

基于区块链的零知识位置证明系统设计

吴炜, 刘儿兀, 杨昌鑫, 王睿

(同济大学电子与信息工程学院, 上海 201804)

摘要: 智能终端设备及定位技术的发展催生了许多基于位置的服务, 如定点打卡服务(访问特定地点获取相应的奖励), 这导致非法用户为了获取利益对位置服务器进行位置欺骗。因此, 需要对用户提供的位置证书进行验证。但现有的位置证明系统在验证用户位置证书的同时对用户隐私保护并不周全, 且用户对位置证书的控制并不灵活。基于区块链技术, 提出一种分布式位置证明系统架构; 进而在此架构基础上, 引入零知识证明, 提出一种零知识位置证明协议。架构与协议组成的零知识位置证明系统允许用户根据需求, 自由选择披露的证书内容及位置精度, 实现分级的位置隐私保护。

关键词: 区块链; 零知识证明; 隐私保护; 位置证明系统

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.20200020

Design of blockchain based zero-knowledge proof of location system

WU Wei, LIU Erwu, YANG Changxin, WANG Rui

School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

Abstract: With the development of precise positioning technology, a growing number of location-based services (LBS) emerge. For example, visit a specific place to get the corresponding reward. This also leads to location fraud by illegal users in order to gain benefits. Thus, it is necessary to verify location certificates provided by users. However, present proof of location systems are deficient in verifying users' location certificates while protecting users' privacy. Users do not flexibly control their location certificates either. Based on blockchain, a kind of distributed proof of location system architecture was proposed and based on the proposed system architecture, a kind of proof of location protocol with zero-knowledge proof was proposed further. With the proposed architecture and the protocol, the proposed proof of location system allows users to freely select disclosed certificate parameters and positional accuracy according to their needs so that hierarchical location privacy protection was achieved.

Key words: blockchain, zero-knowledge proof, privacy protection, proof of location system

收稿日期: 2020-01-03; 修回日期: 2020-03-17

通信作者: 刘儿兀, erwuliu@tongji.edu.cn

基金项目: 上海市军民融合发展专项基金资助项目(No.JMRH-2018-1075); 国家自然科学基金资助项目(No.61571330); 上海市科学技术委员会科研计划基金资助项目(No.19511102002)

Foundation Items: Shanghai Integrated Military and Civilian Development Fund (No.JMRH-2018-1075), The National Natural Science Foundation of China (No.61571330), Science and Technology Commission of Shanghai Municipality Research Project (No.19511102002)

论文引用格式: 吴炜, 刘儿兀, 杨昌鑫, 等. 基于区块链的零知识位置证明系统设计[J]. 网络与信息安全学报, 2020, 6(2): 67-76.

WU W, LIU E W, YANG C X, et al. Design of blockchain based zero-knowledge proof of location system[J]. Chinese Journal of Network and Information Security, 2020, 6(2): 67-76.

1 引言

随着智能终端设备和高精度定位技术的发展,越来越多的位置服务(LBS, location-based service)^[1]使人们生活更加便利。例如,基于位置的定点签到服务,用户需到达指定地点签到获取相应的奖励;基于位置的个性推荐服务,用户需向位置服务提供商(LBSP, location-based service provider)提供位置信息,LBSP则根据该信息,为用户提供兴趣点推荐^[2];基于位置的社交网络:用户可通过智能手机、平板、智能穿戴设备等对当前所在的兴趣点签到,并将签到信息和用户体验与好友分享。为了防止恶意用户滥用LBSP维护的服务资源,LBSP要求请求位置服务的用户提供相关的位置证明(PL, proof of location),证明其位置证书(LC, location certificate)的合法性。但是,用户的某些不必要披露的隐私信息会随PL暴露给LBSP,因此,在实现用户LC可验证的同时保护位置隐私是必要的。

国内外有许多学者对位置证明系统进行研究,并取得了丰富的成果。现有位置证明系统大致可分为两种:中心式系统与分布式系统。在中心式位置证明系统中,用户LC存储在单一的中央服务器或服务器群上,而在分布式位置证明系统中,用户LC存储在分布式网络上。Tyagi等^[3]基于类混合空间模型,结合其他多种方法,提出一种位置证明系统,保护用户隐私。Javali等^[4]基于信道状态信息(CSI, channel state information)以及密码基元fuzzy vault,提出一种包含证书生成与证书签发两部分的位置证明系统,该系统允许用户以高概率证明其LC的有效性。Li等^[5]基于Wi-Fi或蜂窝网接入点(AP, access point),提出一种基于基础设施的位置证明系统,该系统允许数据库在不知道用户精确位置的情况下验证LC的有效性。上述位置证明系统均为中心式,存在以下几点缺陷:第一,中心式位置证明系统需要专用的大功率中央服务器用于服务大量用户,造成负载不均衡,一旦中央服务器过载,整个系统就会瘫痪;第二,中央服务器成为安全的焦点,恶意攻击者更容易攻陷系统;第三,用户LC完全由中央服务器管理,用户失去对其数据控制权,

用户隐私安全建立在中央服务器诚实的基础上。为了解决这些问题,一系列分布式位置证明系统被提出,如Zhu等^[6]提出的APPLAUS、Wang等^[7]提出的STAMP以及Nosouhi等^[8]提出的SPARSE。另外,Amoretti等^[9]提出了基于区块链的位置证明系统,在这些系统中,用户以短程通信方式请求邻近的证人节点为其签发LC。

