Computer Science and Application 计算机科学与应用, 2022, 12(5), 1389-1397
Published Online May 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.125138

Hans汉斯

基于FPGA的AES加密算法优化设计

刘政文，赵曙光

东华大学信息科学与技术学院，上海

收稿日期：2022年4月26日；录用日期：2022年5月24日；发布日期：2022年5月31日

摘　要

现如今随着网络技术的不断发展，网络设备的数据处理速度变得越来越快，与此同时，信息安全相关的问题也逐渐显现了出来。针对高速网络信息的报文加密的优化问题，提出了一种AES加密的优化方法。通过对明文的识别、分组加密和替换，有效地隐藏了明文序列，提高了网络报文的保密性和安全性。经测试，优化后的AES加密算法可以正确有效地完成高速网络报文的加解密部分的工作。

关键词

FPGA，AES加密算法，信息安全

Optimization Design of AES Encryption Algorithm Based on FPGA

Zhengwen Liu, Shuguang Zhao

College of Information Science and Technology, Donghua University, Shanghai

Received: Apr. 26th, 2022; accepted: May 24th, 2022; published: May 31st, 2022

Abstract

Nowadays, with the continuous development of network technology, the data processing speed of network equipment is becoming faster and faster. At the same time, the problems related to information security have gradually emerged. Aiming at the optimization problem of packet encryption in high-speed network, an optimization method of AES encryption is proposed. Plaintext identification, packet encryption and replacement can effectively hide plaintext sequence and improve the confidentiality and security of network messages. After testing, the optimized AES encryption algorithm can correctly and effectively complete the encryption and decryption of high-speed network packets.

文章引用: 刘政文, 赵曙光. 基于 FPGA 的 AES 加密算法优化设计[J]. 计算机科学与应用, 2022, 12(5): 1389-1397.
DOI: 10.12677/csa.2022.125138

Keywords

FPGA, AES Encryption Algorithm, Information Security

1. 引言

目前计算机技术与网络技术都在迅猛地发展，网络数据的交互也日趋频繁与复杂。随着网络电子设备的不断更新升级，用户在享受着网络带来的便利时也开始对网络安全、信息窃取等问题感到担忧[1]。维护信息安全的方法中，一个重要的方法是使用密码学中的加密算法，对传输的数据进行替换，使用密钥序列和数据变换方法，将明文隐藏起来，从而对重要信息进行保护，增强数据的安全性和保密性[2]。

高级加密标准(Advanced Encryption Standard, AES)加密算法是分组加密算法的一种，是目前美国联邦政府采用的一种加密技术[3]。近年来，许多学者都对 AES 算法加以研究。Soltani Abolfazl 提出以电子密码本和计数器模式提高 AES 的吞吐量，并结合内存和非内存的方法实现算法中的 S 盒结构，提高 AES 在应用时的性能[4]。Syed Shahzad Hussain Shah 提出了一种基于混沌的 AES 算法实现架构，由混沌映射生成密钥，同时使用并行 RAM 实现字节替代环节，以优化 AES 算法的执行结构[5]。戴强等人提出了改进后的 CSE 算法对 AES 算法中 S 盒的电路结构进行优化[6]。

针对高速网络报文的信息安全问题，本文提出了一种 AES 加密算法的设计方案，将网络报文中的数据识别分组并进行加密替换以隐藏明文。同时优化了 AES 算法使用时的工作流程和工作模式，使 AES 算法能够有效地应用在高速网络报文的场景中，提高网络报文的保密性和安全性。

2. AES 算法原理

AES 加密算法是分组加密算法中的一种，它是在 Rijndael 加密算法的基础上，做出了一定程度的简化而设计的[7]。AES 规定了加解密流程中的数据和密钥的长度，每个分组的数据长度为 128 bit，而密钥的长度可以设置为 128 bit，192 bit 或是 256 bit，然后依据不同的密钥长度，算法执行相应的迭代轮数。

2.1. 算法描述

AES 加密算法的每一步都是作用于矩阵的形式进行变换，每次输入的数据为 128 bit，以字节为单位分成 16 个字节，将其排列为 4*4 的矩阵，AES 的每一轮变换都是由四个步骤组成，分别是字节替代，行移位，列混淆和轮密钥加，AES-128 加密算法的流程图如图 1 所示。

