# 基于 SGX 的车联网身份认证方案研究

王 冠[1,2],张倩倩[1,2]
(1. 北京工业大学 信息学部,北京 100124;
2. 北京市可信计算重点实验室,北京 100124)

**摘 要**:车联网需要实时采集和处理交通数据,缓解交通拥堵,并保护用户的隐私数据,防止被攻击者窃取和操纵。然而,目前大部分认证协议不能很好地抵抗来自内部的特权用户攻击,且没有高效地利用路边基站单元(RSU)。针对上述问题,该文设计了基于 SGX 的车联网身份认证协议。该协议将身份认证过程中主要的计算工作从 TA 卸载到 RSU 内完成,使 RSU 不再只有简单的转发功能,实现了分布式计算。通过 SGX 远程认证提供的安全通道将主密钥从 TA 传输到 RSU。并将身份认证过程中主密钥的使用过程转移到 SGX 的安全区内,利用可信硬件来存储主密钥,同时用主密钥对 TA 中的车辆关系认证表进行加密。在满足抵抗内部特权用户攻击的同时实现了计算工作的卸载。实验结果表明,该协议的计算时间减少了 23.16%,同时大大降低了 TA 的计算负载,在没有增加网络节点的情况下实现了去中心化的身份认证,具备较好的安全性和实时性。

**关键词**:车联网;身份认证;SGX;计算卸载;安全
**中图分类号**:TP309 **文献标识码**:A **文章编号**:1673-629X(2023)11-0099-07
doi:10.3969/j.issn.1673-629X.2023.11.015

# Research on Identity Authentication Scheme Based on SGX in Internet of Vehicles

WANG Guan[1,2], ZHANG Qian-qian[1,2]
(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;
2. Beijing Key Laboratory of Trusted Computing, Beijing 100124, China)

**Abstract**: IoV needs to collect and process traffic data in real time, alleviate traffic congestion, and protect users' private data from being stolen and manipulated by attackers. However, most of the existing authentication protocols cannot resist the privileged user attack from inside, and do not make efficient use of roadside Base Station Unit (RSU). In view of the above problems, we design an identity authentication protocol based on SGX for IoV. The protocol offloads the main computing work in the process of identity authentication from TA to RSU, so that RSU no longer only has a simple forwarding function and realizes distributed computing. The master key is transferred from TA to RSU through the secure channel provided by SGX Remote attestation. The use of the master key in the process of identity authentication is transferred to the secure area of SGX, and the master key is stored by the trusted hardware. At the same time, the master key is used to encrypt the vehicle relationship authentication table in TA. The offloading of computing work is achieved while satisfying the resistance to internal privileged user attacks. The experimental results show that the computing time of the proposed protocol is reduced by 23.16%, and the computational load of TA is greatly reduced. It realizes decentralized identity authentication without increasing network nodes, and has good security and real-time performance.

**Key words**: Internet of vehicles; identity authentication; SGX; computation offloading; security

## 0 引 言

随着物联网的成熟和普及,一种通过互联网连接车辆的特殊网络应运而生:车联网。车联网是物联网的一个子集,通过车对车(V2V)、车对基础设施(V2I)连接实现通信。车联网是一个复杂的综合网络系统,连接车内外的人、车载智能系统以及城市环境中的各种信息物理系统。通过将车辆、传感器和移动设备集成到一个全球网络中,车联网超越了远程信息处理、车

**收稿日期**:2023-01-06 **修回日期**:2023-05-09
**基金项目**:国家重点研发计划(2019YFB2102303);National Key Research and Development Project(2020YFB1005905)
**作者简介**:王 冠(1968-),男,硕导,副教授,通信作者,研究方向为信息安全、可信计算、数据挖掘与智能信息系统;张倩倩(1997-),女,硕士研究生,研究方向为信息安全。

辆自组织网络和智能交通,使各种服务能够交付给车辆和运输系统以及车上和车辆周围的人。车联网实时采集、处理和共享道路信息,缓解交通管控中的交通拥堵,通过预警减少交通事故,保障车辆安全[1]。传统的车联网环境主要由可信实体(TA)、路边基站单元(RSU)和车载移动单元(OBU)3 个基本网络节点组成。TA 执行车辆的注册、认证等功能,RSU 设置在道路两侧或十字路口,利用专用短程通信协议通过无线信道与车辆通信,OBU 用于采集上传的相关车辆信息[2]。

