计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(2):117−124 [doi: 10.15888/j.cnki.csa.007768]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

车载自组网中基于属性的可搜索加密和属性更新①

张露露, 光笑黎, 刘继增

(长安大学 信息工程学院，西安 710064)
通讯作者: 张露露, E-mail: 2960385828@qq.com

摘　要: 目前, 车载自组网 (VANET) 在汽车行业和研究领域都得到了极大的关注, 尤其是在用户的隐私保护方面. 雾计算是云计算的延伸, 它能够有效的减小网络延迟, 其反应性更强. 相较云计算, 使用雾计算减少了发送到云端和从云端发送的数据量, 安全风险也得到了进一步的降低. 由于基于密文策略的加密 (CP-ABE) 适用于存储在云上的数据的细粒度的访问控制以及基于关键字的可搜索加密可以使用户快速查找存储在云服务器上的感兴趣数据和不泄露任何搜索关键字的信息. 因此, 本文提出了基于属性的可搜索加密和属性更新, 它是将基于属性的加密方案和关键字搜索加密方案相结合. 该方案支持用户属性更新, 不合法车辆用户不会对存储的数据有访问权限, 从而实现对不合法车辆用户的撤销. 同时它也实现了车-雾-云三者之间的通信, 在通信过程中其将部分加密和解密计算外包给雾节点, 减少了用户的计算代价. 此外, 通过性能分析表明了所提方案在功能性和计算复杂度两方面都具有较好的优势.

关键词: 车载自组网; 雾计算; CP-ABE; 可搜索加密; 属性更新

引用格式: 张露露,光笑黎,刘继增.车载自组网中基于属性的可搜索加密和属性更新.计算机系统应用,2021,30(2):117–124. http://www.c-s-a.org.cn/1003-3254/7768.html

Attribute-Based Searchable Encryption Scheme and Attribute Update in VANET

ZHANG Lu-Lu, GUANG Xiao-Li, LIU Ji-Zeng

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: At present, Vehicle Ad hoc NETworks (VANETs) have received great attention in the automotive industry and research areas, especially in terms of users' privacy protection. As an extension of cloud computing, fog computing is more reactive since it can effectively reduce network latency. Compared with cloud computing, fog computing decreases the volume of data sent to and from the cloud and further lowers the security risk. Since Ciphertext Policy Attribute-Based Encryption (CP-ABE) is suitable for fine-grained access control of data stored on the cloud and searchable encryption based on keywords, users can quickly find interesting data stored on cloud servers and not leak information about any search keywords. For this reason, we propose attribute-based searchable encryption and attribute update in this study, which is a combination of attribute-based encryption schemes and keyword-search encryption schemes. The proposed scheme supports user attribute update, and illegal vehicle users will not have access to the stored data, thereby realizing the cancellation of illegal vehicle users. At the same time, it also achieves the communication among the vehicles, fog, and cloud. In the communication process, it outsources part of the encryption and decryption calculations to the fog nodes, reducing the users' calculation cost. In addition, performance analysis shows that the proposed scheme has better advantages in both functionality and computational overhead.

① 收稿时间: 2020-05-30; 修改时间: 2020-06-23, 2020-07-15; 采用时间: 2020-07-17; csa 在线出版时间: 2021-01-27

Key words: Vehicle Ad hoc NETwork (VANET); fog computing; CP-ABE; searchable encryption; attribute update

