[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06K 19/073 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200480008265.0

[43] 公开日 2006 年 4 月 26 日

[11] 公开号 CN 1764920A

[22] 申请日 2004.1.26
[21] 申请号 200480008265.0
[30] 优先权
[32] 2003. 1.24 [33] GB [31] 0301726.6
[86] 国际申请 PCT/GB2004/000303 2004.1.26
[87] 国际公布 WO2004/066196 英 2004.8.5
[85] 进入国家阶段日期 2005.9.26
[71] 申请人 埃塞博斯有限公司
地址 英国南约克郡
[72] 发明人 巴里·希姆·霍克菲尔德
安东尼·布雷斯林

[74] 专利代理机构 永新专利商标代理有限公司
代理人 王 英

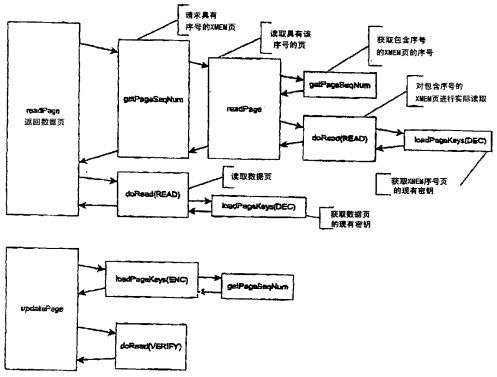
权利要求书 2 页 说明书 5 页 附图 3 页

[54] 发明名称

存储器访问受保护的智能卡

[57] 摘要

一种智能卡包括一个衬底，该衬底上形成一个智能卡芯片和第二存储器器件，如闪存 ROM，该第二存储器器件被可操作地连接到智能卡芯片。 第二存储器器件能够存储多页数据，每个页有一个唯一序号与之相关联，该序号与该数据页分开存储以便当该页被读取时，可以将与该页一起获取的序号与存储的序号相比较以认证该页。 为了最小化智能卡上用于存储序号的存储器，存储的序号中的一些被存储在至少一个存储在第二存储器器件上的数据页中，该至少一个页的序号被存储在智能卡芯片中或者序号被异或并且结果摘要被存储在智能卡 EEPROM 中。 与特定页相关联的序号在每次页被修改或更新时增加。 每个数据页的序号初始被设为一个随机生成的数值，以便不能从页的更新总数中推导出该序号。 因此，本发明的智能卡允许扩展存储器的使用而不损坏安全性。



1、一种包括衬底的智能卡，所述衬底上具有智能卡芯片，所述智能卡的特征在于：它包括所述衬底上的第二存储器器件，并且该器件被可操作地连接到所述智能卡芯片。

2、如权利要求 1 所述的智能卡，其中，所述第二存储器器件是闪存 ROM。

3、如权利要求 1 或 2 所述的智能卡，其中，所述第二存储器器件能够存储多页数据，每页与一个唯一的序号相关联，所述序号与所述数据页分开存储，以便于当所述页被读取时，能够将与所述页一起获取的序号与所述存储的序号进行比较，以认证所述页。

4、如权利要求 3 所述的智能卡，其中，所述存储的序号被存储在所述智能卡芯片中。

5、如权利要求 3 所述的智能卡，其中，所述存储的序号的一部分被存储在所述第二存储器器件上所存储的所述多页数据中的至少一个页中，该至少一个页的序号被存储在所述智能卡芯片中。