当前有大量车联网身份认证协议,它们用密码学方法结合区块链、云雾计算、边缘计算等手段,实现了车联网身份认证协议的安全高效[2-6]。

文献[3]针对多 TA 网络模型提出一种基于区块链的认证和密钥协议,将 TA 的计算负载卸载到 RSU,提高认证效率,但该协议并没有抵抗内部攻击的手段。文献[4]设计了一种基于对称加密算法和雾计算的车联网认证协议,有 4 层架构,并应用了 SGX 抵抗特权攻击,但该协议并没有有效利用 RSU。文献[5]提出一个基于边缘计算的的车联网身份认证隐私保护协议,该协议将车辆作为边缘节点参与认证,RSU 不参与认证,但是没有抵抗特权用户攻击。文献[6]提出了一种基于密钥共享和动态代理机制的去中心化车辆身份验证方案,基于信任管理的区块链聚合子认证结果,实现协同认证,存储在防篡改区块链中的信誉较高的边缘计算节点可以将最终聚合的认证结果上传到中心服务器,实现去中心化认证,但是没有抵抗内部攻击。文献[7]设计了一种基于 SGX 技术的工业物联网认证协议,该协议采用 SGX 存储主密钥,同时借助 SGX 内存保密的特点实现机密计算,该协议可以有效抵抗特权用户攻击和终端节点的追踪攻击[7]。文献[2]设计了一种基于区块链和 secGear 统一机密计算框架的车联网认证协议,在满足抗抵赖要求的同时实现了跨区域认证,并采用 secGear 框架实现了认证表的机密计算,保护了认证表的安全。

但是随着网络中车辆的激增,区块链性能开销大,逐渐不能满足认证所需的安全高效。在车联网中应用边缘计算、雾计算等新兴技术手段提高了认证的高效,但是增加了网络节点,且没有最大程度地利用 RSU。现存的大部分认证协议主要考虑了外部攻击,如重放攻击、窃听攻击、捕获攻击等。随着软件复杂性的增加,攻击者有可能会利用木马等恶意软件来获取系统内部的信息,窃取认证表,从而实现内部攻击,内部攻击危害相比于外部攻击更大,系统内部充满威胁[7]。基于此,该文设计了一种基于可信计算技术的安全高效的身份认证协议,抵抗了内部攻击,同时实现了计算的卸载。

该协议最大程度地利用 RSU,RSU 不再只是实现简单的转发功能,将身份认证过程中的主要计算工作卸载到 RSU 内完成。TA 和 RSU 实现远程认证,TA 将生成的主密钥通过远程认证实现的安全通道传输给可信的 RSU。在身份认证与密钥协商阶段,OBU 通过公共信道将信息发送给 RSU 后,RSU 根据收到的信息,从中取出车辆离线注册时留在 TA 的认证标识,发送给 TA,TA 收到此标识后,从内存的 OBU 认证表中取出对应信息发送给 RSU,在 RSU 内将 TA 发来的信息和 OBU 发来的信息进行计算对比,计算完成后,RSU 将对应的信息分别转发给 OBU 和 TA。

该协议利用 SGX 技术提供的安全机制来保证认证协议的安全性,利用 SGX 的密封机制来保存主密钥,内存隔离机制来实现机密计算,远程认证机制来安全地传输主密钥。

## 1 相关技术

IntelSGX(Intel Software Guard Extension)[8] 是对 Intel 架构的一组扩展,旨在为计算机上执行的敏感计算提供完整性和机密性保护。SGX 是一系列可信计算设计中的最新迭代,旨在通过利用远程计算机中的可信硬件来解决安全远程问题。可信硬件建立安全容器,远程计算服务用户将所需的计算和数据上传到安全容器中,可信硬件在执行计算时保护数据的机密性和完整性[9]。SGX 技术主要包括隔离执行和远程认证两个核心机制[10]。

### 1.1 隔离执行

SGX 技术允许应用程序在一个受保护的容器中执行,称为 Enclave。Enclave 是应用程序地址空间中的一个受保护区域,即使存在特权恶意软件,它也提供了机密性和完整性。如果 Enclave 外的软件试图访问 Enclave 内存区域,即使是特权软件,如虚拟机管理器、BIOS 或操作系统,也会受到阻止。SGX 阻止所有其他软件访问位于 Enclave 内的代码和数据,包括系统软件和来自其他 Enclave 的访问。检测到修改 Enclave 内容时,阻止或中止执行。

为了实现 SGX 内存保护,需要新的硬件结构。EPC(Enclave Page Cache)是用于存储 Enclave 页和 SGX 结构的保护内存,该内存受软硬件访问的保护。EPC 内存储了许多不同 Enclaves 的代码和数据,当 Enclave 执行对 EPC 的内存访问时,处理器决定是否访问。处理器在称为 EPCM(Enclave Page Cache Map)的硬件结构中维护 EPC 中每个页面的安全性和访问控制信息。EPC 为系统中的 Enclave 提供受保护的内存区域。EPCM 是附加到每个 EPC 页面的安全

