计算机应用研究
Application Research of Computers
ISSN 1001-3695,CN 51-1196/TP

《计算机应用研究》网络首发论文

题目：车联网环境下基于多策略访问树的安全访问控制算法研究
作者：张迪，曹利，李原帅
DOI：10.19734/j.issn.1001-3695.2023.03.0125
收稿日期：2023-03-28
网络首发日期：2023-06-20
引用格式：张迪，曹利，李原帅．车联网环境下基于多策略访问树的安全访问控制算法研究[J/OL]．计算机应用研究．
https://doi.org/10.19734/j.issn.1001-3695.2023.03.0125

车联网环境下基于多策略访问树的安全访问控制算法研究

张 迪，曹 利†，李原帅

(南通大学 信息科学技术学院，江苏 南通 226001)

摘 要：车联网环境下传统的资源访问控制技术效率较低且存在安全隐患。因此，提出了一种创新性多策略访问树算法的资源访问控制方案。该方案将传统的单策略访问树改进为多策略访问树，实现多级策略制定，有效提高策略访问效率，实现对资源的细粒度访问。同时，方案通过引入拉格朗日多项式秘密值恢复机制，将秘密值与授权令牌结合，解决了车联网访问控制授权的安全性问题。通过安全性分析证明了方案具有身份匿名性和消息不可否认性；性能分析的结果表明，与其他方案相比，由于该方案采用了多策略访问树存储策略，在策略执行系统中的存储开销较小；且随着该方案的策略匹配效率提高，计算开销明显变小，说明了该方案适用于在车联网环境下的安全，高效的资源访问控制。

关键词：车联网；访问控制模型；多策略访问树；拉格朗日插值

中图分类号：TP393 doi: 10.19734/j.issn.1001-3695.2023.03.0125

Research on security access control algorithm based on multipolicy access tree for VANET

Zhang Di,Cao Li†,Li Yuanshuai

(College of Information Science and Technology, Nantong University, Nantong Jiangsu 226001, China)

Abstract: In the context of the Internet of Vehicles, traditional resource access control techniques are inefficient and present security risks. therefore, this paper propose an innovative resource access control solution based on a multi-strategy access tree algorithm. this solution improves the traditional single-strategy access tree by transforming it into a multi-strategy access tree, achieving multi-level strategy formulation and effectively enhancing the efficiency of policy access and fine-grained access to resources. in addition, the solution solves the security issue of access control authorization in the Internet of vehicles by introducing the Lagrange polynomial secret value recovery mechanism, which combines the secret value with the authorization token. through security analysis, it have proven that the solution has identity anonymity and message non-repudiation. performance analysis reveals that, in comparison to other solutions, the scheme has lower storage overhead in the policy execution system by utilizing the multi-strategy access tree to store policies. moreover, as the efficiency of policy matching increases, the computational overhead significantly diminishes, indicating that the solution is well-suited for secure and efficient resource access control in the context of the Internet of Vehicles.

Key words: VANET; access control model; multi-strategic access trees;Lagrange interpolation

0 引言

车载自组网(Vehicular Ad hoc network，VANET)作为物联网在智慧交通领域的重要分支，按照约定的通信协议和数据交互标准,在车与车、车与路边基础设施、车与外部网络之间,执行无线信息传输及交互,进而实现智能交通管理。同时，车载自组网可通过共享其感知信息、储存资源、多余算力，实现车辆协调通信、智能决策，并有效提高空余资源利用率。但由于车联网通信拓扑的高动态性和自组织性，路边单元和动态车辆之间形成多域、开放的网络环境，导致车辆系统资源的敏感信息易泄露或易受窜改，使车主的财产和人身安全受到威胁[1]。因此，利用访问控制技术对车联网中的资源传播进行必要管控，是资源安全共享的前提。

传统的访问控制技术有自主访问控制[2](Discretionary Access Control,DAC)、强制访问控制[3](Mandatory Access Control，MAC)、基于角色的访问控制[4](role-based access control,RBAC)等，主要适用于集中式安全控制系统中。但车联网因其分布式、开放信道、动态组网、对通信时延高度敏感新等特性，不能直接采用传统访问控制模型。

基于属性的访问控制(Attribute-based Access Control ABAC)[5]是解决分布式环境下的访问控制问题而发展起来的，ABAC 方法的核心思想是主体和资源之间的访问基于属性来授权，访问控制实体的属性是访问控制决策的最小粒度[6]。简单而言，就是访问控制系统根据用户的属性赋予用户访问相应数据的权

收稿日期：2023-03-28；**修回日期**：2023-05-24 **基金项目**：南通市科技项目(JC22022036，JC2021128)

作者简介：张迪(2002-)，女，河南许昌人，主要研究方向为网络与信息安全；曹利(1974-)，男(通信作者)，江苏宜兴人，副教授，主要研究方向为网络空间安全、区块链技术、物联网安全(cl@ntu.edu.cn)；李原帅(2003-)，河南焦作人，主要研究方向为网络与信息安全.

限。由于 ABAC 模型具有分布式、资源可扩展、访问控制授权细粒度的优势，在车联网环境下，可降低授权逻辑复杂性，提高访问控制授权的灵活性。因此，国内外学者对基于 ABAC 模型的车联网访问控制算法研究做了大量工作。如 Huang 等学者[7]提出将 CPABE 加密方法引入到车联网中，利用基于属性制定安全访问策略框架，实现资源共享过程中细粒度的访问控制；Liu 等学者[8]提出通过将部分复杂的计算外包至第三方解决属性基加密的时间消耗大的问题；但对于外包的第三方可信度以及第三方的稳定度条件要求苛刻，容易产生数据泄露或单点故障问题；庞立君等学者[9]提出了一种基于 CP-ABE 的车联网云端数据安全访问方案，解决基于属性的加解密算法时间消耗大的问题，达到加密解密高效率的特性，但车云服务商计算代价较大，容易产生较大的时延。刘雪娇等学者[10]尝试通过引入区块链去中心化的数据存储解决传统存储条件下数据易窜改易泄漏的问题，但去中心化区块链数据上传会影响数据的共享效率，导致产生较大的时延；侯慧莹等[11]提出了一个面向自动驾驶的高效可追踪的车联网匿名通信方案，通过属性集与车辆之间一对多的关系体现出了匿名性；但实现匿名性的同时却损失了访问控制方案的细粒度，难以满足车辆网环境下对细粒度的高要求。

综上，尽管学者在基于 ABAC 模型的基础上对车联网访问控制做了大量优化工作，但其基本思路均为：利用访问控制树实现控制策略，只要属性集合满足对应树节点的条件，就能解出该节点中的秘密值，从而获得访问控制的权限。这些方案最大问题是，它是一种单策略访问控制模型，一棵树只能定义一个访问策略。在车联网实际环境下，由于控制访问规则复杂，单策略访问控制树在高效性、细粒度，低时延，高精准等方面无法满足车联网的访问控制特性需求。

本文创新性的提出一种多策略访问树的数据结构，在此据结构上实现多访问策略绑定，有效提高资源访问效率，并可满足对资源的细粒度访问。同时，本文引入拉格朗日多项式秘密值恢复机制，将秘密值与授权令牌结合，解决了车联网访问控制授权的安全和隐私问题。

1　相关知识

1.1　访问树定义

1)设 $U=\{U_1,U_2,...,U_n\}$ 为系统中所有参与实体的集合，访问结构 A 是集合 U 的非空属性子集，即 $A\subseteq 2^U$ 且非空。其中，U 表示系统的属性域，而访问结构 A 为授权属性集。

2)设 T 是一个访问树。T 中的每个内部节点表示一个门限结构，用 (k_x,num_x) 描述，其中，num_x 表示 x 孩子节点的个数，k_x 表示门限值，且有 $0<k_x\le num_x$。当 $k_x=1$ 时，表示任意子节点至少一个满足，即 OR 门；当 $k_x=num_x$ 时，表示所有子节点都满足，即 AND 门。叶节点 x 用来描述属性，其门限值 k_x 为 1。

