计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20150925
54(2):328-337,2017

一种基于时间约束的分层访问控制方案

马 骏[1,2] 郭渊博[2] 马建峰[1] 张 琦[2]
[1](西安电子科技大学计算机学院 西安 710071)
[2](解放军信息工程大学 郑州 450001)
(sijunhan@163.com)

A Time-Bound Hierarchical Access Control Scheme for Ubiquitous Sensing Network

Ma Jun[1,2], Guo Yuanbo[2], Ma Jianfeng[1], and Zhang Qi[2]
[1](*School of Computer Science and Technology, Xidian University, Xi'an* 710071)
[2](*PLA Information Engineering University, Zhengzhou* 450001)

Abstract In order to realize an effective access control of sensitive data captured by sensor nodes, researchers have made great achievements on secure and efficient hierarchical access control to satisfy the features of widespread distribution, large universe, limited computation and storage capacity of sensor nodes in ubiquitous sensing network. However, time is the main factor that makes the requirements of hierarchical access control scheme in ubiquitous sensing network different from that in traditional Internet networks, leading to the limited actual application scenario. According to the users' requirement on the nodes for gathering resources, an efficient and secure time-bound hierarchical access control scheme is presented in this paper. Based on the characteristics of perception node in ubiquitous sensing network, including the limited power and computation capability, as well as the storage resource, the scheme optimizes the key storage of user, key derivation time, and public information. The advantages of our scheme include that 1) only one key material is required in each users'access; 2) the balance can be achieved between the time for key acquisition and the amount of public information and 3) the scheme is provably secure without random oracle model. Theoretical analysis indicates that our proposed schedule adapts to user' access control requirement of ubiquitous sensing network.

Key words time-bound; centroid of tree; hierarchical access control; ubiquitous sensing; key derivation

摘 要 提出一种时间约束条件下的分层访问控制方案.根据用户对感知节点资源的访问控制需求,充分考虑感知节点计算、存储能力受限且节点数海量的特点,从用户掌握密钥数、密钥获取时间和产生公共信息数3方面进行优化设计,以实现高效、安全的分层访问控制.与现有其他方案对比,该方案的优势在于:1)用户对大量感知节点资源进行的一次访问,仅需要掌握单个密钥材料;2)通过优化设计,使用户访问节点资源密钥的获取时间与产生的公共信息数达到最佳平衡;3)提出的方案是可证明安全的.

收稿日期:2015-10-19;**修回日期**:2016-04-11
基金项目:长江学者和创新团队发展计划基金项目(IRT1078);中央高校基本科研业务费专项资金项目(JY10000903001);国家自然科学基金项目(61602515);河南省科技攻关项目(2016170162);信息保障重点实验室开放课题(KJ-15-103)
This work was supported by the Program for Changjiang Scholars and Innovative Research Team in University (IRT1078), the Fundamental Research Funds for the Central Universities (JY10000903001), the National Natural Science Foundation of China (61602515), the Science and Technology Research Project of Henan Province (2016170162), and the Foundation of Science and Technology on Information Assurance Laboratory (KJ-15-103).

关键词　时间约束;树重心;分层访问控制;泛在感知;密钥获取

中图法分类号　TP309

随着移动互联、物联网等新兴技术的不断普及,金融、电子商务、医疗、军事后勤、军事情报等行业领域的产业结构和服务模式已逐步向无所不在的泛在感知(ubiquitous sensing)方向演进.泛在感知即是5A(anytime, anywhere, anything, anyone, anydevice)情形下的信息获取,现阶段主要利用RFID、蓝牙、红外、GPS、音/视频等芯片,以传感设备和智能移动设备为载体,采集温度、视频、声音、定位等各类数据,为用户提供相应的数据访问[1].在一些典型的泛在感知应用场景中,由传感设备和智能移动设备构成的感知节点数以亿计,且具有分布广、计算和存储能力受限、感知信息敏感等特点,使其为信息服务的泛在化应用带来巨大便利的同时,也对感知节点采集数据的安全高效访问提出了严峻挑战[2].

传统的分层访问控制方案[3],通过方案设计可以实现用户在掌握较少密钥的同时,既能对节点 v 保护的资源进行访问,又能通过密钥推导算法获取与节点 v 构成偏序关系的节点 w 的保护密钥,进而访问节点 w 保护的资源.而引入时间因素之后,如果用户想要在时刻 t_i 访问节点 v 及构成偏序关系的下层节点 w 的资源,则需要获取时刻 t_i 层次节点 v 的密钥材料 k_{t_i};如果想要访问时刻 t_j 层次节点 v 及构成偏序关系的下层节点 w 的资源,则必须另外获取时刻 t_j 层次节点 v 的密钥材料 k_{t_j}.依次类推,如果用户想要访问 $T=[t_i,t_j]$一个时间段内 v 的资源,不得不获取 $t_i\leqslant t\leqslant t_j$ 内每个时刻对应的密钥材料.随着 T 时间段的增大,用户一次需要掌握大量的密钥材料才能访问,显然无法满足泛在感知环境下任何时间能够访问感知节点资源的实际应用需求.如何在时间约束条件下,用户尽可能掌握较少的密钥材料并安全高效地访问更多的资源,是泛在感知环境资源访问必须要解决的实际问题,需要考虑适用于泛在感知环境下基于时间约束的分层访问控制方案[4].

在时间约束条件下的分层访问控制方案中,大多数方案都为了解决因时间分片带来极大的密钥计算和密钥存储消耗,往往忽视了分层访问控制下的多用户合谋攻击的发生[5-8];文献[9-10]基于Akl和Taylor提出方案的扩展,提出了时间约束条件下的安全方案构造,但未考虑方案的安全问题;文献[11]虽然给出了可证明安全的考虑时间约束条件下的分层访问控制方案,但其方案是基于RSA的构造,采用模数运算的计算量,无法适用泛在感知环境感知节点计算能力受限的情况;文献[12]提出了考虑时间约束条件下可证明安全的基于对称密钥算法的访问控制方案,具有计算量较少的特点,但该方案导致较多的密钥产生,会增大感知节点和用户的密钥存储负担,因此也不适用于泛在感知环境.

针对用户在时间约束条件下对泛在感知环境节点资源的分层访问控制需求,本文利用有向无环图的特性,设计了一种基于时间约束的分层访问控制方案,与现有方案相比较,本文的优势主要体现在:1)用户在一次访问过程中仅需要掌握单个密钥材料,通过简单计算即能访问相应级别在某一时刻的资源及该级别以下相应时刻的资源;2)在密钥获取时间与产生的公共信息之间达到最佳平衡;3) 提出的方案在标准模型下的可证明安全的.