尽管这些解决方案克服了中心式系统的弊端且不需要Wi-Fi、AP等基础设施,但它们或存在证人节点作恶不可控造成系统的合谋攻击问题;或存在假名而非匿名造成的用户隐私推导攻击问题。因此,本文结合区块链技术,提出一种分布式的位置证明系统架构,该架构由3种节点组成:用户节点、LBSP节点以及证书签发节点(CIN, certificate issuing node),证书签发节点可以是Wi-Fi、AP、微基站等。特别注意的是,证书签发节点依据地理位置进行二进制编码,其目的是实现不同精度的位置证明。然后,在该系统架构的基础上,提出一种零知识位置证明协议。完整位置证明系统允许用户自主选择披露的LC参数,提供任意精度的位置证明,实现用户可控的分级隐私保护,适应不同的应用场景。此外,本文针对可能的恶意攻击,对系统的安全性进行理论分析,得知系统可抵御这些恶意攻击。最后,对不同位置精度等级的位置证明系统进行实验研究,得出位置精度等级与系统性能关系:位置证明系统执行时延随位置精度等级增加而增大。

2 系统架构

区块链技术是多种技术的融合,包括加密、共识机制、激励机制、分布式存储和智能合约^[10-13]。本质上,区块链是用密码方法保证其不可破坏的、不可篡改的链式数据结构(或公共账本),它由一系列区块组成,后一个区块通过包含前一个区块的加密散列值与前一个区块相连,该区块还包含时间戳、随机数、交易数据等。然后,通过共识算法,如工作量证明(PoW)和权益证明(PoS),网络中的所有对等点对区块数据写入权(公共账本记账权)达成共识。获得记账权的对等节点随后将打包的块添加到链中,链上记录的数据无法修改。这整个过程被称作挖矿,参加记账权竞争

的对等节点被称为“矿工”。对公有链来说，最终获得记账权并完成记账的“矿工”将获得代币奖励，这种机制称为激励机制。当前，至少有4种类型的区块链网络：公有链、私有链、联盟链和混合链。本文介绍的区块链网络是混合链，混合链网络中不同的对等节点根据其不同的职能有不同的权限。

用户的位置数据描绘了用户的数字轨迹。除此之外，攻击者可通过数字挖掘，探寻该数字轨迹更深层次的上下文信息，如个人习惯、兴趣爱好、社交活动等。这些敏感信息的暴露使用户成为追踪广告、垃圾邮箱、恶意跟踪等非法活动的受害者，造成用户的精神及财产损失，甚至使用户受到人身伤害^[14]。传统位置证明系统架构中，用户的LC不由用户控制，或存储在中央服务器上，或存储在分布式数据库上，为了解决这些缺陷，本文提出基于区块链的位置证明系统架构，并结合零知识位置证明（ZK-PL，zero-knowledge proof of location）协议，使用户完全掌握自身的LC。

ZK-PL系统架构由3个组件组成，分别是公共账本、LBSP以及用户，如图1所示。公共账

本是由CIN组成区块链网络，每个CIN为一个点，两个地理位置相邻的CIN组成一个块，两个地理位置相邻的块组成一个区，两个地理位置相邻的区组成一个域，两个地理位置相邻的域组成一个宏，以此方式，公共账本覆盖的地理范围呈指数级方式增加，满足任何的应用场景。对每个CIN进行二进制编码，图中CIN编号范围为0000~1111，其中左上角CIN为0000号，右下角为1111号（按照左0右1、上0下1编码方式）。公共账本的主要职能有：第一，为用户签发LC；第二，记录LC指纹；第三，为用户提供生成PL所需要的公共参数（CRS，common reference string）；第四，保存服务记录。用户主要职能有：第一，以短程通信方式接入区块链网络，向附近CIN请求签发LC；第二，利用LC生成PL，以远程通信方式向LBSP请求相关位置服务。LBSP的主要职能有：第一，验证PL，为用户提供位置服务；第二，以远程通信方式请求CIN保存服务记录。该系统架构实现用户位置数据与位置服务的分离，各组件职能明确，避免了服务器或其他第三方掌握敏感的用户隐私信息。公共账本作为可信之根本，在不暴露用户隐私的情况下杜绝了

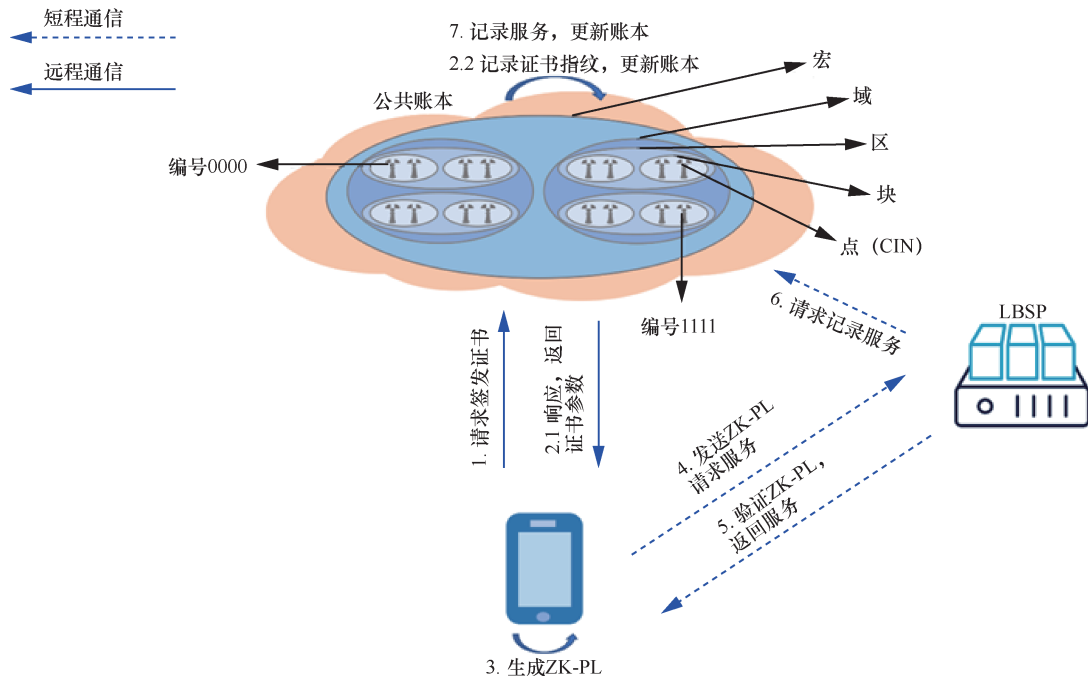


图1 零知识位置证明系统架构
Figure 1 Zero-knowledge location proof system architecture diagram

