无人机网络中基于无证书聚合签名技术的检错批量认证与密钥协商协议

郭超[1]，黄子琛[2]，弓丞[3]，刘培鹤[1]
（1. 北京电子科技学院电子与通信工程系，北京 100070；
2. 北京电子科技学院网络空间安全系，北京 100070；
3. 联通（北京）产业互联网有限公司，北京 100038）

摘　要：在以无人机为中继节点的地空协同通信架构中，实现高效且安全的身份认证与密钥协商是保障系统可靠性的核心问题。针对传统批量认证机制在处理非法用户设备发起的无效接入请求时存在的认证失败与检错能力不足等问题，提出一种基于无证书聚合签名的检错型批量认证与密钥协商协议。该协议包括双向身份认证与密钥协商机制，以及集成群组测试方法的非法请求快速识别模块，可显著提升认证效率。在安全性分析方面，在随机预言机模型下对协议的不可伪造性进行了证明，并借助形式化验证工具完成形式化安全验证。性能评估结果表明，所提方案在通信开销、计算开销及检错复杂度等方面性能优越，适用于大规模无人机网络中资源受限场景下的安全通信需求。

关键词：无人机网络；无证书聚合签名；批量认证；群组测试
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2025204

An error-detectable batch authentication and key agreement protocol based on certificateless aggregate signature for UAV networks

GUO Chao[1], HUANG Zichen[2], GONG Cheng[3], LIU Peihe[1]
1. Department of Electronics and Communication Engineering, Beijing Institute of Electronic Science and Technology, Beijing 100070, China
2. Department of Cyberspace Security, Beijing Institute of Electronic Science and Technology, Beijing 100070, China
3. Unicom (Beijing) industrial Internet Co. Ltd., Beijing 100038, China

Abstract: In geospatial cooperative communication architectures where unmanned aerial vehicles are employed as in-

收稿日期：2025−05−18；**修回日期：**2025−06−17
通信作者：刘培鹤，lph@besti.edu.cn
基金项目：国家重点研发计划项目（No.2023YFB3106505）；中央高校基本科研业务费资助项目（No.3282023001，No. 3282024052）
Foundation Items: The National Key Research and Development Program of China (No. 2023YFB3106505), The Fundamental Research Funds for the Central (No.3282023001, No.3282024052)

termediate nodes, achieving efficient and secure identity authentication and key agreement is regarded as a critical issue for ensuring system reliability. To address the limitations of traditional batch authentication mechanisms—particularly their failure to handle invalid access requests from unauthorized user equipment and their insufficient error detection capability—an error-detectable batch authentication and key agreement protocol based on certificateless aggregate signatures was proposed. The protocol was designed to incorporate mutual identity authentication, key agreement mechanisms, and a group testing-based module for rapid identification of illegitimate access requests, thereby significantly enhancing authentication efficiency. For security analysis, the unforgeability of the protocol was proven under the random oracle model, and its security was verified using formal verification tools. Performance evaluations were conducted, demonstrating that the proposed scheme outperformed existing solutions in terms of communication overhead, computational cost, and error detection complexity. The results suggest that the protocol is well-suited for secure communication in large-scale UAV networks operating under resource-constrained environments.

Key words: UAV network, certificateless aggregate signature, batch authentication, group testing

0　引言

随着无人机（unmanned aerial vehicle，UAV）技术的迅猛发展，UAV网络被广泛部署于物流运输、农作物监测、遥感、灾害管理等多个关键领域[1]。UAV可作为空中中继节点，协助地面用户终端（user equipment，UE）集群进行高效通信。5G及未来网络环境的普及，进一步拓展了UAV网络的应用边界。然而，由于其固有的高动态性、节点资源受限以及通信链路不稳定性[2]，UAV网络在通信安全与效率方面面临诸多挑战。例如，多个UE同时向UAV发起接入认证请求将显著增加UAV的计算负载，同时，系统还易受到如重放攻击、伪造攻击等安全威胁。因此，亟须设计一种高效、安全且具备容错能力的批量认证协议，以实现UAV与UE之间的快速双向认证与密钥协商。

现有的批量认证协议在容错性方面普遍存在不足。若接入请求中存在任意一个无效认证信息，往往导致整批认证失败，从而降低整体认证效率。一些引入错误检测机制的改进方案，在实践中又面临难以准确定位无效签名、检错过程复杂且消耗资源多等问题，无法满足UAV网络中对高实时性与高可靠性的双重需求。此外，当前主流认证体系如基于公钥基础设施（public key infrastructure，PKI）和身份签名（identity-based signature，IBS）等，仍存在证书管理烦琐或密钥托管风险等结构性缺陷，限制了其在动态异构环境中的适应性与扩展性。而基于无证书密码（certificateless signature scheme，CLS）的认证体系主要面对点对点通信，难以解决UE集群同时接入UAV网络的高并发环境。

为解决上述问题，本文提出了一种结合无证书聚合签名（certificateless aggregate signature，CLAS）技术[3]与群组测试方法[4]的高效批量认证与密钥协商协议。CLAS技术有效融合了无证书密码体制与聚合签名的优势，不仅避免了传统PKI系统中的证书管理问题，还可将多个签名聚合为1个短签名，从而显著降低验证开销，特别适用于计算能力受限的UAV场景。为提升协议的容错能力，本文进一步设计了基于群组测试的检错机制，在批量认证失败时可快速检测并定位无效接入认证请求，避免有效请求被误拒，提高整体认证效率与系统鲁棒性。

本文的主要贡献如下。

（1）设计了高效的批量认证机制：基于CLAS技术，构建了一种支持并发接入、无须双线性对运算的批量认证协议，显著降低UAV计算开销并提升认证效率。

（2）提出了快速检错机制：引入群组测试思

想，设计了一种定位精度高、资源开销小的无效请求检测方法，有效解决了传统方案中无法准确识别错误请求以及检错次数多的问题，显著提升了批量认证的成功率和系统稳定性。

1 相关工作

近年来，为应对UAV网络中日益增长的通信安全需求，大量研究聚焦于身份认证机制[5-6]与批量认证协议[7]的设计。现有研究主要可归纳为以下3类：基于传统加密体系的认证方案[8-9]、基于CLAS的高效批量认证方案[10-11]以及具备容错功能的检错批量认证方案[7, 12]。

基于传统加密体系的认证协议主要包括基于PKI、IBS和CLS的安全认证机制。PKI通过数字证书绑定公钥与身份，具备成熟的密钥管理体系，广泛应用于无人机通信环境。Jadhav等[13]提出了一种基于PKI的轻量级认证与加密机制，用于计算资源受限的地空链路环境。Alladi等[14]则设计了SecAuthUAV协议，采用PKI实现了无人机与地面站、无人机之间的双向认证，增强了UAV网络的安全性和灵活性。然而，PKI方案存在密钥管理复杂、证书撤销难度高等问题，在无人机高动态网络中部署成本较高。为解决证书管理问题，提升认证系统在动态网络中的部署效率，研究者提出了基于IBS的方案。该方法通过用户的唯一身份信息生成公钥，避免了传统PKI体系中复杂的证书分发与撤销操作。Jan等[15]提出了一种融合IBS与聚合签名的认证协议，面向军事级场景，在提升验证效率的同时简化了密钥管理。Wani等[16]构建了一种面向异构网络中UAV通信系统的基于IBS的身份认证机制。然而，以上基于IBS的方案存在密钥托管问题。为解决PKI方案中证书管理问题以及IBS方案中密钥托管问题，研究人员提出了CLS方案。CLS体系通过将私钥拆分为用户私有部分与密钥生成中心（key generation center，KGC）生成的部分，既消除了对证书的依赖，又避免了传统IBS方案中的密钥托管风险，成为近年来UAV通信环境中重要的认证机制。Semal等[17]提出了一种适用于不可信UAV网络的CLS群组认证与密钥协商协议。该方案支持在资源受限、节点异构的UAV环境中实现低计算开销、高安全性的双向认证。Li等[18]进一步提出了无须双线性对运算的轻量化CLS认证协议，通过使用配对免除结构大幅提升了UAV网络的认证效率与协议可部署性。虽然，基于CLS的方案在认证效率与密钥安全方面取得良好平衡。但是，这些方案主要面向点对点认证场景，难以支持UE集群同时接入UAV的高并发环境，在实际应用中效率有限。

为了适应UE集群同时接入UAV的高并发环境，研究人员将CLS与聚合签名技术相结合，提出了CLAS机制。Iqbal等[10]提出了适用于车联网环境下的基于CLAS技术的批量认证方案。Samra等[11]结合CLAS技术提出了用于车联网的条件隐私保护认证方案，但是其中的双线性对运算对签名认证带来了一些延迟。为了提高签名生成和认证的效率，Ali等[19]基于椭圆曲线密码技术提出了车联网中无双线性对的基于CLAS的条件隐私保护方案，并实现了批量认证。然而，上述CLAS方案虽可实现高效批量认证，但普遍忽视了认证失败后的无效请求识别问题，缺乏在认证失败场景后对无效请求进行识别与定位的能力，一旦存在单个无效请求，仍需整体重复认证，资源浪费严重，限制了协议在实际高密度并发接入中的应用效果。因此，如何在CLAS框架下引入容错机制与高效检错能力，成为进一步研究的关键方向。

针对批量认证的容错问题，现有研究通过尝试引入错误检测机制以识别无效签名或者引入容错阈值在存在一定无效签名的情况下仍然可以通过认证。在错误检测机制方面，Hartung等[12]提出了基于$d-$不相交矩阵的检错方法，验证过程

可输出有效消息列表，但方案复杂、计算负载大，并且限制了最大出错的签名个数。Bardini等[20]提出了一种基于嵌套无覆盖族的无界容错聚合签名方案，对于出错签名个数没有限制，但是，在进行检错时，存在检错过程复杂，需要检错次数多的问题。在引入容错阈值方面，Wang等[7]提出了面向UAV网络的结合CLS与模糊批量验证机制的认证方案，引入了容错属性提高认证效率。但是，该方案无法检测出无效的签名。根据以上相关工作可知，现有容错批量认证方案在错误识别效率、计算复杂度与实际部署能力方面仍存在较大改进空间，尤其缺乏适用于资源受限UAV网络场景的轻量化检错机制。

本文结合CLAS与群组测试，设计了一种支持快速错误定位的批量认证与密钥协商协议。该方案在提升整体认证效率的同时，显著降低了UAV在处理无效请求时的计算与通信开销，增强了协议的实用性。

