

KASUMI 算法的芯片设计与实现

许静雯, 李树国

(清华大学 微电子学研究所, 北京 100084)

摘 要: KASUMI 算法是第三代移动通信系统(3G)使用的一种加密算法. 提出一种二合一结构来实现 KASUMI 算法的芯片设计. 依据此二合一结构, KASUMI 算法由原始的八轮循环迭代减少为四轮, 从而降低了时钟周期数, 提高了吞吐量. 同时, 该设计还实现了 f8 加密和 f9 完整性验证两种模式. 基于 SMIC 0.13 μm CMOS 工艺的综合结果表明, KASUMI 算法的时钟频率为 123.4 MHz, 面积为 13 979 μm^2 , 吞吐率为 1.97 Gb/s, 与其他同类方法相比, 提高了 23%.

关键词: KASUMI; f8; f9; 二合一

中图分类号: TN49

文献标识码: A

文章编号: 1000-7180(2015)11-0074-04

Design and Implementation of KASUMI Algorithm in Chip

XU Jing-wen, LI Shu-guo

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: KASUMI algorithm is an encryption algorithm used in the third generation mobile communication system (3G). In this paper, we propose a two-in-one architecture to complete the design of KASUMI algorithm in chip. Based on the architecture, the iteration of the loop in KASUMI algorithm reduces from original 8 rounds to 4 rounds. Therefore, the number of clock cycles of the design is reduced and the throughput is improved. Additionally, the design still implements the encryption of f8 mode and the integrity verification of f9 mode. Using SMIC 0.13 μm CMOS technology, clock frequency of this design and the count of the gates are 123.4 MHz and 13979 gates respectively. Most importantly, the throughput achieves 1.97 Gbps, which is 23% faster than previous results.

Key words: KASUMI; f8; f9; 2-in-1

1 引言

KASUMI 算法是第三代移动通信系统中一种标准化核心加密算法, 它的前身是日本三菱公司的分组密码 MISTY1 算法, KASUMI 算法是 MISTY1 算法的改进版本. KASUMI 算法设计简单, 软、硬件实现快, 在硬件实现上功耗低、门电路少, 同时在设计上具有对抗差分分析和线性密码分析的可证明的安全性^[1]. KASUMI 算法在第三代移动通信系统中以 f8 和 f9 两种模式使用^[2], 其中 f8 是加密算法, 用来对 UE(用户设备)和 RNC(无线网络控制器)之间通过

无线接入链路传输的用户数据和信令数据加密, 以保证它的安全性. f9 是完整性验证算法, 用来保证信令数据在传输过程中的完整性, 防止非授权方随意更改信令数据.

2002 年 Akashi Satoh 等人提出了三种结构来实现 KASUMI 算法, 其中第一种结构复用了 FL 函数、九进九出的 S9 和七进七出的 S7 两个 S 盒; 第二种结构复用了 FL 函数和 FI 函数; 第三种结构复用了 FL 函数和 FO 函数, 同时设计了 f8 和 f9 两种模式的数据通路^[3]. 2004 年 Tomas Balderas-Contreras 等人提出了复用两个 FL 函数和一个 FO 函数

收稿日期: 2014-12-29; 修回日期: 2015-01-30

基金项目: 国家“八六三”计划项目(2012AA012402); 清华大学自主研发计划(2011Z05116)

的结构实现了 KASUMI 算法^[4].但是,以上方法在每次计算开始前都需要一个二选一的多路选择器来判断首先进行线性混合 FL 函数计算或是非线性混合 FO 函数的计算.

本文提出一种循环展开二合一的结构来实现 KASUMI 算法,这样节省了每轮计算前二选一的多路选择器,并且计算每个 64 bit 的消息分组只需要 4 个时钟周期.另外,本文的电路设计还实现了 f8、f9 两种模式和随时中断暂停功能.