元数据。EPCM 包含硬件保护 Enclave 内存访问所需的信息。EPC 页面和 EPCM 条目之间存在 1∶1 的映射。内存加密引擎(MEE)是一个硬件单元,加密处理器包主存(DRAM)之间的通信数据。创建 Enclave 时,会将 Enclave 二进制文件加载到 EPC 中,并建立 Enclave 标识。Enclave 创建过程分为多个阶段:初始化 Enclave 控制结构、分配 EPC 页并将 Enclave 内容加载到页中、测量 Enclave 内容并最终建立 Enclave 标识。加载完后,处理器会计算 Enclave 中所有内容的摘要(SHA-256),与开发者签名中的摘要进行对比。只有相匹配,才会通过完整性验证,然后执行相应应用代码。

### 1.2　远程认证

SGX 架构提供了两种机制,一种用于在同一平台上运行的2 个 Enclave 之间进行本地证明,另一种用于扩展本地证明以向平台外的第 3 方进行远程证明。

远程认证需要使用非对称加密。SGX 启用了一个特殊的 Enclave,称为 Quoting Enclave,专门用于远程证明。SGX 引入了 EPID(Enhanced Privacy ID),EPID 是一种组签名方案,它允许平台对对象进行签名,而无需唯一标识平台或链接不同的签名。Quoting Enclave 创建用于签署平台证明的 EPID 密钥,然后由 EPID 后端基础设施认证。EPID 密钥不仅代表平台,还代表底层硬件的可信度。当 Enclave 系统运行时,只有 Quoting Enclave 可以访问 EPID 密钥,并且 EPID 密钥绑定到处理器固件的版本。因此,可以看到一个 QUOTE 是由处理器本身发出的。

首先,应用程序需要来自平台外部的服务,并与服务提供商建立通信。服务提供商向应用程序发出挑战,以证明它确实要在一个或多个 Enclave 内运行必要的组件;应用程序提供 Quoting Enclave 的 Enclave 身份,并将其与服务提供商的挑战一起传递给应用程序的 Enclave;Enclave 生成一个清单,其中包括对质询的响应和一个临时生成的公钥,以供挑战者用于将秘密传送回 Enclave,然后它生成清单的哈希摘要,并将其作为 EREPORT 指令的用户数据包含在内,EREPORT 指令将生成一个将清单绑定到 Enclave 的报告,然后 Enclave 将报告发送到应用程序;应用程序将 REPORT 转发给 Quoting Enclave 进行签名;Quoting Enclave 使用 EGETKEY 指令检索其报告密钥并验证报告。Quoting Enclave 创建 QUOTE 结构并使用 EPID 密钥对其进行签名。Quoting Enclave 将 QUOTE 结构返回给应用程序;应用程序将 QUOTE 结构发送给服务挑战者;挑战者使用 EPID 公钥证书和撤销信息或认证服务来验证报价上的签名,并通过清单摘要验证清单的完整性。

## 2　身份认证协议设计

该系统包括 3 个实体:TA、RSU 和 OBU[11]。在这个模型中,TA 知道部署了哪些 RSU。车辆在接入车联网中要与 TA 进行认证,整体流程如图 1 所示。

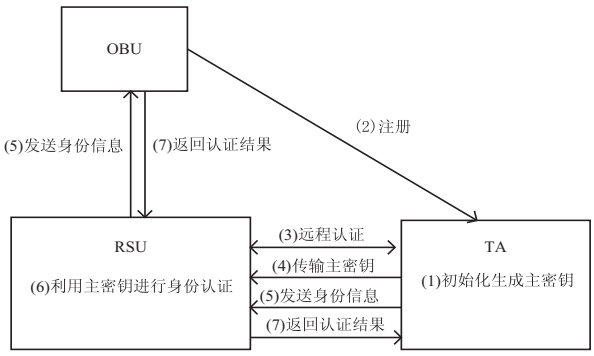


图 1　整体流程

针对车联网应用场景,该文提出一种基于 SGX 的身份认证协议,以确保车联网的数据传输和指令传输的安全高效。该协议主要包括初始化协议、注册阶段、远程认证阶段、身份认证及密钥协商阶段。

### 2.1　初始化协议

(1)超级管理员选取随机数(MK),将 MK 作为主密钥保存在 TA 中的 SGX,TA 生成全局随机数($ID_T$),RSU 内也包含 $ID_T$。

(2)超级管理员将单向哈希函数 h(),消息认证码对 $HMac_k$(Mac,Ver)等写入所有的 OBU 和 RSU 的内存中。

### 2.2　车辆注册阶段

在车辆注册阶段,用户和车辆需要在 TA 进行注册,TA 为车辆生成身份标识($ID_i$),该阶段在离线环境中进行,注册步骤如下:

(1)用户在 TA 中输入密码($PW_i$)和生物信息($BIO_i$),TA 为车辆选择身份标识($ID_i$),创建智能卡($ID_{SC}$),计算 $(\sigma_i,\tau_i)=Gen(BIO_i)$,$RPW=h(ID_i \| PW_i \| \sigma_i)$,$V=h(RPW \| ID_{SC})$。

(2)TA 生成随机数($r_s$,$K_s$),使用 $ID_T$ 计算 $N_i$,$VID_i$ 和 $F_i$,其中 $N_i=r_s \oplus h(ID_i \| PW_i \| \sigma_i \| ID_{SC})$,$VID_i=h(r_s \| ID_i \| ID_T)$,$F_i=h(ID_T \| VID_i) \oplus r_s$。

(3)TA 获取随机数($W$),并将 $W$、$VID_i$、$F_i$、$K_s$、$ID_i$ 输入到 SGX 的安全接口,安全接口使用主密钥(MK)计算并返回 $PVID_i$,$PF_i$,SW,$PK_s$,$PID_i$,其中 $PVID_i=VID_i \oplus h(W \| MK)$,$PF_i=F_i \oplus h(W \| MK)$,$PK_s=K_s \oplus h(W \| MK)$,$PID_i=ID_i \oplus h(W \| MK)$,$SW=W \oplus h(ID_T \| MK)$。

(4)TA 将$\{W, PVID_i, PF_i, PK_s, PID_i\}$存储在自己内存中的认证表中,TA 获取当前时间戳($t_1$),并将$\{N_i, V, VID_i, SW, K_s, \tau_i, ID_i, t_1\}$通过安全信道发送给 OBU,将智能卡交给用户。

(5)OBU 接收到 TA 返回的信息后,判断时间戳($t_1$)的新鲜性,如果 $t_1$ 不新鲜,则拒绝该请求并要求重发;否则,OBU 将$\{N_i, V, \mathrm{VID}_i, \mathrm{SW}, K_s, \tau_i, \mathrm{ID}_i\}$保存在自己的内存中,并给 TA 返回一条确认消息。

### 2.3 远程认证阶段

在基于 SGX 的身份认证协议中,为了确保主密钥的安全性,主密钥的部署方式必须是可信的。TA 首先需要通过认证建立与 RSU 的安全会话通道,然后将主密钥发送到 RSU 中。利用 SGX 提供的远程认证建立 TA 与 RSU 可信区的认证会话,同时通过密钥交换协议协商出双方的会话密钥。最终 TA 通过会话密钥加密主密钥并发送给 RSU。通过远程认证,RSU 中的 Enclave 可以证明自己的身份合法,未经篡改同时运行在启用了 SGX 的正版平台,这保证了主密钥的安全[12-14]。认证流程如图 2 所示。