6、如权利要求 3 所述的智能卡，其中，所述序号被进行异或运算以产生存储在所述智能卡 EEPROM 中的摘要或哈斯数据（HASH）。

7、如权利要求 3 至 5 中的任一项所述的智能卡，其中，与特定数据页关联的序号在每次修改或更新所述页时被改变。

8、如权利要求 3 至 6 中的任一项所述的智能卡，其中，每个数据页的序号被初始设定为随机生成的值。

9、如权利要求 3 至 8 中的任一项所述的智能卡，其中，每个页包含其页号的拷贝。

10、如权利要求 3 至 9 中的任一项所述的智能卡，其中，每个页中的数据是加密的。

11、如权利要求 3 至 10 中的任一项所述的智能卡，其中，用密码 MAC 来保护每页数据的完整性。

12、如权利要求 3 到 11 的任一项所述的智能卡，其中，每个页加密和 MAC 都是使用一个关于页和芯片唯一的密钥来执行的。

存储器访问受保护的智能卡

技术领域

本发明涉及一种改进的智能卡。智能卡包括一个形状通常是矩形的衬底，该衬底与一个电子芯片形成整体，所述电子芯片能够存储与读卡器进行交互的数据和/或程序。

背景技术

智能卡越来越多地应用在识别和认证系统中，但是迄今为止，由于用于形成芯片的晶模（die）的大小限制了存储器的大小，因而限制了智能卡的实用性。随着越来越复杂的卡上应用，这种存储器大小的限制已经成为一个越来越严重的问题。

我们已经认识到通过附加一个第二存储器芯片（例如闪存 ROM（FLASH ROM））能够极大地扩展卡的存储容量。但是，随之而来的问题是：为使智能卡可以使用，保留在第二存储器器件中的数据的安全性必须像该数据存储在智能卡芯片内部时的安全性一样好。

发明内容

根据本发明的智能卡的特征在于：它在衬底上包括一个第二存储器器件并且该器件被可操作地连接到智能卡芯片。

优选地，第二存储器器件能够存储多个数据页，每个页具有一个与其相关联的唯一的序号，序号与数据页分开存储以便当读取页时，与该页一起获取的序号能够与存储的序号相比较，以认证该页。每个页被使用关于页和芯片唯一的密钥进行加密和签名。

附图说明

现在将参照附图，通过实例详细描述根据本发明的智能卡的实施例，其中：

图 1 显示了一个根据本发明的具有智能卡芯片和第二存储器器件的智能卡；

图 2 说明了用于与本发明的智能卡一起使用的外部存储器（XMEM）页的结构；

图 3 是本发明的智能卡中所应用的 XMEM 软件调用树的示意性图表。

具体实施方式

从图 1 中可以看出，本发明的智能卡 10 包括一个单独的衬底 12、主智能卡器件 14 以及一个外部存储器芯片 16，所述衬底 12 类似于一个传统智能卡衬底，但是其带有两个电子芯片，所述主智能卡器件 14 作为安全微控制器，所述外部存储器芯片 16 连接到智能卡微控制器。方便地，XMEM 器件 16 可以是一个闪存 ROM，但也可以使用任何其他非易失性或电子存储器器件。

在较低的层级，需要智能卡操作系统和 XMEM 之间的通信函数，用于安全地通信以及在与智能卡相连的 XMEM 上存储数据。这些功能被更高级的函数调用以读取和更新 XMEM 中的数据。XMEM 可以是，例如，ATMEL AT45DB321B 4M 字节串行数据闪存。与 XMEM 的通信是使用 ATMEL AT903232CS 串行外设接口（SPI）硬件实现的，并且输入/输出线路之一被用作芯片选择。然而，所述原理将适用于任何带有可用 I/O 的智能卡微控制器以与串行闪存器件互连。

在该实例中，每个 XMEM 页包括 528 字节。页结构在图 1 中示出。头 8 个字节（页头）包含一个指示该页未被擦除的字节（值不是 0xFF），5 字节随机数据，指示页号的 2 个字节以及 1 个字节的序号。页头后面是 512 字节的数据。末尾的 8 字节包含和 512 数据字节一起使用加密字组链接而加密的页头的拷贝。页头未被加密以便允许芯片推导页密钥。

我们所提供的 XMEM 驱动器实现了下列特性以增强安全性和可靠性。

- 页内的 512 数据字节是被三重 DES（Data Encryption Standard：

数据加密标准）CBC（Cipher Block Chaining：加密字组链接）加密的。任何对该数据的改变将改变 MAC（Message Authentication Cryptogram：消息认证码）并使其无效。