2 KASUMI 算法原理概述

2.1 KASUMI 算法概述

KASUMI 算法是 Feistel 结构的分组加密算法,密钥是 128 bit,加密过程中对 64 bit 的输入,经过 8 轮迭代产生 64 bit 的输出. KASUMI 算法加密过程如图 1 所示.具体实现如下:输入数据 I 被分为 32 bit 的左右两部分 L_0 和 R_0 ,从第一轮到第八轮,每一轮的输出定义为 $R_i = L_{i-1}$, $L_i = R_{i-1} \oplus f_i(L_{i-1}, RK_i)$,经过八轮的计算后,得到 64 bit 输出.

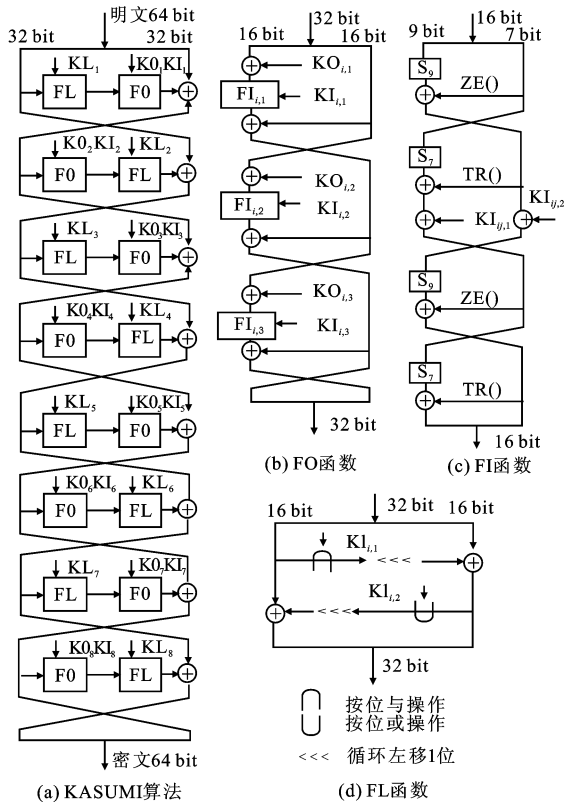


图 1 KASUMI 算法原理图

KASUMI 算法的轮函数包括线性混合 FL 函数和非线性混合 FO 函数,FO 函数由非线性混合 FI 函数进行三轮迭代计算构成,而 FI 函数是由非

线性的 S 盒构成的四轮迭代结构. FO 函数原理如图 1(b)所示, FI 函数原理如图 1(c)所示, FL 函数原理如图 1(d)所示,详见参考文献[5].需要注意的是,轮函数 f_i 在不同的奇偶轮有不同的表达形式.

对于 1、3、5、7 轮:

$$f_i(\text{indata}, RK_i) = \text{FO}(\text{FL}(\text{indata}, KL_i), KO_i, KI_i).$$

对于 2、4、6、8 轮:

$$f_i(\text{indata}, RK_i) = \text{FL}(\text{FO}(\text{indata}, KL_i), KO_i, KI_i).$$

其中 KL_i , KO_i 和 KI_i 是子密钥,是由密钥扩展模块产生的.密钥扩展模块对 128 bit 密钥 K 进行操作,为每轮迭代生成 128 bit 的子密钥.

2.2 f8、f9 模式概述

f8 是一个密钥流加密算法,它通过 KASUMI 算法生成 64 bits 为一组的密钥流,将明文数据流与密钥流进行异或(模 2 加)运算,得到密文流,解密操作与加密操作相同,只要将同样的密钥流与密文流进行异或(模 2 加)运算,即可得到明文数据流. f8 模式的原理图如图 2(a)所示.

