产生可并行执行。

在本设计中假设初始向量所需的 Nounce 和帧头、数据、密钥等都已放入双端口 RAM 中。运算时可直接取出 Nounce，加上 Flag、Dlen、cnt 等，组成校验和加密所需的初始向量。如图 6、7 分别为加密和解密过程的流程。

由图可知，加密和解密都只用到了 AES 加密算法，而没有用到 AES 解密算法。AES-CCM 的解密过程几乎是加密过程的重复，不同的只是某些时序。加

978-1-935068-06-8 © 2009 SciRes.

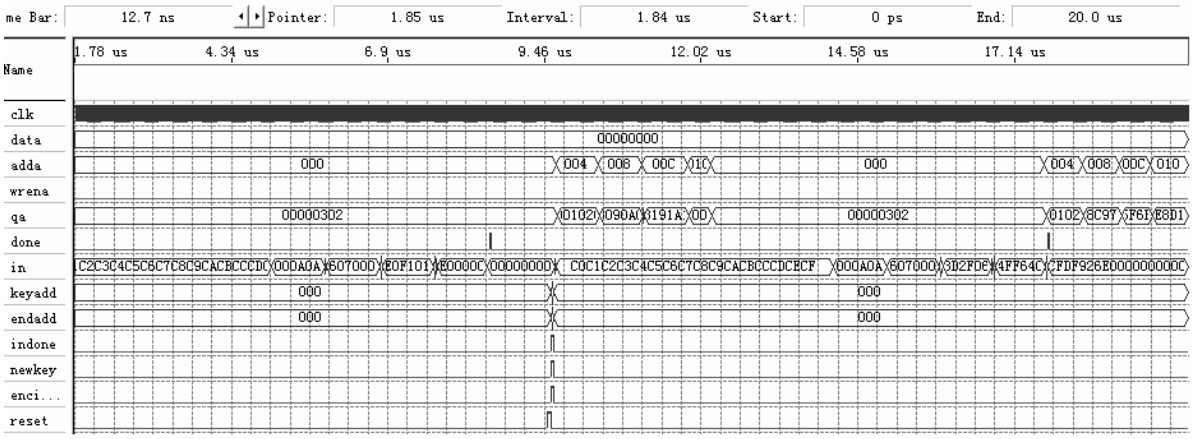

Figure 9. AES-CCM Timing Simulation

图 9. AES-CCM 时序仿真

密流程中，CBC-MAC 和 CTR 的异或运算可以同步进行，而在解密流程中，CBC-MAC 的异或运算需等 CTR 运算解出明文后才能进行。

由于 AES-CCM 加密和解密流程的相似性，因而没有必要为加密和解密建立各自独立的模块。也就是说，加密和解密可在同一模块中进行，只需提供一定的片选信号和两套不同的控制微指令即可。用简图表示如图 9。

将加密和解密整合在同一模块中的实现，无疑节省了大量的资源。虽然这样会使控制信号变得复杂，某些微指令位在加密和解密时也会有不同的意义，但只要建立两套互相独立的微指令寄存器，并利用外部给的加/解密信号控制某些信号意义的转换，就可以很方便的在加密和解密运算之间进行转换。

5 时序仿真

本文用 Quartus 软件和 Altra 公司的 EP1S20F672C7FPGA 芯片作为开发仿真环境，对 AES-CCM 设计进行编译、仿真。

如图 9，是对 AES-CCM 依次进行一次解密和一次加密仿真的结果。先写入一段密文，密钥扩展后进行解密。然后在解得明文的基础上再进行一次加密，结果仍回写入 RAM 中。系统时钟为 50M。仿真中采用的测试量如下：

密钥：C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CBCC CD CE CF ；

Nounce：TA：A0 A1 A2 A3 A4 A5

PN： 00 00 00 03 02 01 00 ；

帧头：00 01 02 03 04 05 06 07；

数据：08090A0B 0C0D0E0F10111213141516171819A 1B1C1D1E；

密文：588C979A61C663D2F066D0C2C0F989806D5F6B61DAC384

MIC：17 E8D12CFDF926 E0

本设计中，默认 Nounce 参量放在 RAM 首地址中，后面依次存放帧头和数据包，在最后一个数据分组之后存放 MIC 值。对解密后的从 RAM 的 32 位端口读取的值和再次加密后 RAM 内容的比较，对应读取的分别是解密后 PN、帧头、第一分组、第二分组和 MIC 地址内的前四字节；并且对应于将解密后数据再次加密所得的结果，同样也是读取各地址的内容。与理论值相比较，结果完全正确。