3)在访问树结构上定义三个函数，$parent(x)$ 用于返回节点 x 的父节点，$att(x)$ 用于返回叶子节点 x 描述的属性，$index(x)$ 返回节点 x 在其兄弟节点中的索引值。

4)设 T 是一个以 r 为根节点的访问树，用 T_x 表示以 x 为根节点的子树。若一个属性集 A_i 满足访问树 T_x，就表示 $T_x(A_i)=1$。通过以下方式递归计算 $T_x(A_i)$。

a)若 x 是非叶子节点，对 x 的所有孩子节点 x' 计算 $T_{x'}(A_i)$。当且仅当至少有 k_x 个孩子节点 x' 返回 $T_{x'}(A_i)=1$ 时，$T_x(A_i)=1$。

b)若 x 是叶子节点，当且仅当 $attr(x)\in A_i$ 时，$T_x(A_i)=1$。

1.2　拉格朗日插值定理

设 $f(x)$ 是一个 n 阶多项式，若已知在该多项式上有 n+1 个互异点 $(x_i,f(x_i))$，则可得到拉格朗日多项式为

$$f(x)=\sum_{i=0}^{n} f(x_i)\Delta_{i,s}(x)$$

其中，拉格朗日系数为 $\Delta_{i,s}(x)=\prod_{i=0,i\neq j}^{n}\frac{x-x_i}{x_j-x_i}$。

本文根据拉格朗日插值实现访问控制。方案在访问树生成过程中，根节点(假设门限值为 2，孩子节点有 3 个)随机生成一个一次多项式，如 $f(x)=6+2x$，秘密值为 6。假设根节点的孩子节点从左到右依次标记为 1、2、3，将其带入根节点的多项式中，得到新生成的秘密值($f(1)=8$，$f(2)=10$，$f(3)=12$)传递给孩子节点保存。同样按照上面的方式，再传递给它们的孩子节点(即叶子节点)，生成新的秘密值。

解密访问树的秘密值，正好是与构造过程相反，从叶子节点属性出发。访问者需要在属性集中寻找出和此叶子节点属性相同的属性，并用找出的属性，根据拉格朗日多项式插值定义解密出它们的秘密值，然后进一步解密其父子节点，从而追溯到最后解密根节点的秘密值。

2　基于多策略访问树的访问控制方案

2.1　系统模型

方案系统架构如图 1 所示，分三个层次，底层为 OBU 所组成的网络，中间层为 RSU 接入网，顶层为可信认证机构(可信认证中心、策略执行系统)。

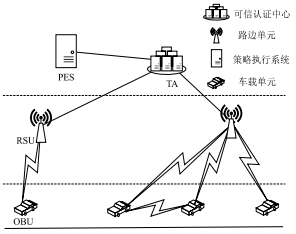

图 1　系统逻辑架构图

Fig.1　System logical architecture diagram

a) 可信认证机构：可信机构 TA 是假定为完全可信的管理者，在本方案中包含可信认证中心 TA 和策略执行系统 PES(Policy Enforcement System)两部分。可信认证中心 TA 是权威机构，与路边单元 RSU 通过有线网络建立连接，主要负责初始化系统参数、为车辆及 RSU 提供 PKI 认证服务，辅助 RSU 实现移动车辆节点的可靠接入；策略执行系统 PES(即访问控制服务器)负责策略初始化定义和授权控制工作。

b) 路边单元(RSU)：RSU 是一些部署在路边的车联网的通信设施，通过无线通信方式与通过其通信范围内的车辆进行通信。同时 RSU 还通过有线通信方式与可信机构或者应用服务器进行通信。为了研究更具集中性，本文主要集中于无线通信中的安全问题，因此假定有线通信过程是安全可靠的。

c) 车载单元(OBU)：每个车辆的装备有车载单元，车载单元具有一定的计算和存储能力，利用 DSRC 车载短程通信协议或者其他无线通信协议与其他 OBU 和 RSU 进行组网并发送各种消息，包括车辆的位置、行驶方向、速度、交通条件、危险警告及其他交通状态相关的信息。

2.2　系统实现

本文创新性的提出一种基于多策略访问树的安全访问控制方案，其过程是：系统初始化、多策略访问树生成、车辆入网双向验证、资源访问控制、资源授权。表 1 为方案描述

时相关符合对应含义。

表 1　符号及含义

Tab. 1　Symbols and meanings

符号	含义
G	双线性群
S_{MK}	主密钥
H_0, H_1, H_2	哈希函数
P_{pub}	系统公钥
A	标准属性集
$Cert_R$	证书
P_{V_i}, S_{V_i}	车辆 V_i 的公私钥
key	会话密钥
F_i	车辆 V_i 的假名
P	策略
$secret$	秘密值
$token$	令牌
$Tree$	访问树
R	资源集
TS	当前时间的时间戳

2.2.1　系统初始化

1) 系统参数初始化

a)TA 随机选取素数 q 和 s，$s \in Z_q^*$，计算 $\beta = \alpha^s (\mathrm{mod} q)$，并将 s 作为系统主密钥 S_{MK}，β 作为系统公钥 P_{pub}。

b)TA 选择三个单向抗碰撞的哈希函数：$H_0, H_1, H_2 : \{0,1\}^* \to Z_q^*$。

c)TA 公布系统参数 $Param = \{q, \alpha, \beta, H_0, H_1\}$，保留系统主密钥 S_{MK}。

2)属性标准化

基于属性的访问控制以属性为基本单位对实体的权限进行控制，属性包括访问主体、访问的资源、访问操作、环境，环境属性包含车辆的网络状况、道路环境、历史信息等。策略执行系统在初始化阶段，定义并产生车联网系统中车辆实体的标准属性集 $A = \{a_1, a_2, ..., a_n\}$，如表 2 所示。

表 2　车辆的标准属性集

Tab. 2　A standard set of attributes for a vehicle

属性类型	主体属性	操作属性	环境属性
固定属性	车辆生产商	-	道路位置
	车辆生产年份	-	道路方向
	车主信息	-	道路属性
	车辆类型	-	-
可变属性	车辆的资源	操作时间	网络状况
	车辆的行为	操作方式	网络类型
	资源的安全等级	-	-
	资源的大小	-	-

3)实体注册

在本阶段，主要实现实体 RSU 和车辆 V_i 的线下初始化注册。RSU 和车辆 V_i 分别向 TA 提交其基本信息(公私钥、身份信息)，TA 为 RSU 和车辆提供的公钥颁发证书。具体流程见图 2。

a)在部署 RSU 前，车管所统一购进 RSU 设备并为其初始化。RSU 选择随机数 $y_k \in Z_q^*$ 作为其的私钥 S_R，计算 RSU 的公钥 $P_R = \tau^{y_k} \mathrm{mod} q$，其中 τ 是一个本原元。

b) TA 为 RSU 颁发证书 $Cert_R$，证书包含：RSU 的公钥、证书有效期、TA 的私钥签名、RSU 的位置信息等。RSU 将证书 $Cert_R$、公钥存储在本地。

c) 车辆 V_i 注册由可信认证中心 TA 完成。车辆 V_i 向 TA 提交基本信息(车牌号、车主身份信息、车辆登记信息、车辆违法信息等)，TA 核实其真实性后，将 ID_i 设置为车辆 V_i 的唯一身份标识。

d) 车辆 V_i 选择随机数 $x_i \in Z_q^*$，计算车辆 V_i 的公钥 $P_{V_i} = \gamma^{x_i} \mathrm{mod} q$，并把 x_i 设置为车辆 V_i 的私钥 S_{V_i} 存储于 OBU 的不可窜改设备中，其中 $\gamma \in Z_q^*$ 是一个本原元。

e)TA 为车辆 V_i 颁发证书 $Cert_{V_i}$，证书包括：车辆 V_i 的公钥 P_{V_i}、身份标识 ID_i、证书的有效期、TA 的私钥签名等，车辆 V_i 将证书存储在 OBU 的不可窜改设备中。

f)TA 选择随机数 n_1 为车辆 V_i 计算伪身份：$ID_i' = ID_i \oplus H_0(ID_i \| n_1)$，并将车辆 V_i 的伪身份 ID_i'、身份标识 ID_i 进行关联存入身份映射表中。

g)策略执行系统 PES 根据车辆 V_i 提交的基本信息，映射到标准属性集 $A = \{a_1, a_2, ..., a_n\}$，产生本车辆 V_i 的属性子集 A_{V_i}。其中，A_{V_i} 为车辆 V_i 在访问控制系统中的授权属性集，并将 A_{V_i} 存储在车载 OBU 中的不可窜改设备内。