1 引言

车载自组网 (VANET) 是指在交通环境中车辆之间, 车辆与固定接入点之间及车辆与行人之间相互通信形成的开放式移动网络. VANET 的架构已被拓展到更广泛的范畴, 分为车内通信 (in-vehicle domain)、车间通信 (Ad-hoc domain) 和车路通信 (Infrastructure domain) 3 个域. 车内通信是车载单元 (OBU) 与用户终端之间的通信, 用户终端可以是某种具体的设备. 车间通信包括 OBU 之间的通信 (V2V) 以及 OBU 与路侧单元 (RSU) 间的通信 (V2R). 车路通信是 OBU, RSU 以及基础设施之间的通信. 然而, 车辆与 RSU 采用的通信方式是专用短距离通信 (Dedicated Short Range Communication), 车辆与车辆之间的通信采用的是 802.11p 通信协议[1]. VANET 是实现智能交通系统 (ITS) 的一种方式, 是将信息和通信技术传授给交通基础设施和车辆的一种技术. 车辆与车辆和车辆与路边基础设施之间可以数据交换, 用这些警告信息来提高乘客的安全, 可以减少交通事故的发生, 改善交通管理. 通过物联网的通信、数据处理等技术在交通领域的支撑性映射, 来提供更好的服务. 一般, 我们都是将大量数据存储在云服务器上, 但是, 云服务器本身的特点会造成网络延迟. 为了更好的解决问题, 美国思科公司的 Bonomi 等在 2012 年首次提出雾计算的概念[2]. 与云计算相比, 它能够过滤信息, 当聚合用户信息时, 只需要将核心信息发送给云, 减少核心网络压力. 而且, 它比云拥有更小的网络延迟, 能够及时的接收信息. 此外, 雾服务器设置在云服务器和物联网设备,以便存储和计算的数据传输尽可能多的雾服务器. 因此, 雾计算有助于减少云服务器的工作负载, 提高整个系统的效率. 由于, 我们往往将注意力放在服务器的存储和计算资源上, 在这样情况下, 如何保证数据的安全, 成为了我们需要关注的焦点. Song 等[3] 提出了可搜索加密, 在该方法中数据所有者将其加密的数据存储在第三个存储服务器中, 允许用户使用适当的关键字搜索加密数据, 并获得所需的加密数据. 此方法, 很好的解决了数据的安全性问题. 虽然我们将大量数据存储在云服务上, 但是它并不是完全可信的. 为了保证存储在它上面的数据的隐私性和对数据的细粒度的访问, Sahai 等[4] 首次提出基于属性加密 (Attribute-Based Encryption, ABE)机制. 接着, Goyal 和 Bethencourt 等[5,6] 提出了基于密钥策略的属性加密方案 (KP-ABE) 和基于密文策略的属性加密方案 (CP-ABE), 两方案均是属性集合满足访问策略时, 用户才可以解密. 然而, 在 2009 年 Huang等[7] 首次针对车联网中数据访问控制提出了一种新的基于属性的安全策略实施方案, 该方案将基于密文策略属性加密方案进行优化, 减少了加密和解密时间. 2011 年, Yeh 等[8] 针对车联网中非安全类应用紧急服务提出基于属性访问控制方案, 该方案能够实现数据机密性和细粒度访问控制. 2013 年, 针对属性加密密文问题, Rao 等[9] 使用析取范式访问策略设计车联网中基于属性访问控制方案, 方案中密文长度与属性数无关, 降低了通信开销. 2014 年, Yeh 等[10]提出面向服务的车联网细粒度访问控制便携式计费方案, 该方案利用基于密钥策略属性加密实现复杂的访问控制策略, 并确保实体的真实性和电子硬币的有效性. 2016 年, Bouabdellah 等[11] 提出车联网中基于属性加密安全协作传输模型, 该模型利用基于密文策略属性加密来实现数据的机密性和细粒度访问控制. Xia 等[12] 提出车联网中隐私保护的自适应多媒体数据转发方案, 该方案利用基于密文策略属性加密将解密外包给路侧单元, 提高车辆的解密效率. 大量的数据存储在云中, 很难满足我们现在的需求, 也会给我们带来一些不便. 为了减轻用户的计算负担, 有学者将雾计算和云计算相结合, 由于雾计算有计算能力, 因此, 可以将部分计算外包给雾, 从而减轻用户的计算代价. 在 2018 年, Xue 等[13] 提出车辆云计算中延迟敏感数据共享的雾辅助可验证隐私保护访问控制方案, 该方案利用基于密文策略属性加密将数据加密和解密外包给云和雾节点, 减少了延迟和计算开销. 为了在车-雾-云环境中实现用户撤销, 计算外包, 可搜索加密, 本文提出了一个 VANET 中基于属性的可搜索加密和属性更新方案. 本文的主要工作是:

(1) 在车-雾-云环境下, 本文设计了一个可搜索加密方案, 将雾节点作为车辆和云服务器之间的桥梁, 数据拥有者和车辆用户都可以直接与雾节点连接, 每个雾节点都与云服务器连接, 减少了数据不必要的传输.

(2) 本文提出的基于属性的可搜索加密方案, 是一

个一对多的通信. 车辆用户可以根据关键字来询问相关密文, 不泄露相关信息.

(3) 文章提出的 CP-ABE 方案, 将雾-云相结合, 部分加密和解密外包给雾计算, 减少了用户的计算负担, 减少了网络延迟.

2 基础知识

2.1 访问结构

令 $P=\{P_1,P_2,\cdots,P_n\}$ 是参与者集合, 存取结构 A 是 2^P 的一个非空子集, 即 $A\subseteq 2^P\backslash\{\emptyset\}$. 对任意集合 B、C, 如果 $B\in A$ 且 $B\subseteq C$, 有 $C\in A$, 则称存取结构 A 是单调的. 存取结构 A 中的集合称为授权集合, 否则称为非授权集合.

2.2 双线性映射

令 G 为循环乘法群, 其生成元为 g, 阶为素数 p. G_T是阶同为 p 的循环乘法群. 假设 e: $G\times G\to G_T$为双线性映射, 也称双线性对, 则其满足下面的性质:

(1) 双线性性: 对于任意 $u,v\in G$, $a,b\in Z_p$, 恒有 $e\left(u^a,v^b\right)=e(u,v)^{ab}$.

(2) 非退化性: 存在 $g\in G$, 使得 $e(g,g)\neq 1$.

(3) 可计算性: 对于任意 $u,v\in G$, 存在有效的算法计算 $e(u,v)$.