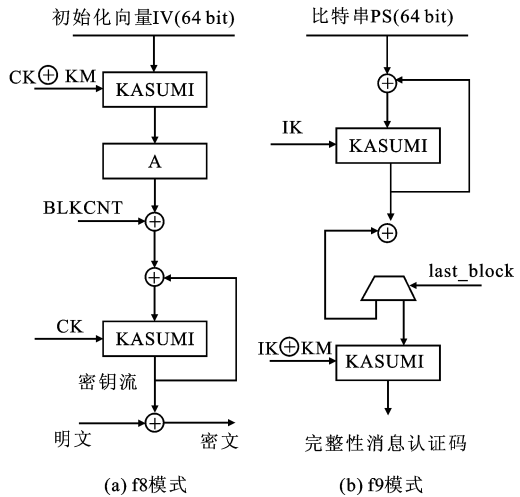


图 2 f8、f9 模式原理图

f9 完整性验证算法是根据现代密码学中认证和鉴权的基本原理,采用哈希(HASH)函数来实现. f9 模式对输入消息的长度没有限制,它的输出是 32 bits 的消息认证码 MAC-I. f9 模式的原理图如图 2(b)所示. f8 和 f9 模式具体原理详见参考文献[6].

3 KASUMI 算法芯片实现

本文的设计旨在实现 KASUMI 算法,并可以应用于 f8、f9 两种模式,同时电路支持随时中断暂停功能. KASUMI 算法的结构框图如图 3 所示.

输入信号 mode 是计算模式的标志位. 密钥 CK

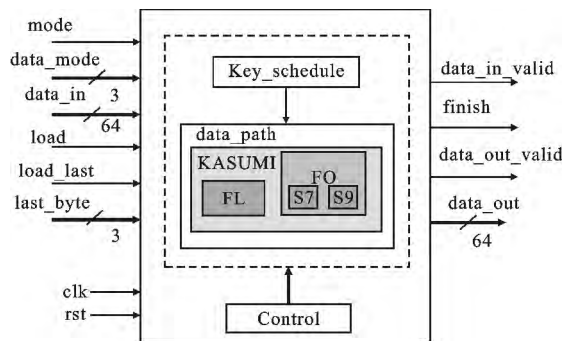


图3 KASUMI算法实现结构框图

或IK、初始向量IV、COUNT-I等输入参数复用64 bits的data_in输入端口,输入信号data_mode是输入数据类型的标志位。

3.1 KASUMI算法实现方法

KASUMI算法的实现策略一般有三种方式,分别是迭代方式、流水线方式和循环展开方式。

迭代方式是将单轮迭代的结果重新反馈到输入端进行计算,这种方式节约资源,但是延时较大,效率较低。

由于上文中所述,KASUMI算法奇数次轮函数和偶数次轮函数中的线性混合FL函数计算和非线性混合FO函数计算次序不同,如果采用单轮迭代方式,那么每次进行KASUMI算法的计算时,需要判断当轮迭代是奇数次还是偶数次迭代,也就是说每次64 bit的消息分组进行KASUMI计算会增加8次二选一操作,这会减慢计算速度,文献[3-4]中提出的KASUMI算法实现方法属于这种迭代方式。本文不采用单轮迭代方式。

流水线方式是提高吞吐率的常用手段,流水线方式将组合逻辑电路分为几个级,在每一级插入寄存器组并暂存中间数据。这种方式每个时钟均有输出,吞吐率高。但是流水线方式在性能上的提高是以消耗较多的寄存器资源为代价的。KASUMI算法的f8和f9模式分别应用了输出反馈模式和密码分组连接模式,所以KASUMI算法无法使用流水线方式来实现。

循环展开方式是将算法的多步运算合并成单步运算,与单轮迭代相比,循环展开方式占用资源多,但是加大了并行计算的规模,从而提高了单步运算的效率。由于KASUMI算法包含八轮轮函数,所以循环展开方式有三种实现方案,分别是两轮循环展开、四轮循环展开和八轮循环展开,以下简称为二合一、四合一和八合一结构。而循环展开四合一和八合一的结构是二合一结构的单纯重复,四合一和八合

一结构的吞吐率会增大,同时面积也会翻倍,本文不采用四合一和八合一的实现结构。

二合一结构的组合逻辑中包含了奇数次和偶数次轮函数的计算,这样就节省了文献[3-4]提出的方法中每轮迭代计算前的二选一多路选择器。

同时,二合一结构节省了每个64 bit的消息分组计算所需的时钟周期数。文献[3]中的三种结构每次计算64 bit的消息分组分别需要56、32和8个时钟周期;文献[4]中的结构每次计算64 bit的消息分组需要8个时钟周期,如图4(a)所示。而二合一结构每次计算64 bit的消息分组时,将原来所需的8个时钟周期减少到4个时钟周期,如图4(b)所示。综上所述,考虑面积和吞吐率,本文采用二合一的结构实现KASUMI算法,二合一结构的具体实现见图4(c)。