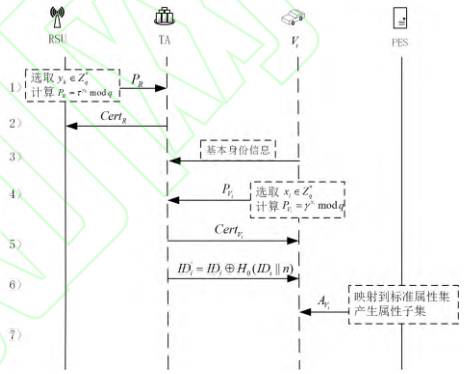

图 2　线下实体注册

Fig. 2　Offline entity Registration

2.2.2　生成多策略访问树

本文利用访问树结构表达车辆的资源访问控制策略。步骤为，首先定义本车的访问策略，生成访问策略表，然后将该访问策略表转换成对应多策略访问树结构，并将访问树的节点赋予秘密值，将不同节点的秘密值和相关的访问策略绑定。具体如下：

1)定义访问策略表

如表 3 所示，车辆 V_i 可以对自己的不同的开放资源定义不同的访问策略，实现细粒度访问。具体为，通过定义多个访问策略，将满足不同的属性集映射到不同的资源 R_k，形成三元组访问策略< P_i，A_i，R_i >。

表 3　访问策略表定义

Tab. 3　Access policy table definition

策略	属性集	资源集
P_1	A_1	R_1
...	...	...
P_i	A_i	R_i
...	...	...
P_n	A_n	R_n

2)生成多策略访问树

多策略树生成的目的是将车辆定义的访问策略表转换成多策略访问树的数据结构。这种创新的数据结构通过一颗生成树的不同节点秘密值映射到不同策略，并将之和授权令牌绑定，从而实现高效灵活安全的多策略访问控制。

将访问策略表转换成对应访问树结构的方法为：假设车

辆属性为(A，B，C，D，E)，非叶子节点为门节点，存储秘密值。将车辆定义的策略表转换成访问树结构的过程为，若根据车辆定义的策略表，车辆开放相关资源需要满足以下 5 个属性集之一，表达为属性集 1：{属性 A，属性 B}，属性集 2：{属性 C }，属性集 3：{属性 D }，属性集 4：{属性 E}，属性集 5：{属性 B}。

而生成的多策略树结构对应的逻辑表达式为(属性 A && 属性 B)||(属性 C ‖ 属性 D)||属性 E。对应的多策略树结构如图 3 所示。

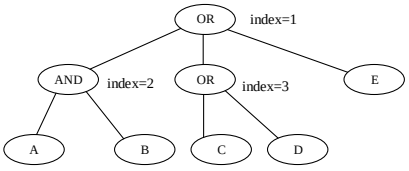

图 3　多授权访问树结构

Fig. 3　Multi-authorization access tree structure

车辆 V_i 根据图 3 多策略树结构即可构造访问树 $Tree_i$。其中，多策略访问树采用广度遍历中的中序遍历方式，叶子节点的值为属性值，而非叶子节点中包含门限值、节点索引值 $index_k,(k\in\{1,2,3,...\})$ 、秘密值 $secret_k$ 和令牌 $H_1(index_k,secret_k)$。只有满足门限值的限制才可以对该非叶子节点解密，从而获得该非叶子节点的秘密值，然后获得对应资源的授权令牌。

具体为，方案将多策略树生成算法定义为 4 阶段：定义访问树节点、初始化数据、建立生成树、执行令牌策略绑定。

步骤 1：定义访问树节点。

为建立访问树，需要定义树的节点，节点分为叶子和非叶子两类，Node 数据结构将两类节点均定义在内，如下所示。

Defining tree nodes algorithm：

```
1) struct node
2) {string attr;/*属性值*/
int gate[][]={ (k_x,num) };/*门限值，节点的孩子节点数*/
3) int index;/*节点索引值*/
4) int children[];/*非叶子节点的孩子节点索引值列表*/
5) f(x)=Polynomia(d,coef[],secret);/*非叶节点的多项式*/
6) int d;/*多项式的次数*/
7) random coef[];/*多项式系数*/
8) Pairing secret;}/*秘密值*/
```

步骤 2：利用 Node 数据结构初始化数组 N，将访问树的 n 个节点的属性值、策略、门限值、孩子节点的索引编号分别存入 N.atts、N.Policy、N.gates、N.children。算法描述如下：

Data initialization algorithm：

```
1) Init_Tree_GetData(N){
2) N.atts[]=atts[].insert(attr);
N.Policy[]=Policy[].insert( P_k );
3) N.gates[][]=gates[][].insert(gate);
4) N.children[]=children[].insert(children[]);
5) return N; }
```

步骤 3：建立生成树，利用数组 N 中的数据为生成树中每个节点赋值。算法描述如下：

Tree initialization algorithm：

```
1) Init_Tree(node N[], node  nodes[])
2) { for i← 0 to 节点个数 do
3)    {if(nodes[i].is leaf)
4)      { nodes[i].attr= N.atts[];
5)     nodes[i].index=i;}
6)     if(nodes[i].is not leaf)
7)       { nodes[i].gate= N.gates[][];
8)       nodes[i].children= N.childrens[];
9)       nodes[i].index=i;
10) }}
11) return nodes[];}
```

算法分析：若节点为叶子节点，则利用初始化数组 N 中的属性集 atts 为叶子节点赋属性值；若是非叶子节点，则根据初始化数组 N 中的门限值集合 gates 和孩子节点集合 children 为节点的门限值 和孩子节点赋值。

步骤 4：执行令牌策略绑定算法。在本步骤主要实现访问树非叶子节点的秘密值和访问策略表的权限映射，从而将访问树定义成一棵多策略访问树。

Algorithm 1　Token-Policy binding algorithm

```
Input: Nodes of the access tree nodes[], A certain node
of the access tree n .
Output: Token policy mapping table T_P_map[][] .
1) Gloabl T_P_map[][];
2) Token-Policy binding (nodes[], node n){
3) node childnode[];
4) nodes[0].secret=random();
5) nodes[0].token= hash(index, nodes[0].secret);
/*映射根节点的令牌和策略*/
6) T_P_map[][]=T_P_map[][].insert{(nodes[0].token,
nodes[0].Policy)};
7) if(n is not leaf) then
/*构造随机多项式*/
   n.f(x) = n.secret + n.coef[1]×x^1 + n.coef[2]×x^2 + ... +
            n.coef[K_x − 1]×x^(K_x − 1) ;
   for j←0 to n.num do
/*计算孩子节点的秘密值*/
8)    childnode[j].secret=f(index);
9)    childnode[j].token=hash(index, childnode[j].
secret);
/*映射孩子节点的令牌和策略*/
10)    T_P_map[][]=T_P_map[][].insert((childnode[j].token,
childnode[j].Policy));
11)     Token-Policy binding (nodes[], childnode[]);
12)   end for
13) return T-P-map[][];}
```

