(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101038619 B
(45)授权公告日 2010.05.19

(21)申请号 200710003363.0
(22)申请日 2007.02.06
(73)专利权人 中国科学院研究生院
地址 100049 北京市石景山区玉泉路 19 号（甲）
(72)发明人 胡磊 鲁力 韩劲松 刘云浩 倪明选
(74)专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 谢安昆 宋志强
(51)Int.Cl.
G06K 7/00(2006.01)
G06K 17/00(2006.01)

(56)对比文件
WO 2006/131861 A1,2006.12.14, 全文.
CN 1818923 A,2006.08.16, 全文.
US 2005/0018853 A1,2005.01.27, 全文.
CN 1726500 A,2006.01.25, 全文.

审查员 鞠博

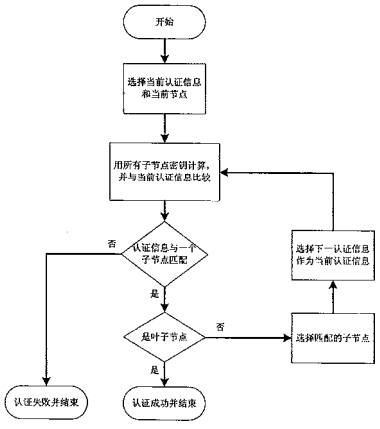
权利要求书 2 页 说明书 8 页 附图 6 页

(54)发明名称

射频识别系统隐私认证方法

(57)摘要

本发明涉及射频识别技术，公开了一种射频识别(RFID)系统中隐私认证的方法，该方法提供了读写器(Reader)和标签(Tag)之间的双向认证协议和系统密钥动态更新方案，使得在射频识别系统中读写器读取标签信息时不泄漏标签中存储的隐私信息。尤其是在不增加硬件的开销的情况下，本发明中所提出的方法能抵抗目前已知的对射频识别系统的所有攻击和追踪。



CN 101038619 B

1. 一种射频识别系统中隐私认证的方法，其特征在于，包含以下步骤：

A 系统初始化步骤，在建立所述射频识别系统时，系统生成密钥树，并对所述密钥树中每个非叶子节点分配多个相同的密钥，每个标签分配的密钥为密钥树中从根节点到每个标签对应的叶子节点路径上的密钥，

并设置密钥树各节点的状态；所述系统生成的密钥树为平衡树；

前向认证过程，即读写器认证所述标签的过程，包括以下 B 和 C 步骤：

B 标签识别步骤，所述读写器向所述标签发出认证请求，所述标签发送认证信息，收到认证信息后，所述读写器识别所述标签；

C 密钥更新步骤，所述读写器完成认证所述标签后，所述读写器更新密钥树，所述读写器生成身份认证码和密钥更新同步信息；

后向认证过程，即所述标签认证所述读写器的过程，包括所述标签验证所述读写器的身份认证码利根据密钥更新同步信息更新存储的密钥；

D 当系统动态变化时，即有所述标签加入或者离开射频识别系统时，执行系统维护步骤，所述系统维护系统密钥树。

2. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，所述方法的 B、C 步骤有三轮通讯，第一轮由所述读写器将认证请求和一个随机数发送给所述标签，第二轮由所述标签计算标签的认证信息，并发送给所述读写器，第三轮由所述读写器计算读写器的认证信息和密钥更新同步信息并发送给所述标签。

3. 根据权利要求 2 所述射频识别系统中隐私认证的方法，其特征在于，前向认证过程在协议的第一和第二轮通讯后完成，后向认证过程在协议的第三轮通讯后完成。

4. 根据权利要求 2 所述射频识别系统中隐私认证的方法，其特征在于，在第二轮通讯完成后，所述读写器先执行标签识别步骤，然后执行密钥更新步骤，并生成读写器身份认证信息和密钥同步信息。

5. 根据权利要求 2 至 4 之一所述射频识别系统中隐私认证的方法，其特征在于，在认证协议第二轮，所述密钥树中此次认证所涉及的密钥更新后，所述读写器生成密钥同步信息在认证协议第三轮发送给所述标签；所述标签根据收到的密钥同步信息更新所述标签中存储的密钥。

6. 根据权利要求 2 所述射频识别系统中隐私认证的方法，其特征在于，在第三轮通讯完成后，所述标签验证所述读写器的身份认证码，验证通过后根据密钥同步信息更新存储的密钥。

7. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，认证信息使用伪随机函数或密码学意义上的哈希函数计算。

8. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，所述步骤 B 中，所述认证信息由所述标签以分配的密钥使用伪随机函数或密码学意义上的哈希函数对所述读写器发出的认证请求计算产生。

9. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，所述步骤 B 中，所述读写器对所述密钥树使用深度优先搜索由上至下来确定所述标签所使用的密钥。

10. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，所述密钥树中每个非叶子节点均引入了一些状态位，对每个非叶子节点来说，状态位反映了其孩子节点

密钥更新的状态，每个状态位对应一个孩子节点，一旦某个孩子节点更新了密钥，所述系统设置父节点中该孩子节点对应的状态位，当某节点所有的孩子节点均更新密钥后，该节点更新其密钥；所述读写器仅对本次认证中所使用的密钥进行更新，并按照从叶子节点往上的顺序渐次更新所述密钥树中的密钥。

11. 根据权利要求 1 所述射频识别系统中隐私认证的方法，其特征在于，所述步骤 D 中，系统在有新的标签加入所述射频识别系统时，为该标签分配所述密钥树中的密钥；系统在有标签从所述射频识别系统中被撤销时，设置所述密钥树中节点状态，使所述标签被撤销后，不影响所述射频识别系统中其它标签的识别和密钥更新。

射频识别系统隐私认证方法

技术领域

[0001] 本发明涉及射频识别技术，特别涉及射频识别标签认证有关的安全技术。

背景技术

[0002] 射频识别 (Radio Frequency Identification，简称“RFID”) 是使用无线射频技术在开放系统环境中进行对象识别。这种技术的优点之一是无需物理或其它任何可见的接触。它是计算机系统和现实世界的联系纽带，为计算机感知和识别现实世界提供了一种高效、价廉的方式。