2　系统模型和问题描述

2.1　系统模型

本文在UAV网络的环境下构建了应用于UAV与UE集群通信的批量认证与密钥协商协议。模型中包括UE、UAV、控制中心（control center，CC）和KGC这4个实体，系统模型如图1所示。

在系统模型中，CC是完全可信的，负责整个系统的初始化、用户注册、为UE生成假名并分发假名以及追踪非法用户身份。KGC负责为UAV生成公私钥对，为UE生成部分私钥。UE作为通信发起方，向UAV发送接入认证请求，并接收来自UAV的响应消息以完成对UAV合法身份的认证。在系统建立时，UE需将自身真实身份发送至CC完成注册，通信过程中通过假名参与认证，保障匿名性。UAV负责接收多个UE的接入请求，聚合多个认证请求，利用CLAS机制降低认证开销，完成批量认证并与每个UE协商会话密钥。

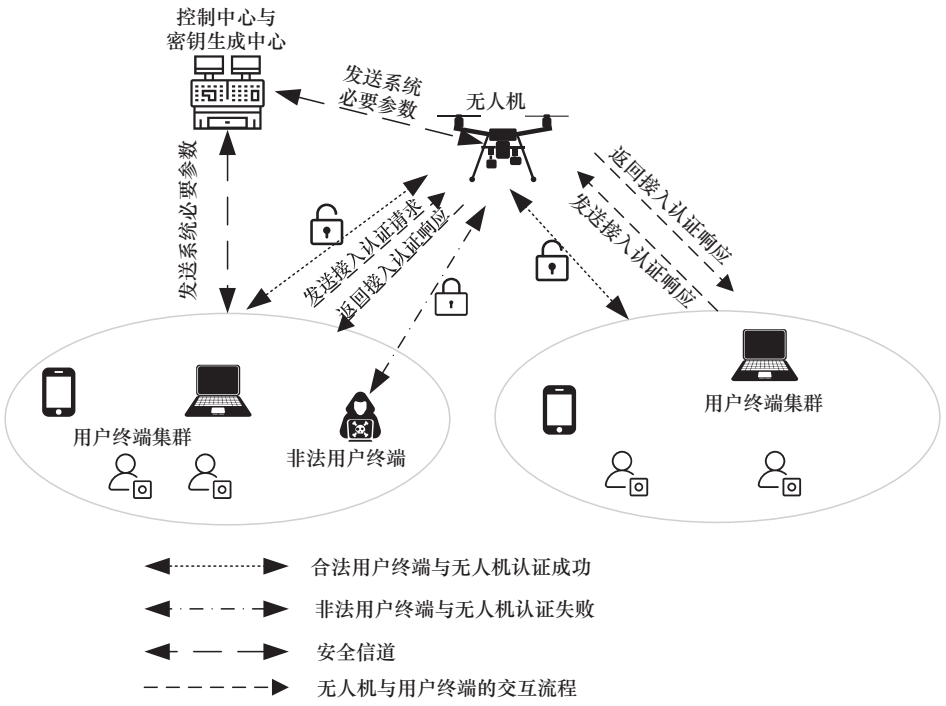


图1　系统模型

在此模型之下，采用批量认证支持UE集群通过UAV节点并发接入的场景。为缓解UAV在批量认证中的计算压力，系统采用CLAS机制进行聚合签名认证；同时，为有效解决非法UE接入带来的批量认证失败问题，系统引入基于群组测试的错误检测机制，从而在保持认证效率的同时，保障认证的安全性与容错性。

2.2 安全模型

为系统地评估协议在随机预言机模型下的安全性，本节描述了随机预言机模型，定义了攻击者的能力边界，并基于此定义了方案满足安全性的条件。

2.2.1 攻击者类别

根据攻击者的行为，考虑以下2种类型的攻击者。

第1类攻击者Adv_1：攻击者可以破坏UE的秘密值，或能够替换任意UE的公钥为其选择的值，但无法访问KGC的主密钥。

第2类攻击Adv_2：攻击者可以访问KGC的主密钥，但无法替换任何UE的公钥，无法为UE生成局部私钥。

2.2.2 预言机

本节定义了攻击者Adv可以访问以下5个预言机。

（1）揭示部分私钥预言机：在接收到Adv查询后，挑战者ξ将PID_i作为输入，计算得到psk_{PID_i}，并发送给Adv。

（2）创建用户预言机：在接收到Adv的查询后，挑战者ξ将PID_i作为输入，计算得到vpk_{PID_i}，并发送给Adv。

（3）揭示私钥预言机：在接收到Adv的查询后，挑战者ξ将PID_i作为输入，计算得到vsk_{PID_i}，并发送给Adv。

（4）替换公钥预言机：Adv输入PID_i和vpk'_{PID_i}，将现在的公钥vpk_{PID_i}替换为需要的公钥vpk'_{PID_i}。

（5）签名预言机：在收到Adv的查询后，预言机根据PID_i，vpk_{PID_i}和消息m，返回一个有效的签名σ。

2.2.3 安全定义

本节通过针对挑战者ξ与攻击者Adv定义的2个博弈来对方案的不可伪造性进行定义。

博弈1在挑战者ξ与攻击者Adv_1之间进行，具体步骤如下。

（1）初始化阶段：在初始化阶段，挑战者ξ运行设置算法得到参数、秘密值s和主公钥。然后，挑战者ξ将参数和主公钥发送给攻击者Adv_1，并保留秘密值s。

（2）查询阶段：在查询阶段，Adv_1向揭示部分私钥预言机、创建用户预言机、揭示私钥预言机、替换公钥预言机和签名预言机进行查询。

（3）伪造阶段：最后，Adv_1输出对应于消息m_i^*，PID_i^*和$vpk_{PID_i}^*$的签名σ^*。当满足以下条件时，Adv_1赢得这个博弈。（a）σ^*是一个有效的签名；（b）PID_i^*没有调用到部分私钥预言机与私钥预言机；（c）在博弈中签名预言机从未被调用。

定义1 如果没有攻击者Adv_1以不可忽略的概率在多项式时间t内赢得博弈1，则这个认证方案满足第一类安全。

博弈2在挑战者ξ和Adv_2之间进行，具体步骤如下。

（1）初始化阶段：挑战者ξ运行初始化算法生成系统参数、秘密值s和主公钥。并将系统参数和主公钥提供给Adv_2。

（2）查询阶段：Adv_2向创建用户预言机、揭示私钥预言机和签名预言机进行查询。

（3）伪造阶段：最后，Adv_2输出对应于消息m_i^*，PID_i^*和$vpk_{PID_i}^*$的签名σ^*。当满足以下条件时，Adv_2赢得这个博弈。（a）σ^*是一个有效的签名；（b）PID_i^*没有调用揭示私钥预言机以获取私

钥；（c）签名预言机从未被调用过。

定义 2　如果没有攻击者 Adv_2 以不可忽略的概率在多项式时间 t 内赢得博弈 2，则这个认证方案满足第二类安全。

2.3　安全目标

为了确保认证协议在 UAV 网络环境中的安全性，本文协议设计必须满足一系列核心安全目标。具体安全目标如下。

（1）消息可认证性：接收者接收到签名消息后可通过认证来确认该消息的合法性和完整性，即该消息由合法用户发送且未被恶意攻击者修改或篡改。

（2）身份条件隐私性：消息发送者的真实身份在通信过程中是匿名的，除了 CC 没有任何第三方能从匿名中获取其真实身份。

（3）可追踪性：当消息来源存在争议或者导致事故发生需要追究责任人时，CC 能够得到消息发送者的真实身份。

（4）不可关联性：恶意攻击者无法关联到两条或多条消息来自同一实体。

（5）抵抗多种传统攻击：攻击者不能通过某种传统攻击得到相关信息从而达到其目的，例如重放攻击、伪造攻击、中间人攻击等。

3　基于无证书聚合签名技术的批量认证与密钥协商协议

在本节中将详细介绍所提出的批量认证方案。方案分为 6 个阶段，分别是系统建立、假名与部分私钥生成、UE 生成密钥对、相互认证与密钥协商、批量认证与密钥协商、批量认证检错。

3.1　系统建立阶段

在系统建立阶段，CC 与 KGC 产生系统必要参数。CC 与 KGC 选择 2 个大素数 p，q，生成椭圆曲线 $E{:}y^2=x^3+ax+b \bmod p$，其中 $a,b\in Z_p^*$ 并且 $\left(4a^3+27b^2\right)\bmod p\neq 0$。CC 选择无穷远的点 P 作为椭圆曲线的基点。KGC 选择随机数 $s\in Z_p^*$ 作为主私钥，并计算主公钥 $P_{\mathrm{pub}}=s\cdot P$。CC 选择随机数 $b\in Z_p^*$ 作为主私钥，并计算主公钥 $T_{\mathrm{pub}}=b\cdot P$。并且，KGC 为 UAV 生成公私钥对。KGC 和 CC 选择 3 个哈希函数 $H_1,H_2,H_3{:}\{0,1\}\rightarrow Z_p^*$。KGC 随机选择 $q_{\mathrm{UAV}}\in Z_p^*$，计算 $\mathrm{pk}_{\mathrm{UAV}}=q_{\mathrm{UAV}}\cdot P$ 作为 UAV 的公钥，计算 $\mathrm{sk}_{\mathrm{UAV}}=q_{\mathrm{UAV}}+h_1\left(\mathrm{ID}_{\mathrm{UAV}}\|\ \mathrm{pk}_{\mathrm{UAV}}\right)\cdot s$ 作为 UAV 的私钥。UE 将真实身份 RID_i 发送到 CC 进行注册。最后，CC 与 KGC 公开系统参数 $\left\{P,p,q,E,G,H_1,H_2,H_3,P_{\mathrm{pub}},T_{\mathrm{pub}},\mathrm{pk}_{\mathrm{UAV}}\right\}$。

