面向车联网车辆的轻量级持续身份认证协议

邹光南[1],尤启迪[2],金星虎[2],马永春[2],李洁榆[2]
(1. 清华大学计算机科学与技术系,北京 100084;2. 航天恒星科技有限公司,北京 100086)

摘 要: 基于云-边缘计算的车联网(Cloud-Edge computing for the Internet of Vehicle,CEIoV)能够支持大规模车辆的实时访问与服务请求,为了保证其内部资源的安全性,需要对车辆进行身份认证而后才能接入CEIoV;但是车辆本身处于运行状态且计算、存储和通信资源受限,给CEIoV车辆的身份认证带来挑战. 本文基于具有简单密码操作的变色龙哈希函数,提出了一个连续轻量级身份认证协议(Lightweight Continuous identity Authentication,LCA),实现了对于资源受限车辆的认证和CEIoV内部资源的安全保障. 本文在随机预言机模型下证明了LCA协议的语义安全性;并通过实验验证LCA协议在连续认证过程中具有较低的计算和通信成本.

关键词: 身份认证;变色龙哈希函数;轻量级;多接入边缘计算;隐私保护
基金项目: 国家自然科学基金(No.62173026)
中图分类号: TN918.91 **文献标识码:** A **文章编号:** 0372-2112(2024)06-1903-08
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230661

Lightweight Continuous Authentication Protocol for Vehicles in Vehicular Networks

ZOU Guang-nan[1], YOU Qi-di[2], JIN Xing-hu[2], MA Yong-chun[2], LI Jie-yu[2]
(1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*;
2. *Spacestar Technology Co., Ltd., Beijing* 100086, *China*)

Abstract: Cloud-edge computing for the Internet of vehicle (CEIoV) can support real-time access and service requests of large-scale vehicles. In order to ensure the security of its internal resources, vehicle identity usually needs to be validated before it can access CEIoV. However, because the vehicle itself is in the running state and moreover its computing, storage and communication resources are limited, the existing identity authentication protocol cannot be directly applied to authenticate a running vehicle in CEIoV. Therefore, this paper proposes a lightweight continuous authentication (LCA) protocol to realize vehicle authentication and guarantee the security of CEIoV internal resources. LCA is designed based on chameleon Hash function, whose implementation requires simple cryptographic operations and is easy to be deployed on the resource-limited devices. By using random oracle model, the semantic security of LCA is proved strictly. At the same time, the experimental results show that LCA has lower computational and communication costs in the continuous authentication process compared with prior schemes.

Key words: identity authentication; chameleon Hash function; lightweight; mobile edge computing; privacy protection
Foundation Item(s): National Natural Science Foundation of China (No.62173026)

1 引言

边缘计算(Mobile Edge Computing,MEC)将计算智能延伸到网络边缘,拥有更强大的数据处理和存储能力,为车辆提供了响应更快、更易于访问的实时服务[1]. MEC与区域云中心(District Cloud Center,DCC)相结合形成的层级体系,通过车路云一体化协同控制,提高了交通的整体管理能力[2]. 然而,MEC分布式布局、开放的应用服务接口和多种不可信终端设备接入,为攻击者提供了攻击入口,存在数据安全和隐私风险;此外,MEC设备大多靠近终端设备,部署密码安全保护措施的能力有限. 因此,MEC在提供服务时,极易受到中间人攻击、重放攻击、消息篡改等攻击.

基于云-边缘计算的车联网(Cloud-edge computing for the Internet of Vehicle,CEIoV)能够支持大规模车辆

收稿日期:2023-07-14;修回日期:2024-04-02;责任编辑:李勇锋

的实时访问与服务请求，为了保证其内部资源的安全性与隐私性，通常车辆需要进行认证而后才能接入CEIoV. 但是车辆本身处于运行状态且计算、存储和通信资源受限，给CEIoV车辆的身份认证带来挑战. 因此，如何高效认证资源受限的车辆，同时保证车辆从源MEC无缝过渡到目标MEC时，为亟待解决的问题.

1.1 相关工作

针对车联网中车辆认证协议的轻量级、易于部署、密码操作简单的需求，研究工作者开展了一系列研究. Almajali等人[3]提出了一种安全高效的MEC框架，适用于移动设备的物联网应用，并且比较了面向物联网资源访问控制框架的经典身份认证协议. 贾等人[4]在MEC环境下提出了一种基于身份的AAKA协议，具有用户匿名和不可追踪的特性；该协议不依赖可信任的第三方，但性能开销较高. 为了规避第三方密钥托管的风险，Sun等人[5]设计了边缘云协同计算场景下的跨域认证协议；该协议存在密码运算复杂、计算开销大、不易部署等问题. 针对车辆频繁切换安全域的问题，Son等人[6]提出了一种仅在车辆切换期间执行轻量级计算的协议；该协议具有较低的网络负荷，但认证过程不适合高速移动车辆实时数据处理的这一场景. Babu等人提出的Ev-auth协议[7]可实现实体间的无缝切换；但是其通信开销很高，给网络带来了高负荷. Yu等人[8]基于3GPP规定的5G架构，提出Bash协议，将车辆信息记录在区块链上，减少了交接认证的步骤和开销，但存在泄露车主个人隐私的风险.

以上研究分析显示，现有工作已经注意到车辆频繁切换安全域时的认证安全和效率问题，但这些成果仍存在用户隐私泄露、跨域认证切换时延高、密码运算复杂，从而导致部署困难问题. 第6节的内容明确地给出了上述讨论的研究成果在连续认证过程中计算和通信负荷较大.