LC 的伪造、篡改及抵赖。用户则可通过生成零知识位置证明，向 LBSP 证明其持有证书的合法性，获取位置服务。相对地，LBSP 只能从 ZK-PL 中获取必要的用户位置信息，避免了用户隐私泄露。

明确了系统架构组成及各组件职能之后，需要明确各个组件之间的交互细节。为此，下文提出一种零知识位置证明协议（ZK-PLP，zero-knowledge proof of location protocol）。

3 ZK-PLP

ZK-PLP 由两部分组成，分别是 LC 签发协议和 LC 验证协议。其中，LC 签发协议主要用于规定用户如何与 CIN 交互，请求签发 LC；LC 验证协议主要用于规定用户如何产生 ZK-PL 以及 LBSP 如何验证 ZK-PL 的合法性。协议流程如图 2 所示。LC 签发阶段，用户首先以短程通信方式接入区块链网络，请求邻近 CIN 签发 LC。CIN 在收到请求后，验证请求的有效性，验证通过则返回 LC 参数；同时，CIN 结合用户请求中包含的信息，生成 LC 指纹并将指纹广播至整个区块链网络，等待网络中节点挖矿保存指纹。而用户在收到 CIN 返回的 LC 参数后，生成 LC 保存至本地，留待请求位置服务时使用。LC 验证阶段，用户使用本地保存的 LC 生成 ZK-PL。然后，以远程通信方式将 ZK-PL 发送给 LBSP，请求位置服务。LBSP 在收到 ZK-PL 后，验证 ZK-PL 的有效性，验证通过则返回位置服务；同时，LBSP 生成服务记录并请求 CIN 将服务记录保存至区块链网络。LBSP 生成服务记录并请求 CIN 将服务记录保存至区块链网络。

链网络，等待网络中节点挖矿保存指纹。而用户在收到 CIN 返回的 LC 参数后，生成 LC 保存至本地，留待请求位置服务时使用。LC 验证阶段，用户使用本地保存的 LC 生成 ZK-PL。然后，以远程通信方式将 ZK-PL 发送给 LBSP，请求位置服务。LBSP 在收到 ZK-PL 后，验证 ZK-PL 的有效性，验证通过则返回位置服务；同时，LBSP 生成服务记录并请求 CIN 将服务记录保存至区块链网络。

3.1 LC 签发协议

在阐述 LC 签发协议细节之前，首先说明 LC 格式及其各参数的含义，LC 格式如下。

$$LC: \left\{ \begin{array}{l} serialno \\ pk_{user} \\ time \\ number \end{array} \right\} \quad (1)$$

各个参数含义如表 1 所示。需要额外说明的是，每个用户拥有唯一的公私钥对，可由权威中心（政府）统一管理，类似于个人身份证。CIN 编号之所以能表征用户出现地点，是因为 CIN 的

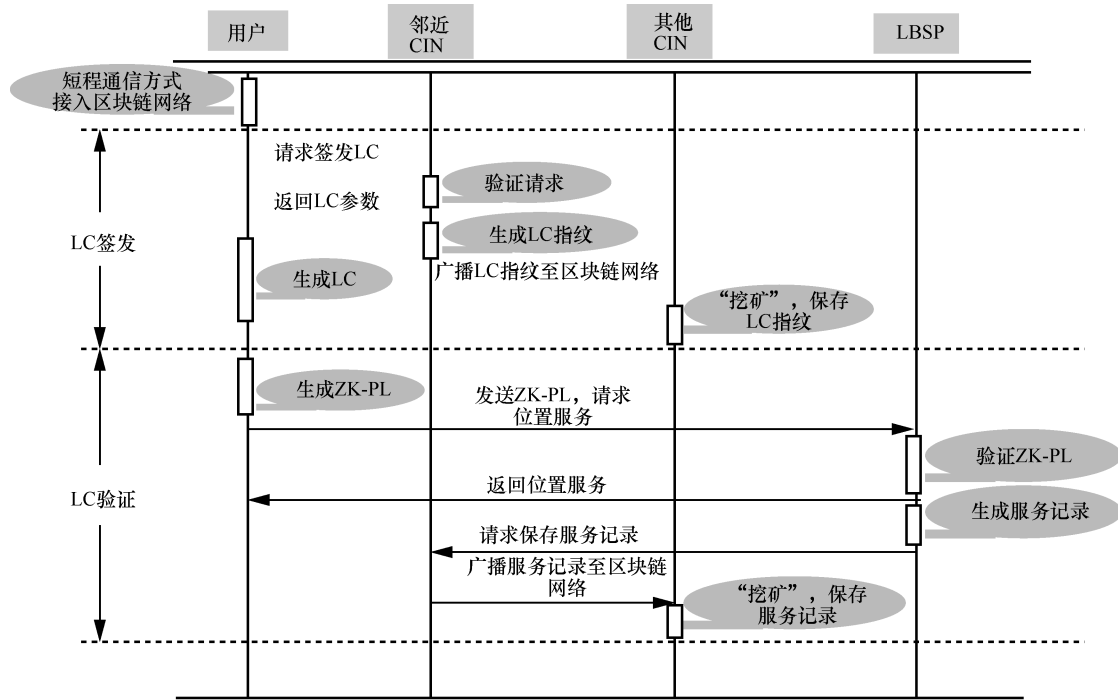


图 2 零知识位置证明协议流程
Figure 2 Zero-knowledge location proof protocol flowchart

地理位置是固定的且用户只有在短程通信范围内才能成功签发 LC，通信范围越小，其表征的用户位置越精确。

表 1 LC 参数
Table 1 Parameters of the LC

参数名	参数含义
<i>serialno</i>	证书序列号
<i>pk_{user}</i>	用户公钥，表征用户身份
<i>time</i>	时间戳，表征用户出现时间
<i>number</i>	CIN 编号，表征用户出现位置