1　时间约束条件下的分层访问控制模型

设 $DAGG=(V,E,S)$是一个有向无环图,其中 $V=\{v_1,v_2,\cdots,v_n\}$,表示 G 的节点集合,每个节点 v_i 代表一个层次节点; $E=\{e_1,e_2,\cdots,e_k\}$,表示 G 的有向边的集合,每条有向边 e_i 连接的2个层次节点之间具有覆盖关系; $S=\{s_1,s_2,\cdots,s_j\}$,表示当前级别的层次节点包含数据资源的集合,可以用函数 $\xi:V\to 2^S$表示,并且存在 $\forall v_i,v_j\in V$,如果 v_i 和 v_j 不构成偏序或覆盖关系,则 $\xi(v_i)\bigcap\xi(v_j)=\varnothing$.设 $T=\{t_1,t_2,\cdots,t_m\}$是一个时间序列,λ 表示时间段,是 T 的一个子集,记作 $\lambda\subset T$.由多个 λ 构成的集合,表示时间段的集合,记作 $\zeta=\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$.一个时间约束条件下的分层访问控制模型描述如下:

给定 $DAGG=(V,E,S)$, $T=\{t_1,t_2,\cdots,t_m\}$, $\zeta=\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$和安全参数$\{0,1\}^\kappa$,在多项式时间内,对 $\forall v\in V$ 初始化产生密钥材料 $k_{v,\lambda}$、保护密钥 $sk_{v,t}$及公共信息,其中 $v\in V,\lambda\in\zeta,t\in\lambda$.利用 $k_{v,\lambda}$和公共信息,通过计算可得到 λ 内的任意 $sk_{v,t}$,同时利用层次节点之间的偏序关系,可以在多项式时间内计算得到 $k_{w,\lambda}$和 $sk_{w,t}$,其中 $v\leqslant w,w\in V$.

在该模型下,用户对泛在感知环境中资源的访问过程为:当用户想要访问时间段 λ 层次节点 v 的资源,首先需要通过安全通路从密钥生成中心TA

获得合法的密钥材料 $k_{v,\lambda}$，然后结合公开信息计算得到时间 t 的资源保护密钥 $sk_{v,t}$. 接着利用公共边信息 E 得到与 v 构成偏序关系的层次节点集合，即可通过密钥推导算法得到层次节点 w 在时间段 λ 的密钥材料，进而结合公共信息计算得到时间 t 的资源保护密钥 $sk_{w,t}$. 只要存在 $e\in E$ 满足从授权层次节点开始到其他层次节点的有向路径，均可通过公开的 e 利用重复的密钥推导算法获得下层密钥，从而使该用户的一次访问过程中仅掌握单个密钥材料 $k_{v,\lambda}$ 情况下，能够访问更多的节点资源，从而增加分层访问控制的实用性.

通过前文分析，我们可以感受到在加入时间因素的分层访问控制中，用户获取层次节点密钥访问节点资源的过程，相比较不考虑时间因素的情况，大大增加了密钥存储负担和密钥获取时间. 因此，在设计时间约束条件下的分层访问控制方案时，要统一考虑以下 3 个实际需求：

1) 用户在一次访问过程尽可能掌握较少的密钥数；

2) 具有较小的密钥获取时间；

3) 产生较少的公共信息.

从用户的访问角度考虑，第 2 节设计的方案中是在首要满足需求 1 的情况下，通过对方案的设计在需求 2 和需求 3 之间达到一定的平衡.

2 基本方案构造

Santis 等人基于对称密钥学在文献[13]中，提出一种建立 2 层加密构造分层访问控制方案(记作 TLEBC). 该方案需要将上层节点的 λ 与本层及下层的 t 建立对应关系. 随着层次的增加，其建立的有向公共边数会呈几何级数的增长. 针对 TLEBC 方案存在的问题，本文首先提出基于 2 类偏序关系的方案构造(记作 TLPOS)，其构造思想为：上层节点 v_λ 仅与下层节点 w_λ 建立偏序关系，而 λ 和 t 仅在同层次之间建立偏序关系，从而形成 2 类偏序关系.

给定 $G=(G_1,G_2)$ 表示时间约束条件下的分层访问控制模型，其中 G_1 表示由层次节点 $V=\{v_1,v_2,\cdots,v_n\}$ 的级联关系构成的有向图，G_1 中的节点表示层次节点，有向边表示层次节点之间形成的偏序关系；G_2 表示由 $\zeta=\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$ 的包含关系形成的有向图，G_2 中节点表示 λ_i，有向边表示 λ_i 之间形成的偏序关系. 通过将 G 按 2 类偏序关系进行分解，TLPOS 方案的构造过程如下：

1) 对于每一个 $v\in V$，每一个 $\lambda\in\zeta$，给出新的层次节点 v_λ，其中 $|\lambda|>1$；

2) $v\in V$，每一个 $t\in T$，给出新的层次节点 v_t；

3) 对于每一个 $v\in V$，每一个 $\lambda\in\zeta$，每一个 $t\in T$，给出新的有向边 $e_{\lambda t}=\langle v_\lambda,v_t\rangle$；

4) 对于 $v,w\in V$，每一个 $\lambda\in\zeta$，给出新的有向边 $e_{vw}=\langle v_\lambda,w_\lambda\rangle$.

其中，我们只在 $|\lambda|>1$ 的情形下给出新的层次节点；当 $|\lambda|=1$ 时，即 $\lambda=\{t_i\}$，此种情况与 $t\in T$ 中的情况相同，可认为是同一层次节点.

与文献[13]给出的示例保持一致，假设 $\zeta=\{\lambda_1,\lambda_2,\lambda_3\}$，其中 $\lambda_1=\{t_1,t_2\}$，$\lambda_2=\{t_1,t_2,t_3\}$，$\lambda_3=\{t_2,t_3\}$. 如果存在 $v,w\in V$，且 $w\prec\cdot v$，则按照 TLPOS 构造过程形成的有向图如图 1 所示：

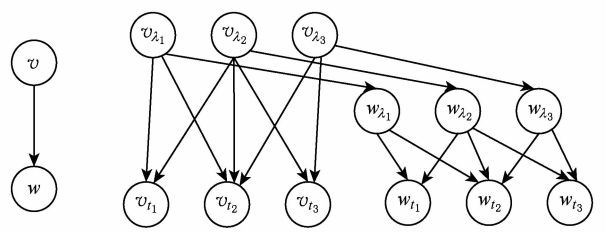


Fig. 1 Directed graph based on two kinds of partial order relation

图 1 基于 2 类偏序关系的有向图

通过图 1 示例可以看出，在加入时间条件后，2 个层次节点之间形成的偏序关系，会随着时间集的变化构成多个层次节点和多个偏序关系. 为了能够获取下层节点在某一时刻的密钥，需要做以下初始化构造：

步骤 1. 对于每个层次节点 v，随机选取密钥材料 $k_{v,\lambda}\in\{0,1\}^\kappa$，利用 $k_{v,\lambda}$ 和散列函数 f 计算得到保护密钥 $sk_{v,\lambda}$ 和推导密钥 $dk_{v,\lambda}$；

步骤 2. 对于每个层次节点 v，随机选取密钥 $k_{v,t}\in\{0,1\}^\kappa$，利用对称密钥加密方法 ε 的加密算法 E，将保护密钥 $sk_{v,\lambda}$ 和 $k_{v,t}$ 建立映射关系，作为公共边信息 $e_{\lambda t}$；

步骤 3. 对于每个层次节点 $v,w,w\prec\cdot v$ 存在，随机选取密钥材料 $k_{w,\lambda}\in\{0,1\}^\kappa$，利用对称密钥加密方法 ε 的加密算法 E' 将推导密钥 $dk_{v,\lambda}$ 和 $k_{w,\lambda}$ 建立映射关系，作为公共边信息 e_{vw}.