1.2 动机和贡献

车辆访问MEC等基础设施资源场景下的身份认证协议应至少满足如下安全、功能和性能要求.

(1)安全性. 协议应具有抵抗传统攻击的能力，如重放攻击、篡改攻击、中间人攻击等.

(2)车辆身份隐私. 保护车辆真实身份.

(3)跨域认证功能. 协议应当提供车辆的跨域认证功能.

(4)轻量性. 由于云-边服务器的计算能力可以通过配置高性能服务器或配置多台服务器来增强，因此车辆认证协议至少要保证车辆端计算负荷和通信负荷是轻量；这种轻量可以满足车辆密集场景里车辆能够快速完成重认证. 因此，认证协议在车辆全流程中应当具有计算、通信开销小的特点，以方便资源受限实体的认证需求；此外，也可满足大规模服务请求的能力；即保证协议在大规模场景下的有效性.

(5)易于部署. 车辆的计算和存储能力有限，协议需要设置简单有效的密码保护.

(6)连续认证. 车辆要定期发送认证信息给MEC，来保证临时身份的有效性.

从1.1节的讨论可以看出，现有研究成果并不能满足上述安全、功能和性能要求. 本文的主要贡献如下.

(1)提出了一种轻量级连续认证协议(Lightweight Continuous Authentication，LCA)，可以实现MEC对于车辆的身份认证，同时保护车辆隐私信息；

(2)相比于以往的工作，LCA协议具有低计算和通信开销，适用于资源有限的车辆；这里通信开销小是指车辆完成第一次认证后，在后续的认证中，只需发送一条通信验证信息，就可以完成身份认证，这也减少了服务中心的处理负荷，因此说LCA协议适合于大规模服务请求的场景. 此外，LCA中包含的密码操作简单，更易于部署.

(3)基于随机预言机模型[9]，证明了LCA协议的语义安全性，即攻击者无法通过某段密文获得与其相关明文的任何信息；并通过非形式化安全分析，证明了LCA协议能够抵抗重放攻击、中间人攻击，并保护车辆身份隐私.

2 基础知识

2.1 哈希函数

定义变量p和q为两个大素数且$p=u\cdot q$，变量u是一个小整数.

(1) 泛哈希函数

泛哈希函数(Universal Hash Function，UHF)[10]是一类抗碰撞的哈希函数，可表示为$K\times X\to Y$，其中K表示密钥空间，X表示输入空间，Y表示UHF输出空间. 本论文假设$K=X=Y=Z_q$. 根据文献[11]，对UHF进行实例化$y=\mathrm{UHF}(k,x)=k_1x+k_2(\mathrm{mod}\,q)$，参数$k_1$、$k_2$都是从$Z_q^*$空间随机选择.

定义1 一个哈希函数如果满足以下两个条件，则被称为泛哈希函数：

(i)通过随机实例化一个哈希密钥$k\xleftarrow{\$}K$来统一选择一个哈希函数$\mathrm{UHF}\in\mathrm{UH}$；(ii)对于任意$\forall(x,y)\in X$，式(1)都成立，其中$\#X$表示集合$X$中元素的个数.

$$\Pr[\mathrm{UHF}(k,x)=\mathrm{UHF}(k,y)]\leqslant\frac{1}{\#X} \tag{1}$$

(2) 变色龙哈希函数

除了UHF，身份认证过程还要使用变色龙函数，定义M、R、SK|PK分别表示消息空间、随机数空间和密钥空间，其范围都是取决于δ值. 变色龙哈希函数包含如下三个概率多项式时间(probability probabilistic polyno-

mial time,PPT)算法[11]:

CHKGen(1^{δ}):变色龙哈希函数的公私钥生成算法,选两个满足如下2个条件的素数P和q:(i)q足够大;(ii)$p=kq+1$. 定义Z_q^*是一个阶为q的群,g是群生成元. 设置私钥(陷门)$\mathrm{sk}\leftarrow Z_q^*$,然后计算公钥$\mathrm{pk}=g^{\mathrm{sk}}(\mathrm{mod}\,p)$.

CHF(pk,m,r):变色龙哈希函数根据输入公钥pk、消息$m\in Z_q^*$、随机数$r\in Z_q^*$,计算出哈希值$\mathrm{CHF}(\mathrm{pk},m,r)=g^m\mathrm{pk}^r(\mathrm{mod}\,p)$. 具有抗碰撞特征是指不基于pk,能得到一对$(m,r)$和$(m',r')$,使得$\mathrm{CHF}(\mathrm{pk},m,r)=\mathrm{CHF}(\mathrm{pk},m',r')$且$m\neq m'$,在安全参数$\delta$下具有可忽略的概率.

CHColl(sk,r,m,m'):CHColl是一种具有陷门碰撞特性的高效确定性算法,输入(r,m,m')和sk,输出$r'=\frac{m+r\times\mathrm{sk}-m'}{\mathrm{sk}}(\mathrm{mod}\,q)$以使得$\mathrm{CHF}(\mathrm{pk},m,r)=\mathrm{CHF}(\mathrm{pk},m',r')$.