2.3 访问树

本文中, 我们用访问树作为访问策略.

T 代表访问树, x 代表它的节点, 其中 T 的非叶节点 x 代表一个阈值门, num_x表示 x 的子节点数, 则 $0<k_x\leq num_x$. 其中, 当 $k_x=1$时, 阈值为“或门”; 当 $k_x=num_x$时, 阈值为“与门”, 每个叶节点与属性相关. 为了方便, 我们定义了以下公式:

(1) $parent(x)$: 代表除根节点外返回节点的父节点.

(2) $index(x)$: 代表假定每个节点的子节点被标记为 1 到 num, 这里返回的是与节点 x 相关的数字.

(3) $att(x)$: 代表一个叶节点相关的属性.

2.4 困难问题及安全假设

问题 1. 判定性双线性 Diffie-Hellman (简称 DBDH) 问题:

令 G、G_T 是阶为素数 p 的群, g 是 G 的生成元, e: $G\times G\to G_T$ 是双线性映射. 给定 $g,g^a,g^b,g^c\in G$ 以及 $Z\in G_T$, 其中 $a,b,c\in Z_p^*$为随机选取的未知数, 判断 $Z=e(g,g)^{abc}$ 是否成立.

定义算法A_{DBDH}解 DBDH 问题的优势为:

$$Adv_{\mathrm{DBDH}}=\left|Pr\left[A_{\mathrm{DBDH}}\left(g,g^a,g^b,g^c,e(g,g)^{abc}\right)\right]-Pr\left[A_{\mathrm{DBDH}}\left(g,g^a,g^b,g^c,Z\right)=1\right]\right|$$

假设 1. 对于任意多项式时间算法A_{DBDH}, 其解 DBDH 问题的优势是可忽略的.

3 系统和安全模型

3.1 系统模型

本文设计的方案由以下 7 部分组成:

云服务器 (CSP): CSP 作为可信的实体, 提供多种服务, 比如: 数据存储. 当收到 VU 的搜索陷门时, CSP 提供关键字搜索. 当对车辆用户进行撤销时, CSP 可以通过更新用户属性来更新密文, 实现用户撤销.

雾节点 (Fog): Fog 作为可信的实体, 提供多种服务, 比如: 低延迟, 存储. Fog 可以产生部分密文并发送给用户, 最终上传给 CSP. 它也可以解密从 CSP 上获得的部分密文.

数据拥有者 (DO): DO 定义访问结构并产生部分密文和关键字索引, 最后将完整密文和索引发送给 Fog, 最终存储在 CSP.

车辆用户 (VU): VU 可以产生搜索的陷门并发送给 CSP, CSP 找到包含此关键字的密文, 然后将其发送给 Fog 进行部分解密. 最后, VU 得到部分解密密文, 并解密.

数据拥有者 (DO): DO 定义访问结构并产生部分密文和关键字索引, 最后将完整密文和索引发送给 Fog, 最终存储在 CSP.

车辆用户 (VU): VU 可以产生搜索的陷门并发送给 CSP, CSP 找到包含此关键字的密文, 然后将其发送给 Fog 进行部分解密. 最后, VU 得到部分解密密文, 并解密.

属性权威 (AA): AA 作为可信的实体, 负责生产系统参数, 属性管理和密钥生成. 当 VU 的属性需要更新时, AA 为 VU 产生更新密钥.

系统模型如图 1 所示.

3.2 方案定义

我们提出的 VANET 中基于属性的可搜索加密和属性更新方案具体包含以下 8 个步骤:

(1) 算法输入安全参数λ和属性集合 L, 输出系统参数 PP, 主密钥 MSK, CSP 的公私钥对 (PK_s, SK_s).

(2) 密钥生成

$KeyGen(MSK,S)\to(SK,A_{\mathrm{pri}},A_{\mathrm{pub}})$: 算法输入主密钥 MSK 和用户的属性集合 S, 输出用户的私钥 SK, 用

户的公私钥对$(A_{\text{pri}}, A_{\text{pub}})$.

(3) 数据加密和生成关键字索引

DU 用对称加密算法的对称密钥 ck 对进行消息 M 加密, 生成$E_{ck}(M)$, 然后用加密算法对 ck 加密.

Fog 执行 $Encrypt(PP, ck, T) \rightarrow (CT_1)$: 算法输入系统参数 PP, 对称密钥 ck, 访问结构 T, 输出部分密文 CT_1.

DO 执行 $Encrypt(PP, CT_1) \rightarrow (CT)$: 算法输入系统参数和部分密文$CT_1$, 输出密文 CT.

DO 执行 $Index(W, A_{\text{pub}}) \rightarrow (I_W)$: 算法输入关键字集合 W, 数据拥有者的搜索公钥A_{pub}, 输出关键字索引I_W.

(4) Trapdoor 生成

$TrapdoorGen(w, PK_s, A_{\text{pri}}) \rightarrow T_w$: 算法输入关键字 w, CSP 的公钥PK_s, DV 搜索私钥A_{pri}, 输出T_w.