AES-128 算法在加密迭代前使 128 bit 的明文数据与初始密钥进行一次异或再进入迭代程序，加密时每次迭代时每轮有四个步骤，即字节替代、行移位、列混淆和与扩展好的密钥异或计算，最后一轮加密时移除列混淆步骤，最终输出的结果即为 128 bit 的密文数据。解密时的步骤与加密时相反，密文输入先与密钥扩展的最后 4 个双字密钥异或后启动解密算法迭代，每次迭代依次进行四个解密步骤，分别是逆列混淆、逆行移位、逆字节替代和轮密钥加，完成所有的迭代轮数就可以完成解密算法与输出 128 bit 的明文数据[8]。

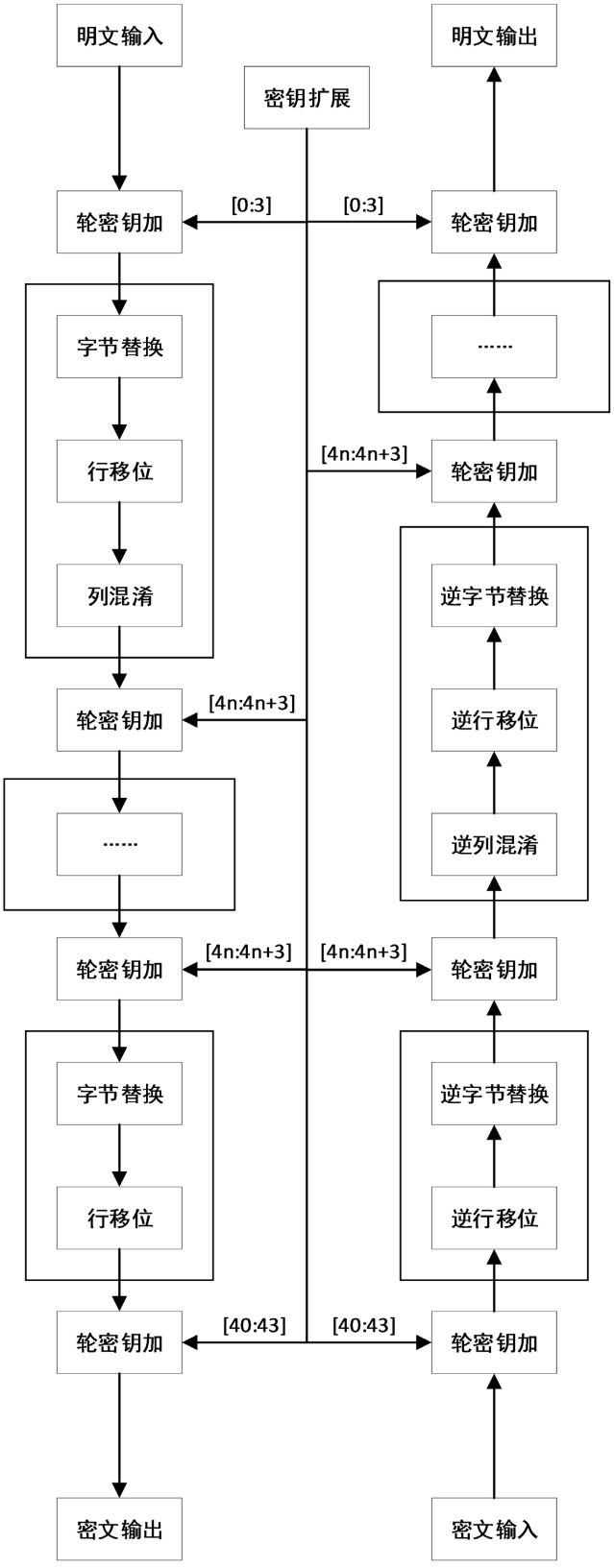

Figure 1. AES-128 algorithm flow chart
图 1. AES-128 算法流程图

2.2. 字节替代

字节替代是一种非线性变换，将一个 8 bit 的数据通过变换用另一个与之唯一相互对应的另一个 8 bit 的数据替换掉原始数据，并且要求能够通过逆变换使新数据恢复成原始数据。

AES 加密算法定义了 S 盒和逆 S 盒，两个表均占用了 16*16 的存储空间，用于完成算法加密和解密中的字节替代工作。S 盒使用查表法的方式执行，旧数据以 8 bit 为单位输入，高 4 bit 数据作为行系数查表，低 4 bit 作为列系数查表，以此将输入的数据变换为 8 bit 的新数据输出，即为字节替代的结果。S 盒内容如图 2 所示。

