第 37 卷 第 17 期
Vol.37 No.17

计 算 机 工 程
Computer Engineering

2011 年 9 月
September 2011

• 安全技术 •

文章编号：1000—3428(2011)17—0078—03 文献标识码：A 中图分类号：TP301.6

Hummingbird 算法在射频识别标签中的应用

肖梦琴，沈 翔，杨玉庆，王俊宇

(复旦大学专用集成电路与系统国家重点实验室，上海 201203)

摘 要： Hummingbird 加密算法因实现所需面积较小、功耗较低而适用于低成本射频识别标签的安全加密。为此采用 SMIC 0.13 μm 工艺对 Hummingbird 算法进行硬件实现，通过降低时钟翻转频率和数据翻转频率减少动态功耗。仿真实验结果证明，Hummingbird 算法可以满足电子标签对硬件开销、功耗及响应时间的实际应用要求。

关键词： 射频识别；安全标签；Hummingbird 算法；时钟翻转；数据翻转

Application of Hummingbird Algorithm in Radio Frequency Identification Tag

XIAO Meng-qin, SHEN Xiang, YANG Yu-qing, WANG Jun-yu

(State Key Laboratory of ASIC & System, Fudan University, Shanghai 201203, China)

【Abstract】 Hummingbird algorithm needs small area for implementation and low power, so it is suitable for safety encryption of Radio Frequency Identification(RFID) tags. This paper uses SMIC 0.13 μm to implement Hummingbird algorithm. By reducing the unnecessary clock toggling and data toggling, it reduces dynamic power. Simulation experimental results prove that Hummingbird algorithm can meet the requirements of hardware cost, power consumption and response time.

【Key words】 Radio Frequency Identification(RFID); security tag; Hummingbird algorithm; clock toggling; data toggling

DOI: 10.3969/j.issn.1000-3428.2011.17.025

1 概述

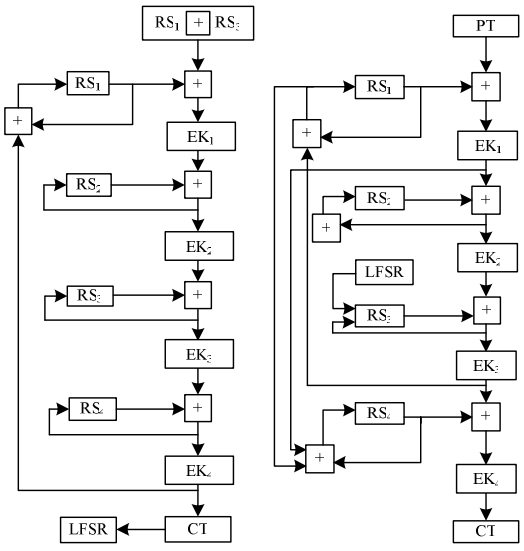
射频识别(Radio Frequency Identification, RFID)是一项利用射频信号通过空间耦合(交变磁场或电磁场)实现无接触信息传递并通过所传递的信息达到识别目的的技术。RFID 系统由 3 个部分组成：标签，读写器以及存储信息记录的后端网络和数据库[1]。正如任何通信设备一样，RFID 系统在应用中也显示出一些安全隐患，RFID 标签易受到非法访问、跟踪、窃听、伪造等攻击，因此标签的安全性成为了 RFID 标签的一个重要部分。除了在标准协议中通过规定的指令来保证标签与读写器的通信安全之外，数据加密算法也被整合到标签中，为标签与读写器的通信提供更可靠的安全保证。

RFID 标签对面积及功耗等的要求较高，文献[2-4]对此提出了一些轻量级对称密钥算法的硬件实现方法。本文则采用 SMIC 0.13 μm 工艺对一种新型的加密算法——Hummingbird 算法进行了硬件实现。

2 Hummingbird 算法

Hummingbird 算法是由 Revere Security 研究团队开发的一种轻量级的加密算法，它所需的实现面积小、功耗低，更能满足低成本无源 RFID 安全标签在响应时间和功耗方面的实际应用要求[5-6]。Hummingbird 算法的密钥长度为 256 bit，明文长度 16 bit。此外，算法结构还包括 4 个 16 bit 内部状态寄存器 RS_i(i=1, 2, 3, 4)和一个 16 bit 线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)。

Hummingbird 算法主要由内部状态寄存器初始化和块加密 2 个过程构成，如图 1 所示，其中，⊞代表 2^{16} 模加；PT 为待加密的明文；CT 为加密后的密文；EK 为 Hummingbird 算法的核心模块。