(5) 文件检索

CSP 执行 $Test(I_W, T_w) \rightarrow (0,1)$: 算法输入关键字索引集合$I_W$, DV 的搜索陷门$T_w$, 输出$(0,1)$.

(6) 数据预解密

Fog 执行 $PDecrypt(CT, SK') \rightarrow CT_1$: 算法输入密文 CT, 密钥SK', 输出部分密文CT_1.

(7) 车辆用户解密

DV 执行 $Decrypt(CT_1, SK) \rightarrow ck$: 算法输入密文 CT_1, 密钥 SK, 输出 ck.

最后, 车辆用户解密$E_{ck}(M)$, 得到 M.

(8) 属性更新

假设用户的一个属性a_j通过 AA 被更新为a_j'. 属性更新算法包括以下几步:

$UKeyGen(PP, MSK, SK, a_j, a_j') \rightarrow (UK, UK')$: 算法输入系统参数 PP, 主密钥 MSK, 用户密钥 SK, 属性 a_j, a_j', 输出更新算法UK, UK'.

$SKUpdate(SK, UK, UK') \rightarrow SK_u$: 算法输入密钥 SK, 更新算法UK, UK', 输出更新的密钥SK_u.

3.3 安全模型

在我们的方案中, 雾节点和云服务器都是诚实可信的, 意味着它们可以执行操作并可以抵制获得非授权数据. 安全模型包括以下几个方面.

(1) 细粒度访问控制: 数据拥有者可以灵活的定义访问结构, 只有用户的属性集合满足定义的访问结构时, 用户才可以访问和更新数据.

(2) 抗共谋: 两个或多个用户不能结合来获得他们的私钥.

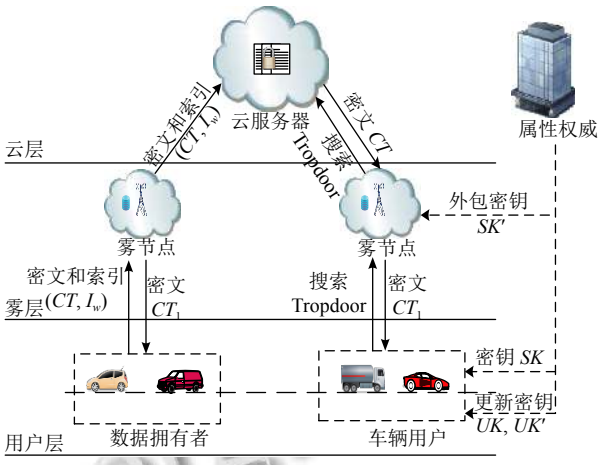

图 1 系统模型

4 方案构造

(1) 系统初始化

AA Setup: AA 输入安全参数λ和属性集合$L = \{a_1, a_2, \cdots, a_m\}$. 它选择一个双线性对$e: G_0 \times G_0 \rightarrow G_T$, 其中$G_0$是阶为 p 双线性群, g 是它的生成元. AA 随机选择两个元素$\alpha, \beta \in Z_p$, 也选择$h \in G_0$, 随机选择$x_s \in Z_p$, 让$SK_s = x_s, PK_s = g^{x_s}$作为云的密钥对, 让$A_{\text{pri}} = u$作为用户搜索私钥, $A_{\text{pub}} = g^u$作为用户搜索公钥, 对于每个属性$a_j \in L$, 它选择一个随机v_j并计算$PK_j = g^{v_j}$. 最后输出公钥和主密钥.

$$PP = \left\{G_0, g, h, g^{\alpha}, g^{\beta}, h^{\beta}, e(g,g)^{\alpha\beta}, \left\{PK_j = g^{v_j} \mid a_j \in L\right\}\right\}$$

(2) 密钥生成

AA 运行算法, 选择$r, \varepsilon \in Z_p$, 生成用户私钥$SK = D = g^{\beta(\alpha+r)}$ 和外包密钥.

$$SK' = \left\{D' = g^r h^{\varepsilon}, D'' = g^{\varepsilon}, \left\{D_j = g^{\frac{r\beta}{v_j}} \mid \forall a_j \in S\right\}\right\}$$

AA 将用户的外包密钥发送给 Fog.

(3) 加密阶段和生成关键字索引

① 加密阶段

数据上传到云之前, 数据拥有者随机选择一个 ck 作为对称密钥数据 M 进行加密, 生成$E_{ck}(M)$, 数据拥有者定义访问结构 T 并发给 Fog.

Fog 加密: Fog 首先从根节点 R 开始自上而下的为每个节点 x 选择一个多项式q_x, 多项式的次数取为$d_x = k_x - 1$, 其中k_x为节点 x 的门限值. num_x表示 x 的子节点数, 则$0 \le k_x \le num_x$.

