文章编号:1007－6735(2026)05－0566－11　DOI: 10.13255/j.cnki.jusst.20250827001

云环境下基于局部敏感哈希的室内定位隐私保护

王　澍，　姚书文，　乐燕芬
（上海理工大学 光电信息与计算机工程学院，上海 200093）

摘要：随着云计算在位置相关服务中的应用与普及，不可信云环境可能导致定位服务中敏感信息的泄露。现有云环境下的定位隐私保护通常需要服务商与云端联合计算实现加密域的安全定位，计算和通信效率较低。针对这些问题，提出一种利用局部敏感哈希构建指纹特征布隆过滤器的室内指纹定位隐私保护方案，将指纹信号距离匹配转化为可安全外包至云服务器的布隆过滤器内积计算，完成最近邻检索；并提出一种局部定位方法，减少在地图大数据中云端在线检索的参考点空间，从而有效降低计算开销。理论分析和实验结果表明，所提方案有效平衡了定位精度、计算效率和数据安全。

关键词：室内定位；隐私保护；布隆过滤器；局部敏感哈希；局部定位
中图分类号：TP 391　　**文献标志码：**A

Privacy preserving indoor localization based on locality-sensitive hashing in cloud environments

Wang Shu,　Yao Shuwen,　Le Yanfen
(*School of Optical−Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*)

Abstract: With the widespread adoption of cloud computing in location-related services, untrusted cloud environments may lead to the leakage of sensitive information in positioning services. Existing privacy-preserving positioning schemes in cloud environments typically require collaborative computation between service provider and the cloud to achieve secure positioning in the ciphertext. Moreover, these approaches are inefficient in computation and communication. To tackle these issues, a privacy-preserving indoor fingerprint positioning scheme was proposed. The scheme leveraged locality-sensitive hashing (LSH) to construct Bloom filters for fingerprint features, and transformed fingerprint signal

收稿日期: 2025−08−27
基金项目: 国家自然科学基金资助项目(62172281)
第一作者: 王　澍(1997—)，男，硕士研究生．研究方向：室内定位、隐私保护．E-mail：wasswaggy@163.com
通信作者: 乐燕芬(1978—)，女，副教授．研究方向：室内定位、隐私安全等．E-mail：leyanfen@usst.edu.cn

引文格式: 王澍，姚书文，乐燕芬．云环境下基于局部敏感哈希的室内定位隐私保护[J]．上海理工大学学报，2026，48(5)：566-576.
Citation: Wang Shu, Yao Shuwen, Le Yanfen. Privacy preserving indoor localization based on locality-sensitive hashing in cloud environments[J]. Journal of University of Shanghai for Science and Technology, 2026, 48(5): 566-576.

distance matching into Bloom filter inner product computation. This computation could be securely outsourced to the cloud for nearest neighbor search. Furthermore, a local positioning method was introduced to reduce the search space of reference points in large-scale map data during cloud-based online retrieval, effectively reducing computational overhead. Theoretical analysis and experimental results demonstrate that the scheme achieves an effective balance among positioning accuracy, computational efficiency, and data security.

Keywords: *indoor localization*; *privacy preserving*; *Bloom filter*; *locality-sensitive hashing*; *local positioning*