2.2　布隆过滤器

布隆过滤器(Bloom Filter,BF)[12]是一种基于概率的数据结构,主要用来判断某个元素是否在集合内,具有运行速度快(时间效率),占用内存小的优点(空间效率),但是有一定的误识别率,其只能判断某个元素一定不在集合内或可能在集合内. 但是根据所要存储的元素个数N,设定布隆过滤器使用位图的大小,可以使得误识别率处于可以接受的范围. 布隆过滤器包含以下四个算法:

(1)Init(N,p):初始化函数根据输入N和误报率ε,来产生一个长度为$1.44\varepsilon N$的布隆过滤器,其中误报率为2^{-p}[13].

(2)Insert(m):插入函数负责将元素m放入到BF.

(3)Check(m):检测函数用于检测元素m是否在BF中,如果在,则返回1,否则返回0.

(4)Pos(m):位置更新算法,用户计算布隆过滤器中m元素的位置.

3　系统模型和攻击模型

3.1　系统模型

LCA协议的系统模型主要包括三个实体,如图1所示,即区域云中心(DCC)、边缘计算节点(MEC)和车辆(V).

(1)DCC:即具有充足计算和存储资源的区域云中心,是多个多接入边缘节点的汇聚中心. DCC负责采集全域车辆和道路数据,提供实时性较弱的驾驶和交通支持服务. 在该系统模型下,DCC作为可信的第三方实体存在,其身份用RID表示,并生成系统参数κ. 它与交通和执法部门合作,对车辆进行注册,并为车辆生成临时身份TVID、公钥/私钥对和验证值. 验证值存储在由布隆过滤器材生成的名为BF的列表里.

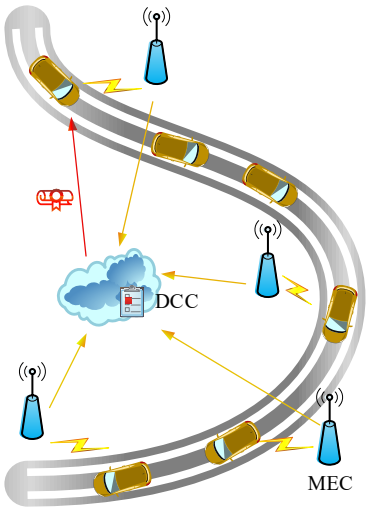


图1　系统模型

(2)MEC:即多个多接入边缘计算节点,部署在路边. 每个边缘计算节点负责采集和计算全域道路的动态交通数据. MEC提供实时服务,可以提高智能网联汽车的安全和效率. 每个边缘计算节点都会存储DCC的身份RID,用于确认从DCC收到的消息. MEC从DCC接收验证值列表BF,用于验证V的身份.

(3)V:即车辆,行驶过程中经过多个MEC. V的身份用VID表示. 为保证动态车辆身份的合法性,每次到达MEC管辖区域都需要进行身份验证. 此外,为了解决仅在服务入口点进行身份验证的传统解决方案的安全威胁,还需要在同一MEC处定期重新进行身份认证. DCC每天清除车辆的TVID存储库,但VID和车辆不当行为记录会永久保存.

3.2　假设

LCA协议针对多项式时间内的攻击者考虑了以下假设.

(1)MEC会诚实地按照协议流程为车辆提供服务.

(2)所有车辆和MEC都将DCC视为完全可信的第三方.

(3)基于变色龙哈希函数进行进行签名的安全性由离散对数困难问题决定.

(4)DCC为注册车辆生成的临时身份,并将更新后的临时身份TVID、r_j及更新后的r_{j+1}和Vsk安全发送给车辆.

3.3　攻击模型

本文所考虑的攻击者,具备如下能力,本文所设计的身份认证协议能够防御具有这些能力的攻击者发起的攻击:

(1)通信信息在无线和开放的环境中传输,因此攻

击者具有窃听、拦截任何信息的能力，由此可能会根据拦截到的信息获取关于认证过程的额外信息.

(2)攻击者具有修改、删除合法内容，或插入恶意内容的能力.

3.4 形式化安全模型

本文通过定义一个敌手与挑战者之间的游戏来描述LCA认证过程的安全模型.

定义2 令 $G_{\mathrm{A}}^{\mathrm{LCA}}(\delta)$ 是一个PPT敌手 $\mathcal{A}$ 和挑战者 $\mathcal{C}$ 根据LCA而进行的一个理想游戏，且安全参数是 δ，$\mathrm{Adv}_{\mathrm{A,LCA}}^{\mathrm{G}}(\delta)=\left|\Pr\left[\mathrm{G}_{\mathrm{A}}^{\mathrm{LCA}}(\kappa)=1\right]-\frac{1}{2}\right|$ 是一个PPT敌手 $\mathcal{A}$ 在安全参数 δ 下赢得游戏 $G_{\mathrm{A}}^{\mathrm{LCA}}(\delta)$ 的优势. 如果没有一个PPT敌手有不可忽略的优势 $\mathrm{Adv}_{\mathrm{A,LCA}}^{\mathrm{G}}(\delta)$，则LCA满足语义安全，即攻击者无法通过某段密文从而得到与其相关明文的任何信息.

