

一种基于 memory 的数据加解密算法

张金旻, 谢小东

(电子科技大学, 四川 成都 610054)

摘 要: 对于 memory 的研究, 大部分都只是停留在对容量的扩充, 速度的提高等性能提升方面, 很少涉及对 memory 进行辅助设计以实现更多的特殊功能. 提出了一种通过对 memory 地址作用加解密算法, 利用 memory 自身的特性来实现对数据的加解密的功能, 并通过实验仿真验证了该方法的可行性.

关键词: Memory; 加解密; 仿真

中图分类号: TN918.91

文献标识码: A

文章编号: 1000-7180(2015)04-0061-04

An Encryption and Decryption Algorithm of Data Based on Memory

ZHANG Jin-min, XIE Xiao-dong

(University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: Most memory researches only focus on improving the performance, which is to enlarge the capacity and to increase the speed. However, aided designs to realize more functions have been rarely achieved in present memory studies. So an algorithm to encrypt and decrypt the address of memory is given out in this paper, and the encryption and decryption of the data is realized, which is based on the characteristics of memory, and the feasibility of this algorithm with simulation is validated.

Key words: memory; encryption and decryption; simulation

1 引言

在当代, 不管是在网络通信方面, 还是在信息存储方面, 数据的安全性越来越受重视^[1]. 人们通过一定的加密算法, 对所传输的数据进行加密, 或者将数据加密后, 存储在 memory 中, 这样即使 memory 里面的数据被盗取, 盗取方仍然不能获取正确的数据. 关于 memory 的数据加密, 有很多方法^[2-4], 但是大部分加密算法比较复杂, 需要消耗大量的软硬件资源来实现.

本文提出了一种新的数据加解密方法, 在对数据利用 memory 进行存取的同时, 通过对地址作用一定的加密算法, 从而实现对数据安全的保护.

2 利用 memory 实现对存储数据加密

通常对 memory 的读取操作中, 地址线和数据

线的逻辑顺序均是按照正常的顺序进行操作, 但是如果改变这种常规的方式, 顺序写、乱序读, 乱序写、顺序读, 或者乱序写, 乱序读. 那么就可以实现对存储数据的一种加密算法.

以一个 2M 的 16 bit 的 SRAM 为例, 16 bit 的 SRAM, 一共有 17 根地址线 $A_{16}-A_0$. 当只对地址进行乱序处理的情况下, 有下列三种情况:

(1) 顺序写, 乱序读

memory 的顺序地址为 $A_{16}-A_0$. 按照该顺序将数据写入到 memory 中, 而读出的时候, 则可以打乱地址线的顺序, 实现地址乱序的读出. 如表 1 所示.

读出数据之前, 首先对地址线进行乱序地址处理, 处理方式可以基于某种算法^[5-7], 也可以简单进行移位, 拼凑等处理, 这里为了方便起见, 只是做一些简单的处理, 处理方式如图 1 所示.

表 1 地址与数据对应表

$A_{16}A_{15}-A_{12}A_{11}-A_8A_7-A_4A_3-A_0$	数据
0_0000_0000_0000_0000(0x00000)	0x0420
0_0000_0000_0000_0001(0x00001)	0xA0B2
0_0000_0000_0000_0010(0x00002)	0x0FC6
0_0000_0000_0000_0011(0x00003)	0xCB23
0_0000_0000_0000_0100(0x00004)	0x841C
0_0000_0000_0000_0101(0x00005)	0x2700
0_0000_0000_0000_0110(0x00006)	0x1095
0_0000_0000_0000_0111(0x00007)	0x1F6D

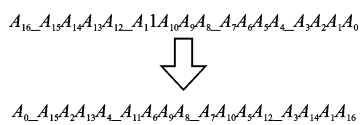


图 1 地址转换图

以 A_9 为对称中心,交替交换两边的地址.按照上述处理之后,各个数据所对应的地址线如表 2 所示:

表 2 转换后地址与数据对应表

$A_0A_{15}A_2A_{13}A_4A_{11}A_6A_9A_8A_7A_{10}A_5A_{12}A_3A_{14}A_1A_{16}$	数据
0_0000_0000_0000_0000(0x00000)	0x0420
1_0000_0000_0000_0000(0x10000)	0xA0B2
0_0000_0000_0000_0010(0x00002)	0x0FC6
1_0000_0000_0000_0010(0x10002)	0xCB23
0_0100_0000_0000_0000(0x04000)	0x841C
1_0100_0000_0000_0000(0x14000)	0x2700
0_0100_0000_0000_0010(0x04002)	0x1095
1_0100_0000_0000_0010(0x14002)	0x1F6D

(2)乱序写,顺序读

乱序写和顺序读,从原理上讲,与顺序写和乱序读几乎是一样的,区别在于地址转换算法一个是由发送方确定,一个是由接收方确定.

(3)乱序写,乱序读

读写两方均作乱序的处理,原理同上.

在只改变地址线顺序的情况下,破解一组的计算量为 131 072 次运算,对于要求比较简单的情

况下,可以用这种方法来实现加密的需求.但是,memory 的另一个自身的特性还没有被利用到,就是数据位的顺序,假设数据位是 8 位,那么计算量将是 $2^{17} \times 2^8 = 33\ 554\ 432$,实际上,现在大部分的存储器的数据位宽都是 16 bit,甚至扩展到了 32 bit 和 64 bit.对本次所使用的例子而言,SRAM 的数据位宽为 16 bit,因此,计算量将会达到 $2^{17} \times 2^{16} = 8\ 589\ 934\ 592$,较之之前的计算量扩大了 256 倍.安全系数相应也随之大大提高,下面介绍一种地址和数据在顺序均加密的情况下的算法.

假设 memory 的顺序地址为 $A_{16}-A_0$,数据位为 $D_{15}-D_0$.F 是地址转换函数,如下:

$$(A_{16}^*, \dots, A_0^*) = F(A_{16}, \dots, A_0) \quad (1)$$

如果想增加算法的复杂度,可以使用转换后的地址组来进行对数据位的转换,如下:

$$(D_{15}^*, \dots, D_0^*) = M(K(A_{16}^*, \dots, A_0^*), D_{15}, \dots, D_0) \quad (2)$$

函数 K 和函数 M 的不同,所加密的方式就不同,因此,完全可以由设计人员根据复杂程度来选择所需求的加密方式.下面就提供一个简单可行的加密方式进行说明.

首先针对数据存储方而言,是需要将数据存入 memory,因此,给定的初始数据和地址是一一顺序对应的,如表 1 所示,为了方便起见,函数 F 就用图 1 所给的转换方式,

其算法如下:

$$A_n^* = \begin{cases} A_{16-n}, & n \text{ 为偶数} \\ A_n, & n \text{ 为奇数} \end{cases} \quad (3)$$

地址转换完成后,便开始对数据进行乱序加密,函数 K 和函数 M 分别如下:

为了方便起见,简化函数 K.

函数 K:

$$K(A_{16}^*, \dots, A_0^*) = (A_{16}^*, \dots, A_0^*) \quad (4)$$

函数 M:

$$M = K + (\overline{D_{15}}, \dots, \overline{D_0}) + C \quad (5)$$

在函数 M 中,后面的 C 是设计人员自定义补位常量,C 的设计是为了防止当由于某种偶然的情况下,经过 M 函数转变之后,连着几组数据全部为 0 的情况,当然这种情况的出现与算法的设定有关,通常情况下,可以将 C 忽略不计,当发现出现上述情况后,再来补位添加.

上面的转换步骤是针对于数据存储方而言,经过上述的转换后,图所示的地址与对应的数据变为如表 3 所示:

表3 地址与数据顺序位均加密结果对应表

$A_0_A_{15}A_2A_{13}A_4_A_{11}A_6A_9A_8_$ $A_7A_{10}A_5A_{12}_A_3A_{14}A_1A_{16}$	数据
0_0000_0000_0000_0000(0x00000)	0xFBDF
1_0000_0000_0000_0000(0x10000)	0x5F4D
0_0000_0000_0000_0010(0x00002)	0xF03B
1_0000_0000_0000_0010(0x10002)	0x34DE
0_0100_0000_0000_0000(0x04000)	0xBBE3
1_0100_0000_0000_0000(0x14000)	0x18FF
0_0100_0000_0000_0010(0x04002)	0x2F6C
1_0100_0000_0000_0010(0x14002)	0x2094