随着数字化服务的发展，位置相关技术在各领域的应用需求不断增长。户外活动时可以通过全球导航卫星系统（global navigation satellite systems, GNSS），如北斗、全球定位系统（global positioning system, GPS）或者伽利略（Galileo），来获取地理位置。然而，当卫星信号被障碍物遮蔽，尤其是在大型商场或机场等室内环境中时，GNSS 的定位精度会降低，甚至不可用。因此，一些针对室内定位的技术逐渐被开发和应用，如利用 Wi-Fi（wireless fidelity）、超宽带（ultra-wide band, UWB）、蓝牙等无线信号中携带的特定定位信息完成定位[1-2]。其中使用 Wi-Fi 信号的指纹定位已在配有 Wi-Fi 接入点（access point, AP）的公共建筑中广泛使用。基于 Wi-Fi 的室内指纹定位通常包括离线指纹库构建和在线定位服务两个阶段。在离线指纹库构建阶段，定位服务提供商（location service provider, LSP）完成对定位区域内预设参考点（reference point, RP）上各 AP 的无线接收信号强度（received signal strength, RSS）采样，并由此构建位置指纹库。在在线定位服务阶段，请求定位服务的用户利用自身扫描获取的 RSS，构建定位请求指纹，由 LSP 采用某种匹配方法[3] 与指纹库中参考点的指纹比较，完成用户的位置估计。在定位服务过程中，用户位置信息与 LSP 指纹库资源均属于敏感隐私数据，一旦泄露，恶意攻击者可以识别预测用户的行动轨迹[4]，并攻击 LSP，造成资产损失。

针对室内定位中用户位置和 LSP 数据的隐私安全问题，国内外已有不少研究提出了多种具有隐私保护的室内定位（privacy-preserving indoor localization, PPIL）方案，涉及的隐私保护技术主要包括同态加密[5-6]、差分隐私[7] 及 K 匿名[8] 等。其中：同态加密方案虽然能保证数据的可用性和安全性，但定位服务过程中用户需要完成大量的加密和解密计算，在拥有众多 AP 和 RP 数量的定位区域中，方案的实时性较低；基于差分隐私的定位方案虽然具有较高的计算效率，但通常会对用户指纹添加噪声，导致定位精度降低；K 匿名方法将用户真实位置信息隐藏在 K–1 个虚假信息中，缺点是如果攻击者拥有额外背景知识，这种匿名性会被破坏，导致隐私强度变弱。

近年来，为了解决 PPIL 方案中较高的通信开销和计算开销问题，越来越多的研究方案选择将繁重的计算外包给云服务器（cloud server, CS）[9-12]。乐燕芬等[9] 基于非对称内积标量保留加密（asymmetric scalar-product-preserving encryption, ASPE）算法对 LSP 指纹信息离线加密后上传给 CS，由其完成在线阶段用户的定位请求，算法的计算和通信开销很小，但要求用户必须是可信的，且无法抵御已知明文攻击。Eshun等[11] 结合半同态加密、不经意传输和混淆电路，把耗费大量计算资源的指纹匹配和最近邻检索外包给不互谋的 LSP 和 CS，从而提高计算效率。该方案能有效保护定位服务过程中用户位置隐私和 LSP 指纹库隐私，但每次定位服务，LSP与 CS 之间存在大量通信开销。张应辉等[12] 也采用了类似的系统架构，由协助 LSP完成每个用户的 K 个匿名定位请求指纹的最近邻匹配。在整个定位服务过程中，云服务器在密文域进行相关运算，保障了 LSP 的指纹资源和用户的位置隐私，但用户只获得 K 匿名的隐私保护强度，且在指纹具有较高维度或为达到较强的保护增大 K 值时都会给用户带来大量的同态加密计算。上述 PPIL 方案[10-12] 为响应每一个定位服务请求，在线阶段都需要 LSP 和 CS 参与计算并频繁通信，并没有真正实现计算任务的外包。

针对以上问题，基于局部敏感哈希（locality-sensitive hashing, LSH）算法，本文提出了一种云环

境下的轻量级室内定位隐私保护方案，利用保持数据特征相似性的LSH算法，将指纹数据进行哈希处理后构建各指纹的特征布隆过滤器（Bloom filter, BF），提交到CS，由其利用用户与RP指纹间特征布隆过滤器的内积确定最近邻。同时为提高大定位场景中的定位效率，基于LSH对指纹数据进行分桶，可实现从局部再精确的二级定位。安全分析与实验结果表明，本文所提方案能够有效保护用户位置信息与LSP资源隐私，且具有较高的计算效率和定位实时性。