3.2　假名与部分私钥生成

在该阶段 CC 为 UE 生成假名，KGC 为 UE 生成部分私钥。首先，UE 随机选择 t_i 并计算 $\mathrm{PID}_{i,1}=t_i\cdot P$，$\mathrm{Key}_i=t_i\cdot T_{\mathrm{pub}}\oplus\mathrm{RID}_i$，发送假名请求消息 $\left\{\mathrm{PID}_{i,1},\mathrm{Key}_i\right\}$ 到 CC。CC 收到假名请求消息后，首先计算 $\mathrm{Key}_i\oplus b\mathrm{PID}_{i,1}\stackrel{?}{=}\mathrm{RID}_i$，来验证 UE 的合法身份。验证通过之后，为 UE 计算假名 $\mathrm{PID}_{i,2}=\mathrm{RID}_i\oplus H_1\left(b\mathrm{PID}_{i,1},\Delta T_i\right)$，假名 $\mathrm{PID}_i=\left\{\mathrm{PID}_{i,1},\mathrm{PID}_{i,2},\Delta T_i\right\}$。其中，$\Delta T_i$ 为假名的有效期。CC 通过安全信道将 PID_i 发送给 KGC。KGC 收到后，随机选择 r_i，计算 $R_i=r_i\cdot P$。为 UE 计算部分私钥 $\mathrm{psk}_{\mathrm{PID}_i}=\left(r_i+s\cdot K_i\right)\bmod p$，其中 $K_i=H_1\left(\mathrm{PID}_i,R_i,P_{\mathrm{pub}}\right)$。KGC 发送 $\left\{\mathrm{psk}_{\mathrm{PID}_i},R_i,\mathrm{PID}_i\right\}$ 到 UE。UE 收到后，验证 $\mathrm{psk}_{\mathrm{PID}_i}\cdot P\stackrel{?}{=}R_i+K_i\cdot P_{\mathrm{pub}}$，验证消息的完整性与有效性。

3.3　UE 生成密钥对

在该阶段 UE 生成自己的公私钥对。UE 随机选择 x_i 作为秘密值，计算 $X_i=x_i\cdot P$，$Q_i=R_i+H_2\left(\mathrm{PID}_i,X_i\right)X_i$。UE 的公钥为 $\mathrm{vpk}_{\mathrm{PID}_i}=\left(Q_i,R_i\right)$，私钥为 $\mathrm{vsk}_{\mathrm{PID}_i}=\left(\mathrm{psk}_{\mathrm{PID}_i},x_i\right)$。

3.4　相互认证与密钥协商阶段

该阶段为单一 UE 与 UAV 进行相互认证与密

钥协商。具体流程主要包括以下步骤：UE首先生成临时公私钥，并构造接入认证请求消息发送至UAV。UAV在接收到请求后，对假名有效性及时间戳进行验证，随后依据签名信息验证UE身份的合法性。认证通过后，UAV生成自身的临时密钥并计算会话密钥，随后返回接入认证响应消息。UE在收到响应后完成认证并计算会话密钥，最终双方完成相互认证与密钥协商过程，UE与UAV相互认证与密钥协商流程如图2所示。

首先，UE发送接入认证请求消息到UAV。UE选择随机数$\mathrm{Tsk}_{\mathrm{UE}_i}$作为临时私钥，并计算临时公钥$\mathrm{Tpk}_{\mathrm{UE}_i}=\mathrm{Tsk}_{\mathrm{UE}_i}\cdot P$。UE随机选择$u_i$，计算$U_i=u_i\cdot P$，$h_{2i}=H_2(\mathrm{PID}_i,X_i)$，$h_{3i}=H_3(\mathrm{PID}_i,M_i,\mathrm{vpk}_{\mathrm{PID}_i},U_i,T_i)$，$S_i=\left[\mathrm{Tsk}_{\mathrm{UE}_i}+u_i+h_{3i}\left(\mathrm{psk}_{\mathrm{PID}_i}+h_{2i}\cdot x_i\right)\right]\bmod q$，$\sigma_i=(U_i,S_i)$。其中，$T_i$为时间戳，$M_i$为要签名的消息。UE发送接入认证请求$\left\{\mathrm{PID}_i,\mathrm{vpk}_{\mathrm{PID}_i},M_i,T_i,\sigma_i,\mathrm{Tpk}_{\mathrm{UE}_i}\right\}$到UAV。

UAV收到接入认证请求之后，首先验证假名PID_i处于有效期，验证时间戳T_i。然后，计算$K_i=H_1(\mathrm{PID}_i,R_i,P_{\mathrm{pub}})$、$h_{3i}=H_3(\mathrm{PID}_i,M_i,\mathrm{vpk}_{\mathrm{PID}_i},U_i,T_i)$，验证$S_i\cdot P\stackrel{?}{=}\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}(Q_i+K_iP_{\mathrm{pub}})$，

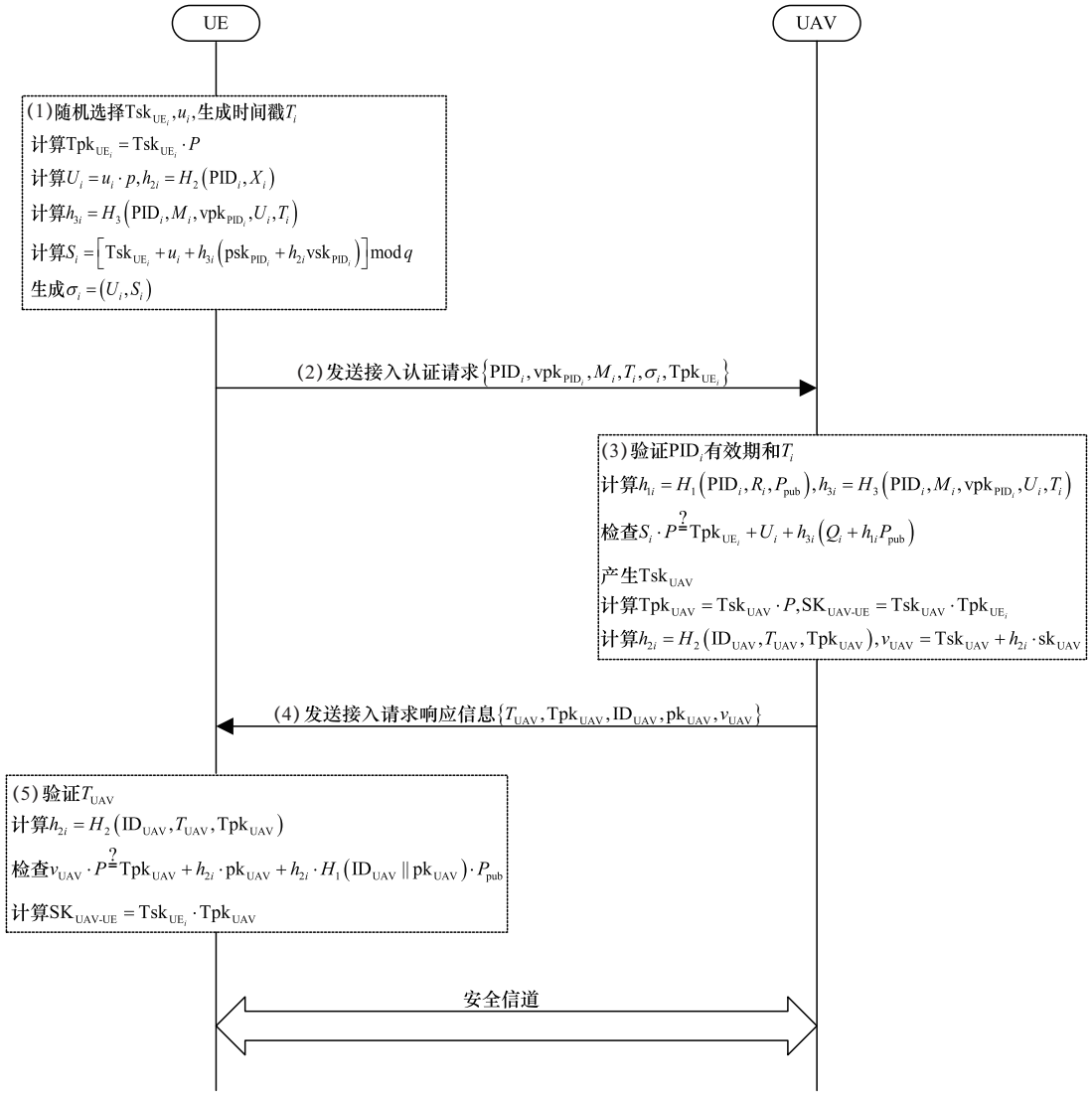


图2　UE与UAV相互认证与密钥协商流程

确定UE的合法身份。若验证失败，则终止认证过程。验证通过后，UAV产生随机数$\mathrm{Tsk}_{\mathrm{UAV}}$作为临时私钥，并计算临时公钥$\mathrm{Tpk}_{\mathrm{UAV}}=\mathrm{Tsk}_{\mathrm{UAV}}\cdot P$。UAV计算与UE之间的会话密钥$\mathrm{SK}_{\mathrm{UAV-UE}}=\mathrm{Tsk}_{\mathrm{UAV}}\cdot\mathrm{Tpk}_{UE_i}$。UAV计算$h_{2i}=H_2(\mathrm{ID}_{\mathrm{UAV}},T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}})$，$v_{\mathrm{UAV}}=\mathrm{Tsk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{sk}_{\mathrm{UAV}}$。其中，$T_{\mathrm{UAV}}$为当前时间戳。UAV发送接入认证响应消息$\{T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}},\mathrm{ID}_{\mathrm{UAV}},\mathrm{pk}_{\mathrm{UAV}},v_{\mathrm{UAV}}\}$到UE。

正确性分析：在UAV对单个消息进行认证的过程中，通过计算$K_i=H_1(\mathrm{PID}_i,R_i,P_{\mathrm{pub}})$、$h_{3i}=H_3(\mathrm{PID}_i,M_i,\mathrm{vpk}_{\mathrm{PID}_i},U_i,T_i)$，使用系统参数$P_{\mathrm{pub}}=s\cdot P$、UE的临时公钥$\mathrm{Tpk}_{\mathrm{UE}_i}=\mathrm{Tsk}_{\mathrm{UE}_i}\cdot P$、UE产生的随机数$U_i=u_i\cdot P$以及UE的公钥$\mathrm{vpk}_i=(Q_i,R_i)$，验证UE的合法身份。由$\mathrm{psk}_{\mathrm{PID}_i}=r_i+s\cdot K_i$、$Q_i=R_i+H_2(\mathrm{PID}_i,X_i)X_i$和$X_i=x_i\cdot P$可得式（1）。若式（1）成立，则说明UE为合法用户，认证通过；否则，说明UE为非法用户，认证失败。

$$\begin{aligned}&S_i\cdot P=\left[\mathrm{Tsk}_{\mathrm{UE}_i}+u_i+h_{3i}\left(\mathrm{psk}_{\mathrm{PID}_i}+h_{2i}x_i\right)\right]\cdot P=\\&\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(r_i\cdot P+s\cdot P\cdot K_i+h_{2i}x_i\cdot P\right)=\\&\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(R_i+K_iP_{\mathrm{pub}}+X_ih_{2i}\right)=\\&\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(Q_i+K_iP_{\mathrm{pub}}\right)\end{aligned}\tag{1}$$