通过以上初始化构造，将公共信息发布到感知层网络，私有信息由层次节点存储.

在该构造方案下，用户对感知环境资源进行访问的过程：首先通过 TA 获取层次节点 v 在时间段 λ 内的密钥材料 $k_{v,\lambda}$，计算得到保护密钥 $sk_{v,\lambda}$ 和推导

密钥 $dk_{v,\lambda}$;利用保护密钥 $sk_{v,\lambda}$ 和公共信息 $e_{\lambda t}$ 通过解密算法 D(与加密算法 E 对应),计算得到 $k_{v,t}$;同时,利用推导密钥 $dk_{v,\lambda}$ 和公共信息 e_{vw} 通过解密算法 D'(与加密算法 E' 对应),计算得到 $k_{w,\lambda}$;然后利用 $k_{w,\lambda}$,计算得到层次节点 w 在时间段 λ 内的保护密钥 $sk_{w,\lambda}$ 和推导密钥 $dk_{w,\lambda}$,并利用保护密钥 $sk_{w,\lambda}$ 按照相同步骤计算得到保护密钥 $k_{w,t}$. 需要说明的是,考虑到通用性,这里我们并未指出具体的密钥推导算法,在实际应用中为了防止同谋攻击的发生,可参考本文作者在文献[14]提出的密钥推导算法,或其他具有同样安全保障的密钥推导算法.

通过上述步骤,我们可以看到,用户在某一时间段 λ,仅需要掌握单个密钥材料 $k_{v,\lambda}$,通过简单的单向散列计算和对称加密运算即能得到当前层次或下层节点对应某一时刻的保护密钥,进而访问相应资源. 与文献[13]中的 TLEBC 方案相比,TLPOS 方案在同样满足需求 1 和需求 2 的同时,能够更好地满足需求 3.

3　基于树重心分解的分层访问控制方案

在第 2 节提出的 TLPOS 方案中,我们可以看到在时间约束条件下的分层访问控制本质上是分成 2 部分进行构造:一部分是实际存在层次节点之间形成的分层访问控制关系(空间维度形成的层次关系),另一部分是由多个时间段与访问时刻形成的层次关系(时间维度形成的层次关系). 而在这 2 部分中,第 2 部分的分层访问控制会因时间的增长($|T|\to n$)和时间段的增多($|\lambda|\to n$),使初始化构造时不得不产生大量的公共信息,以满足用户获取时间 t 的密钥,进而能够访问对应时刻的资源. 虽然 TLPOS 方案相比文献[13]中的 TLEBC 方案已经明显减少公共信息的产生,但产生的公共信息量仍给系统造成负担. 如何减少大量公共信息的产生是需要进一步解决的问题.

3.1　基于 λ 分层的方案构造

给定 $\zeta=\{\lambda_1,\lambda_2,\lambda_3\}$ 和 $T=\{t_1,t_2,\cdots,t_m\}$,在 TLPOS 方案构造中,是将每个 $\lambda(\lambda\in\zeta)$ 与对应的 t_j ($t_j\in T$)直接利用公共边建立对应关系,如图 1 中的 v_λ 到 v_t 的有向边. 这种构造方式虽然对减小密钥获取时间有利(时间维度仅一次推导过程),但会产生大量的公共边信息.

如果利用集合 λ 之间具备的包含关系,首先将 ζ 中的 λ 建立层次关系,然后将部分 λ 与之包含的 t 建立层次关系,无疑会大大缩短 λ 与 t 之间的公共边数. $\zeta=\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$ 中的 λ 之间天然地具备包含与被包含的关系,从而可以利用该特征建立 λ 之间的偏序关系,形成有向无环图. 基于该构造思想,本文在 TLPOS 方案的基础上,为减少初始化构造中产生的公共信息量,将提出基于树重心分解的方案构造(记作 TCDS).

举例来讲,设 $T=\{t_1,t_2,t_3,t_4\}$,$|T|=4$,ζ 是由 T 构成的全集:

$$\lambda_1=\{t_1\},\lambda_2=\{t_2\},\lambda_3=\{t_3\},\lambda_4=\{t_4\},$$
$$\lambda_5=\{t_1,t_2\},\lambda_6=\{t_2,t_3\},\lambda_7=\{t_3,t_4\},$$
$$\lambda_8=\{t_1,t_2,t_3\},\lambda_9=\{t_2,t_3,t_4\},$$
$$\lambda_{10}=\{t_1,t_2,t_3,t_4\}.$$

其中 $|\zeta|=10$. 利用 $\lambda_i(i=1,2,\cdots,10)$ 之间的包含关系,可建立如图 2 的有向无环图. 为表述更加直观,我们将 λ_i 按时间区间方式表示,如 λ_1 表示[1,1],λ_6 表示[2,3],λ_9 表示[2,4]等.

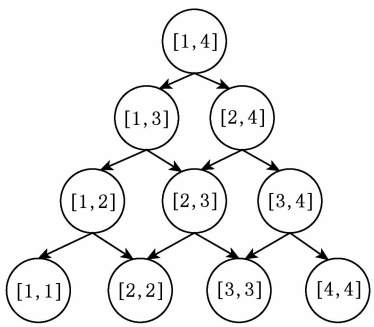


Fig. 2　DAG composed by λ_i

图 2　λ_i 构成的有向无环图

通过图 2 我们可以看到,利用 λ_i 形成的偏序关系,当 $|T|=4$ 时,λ 与 t 之间建立的公共边仅为 6(将 $|\lambda|=1$ 的节点作为时间节点),与 TLPOS 方案构造中同一层次节点内 λ 与 t 之间建立的公共边数(总计为 16)相比,公共边数得到大幅下降. 即使将 λ 之间的公共边数计算在内,总的公共边数也小于基本方案构造建立的公共边总数,且随着 $|T|$ 值的增大,公共边数也会随之大幅减少.