从根节点 R 开始, Fog 随机选择$s_1 \in Z_p$并有$q_R(0) =$

s_1, 对于其他节点 x, 设置 $q_x(0)=q_{parent(x)}(index(x))$, 在访问树中, X 是叶节点相关的属性集合, 生成的部分密文为:

$$CT_1=\left\{T,C_1=g^{s_1\beta},C_2=h^{\beta s_1},\left\{C_j=g^{v_j q_x(0)}\mid\forall a_j=att(x)\in X\right\}\right\}$$

Fog 将 CT_1 发送给数据拥有者.

数据拥有者随机选择 $s_2\in Z_p$, 生成最终的密文 CT:

$$CT=\left\{T,E_{ck}(M),\widetilde{C}=ck\cdot e(g,g)^{\alpha\beta s_2},C=g^{s_2},C'=g^{s_1\beta}g^{\beta s_2},C''=h^{s_1\beta}h^{\beta s_2},\left\{C_j=g^{v_j q_x(0)}\mid\forall a_j=att(x)\in X\right\}\right\}$$

② 生成关键字索引

数据拥有者从文件中提取关键字集合 $W=\{w_1,w_2,\cdots,w_n\}$, 执行索引生成算法

输入关键字和用户搜索公钥, 索引生成算法为每个关键字随机选择 $\zeta_i\in Z_p$ 并计算 $\tau_i=e\left(\left(A_{\text{pub}}\right)^{\xi_i},H_1(w_i)\right)$, 计算 $I_{w_i}=[I_1,I_2]=\left[g^{\xi_i},H_2(\tau_i)\right]$ 最后输出关键字索引的集合 $I_w=\left\{I_{w_i}\right\}_{i\in\{1,2,\cdots,n\}}$. (CT,I_w) 由数据拥有者经过雾节点存储到云上.

(4) Trapdoor 生成

输入关键字, 云的公钥和用户私钥, 算法随机选择 $\sigma\in Z_p$ 并计算 $T_1=g^{\sigma},T_2=(PK_s)^{\sigma}\cdot H_1(w)^{A_{\text{pri}}}$. 输出 Trapdoor T_w.

$$T_w=\{T_1,T_2\}=\left\{g^{\sigma},(PK_s)^{\sigma}\cdot H_1(w)^{A_{\text{pri}}}\right\}=\{g^{\sigma},(g^{x_s})^{\sigma}\cdot H_1(w)^{u}\}$$

(5) 文件检索

输入关键字索引和用户搜索 Trapdoor, 云收到用户的搜索请求后, 首先要检查用户的属性是否满足访问结构. 如果是, 云检查 T_w 是否和 I_w 匹配.

云计算:

$$\theta=\frac{T_2}{(T_1)^{x_s}}=\frac{(g^{x_s})^{\sigma}\cdot H_1(w)^{u}}{(g^{\sigma})^{x_s}}=H_1(w)^{u}$$

再根据关键字索引计算:

$$\theta_1=e(I_1,\theta)=e((g^{u})^{\sigma},H_1(w))$$

判断 $H_2(\theta_1)=I_2$, 如果成立, 说明匹配成功.

(6) 数据预解密

Fog 首先从云上获得关键字索引相关的密文, 用部分密钥解密, 进行如下操作:

如果用户的属性集合满足定义的访问策略, 那么用户就能解密. 算法输入密文 CT, 部分密钥 SK' 和一个节点 x.

① 如果节点 x 是叶节点, 让 $a_j=att(x)$, 如果 $a_j\in S$, 则:

$$\begin{aligned}F_x&=\prod_{n\in S_x}F_n^{\nabla_j,S_x'(0)}\\&=\prod_{n\in S_x}\left(e(g,g)^{r\beta q_{parent(x)}(index(x))}\right)^{\nabla_j,S_x'(0)}\\&=\prod_{n\in S_x}e(g,g)^{r\beta q_x(j)\nabla_j,S_x'(0)}\\&=e(g,g)^{r\beta q_x(0)}\end{aligned}$$

② 如果 $a_j\notin S$ 有:

$$Fog.DecryptNode(CT,SK',x)=\perp$$

③ 如果是 x 非叶节点算法定义是: 对于 x 所有的孩子节点 n, 它执行 $F_n=Fog.DecryptNode(CT,SK',n)$, 让, 如果没有这个集合存在, 则 $F_n=\perp$, 否则, 计算:

$$\begin{aligned}&Fog.DecryptNode(CT,SK',x)\\&=e\left(D_j,C_j\right)\\&=e\left(g^{\frac{r\beta}{v_j}},g^{v_j q_x(0)}\right)\\&=e(g,g)^{r\beta q_x(0)}\end{aligned}$$

让 $j=index(n)$ 和 $S_x'=\{index(n):n\in S_x\}$. 如果属性集合 S 满足访问策略 T. 则整个计算结果是:

$$\begin{aligned}F&=Fog.DecryptNode(CT,SK',R)\\&=e(g,g)^{r\beta q_R(0)}\\&=e(g,g)^{r\beta s_1}\end{aligned}$$

