第 33 卷 第 3 期 计 算 机 工 程 2007 年 2 月
Vol.33 No.3 Computer Engineering February 2007
·安全技术· 文章编号：1000—3428(2007)03—0151—03 文献标识码：A 中图分类号：TP309.1

Key 值更新随机 Hash 锁对 RFID 安全隐私的加强

曾丽华，熊 璋，张 挺
(北京航空航天大学计算机学院，北京 100083)

摘 要：RFID 无线通信的方式和无可视性读写要求带来了很多安全隐患，针对 RFID 技术在安全隐私方面存在的威胁，在分析几种典型的 RFID 安全隐私保护方法的特点和局限的基础上，提出了一种新的方法——Key 值更新随机 Hash 锁。该方法使用单向 Hash 函数添加随机 Hash 锁，并在每次通信过程中更新标签 Key 值，且标签与阅读器之间的数据传输都经过了 Hash 加密，有效地防止了非法读取、位置跟踪、窃听、伪装哄骗、重放等攻击。分析表明，该方法具有成本低、前向安全、负载小、效率高、安全性好等特点，适用于标签数目较多的情况。

关键词：RFID；安全隐私；Key 值；随机 Hash 锁

Key Value Renewal Random Hash Lock for Security and Privacy Enhancement of RFID

ZENG Lihua, XIONG Zhang, ZHANG Ting
(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】RFID brings in security and privacy hidden trouble because of its wireless communication mode and no demand for visibility, a new approach——key value renewal random lock is proposed based on the analysis of some kinds of typical methods. Through taking in the random hash lock with one-way Hash function, renewing the tag's key value in each communication process and encrypting the data transported between tags and reader by Hash function, this approach efficiently prevents unauthorized reading, location track, wiretapping, counterfeit and spoofing, replay attack and so on. Analysis shows that this approach is low-cost, forward secure, low-load, high efficiency, good security and fits for the scenario when the tags number is large.

【Key words】RFID; Security and privacy; Key value; Random Hash lock

RFID 是 20 世纪 90 年代兴起的一项非接触式的自动识别技术，其无线通信方式和无可视性读写的要求，给我们带来了极大方便，也带来了很多安全隐私问题。针对 RFID 的安全隐私问题，目前国内外开展了很多加强 RFID 安全隐私保护的研究，并提出了一系列的方法，如 Hash 锁、随机 Hash 锁和 Hash 链，但这些方法存在安全性不高或效率低等缺陷。本文针对现有方法的不足，进一步对 RFID 的安全隐私保护展开研究。

1 RFID 技术及其安全隐私分析

RFID 系统主要由阅读器、标签及后端数据库组成，如图 1 所示。

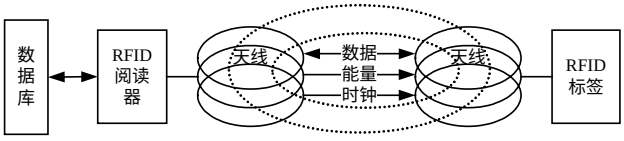

图 1 RFID 系统架构

目前RFID技术已经吸引了工业和学术界越来越多的关注，并已广泛地应用于供应链管理、门禁控制、电子钱包[1]等各种场合。然而RFID技术属于非接触式自动识别技术，其面临的安全隐私威胁主要有：