UE接收到接入认证响应消息之后，首先验证时间戳T_{UAV}。然后，计算$h_{1i}=H_1(\mathrm{ID}_{\mathrm{UAV}},\mathrm{pk}_{\mathrm{UAV}})$、$h_{2i}=H_2(\mathrm{ID}_{\mathrm{UAV}},T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}})$，验证等式$v_{\mathrm{UAV}}\cdot P\stackrel{?}{=}\mathrm{Tpk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{pk}_{\mathrm{UAV}}+h_{2i}\cdot h_{1i}\cdot P_{\mathrm{pub}}$，确定UAV的合法身份。若验证失败，则终止接入认证过程。验证通过后，计算与UE之间的会话密钥$\mathrm{SK}_{\mathrm{UAV-UE}}=\mathrm{Tsk}_{\mathrm{UE}_i}\cdot\mathrm{Tpk}_{\mathrm{UAV}}$。至此，单个UE完成了接入认证，并与UAV之间完成了会话密钥的协商。

正确性分析：在UE对UAV的合法身份进行认证的过程中，通过计算$h_{1i}=H_1(\mathrm{ID}_{\mathrm{UAV}},\mathrm{pk}_{\mathrm{UAV}})$、$h_{2i}=H_2(\mathrm{ID}_{\mathrm{UAV}},T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}})$，使用系统参数$P_{\mathrm{pub}}=s\cdot P$、UAV的临时公钥$\mathrm{Tpk}_{\mathrm{UAV}}=\mathrm{Tsk}_{\mathrm{UAV}}\cdot P$、UAV的公钥$\mathrm{pk}_{\mathrm{UAV}}=q_{\mathrm{UAV}}\cdot P$，验证UAV的合法身份。由$\mathrm{sk}_{\mathrm{UAV}}=q_{\mathrm{UAV}}+h_{1i}\cdot s$可得式（2）。若式（2）成立，则说明UAV为合法的，认证通过；否则，说明UAV为非法的，认证失败。

$$\begin{aligned}&v_{\mathrm{UAV}}\cdot P=\left(\mathrm{Tsk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{sk}_{\mathrm{UAV}}\right)\cdot P=\\&\mathrm{Tpk}_{\mathrm{UAV}}+h_{2i}\cdot\left(q_{\mathrm{UAV}}+h_{1i}\cdot s\right)\cdot P=\\&\mathrm{Tpk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{pk}_{\mathrm{UAV}}+h_{1i}\cdot P_{\mathrm{pub}}\end{aligned}\tag{2}$$

3.5　批量认证与密钥协商阶段

当UAV收到多个UE的接入认证请求之后，对这些接入认证请求进行批量认证。当UAV收到$\mathrm{UE}_1,\mathrm{UE}_2,\cdots,\mathrm{UE}_n$的$n$个接入认证请求$\{\mathrm{PID}_i,\mathrm{vpk}_{\mathrm{PID}_i},M_i,T_i,\sigma_i,\mathrm{Tpk}_{\mathrm{UE}_i}\}$之后。首先，依次验证假名$\mathrm{PID}_i$是否处于有效期，依次验证时间戳$T_i$，计算$K_i=H_1(\mathrm{PID}_i,R_i,P_{\mathrm{pub}})$，$h_{3i}=H_3(\mathrm{PID}_i,M_i,\mathrm{vpk}_{\mathrm{PID}_i},U_i,T_i)$。然后，将这些接入认证请求消息进行聚合，计算$S=\sum_{i=1}^{n}S_i$。聚合签名为$\sigma_{\mathrm{agg}}=(U_1,U_2,\cdots,U_n,S)$。UAV通过验证计算式$S\cdot P\stackrel{?}{=}\sum_{i=1}^{n}\mathrm{Tpk}_{\mathrm{UE}_i}+\sum_{i=1}^{n}U_i+\sum_{i=1}^{n}h_{3i}\left(Q_i+K_i\cdot P_{\mathrm{pub}}\right)$来进行批量认证。若认证失败，说明UE集群中存在非法用户，UAV将进行批量认证检错。认证通过后，UAV与每一个UE进行会话密钥的协商。UAV产生随机数$\mathrm{Tsk}_{\mathrm{UAV}}$作为临时私钥，并计算临时公钥$\mathrm{Tpk}_{\mathrm{UAV}}=\mathrm{Tsk}_{\mathrm{UAV}}\cdot P$。UAV计算与每一个UE的会话密钥$\mathrm{SK}_{\mathrm{UAV-UE}_i}=\mathrm{Tsk}_{\mathrm{UAV}}\cdot\mathrm{Tpk}_{\mathrm{UE}_i}$。然后，UAV计算$h_{2i}=H_2(\mathrm{ID}_{\mathrm{UAV}},T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}})$，$v_{\mathrm{UAV}}=\mathrm{Tsk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{sk}_{\mathrm{UAV}}$。其中，$T_{\mathrm{UAV}}$为UAV的时间戳。UAV向每一个UE发送接入认证响应消息$\{T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}},\mathrm{ID}_{\mathrm{UAV}},\mathrm{pk}_{\mathrm{UAV}},v_{\mathrm{UAV}}\}$。

UE接收到接入认证响应消息之后，首先验证时间戳，计算$h_{1i}=H_1(\mathrm{ID}_{\mathrm{UAV}},\mathrm{pk}_{\mathrm{UAV}})$、$h_{2i}=H_2(\mathrm{ID}_{\mathrm{UAV}},T_{\mathrm{UAV}},\mathrm{Tpk}_{\mathrm{UAV}})$，通过验证等式$v_{\mathrm{UAV}}\cdot P\stackrel{?}{=}\mathrm{Tpk}_{\mathrm{UAV}}+h_{2i}\cdot\mathrm{pk}_{\mathrm{UAV}}+h_{2i}\cdot h_{1i}\cdot P_{\mathrm{pub}}$来确定UAV

的合法身份。若验证失败，则终止认证过程。若验证通过，则计算与UAV之间的会话密钥$SK_{UAV-UE_i}=Tsk_{UE_i}\cdot Tpk_{UAV}$。至此，UAV完成了对于UE的批量认证，并与每一个通过认证的UE建立了会话密钥。

正确性分析：在UAV对多个消息的聚合签名$\sigma_{agg}=(U_1,U_2,\cdots,U_n,S)$进行验证的过程中，通过计算$K_i=H_1(PID_i,R_i,P_{pub})$、$h_{3i}=H_3(PID_i,M_i,vpk_{PID_i},U_i,T_i)$，使用系统参数$P_{pub}=s\cdot P$、每一个UE的临时公钥$Tpk_{UE_i}=Tsk_{UE_i}\cdot P$、每一个UE产生的随机数$U_i=u_i\cdot P$以及每一个UE的公钥$vpk_i=(Q_i,R_i)$，验证聚合签名的有效性。由$psk_{PID_i}=r_i+s\cdot K_i$、$Q_i=R_i+H_2(PID_i,X_i)X_i$和$X_i=x_i\cdot P$可得式（3）。由式（3）可得式（4）。若式（4）成立，则说明每一个UE都为合法用户，批量认证通过；否则，说明UE集群中存在非法用户，批量认证失败，UAV将进行批量认证检错。

$$\begin{aligned}&(psk_{PID_i}+h_{2i}vsk_{PID_i})\cdot P=\\&r_i\cdot P+s\cdot P\cdot K_i+h_{2i}\cdot x_i\cdot P=\\&R_i+K_iP_{pub}+X_i\cdot h_{2i}=Q_i+K_iP_{pub}\end{aligned}\tag{3}$$

$$\begin{aligned}&S\cdot P=\sum_{i=1}^{n}S_i\cdot P=\\&\sum_{i=1}^{n}\left[Tsk_{UE_i}+u_i+h_{3i}(psk_{PID_i}+h_{2i}x_i)\right]\cdot P=\\&\sum_{i=1}^{n}Tpk_{UE_i}+\sum_{i=1}^{n}U_i+\sum_{i=1}^{n}h_{3i}(Q_i+K_i\cdot P_{pub})\end{aligned}\tag{4}$$

3.6 批量认证检错阶段

当批量认证验证不通过时，进入批量认证检错阶段。对于n个签名，最多有d个无效签名的情况下，使用基于(d,d)–可分解矩阵的群组测试的方法来检测无效的签名。

首先，构造(d,d)–可分解矩阵$\boldsymbol{M}$。矩阵$\boldsymbol{M}$为$2t\times n$的矩阵。其中，t为d的倍数。对于矩阵$\boldsymbol{M}$的每一列，任取t/d行，将其对应的值设为1，其余行设置为0，来构造矩阵$\boldsymbol{M}$。当$t\geq 2d\log(en/d)+\log n$时，按照上面方法构造出来的矩阵是$(d,d)$–可分解的。

然后，进行2个阶段的检错过程。按照矩阵$\boldsymbol{M}$将这些签名分别聚合为$2t$个签名。矩阵$\boldsymbol{M}$的每一行代表一个聚合签名，由对应列为1的签名聚合而成。在第1阶段，将这些签名聚合为$2t$个签名进行验证，得到$2d$个可能无效的签名。在第2阶段，对这$2d$个签名单独进行验证得到无效的签名列表，完成无效签名的检测。

4 安全性分析

在本章中，对提出的方案进行了非形式化安全分析、证明了本文方案在随机预言机模型下是不可伪造的以及使用Tamarin对方案进行了形式化验证。

4.1 非形式化安全分析

本节从UAV网络的安全需求方面，采用非形式化的方法分析本文方案所满足的安全目标。

消息可认证性。在方案中，UE发送接入认证请求消息到UAV之前需经过签名S_i，UAV通过验证$S_i\cdot P\overset{?}{=}Tpk_{UE_i}+U_i+h_{3i}(Q_i+h_{1i}P_{pub})$来确定消息是否来自合法的UE，是否被攻击者篡改或伪造。UAV发送接入认证响应消息到UE之前需经过签名v_{UAV}，UE通过验证$v_{UAV}\cdot P\overset{?}{=}Tpk_{UAV}+h_{2i}\cdot pk_{UAV}+h_{2i}\cdot H_1(ID_{UAV}\|\ \|pk_{UAV})\cdot P_{pub}$来确定消息是否来自合法的UAV，是否被攻击者篡改或伪造。