	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	1	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	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	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	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	6	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	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	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 2. S-box
图 2. S 盒

2.3. 行移位

行移位变换是加密算法对输入数据矩阵以每一行为单位做移位变换，矩阵的第一行做变换，第二行到第四行分别左移 1~3 字节。行移位的操作不会改变每个字节的数值，而是将矩阵的每一列重新对齐，使数据排列更加的无序，增强了加密算法的复杂度。在解密过程中使用的是逆行移位变换，同样第一行不做变换，2~4 行分别右移 1~3 字节。行移位变换步骤模型如图 3 所示。

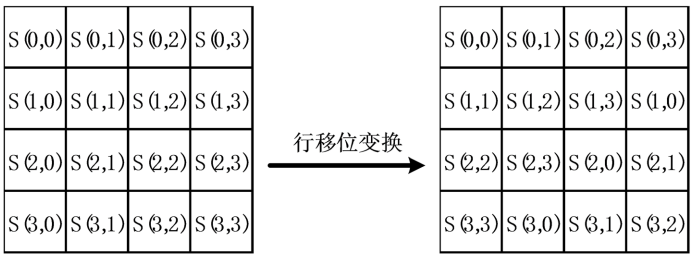

Figure 3. Schematic diagram of row shift transformation
图 3. 行移位变换原理图

2.4. 列混淆

列混淆是对数据矩阵进行列方向的变换，变换的矩阵乘法表达示如下：

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

式中矩阵里的每个元素都是对应位置元素的乘积之和，其中矩阵中的乘法与加法均为基于有限域 GF(2^8)上的二元运算。在 GF(2^8)中加法运算表示为不进位的异或运算，乘法运算为则是对移位、异或、取模的结合，在 GF(2^8)中取模运算的标准为 $M(X)=x^8+x^4+x^3+x+1$。在解密过程中，逆列混淆的操作不变，仅将解密中间矩阵替换为加密的逆矩阵。逆列混淆表达式下：

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

2.5. 轮密钥加

轮密钥加时将准备好的数据与经过密钥扩展提取后的密钥按位进行异或运算，运算的结果作为下一次加密迭代的输入，或是在最后一轮直接成为最终的输出。轮密钥加的过程减弱了明文与密文之间的直接联系，增强了数据与密钥之间的联系，使破解密码的难度大大提升。

2.6. 密钥扩展

AES 算法每一轮都需要使用密钥与数据进行异或运算，所以输入的密钥长度是不足够算法所使用的，在迭代的过程中相应的要对密钥进行扩展工作。AES-128 的密钥扩展算法原理图如图 4 所示。