1 背景知识

1.1 室内Wi-Fi指纹定位

Wi-Fi指纹定位是一种基于RSS特征匹配的室内定位技术，指纹是与特定RP对应的一组AP的RSS向量。LSP在离线阶段构建指纹库：

$$\boldsymbol{D}=[i,(x_i,y_i),\boldsymbol{V}_i=\{v_{i1},v_{i2},\cdots,v_{iN}\}|i=1,2,\cdots,M] \quad (1)$$

式中：i是定位区域也即地图内的索引；(x_i,y_i)是第i个RP坐标；$\boldsymbol{V}_i$表示第i个RP指纹；N和M分别是地图内AP和RP的数量。在线阶段，请求定位的用户扫描各AP信号后的定位请求指纹表示为$\boldsymbol{V}'=\{v'_1,v'_2,\cdots,v'_N\}$，随后将其发送给LSP完成最近邻匹配，该匹配过程通常采用欧氏距离作为相似性度量指标：

$$d_i=\|\boldsymbol{V}_i-\boldsymbol{V}'\|^2=\sum_{j=1}^{N}(v_{ij}-v'_j)^2 \quad (2)$$

目前已有不少PPIL方案通过对此距离公式的分解来构建加密算法，完成距离的秘密计算[6,12]。本文所提方案将对该距离通过哈希运算进行变换，在隐藏指纹明文信息的同时完成最近邻检索。

1.2 布隆过滤器

布隆过滤器（BF）是一种空间效率较高的概率数据结构[13]。通常BF是一个由u个独立哈希函数$\mathcal{H}=\{h_1,h_2,\cdots,h_u\}$生成的长度为$l$比特的位数组。数组中所有元素初始状态设置为“0”，当元素v需要插入时，将$h_i(v)\bmod l$的值作为位数组的索引，对应索引位置设置为“1”，其中$1<i<u$。当查询某个元素时，对其使用相同的哈希函数，若u个索引位置同时为“1”，表明该元素可能存在。由于多个元素可能映射到BF中的相同位置，BF存在误判率。

本文利用哈希映射后的指纹构建BF，将基于欧式距离的指纹匹配转化为BF的内积匹配，这不仅有效提升了匹配速度，还可以隐匿真实指纹信息，保障LSP指纹数据与用户位置指纹的隐私安全。

1.3 局部敏感哈希

局部敏感哈希（LSH）是高维空间近似最近搜索（approximate nearest neighbor, ANN）问题中最常用的解决方法之一[14]。LSH通过随机哈希函数将高维数据映射到低维表示，每个哈希函数独立地将数据点分配到不同的哈希桶中，同时最大程度地保持原数据点之间的相似性，从而确保相似数据点以更高的概率映射到相同的哈希桶。

1.3.1 MinHashLSH

MinHashLSH通过对MinHash签名应用LSH，仅对具有较高相似度概率的数据点进行比较[15]。在处理多组签名时，构建一个签名矩阵，其中每一列表示一组数据的所有MinHash值。针对给定的相似度阈值，可选择一个最优的分组数量B，将签名矩阵按行划分为B个分组，每个分组包含R行，从而构建MinHashLSH结构。当两个数据的MinHash签名在任意分组中被映射到相同的哈希桶时，这两个数据的相似度大于或等于阈值的概率较高。

本方案利用MinHashLSH对离线指纹进行哈希处理，将相似指纹映射到同一个哈希桶内。在线定位时，可将用户的定位请求指纹做同样的处理，从而确定其所在的哈希桶，由此实现对用户的局部定位。