(1)非法读取。商业竞争者可通过未授权的阅读器快速读取超市的商品标签数据，获取重要的商业信息；

(2)位置跟踪。通过 RFID 标签扫描，依据标签的特定输出可对消费者位置进行跟踪定位；

(3)窃听。因 RFID 系统在前向信道的信号传输距离较远，窃听者可轻易窃取阅读器发出的信号数据；

(4)拒绝服务。人为的信号干扰使得合法阅读器不能正常阅读标签数据；

(5)伪装哄骗。通过伪装成合法标签，哄骗阅读器为其提供错误的数据；

(6)重放。根据窃听到的阅读器和标签间的数据通信，重复之前的通信行为从而获取数据信息。

2 RFID 安全隐私保护

RFID 的安全隐私问题阻碍了 RFID 技术的进一步推广，引起了消费者的高度关注，加强对 RFID 的安全隐私保护有着极其重要的意义。

2.1 前提与要求

假定阅读器与后台数据库的通信是在一条安全可靠的有连接信道上进行，但阅读器与标签之间的无线通信易被窃听。

要普及RFID技术，必须保证RFID标签的低成本实现。由于标准的安全机制要求的计算比较复杂，如SHA-1 约需 12K 个门，这在低成本标签上无法实现，因此可采用低成本的单

基金项目：北京市教育委员会重点学科共建项目(XK100060423)
作者简介：曾丽华(1982－)，女，硕士生，主研方向：多媒体技术；熊 璋，教授、博导；张 挺，博士
收稿日期：2006-02-22 **E-mail：**lovely-lihua@163.com

向Hash函数进行加密[2]。

安全的 RFID 系统应能抵御各种攻击，且考虑到较坏的情况，即使外人获得了标签内部的秘密数据，也应保证其无法追踪到跟标签有关的历史活动信息，即保证前向安全性。

2.2 典型方法

典型的加强 RFID 安全隐私保护的访问控制方法主要有 Hash 锁、随机 Hash 锁和 Hash 链，它们都是基于单向 Hash 函数实现的。

2.2.1 Hash锁(Hash Lock)[2]

采用 Hash 锁方法控制标签的读取访问，其工作机制如下：

锁定标签：对于唯一标志号为 ID 的标签，首先阅读器随机产生该标签的 Key，计算 metaID=Hash(Key)，将 metaID 发送给标签；标签将 metaID 存储下来，进入锁定状态。阅读器将(metaID,Key,ID)存储到后台数据库中，并以 metaID 为索引。

解锁标签：阅读器询问标签时，标签回答 metaID；阅读器查询后台数据库，找到对应的(metaID,Key,ID)记录，然后将该 Key 值发送给标签；标签收到 Key 值后，计算 Hash(Key)值，并与自身存储的 metaID 值比较，若 Hash(Key)=metaID，标签将其 ID 发送给阅读器，这时标签进入已解锁状态，并为附近的阅读器开放所有的功能，如图 2 所示。

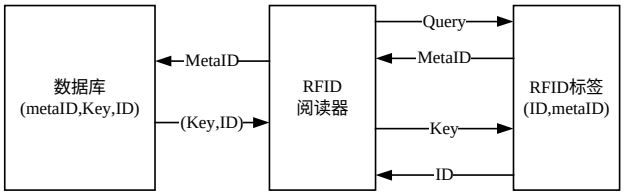

图 2 解锁经“ Hash锁 ”锁定的标签[3]

方法的优点：解密单向 Hash 函数是较困难的，因此该方法可以阻止未授权的阅读器读取标签信息数据，在一定程度上为标签提供隐私保护；该方法只需在标签上实现一个 Hash 函数的计算，以及增加存储 metaID 值，因此在低成本的标签上容易实现。

方法的缺陷[1]：由于每次询问时标签回答的数据是特定的，因此其不能防止位置跟踪攻击；阅读器和标签间传输的数据未经加密，窃听者可以轻易地获得标签Key和ID值。

2.2.2 随机Hash锁(Random Hash Lock)[2]

为了解决 Hash 锁中位置跟踪的问题，将 Hash 锁方法加以改进，采用随机 Hash 锁方法。

首先介绍字符串连接符号“ || ”，如标签ID和随机数R的连接即表示为“ ID||R ”。该方法中数据库存储各个标签的ID值，设为ID_1、ID_2…ID_k…ID_n。