然后, 雾节点计算:

$$\begin{cases}A=\dfrac{B}{F}=\dfrac{e(g,g)^{r\beta(s_1+s_2)}}{e(g,g)^{r\beta s_1}}=e(g,g)^{r\beta s_2}\\B=\dfrac{e(D',C')}{e(D'',C'')}=\dfrac{e\left(g^{r}h^{\varepsilon},g^{s_1\beta}g^{\beta s_2}\right)}{e\left(g^{\varepsilon},h^{s_1\beta}h^{\beta s_2}\right)}=e(g,g)^{r\beta(s_1+s_2)}\end{cases}$$

(7) 车辆用户解密

雾节点把预解密密文发送给用户, 用户用自己的私钥解密得到对称密钥;

$$\frac{\widetilde{C}\cdot A}{e(D,C)}=\frac{ck\cdot e(g,g)^{\alpha\beta s_2}\cdot e(g,g)^{r\beta s_2}}{e\left(g^{\beta(\alpha+r)},g^{s_2}\right)}=ck$$

最后再用对称密钥解密 $E_{ck}(M)$ 得到消息.

(8) 属性更新

更新密钥生成: 算法输入公钥, 主密钥和属性 a_j,a_j'. 计算属性更新密钥 $UK_j=\dfrac{v_j}{v_j'}$. 权威选择一个随机数 $\overline{v}_j\in Z_p\left(\overline{v}_j\neq v_j\right)$, 它为没有更新的用户产生 $UK_j'=\dfrac{v_j}{\overline{v}_j}$,

用以更新云中的密文. 分别发送$UK_j=\frac{v_j}{v_j'}$, $UK_j'=\frac{v_j}{\overline{v_j}}$, 及$UK_j'=\frac{v_j}{\overline{v_j}}$给更新用户, 未更新用户和云. 更新的公共属性密钥$PK_j'=\left(PK_j\right)^{UK_j''}=g^{\overline{v_j}}$

更新的用户私钥:

$$SK_u=\left\{D=g^{\beta(\alpha+r)},D'=g^r h^{\varepsilon},D''=g^{\varepsilon},\left\{D_j'=D_j=g^{\frac{r\beta}{v_j}}\mid j\neq j'\right\},\left\{D_j'=\left(D_j\right)^{UK_{j'}}\mid j=j'\right\}\right\}$$

5 安全分析

假如存在多项式时间的敌手以不可忽略的优势攻破本文方案, 则存在一个算法以一定的优势解决 DBDH 问题. 我们方案的安全性是基于 DBDH 困难问题假设做的, 证明在文献 [14] 中. 本方案的安全性能分析如下:

(1) 细粒度的访问控制

为了能够灵活的访问数据, 就需要细粒度的访问加密数据, 我们利用 CP-ABE 来管理系统的加密密钥. 在加密阶段, 数据拥有者定义访问结构并用对称密钥对消息进行加密. 当有用户要访问数据时, 需要自己的属性集合满足定义的访问策略, 否则, 解密失败. 因此, 本方案在访问控制方面具有灵活性.

(2) 抗共谋

当两个或多个用户想要结合他们的密钥来访问加密数据时, 这个操作是实现不了的. 在本方案中, 属性权威会给每个用户产生不同的私钥, 这个私钥是和一个随机数相关的, 多个用户密钥结合不能恢复出来访问密钥.

6 性能分析

6.1 功能性比较

在这部分, 我们将本方案与其他方案进行了对比, 这里我们更注重功能和计算复杂度的比较. 表 1 是功能比较, 表中√表示满足, ×表示不满足.

在这里可以看出, 文献 [15] 实现了撤销和关键字搜索, 但没有实现雾计算和外包. 文献 [16] 仅实现了撤销, 其他都没实现. 文献 [17] 仅实现了撤销, 其他功能均没有实现. 文献 [18] 实现了外包, 撤销和关键字搜索, 没有实现解密外包和雾计算. 本方案以上功能均实现了, 看出本方案功能齐全, 有优势. 本方案以上功能均实现了, 看出本方案功能齐全, 有优势.

表 1 功能性比较

方案	雾计算	加密外包	解密外包	属性撤销	键字搜索
文献[15]	×	×	×	√	√
文献[16]	×	×	×	√	×
文献[17]	√	√	√	×	×
文献[18]	×	×	√	√	√
本方案	√	√	√	√	√

6.2 计算开销

本节对文献 [15–18] 等方案的计算开销进行了测试. 设 p 表示双线性配对所需的时间, e 表示在群 G 中标量乘法所需的时间. 本方案在 i5-M60, 2.53 GHz, i5 CPU, 4 GB 内存和 64 位 Windows 10 上进行仿真. 表 2 列出了操作的平均运行时间.

表 2 平均运行时间

符号	注释	时间(ms)
p	双线性配对运算的时间	10.31
e	在群G中标量乘法运算的时间	0.52

假设在密文生成, 密钥生成, 密钥更新, 外包解密等阶段数据使用者从属性授权得到的属性数量 l 是一致的. 表 3 中总结了本文方案和文献 [15–18] 方案的计算开销, 对每个过程所需要的计算量进行了量化, 对应的实验仿真结果如图 2–图 5 所示.