下面说明 LC 签发流程细节：假设某用户需要签发一张 LC，首先，该用户通过短程通信的方式接入区块链网络，向邻近的 CIN 发起证书签发请求，请求格式如下。

$$req_{user} : \left\{ \begin{array}{l} inf_{user} : \{pk_{user}\} \\ sig_{user} : \{Hash(inf_{user})\}_{sk_{user}} \end{array} \right\}_{pk_{CIN}} \quad (2)$$

其中， inf_{user} 表示由用户发出的消息，大括号的下标 sk_{user} 和 pk_{CIN} 分别表示用户私钥和 CIN 公钥对大括号内消息进行加密。 sig_{user} 为用户对其发出消息的数字签名，目的是防止恶意中间人对消息进行篡改。CIN 公钥加密确保只有 CIN 才能解密请求，防止恶意中间人窃取请求中的消息。

CIN 收到用户请求之后，做如下检测：

- 1) 请求内容是否被篡改；
- 2) 用户与其通信方式是否为短程通信。

检测通过后，CIN 响应用户，响应格式如下。

$$res_{CIN} : \left\{ \begin{array}{l} inf_{CIN} : \left\{ \begin{array}{l} serialno \\ time \\ number \end{array} \right\} \\ sig_{CIN} : \{Hash(inf_{CIN})\}_{sk_{CIN}} \end{array} \right\}_{pk_{user}} \quad (3)$$

其中，证书序列号为 CIN 生成的随机数，一定程度上，证书序列号代表着证书所有权。CIN 以类似方式对响应及响应中的消息分别进行加密和数字签名，确保恶意中间人无法篡改与窃听。

成功接收到响应后用户告知 CIN，然后用户利用 inf_{user} 与 inf_{CIN} 生成 LC 保存至本地。与此同时，CIN 生成位置证书指纹（DLC，digest of location certificate），形式如下。其中，||为连接符。

$$DLC = Hash(serialno || pk_{user} || time || number) \quad (4)$$

并将其广播至区块链网络，等待挖矿保存至公共账本中。完成 DLC 广播后，CIN 删除 inf_{user} 与 inf_{CIN} 。CIN 的所有行为都可通过嵌入智能合约^[15]的方式使其规范化，防止其作恶。当用户需要 LBSP 提供位置服务时，可利用保存在本地的 LC 生成 ZK-PL，证明其拥有的证书的合法性，即该用户拥有的证书与链上证书指纹匹配。

3.2 LC 验证协议

现有的位置服务种类繁多，不同的位置服务中，LBSP 需要查验的用户位置信息也不尽相同。例如，在打卡获取商场优惠券服务中，LBSP 只需验证用户是否去过特定商场，而不需要知道用户身份（即公钥）、去该商场的的时间以及去到商场的哪个特定位置（假设商场为一个宏，LBSP 不需知道用户去哪个域、哪个区、哪个块甚至哪个点）。再比如，某用户需要向法院提供不在场证明，则他/她需要提供证明，披露其身份以及在特定时间出现在特定地点。基于 ZK-SNARK^[16]的 LC 验证协议允许用户根据位置服务类型，生成不同的 ZK-PL，只披露必要的位置信息，更好地保护用户隐私。

依据前面分析，不同位置服务可能要求用户披露不同的 LC 参数，也可能需要不同的位置精度。因此，按披露参数不同，将 ZK-PL 分为若干类型，如表 2 所示。

表 2 ZK-PL 参数
Table 2 Parameters of the ZK-PL

类型	公共参数（披露）	私密参数（不披露）
1		<i>serialno, pk_{user}, time, number</i>
2	<i>pk_{user}</i>	<i>time, number</i>
3	<i>time</i>	<i>pk_{user}, number</i>
4	<i>pk_{user}, time</i>	<i>number</i>

表 2 中公共参数与私密参数统称为输入参数，为生成 ZK-PL 的输入；公共参数直接披露给 LBSP。位置精度由 CIN 的短程通信范围决定，通信范围越小精度越高，相同覆盖面积所需要的 CIN 个数越多，最终导致 CIN 的编码长度越长。在打卡获取商场优惠券的例子中，商场由 8 个 CIN 覆盖，因此编码长度为 4 位（即精度等级为 4）。

下面说明 LC 验证流程细节：假设上述 LC 签发流程中用户欲使用签发完成的 LC 生成 ZK-PL，向 LBSP 证明他/她去过 0 域 1 区，从而获取相应的奖励。此场景应选择表 2 中类型 1 的 ZK-PL。用户需要证明以下算式成立。

$$\begin{cases} \text{Hash}(\text{serialno} \parallel pk_{\text{user}} \parallel \text{time} \parallel \text{number}) = \text{DLC} \\ \text{number} \wedge 1100 = 0100 \end{cases} \quad (5)$$

其中， $\parallel$ 为连接符； $\wedge$ 为按位与运算；1100 为位置掩码，用于屏蔽 LBSP 不必了解的位置信息，精度等级 4 中的位置掩码如表 3 所示。