1.3.2 E2LSH

E2LSH是一种基于2-stable分布（标准正态分布）的LSH算法[14]，可以将一个高维数据映射为一个整数，并保持映射前后数据间的欧氏距离关系。其核心思想是通过式(3)，将N维向量$\boldsymbol{V}$映射为一个整数，使得在欧氏空间中距离较近的数据映射为相同的哈希值。其中，$\boldsymbol{a}$是一个N维随机向量，每一维随机且独立地服从2-stable分布，b是对每个$\boldsymbol{a}^{\mathrm{T}}\cdot\boldsymbol{V}$分量在区间$[0,w]$内均匀选取的实数，$w$是一个常数。

$$h_{\boldsymbol{a},b}(\boldsymbol{V})=\left\lfloor\frac{\boldsymbol{a}^{\mathrm{T}}\cdot\boldsymbol{V}+b}{w}\right\rfloor \quad (3)$$

本文方案利用E2LSH将指纹向量映射为一组整数，并将其作为布隆过滤器的索引，过程中保

持了指纹在欧氏空间的距离关系。

2　基于 LSP 的隐私定位方案

2.1　方案框图

本文所提方案框架如图 1 所示，CS 是执行定位计算的主体。在离线阶段，LSP 对指纹库指纹分别进行 MinHash 和 E2LSH 处理，得到表征指纹特性的 MinHash 签名和布隆过滤器，并将它们与加密的参考点坐标一起发送给 CS，由 CS 构建 LSH 查询结构。在线阶段，请求定位服务的用户从 LSP 获取哈希函数，将在线测得的 RSS 指纹经哈希处理后发送给 CS。CS 在查询结构中查找相似签名的索引，完成局部定位。随后，CS 计算得到该局部区域内与用户布隆过滤器的前 k 个最大内积对应的索引，在密文域对其相应的参考点执行 k-NN 算法，结果返回给用户，由其解密获得自身的估计位置。

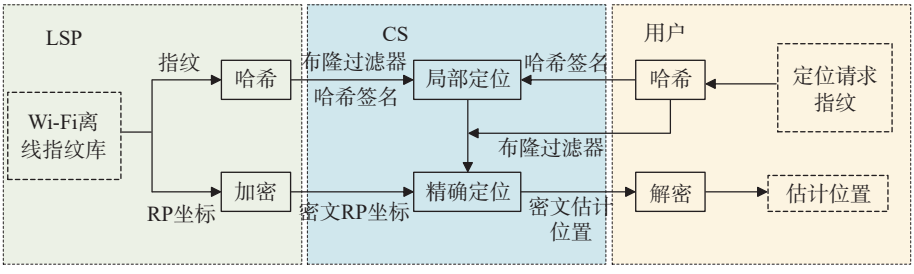

图 1　本文方案框架

Fig. 1　Structure of the proposed scheme

2.2　威胁模型

本方案同时考虑用户的位置隐私和 LSP 的资源隐私。在基于外包云环境的室内定位服务中，设定 CS 是半诚实的，这意味着它可能出于盈利目的窃取用户的位置信息和 LSP 资源。此外，CS 作为一个开放平台，可能会在执行协议的过程中遭到外部敌手的攻击，导致数据泄露。对于用户位置隐私，攻击者主要有以下 2 种攻击方式：

a. 攻击者通过窃取用户的指纹信息，间接推测用户的位置信息；

b. 攻击者直接从定位结果中获取用户位置信息。

对于 LSP 的资源隐私，攻击者包括以下 2 种攻击方式：

a. 攻击者通过窃取参与定位计算的数据和定位结果，构建与 LSP 指纹库 $\boldsymbol{D}$ 相似的指纹库 $\boldsymbol{D}'$；

b. 攻击者通过发起穷举攻击，利用大量用户指纹信息及其估计位置，直接推测真实指纹库 $\boldsymbol{D}$。

2.3　方案具体设计

2.3.1　局部定位

通常 Wi-Fi 指纹定位方案在每次执行定位服务时，需在整个指纹库中匹配用户的最近邻。当这一过程在密文域进行时，往往会产生显著的计算开销。目前已有不少研究提出了有效的方案来提高匹配的实时性[16]。本文方案通过衡量指纹向量之间的相似度，利用 MinHashLSH 算法将指纹库中的指纹映射至多个哈希桶，从而在在线阶段通过确定用户查询签名所属的哈希桶实现快速的局部定位。具体过程包括以下步骤：