[0003] RFID 主要组成部分包括后台数据库 (Back-end database，简称“DB”)、读写器 (Reader) 和标签 (Tag) 等。其中后台数据库是运行于硬件平台的数据库系统，通常认为其具有强大的计算和存储能力，同时它包含系统中所有标签的信息。读写器 (Reader) 是一个带有天线的无线发射与接收设备，负责对标签中的信息进行读写。标签 (tag) 是带有天线的微型电路，通常没有处理器，仅由数千个逻辑门电路组成。标签中存储有唯一的身份标识 (ID) 和一些其它信息。整个 RFID 系统如图 1 所示。

[0004] 由于原始的标签，即只具备存储和通信能力而不具备计算能力的标签不能适应当前应用需求，特别是安全的 RFID 协议要求标签要具有计算能力。目前出现了一种在原始标签存储和通信功能基础之上增加了计算伪随机函数和生成随机数功能的新标签 (如 RF Code 公司的 Mantis™ 系列产品)，本发明中即使用这种新型标签。

[0005] 通常认为，标签与读写器之间的信道是不安全信道。而读写器与后台数据库之间的信道是安全信道。因此，在 RFID 系统安全通信协议的设计中，可以认为读写器与后台数据库是一体的 (即整体看作通信协议中的一方，另一方是标签)。因此，在本发明中，我们用“读写器”指代“读写器和后台数据库”。

[0006] 现有的 RFID 认证协议是典型的两轮“请求 - 响应”协议，其基本模型如图 2 所示。读写器首先向标签发出认证请求 ；然后标签返回认证响应信息，比如标签 ID 和存储的产品信息等等。

[0007] 两轮的 RFID 认证协议虽然简单，但是存在致命的安全缺陷，即使用者在被认证时的隐私信息将被泄漏。因为当读写器发出请求后，标签自动将它的唯一的序列号 (或 ID) 发送给读写器而不通知标签的使用者。在可扫描范围里，恶意的读写器能运行伪造的认证过程来探测标签以得到标签中记录的敏感信息。

[0008] 为了解决使用者隐私泄漏问题，现有的协议在 RFID 认证协议中使用了加密的方法。该认证方案如图 3 所示。每个标签与合法的读写器共享唯一的一个密钥 k。读写器将认证请求和一个随机数 r 发送给标签，标签用一个哈希函数，比如 MD-5、SHA-1 等计算加密结果并返回给读写器，读写器在后台数据库中搜索 k，直到找到某个密钥 k，满足 h(k，r) 与标签返回的认证加密消息相等为止。在这种体制中，存在严重缺陷 ：搜索密钥过程是线性搜索，即穷举搜索，效率太低。

[0009] 为了达到高效的隐私认证，现在出现了基于密钥树结构的隐私认证协议。该类协

议提供了对数级复杂度的搜索算法。在这类协议模型如图 4 所示，每个标签拥有多个密钥（比如 d 个）而不是一个。读写器构造一棵虚拟的层次树结构（称为密钥树，我们以一棵深度为 2 的树为例，如图 5 所示）来组织这些密钥。树中的每个节点存储一个密钥。每个标签与唯一的叶子节点相关联。从根节点到叶子节点路径上的所有密钥就是该叶子对应的标签所掌握的密钥。如果树的深度为 d 并且树的分支因子为 δ（密钥树设为平衡树），那么每个标签拥有 d+1 个密钥，为方便起见，记标签 T_i 中存储的密钥为 $(k_i^0, k_i^1, \ldots k_i^d)$。整棵树能支持 $N = d^{\delta}$ 个标签（即密钥树中有 N 个叶子节点）。在认证协议中，标签用 d 个密钥分别对同一随机数计算 d 次产生认证信息。收到标签的认证信息后，读写器在密钥树上做深度优先搜索找到一个叶子节点。

[0010] 上述基于树的协议具有效率高的特点，但是缺点也非常明显。由于缺少密钥更新体制，在敌手的主动攻击下是不安全的。这些协议中的密钥树为静态树（即树中存储的密钥不变）。由于树结构的特点，树中每个叶子节点，或多或少要与其它叶子节点共享一些密钥。这样，敌手通过物理破解（即用特定设备将标签中信息全部读出）一个或几个标签就能获得其它节点的密钥信息。申请人的研究表明，对于一个拥有 2^{20} 个标签的 RFID 系统，敌手仅仅需要破解 200 个标签，就能以超过 90％的概率成功跟踪识别系统中所有标签。

[0011] 前面所述的所有 RFID 隐私认证方案都是两轮协议，这些协议还有一个共同的缺点，即只有单向认证。所谓“单向认证”，即只有读写器能够认证标签的合法性，而标签不能认证读写器的合法性。这样标签在面对恶意读写器的扫描时，将会泄漏秘密信息。

发明内容

[0012] 基于上述两轮协议的缺点，本发明中提出了一个三轮的隐私认证方法，本发明中的协议方法具有很强的安全性，在克服了上述两轮协议所有缺点的同时，还具有如下三个主要优点：

[0013] 1. 认证过程效率高，具有对数级别的认证效率；

[0014] 2. 动态的密钥更新，能实时、动态的更新系统中的密钥，增强了抵抗主动攻击的能力；

[0015] 3. 双向身份认证（即读写器和标签都能认证对方身份的合法性）；

[0016] 为达到上述特点，本发明提供了一种具有强安全，低负荷特性的隐私认证协议(Strong and lightweightRFID Private Authentication，简称“SPA”)。各部分如下：

[0017] A 系统初始化，在建立 RFID 系统时，系统生成密钥树，并对所述密钥树中每个非叶子节点分配多个相同的密钥，每个标签分配的密钥为密钥树中从根节点到每个标签对应的叶子节点路径上的密钥，并设置密钥树各节点的状态；所述系统生成的密钥树为平衡树；