(a)初始化过程 (b)加密过程

图 1 Hummingbird 算法初始化及块加密过程

具体步骤如下：(1)内部状态寄存器初始化阶段。RS 首先载入随机数 nonce，进行 4 轮如图 1(a)所示的运算，数据输入为 RS_1 和 RS_3 的 2^{16} 模加。每轮运算结束，RS_i 都会被更新

基金项目： 国家“十一五”科技支撑计划基金资助项目“药品安全追溯管理射频识别技术研究”(2008BAI55B07)

作者简介： 肖梦琴(1988－)，女，硕士研究生，主研方向：加密算法，数字集成电路设计；沈 翔，硕士研究生；杨玉庆，博士研究生；王俊宇，副研究员

收稿日期： 2011-01-31 **E-mail：** junyuwang@fudan.edu.cn

一次，第 4 轮运算结束后，RS 中的值就是初始化后的值，运算得到的结果 CT 用来初始化 LFSR。(2)块加密阶段。这一阶段的运算与前一阶段相似，只需进行一轮如图 1(b)所示的运算，且此时的数据输入是待加密的 16 bit 明文 PT，RS_i 是前一阶段初始化的值。加密结束后，RS 再次被更新，用于下一次明文加密。

Hummingbird 算法核心模块 EK 的结构如图 2 所示。EK 是一个典型的代换-置换加密结构，加密块大小为 16 bit，密钥为 64 bit。其加密过程由 4 轮常规变换和一轮最终变换构成。一轮常规变换由密钥混合、代换和置换 3 个操作构成，最终变换只包括密钥混合和代换。代换过程由 4 个 4 bit 输入的 S 盒完成，置换过程为如下线性变换：

$$D=(D=6)\oplus(D=10) \tag{1}$$

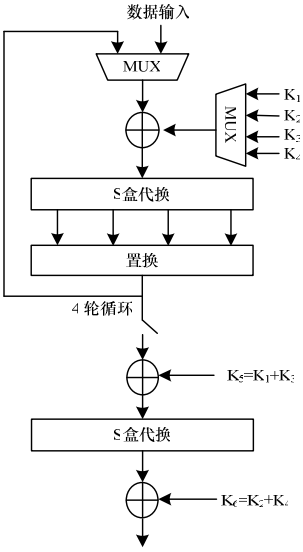


图 2　EK 结构示意图

3　Hummingbird 算法实现

3.1　系统架构

算法的实现需要考虑面积与速度的折中。文献[7]实现了一种速度优化的结构，即将 EK 中的 4 轮常规变换展开。尽管提高了运算速度，但面积的开销很大。本文采用面积优化的结构，4 轮常规运算由同一个模块通过分时复用完成。完成一次 EK 模块的运算需要 4 个时钟周期。因此，完成一次块加密需要 16 个时钟周期。本文设计的结构如图 3 所示。

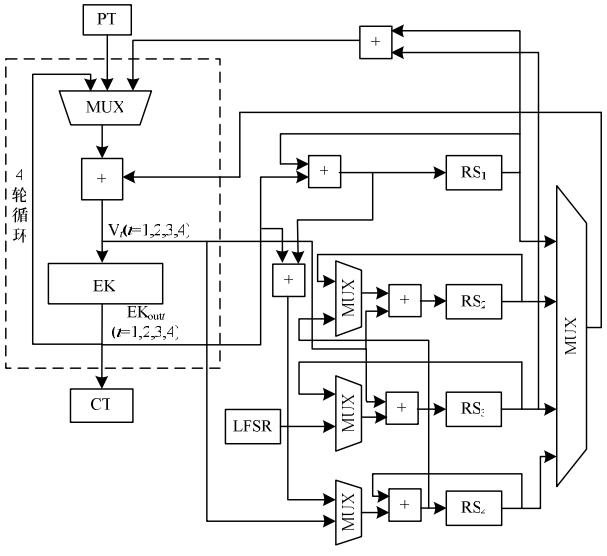


图 3　Hummingbird 结构示意图

RS 在初始化和加密 2 种模式下的更新方法不同，如表 1 所示，其中，加法均为 2^{16} 模加。

表 1　内部状态寄存器的更新方法

初始化运算	加密运算
LFSR=CT|0x1000	$LFSR_{t+1}$<-$LFSR_t$
$RS_{1(t+1)}=EK_{out4}+RS_{1t}$	$RS_{1t+1}=EK_{out3}+RS_{1t}$
$RS_{3(t+1)}=V_3+RS_{3t}$	$RS_{3t+1}=V_3+LFSR_{(t+1)}$
$RS_{4(t+1)}=V_4+RS_{4t}$	$RS_{4(t+1)}=EK_{out1}+RS_{4t}+RS_{1(t+1)}$
$RS_{2(t+1)}=V_2+RS_{2t}$	$RS_{2(t+1)}=V_2+RS_{4(t+1)}$