6 仿真结果分析

为了了解该 AES-CCM 的设计成效，本文结合顺序循环 AES 方式进行以下几个参数的对比。

6.1 资源消耗的比较

FPGA 的芯片资源包括逻辑单元(logic elements)、引脚数（pins）、存储位（memory bits）、DSP 块、PLL 和 DLL 等。DSP、PLL、DLL 在本设计中都没有用到，而引脚数因为这两种设计所使用的接口都相同，也无需比较。所以资源消耗的比较主要只需针对逻辑单元和存储位。逻辑单元即实现设计中使用到的各模块的逻辑描述所需耗费的逻辑门，存储位即设计中 RAM 和 ROM 所使用的 bit 位数。对 EP1S20F672C7，其逻辑单元数为 18460，存储位为 1669248bits。

对于顺序循环 AES 方式，其使用逻辑单元数、存储位分别为 3138、116847bit；对于 AES-CCM 方式，其使用逻辑单元数、存储位分别为 3322、100154bit。

由以上比较结果可知，顺序循环 AES 方式包括加

密和解密两部分，而 AES-CCM 包括 CBC-MAC 和 CTR 两部分，所以这两种实现方式消耗的资源量比较接近。AES-CCM 因为要多一些异或模块和生成参量的模块，所以逻辑单元的使用比顺序循环 AES 方式多；而 AES-CCM 的两部分使用了同一个 roundkey 模块，节省了 RAM 空间，所以存储位的使用又比顺序循环方式少。

6.2 速度的比较

在顺序循环 AES 方式中，完成一个分组的 AES 运算需要 44 个时钟周期，从而得到平均吞吐量为 142.2Mbit/S；在 AES-CCM 中，完成一个长度为 128 分组的 AES 运算需要 45 个时钟周期，对于长为 n 分组的数据（包括帧头），一共要进行 n＋1 次 AES 运算，计算得吞吐量为

$$\frac{128 \times n}{20 \times 45 \ (n+1)} \times 10^{3}$$

Mbit/S。可见数据越长，平均吞吐量越大，最高吞吐量可达 145Mbit/S。

AES-CCM 实现方式由于要多运算一个 AES 周期来计算校验码 MIC，吞吐量也受到了一定限制，但总的来说，已经超过了 802.16 协议理论上的最大传输速率 75Mbps。这两种方式都可以应用于安全 WiMAX 中。

为了进一步提高系统的加解密速度，可以采用更高的系统时钟频率，以及采用多级流水线的方式。

6.3 安全性的分析

顺序循环方式是在 ECB 模式下进行的，这种模式虽然简单，但只适用于短信息的传送，不能提供足够的安全性。AES-CCM 是在 CBC-MAC 认证模式下进行，无疑提供了最好的安全性，它具有分组序号的初始向量，增强了数据的可靠性，可以更好的防御外部的攻击。

7 结束语

本文给出了一种基于 FPGA 的 AES-CCM 设计和实现方案。它在由 Quartus 软件和 FPGA 芯片组成的开发仿真环境下，采用由上至下的设计方法，利用 Verilog 语言设计了 AES-CCM 的硬件实现。该设计方案具有消耗资源少，速度快，安全性能高的特点，已经成功的应用在基于 802.16e 安全协议的无线局域网移动客户端和 AP 端产品中。

References (参考文献)

[1] Chen Jyh Cheng, Jiang Ming Chia. Wireless LAN Security and IEEE 802.11i. IEEE Wireless Communications, 2005, 12(1): 27～36.

[2] William Stallings W. Cryptography and Network Security Principles and Practices [M] .Boston: Academic Press, 2004.

[3] Dawson Ed, Gustafson Helen. Evaluation of RC4 Stream Cipher. Information Research Centre, Queensland University of Technology. July 31, 2002.

[4] IEEE 802.16e-2005. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems [S].Dec 2005: 18-300.

978-1-935068-06-8 © 2009 SciRes.