对以上算法说明为：

a)车辆选择随机数作为多授权访问树根节点 n 的秘密值，计算该秘密值和根节点的索引值的哈希值，将结果作为根节点的授权令牌。同时，在授权令牌和访问策略表中该节点对应的策略之间建立映射关系。

b)以根节点的门限值减 1 为多项式 f(x)的次数，随机数数组 coef[]为多项式 f(x)的系数，根节点的秘密值(secret)为多项式 f(x)的常数项，x 为变量，生成多项式 f(x)。并将该多项式作为根节点的节点多项式。

c)车辆将根节点孩子节点的索引值 index 作为自变量，代入根节点的多项式 f(x)计算其孩子节点的秘密值。

d)车辆对孩子节点的秘密值和索引值进行哈希运算，并将哈希值作为其孩子节点的授权令牌。

e)车辆将访问策略表中节点 n 对应的访问策略 *Policy*

与节点 n 的授权令牌 *token* 进行映射，并将该映射存储在 T-A-map[][]表中。

f)执行递归循环，在递归循环中，将该节点的孩子节点作为新的根节点，但是不再需要初始化根节点的秘密值和令牌；直至该根节点为访问树的叶子节点。

g)将令牌策略映射表 T-P-map[][]作为返回结果，用于其他车辆访问时的授权凭证。

3）多策略访问树存储

车辆 V_i 向可信认证中心的策略执行系统递交多授权访问树，包括访问树 $Tree_i$ 、假名 F_i 和令牌策略映射表 T-P-map[][]。其中，访问树 $Tree_i$ 采用孩子表示法存储，即将每个节点的孩子节点都用单链表连接形成线性结构。策略执行系统 PES 将该多策略访问树根节点的地址和对应车辆伪身份 $ID_i^{'}$ 建立映射关系。

2.2.3　车辆入网双向验证

车辆在接入车联网的过程中，首先和区域接入 RSU 双向认证，获得会话密钥，并向该区域 RSU 申报可开放资源，如图 4 所示。

图 4 车辆入网验证

Fig. 4　Vehicle entry verification

1)双向认证

车辆入网验证过程说明：

a)RSU 周期性广播证书和签名。车辆 V_i 驶入 RSU 的通信区域时，获得证书 Cert_R 和 RSU 公钥，并利用该公钥验证 RSU 签名的有效性，从而确定 RSU 身份合法性，同样， 车辆 V_i 向 RSU 发送其证书、时间戳和签名等，时间戳用于防止重放攻击。RSU 获得车辆 V_i 的证书后，利用 V_i 公钥验证车辆 V_i 签名的有效性，从而验证车辆 V_i 身份合法性。

b)双方验证身份合法性后，执行会话密钥生成操作。过程为：RSU 和车辆 V_i 分别选择一个随机数 a,b， RSU 选择大素数 p 以及整数 g，并将其公开。RSU 计算 $K_1 = g^a \bmod p$ 并发送 $E_{R_{V_i}}(K_1)$ 给车辆 V_i 。车辆 V_i 利用其私钥 S_{V_i} 对 $E_{R_{V_i}}(K_1)$ 解密，并计算会话密钥 $key = (K_1)^b \bmod p$ 。然后，车辆 V_i 计算 $K_2 = g^b \bmod p$ 并发送 $E_{P_R}(K_2)$ 给 RSU。RSU 对 $E_{P_R}(K_2)$ 解密，并计算会话密钥 $key = (K_2)^a \bmod p$ 。会话密钥计算完成后，RSU 为车辆 V_i 计算临时假名 $F_i = H_3(ID_i^{'} \| n_2)$，其中，$n_2$ 为随机数，将临时假名 F_i 与伪身份 $ID_i^{'}$ 相互关联映射存入伪身份映射表中。并发送 $E_{key_i}(F_i \| TS)$ 给车辆 V_i 。在本区域中，车辆利用临时假名进行通信。

2)申报可开放资源

客体车辆 V_j 利用会话密钥生成密文消息 $E_{key_j}(F_j \| R_k \| loc_{V_j} \| TS)$ 发送给 RSU，消息包含：假名 F_j 、资源集 R_k 、车辆 V_j 的位置 loc_{V_j} 和时间戳 TS。RSU 利用会话密钥解密该消息后，首先根据时间戳 TS 验证时效性，若超时，重新验证。然后比较值 $|loc_R - loc_{V_j}|$ 是否大于 ΔL ，若是，说明车辆 V_i 已经驶离当前 RSU 的管辖区域，拒绝本次申报。否则将该车辆 V_j 相关字段值存入表 4 的车辆可用资源表中，表示该资源对本 RSU 区域的车辆开放申请。

表 4　车辆可用资源表

Tab. 4　Table of available resources for vehicles

车辆假名	公钥	资源集	车辆位置	时间戳
F_1	P_{V_1}	R_1	loc_{V_1}	TS_1
F_2	P_{V_2}	R_2	loc_{V_2}	TS_2
...	...	...	...	...
F_n	P_{V_n}	R_n	loc_{V_n}	TS_n

2.2.4　资源访问控制

假设车辆 V_i 和 V_j 已经在本区域的 RSU 完成注册验证，且车辆 V_i 为主体，车辆 V_j 为客体。车辆 V_i 请求资源的访问控制过程如图 5 所示。

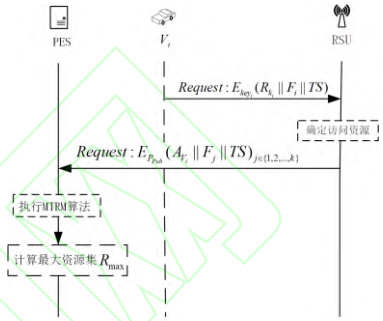

图 5　资源访问控制

Fig. 5　Resource access

1)　V_i 申请资源

a) $V_i \rightarrow RSU : \{E_{key_i}(R_{k_i} \| F_i \| TS)\}$ 。

主体车辆 V_i 向 *RSU* 发送资源请求报文 $E_{key_i}(R_{k_i} \| F_i \| TS)$ 。其中，集合 R_{k_i} 表示车辆 V_i 所需资源请求集，*TS* 表示时间戳，用于防止重放攻击。

b)定位客体车辆。

RSU 接收申请报文，解密验证时效性后，定位满足资源需求的客体车辆 V_j 。过程为

(a)RSU 为主体车辆 V_i 生成请求资源属性集 A_{V_i} 。

(b)根据车辆 V_i 的资源请求集检索车辆可用资源表，并返回符合条件的客体车辆 V_j 假名。若存在多个匹配，返回符合条件的客体车辆假名集 $\{F_1,...,F_k\}$ 。

2)资源访问请求

a) $RSU \rightarrow PES : \{E_{P_{Pub}}(A_{V_i} \| F_j \| TS)_{j\in\{1,2,...,k\}}\}$ 。

RSU 向 PES 转发主体车辆 V_i 的资源访问请求 $\{E_{P_{Pub}}(A_{V_i} \| F_j \| TS)_{j\in\{1,2,...,k\}}\}$，请求报文包含如下字段：主体车辆 V_i 的属性集 A_{V_i} 、满足条件的客体车辆的假名集 $\{F_1,...,F_k\}$ 和时间戳 *TS* ，利用系统公钥 P_{pub} 加密本请求报文。

策略执行系统 *PES* 利用主密钥 S_{MK} 解密该请求报文，并以请求报文字段中车辆假名集 $\{F_1,...,F_k\}$ 为多策略访问树的索引条件，确定多个假名 F_j 对应的多策略访问树，并针对每一棵多策略访问树按以下算法，逐一进行授权匹配检查。

b)多策略树资源匹配算法。

PES 利用 *RSU* 发来的资源请求信息获得多棵访问树 $tree_j$ 后，执行多策略树资源匹配(MTRM)算法为主体车辆资源授权检查。

Algorithm 2　Multi-policy tree resource matching algorithm

Input: Nodes of the access tree *nodes[]*, The attributions of the subject vehicle attrs, A certain node of the access tree *n* .