而针对数据读取方而言,则必须要按照逆过程将数据读取出来,地址解密方式相同:

$$A_n^* = \begin{cases} A_{16-n}, & n \text{ 为偶数} \\ A_n, & n \text{ 为奇数} \end{cases} \quad (6)$$

数据解密:

$$(D_{15}, \dots, D_0) = \overline{M - K - C} \quad (7)$$

需要注意的是,本算法的数据加密是基于加法和取反两种逻辑得以实现的,必然会涉及到数据相加之后,产生算术溢出的情况,如表1中,第六组数据为 0x2700,在写入的时候按照上述的加密方法,

进行加密,计算过程如下:

$$\begin{aligned} & 0x2700 + 0x4000 \\ &= 0xD8FF + 0x4000 \\ &= 0x118FF \end{aligned} \quad (8)$$

进行加密之后,数据产生了算术溢出,因此,为了保证在数据解密的时候,解密的数据不会出错,整个数据加密和解密过程全部按照无符号位的二进制加法进行运算,在运算过程中,若产生算术溢出,则舍弃高位进位,继续参与运算.因此,在解密的过程中,对于上述情况,0x18FF-0x4000 的减法运算,在无符号位二进制运算下,运算结果为 0xD8FF,然后取反,便可以得到正确的数据 0x2700.

这种加密方式,相对与普通的加密方式而言,不需要太多复杂的加密算法,利用了 memory 自身的特性,简易的实现了数据加密的方式.

3 仿真结果

根据上述算法,对数据和地址均乱序的情况下进行数据仿真,仿真结果如图2和图3所示(仿真工具为 Modelsim 10.0c).

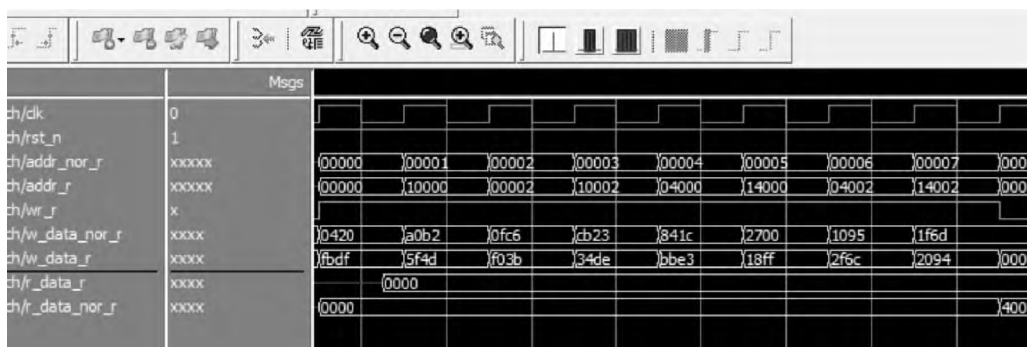


图2 写数据加密仿真波形图

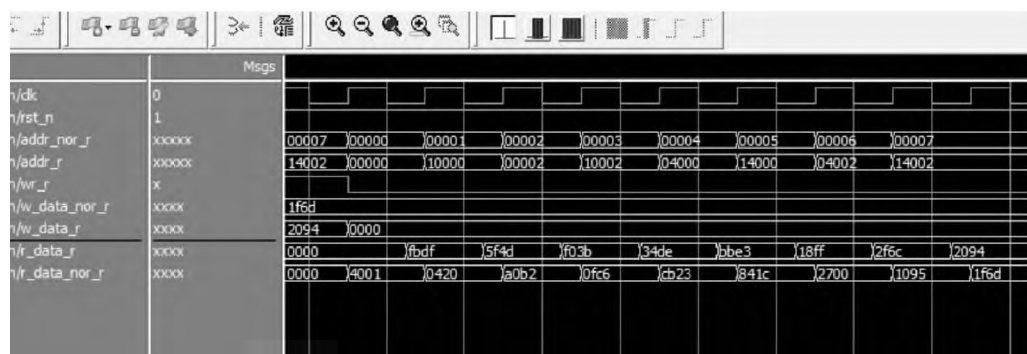


图3 读数据解密仿真波形图

图中,addr_nor_r,w_data_nor_r,r_data_nor_r 为读写 memory 的地址和数据信号,操作方均是按

照正常的顺序对 memory 进行读写控制,然后经过算法加密乱序处理之后,addr_r,w_data_r,r_data_r 为实际读写 memory 的地址和数据信号.

wr_r 为读写控制信号,当 wr_r 为 1 时,表示当前为写操作,否则为读操作.