a. LSP 利用 MinHash 生成哈希签名 $\boldsymbol{S}_i$

离线阶段，LSP 利用一组包含 u 个独立哈希函数的哈希数组 $\boldsymbol{H}=\{h_1,h_2,\cdots,h_u\}$ 对指纹向量 $\boldsymbol{V}_i=\{v_{i1},v_{i2},\cdots,v_{iN}\}$ 中的每个 RSS 值进行哈希计算，生成哈希签名：

$$\boldsymbol{S}_i(\boldsymbol{V}_i)=\{\min(h_1(\boldsymbol{V}_i)),\min(h_2(\boldsymbol{V}_i)),\cdots,\min(h_u(\boldsymbol{V}_i))\}^{\mathrm{T}} \tag{4}$$

式中：$\min(\cdot)$ 是对生成的 N 个哈希值取最小值，作为该哈希函数对应的签名分量；$\boldsymbol{S}_i(\boldsymbol{V}_i)$ 是指纹向量 $\boldsymbol{V}_i$ 对应的哈希签名。随后，LSP 将 $\{\boldsymbol{S}_i\}_{i=1}^{M}$ 发送至 CS。

b. 用户生成查询签名 $\boldsymbol{S}_{\mathrm{u}}$

在线阶段，用户利用从 LSP 获取的哈希函数 $\boldsymbol{H}$ 对自身采集的指纹 $\boldsymbol{V}'$ 进行哈希处理，生成查询哈希签名：

$$\boldsymbol{S}_{\mathrm{u}}(\boldsymbol{V}')=\{\min(h_1(\boldsymbol{V}')),\min(h_2(\boldsymbol{V}')),\cdots,\min(h_u(\boldsymbol{V}'))\}^{\mathrm{T}} \tag{5}$$

并将其发送给 CS。

c. CS 完成用户局部定位

在离线阶段，CS 利用 LSP 上传的哈希签名

$\boldsymbol{S}_i$，构建签名矩阵 $\boldsymbol{S}$：

$$\boldsymbol{S}=[\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_M]=\begin{bmatrix}\min(h_1(\boldsymbol{V}_1)) & \cdots & \min(h_1(\boldsymbol{V}_M))\\ \vdots & & \vdots\\ \min(h_u(\boldsymbol{V}_1)) & \cdots & \min(h_u(\boldsymbol{V}_M))\end{bmatrix} \tag{6}$$

并按照 MinHashLSH 的构建原理将 $\boldsymbol{S}$ 划分为 B 段，每段包含 R 行。接着 CS 对每段签名通过特定哈希函数计算每段的哈希值：$H_B(\boldsymbol{S}_i^{(j)})=\text{hash}(\boldsymbol{S}_i^{(j)}),j=1,2,\cdots,B$，即将每段签名经字节顺序反转后拼接，生成固定长度的哈希值。具有相同哈希值的分段签名会被分配到同一个哈希桶内。

在线阶段，CS 接收到用户的查询签名 $\boldsymbol{S}_\text{u}$ 后对其进行相同的哈希和分段处理，获取与 $\boldsymbol{S}_\text{u}$ 位于同一哈希桶内的签名索引，即对应参考点坐标的索引，完成局部定位。