[0018] 前向认证过程，即读写器认证标签的过程，包括以下 B 和 C 步骤：

[0019] B 标签识别步骤，读写器向所述标签发出认证请求，标签发送认证信息，收到认证信息后，读写器识别标签；

[0020] C 密钥更新步骤，读写器完成认证标签后，读写器完成密钥树更新，并将更新信息通知标签；

[0021] 后向认证过程，即标签认证读写器的过程，包括读写器身份认证码生成和密钥更新同步。

[0022] D 当系统动态变化时，即有标签加入或者离开 RFID 系统时，执行系统维护步骤，系统维护密钥树。

[0023] 在所述方法中，前向认证过程在协议的第一和第二轮通信后完成，后向认证过程在协议的第三轮通信后完成。

[0024] 在所述方法中，认证信息使用伪随机函数计算。

[0025] 在所述方法中，认证信息可以使用密码学意义上的哈希函数计算。

[0026] 在所述方法中，系统生成密钥树，并对密钥树中每个节点分配多个密钥，并设置各个节点状态。

[0027] 在所述方法中，认证信息由标签以分配的密钥使用伪随机函数对读写器发出的认证请求计算产生。

[0028] 在所述方法中，认证信息由标签以分配的密钥使用密码学意义上的哈希函数对读写器发出的认证请求计算产生。

[0029] 在所述方法中，读写器对密钥树使用深度优先搜索来确定标签使用的密钥。

[0030] 在所述方法中，读写器仅对本次认证中所使用的密钥进行更新，并按照由底至顶的顺序渐次更新密钥树中的密钥。

[0031] 在所述方法中，密钥树中每个非叶子节点均引入了一些状态位，对每个非叶子节点来说，状态位反映了其孩子节点密钥更新的状态，每个状态位对应一个孩子节点，一旦某个孩子节点更新了密钥，系统设置父节点中该孩子节点对应的状态位，当某节点所有的孩子节点均更新密钥后，该节点更新其密钥。

[0032] 在所述方法中，在认证协议第二轮，密钥树中此次认证所涉及的密钥更新后，读写器生成密钥同步信息在认证协议第三轮发送给标签；标签根据收到的密钥同步信息更新标签中存储的密钥。

[0033] 在所述方法中，根据一个旧密钥生成新密钥，使用伪随机函数计算。

[0034] 在所述方法中，根据一个旧密钥生成新密钥，使用密码学意义上的哈希函数计算。

[0035] 在所述方法中，系统在有新的标签加入 RFID 系统时，为该标签分配密钥树中的密钥；系统在有标签从 RFID 系统中被撤销时，设置密钥树中节点状态，使标签被撤销后，不影响系统中其它标签的识别和密钥更新。

[0036] 在所述方法中，标签的身份，是该标签与读写器共享的多个密钥。

[0037] 通过比较可以发现，本发明的技术方案 (SPA) 与现有技术相比，主要优点在于，SPA 基于树结构，并为树结构提供了一致的动态密钥更新，在不增加硬件开销的同时，为 RFID 系统提供了双向认证。在认证效率上，与现有技术相比，其密钥搜索效率还是对数级，没有降低认证效率。SPA 提供了动态密钥更新，能有效抵抗现有的对 RFID 系统的所有被动和主动攻击，特别是现有技术不能抵抗的破解攻击。

附图说明

[0038] 图 1 是现有 RFID 系统示意图。

[0039] 图 2 是现有 RFID 系统认证协议示意图。

[0040] 图 3 是现有 RFID 系统中基本的隐私认证协议示意图。

[0041] 图 4 是现有 RFID 系统中基于树结构的隐私认证协议示意图。

[0042] 图 5 是现有 RFID 系统中基于树结构的隐私认证协议密钥树示意图。

[0043] 图 6 是本发明 RFID 系统隐私认证协议，读写器和标签间认证过程示意图。

[0044] 图 7 是本发明 RFID 系统隐私认证协议，标签识别流程图。

[0045] 图 8 是本发明 RFID 系统隐私认证协议，密钥更新流程图。

[0046] 图 9 是根据本发明小规模实施例的 RFID 系统密钥树示意图。

[0047] 图 10 是根据本发明小规模实施例的 RFID 系统隐私认证协议，密钥树更新的过程示意图。

[0048] 图 11 是根据本发明小规模实施例的 RFID 系统隐私认证协议，有新标签加入时密钥树状态示意图。

[0049] 图 12 是根据本发明小规模实施例的 RFID 系统隐私认证协议，有标签离开时密钥树状态示意图。

具体实施方式

[0050] 为使本发明的目的、技术方案和优点更加清楚，下面给出本发明的具体实施方式。

[0051] A. 系统初始化

[0052] 建立密钥树，将每个标签与一个叶子节点相关联。若系统中有 N 个标签，密钥树 S 的分支因子为 δ，深度为 d。密钥树中每个非叶子节点分配 a 个密钥，初始这些密钥均相同。另外每个非叶子节点分配 δ 个状态位，分别对应 δ 个孩子节点，初始每个状态位均为 0。标签中的密钥为从根节点到该标签对应的叶子节点路径上所有节点的密钥。注意，因为每个非叶子节点有 a 个密钥，要从其中选择一个，又因为初始这 a 个密钥相同，故只需要取任意一个密钥即可。

[0053] 下面介绍本发明的三轮认证协议，如图 6 所示。

[0054] 前向认证过程：

[0055] 在协议第一轮通信中，读写器生成一个随机数 r_1，并将 r_1 同认证请求发送给标签。

[0056] 标签收到认证请求和随机数 r_1 后，使用伪随机函数或者密码学意义上的哈希函数进行计算，函数的输入为随机数 r_1 和标签中存储的密钥，输出为标签的认证信息。最后生成一个随机数 r_2，并将 r_2 和认证信息作为协议第二轮通信发送给读写器。

[0057] 读写器收到标签的认证信息后，执行以下步骤：

[0058] 步骤 1，根据标签的认证信息执行“标签识别”过程，即所述读写器向所述标签发出认证请求，所述标签发送认证信息，收到认证信息后，所述读写器识别所述标签，以认证标签的合法身份；