3.2　基于树重心分解的优化

上述构造方案虽使公共边数得以减少,但是当用户通过掌握的 λ 对应密钥,推导得到时刻 t 密钥时,不得不需要经过由 λ 与 t ($|\lambda|=1$)建立的多条公共边信息才能计算得到相应密钥,这无疑会增加用户的密钥获取时间(不满足具有较小密钥获取时间的需求),且随着 $|T|$ 值的增大,λ 形成的有向图深度增加,用户的密钥获取时间也会随之增加,从而增大用户获取密钥时间的负担.

为此,我们需要通过方案设计优化用户的密钥获取过程,即缩短用户的密钥获取时间.这里我们引入树重心的概念和特性.

1) 树重心的定义.对于树 T 中的任一节点 k,若将它删去(连同边)后,树 T 会变为若干棵子树 $\{T_1,T_2,\cdots,T_n\}$,取这些子树的节点数中最多的作为节点 k 的值.如果对应树 T 中所有节点中对应的 k 值最小,则称 k 为树 T 的重心,记作 c[15].

2) 树重心分解的特性.设 T 的节点数为 n,将以重心 c 为根的树从 T 中删除,T 中剩余部分的节点数小于 $n/2$[16].

通过对树重心的逐次分解,可将每次分解确定的树重心依序两两通过有向边进行连接,进而能够缩短整个树从树根到叶子结点的距离(减小树的深度).将重心之间连接的有向边作为新添加的公共边信息,则用户可通过新增的公共边信息进行密钥推导过程,从而减少用户的密钥获取时间.

我们设计基于树重心分解的构造过程如下:

步骤 1. 计算树 T 的重心 c[17],并以树 T 的根 r 作为起点,重心 c 为终点,建立一条有向边作为新添加的公共边,如果 $c\prec\cdot r$ 已经存在,则不需要添加新的有向边.

步骤 2.1. 以 c 为根建立的树 T_1 中,计算重心 c_1,并添加一条以 c 为起点、c_1 为终点的有向公共边,如果 $c_1\prec\cdot c$ 存在,则不需要添加新的有向边.

步骤 2.2. 树 T 除去树 T_1 的部分构成新树 T_2,计算 T_2 的重心 c_2,并建立一条从 r 到 c_2 的有向边,如果 $c_2\prec\cdot r$ 存在,则不需要添加新的有向边.

步骤 3.1. 以 c_1 为根建立的新树 T_3,分别重复步骤 2.1、步骤 2.2 的过程建立新的公共边信息.

步骤 3.2. 树 T_1 除去树 T_3 的部分构成的新树 T_4,分别重复步骤 2.1、步骤 2.2 的过程建立新的公共边信息.

步骤 4. 对整个树 T 重复以上的步骤,直至构成新树 T'的根 r'与 T'的重心 c'满足 $r'=c'$,则树初始化过程结束.

以上构造过程,每添加一条公共边均需要进行公共边增加的操作.增加有向公共边的操作过程,可参见本文作者在文献[14]提出的操作步骤,能够使新增公共边信息的过程无需更改原层次节点的私有信息,只需要调用原层次节点的私有信息计算即可生成新的公共边信息.

通过分析我们可以得到,上述添加新公共边的过程,本质上是重复利用树重心向量的分解来进行初始化构造.而每次基于树重心向量的分解剩余部分的层次节点数均小于原树层次节点数的一半.这样,以树 T 的根节点 r 和所有计算出的重心节点集合 $\{c_1,c_2,\cdots,c_i\}(i<n)$ 作为新的特殊层次节点,并通过新添加的公共边进行连接,可构成如图 3 所示的完全 2 叉树.

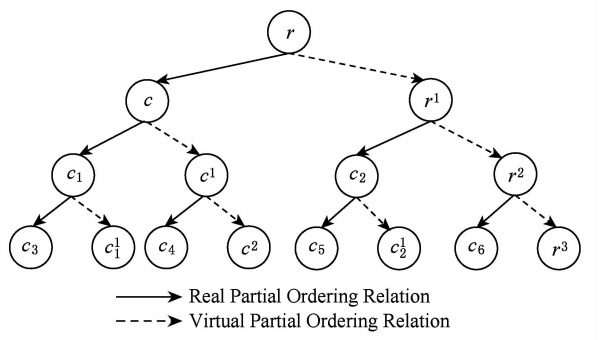


Fig. 3 Binary tree based on the decomposition of the centroid of tree

图 3 基于重心分解的 2 叉树

图 3 中父节点与左儿子节点之间的有向边是真正添加的公共边,表示的是重心节点之间的偏序关系;父节点与右儿子之间的连接用虚线表示,表示的是树 T_i 在去除重心 c_i 之前的根节点与去除重心 c_i 之后的根节点之间的连线(即虚线 2 端为同一层次节点,如 $r=r^1=r^2=r^3$).

3.3 基于树重心分解的方案构造

由于 3.1 节 λ_i 构成的层次关系并不符合树的特征(层次节点存在一个以上的父节点),因此,如果想要利用 3.2 节的优化方案,首先需要将有向无环图转换成有向树.

仍以 $T=\{t_1,t_2,t_3,t_4\}$,$|T|=4$ 为例,我们将图 2 转换成有向树如图 4 所示.

自此,我们可以基于树重心的特性,利用 3.2 节的树重心分解优化,展开基于树重心分解的构造.在现有节点中挑选特殊的节点(利用树重心分解算法找树的重心),依次添加有向边建立连接关系,从而利用特殊节点建立一条有向路径.

如果 $|T|=n$,通过简单计算可得,由 λ_i 形成的有向树节点数为 $4n-1$,公共边数为 $4n-2$.则由 $4n-1$ 个节点和 $4n-2$ 条有向公共边建立的有向 2 叉树中,利用树重心分解算法,需要添加的公共边数为 $\lceil \mathrm{lb}(4n-1)\rceil$,节点 λ_i 到节点 t 的最长距离为 $\lceil \mathrm{lblb}(4n-1)\rceil$.如果 $n=4$,则仅需添加 4 条公共边,λ_i 最远经过 2 跳距离即可到达 t,即一次访问过

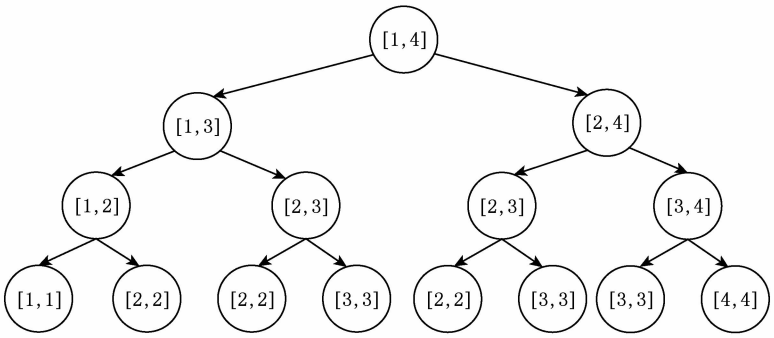


Fig. 4　The direct tree composed by λ_i

图 4　λ_i 构成的有向树

程中,用户通过 2 次密钥推导过程可获得 t 的密钥,进而访问到该时刻保护的资源信息.