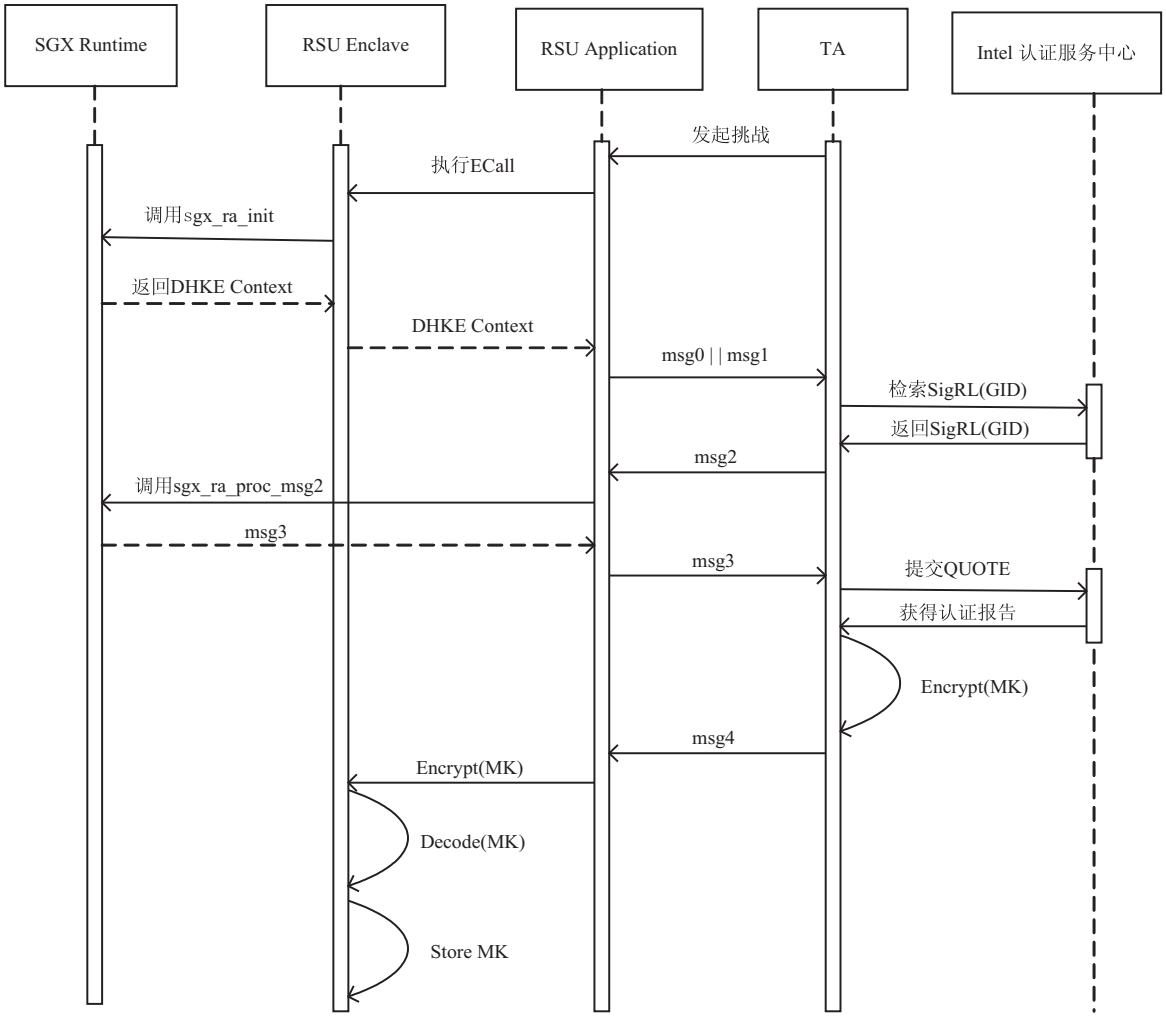


图 2 远程认证流程

认证过程如下:(1)TA 向 RSU 发起挑战,RSU 执行几个步骤来构造远程认证流的初始消息。RSU Enclave 初始化后,RSU Application 执行 Ecall 进入 Enclave,RSU Enclave 调用 sgx_ra_init( )并将结果和 DHKE Context 返回给 RSU Application,sgx_ra_init( )函数的参数为 TA 的公钥,该公钥应该硬编码到 RSU Enclave,可确保最终用户无法更改密钥,从而 Enclave 只能与预期的远程服务通信。然后,RSU Enclave 调用 sgx_get_extended_epid_group_id( )来检索 EPID 的扩展 GID 以生成 msg0。

(2)RSU 调用 sgx_ra_get_msg1( )来生成包含 DHKE 客户端公钥(Ga)的 msg1,该方法的其他参数包括在上一步中获得的 DHKE Context 和一个指向用于计算客户端 DHKE 密钥的 sgx_ra_get_ga( )存根函数的指针。当 RSU Enclave 链接到 sgx_tkey_exchange 库并导入 sgx_tkey_exchange 时,该函数由 SDK 自动生成。将 msg0 和 msg1 一起发送给 TA, msg0 ‖ msg1 = (ExGID ‖ Ga ‖ GID),其中“ ‖ ”表示连接。

(3)从 RSU 接收 msg1 后,TA 检查请求中的值,生成自己的 DHKE 参数,并向 Intel 认证服务中心(IAS)发送查询用以检索客户端发送的 Intel EPID GID 的签名撤销列表(SigRL),Intel 认证服务只支持扩展 GID 的值为 0。TA 使用 P-256 曲线生成一个随机的 EC 密钥(Gb),通过 Ga 和 Gb 派生出密钥

(KDK),对 KDK 执行 AES-128CMAC 算法派生出 SMK。TA 将 msg2 = (Gb ‖ SPID ‖ Quote_Type ‖ KDF_ID ‖ SigSP(Ga,Gb)) ‖ $CMAC_{SMK}$(Gb ‖ SPID ‖ Quote_Type ‖ KDF_ID ‖ SigSP(Ga,Gb)) ‖ SigRL(GID)发给 RSU。其中 SPID 为服务提供商 ID,Quote_Type 为客户端请求的 Quote 类型(0x0 表示不可链接,0x1 表示可链接)。KDF_ID 通常为 0x1,使用服务提供商的 EC 私钥计算 ECDSA 签名得到 SigSP(Ga,Gb)。

(4)RSU 接收到 msg2 后,应用程序调用 sgx_ra_proc_msg2( )函数来生成 msg3。该调用执行以下任务。验证 TA Enclave 的签名、检查 SigRL、返回 msg3,其中包含用于验证特定 Enclave 的 Quote,msg3 = $CMAC_{SMK}$(Ga ‖ Ps_Security_Prop ‖ Quote) ‖ Ga ‖ Ps_Security_Prop ‖ Quote。