$\mathrm{G}_{\mathrm{A}}^{\mathrm{LCA}}(\delta)$ 的具体描述如下：

敌手 $\mathcal{A}$ 选择两个消息 $m_0,m_1\in M$，发送给挑战者 $\mathcal{C}$. 挑战者 $\mathcal{C}$ 随机选择一个比特 $b\in\{0,1\}$，同时生成变色龙哈希函数的sk并生成pk，随机选择种子并使用UHF生成随机数 r，然后为 m_b 生成签名 S_b，同时将签名结果发送给敌手 $\mathcal{A}$. 敌手 $\mathcal{A}$ 收到 S 后，发送比特 $b'\in\{0,1\}$ 给挑战者 $\mathcal{C}$，则如果 $b=b'$，敌手 $\mathcal{A}$ 赢得游戏，输出1，否则输出0.

4 LCA协议

本节将介绍LCA协议流程. 表1列出了LCA协议使用的符号，LCA协议是基于Lis[9]签名方案而创新提出的，认证原理如图2所示，流程如图3所示，由基于泛哈希函数族和变色龙哈希函数的算法组成. 算法描述如下：

RegV(VID)：由DCC运行的车辆身份注册算法为车辆生成一个临时身份 $\mathrm{TVID}\leftarrow Z_q^*$. 此外，该算法还为车辆生成私钥 $\mathrm{Vsk}\leftarrow Z_q^*$ 和公钥 $\mathrm{Vpk}=g^{\mathrm{Vsk}}(\mathrm{mod}\,q)$.

UpdR(r_{j-1})：DCC运行随机数更新算法. DCC获取随机数后，撤销TVID的本地存储并生成一个新的随机数. 该算法在以下三种情况下执行：第一种是车辆在规定的时间间隔内没有发起认证请求；其次，车辆在跨越到另一个MEC的管辖范围内行驶并需要认证时；最后，车辆现有的随机数已失效. 随机值 r_j 的更新为 $r_j=\mathrm{UHF}\left(k,r_{j-1}\right)(j>0)$，其中，$r_0\in X$.

$\mathrm{CreReq}\left(\mathrm{Vsk},r_j,\mathrm{TVID},Q_i\right)$：车辆第 i 次与DCC通信请求的创建算法，以Vsk、随机值 r_j、TVID和 Q_i（例如认证请求消息或通信消息）作为输入. 然后该算法计算并输出 Q_i 的签名为 $\mathrm{QS}_i=\mathrm{CHColl}\left(\mathrm{Vsk},r_j,\mathrm{TVID},Q_i\right)$. 当随机值 r_j 更新后，车辆需要使用新的随机值进行签名的生成.

$\mathrm{ComVP}\left(\mathrm{Vpk},\mathrm{TVID},r_j,\mathrm{BF}\right)$：DCC运行的验证点计算算法，以TVID和随机值 r_j 作为输入. 然后计算验证点 $\mathrm{VP}_i=\mathrm{CHF}\left(\mathrm{Vpk},\mathrm{TVID},r_j\right)$ 并通过BF. $\mathrm{Insert}\left(\mathrm{VP}_i\right)$ 将其存储到BF中. DCC为同一个车辆第一次运行此算法时，$j=0$. 当随机值 r_j 更新后，DCC需要使用新的随机值进行验证点的生成.

$\mathrm{VerifyV}\left(\mathrm{Vpk},Q_i,\mathrm{QS}_i,\mathrm{BF}\right)$：车辆身份鉴别签名验证算法以Vpk、认证请求 Q_i 及其签名 QS_i 作为输入，接下来通过BF. $\mathrm{Check}\left(V_i\right)$ 判断认证信息 $V_i=\mathrm{CHF}\left(Q_i,\mathrm{QS}_i\right)$ 是否为BF中的一个元素，如果是，算法返回1，否则返回0.

LCA协议由三个阶段组成，称为DCC系统初始化、注册阶段、V-MEC认证阶段，分别在4.1~4.3节中详细介绍.

表1 LCA协议符号

符号	描述
$(\mathrm{Vpk},\mathrm{Vsk})$	车辆的公钥和私钥
r_j	用于认证的秘密值,通过UHF进行更新
R_0	重置后初始随机数值
Q_i	车辆身份鉴别请求
QS_i	车辆身份鉴别请求签名
VP_i	DCC预先计算的身份验证点
V_i	由MEC计算的认证信息
BF	存储 VP_i 的布隆过滤器

4.1 DCC系统初始化

在这个阶段，DCC产生如下参数：(1)两个大素数 p 和 q；(2)$(k_1,k_2)\leftarrow Z_q^*$ 给UHF；(3)用于初始化系统的参数 $g\leftarrow Z_q^*$. 此外DCC调用BF. $\mathrm{Init}(N,\varepsilon)$ 完成BF验证值的初始化.