表 3 计算开销比较

步骤	文献[15]	文献[16]	文献[17]	文献[18]	本文
参数	$(3+m)e+p$	$(2+m)e+p$	$4e+p$	$(3+2m)e+p$	$(4+m)e+p$
私钥	$(4+l)e$	$(5+3l)e$	$(5+2l)e$	$(2+2l)e$	$(5+l)e$
密文	$(2+2l)e+p$	$(4+3l)e+p$	$(6+2l)e+p$	$(2+3l)e+p$	$(6+l)e+p$
密钥更新	le	$(1+2l)e$	×	$(1+l)e$	le
Trapdoor	$3e+m$	×	×	$2e$	$2e$
索引	$5e$	×	×	$e+2p$	$e+2p$
外包加密	×	×	$2e+2l$	×	$2e+l$
外包解密	×	×	$2e+(2+2l)+p$	$2lp$	$2e+(2+l)p$
用户解密	$2e+(2+2l)p$	$2e+(3+3l)p$	$2e+p$	$2e+p$	$2e+p$

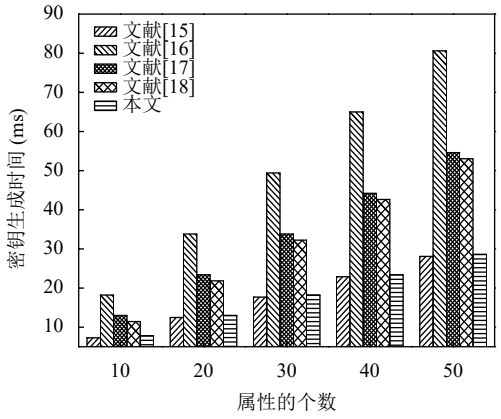

图 2 密钥生成时间

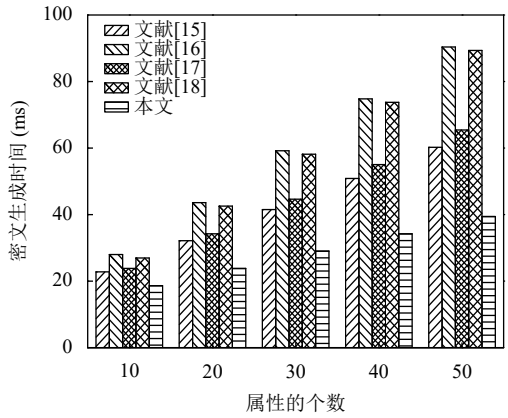

图 3 密文生成时间

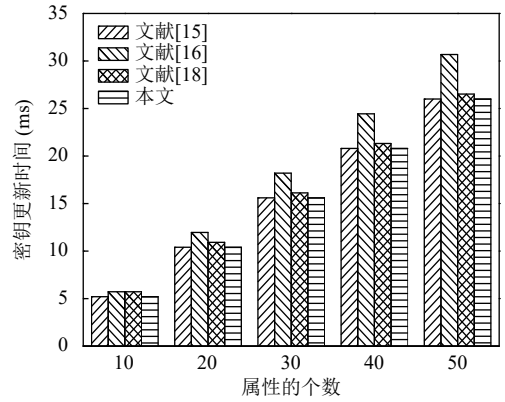

图 4 密钥更新时间

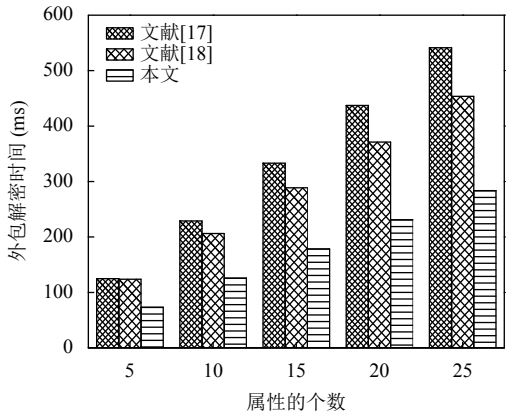

图 5 外包解密时间

在表 3 中, m 表示集合中属性的个数; e 表示G_T,G 中的指数运算; p 表示一个对运算; l 表示各个操作中涉及的属性的个数.

图 2–图 5 分别表示的是密钥生成时间, 密文生成时间, 密钥更新时间以及外包解密时间与属性数量的关系, 其中密钥中涉及到的属性数量被设置为 10, 即 $l = 10$. 从图 2–图 5 可知, 这几个阶段的计算开销随着涉及到的属性数量的变化呈线性增长.

在图 2 中, 本文方案的密钥生成时间低于文献 [16–18] 等对比方案, 计算成本更低. 在图 3 中, 本文方案的密文生成时间低于所有对比方案, 计算成本低且更有效. 在图 4 中, 本文方案的密钥更新生成时间低于文献 [15–17] 等对比方案, 计算成本低. 在图 5 中, 本文方案的外包解密计算时间低于文献 [17,18] 等对比方案, 计算成本低, 其他方案没有实现外包.