仿真过程中,首先写入数据到 memory 中,然后依次将数据读出,从仿真图中可以看出,其数据结果与理论分析的数据完全相同,从而证明了此加密方法的可行性.

此外,从图 3 中看书,在给出读信号的同时,r_data_nor_r 有一个无效的返回值,这是因为此时地址还没被更新为有效地址,因此读出一个无效的值,但是由于读周期需要一个时钟周期,所以在给出读数据地址一个周期后才能得到正确的值,正如波形图中所示,给出地址 0x00000 一个周期后,得到正确的数值 0x0420. 所以该无效值不影响数据的正确读写.

4 结束语

本文利用 memory 自身的特性,通过对地址作用一定的加密算法,实现了对数据的加密,提高了数据存储的安全性.

参考文献:

- [1] 李聪. 数据加密技术在网络安全中的重要性[J]. 企业导报, 2011, 14(10): 296.

- [2] Gao Lan, Yang Jun, Chrobak Marek, et al. A low-cost memory remapping scheme for address bus protection[C] // Proceedings of the 15th international conference on Parallel architectures and compilation techniques. USA:Seattle, Washington, 2006: 74-83.
- [3] Xu Huachun, Li Zhengsheng, Wang Zenghong. Study on hardware protection technique of flash disk[C] // The 8th International Conference on Electronic Measurement and Instruments. Xi'an, 2007: 55-59.
- [4] 吴延军. 基于 FLASH 芯片的加密存储技术研究[D]. 广州: 暨南大学, 2006.
- [5] 苏长龄, 庄晋林. 基于序列变换的加密方法[J]. 微电子学与计算机, 1998, 15 (5): 46-47.
- [6] Feuser Markus, Sommer Sabine. Address encryption method for flash memories; USA, US7640437B2[P]. 2009-12-29.
- [7] Mei Hong, Hui Guo, Parameswaran S. Dynamic encryption key design and management for memory data encryption in embedded systems[C] // IEEE Computer Society Annual Symposium on VLSI (ISVLSI). Brazil, Natal, 2013: 70-75.

作者简介:

张金旻 男,(1990-),硕士研究生. 研究方向为专用集成电路与系统设计. E-mail:gzyd2008@yeah.net.

谢小东 男,(1977-),博士,副教授,硕士生导师. 研究方向为超大规模集成电路设计、专用集成电路设计、微系统与 SOC 设计等.

(上接第 60 页)

- [9] Li Bo, Zheng Chunhou, Huang Deshuang. Locally linear discriminant embedding: an efficient method for face recognition [J]. Pattern Recognition, 2008, 41 (12): 3813-3821.
- [10] Sam T Roweis, Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290(5500): 2323-2326.
- [11] Yamada K, Komine H, Kinukawa H, et al. Abstract of abstract: a new summarizing method based on document frequency and clause length [C] // 8th World Multi-Conference on Systemics, Cybernetics and Informatics. Finland Orlando, 2004: 56-61.
- [12] Chen Chih Ming, Lee Hahn Ming, Chang Yu jung. Two novel feature selection approaches for web page

classification [J]. Expert systems with Applications, 2009, 36 (4): 260-272.

- [13] Ferjani Fethi, Elloumi Samir, Jaoua Ali, et al. Formal context coverage based on isolated labels: an efficient solution for text feature extraction [J]. Information Sciences, 2012, 188(4): 198-214.

作者简介:

李卫平 男,(1974-),博士研究生,副教授. 研究方向为计算机软件与理论、数据挖掘. E-mail:liwp360@163.com.

杨杰 女,(1960-),博士生导师,教授. 研究方向为信息传输理论及应用、模式识别.

王钢 男,(1958-),博士,教授. 研究方向为云计算、计算机犯罪侦查.