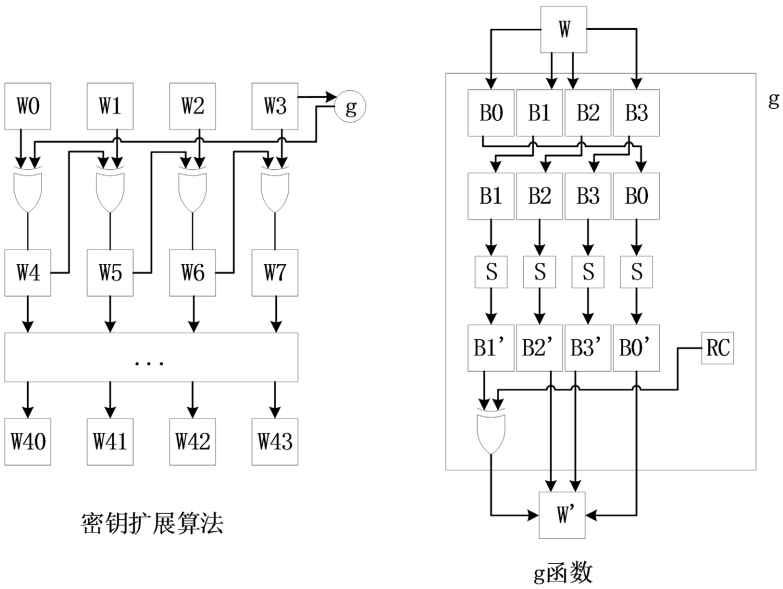

Figure 4. Key extension schematic diagram
图 4. 密钥扩展原理图

将密钥以 1 个字节为一组分割成 16 个部分，排列组成一个 4*4 的矩阵。每一列用 4 维向量所表示，注为[w0, w1, w2, w3]，其中位于 w3 位置的 4 字节数据需要进行一次 g 函数的运算，g 函数会输入进来 4 字节的数据，分成四份，每份一个字节，4 字节的数据排列好后左循环 1 字节，并带入 S 盒中进行字节替代，经过以上流程 w3 替换成了 w’。生成的 w’输入进原矩阵依此进行一系列的异或运算以生成本轮的密钥。

3. AES 模块设计与分析

为了将 AES 加密算法的数据处理速度能够与高速网络报文传输速度相匹配，需要合理设计算法内部执行模块的连接结构，以保证模块内部的数据流不会丢失或是存在逻辑冲突，使数据流能够互不影响的完整各自的阶段性变换。AES 工作流程的设计还要兼顾占用资源的多少以及是否容易实现的问题。

AES-128 加密算法中，每个加密分组要迭代 10 次，若采用环展开结构在一个时钟周期完成 10 次迭代的全部流程，需要设计出一个巨大的组合逻辑电路，对 FPGA 内部电路，造成巨大的时序压力，性价比不高，因此需要通过流水线的结构完成 AES 中迭代算法的连接。流水线化的设计是对于比较复杂的结构进行分解，分成多个层次分成多个周期执行，这种设计模式适用于同一组程序需要反复执行的情况。流水线化的设计在每个层级都要设置寄存器来暂存数据的状态，在下个时钟周期会被下一个层级抽取，而上一个层级同时也会被新的数据填充，像流水线一样将顺序程序一层一层的串联起来，并以此完成数据的并行处理。AES-128 算法流水线化的设计中，第一组数据输入进模块完成一轮加密，下一个周期这一组数据进行第二轮加密，同时第二组数据输入进模块完成第一轮加密，经过十个周期的迭代，此时第一组数据完成了 AES 的加解密，而第二组数据的加解密流程也即将完成。流水线结构完成 AES 加密的结构示意图如图 5 所示。