Output: The policy set *PolicyList[]* .

```
1) for n←节点个数 to 0 do
2)   if(n is leaf)
3)    if (atts.contains(n.attr))
4)      n.valid = true;
5)      if(n is not leaf)
/*对 n 的每一个子节点进行判断是否有效*/
6)      foreach(j:childnode[])
7)        if (childnode[j].valid)
8)           validChildren[].insert(n.childnode[j]);
            if(validChildren[].size() ≥ n.k_x )then
9)              n.valid = true;
10)             foreach(i:validChildren[]){
11) n.secret=n.secret.add(nodes[i].secret.multiply
(lagrange(i, validChildren[], 0))；}
/*对有效节点计算令牌，并插入策略列表*/
                n.token = hash(n.index, n.secret);
12)             if (n.token == T_P_map[][].token) then
13)                PolicyList[].insert(T_P_map[][].
Policy);
14)    end for
15) return PolicyList[];
```

算法流程如下：

(a) PES 从该树的最后一个节点依次向前遍历，直到根节点结束。

(b) PES 根据每一个节点判断是否为叶子节点，如果是叶子节点，判断主体车辆的属性值是否满足该节点的属性值。如果是非叶子节点，对该非叶子节点的每一个子节点进行判断是否为有效节点，若是有效节点，则插入该节点可用孩子数组。

(c) PES 判断该节点 n 可用孩子数组中节点的个数是否达到门限值，确定是否可以恢复节点 n 的秘密值。

(d) PES 遍历可恢复的非叶子节点 n 可用孩子数组，计算有效子节点的拉格朗日插值因子，并利用有效子节点的拉格朗日插值因子和秘密值恢复该非叶子节点 n 的秘密值。

(e) PES 将该非叶子节点 n 的秘密值和该节点索引值进行哈希运算，并将计算结果作为该节点 n 的令牌。

(f) PES 利用该节点的令牌与该多策略访问树的资源拥有者产生的 T_P_map[][]中的令牌属性列中的元素进行匹配，将满足令牌匹配条件的结果插入策略列表。

(g) PES 返回策略列表 PolicyList[]，PolicyList[]中包含了多策略访问控制树中所有满足属性匹配的非叶子节点秘密值对应的访问控制策略，实现了一个访问控制树支持多策略的细粒度、高效访问控制。

c) 生成最高权限资源集。

PES 根据 MTRM 算法获得的 PolicyList[]检索表 3， 若 PolicyList[]获得的策略分别为 $\{P_1,P_2,...,P_k\}$，表 3 访问策略表中对应的资源集应为 $\{R_1,R_2,...,R_k\}$，则可生成最大策略资源集 $R_{\max}=R_1\bigcup R_2\bigcup...\bigcup R_k$。*PES* 将 $R_{\max}$ 作为主体车辆 V_i 可访问的资源集。

2.2.5　资源授权

资源授权如图 6 所示。

1) $PES\rightarrow RSU:\{E_{P_k}(F_i\|R_{\max}\|F_j\|TS)\}$

PES 以 *RSU* 公钥加密授权通知报文后发送给 *RSU*，包括以下字段：主体车辆假名 F_i、客体车辆假名 F_j、V_i 的可访问资源集 $R_{\max}$ 和时间戳 *TS*。

2) $RSU\rightarrow V_i:\{E_{key_i}(success\|TS)\}$

RSU 生成授权通知报文 1 给主体车辆 V_i，该报文包含车辆 V_i 资源请求的状态和时间戳 TS，并利用会话密钥 key_i 加密。车辆 V_i 若接收到该报文，则表明资源请求授权成功。

3) $RSU\rightarrow V_j:\{E_{key_j}(R_{\max}\|TS)\}$

RSU 生成授权通知报文 2 给客体车辆 V_j，该报文包含 V_j 的可访问资源集 $R_{\max}$ 和时间戳 TS， V_j 开放资源等待主体车辆 V_i 的访问。

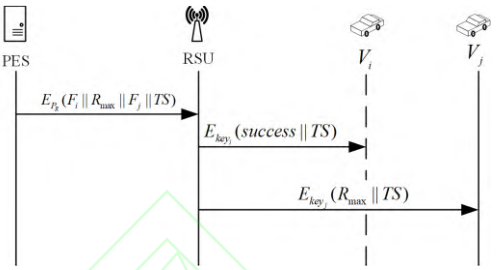

图 6　资源授权

Fig. 6　Resource authorization

3　实验结果与分析

3.1　车联网资源访问控制模型比较

将本方案模型与其他方案比较，如表 5 所示。

表 5 资源访问控制模型比较

Tab. 5　Comparison of resource access control models

方案	访问控制	决策设备	资源存储方式	资源存储位置	访问策略构造
Pan 等[12]	基于属性	云端	集中式	部分车辆与云端	LSSS
Kanumalli 等[13]	基于角色	信息中心	集中式	信息中心	-
Ma 等[14]	基于属性	RSU	分布式	各车辆	访问树
刘雪娇等[10]	基于属性	RSU	分布式	RSU	LSSS
本方案	基于属性	PES	分布式	各车辆	访问树

从访问控制方面来看，Kanumalli 等[13]的方案模型使用基于角色的访问控制方式，在涉及大量角色的车联网环境下可能会存在角色爆炸问题，相较于基于角色的访问控制方式将访问权限与角色相关联，本方案使用基于属性的访问控制方式，并通过相关属性进行识别，更具灵活性与细粒度。

从决策设备方面来看，刘雪娇等[10]、Ma 等[14]的方案模型使用 RSU 作为决策设备，作为半可信的路侧基础设施，相较于本方案使用 PES 等独立决策中心进行决策，RSU 存在计算能力较弱且安全性较低的缺点，而本方案模型更具安全性与高效性。

从资源存储方式与位置来看，Pan 等[12]、Kanumalli 等[13]的方案模型使用集中式存储方式存储在特定资源中心中，刘雪娇等[9]使用分布式存储方式存储在 RSU 中，本方案使用分布式存储方式存储在各车辆中，分布式存储方式能有效防止单点故障问题，资源分布式存储在各车辆中，可共享资源种类更加丰富，从访问策略构造来看，刘雪娇等[10]、Pan 等[12]的方案模型使用 LSSS 构造的访问策略，本方案使用访问树构造的访问策略，然而由于 LSSS 的计算复杂度高，计算开销大。访问树结构计算开销小的同时更具灵活性以及扩展性，更适用于轻量级车联网环境下。

3.2　基于多策略访问树的访问控制安全性分析

3.2.1　基于多策略访问树的访问控制安全性证明

定理 1　已知 $a_1,a_2,...,a_{n+1}$ 是数域 F 中互不相同的数，$b_1,b_2,...,b_{n+1}$ 是数域 F 中任意的不全为 0 的数，若在数域 F 存在唯一的次数不超过 n 的多项式 $f(x)$，使得

$$f(a_i)=b_i(i=1,2,...,n+1) \tag{1}$$

并且多项式 $f(x)$ 可以表示为

$$f(x)=\sum_{i=1}^{n+1}\frac{b_i(x-a_1)...(x-a_{i-1})(x-a_{i+1})...(x-a_{n+1})}{(a_i-a_1)...(a_i-a_{i-1})(a_i-a_{i+1})...(a_i-a_{n+1})} \tag{2}$$