身份条件隐私保护。在方案中，UE使用假名$PID_i=\{PID_{i,1},PID_{i,2},\Delta T_i\}$进行通信，攻击者无法从假名中得到UE的真实身份。

可追踪性。在方案中，当遇到争议情况时，CC能够通过计算$RID_i=PID_{i,2}\oplus H_1(bPID_{i,1},\Delta T_i)$由假名$PID_i=\{PID_{i,1},PID_{i,2},\Delta T_i\}$来得到UE的真实身份，其中，$b$是CC的私钥。因此，方案具有

可追踪性。

不可关联性。在方案中，UE 发送$\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, M_i, T_i, \sigma_i, \mathrm{Tpk}_{\mathrm{UAV}_i}\}$到UAV。由于签名$\sigma_i$中存在随机数$u_i$，因此攻击者无法将同一UE的两条消息关联起来。因此，方案满足不可关联性。

抵抗多种传统攻击。

重放攻击。在方案中，签名生成阶段利用时间戳T_i或者T_{UAV}来确保每一次签名都是最新消息，消息接收者通过验证时间戳来检测消息的新鲜性，抵抗重放攻击。

伪造攻击。在方案中，UE发送接入认证请求消息到UAV之前需经过签名S_i。UAV发送接入认证响应消息到UE之前需要经过签名v_{UAV}。因此，可以检测出消息是否来自于合法的UE，抵抗伪造攻击。

中间人攻击。在方案中，UAV与UE之间发送消息之前都需要进行签名，攻击者无法伪造成合法的UAV与UE。即使攻击者在接入认证期间窃听消息，攻击者与无法获得UAV与UE之间的会话密钥。因此，方案可以抵抗中间人攻击。

4.2　安全证明

根据第2.2节定义的两类攻击者Adv_1和Adv_2，本节通过挑战者ξ与攻击者Adv_1、Adv_2之间的博弈来证明本文方案的安全性。

定理1　在随机预言机模型下，假设椭圆曲线离散对数问题（elliptic curve discrete logarithm problem，ECDLP）是困难的，则本文方案在第一类攻击者Adv_1的适应性选择消息攻击、选择身份攻击以及公钥替换攻击下是安全的。

证明　假设有一个攻击者Adv_1能够伪造一个消息$\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, M_i, T_i, \sigma_i=(U_i, S_i), \mathrm{Tpk}_{\mathrm{UE}_i}\}$，通过构造一个挑战者$\xi$，运行子程序$\mathrm{Adv}_1$以不可忽略的概率解决ECDLP。给定一个ECDLP问题的实例$(P, Q=s\cdot P)$，挑战者ξ将模拟预言机与攻击者Adv_1进行如下博弈。

（1）系统初始化阶段。挑战者ξ执行系统初始化程序，随机选取$s\in Z_q^*$，计算$P_{\mathrm{pub}}=s\cdot P$。令$P_{\mathrm{pub}}\leftarrow Q$，生成系统公共参数$\{P, p, q, E, G, H_1, H_2, H_3, P_{\mathrm{pub}}, T_{\mathrm{pub}}\}$，并将参数发送给攻击者$\mathrm{Adv}_1$。

（2）H_3询问。挑战者ξ建立列表L_{H_3}，初始为空，元素类型为$\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$。攻击者$\mathrm{Adv}_1$请求关于$\mathrm{PID}_i$的查询时，挑战者$\xi$查询列表$L_{H_3}$是否存在$\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$。若存在，挑战者$\xi$返回$h_{3i}$。若不存在，挑战者$\xi$选择一个随机值$h_{3i}\in Z_q^*$作为应答，并将$\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$存入列表$L_{H_3}$中。

（3）用户创建。挑战者ξ建立列表L_{PK}，初始为空，元素类型为$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$。攻击者$\mathrm{Adv}_1$请求关于$\mathrm{PID}_i$的查询时，挑战者$\xi$查询列表$L_{\mathrm{PK}}$是否存在$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$。若存在，挑战者$\xi$返回$\mathrm{vpk}_{\mathrm{PID}_i}$。若不存在，执行以下操作：挑战者$\xi$随机选取$r, b, x_i\in Z_q^*$，设置$R_i=rP_{\mathrm{pub}}+bP$，$K_i=-r \bmod q$，$\mathrm{psk}_{\mathrm{PID}_i}=b$，$\mathrm{vsk}_{\mathrm{PID}_i}=x_i$，$\mathrm{vpk}_i=x_i\cdot P$。等式$\mathrm{psk}_{\mathrm{PID}_i}\cdot P=R_i+K_iP_{\mathrm{pub}}$成立，将$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$存入列表$L_{\mathrm{PK}}$中，并返回$\mathrm{vpk}_{\mathrm{PID}_i}$给$\mathrm{Adv}_1$。

（4）部分私钥。攻击者Adv_1请求关于PID_i的部分私钥查询时，挑战者ξ查询列表L_{PK}是否存在$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$。若存在，挑战者$\xi$返回$\mathrm{psk}_{\mathrm{PID}_i}$。若不存在，挑战者$\xi$返回$\perp$。

（5）用户私钥。攻击者Adv_1请求关于PID_i的私钥查询时，挑战者ξ查询列表L_{PK}是否存在$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$。若存在，挑战者$\xi$返回$\mathrm{vsk}_{\mathrm{PID}_i}$。若不存在，挑战者$\xi$返回$\perp$。

（6）用户公钥替换。攻击者Adv_1请求关于PID_i的公钥替换查询时，挑战者ξ查询列表L_{PK}是否存在$\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\}$。若存在，挑战者$\xi$将$\mathrm{vpk}_{\mathrm{PID}_i}$更新为$\mathrm{vpk}'_{\mathrm{PID}_i}$。

（7）签名查询。攻击者 Adv_1 请求关于 PID_i 的签名查询时，挑战者 ξ 计算 S_i，使其满足 $S_i \cdot P = \mathrm{Tpk}_{\mathrm{UE}_i} + U_i + h_{3i}\left(Q_i + K_i P_{\mathrm{pub}}\right)$，返回 S_i 给攻击者 Adv_1。

（8）输出。最后，攻击者 Adv_1 输出 $\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, M_i, T_i, \sigma_i = (U_i, S_i), \mathrm{Tpk}_{\mathrm{UE}_i}\}$。满足式 $S_i \cdot P = \mathrm{Tpk}_{\mathrm{UE}_i} + U_i + h_{3i}\left(Q_i + K_i P_{\mathrm{pub}}\right)$。攻击者 Adv_1 重放以上过程伪造生成另一组有效消息 $\left\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, M_i, T_i, \sigma_i = \left(U_i, S_i^*\right), \mathrm{Tpk}_{\mathrm{UE}_i}\right\}$，消息满足式 $S_i^* \cdot P = \mathrm{Tpk}_{UE_i} + U_i + h_{3i}\left(Q_i + K_i^* P_{\mathrm{pub}}\right)$。根据这 2 个等式可以得到：

$$\left(S_i - S_i^*\right) \cdot P = h_{3i}\left(K_i - K_i^*\right) P_{\mathrm{pub}} = h_{3i}\left(K_i - K_i^*\right) \cdot s \cdot P \tag{5}$$

$$S_i - S_i^* = \left(K_i - K_i^*\right) \cdot s \tag{6}$$

挑战者 ξ 输出 $s = \left(K_i - K_i^*\right)^{-1}\left(S_i - S_i^*\right)$ 以解决给定的 ECDLP 实例，否则博弈失败。

因此，在随机预言机模型和 ECDLP 假设下，该方案在第一类攻击者 Adv_1 的攻击下是安全的。

定理 2 在随机预言机模型下，假设 ECDLP 是困难的，则本文方案在第二类攻击者 Adv_2 的适应性选择消息攻击、选择身份攻击下是安全的。

证明 假设有一个攻击者 Adv_2 能够伪造一个消息 $\left\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, M_i, T_i, \sigma_i = (U_i, S_i), \mathrm{Tpk}_{\mathrm{UE}_i}\right\}$，通过构造一个挑战者 ξ，运行子程序 Adv_2 以不可忽略的概率解决 ECDLP。给定一个 ECDLP 问题的实例 $\left(P, Q = x \cdot P\right)$，挑战者 ξ 将模拟预言机与攻击者 Adv_2 进行如下博弈。

（1）系统初始化阶段。挑战者 ξ 执行系统初始化程序，随机选取 $s \in Z_q^*$，计算 $P_{\mathrm{pub}} = s \cdot P$。令 $P_{\mathrm{pub}} \leftarrow Q$，生成系统公共参数 $\{P, p, q, E, G, H_1, H_2, H_3, P_{\mathrm{pub}}, T_{\mathrm{pub}}\}$，并将公共参数和 s 发送给攻击者 Adv_2。

（2）H_2 询问。挑战者 ξ 建立列表 L_{H_2}，初始为空，元素类型为 $\{M_i, \mathrm{PID}_i, T_i, h_{2i}\}$。攻击者 Adv_2 请求关于 PID_i 的查询时，挑战者 ξ 查询列表 L_{H_2} 是否存在 $\{M_i, \mathrm{PID}_i, T_i, h_{2i}\}$。若存在，挑战者 ξ 返回 h_{2i}。若不存在，挑战者 ξ 选择一个随机值 $h_{2i} \in Z_q^*$ 作为应答，并将 $\{M_i, \mathrm{PID}_i, T_i, h_{2i}\}$ 存入列表 L_{H_2} 中。

（3）H_3 询问。挑战者 ξ 建立列表 L_{H_3}，初始为空，元素类型为 $\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$。攻击者 Adv_2 请求关于 PID_i 的查询时，挑战者 ξ 查询列表 L_{H_3} 是否存在 $\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$。若存在，挑战者 ξ 返回 h_{3i}。若不存在，挑战者 ξ 选择一个随机值 $h_{3i} \in Z_q^*$ 作为应答，并将 $\{M_i, \mathrm{PID}_i, t_i, h_{3i}\}$ 存入列表 L_{H_3} 中。