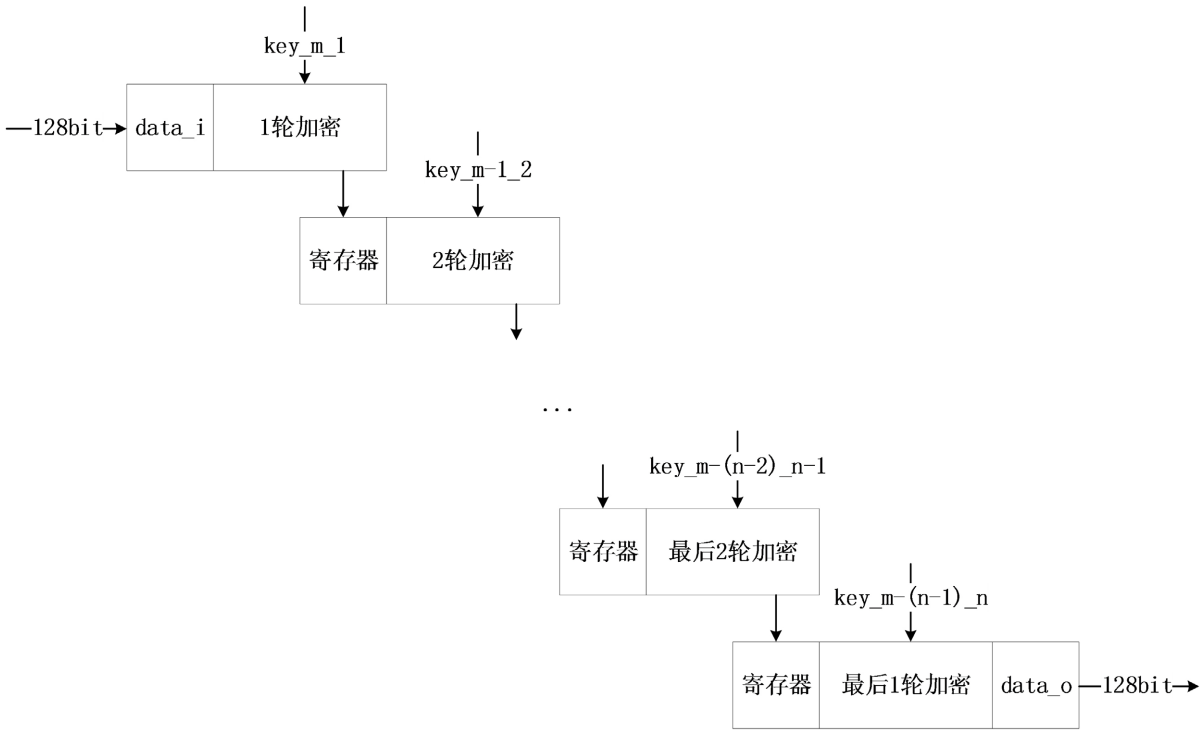

Figure 5. AES-128 pipeline structure design
图 5. AES-128 流水线结构设计

AES-128 的输入模式选取了计数器(Counter, CTR)模式，CTR 模式支持 AES 算法的流水线结构设计，使 AES 加密算法的加密速度可以与高速网络报文的数据处理效率所匹配。CTR 模式中加密数据由计数器生成，加密数据通过 10 次迭代加密生成加密的结果，此时取出网络报文的明文数据，与加密结果异或，得到密文数据，在解密操作中，同样需要通过计数器生成一个相应额序列，并使用 AES 加密算法进行解密，加密后的结果与密文异或即可生成网络报文明文。使用 CTR 加密模式的优点是有效解决了报文拉长的问题，AES 算法的最小单位是 128 bit，不能保证网络报文长度是 128 bit 的倍数，使用 CTR 模式，网络报文数据只在加密结束时参与加密计算，多余的密文或原文长度的舍弃不影响加解密的结果，有效的解决了报文拉长的问题[9]。CTR 模式的 AES 加解密算法示意图如图 6 所示。

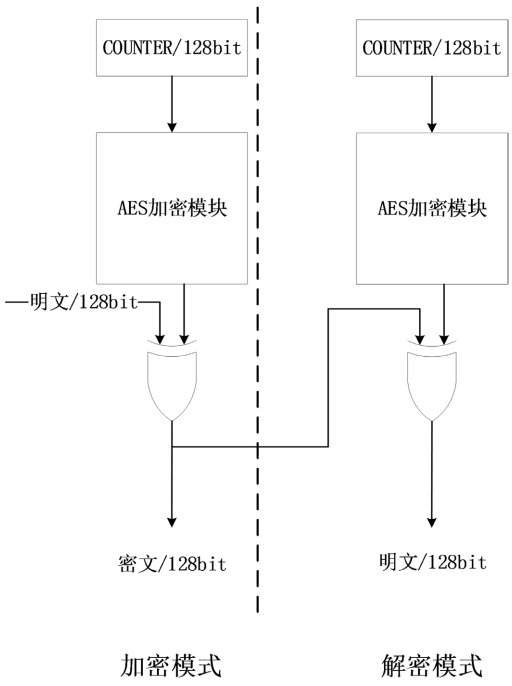

Figure 6. CTR mode AES encryption and decryption algorithm
图 6. CTR 模式的 AES 加解密算法

4. AES 算法的仿真

本文使用了 Verilog HDL 硬件描述语言在 FPGA 上对 AES 加解密流程进行了设计，并使用了 ModleSim 10.4 仿真软件来验证程序逻辑的正确性。

对 AES-128 模块的加密逻辑进行验证时，要输入初始的密钥和待加密的数据，再由系统自动扩充密钥，将密钥与待加密的数据进行迭代运算，最终输出加密好的密文，加密仿真图如图 7 所示。