式(2)称为拉格朗日插值公式，则上述多策略访问树访问控制方案中的秘密值的生成和恢复是安全的。

证明　设满足(1)式的次数不超过 n 的多项式 $f(x)=c_0+c_1x+c_2x^2+...+c_nx^n$。

1)多项式存在性和唯一性证明

由式(1)可得：

$$\begin{cases} c_0+c_1a_1+c_2a_1^2+...+c_na_1^n=b_1 \\ c_0+c_1a_2+c_2a_2^2+...+c_na_2^n=b_2 \\ \cdots \\ c_0+c_1a_n+c_2a_n^2+...+c_na_n^n=b_n \\ c_0+c_1a_{n+1}+c_2a_{n+1}^2+...+c_na_{n+1}^n=b_{n+1} \end{cases} \tag{3}$$

将资源请求者的属性集 $A_{V_i}=\{A_1,A_2,...,A_{n+1}\}$ 与资源拥有者多策略访问树的叶子节点相匹配的秘密值 *secret*，分别作为式(3)的常数项 $\{b_1,b_2,...,b_{n+1}\}$，每个叶子节点在其父节点(即非叶子节点)的索引值 *index* 为式(3)中的系数 $a_i(i=1,2,...n+1)$，资源请求者的有效属性值个数为式(3)线性方程组中的最高次数 n。又式(3)是一个以 $c_0,c_1,...,c_n$ 为未知数的 n+1 个线性方程的线性方程组，则其系数行列式(记为 D)是 n+1 阶 Vandermonde 行列式，且

$$D=\begin{vmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^n \\ 1 & a_2 & a_2^2 & \cdots & a_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^n \\ 1 & a_{n+1} & a_{n+1}^2 & \cdots & a_{n+1}^n \end{vmatrix}=\prod_{1\le i<j\le n+1}(a_j-a_i)$$

又因为 $i\neq j$ 时，叶子节点的索引值 a_i,a_j 不相等 $(a_i\neq a_j)$，所以 $D\neq 0$，即线性方程组式(3)有唯一解 $c_0,c_1,...,c_n$。因此，在数域 F 中只存在唯一的次数不超过 n 的多项式 $f(x)=c_0+c_1x+c_2x^2+\cdots+c_nx^n$，使得 $f(a_i)=b_i(i=1,2,\cdots,n+1)$ 成立。从而得出，非叶子节点的多项式函数 $f(x)$ 都可以由其 n 个叶子节点的秘密值唯一确定，且满足 $f(index)=secret$。

2)多项式正确性证明

由 Cramer 法则可得 $c_j=\frac{D_{j+1}}{D}(j=0,1,\cdots,n)$，其中 D_{j+1} 是将 D 中第 $j+1$ 列换成式(3)中的常数列 $b_1,b_2,\cdots,b_{n+1}$ 所得到的行列式

$$D_{j+1}=\begin{vmatrix} 1 & a_1 & \cdots & a_1^{j-1} & b_1 & a_1^{j+1} & \cdots & a_1^n \\ 1 & a_2 & \cdots & a_2^{j-1} & b_2 & a_2^{j+1} & \cdots & a_2^n \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & a_n & \cdots & a_n^{j-1} & b_n & a_n^{j+1} & \cdots & a_n^n \\ 1 & a_{n+1} & \cdots & a_{n+1}^{j-1} & b_{n+1} & a_{n+1}^{j+1} & \cdots & a_{n+1}^n \end{vmatrix}$$

将 $c_j=\frac{D_{j+1}}{D}$ 代入多项式 $f(x)$ 可得式(4)

$$\begin{aligned} f(x)&=\frac{D_1}{D}+\frac{D_2}{D}x+\frac{D_3}{D}x^2+\cdots+\frac{D_{n+1}}{D}x^n= \\ &\sum_{j=1}^{n+1}\frac{D_j}{D}x^{j-1}=\sum_{j=1}^{n+1}(\sum_{i=1}^{n+1}b_iA_{ij})\frac{x^{j-1}}{D} \end{aligned} \tag{4}$$

其中，A_{ij} 是 D 中 a_{ij} 的代数余子式。交换式(4)中的双重求和符号，得到

$$f(x)=\sum_{i=1}^{n+1}b_i(\sum_{j=1}^{n+1}\frac{A_{ij}}{D}x^{j-1}) \tag{5}$$

而 $\sum_{j=1}^{n+1}A_{ij}x^{j-1}=\begin{vmatrix} 1 & a_1 & \cdots & a_1^{n-1} & a_1^n \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & a_{i-1} & \cdots & a_{i-1}^{n-1} & a_{i-1}^n \\ 1 & x & \cdots & x^{n-1} & x^n \\ 1 & a_{i+1} & \cdots & a_{i+1}^{n-1} & a_{i+1}^n \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & a_{n+1} & \cdots & a_{n+1}^{n-1} & a_{n+1}^n \end{vmatrix}$，这也是一个 Vandermode 行列式，于是有式(6)：

$$\sum_{j=1}^{n+1}\frac{A_{ij}x^{j-1}}{D}=\frac{(x-a_1)(x-a_2)...(x-a_{i-1})(x-a_{i+1})...(x-a_{n+1})}{(a_i-a_1)(a_i-a_2)...(a_i-a_{i-1})(a_i-a_{i+1})...(a_i-a_{n+1})} \tag{6}$$

将式(6)代入式(5)可得 $f(x)=\sum_{i=1}^{n+1}\frac{b_i(x-a_1)...(x-a_{i-1})(x-a_{i+1})...(x-a_{n+1})}{(a_i-a_1)...(a_i-a_{i-1})(a_i-a_{i+1})...(a_i-a_{n+1})}$，进而得到拉格朗日插值公式。

而在资源拥有者的多策略访问树中，非叶子节点所定义的多项式的次数由该节点的门限值 k_x 确定，进而在数域 F 中存在唯一次数不超过 k_x 的多项式，又根据式(3)非叶子节点的孩子节点的秘密值 *secret* 和索引值 *index* 满足等式 *secret* = *f(index)*。因此，将 k_x 个资源请求者的有效属性值与资源拥有者的多策略访问树的叶子节点匹配，确定 k_x 个叶子节点对应的秘密值 *secret*，结合该叶子节点在非叶子节点下的索引值 *index*，根据式(4)~(6)分别代入对应值，可以确定一个唯一已知的非叶子节点的多项式函数：

$$f(x)=\sum_{i=1}^{n+1}\frac{secret_i(x-index_1)...(x-index_{i-1})...(x-index_{n+1})}{(index_i-index_{11})...(index_i-index_{i-1})...(index_i-index_{n+1})}$$

(注：该函数右表达式已去除因子 $\frac{(x-index_i)}{(index_i-index_i)}$)。

3)秘密恢复安全性证明

由于非叶子节点的多项式函数自变量为 0 时，$f(x)$ 的计算结果 $f(0)$ 即为当前非叶子节点的秘密值。因此，只有资源请求者的属性值包含在当前非叶子节点的属性集中，且满足门限值 k_x 时，才可以恢复当前非叶子节点的秘密值。则有，资源请求者的属性若未满足资源拥有者多策略访问树非叶子节点的门限值，则无法通过本方案获取该节点的秘密值，从而也无法获得访问权限。

推论 1　基于多策略访问树的访问控制具有安全性。

1)多策略访问树的多项式存在性和唯一性证明

由上述多项式存在性和唯一性证明可得，当 $k_x>n$，资源请求者的属性值个数小于时，在数域 F 上存在的唯一的多项式 $f(x)=c_0+c_1x+c_2x^2+\cdots+c_nx^n$ 的最高次将小于门限值减 1。

2)多策略访问树的拉格朗日插值多项式正确性证明

由上述拉格朗日值多项式正确性证明可得，若 $k_x>n$，则无法生成当前非叶子节点预先定义的 k_x-1 阶多项式 $f(x)$。此时由 n 个叶子节点构造的多项式 $f(x)^*$，且 $f(x)^*\neq f(x)$。