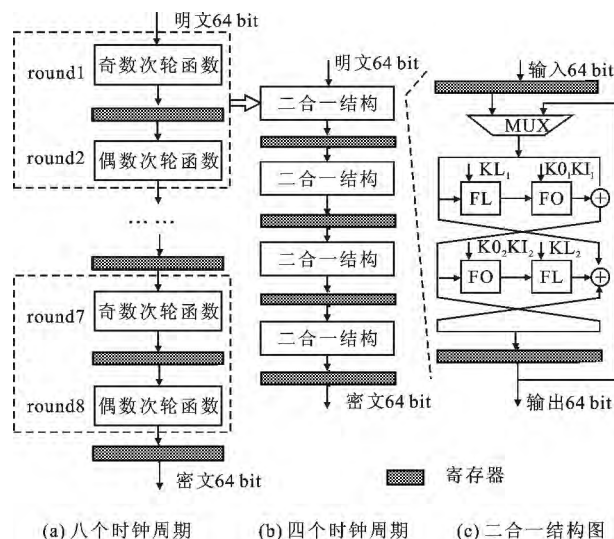


图4 KASUMI算法二合一结构图

3.2 KASUMI算法轮函数及密钥扩展部分

如图4(c)所示,KASUMI算法结构中包含线性混合FL函数和非线性混合FO函数。

线性混合FL函数由循环移位、按位与、按位或和异或操作构成,结构简单,由纯组合逻辑构成。

非线性混合FO函数中包含了6次七进七出的S盒和6次九进九出的S盒计算,可见S盒的计算速度大大影响了KASUMI算法的计算速度。S盒一般由查表法或表达式法实现。综合结果表明,表达式法对比查表法,表达式法占用面积小,延时短,所以本文采用表达式法来实现KASUMI算法中的S盒。S盒的具体表达式见参考文献[5]。

KASUMI算法每轮使用的八个子密钥 $KL_{i,1}$ 、 $KL_{i,2}$ 、 $KO_{i,1}$ 、 $KO_{i,2}$ 、 $KO_{i,3}$ 、 $KI_{i,1}$ 、 $KI_{i,2}$ 和 $KI_{i,3}$ 是由

128 bit 的密钥 K 扩展而成的. 密钥 K 被分为每组 16 bit 的 8 个组: $K = K_1 || K_2 || K_3 || K_4 || K_5 || K_6 || K_7 || K_8$, K_j' 根据密钥 K_j 来生成: $K_j' = K_j \oplus C_j$, C_j 是 16 进制的常量, 每轮的子密钥由 K_j 和 K_j' 按照文献[5]定义的规则生成. 文献[3]中采用移位寄存器产生子密钥的值. 本文则分别写出了每轮子密钥与 128 bit 输入密钥之间的表达式, 并在硬件中使用连线实现, 其延时与其他逻辑相比可以忽略.

3.3 f8、f9 模式设计

本文实现的设计支持 f8 和 f9 两种模式, f8、f9 模式下数据通路如图 5 所示.