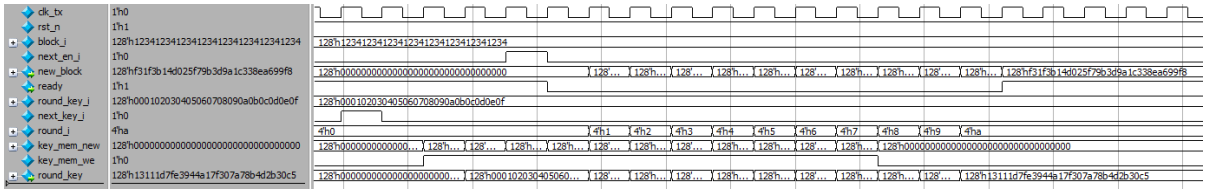

Figure 7. AES-128 encryption simulation diagram
图 7. AES-128 加密仿真图

仿真激励将初始明文信号设为 0x12341234_12341234_12341234_12341234，也给出了 128bit 的初始密钥 0x00010203_04050607_08090a0b_0c0d0e0f，用来对明文进行加密。128 bit 的初始密钥要对 AES 模块完成 10 轮的加密迭代，在迭代之前要对初始明文和初始密钥先完成一次轮密钥加的运算，之后在迭代中要分别完成字节替代、行移位变换、列混淆变换和与扩展后的密钥完成轮密钥加变换，除最后一轮不必完成列混淆变换。仿真图表明经过 10 轮迭代后得到最终的密文序列为 0xf31f3b14_025f79b3_3d9a1c33_8ea699f8。使用第三方软件设置相同的初始状态，以对 AES 加密模块的功能逻辑正确性进行验证，第三方的加密结果如图 8 所示。通过将 AES 加密模块的结果和第三方的 128 bit 输入密钥的 AES 加密结果进行对比，两者的解密结果是完全一样的，可以证明 AES 加密功能是正确的。

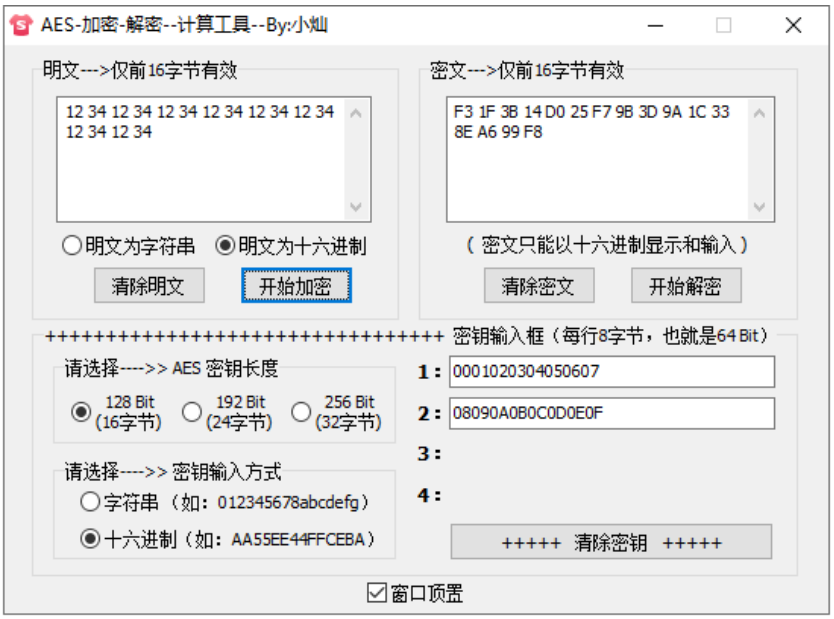

Figure 8. Third-party software verifies the AES encryption algorithm
图 8. 第三方软件验证 AES 加密算法