3.2　面积优化

由于 Hummingbird 算法使用的 S 盒处理的数据为 4 bit，因此面积较小，可以直接采用查表的方式实现。此外，由于算法中的内部状态寄存器在一次加密完成后需要更新，因此要使用较多的寄存器来保持加密过程中产生的中间值用于更新，如表 1 所示，这样必然会消耗大量寄存器，增加面积开销。通过分析电路的时序可知，在 4 轮运算中，每个内部状态寄存器只在一轮运算中被用到，所以，不需要等到一次加密的 4 轮运算全部结束后再更新，当 RS 本轮的任务完成时，等到产生所需的更新中间值就立即对 RS 进行更新。

3.3　功耗优化

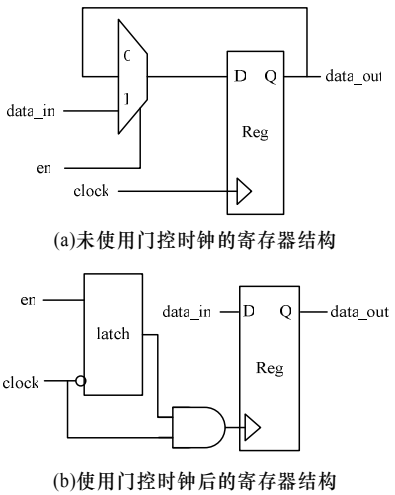
电路的功耗主要分为静态功耗和动态功耗。静态功耗主要来自于亚阈值漏电，减小这部分功耗主要采用多阈值电压的 MOS 管。电路的动态功耗可表示为：

$$P_{\mathrm{dyn}}=\alpha C_{\mathrm{L}}V_{\mathrm{DD}}{}^2 f_{\mathrm{CLK}} \tag{2}$$

其中，C_{L} 为器件的负载电容；V_{DD} 为电源电压；α 为器件的翻转概率。

通过对电路结构的设计，翻转概率最容易减少。本文对电路功耗的优化主要是通过降低电路中器件的翻转概率来实现的。

电路中翻转频率最大的是时钟信号，并且时钟的扇出负载很大，时钟网络的功耗占据了整个电路很大一部分。本文采用门控时钟来减少不必要的时钟翻转。图 4 给出使用门控时钟前后的寄存器结构。未采用门控时钟时，若使能信号无效，输出保持不变，但此时时钟仍然翻转；采用门控时钟后，使能信号控制时钟，仅当使能信号有效时，控制寄存器的时钟才产生翻转。



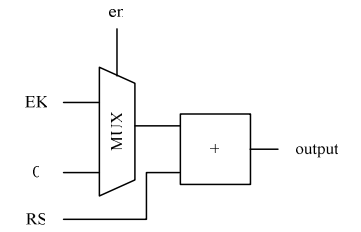
(a)未使用门控时钟的寄存器结构

(b)使用门控时钟后的寄存器结构

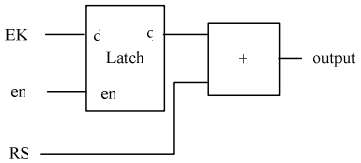
图 4　使用门控时钟前后的寄存器结构

除了采用门控时钟降低时钟翻转概率外，通过分析电路还可以减少不必要的数据翻转。EK 的输出在每个周期都会改变，导致加法器在每个周期都会翻转，如图 1 所示。EK 模块

的运算需要 4 个时钟周期，而实际上加法器的输出只在第一个时钟周期内被用到。有 2 种方法可以减少加法器在其余的 3 个时钟周期内不必要的翻转。如图 5(a)所示，在第 1 个时钟周期内，EK 的输出 EK_{out} 被选中作为加法器输入，而在其余的 3 个时钟周期内，数据“0”被选中作为加法器输入。在 4 个时钟周期内，加法器的输出只翻转 2 次。如图 5(b)所示，通过引入锁存器来保存第 1 个时钟周期输出 EK_{out} 的值，使 4 个时钟周期内加法器的输入值都保持不变。这样，加法器的输出在 4 个时钟周期内只翻转了一次，与图 5(a)的方法相比，翻转概率更小。但由于该方法的实现需要引入 16 个锁存器，会增大面积开销和漏电功耗，因此本文采用了图 5(a)的改进方法。



(a)基于组合逻辑的方法



(b)基于锁存器的方法

图 5 2 种降低数据翻转频率的方法

4 结果分析

4.1 仿真结果

对所做设计在 Synopsys Design Complier 中进行了综合，所用工艺库为 SMIC 0.13 μm 工艺。为了与同类研究进行比较，时钟频率选取 100 kHz。对设计的面积，功耗及运算周期进行了仿真，结果表明，电路的面积为 14 735 μm^2，取库中的 2 个输入与非门(面积 6.8 μm^2)为一个单元计算等效门数，可以得到电路大约为 2 167 门。