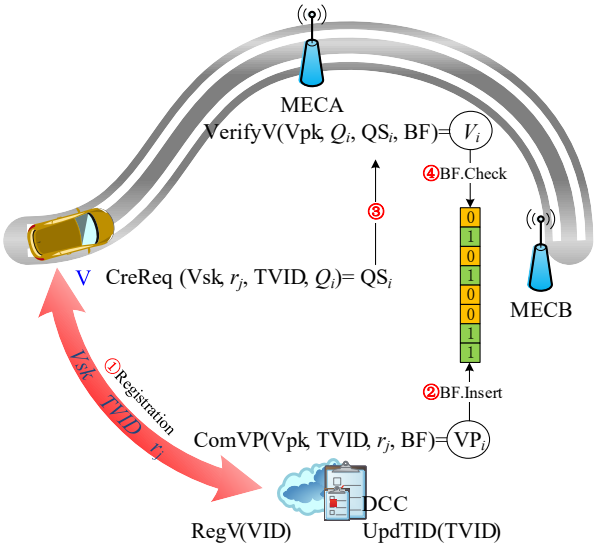


图2 MECA连续对车辆进行身份鉴别流程

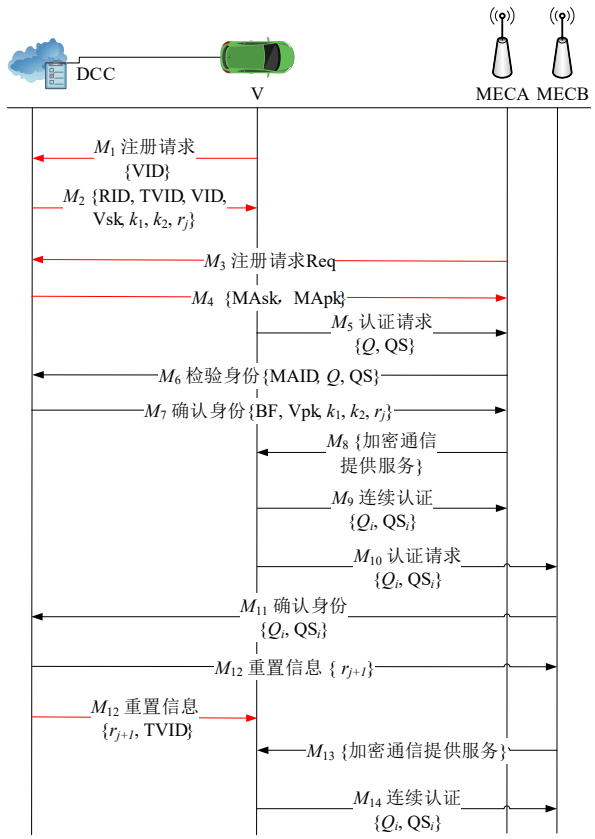


图 3　LCA 协议流程

4.2　注册阶段

（1）　车辆注册

车辆必须先通过安全信道在 DCC 注册个体信息，才能去访问 MEC 服务. 注册步骤如下：

M_1：请求注册，V→DCC：{VID}

V 向 DCC 发出注册请求，请求中包含了车辆 V 的身份信息，例如{VID}.

M_2：DCC 产生 ID，DCC→V：{RID, TVID, VID, Vsk, r_j}

DCC 收到注册信息后，DCC 会检查本地数据库，确认车辆是否有过恶意行为记录. 如果有不良记录，则拒绝改注册请求，该车辆 V 将无法访问 MEC 提供的服务.

如果没有恶意行为记录，DCC 则调用 RegV(VID) 为车辆产生 TVID、公私钥对(Vpk, Vsk). 同时 DCC 调用 ComVP(Vpk, TVID, r_j, BF)，为 V-MEC 认证阶段产生初始验证点 VP，并存储到 BF 中. 对于之后 r_j、TVID 需要更新，则 DCC 需要重新计算认证值 VP 并将其存到 BF 中.

DCC 存储参数{VP, TVID, VID, Vpk, r_j}，同时车辆安全存储{RID, TVID, VID, Vsk, r_j}.

（2）　MEC 节点注册

MEC 节点的注册方式均相同，这里以节点 MECA 为例：

M_3：请求注册，MECA→DCC：{MAID, Location}

MECA 向 DCC 发出注册请求，请求中包含了其身份和地理位置等信息，例如 Req={MAID, Location}.

M_4：DCC 生成公私钥对(MApk, MAsk)，DCC→V：{MApk, MAsk

DCC 收到注册信息后等待带外信息确认 MECA 的相关信息，如果信息准确，其通过 $\mathrm{MAsk} \leftarrow Z_q^*$ 和 $\mathrm{MApk} = g^{\mathrm{MAsk}} (\mathrm{mod}\, q)$ 为 MECA 生成私钥和公钥，并通过安全信道返回给 MECA. 此外，DCC 会不定期向 MECA 同步已经注册车辆的 RID 以及公钥，用于后续的安全认证.

4.3　V-MEC 认证阶段

根据车辆的活动范围，V 与 MEC 之间的认证分为两种情况. 以 V 在 MECA 和 MECB 中的认证为例. 第一种情况是 V 只在一个 MEC 负责的区域行驶，即 V 每次只需要在 MECA 中进行认证. 第一种情况是 V 从 MECA 管辖区域移动到 MECB 管辖区域.

（1）　情况 1：车辆只在 MECA 管辖范围内行驶

M_5：车辆生成并发送第 i 次身份认证请求，V→MECA：{Q_i, QS_i}

如果 V 要访问 MECA 提供的服务，它首先发出身份认证请求 $Q_i = I_i \| \mathrm{RID}$，请求中包含认证请求内容 I_i 和身份信息 RID，然后调用 CreReq(Vsk, r_j, TVID, Q_i) 计算 Q_i 的签名 QS_i.

M_6：MECA 转发认证请求，MECA→DCC：{MAID, Q_i, QS_i}

MAID 为 MECA 身份标识，当 MECA 收到车辆 V 的身份认证请求后，MECA 将信息{MAID, Q_i, QS_i}发送给 DCC.