锁定标签：通过向未锁定的标签发送简单的锁定指令，即可锁定该标签。

解锁标签：阅读器向标签ID发出询问，标签产生一随机数R，计算Hash(ID||R)，并将(R,Hash(ID||R))数据对传送给阅读器；阅读器收到数据对后，从后台数据库中取到所有的标签ID值，分别计算各个Hash(ID||R)值，并与收到的Hash(ID||R)比较，若$Hash(ID_k||R)=Hash(ID||R)$，则向标签发送$ID_k$；若标签接收到的$ID_k=ID$，此时标签即被解锁，如图 3 所示。

在该方法中，标签每次回答是随机的，因此可以防止依据特定输出而进行的位置跟踪攻击。但是，该方法也有一定的缺陷：(1)阅读器需要搜索所有标签ID，并为每一个标签计算$Hash(ID_k||R)$，因此标签数目很多时，系统延时会很长，效率并不高；(2)随机Hash锁不具备前向安全性，若敌人获得了标签ID值，则可根据R值计算出Hash(ID||R)值，因此可追踪到标签历史位置信息。

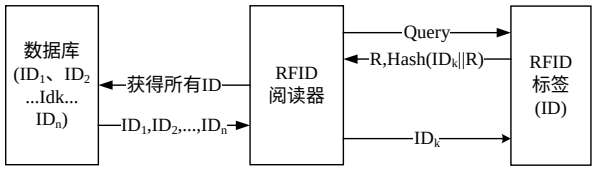

图 3 解锁经“ 随机Hash锁 ”锁定的标签[3]

2.2.3 Hash链(Hash Chain) [4]

NTT 实验室提出了一个 Hash 链方法，其保证了前向安全性，工作机制如下：

锁定标签：对于标签ID，阅读器随机选取一个数S_1发送给标签，并将(ID,S_1)存储到后台数据库中，标签存储接收到S_1后，进入锁定状态。

解锁标签：在第i次事务交换中，阅读器向标签发出询问消息，标签回答$a_i=G(S_i)$，并更新$S_{i+1}=H(S_i)$，其中G和H为单向Hash函数，如图 4 所示。

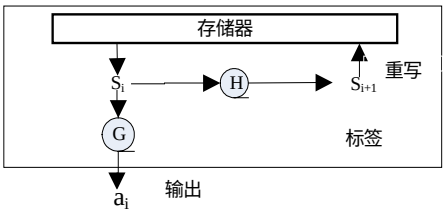

图 4 “ Hash 链 ”方法

阅读器接收到a_i后，搜索数据库中所有的(ID,S_1)数据对，并为每个标签计算$a_i^*=G(H^i(S_1))$，比较a_i^*是否等于a_i，若相等，则返回相应ID。

方法优点[1]：具有不可分辨性，因为G是单向Hash函数，外人获得a_i值不能推算出S_i值，当外人观察标签输出时，G输出的是随机数，所以不能将a_i和a_{i+1}联系起来；具有前向安全性，因为H是单向Hash函数，即使窃取了S_{i+1}值，也无法推算出S_i值，所以无法获得标签历史活动信息。

方法缺点[1]：需要为每一个标签计算$a_i^*=G(H^i(S_1))$，假设数据库中存储的标签个数为N，则需进行N个记录搜索，2N个Hash函数计算，N次比较，计算和比较量较大，不适合标签数目较多的情况。

3 Key 值更新随机 Hash 锁

鉴于上述几种安全隐私保护方法存在的缺陷，并结合几种方法的思想，本文提出了一种“ Key 值更新随机 Hash 锁 ”方法，实现了安全高效的读取访问控制。

3.1 工作原理

数据库记录主要包括 4 列：H(Key)，ID，Key，Pointer，主键为H(Key)。其中ID为标签唯一标志号，Key是阅读器为每个标签选取的随机关键字，H(Key)是Key的单向Hash函数H计算值，Pointer是数据记录关联指针，主要用来保证数据的一致性[5]。