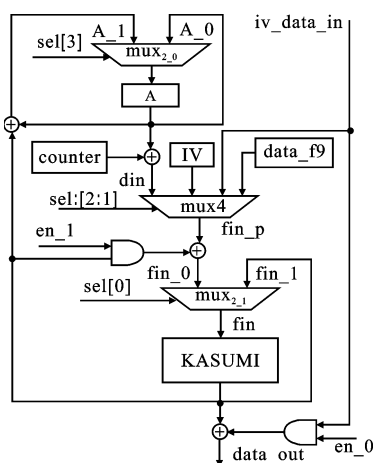


图 5 f8、f9 模式数据通路

f8 模式的控制模块由状态机实现, 它包含两部分, 首先进行预白化操作, 即初始向量 IV 代入一个 CK 修正模式下的 KASUMI 算法, 计算出预白化值 w , w 被存入寄存器 A , 当 f8 以这种方式初始化后, 就可以产生所需要的密钥流. 寄存器 A 中的值与计数器 counter 的值异或后通过多路选择器 mux4, 上轮 KASUMI 算法的结果与 en_1 做按位与操作后与 mux4 的输出进行异或, 异或后的结果进行 KASUMI 计算, 重复这个过程直至产生足够的密钥流输出. 密钥流产生后, 明文与密钥流做异或操作, 64 bit 的 $data_out$ 就是输出的密文分组. 同理, 解密操作时, 64 bit 的 $data_out$ 就是输出的明文分组.

f9 模式的控制模块与 f8 模式类似, f9 模式同样包含两部分. 首先, 处理好的长度为 64 bit 的比特单元 $data_f9$ 通过多路选择器 mux4, 上轮 KASUMI 算法的结果与 en_1 做按位与操作后与 mux4 的输出进行异或操作, 异或值作为输入进行 KASUMI 计算, 结果与寄存器 A 中的值做异或后重新存入寄存器 A , 重复这一过程直至处理全部的比特单元. 最后, 将寄存器 A 中的值代入一个 IK 修正模式下

的 KASUMI 算法, 计算结果取最左边的 32 bits 作为 f9 的消息认证码 MAC-I.

4 ASIC 硬件实现性能比较

完成 RTL 级 Verilog 代码编写后, 对模块的功能进行仿真和验证. 算法通过了文献[6]中标准向量的测试, 并利用 C 语言随机函数产生百万组随机输入的消息, C 程序的结果与 Verilog 代码计算出的结果进行比对, 若不正确则中断进行调试直至程序功能完全正确. RTL 级代码通过了百万组随机测试向量的验证, 证明了电路功能的正确性. 为方便对比, 本文的设计采用与文献[3]中相同的工艺, 在 SMIC 0.13 μm CMOS 工艺下综合, 本设计的面积是 13 979 门, 关键路径的延时为 8.1 ns, 时钟频率可达 123.4 MHz, 吞吐率达到 $64/4 \times 123.4 = 1.97$ Gb/s. 查阅相关文献, KASUMI 算法在 f8 和 f9 模式下芯片实现的单核性能对比如表 1 所示.

表 1 KASUMI 算法芯片实现性能比较

迭代 轮数	面积/ 门	频率/ MHz	吞吐率/ Gb/s	工艺/ μm	
文献[3]	56	6 512	454.55	0.519 49	0.13
文献[3]	32	6 871	333.33	0.666 66	0.13
文献[3]	8	8 270	200	1.6	0.13
文献[7]	54	2 990	97.6	0.110 3	0.11
本文	4	13 979	123.4	1.97	0.13

本文复用了 KASUMI 算法的两轮轮函数, 因此本设计占用的面积有所增加. 但是从表 1 中可以看出, 本设计的吞吐率高于其他同类设计, 比文献[3]在吞吐率上提高了 23%. 同时本设计降低了时钟频率, 功耗低, 而且在功能上更加全面, 支持随时中断暂停功能, 更方便应用.

5 结束语

本文实现了 KASUMI 密码算法的芯片设计, 该设计同时支持 f8 和 f9 两种模式. 在 SMIC 0.13 μm CMOS 工艺下综合, 电路的面积为 13 979 门, 吞吐率达到了 1.97 Gb/s. 与其他实现方法相比, 本文的实现方法吞吐率高, 同时功能全面, 支持随时中断暂停功能[8]. 实际应用中可以根据需要的性能指标, 例化多核, 实现灵活配置. 该电路适用于移动通信系统, 应用价值高.

