基于 FPGA 的 IDE 硬盘数据 AES 加解密研究与实现

陈灯川,马维华,张有成
(南京航空航天大学 信息科学与技术学院,江苏 南京 210016)

摘 要:提出了基于 FPGA 对 IDE 硬盘数据进行 AES 加解密的方法。对算法进行了改进和优化,以降低加解密过程对 IDE 硬盘数据传输速度的影响。

关键词:FPGA AES IDE

随着信息量的急速增加,对硬盘数据的保护日益受到重视。如何使加密过程不影响硬盘数据传输性能成为应用研究的课题。同时,随着密码分析技术的不断发展,超期服役的 DES 算法已被攻破,于是美国商业部提出以 Rijndael 算法的 AES 作为新一代的加密算法。AES 算法的加密强度相比 DES 算法大大提高。采用 AES 算法对硬盘上的数据进行加密与解密,能够使企图非法获得硬盘数据的门槛提高。若采用 256 位 AES 算法加密,将使非法窃取硬盘数据几乎不可能。采用 ALTERA 的 FPGA 芯片 Cyclone 2 EP2C8 实现 AES 加密,其吞吐率足以满足 IDE 总线的传输速率要求。

1 硬件电路设计

硬件电路分成三大模块:电源模块、IDE 接口过滤处理模块和获取密钥模块。电源模块负责产生 EP2C8 芯片所需要 3.3V 和 1.2V 的电压。IDE 接口过滤处理模块负责对 IDE 接口信号线上的指令和数据过滤,对数据进行加解密:当从硬盘读取数据时,为解密操作;当向硬盘写数据时,为加密操作。因此必须对 IDE 总线传输采用的 PIO 模式和 Ultra DMA 模式进行识别和操作。限于篇幅,本文简要介绍采用 PIO 模式的传输方式。获取密钥模块通过 89C51 单片机模拟 I²C 总线,从智能卡上读取 ID,根据一定的算法,生成密钥再传给 EP2C8。硬件电路框架如图 1 所示。

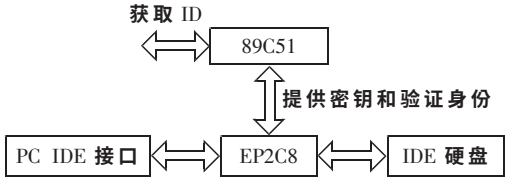

图 1 硬件电路框架图

2 IDE 接口过滤实现

在采用 PIO 模式的传输方式下,IDE 接口通过数据寄存器读写数据,可进行 8 位及 16 位数据的读写。对于 AES 加解密,需要成块地对数据进行操作。现设定对 128 位的数据进行加解密操作,因此应在 EP2C8 中设计双口 RAM。当每次读一个字节时,将地址位低 7 位为 0 的数据连续读取 128 位(16 字节),并在其中对数据成块地进行解密操作。对 IDE 接口信号过滤的所得数据块由 AES 模块进行处理,处理完后将数据按照 IDE 接口时序转发出去。对 IDE 接口信号,首先截获其命令字和控制信号,获知其中的信息,分析何时传输的是数据,何时传输的是参数,采用 PIO 传输的是何种模式等。每种模式的时序都有所差别,这对最后转发给硬盘的时序很重要。

PIO 模式传输相对于 Ultra DMA 模式传输更为简单,速度要求不是很高,实现起来也简单很多。这里不对 PIO 模式传输的过滤详述。下面详细介绍如何对基于 PIO 模式传输的数据进行 AES 加解密。

3 AES 算法的简述

2000 年 10 月,NIST(美国国家标准和技术协会)通过考核,从 15 种侯选算法中选出 Rijndael 算法的 AES 作为新的加密算法标准,代替 DES 算法。Rijndael 算法采用 128 位的分组长度,支持长度为 128 位、192 位和 256 位的密钥。到目前为止,AES 算法还没有出现任何致命缺陷。

下面以 PIO 模式传输 16 字节,即 128 位的数据和 128 位密钥为例,说明 AES 加/解密过程,如图 2 所示。

AES 加密和解密算法的输入都是一个 128 位的分组。这个分组被编排为一个称做状态阵列的 4×4 字节矩阵,这个状态阵列在每一层加密或解密时都要进行修改。在加密或解密的最后阶段过后,这个状态阵列又被重新转换为 128 位的线性串。128 位密钥也类似地被变为字节方矩阵,密钥经过 10 层处理后,扩展为 10 组不同的密钥。一层典型的加密处理都由四个阶段构成:ByteSub(字节代替)、RowShift(行移位)、MixColumns(列混合)和 AddRoundKey(加密钥)。ByteSub 使用一张表(即 S 盒)对分组的字节进行代替,也就是说,每一个输入字节被映射到惟一的一个输出字节。在 RowShift 中,状态阵列的第一行保持不动,第二行执行 1 字节循环左移,第三行执行 2 字节循环左移,第四行执行 3 字节循环左移。在