[0059] 步骤 2，标签的身份认证通过后，执行“密钥更新”过程，即所述读写器完成认证所述标签后，所述读写器更新密钥树，所述读写器生成身份认证码和密钥更新同步信息，从而更新本次认证所涉及的密钥树中密钥，并使用随机数 r_1、r_2 和更新后的一个密钥生成读写器的身份认证信息。

[0060] 后向认证过程：

[0061] 在协议第三轮，读写器将上面步骤 2 生成的身份认证信息和密钥更新同步信息发送给标签。收到这些信息后，标签首先验证读写器身份认证信息的合法性，然后根据密钥更新同步信息更新存储的密钥。

[0062] D. 系统维护

[0063] 当有标签加入RFID系统时，读写器检查密钥树S中是否有空的叶子节点。如果有，读写器将该标签与空叶子节点关联，并将从根到该叶子节点路径上的密钥分配给标签。如果没有空叶子节点，读写器先创建一个新的深度为d-1（即比系统密钥树层数少1层）的密钥树S'，S'的分支因子也为 δ 。S'的创建过程如系统初始化步骤A所述。然后将S'的根节点与S的根节点相连，即S'根节点的父节点为S的根节点，S'成为S的一棵子树。再将标签与S'中一个空叶子节点相关联并分配密钥。

[0064] 当系统中有标签离开后，读写器锁住相应叶子节点的父节点中对应状态位为1，直到有新标签分配到该空叶子节点，再将相应状态位置为0。这样做的目的是保护系统中其它标签的密钥更新不受影响。

[0065] 所述B“标签识别”过程如下：

[0066] 从密钥树根节点开始，读写器首先用根节点的a个密钥分别计算r_1，并与标签发送的认证信息中第一项比较，如果其中一个计算结果与认证信息第一项（第零项是标签生成的随机数r_2）存在匹配，读写器进入密钥树下一层，用该层中各个节点的密钥计算r_1，如果有一个节点的计算结果与认证信息第二项匹配，则从该节点进入下一层。执行同样的过程，直到标签所属叶子节点结束，由上至下，逐层识别。上述过程是递归过程，其流程如图7所示。

[0067] 所述C“密钥更新”过程如下：

[0068] 每个非叶子节点有 δ 个状态位（δ 是树的分支因子）。对每个非叶子节点来说，各状态位反映了对应孩子节点密钥更新的状态。每个状态位对应一个孩子节点。

[0069] 密钥更新条件：一旦某个孩子节点更新了密钥，父节点中该孩子节点对应的状态位就设置为1。当所有的状态位均为1时，该节点更新其密钥。

[0070] 每次认证成功后，读写器先更新标签对应叶子节点的密钥，然后用上述方法从叶子节点的父节点开始，由下至上，逐层更新。密钥更新的过程是递归过程，其流程如图8所示。

[0071] 样例

[0072] 为使本发明的目的、技术方案和优点更加清楚，下面介绍一个小实施例，即系统中密钥树是深度为2的二叉树，每个非叶子节点包含两个密钥和两个状态位，系统共能容纳四个标签，并结合附图9至12对本发明做进一步的详细描述。

[0073] A 系统初始化

[0074] 如图9所示，该实施例包含四个标签，即系统的密钥树S包含四个叶子节点，每个标签与一个叶子节点相关联。S中每个非叶子节点j被分配两个密钥，工作密钥k_j和临时密钥tk_j。每个密钥都是由系统随机均匀生成，初始$tk_j = k_j$。另外，每个非叶子节点j分配两个1比特状态位s_j^1和s_j^r，初始$s_j^l = s_j^r = 0$。s_j^1和s_j^r将会在密钥更新步骤中使用。

[0075] 系统初始化时，每个标签分配的密钥为密钥树中从根节点到其对应的叶子节点路径上的密钥。例如图9中标签T_1，读写器为其分配的密钥为$(k_0, k_{1,1}, k_{2,1})$。由于初始$tk_j = k_j$，故为T_1分配tk或k均可。为描述方便起见，我们记标签T_i中分配的密钥为$(k_0^i, k_1^i, \ldots, k_d^i)$，这里d为密钥树的深度，图9中所示实施例中，标签$T_1$中分配的密钥记为$(k_0^1, k_1^1, k_2^1)$。

[0076] 前向认证过程

[0077] 如图 6 所示，在协议第一轮中，读写器向标签 T_i 发送一个认证请求和一个随机数 r_1。当 T_i 收到请求后，随机生成一个随机数 r_2 并计算序列 $(h(k_0^i, r_1), h(k_1^i, r_1), h(k_2^i, r_1))$。$T_i$ 向读写器返回认证信息 $U = \left(r_2, h\left(k_0^i, r_1\right), h\left(k_1^i, r_1\right), h\left(k_2^i, r_1\right)\right)$。为描述方便起见，我们记 U = (u, v_0, v_1, v_2)。

[0078] 读写器收到标签的认证信息后，执行以下步骤：

[0079] 步骤 1，根据标签的认证信息执行“标签识别”过程，即所述读写器向所述标签发出认证请求，所述标签发送认证信息，收到认证信息后，所述读写器识别所述标签，以认证标签的合法身份；

[0080] 步骤 2，标签的身份认证通过后，执行“密钥更新”过程，即所述读写器完成认证所述标签后，所述读写器更新密钥树，所述读写器生成身份认证码和密钥更新同步信息，从而更新本次认证所涉及的密钥树中密钥，并使用随机数 r_1、r_2 和更新后的一个密钥生成读写器的身份认证信息和密钥更新同步信息。

[0081] 步骤 1 中，“标签识别”过程如下：从密钥树根节点开始，读写器首先用 k_0 和 tk_0 加密 r_1（即计算 $h(k_0, r_1)$ 和 $h(tk_0, r_1)$），并与 T_i 发送的认证信息中 v_0 比较，如果存在匹配，读写器调用递归算法来识别 T_i，在密钥树中，从根节点开始到 T_i 所属叶子节点结束，由上至下，逐层识别。以图 9 中标签 T_1 为例。读写器从根节点开始计算 $h(k_0, r_1)$ 和 $h(tk_0, r_1)$，并与 T_1 发送的认证信息中 v_0 比较。若存在匹配，读写器计算 $h(k_{1,1}, r_1)$、$h(tk_{1,1}, r_1)$、$h(k_{1,2}, r_1)$ 和 $h(tk_{1,2}, r_1)$，并与 v_1 比较。若 v_1 与 $h(k_{1,1}, r_1)$ 或 $h(tk_{1,1}, r_1)$ 匹配，读写器计算 $h(k_{2,1}, r_1)$ 和 $h(tk_{2,2}, r_1)$，并与 v_2 比较，因为 $v_2 = h(k_{2,1}, r_1)$，读写器识别该标签为 T_1。