(下转第 81 页)

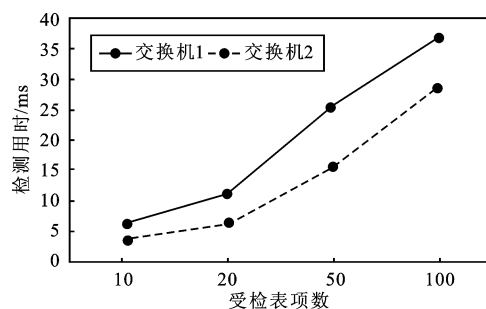


图4 不同交换机数据的检测用时

5 结束语

本文基于协议无感知转发的工作原理,探讨了POF环境下流表冲突的问题,设计出一种冲突检测方案,并在POF控制器上实现。通过实验证明检测的效率与受检规则包含的表项数及流表的规模呈负相关。

参考文献:

- [1] 左青云, 陈鸣, 赵广松, 等. 基于 Open Flow 的 SDN 技术研究[J]. 软件学报, 2013, 24(5): 1078-1097.
- [2] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
- [3] Gude N, Koponen T, Pettit J, et al. NOX: towards an operating system for networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38

(3): 105-110.

- [4] Big Switch Networks. Floodlight open flow controller [EB/OL]. [2014-12-13]. <http://www.projectfloodlight.org/floodlight/>
- [5] Song H. Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane [C] // Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. China, Hongkong, ACM, 2013: 127-132.
- [6] Huawei Technologies. Protocol oblivious forwarding [EB/OL]. [2014-12-10]. <http://www.poforwarding.org/>
- [7] Ghodsi A, Shenker S, Koponen T, et al. Information-centric networking: seeing the forest for the trees [C] // Proceedings of the 10th ACM Workshop on Hot Topics in Networks. Cambridge, ACM, 2011: 1.
- [8] Porras P, Shin S, Yegneswaran V, et al. A security enforcement kernel for open flow networks [C] // Proceedings of the first workshop on Hot topics in software defined networks. Finland, Helsinki, ACM, 2012: 121-126.

作者简介:

吕威男, (1990-), 硕士研究生. 研究方向为基于协议无感知的 SDN 技术.

王雷(通讯作者)男, (1972-), 博士, 副教授. 研究方向为未来网络、云计算以及分布式系统优化.

E-mail: wangl@ustc.edu.cn.

(上接第 77 页)

参考文献:

- [1] 毛光灿, 何大可. 3G 核心加密算法 KASUMI 算法 [J]. 通信技术, 2002, 35(11): 92-94.
- [2] 马炯. 分组密码算法 KASUMI 及其在 WCDMA 中的应用 [J]. 信息安全与通信保密, 2005, 26(2): 123-126.
- [3] Satoh A, Morioka S. Small and high-speed hardware architectures for the 3GPP standard cipher KASUMI [C] // 5th International Conference on Information Security. Sao Paulo, 2002: 48-62.
- [4] Balderas Contreras T, Cumplido Parra R A. An efficient reuse-based approach to implement the 3GPP KASUMI block cipher [C] // 1st International Conference on Electrical and Electronics Engineering. China, Shenzhen, 2004: 113-118.
- [5] 3GPP TS 35.202. Specification of the 3GPP Confidentiality and integrity algorithms; Document 2: KASUMI specifications [S]. USA: 3GPP, 2009.
- [6] 3GPP TS 35.201. Specification of the 3GPP confiden-

tiality and integrity algorithms; Document 1: f8 and f9 specifications [S]. USA: 3GPP, 2009.

- [7] Dai Yamamoto, Kouichi Itoh, Jun Yajima. A very compact hardware implementation of the KASUMI block cipher [C] // Information Security Theory and Practices. Berlin, Heidelberg, Springer, 2010 (6033): 293-307.
- [8] Marinis K, Moshopoulos N K, Karoubalis F, et al. On the hardware implementation of the 3GPP confidentiality and integrity algorithms [C] // 4th International Conference on Information Security. Milaga, Spain, 2001: 248-265.

作者简介:

许静雯女, (1990-), 硕士. 研究方向为信息安全算法的数字大规模集成电路设计与实现.

E-mail: xujingwen_career@163.com.

李树国男, (1963-), 教授, 博士生导师. 研究方向为信息安全算法的数字大规模集成电路设计与实现.