2.3.2 精确定位

在基于 Wi-Fi 指纹的 PPIL 中，不少方案利用同态加密算法完成密文域的指纹距离计算后，通过 k-NN 算法完成用户位置估计。然而，1 024 bit 或 2 048 bit 的密钥长度虽然增强了安全性，却显著增加了计算开销，影响定位服务的实时性。本文方案通过结合 E2LSH 算法和布隆过滤器，将原指纹之间的欧氏距离关系映射到布隆过滤器做内积对比，完成 k-NN 定位。避免了同态加密的计算负担，在隐匿原指纹信息的同时，提高定位的实时性。具体包括以下步骤。

a. 准备阶段

在离线阶段，LSP 为每个参考点初始化一个布隆过滤器，并通过式 (3) 生成一组哈希函数：

$$\mathcal{H}=\{\text{h}_{a1,b1},\text{h}_{a2,b2},\cdots,\text{h}_{au,bu}\} \tag{7}$$

之后，构建每个指纹的布隆过滤器：

$$\text{BF}_\text{index}(\boldsymbol{V}_i)=\left(\sum_{j=1}^{u}r_j\text{h}_{aj,bj}(\boldsymbol{V}_i)\right)\text{mod}\,l \tag{8}$$

式中：r_j 为随机整数；l 为 BF 长度。

具体来说，就是将指纹库中的每个指纹 $\boldsymbol{V}_i=\{v_{i1},v_{i2},\cdots,v_{iN}\}$ 映射为 u 个整数作为索引，并将第 i 个布隆过滤器 BF_i 中对应索引位置的值设为元素“1”。由于 E2LSH 能保持指纹数据间欧氏距离的大小关系，距离相近的指纹会以更高的概率映射在 BF 索引相同的位置，即以高概率发生哈希冲突，因此映射后 BF 会有更多“1”的位置相同。

图 2 展示了针对 3 条指纹 $\boldsymbol{V}_1$、$\boldsymbol{V}_2$ 和 $\boldsymbol{V}_3$，基于本文方法构建 3 个长度为 200 bit 的指纹 BF 实例。其中，指纹 $\boldsymbol{V}_1$ 与 $\boldsymbol{V}_2$ 的欧式距离较近，$\boldsymbol{V}_2$ 与 $\boldsymbol{V}_3$ 的欧式距离较远。BF 的具体构建流程如下：首先对 3 条指纹分别通过式 (8) 进行映射，将 RSS 指纹向量转化为 3 组整数索引，分别为 (2,3,⋯,198,200)、(2,3,⋯,197,198,200) 及 (1,⋯,198,200)；随后以每组整数作为 BF 的比特位索引，将对应索引位置的值置“1”，其余位置保持“0”，最终生成图 2 所示的 3 个 BF 结构。由于 E2LSH 的局部敏感特性，欧式距离更近的 $\boldsymbol{V}_1$ 和 $\boldsymbol{V}_2$ 在 BF 中重合的“1”位数更多，二者的内积值更大。而 $\boldsymbol{V}_2$ 和 $\boldsymbol{V}_3$ 则相反，BF 中同一位置为“1”的位数较少，其内积值更小。通过这种映射，哈希签名之间的内积保留了原指纹之间欧式距离的关系。

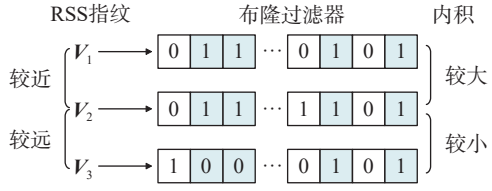

图 2 构建布隆过滤器

Fig. 2 Construction of Bloom filter

值得注意的是，使用 E2LSH 将指纹映射到 BF 的过程中，局部敏感特性只发生在很小的范围内，因此，需要对指纹做标准化处理：

$$\boldsymbol{V}_{\text{nor}i}=\boldsymbol{V}_i\cdot\frac{1}{\max\|\boldsymbol{V}\|_2},\ i=1,2,\cdots,M \tag{9}$$