通过以上分析,我们建立基于树重心分解的方案构造.

给定 $G=(G_1,G_2)$表示时间约束条件下的分层访问控制模型,其中 G_1 表示由层次节点 $V=\{v_1,v_2,\cdots,v_n\}$的级联关系构成的有向图,图中节点表示层次节点,图中的有向边表示层次节点之间形成的偏序关系;G_2 表示由 $\zeta=\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$的包含关系形成的有向树,树中节点表示 λ_i,树中的有向边表示 λ_i 之间形成的偏序关系. 通过将 G 按 2 类偏序关系进行分解,整个方案的构造过程如下:

1) 对于每一个 $v\in V$,每一个 $\lambda\in\zeta$,给出新的层次节点 v_λ,其中$|\lambda|>1$;

2) 对于每一个 $v\in V$,每一个 $t\in T$,给出新的层次节点 v_t;

3) 对于每一个 $v\in V,\lambda_i,\lambda_j\in\zeta,\lambda_j\prec\cdot\lambda_i$,给出新的有向边 $e_{\lambda_i\lambda_j}=\langle v_{\lambda_i},v_{\lambda_j}\rangle$;

4) 对于每一个 $v\in V,\lambda c_i,\lambda c_j\in\zeta$(多次树重心分析指定的特殊节点),$\lambda c_j\prec\cdot\lambda c_i$,给出新的有向边 $e_{\lambda c_i\lambda c_j}=\langle v_{\lambda c_i},v_{\lambda c_j}\rangle$,其中$|\lambda|>1$;

5) 对于每一个 $v\in V$,每一个 $\lambda\in\zeta$ 且$|\lambda|=2$,每一个 $t\in T$,给出新的有向边 $e_{\lambda t}=\langle v_\lambda,v_t\rangle$;

6) 对于 $v,w\in V$,每一个 $\lambda\in\zeta$,给出新的有向边 $e_{vw}=\langle v_\lambda,w_\lambda\rangle$.

其中,我们只在$|\lambda|>1$ 的情形下给出新的层次节点;当$|\lambda|=1$ 时,即 $\lambda=\{t_i\}$,此种情况,与 $t\in T$ 中的情况相同,我们认为其可表示为同一层次节点.

3.4　优化后的分层访问控制方案

分层访问控制方案的初始化过程通过以下步骤进行构造:

步骤 1. 对于每个层次节点 v,对于每个时间层次节点 $\lambda(|\lambda|>1)$,随机选取密钥材料 $k_{v,\lambda}\in\{0,1\}^\kappa$ 与之对应,并利用 $k_{v,\lambda}$ 和散列函数 f 计算得到保护密钥 $sk_{v,\lambda}$ 和推导密钥 $dk_{v,\lambda}$.

步骤 2. 对于每个层次节点 v,对于构成覆盖关系的层次节点 $\lambda_i,\lambda_j(\lambda_j\prec\cdot\lambda_i)$,利用对称密钥加密方法 ε 的加密算法 E,将保护密钥 sk_{v,λ_i} 和 sk_{v,λ_j} 建立映射关系,作为公共边信息 $e_{\lambda_i\lambda_j}$.

步骤 3. 对于每个层次节点 v,随机选取密钥 $k_{v,t}\in\{0,1\}^\kappa$,利用对称密钥加密方法 ε 的加密算法 E,将保护密钥 $sk_{v,\lambda}$ 和 $k_{v,t}$ 建立映射关系,其中$|\lambda|=2$,作为公共边信息 $e_{\lambda t}$.

步骤 4. 对于每个层次节点 v,w,$v\prec\cdot w$ 存在,随机选取密钥材料 $k_{w,\lambda}\in\{0,1\}^\kappa$,$|\lambda|>1$,利用对称密钥加密方法 ε 的加密算法 E'将推导密钥 $dk_{u,\lambda}$ 和 $k_{w,\lambda}$ 建立映射关系,作为公共边信息 e_{vw}.

经过以上初始化构造,将公共信息发布到感知层网络,私有信息由层次节点存储.

在该构造方案下,当用户想要访问感知层节点资源,首先通过 TA 获取层次节点 v 在时间段 λ 内的密钥材料 sk_{v,λ_i},通过计算得到保护密钥 sk_{v,λ_i} 和推导密钥 dk_{v,λ_i}. 利用保护密钥 sk_{v,λ_i} 获取 $k_{v,t}$ 的过程按下列 3 种情形进行:

情形 1. 如果$|\lambda_i|=2$,则利用 $e_{\lambda_{it}}\subseteq e_{\lambda t}$,通过解密算法 D(与加密算法 E 对应),计算得到 $k_{v,t}$;

情形 2. 如果 $\lambda_i\in\lambda_c$,则利用 $e_{c_jc_j}$,通过解密算法 D(与加密算法 E 对应),计算得到 sk_{v,λ_j},其中$|\lambda_j|=2$;然后继续进行情形 1 的操作,最终得到 $k_{v,t}$;

情形 3. 如果 $\lambda_i\notin\lambda_c$,则按以下步骤进行操作.

步骤 1. 找到特殊层次节点 $c_i,c_i\in\{c_1,c_2,\cdots,c_n\}$,满足 $v\leqslant c_i$,且不存在 c_k,使得 $v\leqslant c_k\leqslant c_i$ 存在;然后利用解密算法 D(与加密算法 E 对应)推导得到 c_i 对应密钥.

步骤 2. 找到特殊层次节点 $c_j,c_j\in\{c_1,c_2,\cdots,c_n\}$,满足 $c_j\leqslant w$,且不存在 $c_{k'}$,使得 $c_i\leqslant c_{k'}\leqslant w$ 存在;

然后利用解密算法 D（与加密算法 E 对应），由 c_i 的私有信息和$\langle c_i, c_j\rangle$的公共边信息，至多经过 $\text{lblb}\,n$ 轮推导得到 c_j 的密钥. 进而利用 c_j 的密钥和$\langle c_j, w\rangle$的公共边信息，采用解密算法 D(与加密算法 E 对应)最终得到 w 的密钥信息. 由于以 c_i 为根构成的树中 $c_i \in \{c_1, c_2, \cdots, c_n\}$，2 个相邻的特殊层次节点将树中包含的层次节点保持在常量级，因此，$\langle w, c_i\rangle$和$\langle c_j, w\rangle$的推导过程经过常数级的推导轮数，设$\langle w, c_i\rangle$的推导轮数为 a_1，$\langle c_j, w\rangle$的推导轮数为 a_2，则整个推导过程的密钥推导轮数至多为 $a_1 + a_2 + \text{lblb}\,n$.