- 每个页包含 8 字节三重 DES MAC。
- 利用密码在每个页插入它的页号以允许确定所读的页就是所请求的页。通过页 MAC 来保护页号不被修改。
- 用于推导页密钥的主 DES 密钥对于芯片是唯一的，并在芯片第一次重置时自动地在内部生成。这些 DES 密钥不能从外部读取或更新。
- 在每次更新页时，由主密钥、随机数据、页号以及页序号重新生成对页进行加密和签名的 DES 密钥。这是一种额外的安全特性，当每次更新页而改变密钥进而因此改变 MAC 时，该安全特性增大了已知的为获取密钥而进行的文本攻击的复杂度。
- 每个页包含一个字节的序号，每次页更新时该序号都增加。该序号在页读取操作被验证。该序号是个随机数，在每次页更新时改变；因此不能从页的总更新次数中推导出该序号。页序号的使用提高了通过向同一页提供先前的内容而进行的攻击的复杂度。由于页将具有有效的 MAC 和有效的页号，如果没有序号这种攻击将是可能的。
- 所有对 XMEM 的更新都通过在编程后读取 XMEM 来验证。
- 硬件抽象层（HAL）函数将试图在退出前读取或更新该页 3 次。
- 如果一个页被发现要擦除，它在读取时被初始化为一个随机数值。那么，在外部擦除一个页以使在内部读取具有已知擦除值的页是不可能的。

序号

ATMEL AT45DB321B 包含 8192 页的闪存存储器。如上所述，每个 XMEM 页具有一个单独的序号（1 字节）。序号的拷贝必须被存储

在其他地方以在读取时与该页进行比较。为了阻止序号的拷贝被修改，它必须被保护。这可以通过在内部将所有序号存储到智能卡上来实现。这可能不适合于所有情况，因为它需要为序号保留 8192 字节 EEPROM。通过预留 32 页 XMEM，其中每个页存储另外 8160 个页的 256 个序号来解决该问题。这些页正常地被保护，但是它们的序号存储在智能卡 EEPROM 中。任选地，这 32 个序号可以被异或（XOR）以产生一个存储在智能卡 EEPROM 中的单独的字节。这导致节省了 EEPROM 的使用，使其从 8192 字节降到了 256 字节。

下面的图 2 详细描述了对如下的外部读取和更新页函数的调用顺序。

ReadXMEMPage：

void readXMEMPage（word pageNum）

这个函数将从XMEM读取一个页。该函数将调用函数getPageSeqNum以读取期望的页序号以与接收到的页进行比较。

该函数将调用 **doRead** 函数以执行页的实际读取、解密和 MAC 校验。

如果读取的页序号不正确的话将返回一个错误。

updateXMEMPage：

void readXMEMPage（word pageNum）

这个函数将更新 XMEM 中的一个页并要求提前读取该页以获取现有的页序号。

这个函数将调用 **loadPageKeys** 函数以基于页号、已更新的序号和随机数据为该页推导新的密钥。

这个函数将通过使用 SPI 硬件向 XMEM 芯片发送程序指令和数据来执行实际的 XMEM 页的更新。

这个函数将调用 **doRead** 函数以验证正确编入 XMEM 中的已更新的数据。

doRead:（internal Function）

void doRead（word pageNum, byte mode）

这个函数将从 XMEM 中读取一个页或者验证一个 XMEM 中的页。它具有两个模式：

1. 它将读取页，调用 **loadPageKeys** 以推导出页密钥来解密数据，并检查 MAC 和页号是否正确。
2. 它将读取该页以验证发送来更新XMEM页的加密数据是否被正确地编写。

loadPageKeys:

void loadPageKeys（byte mode,word pageNum）

这个函数将加载该密钥以加密/解密一个页，并且如果更新页，则使用随机数据、页号和已增加的页序号生成密钥多样化字符串，或者当读取它时，使用第一 XMEM 页作为多样化字符串。多样化字符串被使用关于芯片唯一的主 XMEM 密钥加密以给出关于芯片、页和序号唯一的密钥。该密钥被加载进 DES 硬件中。

getPageSeqNum:

byte readXMEMPage（word pageNum）

这个函数将为页返回页序号。这个函数可能已经被 **readXMEMPage** 函数调用以返回页序号。它将对 **readXMEMPage** 函数进行递归调用以读取包含所请求的最初的页序号的 XMEM 页。

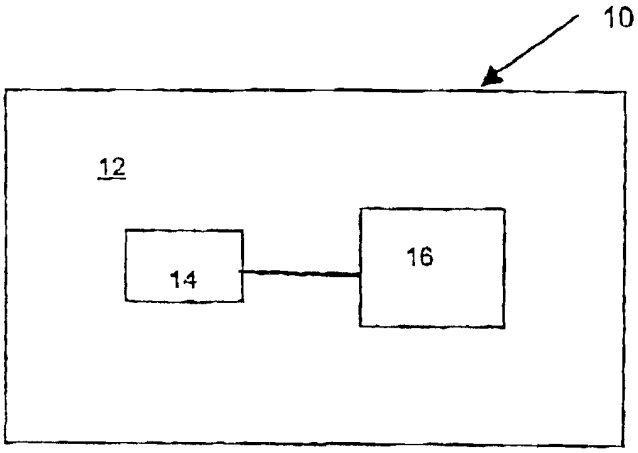


图1

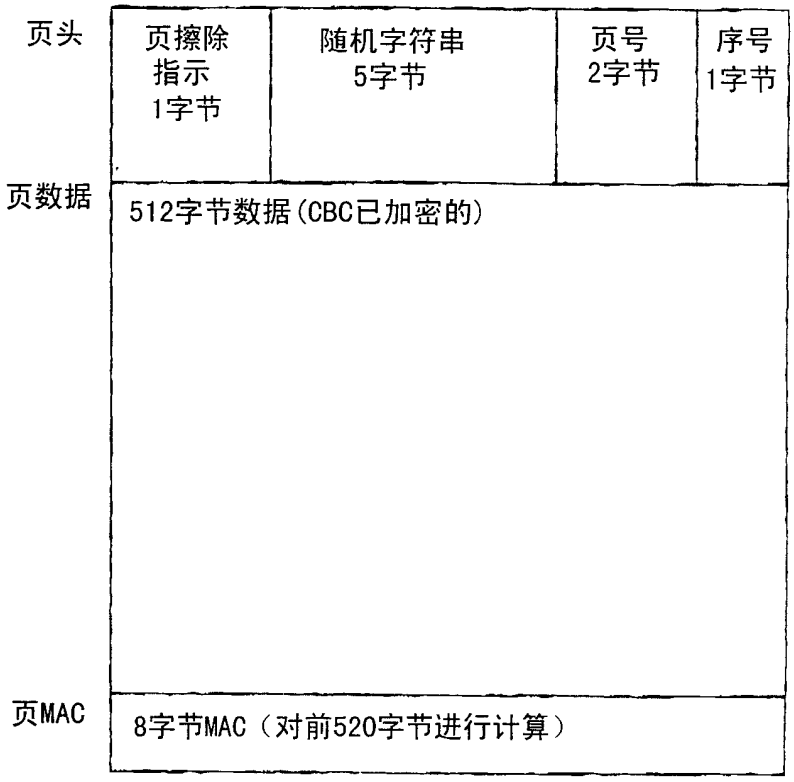

页头
页擦除
指示
1字节
随机字符串
5字节
页号
2字节
序号
1字节
页数据
512字节数据(CBC已加密的)
页MAC
8字节MAC（对前520字节进行计算）