表 3 位置掩码
Table 3 Position mask

位置掩码	位置掩码的作用
1000	屏蔽区、块、点信息
1100	屏蔽块、点信息
1110	屏蔽点信息
1111	不屏蔽任何位置信息

(1) 计算问题到 QAP 问题^[17]的转化

一般计算问题需转化为 QAP 问题才能通过 ZK-SNARK 协议，以零知识的方式证明其计算的正确性。QAP 问题可以简单地描述为：对于一系列多项式以及一个目标多项式，寻找这些多项式的线性组合，使目标多项式能够整除此多项式组合。计算问题(5)可转化为以下 QAP 问题。

$$\begin{cases} \text{variable polynomials} : \{l_i(x), r_i(x), o_i(x)\}_{i \in \{0, \dots, m, \dots, n\}} \\ \text{target polynomial} : \{t(x) = (x-1) \cdots (x-d)\} \end{cases} \quad (6)$$

其中， $\{l_i(x), r_i(x), o_i(x)\}$ 为变量多项式^[18]， $i=0$ 代表伪变量 *one*，其变量赋值固定为 1。 $i=1, \dots, m$ 代表公共变量（本例中 $m=0$ ），即表 2 中公共参数，其赋值是 LBSP 可见的。 $i=m+1, \dots, n$ 代表私密变量，即表 2 中私密参数以及一系列私密中间值。 $t(x)$ 为 d 阶目标多项式。至此，用户提供 LC 参数使之满足式 (5) 的问题转化为用户提供见证。

$$\text{witness} : \{v_i\}_{i \in \{1, \dots, m, \dots, n\}, v_i \in \mathbb{F}_q} \quad (7)$$

使之满足

$$\left(1 \cdot l_0(x) + \sum_{i=1}^n v_i l_i(x)\right) \cdot \left(1 \cdot r_0(x) + \sum_{i=1}^n v_i r_i(x)\right) -$$

$$\left(1 \cdot o_0(x) + \sum_{i=1}^n v_i o_i(x)\right) = L(x)R(x) - O(x) = t(x)h(x) \quad (8)$$

(2) 计算 CRS

计算问题到 QAP 的转化完成之后，需要选取随机值组 $\{s, g, \rho_l, \rho_r, \alpha_l, \alpha_r, \alpha_o, \beta, \gamma\}$ 。然后，利用随机值组以及上述 QAP，计算证明密钥 (proving key) 以及验证密钥 (verification key)。

$$\text{proving key} : \left\{ \begin{array}{l} \{g^{s^i}\}_{i \in \{1, \dots, d\}} \\ \{g_l^{l_i(s)}, g_r^{r_i(s)}, g_o^{o_i(s)}\}_{i \in \{m+1, \dots, n\}} \\ \{g_l^{\alpha_l l_i(s)}, g_r^{\alpha_r r_i(s)}, g_o^{\alpha_o o_i(s)}\}_{i \in \{m+1, \dots, n\}} \\ \{g_l^{\beta l_i(s)}, g_r^{\beta r_i(s)}, g_o^{\beta o_i(s)}\}_{i \in \{m+1, \dots, n\}} \end{array} \right\} \quad (9)$$

$$\text{verification key} : \left\{ \begin{array}{l} g^1, g_o^{t(s)} \\ \{g_l^{l_i(s)}, g_r^{r_i(s)}, g_o^{o_i(s)}\}_{i \in \{0, \dots, m\}} \\ g^{\alpha_l}, g^{\alpha_r}, g^{\alpha_o}, g^{\gamma}, g^{\beta \gamma} \end{array} \right\} \quad (10)$$

其中， $s, \rho_l, \rho_r, \alpha_l, \alpha_r, \alpha_o, \beta, \gamma \in \mathbb{F}_q$ ， g 为椭圆曲线 $\mathcal{G}$ 的生成元； $g_l = g^{\rho_l}$ ， $g_r = g^{\rho_r}$ ， $g_o = g^{\rho_l \rho_r}$ 。

用户利用证明密钥及本地计算的见证生成 ZK-PL 并发送给 LBSP。然后，LBSP 利用验证密钥验证 ZK-PL 的有效性。一旦 CRS 计算完毕，随机值组立刻删除。因为任何拥有该随机值组的恶意中间人可在不知道问题正确解的情况下伪造 ZK-PL，通过 LBSP 的验证，骗取位置服务。因此，随机值组也被称为“毒废料”。步骤 1)、步骤 2) 统称为可信初始化 (TS, trusted setup)，由系统中的 CIN 完成。正如前文所述，CIN 中嵌入的智能合约可规范其行为，保证用户隐私不泄露，保证 ZK-PL 不可伪造。

(3) 计算见证

初始化完成以后，为了生成 ZK-PL，用户需要根据自己的 LC 证书，本地生成 QAP 问题的解，类似于式(7)。

$$\text{witness} : \{v_i\}_{i \in \{m+1, \dots, n\}, v_i \in \mathbb{F}_q} \quad (11)$$

此处之所以不需计算 $v_1, \dots, v_m$ ，是因为 $v_1, \dots, v_m$ 是公共变量多项式的赋值，该赋值（即需要披露的位置信息）可直接披露给 LBSP。需

要注意的是，私密变量及中间变量的赋值不可直接披露给 LBSP，因为这些赋值包含了 LBSP 不需要知道的位置信息。

(4) 生成 ZK-PL

用户使用变量多项式计算 $L(x)$ 、 $R(x)$ 和 $O(x)$ 。

$$\begin{cases} L(x) = 1 \cdot l_0(x) + \sum_{i=1}^n v_i l_i(x) \\ R(x) = 1 \cdot r_0(x) + \sum_{i=1}^n v_i r_i(x) \\ O(x) = 1 \cdot o_0(x) + \sum_{i=1}^n v_i o_i(x) \end{cases} \quad (12)$$