M_7：确认身份，DCC→MECA：{RID, BF, Vpk, r_j}

DCC 收到 MECA 发送的信息后，首先确认 MECA 是可信的，然后调用 VerifyV(Vpk, Q_i, QS_i, BF)，如果返回结果是满足要求，则说明 Q_i 没有被篡改且 V 身份是合法的，用来确定车辆身份；并将验证参数{RID, BF, Vpk, r_j}发送给 MECA，同时附上自己对于验证参数的签名. 如果返回结果是不满足要求，则拒绝请求. 对于布隆过滤器本身导致身份认证错误的情况，通过将布隆过滤器位图设置的足够大，从而可将“假阳”概率降低到可以接受的范围.

M_8：确认身份，通过加密通信安全提供服务 MECA→V：{encrypted communication using sk}

MECA 首先根据 RID 以及 Vpk 来确认消息确实是从 DCC 发送的. 因为对于合法的 V，由于其无法伪造 BF，那么其伪造行为会被 MECA 发现从而进一步发现消息不

来自于DCC;对于恶意的V,由于其无法伪造合法的RID,Vpk以及DCC,从而MECA也可以发现消息不来自于DCC. 因为MECA相信DCC的权威以及V身份的合法性,因此其计算会话密钥$\mathrm{sk}=\mathrm{Vp\,k}^{\mathrm{MAsk}}(\mathrm{mod}\,q)$并使用sk加密通信为V提供服务. V通过计算$\mathrm{MAp\,k}^{\mathrm{Vsk}}(\mathrm{mod}\,q)=\mathrm{sk}$获得相同的会话密钥进行安全通信.

M_9:再次身份认证,V→MECA:$\{Q_i,\mathrm{QS}_i\}$

如果车辆V收到MAID,则意味MECA愿意提供服务给车辆. 每间隔T时间,V会发送一个重新身份认证请求$Q_i\in M$以及通过$\mathrm{CreReq}\left(\mathrm{Vsk},r_j,\mathrm{TVID},Q_i\right)$计算的请求的签名$\mathrm{QS}_i$,以确认MECA服务的安全性,其中,$Q_i=I_i\|\mathrm{MAID}$,即包含车辆的服务请求指令$I_i$和MECA身份,请求指令$I_i$会根据认证次数而进行相应的变化. MECA运行$\mathrm{VerifyV}\left(\mathrm{Vpk},Q_i,\mathrm{QS}_i,\mathrm{BF}\right)$,如果返回结果是1,则说明身份认证请求是完整的,且成功认证车辆身份. 对于布隆过滤器本身导致身份认证错误的情况,通过将布隆过滤器位图设置的足够大,从而可将"假阳"概率降低到可以接受的范围.

如果在特定时间里,MECA没有收到车辆V重认证请求,则MECA中止服务并通知V重新申请一个新的临时身份TVID. 同时,MECA将TVID无效的信息通知DCC. 此时,车辆将在一段时间内无法继续获得服务,同时重新申请TVID的行为会被记录,如果车辆频繁多次发起重新申请TVID服务,则其会被加入黑名单. 所以,车辆为了保证自己能够获取服务,会保持定时的重新认证,从而也避免了TVID频繁更换带来的开销.

(2) 情况2:车辆从MECA管辖范围进入MECB管辖范围

在V与MECA完成认证后,它行驶进入MECB管辖的范围,需要向DCC应用新的随机数r_j,步骤如下:

M_{10}:请求身份认证,V→MECB:$\{Q_i,\mathrm{QS}_i\}$

V移动到MECB的管辖区域,并通过$\mathrm{CreReq}\left(\mathrm{Vsk},r_j,\mathrm{TVID},Q_i\right)$发起认证请求,将$\{Q_i,\mathrm{QS}_i\}$发送给MECB,在MECB的管辖范围内,应重新应用新的随机数,因此Q_i中的I_i是更新当前随机数的服务请求指令.

M_{11}:确认身份,MECB→DCC:$\{Q_i,\mathrm{QS}_i\}$

从V收到认证请求后,MECB转发车辆的身份信息给DCC.

M_{12}:重置信息,DCC→MECB:$\{\mathrm{BF},r_{j+1}\}$,DCC→V:$\{r_{j+1},\mathrm{TVID}\}$

DCC根据Q_i通过$\mathrm{VerifyV}\left(\mathrm{Vpk},Q_i,\mathrm{QS}_i,\mathrm{BF}\right)$验证V的身份后,重置随机值$r_j$,记作$r_{j+1}$,然后DCC调用ComVP(Vpk, TVID, r_{j+1}, BF),为V-MEC认证阶段计算新的验证点VP,并存储到BF中,然后DCC返回{BF, r_{j+1}}给MECB. 与此同时,DCC通过安全信道发送{r_{j+1}, TVID}给V.

M_{13}:确认身份,通过加密通信安全提供服务MEVB→V:{encrypted communication using sk}

MECB计算会话密钥$\mathrm{sk}'=\mathrm{Vp\,k}^{\mathrm{MBsk}}(\mathrm{mod}\,q)$并使用sk′加密通信愿意继续为V提供服务,V通过计算$\mathrm{MBp\,k}^{\mathrm{Vsk}}(\mathrm{mod}\,q)=\mathrm{sk}'$获得相同的会话密钥进行安全通信. 之后V的常规认证过程同其与MECA的认证步骤相同,只是需要更换请求中的MEC身份信息.

5 安全分析

本节介绍用于车辆安全访问多访问边缘计算节点的LCA协议的形式化和非形式化安全分析. 根据攻击模型,由于认证过程完全处于公开的环境中,且攻击者可以对认证过程进行窃听,所以语义安全对于LCA尤其重要,因为其可以使得攻击者除了能够获取密文外,不能够得到与其相对应明文的任何信息,所以本文在5.1节中对LCA的语义安全进行了形式化的分析.