图2

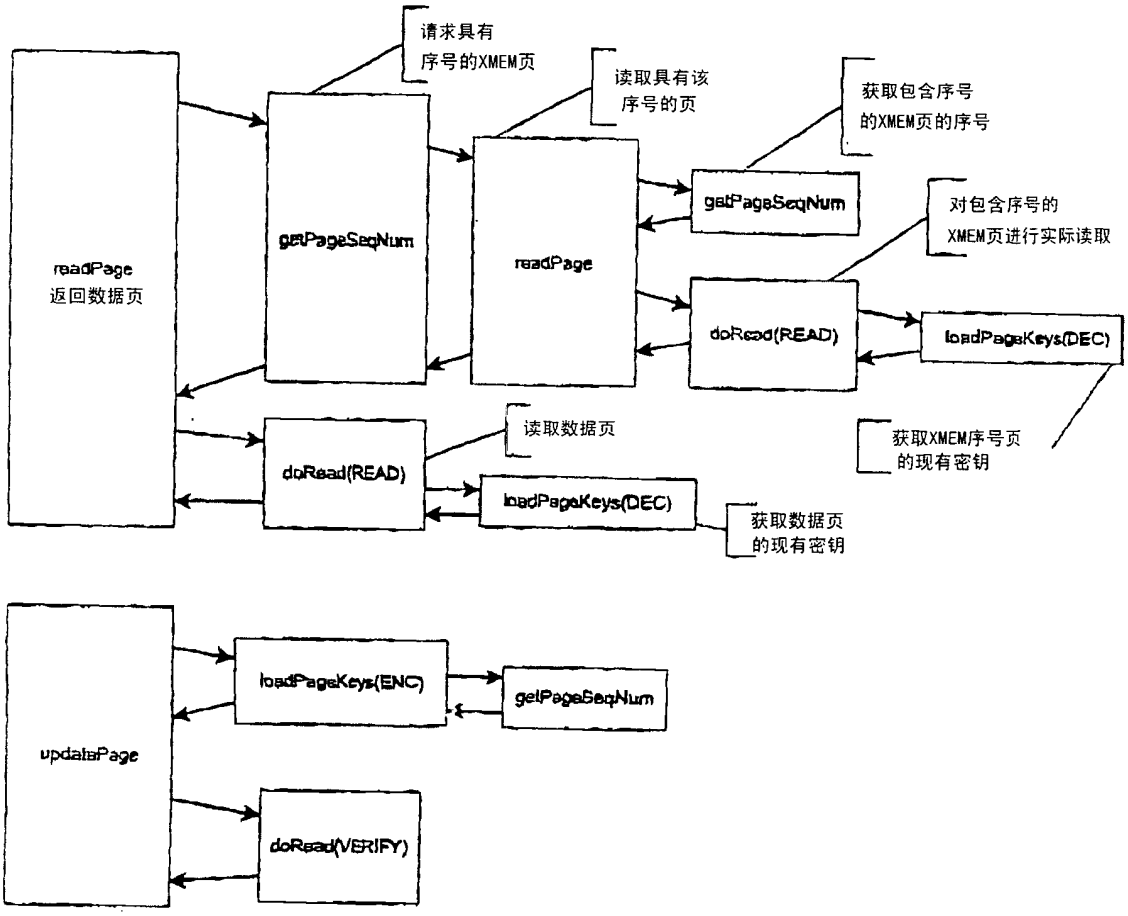

请求具有
序号的XMEM页
读取具有该
序号的页
获取包含序号
的XMEM页的序号
对包含序号的
XMEM页进行实际读取
readPage
返回数据页
getPageSeqNum
readPage
getPageSeqNum
doRead(READ)
loadPageKeys(DEC)
获取XMEM序号页
的现有密钥
读取数据页
doRead(READ)
loadPageKeys(DEC)
获取数据页
的现有密钥
updatePage
loadPageKeys(ENC)
getPageSeqNum
doRead(VERIFY)

图3