4 方案的性能分析

为了便于和其他相关文献进行比较，我们假设一个时间约束条件下的分层访问控制方案 $G=(G_1, G_2)$，其中 G_1 包含的节点数 $|V|=m$，G_2 包含的时刻数 $|T|=n$，且 G_1 为有向树(该假设的目的是便于统计公共边数，如果 $|V|=m$，则 $|E|=m-1$). 我们将从 3 个方面比较：1)用户的一次访问过程要尽可能地考虑掌握较少的密钥；2)具有较小的密钥获取时间；3)产生较少的公共信息，与其他相关方案进行比较.

对于本文提出的基于构造方案 TLPOS，一次用户访问过程仅需要掌握单个密钥(材料)，即能进行访问；而获取访问资源密钥时间共需要经过 2 步操作；对于产生的公共信息，每个层次节点包括公共边数为 $\sum_{i=1}^{n} i\times(n-i+1)$，$m$ 个层次节点包含公共边数为 $m\sum_{i=1}^{n} i\times(n-i+1)$，$m$ 个层次节点之间包含公共边数为 $(m-1)\sum_{i=1}^{n} i\times(n-i+1)$，总共产生的公共边数为 $m(m-1)\sum_{i=1}^{n} i\times(n-i+1)$，则 TLPOS 的公共边数达到 $O(m\times n^2)$.

对于本文提出的基于树重心分解的方案构造 TCDS，一次用户访问过程仅需要掌握单个密钥(材料)，即能进行访问；而获取访问资源密钥时间，在 $|T|=n$ 的情况下，一个层次节点经过树型构造后总的时间节点数为 $2(2n-1)$，则用户获取访问资源的密钥最多经过 $\text{lblb}\,2(2n-1)+1$ 次操作，达到 $O(\text{lblb}\,n)$级；对于产生的公共边，每个层次节点包括的公共边数为 $2(2n-1)+\text{lblb}\,2(2n-1)+1$，$m$ 个层次节点包含公共边数为 $m\times 2(2n-1)+\text{lb}\,2(2n-1)$，$m$ 个层次节点之间包含公共边数为 $m-1$，总共产生的公共边数为 $m\times(2(2n-1)+\text{lb}\,2(2n-1))+(m-1)$，则 TCDS 的公共边数达到 $O(m\times n+\text{lb}\,n)$.

本文最终设计的方案 TCDS 与现有文献设计方案进行比较，如表 1 所示：

Table 1 Comparison of Our Scheme with Previous Work

表 1 方案比较

Scheme	Private Key	Key Derivation	Public Message
TLEBC [13]	1	$O(\text{lb}\,n)$	$O(m\times n^3)$
Naive +EBC[18]	$O(n)$	$O(diam(G))$	$O(m\times n)$
ISPIT + IHBT +EBC[18]	1	$O(\text{lb}\,n)$	$O((m-1)\times n+mn^2\text{lb}\,n)$
(4, 2)CSBT+EBC[18]	4	$O(diam(G))$	$O((m-1)\times n+mn^2/\text{lb}\,n)$
TCDS	1	$O(\text{lblb}\,n)$	$O(m\times n+\text{lb}\,n)$

Note: $diam(G)=diam(G_1)+diam(G_2)$ represents the depth sum of graph G_1 and G_2.

通过比较可以看到，在综合考虑用户掌握私钥数、用户密钥获取时间和产生的公共信息数 3 方面的整体优化情况下，本文设计的方案与现有其他方案比均具有明显的优势.

5 方案的安全性证明

下面我们针对 TCDS 方案(TLPOS 方案是 TCDS 方案的特例)，利用标准模型进行安全性证明.

5.1 方案分析

本文提出的基于时间约束的分层访问控制方案在空间维度上，可看作是利用层次节点 v 和 w 之间构成的偏序关系 $w\leqslant v$，使用户获取上层节点 v 的密钥材料 k_v 通过计算推导出下层层次节点 w 的密钥材料 k_w；在时间维度上，则是利用了 λ 到 t 的包含与被包含关系，建立了时间维度上的偏序关系，即存在 $v_t\leqslant v_\lambda$. 利用 λ 和 t 形成的偏序关系，用户在掌握 $k_{v,\lambda}$ 的情况下，可通过设计安全的密钥推导算法获得

$k_{v,t}$,进而能够访问由 $k_{v,t}$ 保护的节点资源.

因此,TCDS 方案可利用偏序关系 $w \leqslant v$ 和偏序关系 $v_t \leqslant v_\lambda$,在空间和时间维度采用文献[14]提出的密钥推导算法或其他安全算法,建立适用于泛在感知环境的密钥推导过程,将方案的安全性规约到单向散列函数的不可逆性和对称密码选择明文攻击的安全性上. 如果存在敌手 A 攻陷方案的概率等价于存在敌手 A' 攻陷方案采用的一个多项式时间的计算复杂难题上,那我们认为该方案是可证明安全的.

5.2　敌手模型与系统目标

我们将敌手 A 的攻击能力分为 3 种类型.

A_0:通过质询挑战者,获取泛在感知网络 G 中所有公共信息 $Pub=\{ID_i \in ID, e \in E\}$. 能够攻陷方案的方式为成功猜测节点 v_{it} 的密钥 k'_{it},成功的概率记为 $Pr[k'_{it}=k_{it}]$.

A_1:通过质询挑战者获取 G 中公共信息 Pub、部分感知节点$\{v_{it}\}$的密钥材料. 攻陷方案的方式是能够成功恢复节点 v_{mt} 的密钥 k_{mt},其中 $v_{mt} \notin \{v_{jt}\}$,$v_{jt} \leqslant v_{it}$,成功的概率记为 $Pr[k'_{mt}=k_{mt}]$.

A_2:通过质询挑战者获取 G 中公共信息 Pub、部分节点$\{v_{it}\}$的密钥材料,并且可以有针对性的选择任一节点 $v_{jt} \in \{v_{it}\}$并质询挑战者取得相应的密钥,攻陷方案的方式是能够成功区分挑战者返回给敌手的密钥是对应该节点的真实密钥还是与密钥等长的一串随机数,概率记为 $Pr[k'_j=k_j]$.