（4）用户创建。挑战者 ξ 建立列表 L_{PK}，初始为空，元素类型为 $\left\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\right\}$。攻击者 Adv_2 请求关于 PID_i 的查询时，挑战者 ξ 查询列表 L_{PK} 是否存在 $\left\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\right\}$。若存在，挑战者 ξ 返回 $\mathrm{vpk}_{\mathrm{PID}_i}$。若不存在，执行以下操作：挑战者 ξ 随机选取 $r, x_i^* \in Z_q^*$，设置 $R_i = r \cdot P$，$K_i = H_1\left(\mathrm{PID}_i, R_i, P_{\mathrm{pub}}\right)$，$\mathrm{psk}_{\mathrm{PID}_i} = \left(r_i + sK_i\right) \bmod q$，$\mathrm{vsk}_{\mathrm{PID}_i} = \perp$，$x_i \cdot P = x_i^* \cdot Q$。等式 $\mathrm{psk}_{\mathrm{PID}_i} \cdot P = R_i + K_i P_{\mathrm{pub}}$ 成立，将 $\left\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\right\}$ 存入列表 L_{PK} 中，并返回 $\mathrm{vpk}_{\mathrm{PID}_i}$ 给 Adv_2。

（5）部分私钥。攻击者 Adv_2 请求关于 PID_i 的部分私钥查询时，挑战者 ξ 查询列表 L_{PK} 是否存在 $\left\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\right\}$。若存在，挑战者 ξ 返回 $\mathrm{psk}_{\mathrm{PID}_i}$。若不存在，挑战者 ξ 执行用户创建询问并返回 $\mathrm{psk}_{\mathrm{PID}_i}$。

（6）用户私钥。攻击者 Adv_2 请求关于 PID_i 的私钥查询时，挑战者 ξ 查询列表 L_{PK} 是否存在 $\left\{\mathrm{PID}_i, \mathrm{psk}_{\mathrm{PID}_i}, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{vsk}_{\mathrm{PID}_i}\right\}$。若存在，挑战者 ξ

返回$\mathrm{vsk}_{\mathrm{PID}_i}$。若不存在，挑战者$\xi$返回$\perp$。

（7）签名查询。攻击者Adv_2请求关于PID_i的签名查询时，挑战者ξ计算S_i，使其满足$S_i\cdot P=\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(Q_i+K_iP_{\mathrm{pub}}\right)$，返回$S_i$给攻击者$\mathrm{Adv}_2$。

（8）输出。最后，攻击者Adv_2输出$\left\{\mathrm{PID}_i,\mathrm{vpk}_{\mathrm{PID}_i},M_i,T_i,\sigma_i=\left(U_i,S_i\right),\mathrm{Tpk}_{\mathrm{UE}_i}\right\}$。满足等式$S_i\cdot P=\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(\mathrm{psk}_{\mathrm{PID}_i}\cdot P+h_{2i}x_i\cdot P\right)$。攻击者$\mathrm{Adv}_2$重放以上过程伪造生成另一组有效消息$\left\{\mathrm{PID}_i,\mathrm{vpk}_{\mathrm{PID}_i},M_i,T_i,\sigma_i=\left(U_i,S_i^*\right),\mathrm{Tpk}_{\mathrm{UE}_i}\right\}$，消息满足等式$S_i^*\cdot P=\mathrm{Tpk}_{\mathrm{UE}_i}+U_i+h_{3i}\left(\mathrm{psk}_{\mathrm{PID}_i}\cdot P+h_{2i}^*x_i\cdot P\right)$。根据这2个等式可以得到：

$$\left(S_i-S_i^*\right)\cdot P=h_{3i}\left(h_{2i}-h_{2i}^*\right)x_i\cdot P=h_{3i}\left(h_{2i}-h_{2i}^*\right)\cdot x_i^*\cdot xP \tag{7}$$

$$S_i-S_i^*=h_{3i}\left(h_{2i}-h_{2i}^*\right)\cdot x_i^*x \bmod q \tag{8}$$

挑战者ξ输出$x=\left(h_{3i}\left(h_{2i}-h_{2i}^*\right)\cdot x_i^*\right)^{-1}\left(S_i-S_i^*\right)$以解决给定的ECDLP实例，否则博弈失败。

因此，在随机预言机模型和ECDLP假设下，该方案在第二类攻击者Adv_2的攻击下是安全的。

4.3　形式化验证

本节中，使用形式化验证工具Tamarin对所提出的方案进行形式化分析。Tamarin是基于多重集重写规则和一阶逻辑的符号分析工具。Tamarin内置了Dolev-Yao模型以及一系列支持常见加密函数的安全理论。在Tamarin中，All代表全称量化，Ex代表存在量化，#代表时间变量的前缀，$\mathrm{Fr}(\sim x)$代表新鲜值x的生成，$F\left(t_1,t_2,\cdots,t_n\right)$代表事实的形式，rule代表要验证的具体的协议，lemma基于rule中定义的动作事实，代表需要证明的安全属性。

在本方案的Tamarin的模型中，定义了2个基本角色。其中，UE代表方案中的UE，UAV代表方案中的UAV。在形式化验证的过程中，将方案建模为Setup、UE_1、UAV_1和UE_2这4个规则。其中，规则Setup表示系统建立阶段，为系统、UE和UAV生成系统参数。规则UE_1表示UE生成接入认证请求并发送的过程。规则UAV_1表示UAV接收接入认证请求，验证接入认证请求，计算会话密钥以及生成接入认证响应并发送的过程。规则UE_2表示UE收到接入认证响应，验证接入认证响应，并计算会话密钥的过程。形式化验证结果如图3所示，方案的形式化验证结果，在形式化验证的过程中通过4个引理来对方案的安全性进行验证。引理UAV_auth_UE验证UAV是否能够认证UE。引理UE_auth_UAV验证UE是否能够认证UAV。引理Executable Request、ExecutableConfirm确保引理不会因为模型的无法执行而空洞成立。所提出的方案能够成功实现UE到UAV的认证与UAV到UE的认证。

```
==============================================================================
summary of summaries:

analyzed: BAKA.spthy

  UAV_auth_UE (all-traces): verified (14 steps)
  UE_auth_UAV (all-traces): verified (49 steps)
  ExecutableRequest (exists-trace): verified (12 steps)
  ExecutableConfirm (exists-trace): verified (17 steps)

==============================================================================
```

图3　形式化验证结果

5　性能分析

本章对本文方案和其他相关文献[21-24]的计算成本、通信成本和检错复杂度进行分析比较。

对于基于双线性对的方案，使用双线性对$\bar{e}$:$G_1\times G_1\to G_2$来实现80位的安全强度，其中G_1为由点P生成的加法群。点P定义在q阶超奇异椭圆曲线$E:y^2=x^3+x \bmod \bar{p}$上，嵌入度为2，其中$\bar{p}$是1个512位的素数，$q$是1个160位的Solinas素数，并且满足$\bar{p}+1=12q^r$。

对于基于椭圆曲线密码的方案，使用定义在非奇异椭圆曲线$E:y^2=x^3+ax+b \bmod p$上的，由

点P生成的q阶加法群G，实现80位的安全强度，其中p、q为2个160位的素数，$a,b\in Z_p^*$。

5.1 计算开销

本节分析了所提出的方案的计算效率，并将其与其他的批量认证文献[21-24]进行了对比。为了方便对比，令T_{BP}为执行双线性对运算所耗时间，T_H为执行单向哈希函数运算所耗时间，T_{MTP}为执行Map-to-point哈希函数运算所耗时间，T_{BPA}为执行1个椭圆曲线加法运算所耗时间，T_{BPM}为执行1个椭圆曲线乘法运算所耗时间，T_s为执行1个标量乘法运算所耗时间，T_e为执行1个指数运算所耗时间。本文采用方案[7, 25]中的实验数据，即$T_H=0.000\ 1$ ms，$T_{MTP}=4.109\ 1$ ms，$T_{BP}=1.644\ 7$ ms，$T_{BPA}=0.009\ 9$ ms，$T_{BPM}=1.797\ 5$ ms，$T_s=0.442$ ms，$T_e=0.6$ ms，实验采用MIRACL密码库，运行环境为Intel Core i5 2.90 GHz，16 GB内存，Windows 7操作系统。

在文献[21]中，车辆需要进行1次椭圆曲线乘法运算、3次单向哈希函数运算、1次椭圆曲线加法运算和2次标量乘法运算。因此，生成单个签名的计算开销为$T_{BPM}+3T_H+T_{BPA}+2T_s=2.691\ 7$ ms。验证者验证一个签名需要进行5次椭圆曲线乘法运算、3次单向哈希函数运算和1次椭圆曲线加法运算。因此，验证单个签名的计算开销为$5T_{BPM}+3T_H+T_{BPA}=5\times1.797\ 5+3\times0.000\ 1+0.009\ 9=8.997\ 7$ ms。验证者验证n个签名需要进行$2n+1$次椭圆曲线乘法运算、$2n$次椭圆曲线加法运算、n次标量乘法运算和$2n$次单向哈希运算。因此，验证n个签名的计算开销为$(2n+1)T_{BPM}+2nT_{BPA}+nT_s+2nT_H=(4.057n+1.797\ 5)$ ms。

在文献[22]中，车辆需要进行3次单向哈希运算、3次标量乘法运算、2次椭圆曲线乘法运算和1次椭圆曲线加法运算。因此，计算单个签名的计算开销为$2T_{BPM}+T_{BPA}+3T_H+3T_s=2\times1.797\ 5+0.009\ 9+0.000\ 3+3\times0.442=4.931\ 2$ ms。验证者验证单个签名需要进行3次双线性对运算、2次椭圆曲线乘法运算和1次单向哈希函数运算。因此，验证单个签名的计算开销为$3T_{BP}+2T_{BPM}+T_H=3\times1.644\ 7+2\times1.797\ 5+0.000\ 1=8.529\ 2$ ms。验证者验证n个签名需要进行$3n$次双线性对运算、$2n$次椭圆曲线加法运算和1次椭圆曲线乘法运算。因此，验证n个签名的计算开销为$3nT_{BP}+2nT_{BPA}+T_{BPM}=(8.529\ 1n+1.797\ 5)$ ms。