[0082] 简单地来说，读写器识别标签的过程就是读写器在密钥树上做深度优先搜索的过程。即读写器从密钥树根节点开始，用节点密钥和认证请求中的随机数作为输入计算哈希函数，并将结果与标签发出的认证信息比对。由上至下，在密钥树中每一层找到一个匹配节点后，从该节点进入下一层，直到叶子节点为止，此时读写器已识别出标签。

[0083] 步骤 2 中，“密钥更新”过程如下：在生成新密钥时，与认证信息生成一样，使用哈希函数 h。设 k_j 是密钥树中节点 j 的旧密钥，读写器从该旧密钥计算新密钥 $k_j' = h\left(k_j\right)$。为不影响其它标签的认证，对更新了密钥的非叶子节点 j，本发明使用临时密钥 tk_j 来存储 j 的旧密钥。这样可以不影响属于 j 的子树中其它标签的认证过程。

[0084] 如图 9 所示，该实施例密钥树中每个非叶子节点 j 包含两个孩子，那么有两个状态位 s_j^l 和 s_j^r，其中 $s_j^l, s_j^r \in \{0,1\}$。系统初始化时 $s_j^l = s_j^r = 0$。在任意时刻，如果 j 的左或右孩子节点更新了密钥，则读写器将相应的 s_j^l 或 s_j^r 置为 1。此时，如果 $s_j^l = s_j^r = 1$，读写器先将 j 的旧密钥 k_j 存储到 tk_j 中，然后更新 k_j，并将 j 的父节点相应状态位置 1。由下之上重复上述过程，直到根节点或某个节点不满足更新密钥条件为止。

[0085] 读写器身份认证信息和密钥更新同步信息生成过程如下：当读写器完成密钥更新，计算 $\sigma = h\left(k_2^i, r_1, r_2\right)$，并将 σ 与“同步信息”发送给标签 T_i。σ 是读写器的身份认证码，用于标签 T_i 认证读写器；“同步信息”包括密钥树中更新了密钥的节点所在的层次信息，用以通知标签在认证读写器之后更新标签中与读写器共享的密钥，以达到与密钥树当前状态同步。

[0086] 举例来说，如图 9 所示，该实施例中，假设读写器完成对 T_2 的认证，并更新第一层

和第二层的密钥 $k_{1,1}$ 和 $k_{2,2}$，那么“同步信息”为 (1,2)。

[0087] 为了更好地描述本发明中密钥更新的过程，我们以下面的例子来说明本发明中小规模实施例中的密钥更新的过程。假设系统中四个标签 T_1，T_2，T_3 和 T_4 的认证顺序为 (T_1，T_2，T_3，T_4)。密钥树状态变化如图 10 所示。

[0088] 初始状态如图 10(a) 所示。当标签 T_1 被认证后，读写器置 T_1 的父节点中与 T_1 相关的状态位为 1。另外，读写器为叶子节点 T_1 生成一个新密钥。此时密钥树状态如图 10(b) 所示。

[0089] 当标签 T_2 被认证后，T_2 的父节点中相应的状态位被置为 1，读写器更新叶子节点 T_2 及其父节点的密钥。如图 10(c) 所示。

[0090] 当标签 T_3 被认证后，此时情况与标签 T_1 被更新后情况类似。需要指出的是，因为 T_1 和 T_2 的父节点的所有状态位均被置为 1，读写器会将 T_1 和 T_2 的父节点的所有状态位重置为 0，这样当 T_1 或 T_2 下次被认证时，其密钥将可以再次更新。此时密钥树状态如图 10(d) 所示。

[0091] 当标签 T_4 被认证后，T_4 的父节点和根节点所有的状态位都为 1。因此读写器更新从叶子节点 T_4 到根节点路径上所有的密钥。此时密钥树状态如图 10(e) 所示。

[0092] 因为标签 T_1 不知道 T_1 的父节点和根节点的密钥已被更新，在第二次认证 T_1 时，它将继续使用旧密钥。如本发明中密钥更新算法所描述，每个节点将旧密钥存储在临时密钥 tk 中。根据本发明中标签识别算法所描述，T_1 能被成功认证。在 T_1 被认证后，读写器通过发送给 T_1 的“同步信息”通知 T_1 更新从 T_1 的叶子节点到根节点的密钥。此时情况如图 10(f) 所示。

[0093] 后向认证过程

[0094] 在协议第三轮，读写器将所述步骤 2 生成的身份认证信息和密钥更新同步信息发送给标签。收到这些信息后，标签首先验证读写器身份认证信息的合法性，然后根据密钥更新同步信息更新存储的密钥。

[0095] 在本实施例中，当 T_i 收到这些消息后，T_i 首先校验 $\sigma = h(k_2^i, r_1, r_2)$ 是否成立。如果成立，T_i 根据“同步信息”更新自己掌握的密钥。比如，在本实施例中，如果 T_2 收到“同步信息”为 (1,2)，则分别计算新密钥 $k_1^2 = h(k_1^2)$ 和 $k_2^2 = h(k_2^2)$。