经分析可知, 本文提出的方案在加密、解密和密文更新阶段所需的计算开销是最低的, 比文献 [15–18] 方案有更多的优势.

7 总结

本文提出了车载自组网中基于属性的可搜索加密和属性撤销方案. 首先, 将车辆用户的敏感数据和访问策略经过加密通过雾节点外包给云服务器, 并将部分加密和解密外包给雾节点, 从而减轻了车辆用户的计算负担. 其次, 我们的应用场景是车载自组网, 方案中车辆拥有者能够将数据通过雾节点安全的存储在云服务器上, 以便其他用户可以有效的访问. 再者, 我们的方案支持关键字的可搜索加密和细粒度的访问控制以及属性更新, 避免不合法用户对数据的访问. 最后, 经过比较, 我们的方案功能更加齐全, 拥有更大的优势. 在未来工作中, 我们将继续研究多授权、分层、多关键字搜索等在基于属性的加密中的应用.

参考文献

1 Levinson D, Chang E. A model for optimizing electronic toll collection systems. Transportation Research Part A: Policy and Practice, 2003, 37(4): 293–314. [doi: 10.1016/S0965-8564(02)00017-4]

2 Bonomi F, Milito R, Zhu J, *et al*. Fog computing and its role in the Internet of Things. Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing. Helsinki, Finland. 2012. 13–16.

3 Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. Proceeding of 2000 IEEE Symposium on Security and Privacy. Berkeley, CA, USA. 2000. 44–55.

4 Sahai A, Waters B. Fuzzy identity-based encryption. Proceedings of the 24th Annual International Conference on

the Theory and Applications of Cryptographic Techniques. Aarhus, Danmark. 2005. 457–473.

5 Goyal V, Pandey O, Sahai A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security. Alexandria, IN, USA. 2006. 89–98.

6 Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. Proceedings of 2007 IEEE Symposium on Security and Privacy (SP'07). Berkeley, CA, USA. 2007. 321–334.

7 Huang DJ, Verma M. ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks. Ad Hoc Networks, 2009, 7(8): 1526 –1535. [doi: 10.1016/j.adhoc.2009.04.011]

8 Yeh LY, Chen YC, Huang JL. ABACS: An attribute-based access control system for emergency services over vehicular ad hoc networks. IEEE Journal on Selected Areas in Communications, 2011, 29(3): 630–643. [doi: 10.1109/JSAC.2011.110312]

9 Rao YS, Dutta R. Efficient attribute based access control mechanism for vehicular ad hoc network. Proceedings of the 7th International Conference on Network and System Security. Madrid, Spain. 2013. 26–39.

10 Yeh LY, Huang JL. PBS: A portable billing scheme with fine-grained access control for service-oriented vehicular networks. IEEE Transactions on Mobile Computing, 2014, 13(11): 2606–2619. [doi: 10.1109/TMC.2013.45]

11 Bouabdellah M, El Bouanani F, Ben-Azza H. A secure cooperative transmission model in VANET using attribute based encryption. Proceedings of 2016 International Conference on Advanced Communication Systems and Information Security. Marrakesh, Morocco. 2016. 1–6.

12 Xia YJ, Chen WZ, Liu XJ, *et al*. Adaptive multimedia data forwarding for privacy preservation in vehicular ad-hoc networks. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(10): 2629–2641. [doi: 10.1109/TITS.2017.2653103]

13 Xue KP, Hong JN, Ma YJ, *et al*. Fog-aided verifiable privacy preserving access control for latency-sensitive data sharing in vehicular cloud computing. IEEE Network, 2018, 32(3): 7–13. [doi: 10.1109/MNET.2018.1700341]

14 Zhang P, Chen ZH, Liu JK, *et al*. An efficient access control scheme with outsourcing capability and attribute update for fog computing. Future Generation Computer Systems, 2018, 78: 753–762. [doi: 10.1016/j.future.2016.12.015]

15 Sun J, Ren LL, Wang SP, *et al*. Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage. IEEE Access, 2019, 7: 66655–66667. [doi: 10.1109/ACCESS.2019.2917772]

16 Zhang P, Chen ZH, Liang KT, *et al*. A cloud-based access control scheme with user revocation and attribute update. Proceedings of the 21st Australasian Conference on Information Security and Privacy. Melbourne, Australia. 2016. 525–540.

17 Huang QL, Yang YX, Wang LC. Secure data access control with Ciphertext update and computation outsourcing in fog computing for Internet of Things. IEEE Access, 2017, 5: 12941–12950. [doi: 10.1109/ACCESS.2017.2727054]

18 Wang SP, Ye J, Zhang YL. A keyword searchable attribute-based encryption scheme with attribute update for cloud storage. PLoS One, 2018, 13(5): e0197318. [doi: 10.1371/journal.pone.0197318]