用户计算

$$h(x) = \frac{L(x)R(x) - O(x)}{t(x)} \quad (13)$$

然后用户根据证明密钥计算

$$g^{h(s)} = g^{\sum_{i=1}^k s^i} \quad (14)$$

其中， $k < d$ 。用户根据证明密钥计算证明变量多项式及其 α 偏移。

$$\begin{cases} g_l^{L_p(s)} = g_l^{\sum_{i=m+1}^n v_i l_i(s)}, g_l^{\alpha_l L_p(s)} = g_l^{\alpha_l \sum_{i=m+1}^n v_i l_i(s)} \\ g_r^{R_p(s)} = g_r^{\sum_{i=m+1}^n v_i r_i(s)}, g_r^{\alpha_r R_p(s)} = g_r^{\alpha_r \sum_{i=m+1}^n v_i r_i(s)} \\ g_o^{O_p(s)} = g_o^{\sum_{i=m+1}^n v_i o_i(s)}, g_r^{\alpha_o O_p(s)} = g_r^{\alpha_o \sum_{i=m+1}^n v_i o_i(s)} \end{cases} \quad (15)$$

然后用户计算变量一致性多项式。

$$g^{Z(s)} = \prod_{i=m+1}^n \left(g_l^{\beta l_i(s)} g_r^{\beta r_i(s)} g_o^{\beta o_i(s)} \right)^{v_i} \quad (16)$$

最后，用户将计算的各部分组成 ZK-PL。

$$ZKPL : \left\{ \begin{array}{l} g_l^{L_p(s)}, g_l^{\alpha_l L_p(s)} \\ g_r^{R_p(s)}, g_r^{\alpha_r R_p(s)} \\ g_o^{O_p(s)}, g_o^{\alpha_o O_p(s)} \\ g^{Z(s)}, g^{h(s)} \end{array} \right\} \quad (17)$$

用户将生成的 ZK-PL 发送给 LBSP，请求位置服务，请求格式如下。

$$req_{user} : \{ZKPL, index\} \quad (18)$$

其中， $index$ 为服务编号，指明用户所请求的服务

类型。需要说明的是，用户向 LBSP 请求服务时采用远程通信方式，其加密方式与传统位置服务系统一致，本文不作讨论。此外，用户远程通信中使用的 ID 不再是其唯一公钥。因此，用户真实身份不会暴露。

(5) 验证 ZK-PL

LBSP 收到用户请求后，首先检测该 LC 是否失效。LC 失效是因为某些位置服务存在响应次数限制。例如，打卡获取商场优惠券服务中，每张 LC 只能兑换一张或者数张优惠券。LC 有效性检测通过查询公共账本服务记录的方式实现，每次 LBSP 完成一次服务之后，会请求 CIN 保存服务记录，服务记录具体格式后文说明。

检测 LC 有效后，LBSP 检测 ZK-PL 是否合法：LBSP 使用验证密钥计算验证变量多项式。

$$\begin{cases} g_l^{L_v(s)} = g_l^{1 \cdot l_0(s) + \sum_{i=1}^m v_i l_i(s)} \\ g_r^{R_v(s)} = g_r^{1 \cdot r_0(s) + \sum_{i=1}^m v_i r_i(s)} \\ g_o^{O_v(s)} = g_o^{1 \cdot o_0(s) + \sum_{i=1}^m v_i o_i(s)} \end{cases} \quad (19)$$

然后检测下面式子是否成立。

$$\begin{cases} e(g_l^{L_p(s)}, g_l^{\alpha_l}) = e(g_l^{\alpha_l L_p(s)}, g^1) \\ e(g_r^{R_p(s)}, g_r^{\alpha_r}) = e(g_r^{\alpha_r R_p(s)}, g^1) \\ e(g_o^{O_p(s)}, g_o^{\alpha_o}) = e(g_o^{\alpha_o O_p(s)}, g^1) \end{cases} \quad (20)$$

$$e(g_l^{L_p(s)} g_r^{R_p(s)} g_o^{O_p(s)}, g^{\beta \gamma}) = e(g^{Z(s)}, g^{\gamma}) \quad (21)$$

$$\begin{aligned} & e(g_l^{L_p(s)} g_l^{L_v(s)}, g_r^{R_p(s)} g_r^{R_v(s)}) = \\ & e(g^{t(s)}, g^{h(s)}) e(g_l^{O_p(s)} g_l^{O_v(s)}, g^1) \end{aligned} \quad (22)$$

其中， $e(g^x, g^y) = e(g^y, g^x) = e(g^{xy}, g^1) = e(g^1, g^{xy}) = g^{xy}$ ，称为双线性映射。双线性映射将加密空间 $\mathcal{G}_1$ 中的两点 x, y 映射为另一加密空间 $\mathcal{G}_2$ 中的两点的积 xy ，如图 3 所示。

检测 ZK-PL 合法后，LBSP 返回用户请求的服务并生成服务记录。

$$rec : \{pk_{LBSP}, DLC, index, times\} \quad (23)$$

其中， pk_{LBSP} 为该 LBSP 的唯一公钥； DLC 为本

次服务用户所用的 DLC; $index$ 为本次提供的位置服务类型; $times$ 为截至当前提供该类型位置服务给该用户的次数。生成服务记录之后, LBSP 以远程通信方式请求区块链网络中某个 CIN 保存服务记录。收到请求的 CIN 以同样的方式将服务记录广播至区块链网络, 等待“挖矿”保存在公共账本中。如上文所述, 之后 LBSP 可通过查询账本中服务记录的内容判断用户使用的 LC 是否有效。

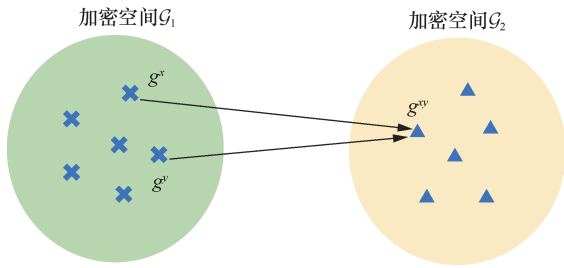


图3 双线性映射示意
Figure 3 Bilinear mapping diagram

至此, 用户完成了证明其在 0 域 1 区并获得了相应的奖励, 其他应用场景交互流程类似, 由于篇幅限制, 本文不再详细讨论。下文将对提出的位置证明系统的安全性进行分析, 并进行实验研究位置精度等级对该位置证明系统效率的影响。