[0096] D 系统维护

[0097] 当有标签 T_i 加入 RFID 系统时，读写器检查密钥树 S 中是否有空的叶子节点。如果有，读写器将 T_i 与空叶子节点关联，即将从根到该叶子节点路径上的密钥分配给 T_i。如果没有空叶子节点，读写器先创建一个新的深度为 d-1（即比系统密钥树层数少 1 层）的密钥树 S’，S’ 的分支因子与 S 一样，S’ 的创建过程如系统初始化步骤所述。然后将 S’ 的根节点与 S 的根节点相连，即 S’ 根节点的父节点为 S 的根节点，S’ 成为 S 的一棵子树。再将 T_i 与 S’ 中一个空叶子节点相关联并分配密钥。如图 11 所示，当有新标签 T_5 加入系统时，读写器创建一棵新的子树并为 T_5 分配一个叶子节点。为 T_5 分配的密钥为 (k_0，$k_{1,3}$，$k_{2,5}$)。当系统中有标签 T_i 离开后，读写器锁住 T_i 父节点的对应状态位为 1，直到有新标签分配到该空叶子节点，再将相应状态位置为 0。为状态位加锁的目的是保护系统中其它标签的密钥更新不受影响。如果不这样做，因为叶子 T_i 为空，那么 T_i 父节点对应状态位将一直为 0，根据本发明中密钥更新算法，从 T_i 到根节点路径上所有非叶子节点中的密钥将不会得到更新，

这样就影响了其它标签的密钥更新。所以必须将 T_i 父节点对应状态位置 1，以允许其它标签更新密钥。如图 12 所示为当 T_3 离开系统时，系统密钥树状态。

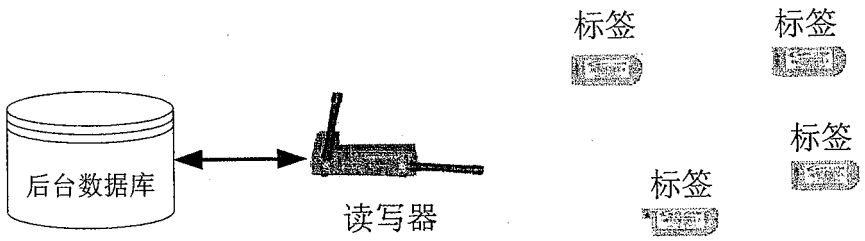

标签
标签
标签
标签
后台数据库
读写器

图 1

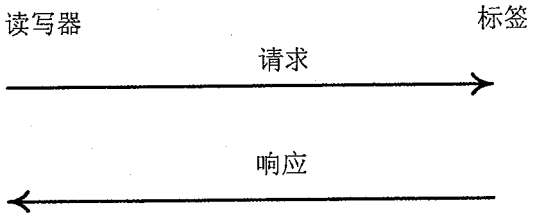

读写器
标签
请求
响应

图 2

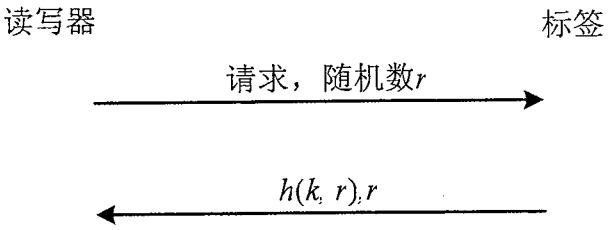

读写器
标签
请求，随机数r
h(k, r), r

图 3

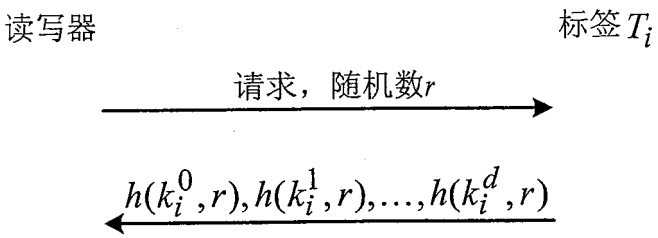

读写器
标签 T_i
请求，随机数r
$h(k_i^0, r), h(k_i^1, r), \ldots, h(k_i^d, r)$

图 4

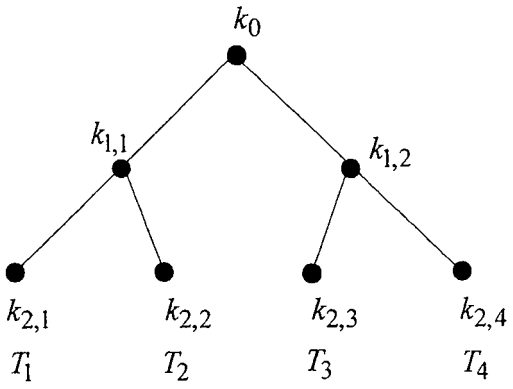

k_0
$k_{1,1}$
$k_{1,2}$
$k_{2,1}$
$k_{2,2}$
$k_{2,3}$
$k_{2,4}$
T_1
T_2
T_3
T_4

图 5

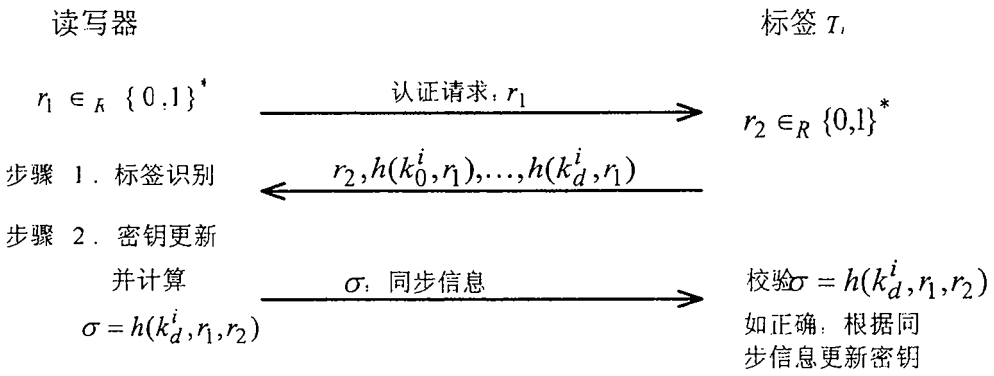

读写器
标签 T_i
$r_1 \in_R \{0,1\}^*$
认证请求：r_1
$r_2 \in_R \{0,1\}^*$
步骤 1．标签识别
$r_2, h(k_0^i, r_1), \ldots, h(k_d^i, r_1)$
步骤 2．密钥更新
并计算
$\sigma = h(k_d^i, r_1, r_2)$
σ：同步信息
校验$\sigma = h(k_d^i, r_1, r_2)$
如正确：根据同
步信息更新密钥