下面详细阐述该方法的基本工作原理：

(1)锁定标签

对于标签 ID，首先阅读器随机选取一个数作为该标签的 Key，将 Key 值发送给该标签，并建立标签在数据库中的初始记录(H(Key),ID,Key,0))，标签将接收到的 Key 值存储下来

后，进入锁定状态。

(2)解锁标签

1)数据库首先产生一个随机数 R，传送给阅读器，然后阅读器将询问消息 Query 和 R 都发送给标签；

2)标签根据接收到的 R 和自身 Key 值，计算 H(Key)和 H(Key||R)的值，然后将(H(Key),H(Key||R))数据对发送给阅读器，接着自行计算 H(ID||R)和 Key*=S(key)，但此时 Key 值并不更新。

3)阅读器查找数据库中的记录，若找到记录 i：($H(Key_i),ID_k,Key_i,Pointer_i$)，其中$H(Key_i)=H(Key)$，则数据库计算$H(Key_i||R)$，并比较$H(Key_i||R)$与接收到的H(Key||R)值是否相等。若不相等，则忽略此消息，表明标签是非法标签，在此阅读器完成对标签的合法性验证；若相等则继续下一步；

4)数据库计算$H(ID_k||R)$的值，并将ID_k和$H(ID_k||R)$的值都传送给阅读器。然后阅读器将$H(ID_k||R)$发送给标签；

5)数据库计算$Key^*_i=S(key_i)$和$H(Key^*_i)$的值。若$Pointer_i=0$，则在数据库中添加新的记录 j：($H(Key^*_i)$, ID_k,Key^*_i,i)，并将记录 i 修改成($H(Key_i),ID_k,Key_i,j$)；若$Pointer_i!=0$，则找到第$Pointer_i$条记录，将其修改成($H(Key^*_i),ID_k,Key^*_i,i$)；

6)在标签接收到$H(ID_k||R)$后，比较其与标签在第 2 步中计算的H(ID||R)是否相等，若相等，则将自身的Key值更新为 Key*，标签进入解锁状态，对阅读器开放其所有功能；若不相等，表明阅读器是非法阅读器，标签保持沉默，在此标签完成对阅读器的验证。如图 5 所示。

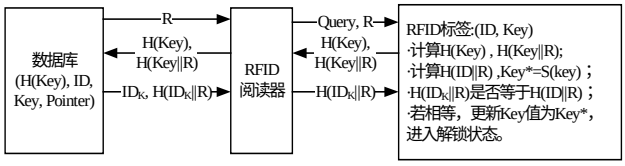

图 5 解锁经“Key 值更新随机 Hash 锁”锁定的标签

3.2 数值实验

设数据库初始时存储了两个标签，ID 分别为 1、2，随机选择的 Key 分别为 5、12，数据库初始化如表 1 所示。

表 1 数据库初始化数据建立

H(Key)	ID	Key	Pointer
H(5)	1	5	0
H(12)	2	12	0

设阅读器要询问ID为 1 的标签，首先阅读器向标签发送询问消息和随机数 3，标签向阅读器回答数据(H(5),H(5||3))，接着自行计算自身的H(ID||R)＝H(1||3)值和Key*＝S(5)；阅读器根据H(5)查找后台数据库，找到记录 1：(H(5),1,5,0)，数据库计算H(Key||R)=H(5||3)，与接收到的H(5||3)相等，至此验证了标签是合法的；接着数据库计算H(ID||R)=H(1||3)，并将(1,H(1||3))传送给阅读器，由此阅读器知道了该标签的ID为 1，然后阅读器将H(1||3)发送给标签；数据库计算$Key^*_1=S(5)$，由于$Pointer_1=0$，数据库中新建一条记录 3：(H(S(5)),1,S(5),1)，并将记录 1 修改为(H(5),1,5,3)。标签接收到数据H(1||3)后，比较发现其等于之前计算的H(ID||R)，于是将自身Key值更新为S(5)。此时数据库中的数据记录如表 2 所示。