在文献[23]中，车辆计算单个签名需要进行5次单向哈希运算、4次椭圆曲线加法运算和2次椭圆曲线乘法运算。因此，计算单个签名的计算开销为$2T_{BPM}+4T_{BPA}+5T_H=2\times1.797\ 5+4\times0.009\ 9+0.000\ 5=3.635\ 1$ ms。验证者验证单个签名需要进行4次单向哈希函数、1次椭圆曲线加法运算和7次椭圆曲线乘法运算。因此，验证单个签名的计算开销为$4T_H+T_{BPA}+7T_{BPM}=0.000\ 4+0.009\ 9+7\times1.797\ 5=12.592\ 8$ ms。验证者验证n个签名需要进行n次单向哈希运算、$2n+5$次椭圆曲线乘法运算、$6n+1$次椭圆曲线加法运算。因此，验证n个签名的计算开销为$(2n+5)T_{BPM}+(7n+1)T_{BPA}+nT_H=(3.664\ 4n+8.997\ 4)$ ms。

在文献[24]中用户计算单个签名需要进行3次椭圆曲线乘法运算、2次标量乘法运算和2次单向哈希运算。因此，计算单个签名的计算开销为$3T_{BPM}+2T_s+2T_H=3\times1.7975+2\times0.442+0.000\ 2=6.276\ 7$ ms。验证单个签名需要进行2次单向哈希运算、4次椭圆曲线乘法运算和2次椭圆曲线加法运算。因此，验证单个签名的计算开销为$4T_{BPM}+2T_{BPA}+2T_H=4\times1.797\ 5+2\times0.009\ 9+0.000\ 2=7.21$ ms。验证者验证n个签名需要进行$2n+2$次椭圆曲线乘法运算、$2n+2$次椭圆曲线加法运算和$2n$次标量乘法运算。因此，验证n个签名的计算开销为$(2n+2)T_{BPM}+(2n+2)T_{BPA}+2nT_s=(4.498\ 8n+3.614\ 8)$ ms。

在本文方案中，UE计算单个签名需要进行2次标量乘法运算。因此，生成单个签名所需要

的计算开销为 $2T_s=2\times0.442=0.884$ ms。验证者验证一个签名需要进行 3 次椭圆曲线加法运算和 3 次椭圆曲线乘法运算。因此，验证单个签名所需要的计算开销为 $3T_{\mathrm{BPM}}+3T_{\mathrm{BPA}}=3\times1.7975+3\times0.009\,9=5.422\,2$ ms。验证者验证 n 个签名需要进行 $2n+1$ 次椭圆曲线乘法运算和 $4n-3$ 次椭圆曲线加法运算。因此，验证 n 个签名所需要的计算开销为 $(2n+1)T_{\mathrm{BPM}}+(4n-3)T_{\mathrm{BPA}}=(3.634\,6n+1.767\,8)$ ms。

方案[21-24]与本文所提出的方案的计算开销进行对比，计算开销对比见表 1。与最优对比方案相比，本文所提出的方案在生成签名时将计算开销降低了 67%，在验证单个签名时将计算开销降低了 25%，生成单个签名与验证单个签名的计算开销对比，计算开销对比如图 4 所示。本文所提出的方案的批量认证的计算开销降低了 16%。批量认证时间与被验证的签名的数量的关系，批量认证时间对比如图 5 所示。因此，本文所提出的方案在计算开销方面有更优的性能。

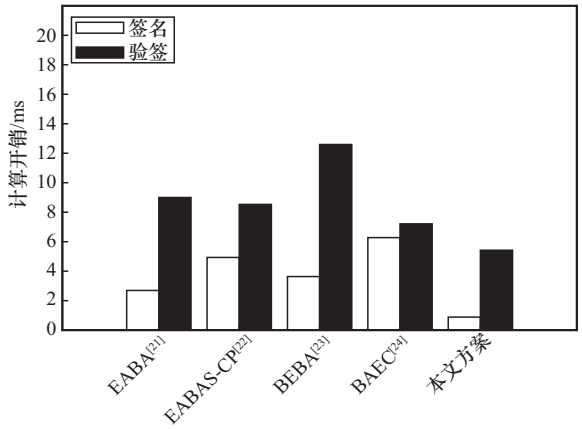


图 4　计算开销对比

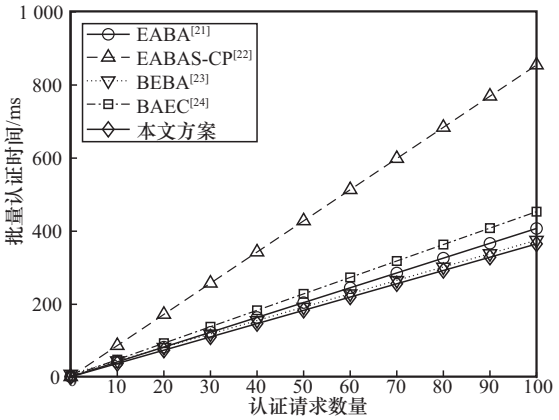


图 5　批量认证时间对比

5.2　通信开销

在本节将会将本文提出的方案与文献[21-24]的通信开销进行对比。由于，$\bar{p}$ 和 p 的大小分别是 64 byte 和 20 byte，因此 G_1 和 G 内的元素尺寸分别是 128 byte 和 40 byte。此外，根据 He 等[26]可知，普通的时间戳的大小为 4 byte，单向哈希函数的输出为 20 byte，Z_p^* 的大小为 20 byte。

在文献[21]中，车辆发送的消息为 $\left\{\mathrm{PID}_i=\left(\mathrm{PID}_{i,1},\mathrm{PID}_{i,2}\right),R_i,m_i,T_i,\sigma_i\right\}$，其中，$\left\{\mathrm{PID}_{i,1},\mathrm{PID}_{i,2},R_i\right\}\in G_1$，$\left\{m_i,\sigma_i\right\}\in Z_p^*$，$T_i$ 为时间戳。方案总的通信开销为 $3|G_1|+2\left|Z_p^*\right|+4=428$ byte。在文献[22]中，车辆发送的消息为 $\left\{\mathrm{PID}_{V_j}=\left(K_1,K_1'\right),\sigma,m_j,\mathrm{TS}_{V_j},R_{v1},R_{\mathrm{b}},\delta\right\}$，其中，$\left\{K_1,K_1',\sigma,R_{v1},R_{\mathrm{b}}\right\}\in G_1$，$\left\{m_j,\delta\right\}\in Z_p^*$，$\mathrm{TS}_{V_j}$ 为时间戳。方案总的通信开销为 $5|G_1|+2\left|Z_p^*\right|+4=684$ byte。在文献[23]中，车辆发送的消息为 $\left\{\sigma_k=\left(S_{k_1},S_{k_2}\right),m_k,L_k,N_k,T_k,W_k,Z_k\right\}$，其中

表 1　计算开销对比

方案	签名	单个签名认证	n 个签名认证
Yan 等[21]	$T_{\mathrm{BPM}}+3T_H+T_{\mathrm{BPA}}+2T_{\mathrm{s}}$	$5T_{\mathrm{BPM}}+3T_H+T_{\mathrm{BPA}}$	$(2n+1)T_{\mathrm{BPM}}+2nT_{\mathrm{BPA}}+nT_{\mathrm{s}}+2nT_H$
Maurya 等[22]	$2T_{\mathrm{BPM}}+T_{\mathrm{BPA}}+3T_H+3T_{\mathrm{s}}$	$3T_{\mathrm{BP}}+2T_{\mathrm{BPM}}+T_H$	$3nT_{\mathrm{BP}}+2nT_{\mathrm{BPA}}+T_{\mathrm{BPM}}$
Dwivedi 等[23]	$2T_{\mathrm{BPM}}+4T_{\mathrm{BPA}}+5T_H$	$4T_H+T_{\mathrm{BPA}}+7T_{\mathrm{BPM}}$	$(2n+5)T_{\mathrm{BPM}}+(7n+1)T_{\mathrm{BPA}}+nT_{\mathrm{H}}$
Cui 等[24]	$3T_{\mathrm{BPM}}+2T_{\mathrm{s}}+2T_H$	$4T_{\mathrm{BPM}}+2T_{\mathrm{BPA}}+2T_H$	$(2n+2)T_{\mathrm{BPM}}+(2n+2)T_{\mathrm{BPA}}+2nT_{\mathrm{s}}$
本文方案	$2T_s$	$3T_{\mathrm{BPM}}+3T_{\mathrm{BPA}}$	$(2n+1)T_{\mathrm{BPM}}+(4n-3)T_{\mathrm{BPA}}$

$\{S_{k_1}, S_{k_2}, L_k, N_k, T_k, W_k, Z_k\} \in G_1$，$m_k \in Z_p^*$。方案总的通信开销为$7|G_1|+|Z_p^*|$=916 byte。在文献[24]中，用户发送的消息为$\{\sigma_{i,j}=(R_{i,j}, U_{i,j}, \delta_{i,j}), M_i, T_i, \mathrm{PID}_{i,j}\}$，其中，$\{R_{i,j}, U_{i,j}, \mathrm{PID}_{i,j}\} \in G_1$，$\{\delta_{i,j}, M_i\} \in Z_p^*$，$T_i$为时间戳。方案总的通信开销为$3|G_1|+2|Z_p^*|+4$=428 byte。在本文方案中，UAV发送的消息为$\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, T_i, \sigma_i, \mathrm{Tpk}_{\mathrm{UAV}_i}\}$，其中，$\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, \mathrm{Tpk}_{\mathrm{UAV}_i}\} \in G_1$，$\sigma_i \in Z_p^*$。因此，本文方案的通信开销为$3|G_1|+|Z_p^*|+4$=408 byte。

所有方案的通信开销对比见表2，与最优对比方案相比，本文所提出的方案的通信开销降低了4%，对比得出本文所提出的方案在通信开销方面具有更优的性能。

表2　所有方案的通信开销对比

方案	消息	通信成本/byte
Yan等[21]	$\{\mathrm{PID}_i=(\mathrm{PID}_{i,1}, \mathrm{PID}_{i,2}), R_i, m_i, T_i, \sigma_i\}$	428
Maurya等[22]	$\{\mathrm{PID}_{V_j}=(K_1, K_1'), \sigma, m_j, \mathrm{TS}_{V_j}, R_{v1}, R_b, \delta\}$	684
Dwivedi等[23]	$\{\sigma_k=(S_{k_1}, S_{k_2}), m_k, L_k, N_k, T_k, W_k, Z_k\}$	916
Cui等[24]	$\{\sigma_{i,j}=(R_{i,j}, U_{i,j}, \delta_{i,j}), M_i, T_i, \mathrm{PID}_{i,j}\}$	428
本文方案	$\{\mathrm{PID}_i, \mathrm{vpk}_{\mathrm{PID}_i}, T_i, \sigma_i, \mathrm{Tpk}_{\mathrm{UAV}_i}\}$	408