3)多策略访问树的秘密恢复安全性证明

由上述秘密恢复安全性证明可得，若 $f(x)^*\neq f(x)$，则 $f(0)^*\neq f(0)$。因此，若资源请求者的属性未满足资源拥有者多策略访问树非叶子节点的门限值($k_x>n$)，则无法通过拉格朗日插值多项式获取该节点的秘密值。

综上所述，若资源请求者的属性集不满足多策略访问树预定义的属性集和门限值，则无法获取资源拥有者的资源。因此，本方案在访问控制上是安全的。

3.2.2　车辆身份匿名性分析

车辆在消息传递过程中使用假名与车联网系统中的其他实体通信，保证车辆身份的匿名性以及不可追踪性。在本方

案中，只有 TA 可以根据身份映射表通过车辆的伪身份 $ID_i^{'}$ 还原其真实身份 ID_i，RSU 可以根据伪身份映射表通过车辆的临时假名 F_i 还原其伪身份 $ID_i^{'}$。在车辆进入 RSU 域内与 RSU 双向认证阶段，车辆使用伪身份 $ID_i^{'}=ID_i\oplus H_0(ID_i\|n_1)$ 与 RSU 进行通信，RSU 为其生成临时假名 $F_i=H_3(ID_i^{'}\|n_2)$ 用于与其他车辆的通信，由于真实身份 ID_i 仅有车辆 V_i 持有且只有 TA 可通过伪身份 $ID_i^{'}$ 进行还原，且 n_1 与 n_2 均为身份生成时的随机数，任何攻击者若想要从临时假名 F_i 还原其伪身份 $ID_i^{'}$，则需要计算 n_2 恰好满足 $F_i=H_3(ID_i^{'}\|n_2)$，若想要从伪身份 $ID_i^{'}$ 还原其真实身份 ID_i，则需要计算 n_1 恰好满足 $ID_i=ID_i^{'}\oplus H_0(ID_i\|n_1)$，然而基于安全散列函数抗碰撞的特性，攻击者难以计算出满足要求的 n_1 与 n_2。因此，方案提供了车辆对其他实体的匿名性保护。

3.2.3　消息不可否认性分析

车辆在申报开放资源和资源请求的过程中，可对恶意车辆发送的恶意请求进行溯源，揭露其真实身份 ID_i 以达到消息的不可否认性。若恶意车辆 V_i 在向 RSU 请求资源时或利用请求到的资源进行恶意操作。RSU 根据其临时假名 F_i 通过伪身份映射表查找出其伪身份 $ID_i^{'}$，将该伪身份 $ID_i^{'}$ 上报至 TA，TA 根据伪身份 $ID_i^{'}$ 通过身份映射表查找出其真实身份 ID_i。完成对恶意车辆 V_i 的真实身份 ID_i 溯源，因此该方案具有消息不可否认性。

3.3　性能分析

3.3.1　存储开销分析

将本方案与文献[15，16]的存储开销对比，参数定义如下：p 表示 Z_p 中数据元素的长度，k 表示访问单个多策略树的策略数，num_{attr} 表示系统中属性的总个数，num_{tree} 表示系统中多策略树的个数，num_u 表示资源请求者 u 拥有的属性个数。

从表 6 可知，PES 在系统中主要用于存储访问策略和属性。由于本文方案利用多策略访问树存储策略和属性，相较于其他方案，本方案的数据存储实体具有较明显优势。而资源请求者和资源拥有者的总存储开销，由于密文加解密方式和属性基本一致，所以并没有太大差距。本文方案更适用于复杂的车联网环境。

表 6　存储成本比较

Tab. 6　Storage cost comparison

数据存储实体	文献[15]	文献[16]	本方案
PES	$(num_{attr}+knum_{tree})p$	$knum_{tree}p$	$num_{tree}p$
资源请求者	$(3+num_u)p$	num_up	$(3+num_u)p$
资源拥有者	p	$3num_up$	p

3.3.2　计算开销分析

将本文方案与文献[17~20]的方案进行性能比较。

仿真实验环境为：开发套件使用 JPBC 密码库，硬件平台 CPU 为 AMD Ryzen 5 5600H ，内存为 16GB。文献[17~20]的方案通过在群 G 上进行双线性运算对实体的属性值加解密，设 l 为实体的属性个数，根据文献[18]方案的方法，得到如表 7 的双线性密码学运算的执行时间。

表 7　各类运算的执行时间　　单位:ms

Tab. 7　Execution time of different operations　　unit:ms

运算类型	含义	执行时间
T_p	在群 G 上执行一次双线性对运算	13.597
$T_{G,mul}$	在群 G 上执行一次标量乘运算	21.434
$T_{GT,mul}$	在群 G_T 上执行一次标量乘运算	1.660

分析计算文献[17~20]的方案在策略匹配计算开销，如表 8 所示。

表 8　计算开销

Tab. 8　Computational overhead

方案	策略匹配开销
文献[17]	$(3l+2)T_p+2l\cdot T_{GT,mul}+T_{G,mul}=44.111l+48.628$
文献[18]	$(3l+1)T_p+l\cdot T_{GT,mul}=42.451l+13.597$
文献[19]	$2l\cdot T_p+l\cdot T_{GT,mul}=28.854l$
文献[20]	$(3l+1)T_p+l\cdot T_{GT,mul}+T_{G,mul}=42.451l+35.031$

分析本方案计算开销：由于本方案采用非双线性运算，极大地降低了运算的复杂性，计算开销则主要体现在对多策略访问树节点地秘密值生成和策略匹配的运算。本实验性能仿真测试过程中，因单次访问策略匹配计算所需时间值较小，造成不同样本数据相对误差较大，因此仿真实验通过求 100 次的平均值来减小相对误差。仿真实验的多策略访问树只包含 1 个策略时，令属性个数分别为 5，10，15，20，进行性能测试。然后，固定属性个数为 20 个时，令多策略访问树的策略数目分别为 4，8，12，16，进行性能测试。以上均采用运行多次计算平均值的方法，用于消除由于环境差异造成的数据误差。将资源请求者的属性个数作为横坐标，时间开销作为纵坐标。方案的策略数均设为 1 个。根据本次仿真数据结果，分析其与其他方案的执行策略匹配计算开销对比，结果如图 7 所示。

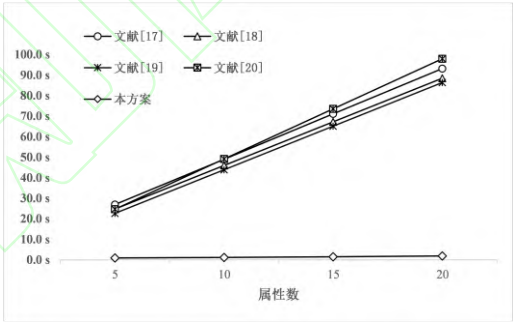

图 7　策略匹配计算开销对比图一

Fig. 7　Comparison of policy matching computation

由图 7 可知，由于本方案需要为多策略访问树的每个节点计算秘密值以及为与多策略访问树匹配的有效属性值完成策略匹配，且在恢复秘密值并完成策略匹配的过程中，有效属性值所对应的叶子节点个数的增加，将导致非叶子节点多项式的拉格朗日因子增加。因此，随着属性个数的增加，策略匹配的计算开销也会递增。但当访问控制的策略唯一，随着属性数逐渐递增，由于本方案并未涉及双线性运算，计算开销极大的减少，因此，本方案执行 100 次策略匹配的计算开销明显小于其他涉及双线性运算的方案。

图 8 将策略个数作为横坐标，时间开销作为纵坐标，各方案的属性数设为 20 个。根据本次仿真数据结果，分析其与其他方案的执行策略匹配计算开销对比结果，如图 8 所示。