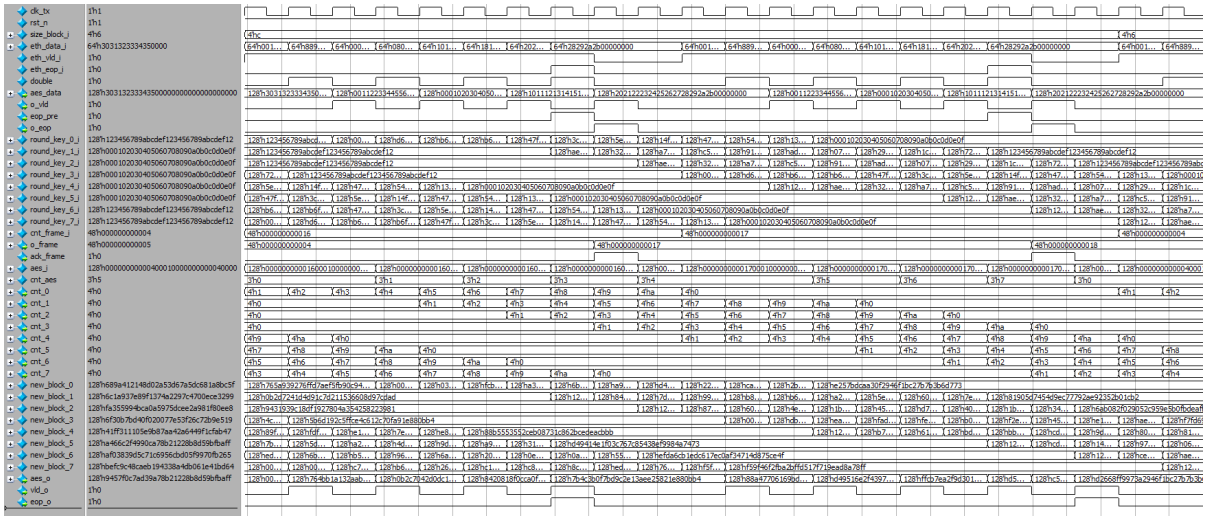

Figure 9. Network packet encryption process simulation diagram
图 9. 网络报文加密流程仿真图

将经过优化后的 AES 加密流程应用在 10 G 以太网的环境中，系统搭建了仿真平台验证网络报文高速加解密功能的有效性。在设计高速加解密的流程时，先要每两个发送周期采集一次 10 G 以太网帧格式的网络报文，生成 128 bit 的明文序列，再根据报文的发送状态生成 CTR 工作模式下的 AES 加密模块的输入，完成加密后序列与 128 bit 的明文相加，即可得到密文序列，生成的 128 bit 密文序列再按时序进行分解，变成每个周期 64 bit 的密文序列，再重新校验以及组装成以太网报文的格式，即完成了对网络报文进行高速加解密的功能。网络报文加密流程仿真图如图 9 所示。

图中使用了仿真激励模拟生成了以太网报文待加密的数据的时序逻辑，将网络报文内容输入进模块中，最终以流水线的形式得到加密好的 128 bit 密文序列。通过仿真实验模拟了真实连续发送高速网络报文的情况，验证了连续工作状态下 AES 加密模块的工作状态，经过多次以及多种情况的仿真实验，实验结果表明模块能够正确有效的完成高速网络报文的加解密部分的工作，证明了设计模型是先进且可靠的。

5. 总结

本文通过对 AES 加密算法的输入模式和工作流程进行了优化，使其能够应用于高速网络报文的加密和解密工作，完成了网络报文的明文提取加密和密文替换报文内容的工作，为高速网络报文的信息安全工作提供了解决思路。本文通过仿真实验模拟了算法在 10 G 以太网报文中的表现结果，证明了方案的有效性。

参考文献

[1] 李炽阳, 雷倩倩, 杨延飞. 全通用 AES 加密算法的 FPGA 实现[J]. 计算机工程与应用, 2020, 56(10): 83-87.

[2] 窦贤振. 基于 FPGA 的 AES 加密算法设计与实现[J]. 科技风, 2019(36): 64.

[3] 王常磊. 基于 FPGA 的数据加密算法设计与实现[D]: [硕士学位论文]. 哈尔滨: 黑龙江大学, 2020.

[4] Soltani, A. and Sharifian, S. (2015) An Ultra-High Throughput and Fully Pipelined Implementation of AES Algorithm on FPGA. *Microprocessors and Microsystems*, **39**, 480-493.

[5] Shah, S.S.H. and Raja, G. (2015) FPGA Implementation of Chaotic Based AES Image Encryption Algorithm. 2015 *IEEE International Conference on Signal and Image Processing Applications* (*ICSIPA*), Kuala Lumpur, 19-21 October 2015, 574-577. https://doi.org/10.1109/ICSIPA.2015.7412256

[6] 戴强, 戴紫彬, 李伟. 基于增强型延时感知 CSE 算法的 AES S 盒电路优化设计[J]. 电子学报, 2019, 47(1): 131-138.

[7] 钱浩. 基于 FPGA 的千兆网络安全通信研究与实现[D]: [硕士学位论文]. 杭州: 杭州电子科技大学, 2017.

[8] 赵新杰, 郭世泽, 王韬, 刘会英. 针对 AES 和 CLEFIA 的改进 Cache 踪迹驱动攻击[J]. 通信学报, 2011, 32(8): 101-110.

[9] 阮景林, 刘林. GPON 中的 AES 加密[J]. 中国集成电路, 2007(12): 32-35.