5.3　检错次数对比

在本节将会对比本文提出的方案、基于d–不相交矩阵的文献[12]和传统的基于二分法的文献[24]，在接收到n个接入认证请求，最多存在d个无效接入认证请求的情况下，在批量认证出错时，检测错误的接入认证请求所需要的检测次数。

在传统的基于二分法的文献[24]中检错复杂度为$O(d\,\mathrm{lb}(n))$。文献[12]提出的检错方法基于d–不相交矩阵检错复杂度为$O(d^2\,\mathrm{lb}(n/d))$。本文方案基于(d,d)–可分解矩阵检错复杂度为$O(d\,\mathrm{lb}(n/d))$。虽然传统的基于二分法的方案所需要的检错次数较少，但是基于二分法的检错方法，无法并行运行，只能依次按照二分法进行聚合，逐个验证聚合签名，导致检错时间过长。检错复杂度对比如图6所示，分别比较了在d=2、d=5和d=10的情况下，3种方案的检错复杂度。其中，d设置为2考虑的是方案应用于安全的通信环境的情况，此时非法用户较少，因此将最大的无效接入认证请求数量设置为2。d设置为5考虑的是方案应用于一般的通信环境的情况，此时存在一定数量的非法用户，因此将最大的无效接入认证请求数量设置为5。d设置为10考虑的是

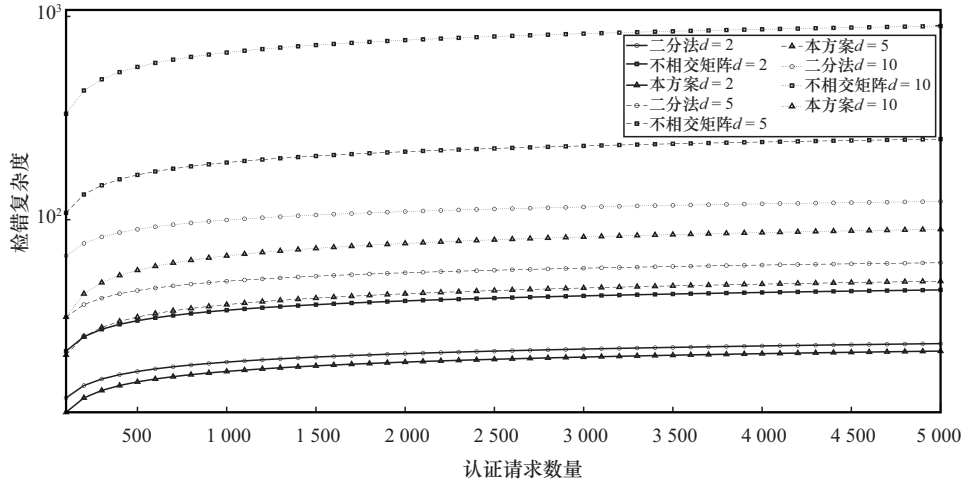


图6　检错复杂度对比

方案应用于危险的通信环境的情况，此时存在较多的非法用户，因此将最大的无效接入认证请求数量设置为10。通过将 d 分别设置为2、5和10考虑了本文方案应用于不同通信环境的情况。由图6可以看出，本文方案在3种情况下检错复杂度均最低。经过对比得出，本方案检错复杂度最低，认证效率最高，具有最优的性能。

6　结束语

本文针对UAV网络中多用户高并发接入场景下，传统批量认证协议存在的容错性能差、检错能力不足等问题，提出了一种结合CLAS与群组测试的高效批量认证与密钥协商协议。该方案一方面构建了支持并发接入与快速签名验证的CLAS批量认证协议，降低了系统的计算与通信开销；另一方面，引入群组测试构建高效检错机制，可在认证失败场景中快速识别无效请求，避免有效请求被误拒，提升整体认证成功率与系统鲁棒性。在安全性方面，本文对所提出的方案进行了非形式化安全分析，在随机预言机模型下进行了安全性证明，在Tamarin下进行了形式化验证，证明了本文方案的安全性。在性能评估中，与现有主流方案对比结果显示，本文方案在计算开销、通信成本及检错复杂度方面均表现出更优的性能。本文所提出的方案在保障认证安全性的同时兼顾了系统开销与容错能力，为资源受限的UAV网络中UE并发接入认证提供了一种高效、安全的解决方案。

参考文献：

[1] NAWAZ H, ALI H M, ALI LAGHARI A. UAV communication networks issues: a review[J]. Archives of Computational Methods in Engineering, 2021, 28(3): 1349-1369.

[2] ZHI Y Y, FU Z J, SUN X M, et al. Security and privacy issues of UAV: a survey[J]. Mobile Networks and Applications, 2020, 25(1): 95-101.

[3] XIONG H, GUAN Z, CHEN Z, et al. An efficient certificateless aggregate signature with constant pairing computations[J]. Information Sciences, 2013, 219: 225-235.

[4] SOBEL M, GROLL P A. Group testing to eliminate efficiently all defectives in a binomial sample[J]. Bell System Technical Journal, 1959, 38(5): 1179-1252.

[5] LI T, MA J F, FENG P B, et al. Lightweight security authentication mechanism towards UAV networks[C]//Proceedings of the 2019 International Conference on Networking and Network Applications (NaNA). Piscataway: IEEE Press, 2019: 379-384.

[6] YOON K, PARK D, YIM Y, et al. Security authentication system using encrypted channel on UAV network[C]//Proceedings of the 2017 First IEEE International Conference on Robotic Computing (IRC). Piscataway: IEEE Press, 2017: 393-398.

[7] WANG Z Z, ZHANG J W, LIU Y, et al. A certificateless authentication scheme with fuzzy batch verification for federated UAV network[J]. International Journal of Intelligent Systems, 2022, 37(9): 6048-6079.

[8] AZEES M, VIJAYAKUMAR P, DEBOARH L J. EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(9): 2467-2476.

[9] RAYA M, HUBAUX J P. Securing vehicular ad hoc networks[J]. Journal of Computer Security, 2007, 15(1): 39-68.

[10] IQBAL A, ZUBAIR M, KHAN M A, et al. An efficient and secure certificateless aggregate signature scheme for vehicular ad hoc networks[J]. Future Internet, 2023, 15(8): 266.

[11] SAMRA B, FOUZI S. New efficient certificateless scheme-based conditional privacy preservation authentication for applications in VANET[J]. Vehicular Communications, 2022, 34: 100414.

[12] HARTUNG G, KAIDEL B, KOCH A, et al. Fault-tolerant aggregate signatures[C]//Proceedings of the 19th International Conference on Practice and Theory in Public-Key Cryptography (PKC 2016). Piscataway: Springer Press, 2016: 331-356.

[13] JADHAV P, MISBAHUDDIN M, CHIPPALKATTI S S, et al. PKI-enabled authentication and encryption for enhanced drone communication[C]//Proceedings of the 2024 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA). Piscataway: IEEE Press, 2024: 1-10.

[14] ALLADI T, NAREN, BANSAL G, et al. SecAuthUAV: a novel authentication scheme for UAV-ground station and UAV-UAV communication[J]. IEEE Transactions on Vehicular Technology, 2020, 69(12): 15068-15077.

[15] JAN S U, KHAN H U. Identity and aggregate signature-based

authentication protocol for IoD deployment military drone[J]. IEEE Access, 2021, 9: 130247-130263.

[16] WANI A R, GUPTA S K, KHANAM Z, et al. A novel approach for securing data against adversary attacks in UAV embedded HetNet using identity based authentication scheme[J]. IET Intelligent Transport Systems, 2023, 17(11): 2171-2189.

[17] SEMAL B, MARKANTONAKIS K, AKRAM R N. A certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks[C]//Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). Piscataway: IEEE Press, 2018: 1-8.

[18] LI J Y, WANG Y J, DING Y, et al. A certificateless pairing-free authentication scheme for unmanned aerial vehicle networks[J]. Security and Communication Networks, 2021, 2021(1): 9463606.

[19] ALI I, CHEN Y, ULLAH N, et al. An efficient and provably secure ECC-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs[J]. IEEE Transactions on Vehicular Technology, 2021, 70(2): 1278-1291.

[20] BARDINI IDALINO T, MOURA L. Nested cover-free families for unbounded fault-tolerant aggregate signatures[J]. Theoretical Computer Science, 2021, 854: 116-130.

[21] YAN C Z, WANG C, SHEN J, et al. Edge-assisted hierarchical batch authentication scheme for VANETs[J]. IEEE Transactions on Vehicular Technology, 2024, 73(1): 1253-1262.

[22] MAURYA C, CHAURASIYA V K. Efficient anonymous batch authentication scheme with conditional privacy in the Internet of vehicles (IoV) applications[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(9): 9670-9683.

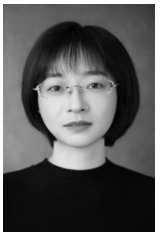
[23] DWIVEDI S K, AMIN R, VOLLALA S, et al. Design of blockchain and ECC-based robust and efficient batch authentication protocol for vehicular ad-hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(1): 275-288.

[24] CUI J, WANG F Q, ZHANG Q Y, et al. Efficient batch authentication scheme based on edge computing in IIoT[J]. IEEE Transactions on Network and Service Management, 2023, 20(1): 357-368.

[25] 熊婉君, 王若梅, 王玉珏, 等. 车联网中基于无证书聚合签名的条件隐私保护批量认证方案[J]. 密码学报, 2023, 10(3): 462-475.
XIONG W J, WANG R M, WANG Y J, et al. A conditional privacy-preserving batch authentication scheme based on certificateless aggregate signature for VANETs[J]. Journal of Cryptologic Research, 2023, 10(3): 462-475.

[26] HE D B, ZEADALLY S, XU B W, et al. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2681-2691.

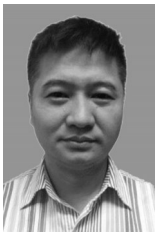
[作者简介]



郭超（1987– ），女，博士，北京电子科技学院电子与通信工程系副教授，主要研究方向为网络信息、数据安全。

黄子琛（2002– ），男，北京电子科技学院网络空间安全系硕士生，主要研究方向为安全协议、网络安全。

弓丞（1986– ），男，博士，联通（北京）产业互联网有限公司工程师，主要研究方向为数据安全。



刘培鹤（1972– ），男，现就职于北京电子科技学院电子与通信工程系，主要研究方向为网络安全、物联网、无线通信。