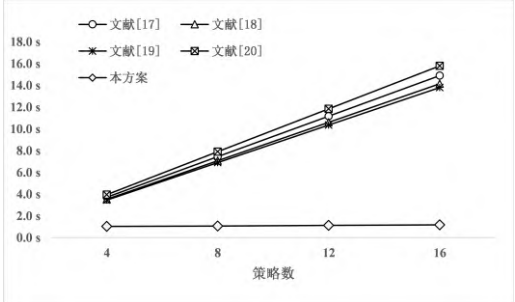

图 8　策略匹配计算开销对比图二

Fig. 8　Comparison of policy matching computation

本方案在多策略访问树上进行策略匹配时，将其所有非叶子节点都作为策略，若与多策略访问树匹配的有效属性值的个数相同，则须计算的非叶子节点多项式的拉格朗日插值因子个数也大致相同。因此，相同属性个数时，访问控制策略的增加对于策略匹配的计算开销是可以忽略的。如图 8 所示，本方案策略匹配的计算开销随着策略数的增加变化趋势几乎接近于水平。而当属性数相同，访问控制的策略递增时，在策略数较少的情况下，由于本方案策略匹配过程中并未涉及双线性运算，计算开销相较于其他方案具有相对的优势。随着策略数的增加，本方案中所使用的多策略访问树在同一访问树上完成多次的策略匹配，而其余方案需要多颗访问树来完成多次策略匹配，因此本方案优势更加明显。

综上，本方案策略执行节点的存储开销明显低于其他方案。同时，由于本方案在构造多策略访问树以及策略匹配时，并未使用较为复杂的双线性运算，因此在同属性值数目或同策略数目条件下，本方案的策略匹配计算开销明显小于其他方案。在策略数较多的情况下，本方案的策略匹配计算开销也明显小于其余方案。且本方案创新性的多策略访问树数据据结构可以实现细颗粒度的访问控制，故方案适合在资源访问者较多、策略数量大、通信时延较小的车联网环境下使用。

4　结束语

为了解决车联网环境下传统资源访问控制效率低下和安全问题，本文提出一种基于多策略访问树的资源访问控制方案。方案将传统的单策略访问树改进为多策略访问树，实现多级策略制定，有效提高策略访问效率，并实现了对资源的细粒度访问。同时，方案通过引入拉格朗日多项式秘密值恢复机制，将秘密值与授权令牌结合，解决了车联网访问控制授权的安全性问题。本方案被证明在安全上具有身份的匿名性和消息不可否认性；同时，性能分析的结果表明，本方案与同类方案相比，在策略执行节点的存储开销明显低于其他方案；策略匹配效率显著提高，计算开销明显变小。本方案适用于在 VANET 环境中高效的资源访问控制。

参考文献：

[1] 邓雨康, 张磊, 李晶. 车联网隐私保护研究综述 [J]. 计算机应用研究, 2022, 39 (10): 2891-2906. (Deng Yukang, Zhang Lei, Li Jing. Overview of privacy protection research on the Internet of Vehicles [J]. Computer Application Research, 2022, 39 (10): 2891-2906)

[2] Graham G S, Denning P J. Protection-Principles and practice [J]. ACM, 2017.

[3] Electronic N A, Lapadula L, Bell D E, *et al.* Secure Computer Systems: Mathematical Foundations. MITRE Corp, 1973.

[4] Wang G H. Role-based access control models [J]. Computer, 1996, 29 (2): 38-47.

[5] Yuan E, Tong J. Attributed based access control (ABAC) for web services [C]// IEEE International Conference on Web Services (ICWS'05) , Orlando, IEEE Press, 2005, 569.

[6] 董国芳, 鲁烨堃, 张楚雯等. 支持撤销属性的 CP-ABE 密钥更新方法 [J]. 计算机应用研究, 2023, 40 (02): 583-588. (Dong Guofang, Lu Yekun, Zhang Chuwen *et al.* CP-ABE key update method supporting revocation attribute [J]. Application Research of Computers, 2023, 40 (02): 583-588.)

[7] Huang D, Verma M. ASPE: attribute-based secure policy enforcement in vehicular ad hoc networks [J]. Ad Hoc Networks, 2009, 7 (8): 1526-1535.

[8] Liu X, Xia Y, Chen W, *et al.* SEMD: Secure and efficient message dissemination with policy enforcement in VANET [J]. Journal of Computer & System Sciences, 2016: 1316-1328.

[9] 庞立君, 刘南杰, 赵海涛, 等. 基于 CP-ABE 的车联网云端数据安全访问控制方案 [J]. 电视技术, 2015, 39 (18): 6. (Pang Lijun, Liu Nanjie, Zhao Haitao, *et al.* Secure access control scheme for cloud data in Internet of vehicles based on CP-ABE [J]. Television Technology, 2015, 39 (18): 6.)

[10] 刘雪娇, 殷一丹, 陈蔚, 等. 基于区块链的车联网数据安全共享方案 [J]. 浙江大学学报: 工学版, 2021. (Liu Xuejiao, Yin Yidan, Chen Wei, *et al.* Secure data sharing scheme of Internet of Vehicles based on blockchain [J]. Journal of Zhejiang University: Engineering Science Edition, 2021.)

[11] 侯慧莹, 廉欢欢, 赵运磊. 面向自动驾驶的高效可追踪的车联网匿名通信方案 [J]. 计算机研究与发展, 2022 (004): 059. (Hou Huiying, Lian Huanhuan, Zhao Yunlei. Efficient and traceable anonymous communication scheme of Internet of Vehicles for autonomous driving [J]. Journal of Computer Research and Development, 2022 (004): 059.)

[12] Pan J, Cui J, Wei L, *et al.* Secure data sharing scheme for VANETs based on edge computing [J]. EURASIP Journal on Wireless Communications and Networking, 2019 (1): 1–11.

[13] Kanumalli S S, Ch A, Murty P S R C. Secure V2V Communication in IOV using IBE and PKI based hybrid approach [J]. International Journal of Advanced Computer Science and Applications, 2020, 11 (1) .

[14] Ma J, Li T, Cui J, *et al.* Attribute-Based secure announcement sharing among vehicles using blockchain [J]. IEEE Internet of Things Journal, 2021, 8 (13): 10873-10883.

[15] 李玲. 雾计算中基于属性加密的可撤销访问控制研究 [D]. 太原: 太原理工大学, 2021.

[16] Fan K, Xu H, Gao L, *et al.* Efficient and privacy preserving access control scheme for fog-enabled IoT [J]. Future Generation Computer Systems, 2019.

[17] 仲红, 崔杰, 朱文龙, 等. 高效且可验证的多授权机构属性基加密方案 [J]. 软件学报, 2018, 29 (7): 12. (Zhong Hong, Cui Jie, Zhu Wenlong, *et al.* Efficient and verifiable multi-authority attribute-based encryption scheme [J]. Journal of Software, 2018, 29 (7): 12.)

[18] 吴静雯, 殷新春, 宁建廷. 车载自组网中可追踪可撤销的多授权中心属性基加密方案 [J]. 计算机应用, 2022, 42 (6): 7. (Wu Jingwen, Yin Xinchun, Ning Jianting. Traceable and revocable multi-authority attribute-based encryption scheme in Vehicular AD Hoc networks [J]. Computer Applications, 2022, 42 (6): 7.)

[19] 明洋, 何宝康. 支持属性撤销的可验证外包的多授权属性加密方案 [J]. 计算机应用, 2019, 39 (12): 7. (Ming Yang, He Baokang. Verifiable outsourcing multi-authority attribute-based encryption scheme supporting attribute revocation [J]. Computer Applications, 2019, 39 (12): 7.)

[20] Yang Y, Chen X, Chen H, *et al.* Improving Privacy and Security in Decentralizing Multi-Authority Attribute-Based Encryption in Cloud Computing [J]. IEEE Access, 2018: 18009-18021.