(5)TA 接收到 msg3 后,需要做以下操作。验证 msg3 中的 Ga 是否与 msg1 中的 Ga 匹配;验证 $CMAC_{SMK}$(M);验证 Quote 中的前 32 个字节是否匹配 SHA-256 摘要(Ga ‖ Gb ‖ VK),其中 ‖ 表示连接;验证 RSU 提供的认证证据,验证证据要求服务提供者向 IAS 提交 QUOTE 并获得认证报告。该报告由 IAS 报告签名私钥签名,服务提供者必须使用 IAS 报告签名公钥验证该签名。TA 从 KDK 派生出 128 比特的会话密钥,用会话密钥通过该 AES/GCM/NoPadding 算法加密要传输的主密钥。TA 将验证结果和加密的 MK 作为 msg4 发送给 RSU[15]。

(6)RSU 收到 msg4 后在 Enclave 中调用 sgx_ra_get_keys( )获取会话密钥,并调用 SGX 加密库提供的 AES 解密函数 sgx_rijindael128GCM_decrypt( )解密出 MK,然后通过 SGX 密封函数 sgx_seal_data( )密封 MK。

### 2.4　用户登录认证与密钥协商阶段

在该阶段中,用户首先进行登录,登录成功后进入身份认证阶段,TA、RSU 和 OBU 三方都要进行认证,并协商出会话密钥,该阶段如图 3 所示。

(1)用户在车辆的设备终端输入登录需要的身份信息,登录验证通过后进入车辆身份认证与密钥协商阶段。车辆在进入车联网区域时,车辆的 OBU 通过不安全的网络信道将$\{Q_1, M_1, \mathrm{SW}, t_1\}$发送给 RSU。

(2)RSU 接收到 OBU 发送的信息后,取出 SW 并将其输入到 RSU 的 SGX 内计算并返回 $W$,将 $W$ 发送给 TA,TA 从其 OBU 的认证表中取得$\{W, \mathrm{PVID}_i, \mathrm{PF}_i, \mathrm{PK}_s, \mathrm{PID}_i\}$发送到 RSU 内,RSU 将 $W$, $\mathrm{PVID}_i$, $\mathrm{PF}_i$, $\mathrm{PK}_s$, $\mathrm{PID}_i$ 输入到 SGX 内进行计算,判断验证码。计算 S2,通过公共信道发送$\{Q_2, M_2, \mathrm{SW}^*, t_2\}$给 OBU,将$\{W^*, \mathrm{PVID}_i^{\mathrm{new}}, \mathrm{PF}_i^*, \mathrm{PTK}_s, \mathrm{PID}_i, t_2\}$发送给 TA。