5.1 LCA语义安全的形式化分析

根据3.4中的安全模型与随机预言机模型来证明LCA满足语义安全.

定理1 如果UHF的输出具有"完美哈希"性质,即统计意义上的不可区分性,且变色龙哈希函数抗选择明文攻击,即满足于语义安全,则LCA满足语义安全,即

$$\mathrm{Adv}_{\mathrm{A,LCA}}^{\mathrm{G}}(\delta)\leqslant\varepsilon$$

其中,ε是一个可以忽略的优势.

证明 Game 0:Game 0与3.4节中定义的理想游戏$\mathrm{G}_{\mathrm{A}}^{\mathrm{LCA}}(\delta)$相同,所以令$\Pr\left[G_0\right]$为敌手$\mathcal{A}$赢得Game 0的概率,则

$$\left|\Pr\left[G_0\right]-\frac{1}{2}\right|=\mathrm{Adv}_{\mathrm{A,LCA}}^{\mathrm{G}}(\delta)$$

Game 1:由于sk为随机选择,所以可使用等长随机字符串来代替其,则认为sk与随机字符串在统计意义上是不可区分的,所以令$\Pr\left[G_1\right]$为敌手$\mathcal{A}$赢得Game 1的概率,则

$$\left|\Pr\left[G_0\right]-\Pr\left[G_1\right]\right|\leqslant\varepsilon$$

Game 2:由于r为UHF输出,且在随机预言机模型下,UHF的输出具有统计意义上的不可区分性,因此可以使用等长字符串代替r,且认为r与随机字符串在统计意义上是不可区分的,所以令$\Pr\left[G_2\right]$为敌手$\mathcal{A}$赢得Game 2的概率,则

$$\left|\Pr\left[G_1\right]-\Pr\left[G_2\right]\right|\leqslant\varepsilon$$

Game 3:根据文献[9],变色龙哈希生成签名时满足选择明文攻击,即$\mathrm{CHColl}(\mathrm{sk},r,m,m')$的输出服从均匀分布,因此可使用等长字符串代替签名$S_b$,则认为$S_b$

与随机字符串在统计意义上是不可区分的,所以令$\Pr[G_3]$为敌手$\mathcal{A}$赢得Game 3的概率,则

$$\left|\Pr[G_2]-\Pr[G_3]\right|\leqslant\varepsilon$$

由于此时所有的可被观测的字符串均为随机字符串,所以可得

$$\left|\Pr[G_3]\right|\leqslant\varepsilon$$

综上可得

$$\begin{aligned}&\mathrm{Adv}^{G}_{A,LCA}(\delta)=\Pr[G_0]\\&=\left|\begin{matrix}\Pr[G_0]-\Pr[G_1]+\Pr[G_1]-\Pr[G_2]\\+\Pr[G_2]-\Pr[G_3]+\Pr[G_3]\end{matrix}\right|\\&\leqslant\left|\Pr[G_0]-\Pr[G_1]\right|+\left|\Pr[G_1]-\Pr[G_2]\right|\\&+\left|\Pr[G_2]-\Pr[G_3]\right|+\left|\Pr[G_3]\right|\\&=\varepsilon+\varepsilon+\varepsilon+\varepsilon\\&=\varepsilon\end{aligned}$$

证毕.

5.2　非形式化安全分析

(1)抵抗重放攻击

车辆在同一个域内进行连续认证时,每次发送的请求认证内容不同,所以会抵抗重放攻击. 而车辆每次进行跨域认证时,都需要更新随机值r_j,r_j更新过程使用UHF,保证了更新过程的不可预测性. 之后的认证过程中,由于r_j的更新将导致之前的签名值无法正确被验证,也因此保证了认证过程的新鲜度. 所以LCA协议能够抵抗上述的重放攻击.

(2)抵抗中间人攻击

在开放通道中,攻击者有可能与车辆和认证器建立独立的关联,诱使对方交换消息. 但是由于攻击者无法得到车辆的私钥,所以其无法伪造正确的签名,因而无法通过验证. 所以LCA协议能够抵抗上述的中间人攻击.

(3)车辆身份隐私保护

认证过程中除了通过安全信道进行车辆身份TVID的传输,其他通信交互过程中均没有进行TVID的传输,因此,能够保证测量身份信息的隐私性.

6　性能分析

本节将介绍LCA与现有相关协议[5~8]之间的性能比较.

LCA协议在带有8KB SARM的Arduino Mega 2560上执行,如图4所示,波特率设置为115 200. 使用的函数库是nano-ecc[14]和BigNumber[15],选择secp256作为计算开销比较的椭圆曲线.

表2给出了LCA协议和其他方案的计算开销. 通信负荷的比较如表3和图5所示,需要指出,LCA协议在连续认证阶段之前的通信开销很高,即表3给出的

图4　汽车模拟

LCA的2 560 bits,但后面只需要发送一条消息即可对于车辆进行实时认证;因而总体来讲,LCA在车辆端具有较低的计算开销,且车辆与MEC间通信开销较低,从而跨域认证切换时延低,容易部署.