式中：$\boldsymbol{V}_{\text{nor}i}$ 是标准化后的指纹向量；$\max\|\boldsymbol{V}\|_2$ 是所有原始指纹向量的欧几里德范数的最大值。但即使这样，LSH 仍然是一种概率性算法，两个较近指纹只会以一定概率发生哈希冲突，导致映射到 BF 后距离保持也是概率性的，这样会影响指纹的准确匹配。为了增大碰撞概率，保证定位精度，本文方案采用 L 组独立的 E2LSH 将指纹向量映射为 L 组 BF，并将每组 BF 按位级联，从而使距离相近的指纹有更多元素“1”的位置相同。图 3 给出了对指纹 $\boldsymbol{V}_i$ 构建完整 BF_i 的过程。

b. 用户指纹映射为 BF_u

在线阶段，待定位用户完成认证后，利用从 LSP 收到的哈希函数对自身的定位请求指纹按上述相同的方式映射为 BF_u，并将其发送给 CS。

c. CS 完成 k 近邻检索

假设 CS 已根据局部定位确定了与 $\boldsymbol{S}_\text{u}$ 位于同一哈希桶内的签名索引，设为 λ 个，提取对应的 λ 个布隆过滤器 $\{\text{BF}_{\text{index}1},\cdots,\text{BF}_{\text{index}\lambda}\}$，并计算其与 BF_u

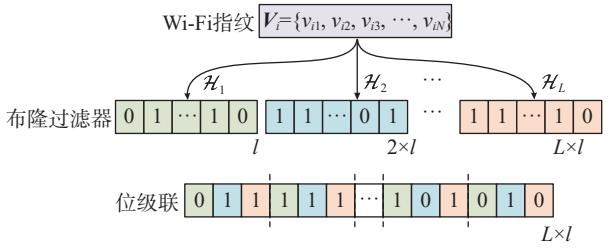

图 3　分组映射指纹与位级联

Fig. 3　Group-mapped fingerprint and bit-level concatenation

的内积，如式 (10) 所示。

$$\mathrm{BF_u}\odot\mathrm{BF_{index}}_{\lambda}=\sum_{i=1}^{L\times l}\mathrm{BF_u}[i]\cdot\mathrm{BF_{index}}_{\lambda}[i] \tag{10}$$

式中：$\odot$为 BF 之间的内积运算；$\mathrm{BF_u}[i]$和$\mathrm{BF_{index}}_{\lambda}[i]$是两个布隆过滤器在位置 i 上的值。内积计算的核心是统计两个 BF 中共同为“1”的位数量。根据指纹 BF 的构建原理可知，内积大小代表了原指纹之间特性的相似程度，也即反映了用户指纹与这 λ 个 BF 对应的参考点指纹之间的欧氏距离。两个 BF 的内积越大，说明原指纹数据越相似，其欧式距离则越近，反之则越远。例如，若：

$$\mathrm{BF_u}\odot\mathrm{BF_{index1}}\leqslant\mathrm{BF_u}\odot\mathrm{BF_{index2}} \tag{11}$$

则有

$$\mathrm{dist}(\boldsymbol{V}',\boldsymbol{V}_{\mathrm{index1}})\geqslant\mathrm{dist}(\boldsymbol{V}',\boldsymbol{V}_{\mathrm{index2}}) \tag{12}$$

随后，CS 选取内积值最大的 k 个布隆过滤器的索引，确定用户的 k 个最近邻坐标。

现有基于同态加密的 PPIL 方案通常将指纹间的距离公式，如式 (2)，分解后在密文域进行运算，其计算量会随着指纹维度增加而急剧增加。本文方案中，布隆过滤器内积运算为明文域向量点积，在线运算复杂度只与预设 BF 长度有关，与指纹维度，即 AP 数量无关，避免了密文计算的复杂性。