图 6

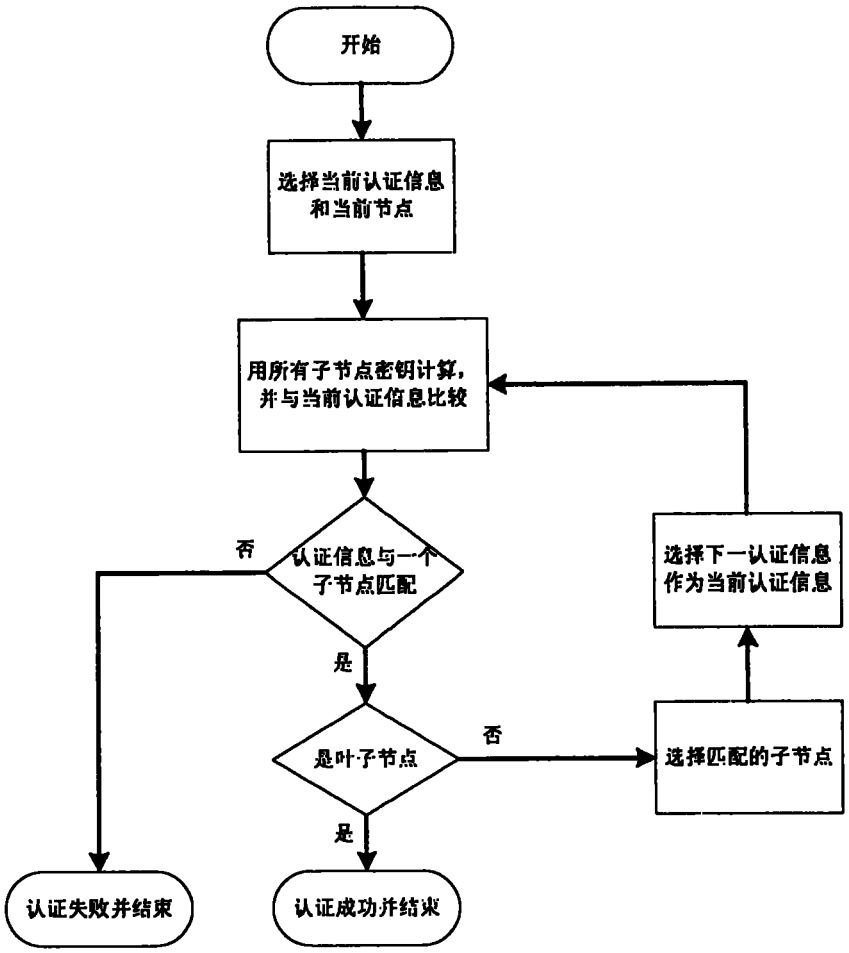

开始
选择当前认证信息和当前节点
用所有子节点密钥计算，并与当前认证信息比较
认证信息与一个子节点匹配
否
是
是叶子节点
否
是
选择匹配的子节点
选择下一认证信息作为当前认证信息
认证失败并结束
认证成功并结束

图 7

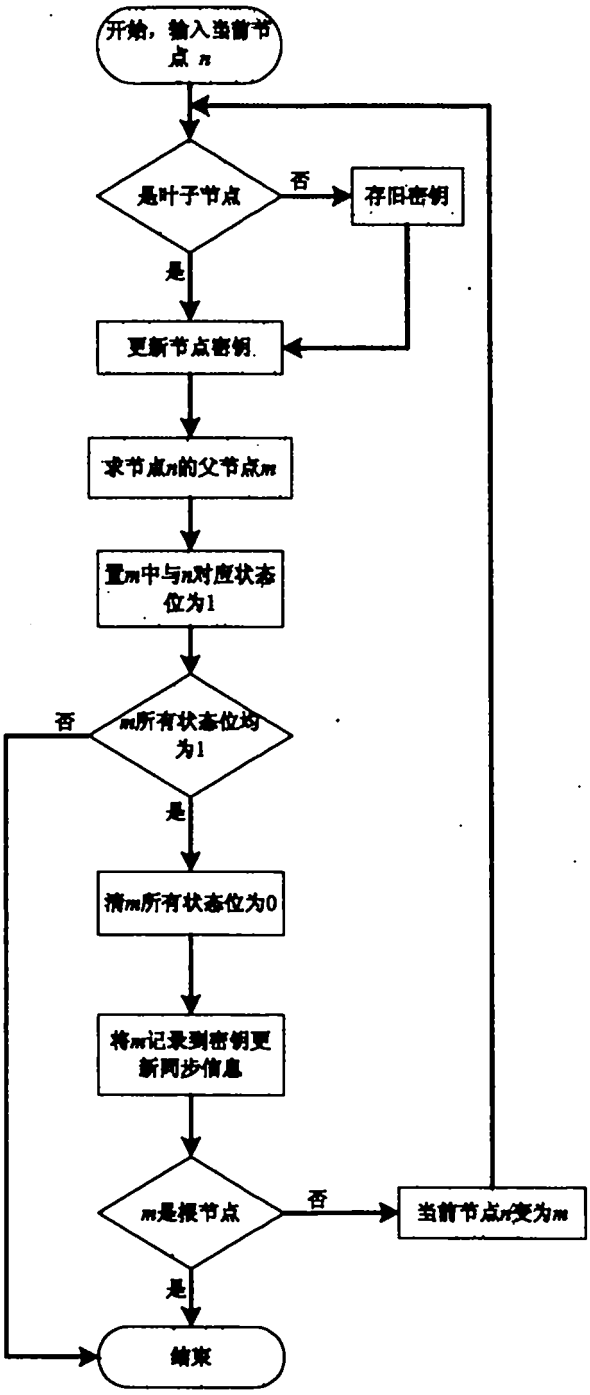

开始，输入当前节点 n
是叶子节点
否
存旧密钥
是
更新节点密钥
求节点n的父节点m
置m中与n对应状态位为1
否
m所有状态位均为1
是
清m所有状态位为0
将m记录到密钥更新同步信息
m是根节点
否
当前节点n变为m
是
结束