由于敌手 A_0 攻陷方案掌握的信息均包含在 A_1 和 A_2 中,因此我们将敌手能力规约为敌手具有 A_1 密钥恢复(key recovery)和 A_2 密钥的不可区分(key indistinguishability)上. 本文设计的系统目标即是抗敌手的 key recovery 和保证方案中的 key indistinguishability. 如果 A_1 攻陷系统的概率 $\varepsilon_1=Pr[k'_{jt}=k_{jt}]$,$A_2$ 攻陷系统的概率 $\varepsilon_2=Pr[k'_{jt}=k_{jt}]$ 均是可忽略的,则我们认为方案是安全的,即敌手不能以不可忽略的概率攻陷系统设计的方案.

5.3　证明过程

为了便于描述,令 $\zeta(Setup, Derivation)$ 表示本文提出的 TCDS 方案,其中 $Setup$ 表示方案的初始化构造过程,$Derivation$ 表示密钥推导获取过程. 根据敌手类型的不同,分别提出命题并进行安全性证明.

命题 1. 设$\{f_v\}$是一个伪随机函数集合,E 是一个安全的加密方案,对于 $\forall$ DAG G,如果存在一个类型为 A_1 的敌手能以 ε 的概率攻陷 ζ,则存在敌手 A'_1 以 ε' 的概率攻陷 f_v 和 E.

证明. 定义一系列的 $Game_0, Game_1, \cdots, Game_h$,其中 $Game_0$ 是敌手真正发起的游戏. 每个游戏 $Game_i$ 对应任意的拓扑序列中节点 v_{it} 展开密钥恢复的操作. 我们通过 $Game_i$ 之间演变的不可区分性,从而证明敌手 A_1 进行 $Game_0$ 攻陷方案的概率是可忽略的.

1) $Game_0$:

构造过程. 挑战者 C 执行方案 ζ 的 $Setup$ 操作,即输入安全参数 1^κ,将输出结果中的公共信息交给 A'_1.

质询过程. A'_1 向 C 发起质询,质询任意节点在任意时间 v_{it} 对应的密钥材料,挑战者 C 通过$\{0,1\}^\kappa$计算生成对应密钥材料 sk_{it} 并返回给 A'_1.

分析过程. 敌手 A'_1 指定一个节点 v_{0t},v_{0t} 与质询阶段的任意 v_{it} 在空间和时间维度上不构成任何偏序关系,即不存在 $v_{0t} \leqslant v_{it}$. A'_1 通过猜测得到最接近 v_{0t} 真实密钥 k_{0t} 的密钥 k'_{0t},A'_1 赢得 $Game_0$ 优势定义为 $Adv_{A'_1,0}=\varepsilon_0=Pr[k'_{0t}=k_{0t}]$.

2) $Game_1$:

$Game_1$ 的过程与 $Game_0$ 基本相似,其唯一区别在于推导过程获取得到的 v_{1t} 所需要的公共参数 $e_{0t,1t}$ 是由真正随机数来替代,即 $e_{0t,1t} \approx E_{e'}(sk_{1t})$. 利用 $Game_1$ 和 $Game_0$ 之间的可区分性,能构造一个多项式时间算法,以不可忽略的优势攻陷伪随机函数 f_v 和对称加密方案 E. 因此有

$$|\varepsilon_1-\varepsilon_0| < negl(f_v)+negl(E) \tag{1}$$

存在.

3) 不失一般性,对 $Game_i(i=1,2,\cdots,h)$ 进行同样操作.

$Game_i$ 的过程与 $Game_{i-1}$ 相同,唯一的区别在于推导得到 v_{1t} 密钥材料需要的公共参数 $e_{(i-1)t,it}$ 是由另外一个随机产生的值来替代,即 $e' \approx R'(ID_0)$,$e_{(i-1)t,it} \approx E_{e'}(sk_{it})$,其中 $R' \leftarrow \{0,1\}^\kappa$. 利用 $Game_i$ 和 $Game_{i-1}$ 之间的可区分性,能用于构造一个多项式时间算法,以不可忽略的优势攻陷伪随机函数 f_v 和对称加密方案 E. 即

$$|\varepsilon_i-\varepsilon_{i-1}| < negl(f_v)+negl(E) \tag{2}$$

存在.

此外,对于 $Game_h$ 其在整个计算推导中,敌手 A' 始终无法通过质询得到真实的密钥材料,因此,敌手 A' 在 $Game_h$ 中能够成功猜测 k_h 的优势定义为 $Adv_{A'_1,h}$:

$$\varepsilon_h=Pr[k'_h=k_h]=1/2^\kappa. \tag{3}$$

将式(1)～(3)进行合并得到:$\varepsilon_0 < h(negl(f_v)+negl(E))+1/2^\kappa$. 证毕.

命题 2. 设$\{f_v\}$是一个伪随机函数集合,E是一个安全的加密方案,对于$\forall$DAG G,如果存在一个类型为A_2的敌手能以ε的概率攻陷ζ,则存在敌手A_2'以ε'的概率攻陷f_v和E.

证明. 仍然定义一系列的$Game_0, Game_1, \cdots$,其中$Game_0$是敌手真正发起的游戏.每个$Game_i$对应任意的拓扑序列中节点v_{it}展开密钥获取的操作,敌手的优势在于能够成功区分挑战者字节流是真实密钥还是等长随机串.我们通过$Game_i$之间的不可区分性,从而证明敌手A_2攻陷方案的概率是可忽略的.

1) $Game_0$:

构造过程. 挑战者C执行方案ζ的$Setup$操作,即输入安全参数1^κ,将输出结果中的公共信息交给A_2'.

质询过程. ①阶段1,A_2'向C发起质询,质询任意节点v_{it}对应的密钥材料,挑战者C通过$\{0,1\}^\kappa$计算生成对应密钥材料sk_{it}并返回给A_2';②阶段2,A_2'指定任一节点v_{0t}(v_{0t}与质询阶段1中的任意v_{it}在空间维度和时间维度上不构成偏序关系,即不存在$v_{0t} \leqslant v_{it}$),质询挑战者C v_{0t}对应的密钥.挑战者C从$\{0,1\}^k$中随机选择r',如果$r'=1$,返回v_0的真实密钥k_{0t},如果$r'=0$,则返回等长随机值k'_{0t}.

分析过程. 敌手A_2'从$\{0,1\}^k$中随机选择r作为对k_{0t}成功的k'_{0t},则A_2'赢得$Game_0$优势定义为$Adv_{A_2',0}=\varepsilon_0=Pr[k'_{0t}=k_{0t}]-1/2$.

2) $Game_1^a$:如果v_{1t}是G中的一个根节点,则$Game_1^a$的过程与$Game_0$类似,唯一区别在于生成k_{1t}的算法是由随机值替代.利用$Game_1^a$和$Game_0$之间的可区分性,能用于构造一个多项式时间算法,以不可忽略的优势攻陷伪随机函数f_v,即

$$|\varepsilon_1-\varepsilon_0|<negl(f_v) \tag{4}$$

存在.