表2　计算负荷比较

协议	需要的操作	总运行时间/ms
[5]	$2T_{ecc\text{-}m}+2T_{inv}+3T_h$	12 180
[6]	$18T_h+4T_{ecc\text{-}m}+T_{ecc\text{-}a}$	24 599
[7]	$16T_h$	384
[8]	$18T_h+4T_{ecc\text{-}m}$	24 512
LCA	$5T_{mul}+3T_a$	39

表3　通信负荷比较

文献方法	初次认证		重认证	
	消息数目	通信负荷	消息负载	通信负荷
[5]	10	1 600 bits	5	800 bits
[6]	6	1 536 bits	2	800 bits
[7]	3	896 bits	4	416 bits
[8]	6	2 338 bits	2	512 bits
LCA	10	2 560 bits	1	160 bits

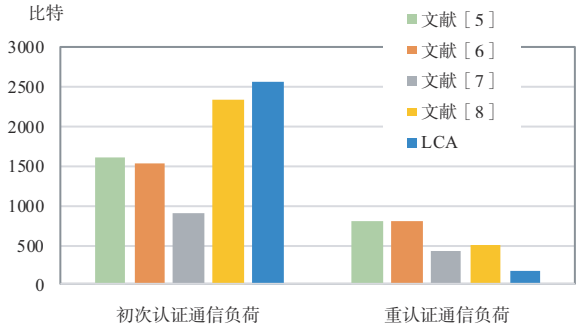


图5　通信负荷比较

大数运算执行时间:

$T_a\approx 3$ ms(加法运算);

$T_{mul}\approx 6$ ms(乘法操作);

$T_h\approx 24$ ms(SHA-256哈希函数);

$T_{inv}\approx 34$ ms(模逆运算);

$T_{ecc\text{-}m}\approx 6\,020$ ms(ECC点乘);

$T_{ecc\text{-}a}\approx 87$ ms(ECC加法运算).

7 结论

在这项研究中,提出了一种连续的轻量级认证协议. LCA 协议的实现利用变色龙哈希冲突. 实现了访问 MEC 资源的永不信任和持续认证的安全保护措施. 实验表明,LCA 协议的计算和通信成本是轻量级的,车辆只需要三次加法和两次乘法. 因此,很容易部署在资源受限的设备上. 这种低通信负载的再次认证能力也减少了服务中心的处理负荷,因此 LCA 协议适合于大规模服务请求的场景. 未来将探索车辆访问 MEC 资源的个性化访问控制策略.

参考文献

[1] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing—A key technology towards 5G[J]. ETSI White Paper, 2015, 11(11): 1-16.

[2] CHU W, WUNIRI Q, DU X, et al. Cloud control system architectures, technologies and applications on intelligent and connected vehicles: A review[J]. Chinese Journal of Mechanical Engineering, 2021, 34(1): 139.

[3] ALMAJALI S, SALAMEH H B, AYYASH M, et al. A framework for efficient and secured mobility of IoT devices in mobile edge computing[C]//2018 Third International Conference on Fog and Mobile Edge Computing (FMEC). Piscataway: IEEE, 2018: 58-62.

[4] JIA X, HE D, KUMAR N, et al. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing[J]. IEEE Systems Journal, 2019, 14(1): 560-571.

[5] SUN H P, TAN Y A, LI C W, et al. An edge-cloud collaborative cross-domain identity-based authentication protocol with privacy protection[J]. Chinese Journal of Electronics, 2022, 31(4): 721-731.

[6] SON S, LEE J, PARK Y, et al. Design of blockchain-based lightweight V2I handover authentication protocol for VANET[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(3): 1346-1358.

[7] BABU P R, REDDY A G, PALANISWAMY B, et al. EV-Auth: Lightweight authentication protocol suite for dynamic charging system of electric vehicles with seamless handover[J]. IEEE Transactions on Intelligent Vehicles, 2022, 7(3): 734-747.

[8] YU F Y, MA M D, LI X H. A blockchain-assisted seamless handover authentication for V2I communication in 5G wireless networks[C]//ICC 2021 - IEEE International Conference on Communications. Piscataway: IEEE, 2021: 1-6.

[9] YANG Z, JIN C, TIAN Y, et al. Lis: Lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. New York: ACM, 2020: 719-731.

[10] CARTER J L, WEGMAN M N. Universal classes of hash functions[C]//Proceedings of the Ninth Annual ACM Symposium on Theory of Computing. New York: ACM, 1977: 106-112.

[11] KRAWCZYK H. Chameleon signatures[C]//NDSS 2000. California: Internet Society, 2000: 143-154.

[12] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.

[13] PAGH A, PAGH R, RAO S S. An optimal Bloom filter replacement[C]//Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms. New York: ACM, 2005: 823-829.

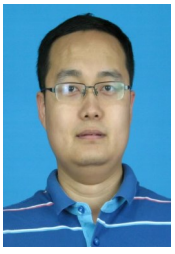
[14] nano-ecc[EB/OL]. (2023). https://github. com/iSECPartners/nano-ecc.

[15] BigNumber[EB/OL]. (2023). https://github.com/nickgammon/BigNumber/blob/master/README.md.

作者简介

邹光南 男,1977 年 9 月出生,四川省自贡市荣县人,博士,研究员. 长期从事移动通信技术和网络安全技术研究工作.

E-mail: ezgnac@163.com



尤启迪 男,1982 年 1 月出生,黑龙江省哈尔滨人,博士,研究员. 长期从事密码安全技术研究工作.

E-mail: youqd@spacestart.com.cn