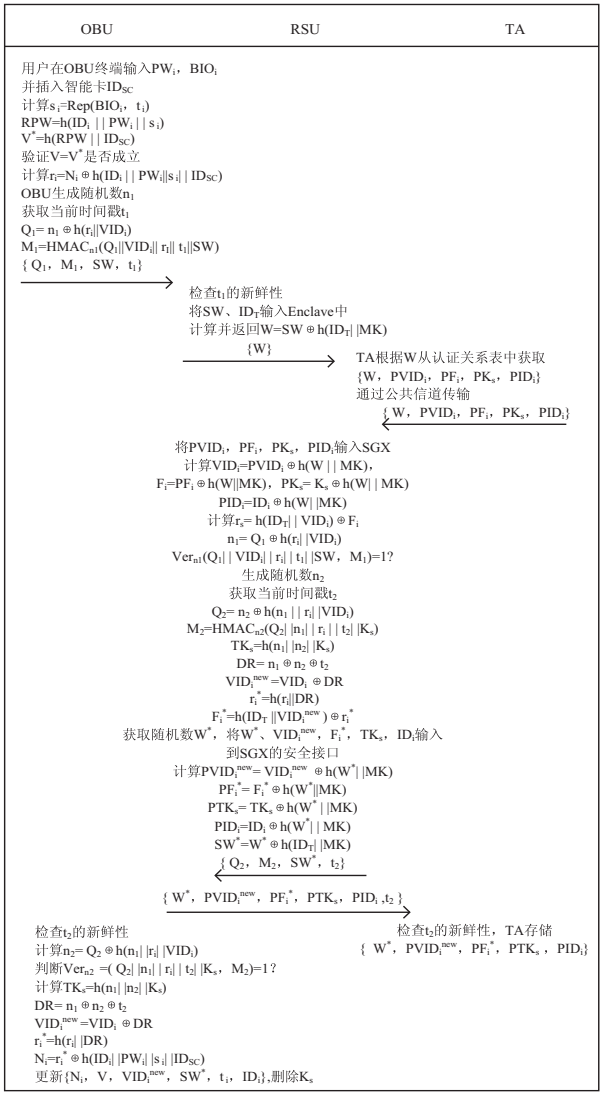


图 3　三方身份认证阶段

## 3　安全性分析

### 3.1　重放攻击

协议使用时间戳($t_1$、$t_2$)来抵抗重放攻击,在每次接收到消息时,会先检查时间戳的新鲜性,如果时间戳不新鲜,丢弃此消息要求重发。

### 3.2　特权内部攻击

该协议主密钥存储在 SGX 内,TA 中的 OBU 认证表的内容都经过主密钥的加密,包括存储的会话密钥、OBU 的 ID 等敏感信息,即使内部特权用户获得认证表,也无法获取有效消息。且 RSU 中的计算都是在 SGX 中进行,特权用户攻击者也无法窥得该机密计算的过程。

### 3.3　追踪攻击

该协议使用的变量在每个会话中都是动态变化的,其中的重要参数$\{Q_1, M_1, \mathrm{SW}\}$,在每一轮的认证过程中都是不同的,包括在 TA 和 RSU 之间传输的$\{W, \mathrm{PVID}_i, \mathrm{PF}_i, \mathrm{PK}_s, \mathrm{PID}_i\}$也是在认证过后实时

更新的。不存在可以一直追踪的常量,因此攻击者无法追踪某个参数进而获取敏感信息。

### 3.4 窃听攻击

攻击者可以在网络中截获{ $Q_1$,$Q_2$,$M_1$,$M_2$,SW, $PVID_i$ , $PF_i$ ,$PK_s$, $PID_i$ },其中攻击者要想从 $Q_1$,$Q_2$ 中获得敏感信息,需要获取 $r_i$ , $VID_i$ ,但这两个参数不在网络中传输。$M_1$,$M_2$ 已经被单向哈希函数加密,从中无法获取有效信息。SW 经过主密钥的计算, $PVID_i$ , $PF_i$ ,$PK_s$, $PID_i$ 均已经通过主密钥进行加密计算,因此也无法获取有效信息。

### 3.5 认证表泄露攻击

在该协议中,认证表存储在内存中,假设认证表泄露,认证表中的内容{ $W$ , $PVID_i$ , $PF_i$ ,$PK_s$, $PID_i$ }即使被攻击这获取,他也无法获取有效内容。因为 $W$ 为随机数,在下一次认证时 $W$ 也会更新,并且在此次认证后的会话过程中也无法通过 $W$ 获取到会话密钥或者其他有效信息,{ $PVID_i$ , $PF_i$ ,$PK_s$, $PID_i$ }都已经通过主密钥进行计算,因此也无法获取到有效信息。综上,该协议抵抗了认证表泄露攻击。

## 4 性能分析

在比较计算成本时,通过模拟实验估算协议中每个实体的计算时间,其中用安卓手机来模拟车辆,用联想笔记本电脑来模拟 RSU 和 TA。两台设备的配置如表 1 所示,得出的运行时间如表 2 所示,其中定义 $t_h$ 为哈希函数的运算时间, $t_{sym}$ 为一次对称加解密的运算时间, $t_e$ 为一次模指数运算的时间。

表 1 实验环境

| 设备 | 荣耀 x30i | 联想笔记本电脑 |
|---|---|---|
| 操作系统 | Android | Windows10 |
| CPU | Media TekMT6833P | Intel(R)Core (TM)i5-6200UCPU |
| 内存 | 8 GB | 8 GB RAM |

表 2 实验结果 ms

| 操作 | OBU/ms | RSU/TA/ms |
|---|---|---|
| $t_h$ | 0.008 2 | 0.005 0 |
| $t_{sym}$ | 0.026 0 | 0.023 0 |
| $t_e$ | 1.188 1 | 0.911 0 |

该协议将 TA 中的主要计算工作卸载到 RSU 内进行,实现了分布式计算,解决了可信中心(TA)负载过大的问题。在分析计算成本时,需要分别分析车辆(OBU)、路边单元(RSU)和可信中心(TA)的计算成本,其中简单的异或运算⊕小到忽略不计,HAMC 签名所用时间与哈希函数所用时间相同。TA 将主密钥通过 Intel SGX 远程认证建立的安全通道传输给 RSU,这个操作只在系统远程认证时部署一次,不会在系统运行过程中带来时间消耗。系统中使用 SGX 主要带来模式切换开销和访存开销,在文献[12]中经过评测得到模式切换开销仅为 8 μs,访存开销为 3 μs,故在以下的计算分析中忽略 SGX 的计算开销。

### 4.1 计算开销分析

计算消耗的比较结果如表 3 所示。从表中可以看出,文中协议的计算消耗是最低的,并且实现了 TA 的计算卸载,因此文中协议在具备安全性的保证下计算耗能也更低,实现了分布式计算,具备车联网协议要求的高效及时性。