图 8

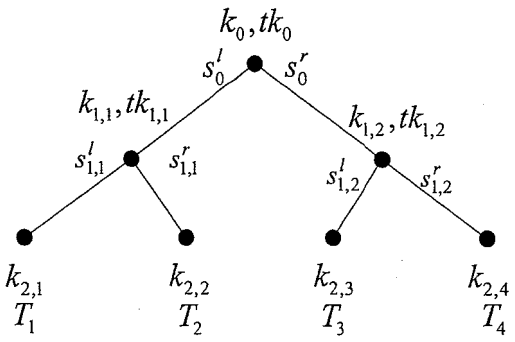
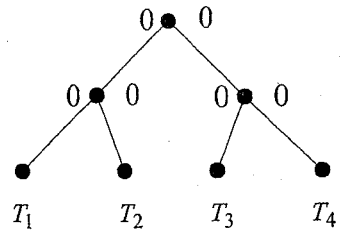
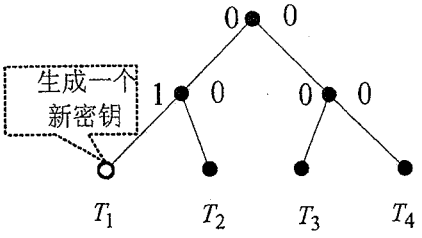

k_0, tk_0
s_0^l
s_0^r
$k_{1,1}, tk_{1,1}$
$k_{1,2}, tk_{1,2}$
$s_{1,1}^l$
$s_{1,1}^r$
$s_{1,2}^l$
$s_{1,2}^r$
$k_{2,1}$ T_1
$k_{2,2}$ T_2
$k_{2,3}$ T_3
$k_{2,4}$ T_4

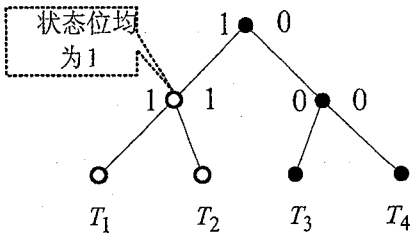
图 9



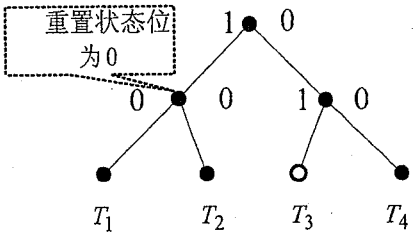
(a)密钥树初始状态



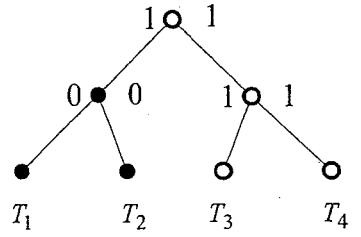
(b)认证标签 *a* 后密钥树状态



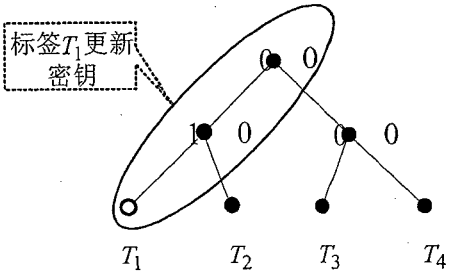
(c)认证标签 *b* 后密钥树状态



(d)认证标签 *c* 后密钥树状态



(e)认证密钥树 *d* 后密钥树状态



(f)第二次认证 *a* 后密钥树状态

图 10

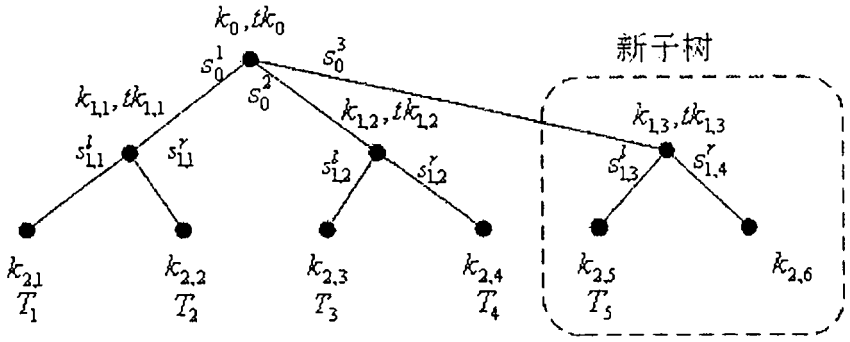


图 11

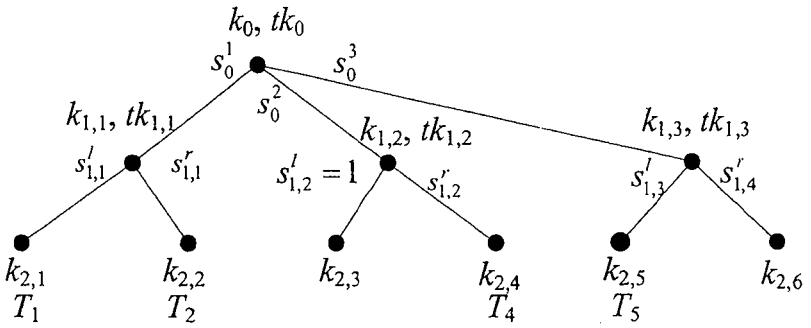

k_0, tk_0
s_0^1
s_0^2
s_0^3
$k_{1,1}, tk_{1,1}$
$k_{1,2}, tk_{1,2}$
$k_{1,3}, tk_{1,3}$
$s_{1,1}^l$
$s_{1,1}^r$
$s_{1,2}^l = 1$
$s_{1,2}^r$
$s_{1,3}^l$
$s_{1,4}^r$
$k_{2,1}$ T_1
$k_{2,2}$ T_2
$k_{2,3}$
$k_{2,4}$ T_4
$k_{2,5}$ T_5
$k_{2,6}$

图 12