下一次再与标签 1 通信时，数据库根据标签的 H(Key)＝H(S(5))查找到第 3 条记录，该记录的 Pointer 为 1，则第 2 次更新 Key 值的记录将会覆盖第 1 条记录。

表 2 通信一次后数据库中的数据

H(Key)	ID	Key	Pointer
H(5)	1	5	3
H(12)	2	12	0
H(S(5))	1	S(5)	1

当标签被询问过一次之后，数据库中始终保持了两条与该标签有关的数据记录，这主要是为了保证数据的一致性。假设在这次通信中阅读器发送的数据 H(1||3)并未成功地被标签接收到，则标签 1 的 Key 值将不会更新，此时数据库的第 3 条记录是错误的。那么在下次与标签 1 的通信中，查找到的仍是记录 1，数据库根据记录 1 的 Pointer 值为 3，将修改第 3 条记录，如此就保证了数据的一致性。

3.3 性能分析与方法特点

(1)简单实用。将随机数产生器等复杂的计算移到了后台数据库中实现，降低了标签的复杂性，标签只需要实现两个 Hash 函数 H 和 S，这在低成本的标签上较易实现。

(2)前向安全。因为标签的 Key 值在每次事务交换后被单向 Hash 函数 S 更新，外人即使获取了当前标签 Key*值，也无法推算出之前的 Key 值，所以无法获得标签相关的历史活动信息。

(3)机器运算负载小，效率高。在每次询问过程中，设数据库中存储的标签个数为 N，本方法中后台数据库需执行 2N 个记录搜索(因每个标签存在两条记录)，进行 3 个 Hash 函数 H(Key||R)、S(Key)、H(ID||R)计算和 1 次值比较，以及产生 1 个随机数 R。相比于 Hash 链方法需计算 2N 个 Hash 函数、N 个记录搜索和 N 个值比较，因为 Hash 函数的计算时延较长，资源消耗大，所以当 N 很大时，本方法系统的负载将要小得多，速度较快，延时较短，效率较高，但安全性更高。

(4)适应标签数目较多的情况。随着标签数目的增加，计算机搜索与计算所需要的时间缓慢增加，可适应标签数目较多的情况。

(5)实现了身份的双向验证。通过 Hash(Key||R)的计算比较，阅读器实现了对标签的验证；通过 Hash(ID||R)的计算比较，标签实现了对阅读器的验证。

(6)有效实现安全隐私保护。

1)防非法读取：只有经过合法认证的阅读器才可读取标签的数据信息；

2)防位置跟踪：由于随机数 R 和标签的 Key 值是更新变化的，因此每次回答的数据(H(Key)，H(Key||R))值也是不同的，可以防止外人根据特定输出而进行的跟踪定位；

3)防窃听：传输的 ID 值和 Key 都经过了 Hash 函数加密，外人很难解密得出 ID 和 Key 的值，因此有效地防止了窃听；

4)防伪装哄骗：由于外人无法获知 Key 值，因此无法模拟合法标签发送(H(Key), H(Key||R))数据，故有效地防止了伪装哄骗攻击；

5)防重放：每次产生的R值是随机的，外人即使窃听了合法阅读器前一次发送的$H(ID_k||R)$数据，也无法再次模拟出$H(ID_k||R)$值，有效地防止了重放攻击。

4 结束语

“Key 值更新随机 Hash 锁”方法具有成本低、负载小、效率高、安全性好等特点，且能保证前向安全性，基本上弥补了目前安全保护方法安全性不够和效率低等缺陷，是一种较为实用的算法。但此方法还存在一些不足，如尚无法防止敌人根据流量分析(计算标签的个数)而进行的定位跟踪，同时安全性提高也增加了标签部分计算时延，这些尚需进一步研究改进。

（下转第 159 页）