表 3 计算开销对比

| 协议 | OBU | RSU | TA | 总开销/ms |
|---|---|---|---|---|
| 文献[11] | $t_e+5t_h+t_{sym}$ | $t_h$ | $t_e+5t_h+t_{sym}$ | 2.219 1 |
| 文献[2] | $6t_h+2t_{MAC}$ | $t_h$ | $11t_h+2t_{MAC}$ | 0.135 6 |
| 文中协议 | $4t_h+2t_{MAC}$ | $9t_h+2t_{MAC}$ | | 0.104 2 |

### 4.2 通信开销分析

在分析通信开销时,假设时间戳的长度为 32 bit,随机数、密钥、身份标识的长度为 160 bit,哈希函数和对称加解密的输出为 256 bit,得出通信开销对比如表 4 所示。

表 4 通信开销对比

| 协议 | 总开销/bit |
|---|---|
| 文献[11] | 1 728 |
| 文献[2] | 2 208 |
| 文中协议 | 2 816 |

在计算开销和通信开销在总成本中占比相等的情况下进行对比,文中协议的计算开销远小于文献[11]的计算开销,与文献[2]相比,文中的通信开销增加了21.59%,但计算开销减少了 23.16%。由此得出文中协议优于以上两种协议。同时,文献[2]在模型中设置了多个 TA,增加了网络节点。文中协议实现了分布式计算,将 TA 的计算负载卸载到 RSU 中完成,同时也没有在网络中增加节点,相较于其他协议降低了成本。

## 5　结束语

车联网的身份认证协议目前多数使用基于区块链的方式,或者使用复杂的加密运算方式,或是采用在网络中增加云雾节点等方式,使得计算通信开销过大。为了解决这个问题,文中协议在车联网中使用了可信计算(SGX)技术,将 TA 的计算负载卸载到 RSU 中完成,实现了分布式计算,大大地降低了 TA 的计算成本,实现了去中心化的身份认证,通过与现有的认证协议进行对比,该协议的计算开销有所减少。另外,该文采用了 SGX 技术,将主密钥保存在由可信硬件保护的环境中,并利用主密钥对 TA 中的车辆关系认证表进行加密,防止了认证表泄露攻击,同时抵御了内部特权用户的攻击。

**参考文献:**


[1] YANG F, WANG S, LI J, et al. An overview of Internet of vehicles[J]. China Communications, 2014, 11(10): 1-15.

[2] 刘　忻,王家寅,杨浩睿,等. 一种基于区块链和 secGear 框架的车联网认证协议[J]. 信息网络安全,2022,22(1):27-36.

[3] XU Zisang, LIANG Wei, LI Kuan-Ching, et al. A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles[J]. Journal of Parallel and Distributed Computing, 2021, 149: 29-39.

[4] WU Tsu-Yang, GUO Xinglan, CHEN Yeh-Cheng, et al. SGXAP: SGX-based authentication protocol in IoV-enabled fog computing[J]. Symmetry, 2022, 14(7): 1393.

[5] ZHANG J, ZHONG H, CUI J, et al. Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7940-7954.

[6] LIU H, ZHANG P, PU G, et al. Blockchain empowered cooperative authentication with data traceability in vehicular edge computing[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4221-4232.

[7] 刘　忻,郭振斌,宋宇宸. 一种基于 SGX 的工业物联网身份认证协议[J]. 信息网络安全,2021,21(6):1-10.

[8] COSTAN V, DEVADAS S. Intel SGX explained[J]. Cryptology ePrint Archive, 2016, 2016(86): 1-118.

[9] 王　冠,梁世豪. 基于 SGX 的 Hadoop KMS 安全增强方案[J]. 信息安全研究,2019,5(6):514-520.

[10] 董春涛,沈晴霓,罗　武,等. SGX 应用支持技术研究进展[J]. 软件学报,2021,32(1):137-166.

[11] YING B, NAYAK A. Anonymous and lightweigh authentication for secure vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2017, 66(12): 10626-10636.

[12] 李明煜,夏虞斌,陈海波. 面向 SGX2 代新型可信执行环境的内存优化系统[J]. 软件学报,2022,33(6):2012-2029.

[13] 王　冠,苗艺雪. 基于 Intel SGX 的 Kerberos 安全增强方案[J]. 信息安全研究,2021,7(4):374-383.

[14] 赵　波,袁安琪,安　杨. SGX 在可信计算中的应用分析[J]. 网络与信息安全学报,2021,7(6):126-142.

[15] ANATI I, GUERON S, JOHNSON S. Innovative technology for CPU based attestation and sealing[C]//Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy. New York: ACM, 2013: 7.