4 性能评估

本节讨论提出的位置证明系统面对数种可能攻击的安全性, 以及位置精度对系统性能的影响。

4.1 安全性

(1) 身份欺骗

身份欺骗指的是恶意用户向 CIN 提供与签名私钥不匹配的公钥, 顶替其他用户生成 LC 证书。在 LC 签发协议中, LC 证书参数 $serialno$ 、 $time$ 以及 $number$ 均由 CIN 产生, 而 CIN 由嵌入的智能合约保证其行为诚实, 因此恶意的用户只能对 CIN 进行身份欺骗, 即发送假身份 pk'_{user} 给 CIN。但 CIN 可通过检测式(24)是否成立来判断用户提供的 pk'_{user} 是否合法。

$$D_{pk'_{user}}(E_{sk_{user}}(Hash(inf_{user}))) = Hash(inf_{user}) \quad (24)$$

其中, $D_{pk'_{user}}$ 与 $E_{sk_{user}}$ 分别表示用 pk'_{user} 解密和用 sk_{user} 加密。如果用户提供的 pk'_{user} 与用户用于签

名的 sk_{user} 不匹配, 则检测不通过。

(2) 服务欺骗

服务欺骗指的是不拥有 LC 的恶意用户生成假证明骗取 LBSP 的位置服务以及恶意用户重复利用同一 LC 骗取某些特殊的、请求次数有限的位置服务。可通过检测式(18)~式(20)是否成立来杜绝第一种行为。可通过查询公共账本中的服务记录来杜绝第二种行为, 即查询用户使用的 LC 是否已经超过其使用次数限制。

(3) 隐私推导

隐私推导指的是恶意攻击者利用可见协议漏洞、DLC 及 ZK-PL 推导用户位置隐私及位置隐私包含的其他上下文信息。在 LC 签发过程中, 用户位置信息只有用户自身以及 CIN 可获得, 但是因为 CIN 由内嵌智能合约限制, 不会泄露用户位置隐私。并且, 公共账本上只存储了 DLC, 哈希函数的抗碰撞性及其逆向不可计算性可保证无法从 DLC 中得到 LC, 从而不会泄露用户隐私。在 LC 验证协议中, 单项陷门函数可保证恶意用户无法从 CRS 以及 ZK-PL 中获取见证。因此, LBSP 只能获得其应知晓的信息而不会获得额外的隐私信息。

4.2 系统效率

下面通过实验仿真研究位置精度等级, 即 CIN 编码位数对系统性能的影响。实验环境如下: 主机型号 MateBook 13、1.8 GHz i7 处理器、8 GB RAM。实验基于 Github 开源项目 Circom^[19]和 Snarkjs^[20]进行二次开发, 主要开发语言为 JavaScript。Circom 是将计算问题(程序)转化为算术电路的电路生成器, 而 Snarkjs 是可以对 Circom 生成的计算电路进行满足性验证的密码学证明系统。由于提出的位置证明系统主要时延来自 ZK-PL 的产生与验证, 因此实验针对位置精度等级 4 到位置精度等级 70 的零知识位置证明系统, 测试其 LC 验证过程各阶段执行时间, 得到如图 4 所示结果(位置精度等级 70 对应 CIN 编码位数为 70, 可对量级 10^{21} 的 CIN 进行编号。假设每个 CIN 覆盖范围为 1 m^3 , 则 70 位的 CIN 可覆盖范围为 $1 \times 10^{21} \text{ m}^3$ 。而地球体积约为 $1 \times 10^{12} \text{ m}^3$, 因此, 70 位的 CIN 编码满足实际需求)。

从实验结果图可知: 计算见证的执行时间基

本不随位置精度等级变化, 稳定在 650 ms 左右; 同样地, 验证 ZK-PL 的执行时间不随位置精度变化, 稳定在 3 800 ms 左右。可信初始化的执行时间大致随位置精度等级呈线性增长, 最小值为 1 024 ms, 最大值为 4 299 ms; 生成 ZK-PL 的执行时间随位置精度等级增加而始化时执行一次, 此后每次不同位置服务过程无须再执行, 因此提出的位置证明系统的总执行时间可大致估计为其余三者时间总和, 约为 5 240~6 686 ms。该执行时延满足大部分位置服务, 如基于位置的个性推荐服务、基于位置的社交网络、基于位置的定点签到服务等。但对于一些时延十分敏感的服务, 如实时定位导航, 该系统有所欠缺。因此, 此后的工作将围绕如何缩短系统时延展开, 将从协议流程、电路生成器、密码学证明系统^[21]等方面出发, 优化提出的零知识位置证明系统。

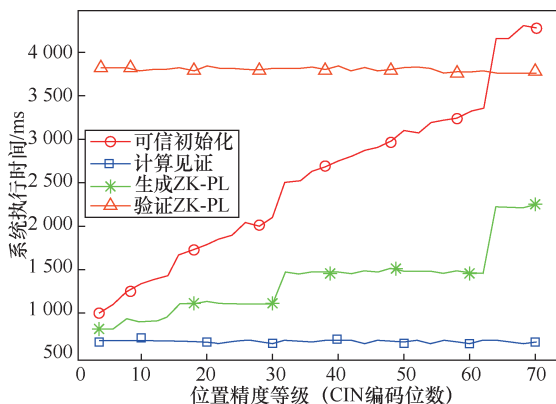


图4 系统效率与位置精度等级关系

Figure 4 Diagram of system efficiency and position accuracy

零知识位置证明系统首次将零知识证明引入位置证明系统中。零知识证明协议保证了它的隐私保护性, 另外, 5~7 s 的延迟可满足绝大多数位置服务。因此, 零知识位置证明系统有着较好的隐私性和效率。