$Game_1^b$:如果v_{1t}是G中一个根节点p的子节点,即$v_{1t} \prec \cdot p$. $Game_1^b$的过程与$Game_0$类似,唯一区别在于生成k_{1t}的密钥推导算法由随机值代替.利用$Game_1^b$和$Game_0$之间的可区分性,能构造一个多项式时间算法,以不可忽略的优势攻陷伪随机函数f_v和对称加密方案E.即

$$|\varepsilon_1-\varepsilon_0|<2negl(f_v)+negl(E) \tag{5}$$

存在.

3) 不失一般性,我们对$Game_i(i=1,2,\cdots)$作同样描述.

$Game_i$的过程与$Game_{i-1}$相同,唯一的区别在于生成k_{it}的密钥推导算法由随机值代替.利用$Game_i$和$Game_{i-1}$之间的可区分性,能用于构造一个多项式时间算法,以不可忽略的优势攻陷伪随机函数f_v和对称加密方案E.即

$$|\varepsilon_i-\varepsilon_{i-1}|<2negl(f_v)+negl(E) \tag{6}$$

存在.

4) $Game_h$:

假设A_2'已未完成h轮的游戏过程,在整个密钥获取计算中,敌手A_2'始终无法区分挑战者处给其密钥是真实密钥还是等长随机字节流,从而使$Game_h$过程也不存在任何公共信息能够使敌手A_2'通过推导得到真实密钥,则$Game_h$中A_2'成功猜测挑战者返回信息是真实密钥k_{ht}的优势定义为$Adv_{A_2',h}$:

$$\varepsilon_h=Pr[k'_{ht}=k_{ht}]=1/2. \tag{7}$$

将式(4)～(7)结论进行合并,可得,$\varepsilon_0 < negl(f_v)+(h-1)(2negl(f_v)+negl(E))+1/2$. 证毕.

6 总 结

泛在感知环境下感知节点数量多,计算、存储能力受限的特点,使用户在对节点资源访问过程中,需要在尽可能掌握较少密钥、并通过较短的密钥获取时间和产生较少量的公共信息情况下,安全高效地访问更多感知节点资源.而加入时间因素的情况下,使满足该实际应用需求的分层访问控制方案设计变得更加困难.本文在提出TLPOS方案的基础上,创新性地利用树重心的特性,提出基于树重心分解的分层访问控制方案TCDS.相比较现有其他方案,方案能够满足在用户掌握单个密钥的前提下,使密钥获取时间和产生的公共信息之间达到最佳平衡,从而更好地适用于泛在感知环境的实际访问控制需求.

参 考 文 献

[1] Want R. Enabling ubiquitous sensing with RFID [J]. Computer, 2004, 37(4): 84-86

[2] Puccinelli D, Haenggi M. Wireless sensor networks: Applications and challenges of ubiquitous sensing [J]. IEEE Circuits & Systems Magazine, 2010, 5(3): 19-31

[3] Akl S, Taylor P. Cryptographic solution to a problem of access control in a hierarchy [J]. ACM Trans on Computer Systems, 1983, 1(3): 239-248

[4] Tzeng W G. A time-bound cryptographic key assignment scheme for access control in a hierarchy [J]. IEEE Trans on Knowledge and Data Engineering, 2002, 14(1): 182-188

[5] Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks [C] //Proc of IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2003: 197-213

[6] Santis A D, Ferrara A L, Masucci B. Enforcing the security of a time-bound hierarchical key assignment scheme [J]. Information Sciences, 2006, 176(12): 1684-1694

[7] Tang Q, Mitchell C J. Comments on a cryptographic key assignment scheme [J]. Computer Standards & Interfaces, 2005, 27(3): 323-326

[8] Yi X. Security of Chien's efficient time-bound hierarchical key assignment scheme [J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17(9): 1298-1299

[9] Wang S Y, Laih C S. Merging: An efficient solution for a time-bound hierarchical key assignment scheme [J]. IEEE Trans on Dependable and Secure Computing, 2006, 3(1): 91-100

[10] Tzeng W G. A secure system for data access based on anonymous authentication and time-dependent hierarchical keys [C] //Proc of ACM Symp on Computer and Communications Security. New York: ACM, 2006: 223-230

[11] Dacro P, De Santis A, Ferrara A L, et al. Variations on a theme by Akl and Taylor: Security and tradeoffs [J]. Theoretical Computer Science, 2010, 411(1): 213-227

[12] Ateniese G, Santis A D, Ferrara A L, et al. Provably-secure time-bound hierarchical key assignment schemes [J]. Journal of Cryptology, 2012, 25(2): 243-270

[13] Santis A D, Ferrara A L, Masucci B. New constructions for provably-secure time-bound hierarchical key assignment schemes [C] //Proc of the 12th ACM Symp on Access Control Models and Technologies. New York: ACM, 2007: 133-138

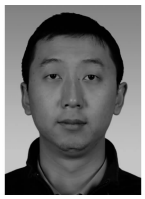
[14] Ma Jun, Guo Yuanbo, Ma Jianfeng, et al. A hierarchical access control scheme for perceptual layer of IoT [J]. Journal of Computer Research and Development, 2013, 50(6): 1267-1275 (in Chinese)
(马骏, 郭渊博, 马建峰, 等. 物联网感知层一种分层访问控制方案[J]. 计算机研究与发展, 2013, 50(6): 1267-1275)

[15] Steeb W H. Sorting and searching [J]. Art of Computer Programming, 2005, 15(5): 27-41

[16] Kang A N C, Ault D A. Some properties of a centroid of a free tree [J]. Information Processing Letters, 1975, 4(1): 18-20

[17] Alstrup S, Holm J, Lichtenberg K D, et al. Maintaining information in fully dynamic trees with top trees [J]. ACM Trans on Algorithms, 2005, 1(2): 243-264

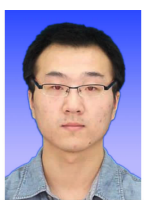
[18] Santis A D, Ferrara A L, Masucci B. Efficient provably-secure hierarchical key assignment schemes [J]. Theoretical Computer Science, 2011, 412(41): 5684-5699



Ma Jun, born in 1981. PhD candidate at Xidian University. His main research interests include wireless network security, key assignment.

Guo Yuanbo, born in 1975. PhD, professor and master supervisor. His main research interests include wireless network security, cryptography, computer network and information security (yuanbo_g@hotmail.com).

Ma Jianfeng, born in 1963. PhD, professor and PhD supervisor. Senior member of CCF. His main research interests include cryptography, computer network and information security (jfma@mail.xidian.edu.cn).



Zhang Qi, born in 1983. Master from Wuhan University of Technology. His main research interests include wireless network security, computer network and information security (zhangqi_qian@163.com).