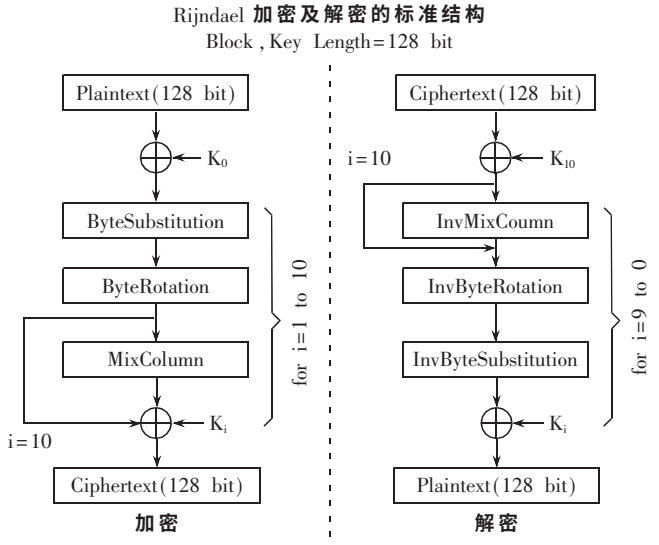

图 2 加解密的过程图解

MixColumn 中，代替算法作为作用于一列中所有字节的函数，改变列中的每个字节。在 AddRoundKey 中，使用扩展密钥中的一块 4×4 字节，其每一个字节与状态阵列的对应字节进行“异或”操作。

不论是加密还是解密，AES 算法总是从 AddRoundKey 阶段开始，然后进行 9 层运算，每一层都包括所有四个阶段，最后进行只包括三个阶段的第 10 层操作，第 10 层操作没有 MixColumns 阶段，且只有 AddRoundKey 阶段使用密钥。因此，AES 算法都是以 AddRoundKey 阶段开始和结束。AES 算法通过四种阶段交替地对数据分组进行处理：首先利用 XOR 函数进行 AddRoundKey 操作，接着在其他三个阶段对分组进行错乱处理，然后再利用 XOR 函数加密，如此反复。AES 的解密算法使用逆序的扩展密钥，但解密算法与加密算法并不完全一样。

4 AES 算法的 FPGA 实现

通过分析发现，S 盒在整个设计中占有很大的比重。S 盒性能的提高对于整个设计的性能会有很大的改善，因此 S 盒是整个设计优化的重点。AES 的 S 盒是一个 8×8 的变换，可采用 ROM 实现，即把输入的 8 位作为地址，对应的地址空间里存放的就是输出的 8 位，从而实现 8×8 的变换，所需时间只是 FPGA 中 LE 的传输时间加上传输线上的延时。为了满足速度要求，加入了流水线方式。加密部分采用 Cyclone 2 EP2C8 芯片实现。

在本设计中，轮密钥的产生和加密轮的计算并行进行，加密模块和解密模块如图 3 和图 4 所示。

密钥和明文的输入模块支持 128 位的密钥和明文。密钥和明文的输入可采用同一个模块。采用每次 32 位的输入，然后移到下一个寄存器，一个完整的 128 位输入需要 4 个时钟周期。此外还需一些简单的控制来确定是否接收了 128 位。

用 MODELSIM 对 AES 加密过程进行仿真，仿真波形如图 5 所示。

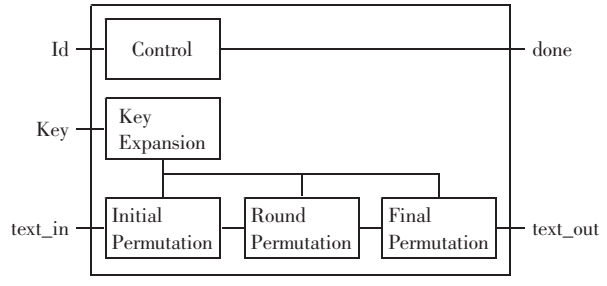

图 3 加密模块

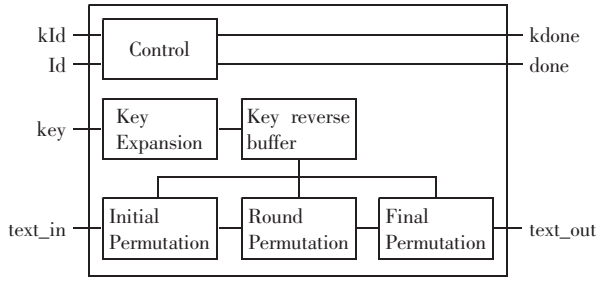

图 4 解密模块

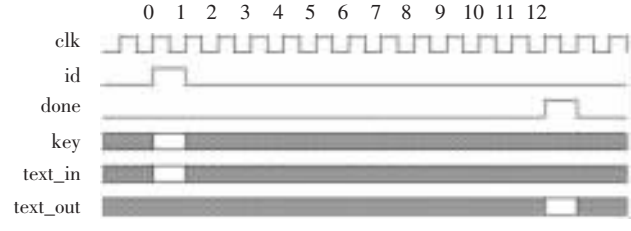

图 5 仿真波形

用 quartus 进行综合和优化的结果如表 1 所示（采用 Altera Cyclone 2 EP2C8-208QC7）。

表 1 用 quartus 进行综合和优化的结果

速度	PIO 模式，数据最高传输速率为 16.7MB/s
面积	4913 LE

单独的 AES 加解密模块的速度可以达到 6Gbps，但由于受硬盘的 PIO 模式传输速度的约束，速度仅为 16.7Mbps。LE 单元耗费较大的是 IDE 接口时序的过滤与模拟。

本文提出了基于 FPGA 对 IDE 硬盘数据进行实时加解密的方法。介绍了 EP2C8 对 IDE 接口信号的过滤，之后详细介绍了 AES 对数据的加解密，并做了进一步的优化，但更进一步的优化有待今后研究。在当前设计中，只实现了 PIO 模式，对于 Ultra DMA 模式的支持有待进一步研究和实现。

参考文献

1 精英科技译.SCSI 总线和 IDE 接口：协议、应用和编程（第二版）.北京：中国电力出版社，2001
2 张 亮.数字电路设计与 Verilog HDL.北京：人民邮电出版社，2000
3 夏宇闻.Verilog 数字系统设计教程.北京：北京航空航天大学出版社，2003

（收稿日期：2006-06-21）