5 结束语

本文设计了一种新型的位置证明系统架构, 并且在此系统架构的基础上, 首次引入零知识证明, 设计了一种基于 ZK-SNARK 的位置证明协议, 由此二者组成零知识位置证明系统。该零知识位置证明系统允许用户自主选择披露的 LC 参数, 提供任意精度的位置证明, 实现用户可控的

分级隐私保护。此外, 本文对提出的零知识位置证明系统安全性进行理论分析, 分析表明, 该系统可应对可能的安全攻击。最后, 针对位置精度等级对系统效率的影响, 进行实验探究。实验结果表明: 位置证明系统执行时延随位置精度等级增加而增大, 等级 4~70 的执行时延为 5 240~6 686 ms。70 位的编号可对量级为 10^{21} 的 CIN 进行编码, 该量级的 CIN 可覆盖足够大的范围, 具有较好的实用性。另外, 除了少部分时延非常敏感的位置服务外, 如实时定位导航, 提出的零知识位置证明系统的时延可满足大部分位置服务, 如基于位置的个性推荐服务、基于位置的社交网络、基于位置的定点签到服务等。之后, 笔者将考虑通过优化协议流程, 采用不同的电路生成器以及密码学证明系统等方式优化零知识位置证明系统性能, 使之适用于所有的位置服务。

参考文献:

- [1] JUNGLAS I A, WATSON R T. Location-based services[J]. Communications of the ACM, 2008, 51(3): 65.
- [2] 任星怡, 宋美娜, 宋俊德. 基于位置社交网络的上下文感知的兴趣点推荐[J]. 计算机学报, 2017, 40(4): 824-841.
REN X Y, SONG M N, SONG J D. Context-aware point-of-interest recommendation in location-based social networks[J]. Chinese Journal of Computers, 2017, 40(4): 824-841.
- [3] TYAGI A K, SREENATH N. Location privacy preserving techniques for location based services over road networks[C]//2015 International Conference on Communications and Signal Processing (ICCSP). 2015: 1319-1326.
- [4] JAVALI C, REVADIGAR G, RASMUSSEN K B, et al. I am Alice, I was in wonderland: secure location proof generation and verification protocol[C]//2016 IEEE 41st Conference on Local Computer Networks (LCN). 2016: 477-485.
- [5] LI Y, ZHOU L, ZHU H, et al. Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks[J]. IEEE Internet of Things Journal, 2015, 3(4): 563-571.
- [6] ZHU Z, CAO G. Toward privacy preserving and collusion resistance in a location proof updating system[J]. IEEE Transactions on Mobile Computing, 2013, 12(1): 51-64.
- [7] WANG X, PANDE A, ZHU J, et al. STAMP: enabling privacy-preserving location proofs for mobile users[J]. IEEE/ACM Transactions on Networking, 2016, 24(6): 3276-3289.
- [8] NOSOUHI M R YU, S, GROBLER M., et al. PARSE: privacy-aware and collusion resistant location proof generation and verification[C]//2018 IEEE Global Communications Conference (GLOBECOM). 2018: 1-6.
- [9] AMORETTI M, BRAMBILLA G, MEDIOLI F, et al. Blockchain

- based proof of location[C]//2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). 2018: 146-153.
- [10] TSCHORSCH F, SCHEUERMANN B. Bitcoin and beyond: a technical survey on decentralized digital currencies[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3): 2084-2123.
- [11] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. 2008.
- [12] 章峰, 史博轩, 蒋文保. 区块链关键技术及应用研究综述[J]. 网络与信息安全学报, 2018, 4(4): 22-29.
- ZHANG F, SHI B X, JIANG W B. Review of key technology and its application of blockchain [J]. Chinese Journal of Network and Information Security, 2018, 4(4): 22-29.
- [13] 沈鑫, 裴庆祺, 刘雪峰. 区块链技术综述[J]. 网络与信息安全学报, 2016, 2(11): 11-20.
- SHEN X, PEI Q Q, LIU X F. Survey of block chain[J]. Chinese Journal of Network and Information Security, 2016, 2(11): 11-20.
- [14] SHOKRI R, THEODORAKOPOULOS G, BOUDEDEC J Y. et al. Quantifying location privacy[C]//2011 IEEE Symposium on Security and Privacy. 2011: 247-262.
- [15] Ethereum: a next-generation smart contract and decentralized application platform[EB].
- [16] BITANSKY N, CANETTI R, CHIESA A, et al. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again[C]//The 3rd Innovations in Theoretical Computer Science Conference. 2012: 326-349.
- [17] GENNAROR, GENTRY C, PARNO B, et al. Quadratic span programs and succinct NIZKs without PCPs[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2013: 626-645.
- [18] PETKUS M. Why and how ZK-SNARK works[J]. ArXiv, abs/1906.07221, 2019.
- [19] Circom: Circuit compiler for zk-SNARKs[EB].
- [20] Snarkjs: zk-SNARK implementation in JavaScript[EB].
- [21] BEN-SASSON E, CHIESA A, TROMER E, et al. Succinct noninteractive zero knowledge for a von neumann architecture[C]//23rd USENIX Security Symposium (USENIX Security 14), 2014: 781-796.

[作者简介]



吴炜 (1995-), 男, 安徽安庆人, 同济大学硕士生, 主要研究方向为区块链和隐私保护。



刘儿元 (1973-), 男, 福建福安人, 同济大学教授、博士生导师, 主要研究方向为位置服务、定位导航、人工智能和区块链。



杨昌鑫 (1997-), 男, 河南信阳人, 同济大学硕士生, 主要研究方向为区块链。



王睿 (1984-), 男, 安徽宣城人, 同济大学副教授、博士生导师, 主要研究方向为无线通信和智能信号处理。