Hummingbird 算法在加密之前需要进行初始化，初始化需要 69 个时钟周期。之后进入数据加密阶段，加密 16 bit 数据需要 16 个时钟周期。

功耗分析采用的是 Synopsys 的 Nanosim。它的结果较为精确，可以达到类似于 HSPICE 的精度。表 2 比较了采用功耗优化方法前后的最大及平均电流大小。

表 2 优化前后最大及平均电流比较

V_{DD}/V	优化前		优化后	
	i_{avg}/μA	i_{max}/μA	i_{avg}/μA	i_{max}/μA
1.2	1.270	10 000	0.900	9 160
1.0	1.140	7 190	0.617	5 500
0.8	0.655	4 370	0.227	2 620

在时钟频率为 100 kHz、电压为 0.8 V~1.2V 的条件下，模块的功耗为 0.182 μW~1.08 μW。结果表明，Hummingbird 算法的功耗较低，适合 RFID 安全标签的应用。

4.2 与其他加密算法的比较

已有不少关于 RFID 标签安全加密算法的硬件实现，如 AES 算法、DESL 算法。下面主要从面积、时钟周期和功耗 3 个方面对本文的加密算法和 AES[2]、DESL[3]、PRESENT[4] 进行比较。实验结果如表 3 所示。

表 3 不同加密算法的硬件性能比较

加密算法	采用工艺/μm	等效门数	明文长度/bit	加密周期/时钟	时钟频率/kHz	功耗/μW
AES[2]	0.13	3 200	128	160	100	3.00
DESL[3]	0.18	1 848	64	144	100	1.60
PRESENT[4]	0.18	1 075	64	563	100	2.52
本文算法	0.13	2 167	16	16	100	1.08

从表中可以看出，虽然文献[4]给出的面积只有 1 000 多门，但是运算时间太长。本文设计实现的 Hummingbird 算法加密 16 bit 数据需要 85 个周期，其中包括初始化的 69 个时钟周期。由于初始化在一次数据通信过程只需进行一次，因此加密的数据长度越长，Hummingbird 算法在运算时间上的优势越明显。例如，本文的设计在 581 个时钟周期内能完成 512 bit 的数据加密，而 AES[2]需要 640 个时钟周期，DESL[3]需要 1 152 个时钟周期，PRESENT[4]则需要 4 204 个时钟周期。

5 结束语

本文对一种新型的加密算法 Hummingbird 进行了低功耗的硬件实现。通过分析算法加密流程，优化了电路所需面积。在设计中采用门控时钟和组合逻辑电路，减少了不必要的数据翻转，降低了电路的功耗。从等效门数、时钟周期和功耗 3 个方面与其他算法进行实验对比，结果证明，Hummingbird 算法适用于低成本低功耗的电子标签安全加密。

参考文献

[1] 范文兵, 葛 峥, 王 耀. 超高频 RFID 系统设计与仿真[J]. 计算机工程, 2010, 36(17): 90-92.

[2] Hamalainen P, Alho T, Hannikainen M. Design and Implementation of Low-area and Low-power AES Encryption Hardware Core[C]//Proc. of EUROMICRO’06. [S. l.]: IEEE Press, 2006.

[3] Poschmann A, Leander G, Schramm K. New Light-weight Cryptographic Algorithms for RFID[C]//Proc. of International Symposium on Circuits and Systems. New Orleans, USA: IEEE Press, 2007.

[4] Rolfes C, Poschmann A, Leander G, et al. Ultra-lightweight Implementations for Smart Devices Security for 1 000 Gate Equivalents[C]//Proc. of Smart Card Research and Advanced Application Conference. [S. l.]: Springer, 2008.

[5] Engels D, Fan Xinxin, Gong Guang, et al. Ultra-lightweight Cryptography for Low-cost RFID Tags: Hummingbird Algorithm and Protocol. Center for Applied Cryptographic Research[EB/OL]. [2010-09-29]. http://www.cacr.math.uwaterloo.ca/techreports/2009/cacr2009-29.pdf.

[6] Recere Security. Revere Security Introduction[EB/OL]. (2010-08-13). http://www.reveresecurity.com/resources.html.

[7] Fan Xinxin, Gong Guang, Lauffenburger K, et al. FPGA Implementations of the Hummingbird Cryptographic Algorithm[C]//Proc. of IEEE International Symposium on Hardware-oriented Security and Trust. Anaheim, USA: IEEE Press, 2010.

编辑 张 帆