同时，为避免敌手或 CS 通过用户近邻 RP 推测用户位置，LSP 在离线阶段已采用 DGK[17] 加密算法对 RP 坐标进行加密处理。CS 在确定用户的 k 近邻坐标后，计算式 (12) 和 (13)，并将结果发送给用户。式中$[[\hat{x}]]$和$[[\hat{y}]]$表示确定的 k 个最近邻坐标密文同态加法结果。

$$[[\hat{x}]]=\left[\left[\sum_{i=1}^{k}x_i\right]\right]=\prod_{i=1}^{k}[[x_i]] \tag{13}$$

$$[[\hat{y}]]=\left[\left[\sum_{i=1}^{k}y_i\right]\right]=\prod_{i=1}^{k}[[y_i]] \tag{14}$$

d. 用户解密完成定位

用户收到$[[\hat{x}]]$和$[[\hat{y}]]$后，利用从 LSP 获取的私钥解密，计算质心得到自己的估计位置x_{u}和y_{u}。

3　安全性分析

本文方案涉及 3 个实体，分别是申请定位服务的用户、LSP 和 CS，其中 LSP 拥有指纹库资源，CS 是定位计算的主体。方案将用户在线测量指纹与指纹库指纹通过哈希函数分别映射为哈希签名和布隆过滤器，隐匿了真实指纹信息。本节从用户位置信息安全和 LSP 资源安全两个方面分析方案的安全性。

3.1　用户位置信息安全

用户请求定位服务时，向 CS 发送的是经过哈希运算后的查询签名$\boldsymbol{S}_{\mathrm{u}}$和布隆过滤器$\mathrm{BF_u}$。在未知哈希函数时，仅从哈希后的结果，攻击者无法获取用户位置的任何信息。进一步假设攻击者通过某种方式，比如与 CS 联合，获得了哈希函数，但哈希函数的单向性使得攻击者无法从哈希值还原用户的原始指纹信息。若攻击者利用哈希元素查询的方式进行蛮力攻击，只能获得用户可能的定位请求指纹集合$\tilde{\Lambda}=\{\tilde{V}'|\forall h_l\in\mathcal{H},\mathrm{BF_u}\left[h_l(\tilde{v}_i')\right]=1,i\in[1,N],l\in[1,L]\}$，$V'\in\tilde{\Lambda}$，且代价是指数级的，至少有$O\left(n^{NL}\right)$种可能。因此，用户真实指纹数据隐私安全性得到保障。

针对定位结果的潜在攻击，局部定位过程 CS 仅提取部分索引号，而这些索引号对应的实际数据仅为布隆过滤器的条目，因此 CS 和攻击者无法通过此过程推测用户的局部定位信息。在精确定位过程中，攻击者仅能获取 k 个二进制向量的内积计算结果和密文 RP 坐标，无法直接解析出用户的真实定位结果。因此，用户的位置隐私安全性得到保障。

3.2　LSP 资源安全

LSP 资源主要包含各个预设的参考点坐标及其对应的指纹向量。这些指纹向量通过 LSH 算法生成哈希签名和布隆过滤器参与定位过程。如 3.1 小节所述，在整个过程中攻击者即使获取了哈希函数组，也无法通过分析哈希签名和布隆过滤器来推断真实的指纹信息。对于预设的 RP 坐标值，本文采用 DGK 加密算法对其进行加密处理，使得 CS 在执行 k 近邻计算时仅能操作密文坐标。

在未拥有私钥的前提下，CS 和攻击者均无法获取真实坐标值。因此，本方案中 LSP 的资源数据在定位服务中的隐私安全性能够得到有效保障。

4 方案性能评估

4.1 理论分析

本文方案分别与文献 [11] 和文献 [12] 的方案对定位服务在线阶段的计算开销和通信开销进行了理论分析对比。3 种方案皆采用三方 PPIL 协议，但各自性能开销存在差异。文献 [11] 结合不经意传输、混淆电路技术和 DGK 加密算法，提出了一种三方数据保护的 PPIL 方案；文献 [12] 通过结合 Paillier 加密算法与 K 匿名算法，设计了一种将距离计算过程外包给 CS 的三方协议。3 种方案的理论分析如表 1 所示，其符号定义如下：h 表示一次哈希计算；T 为 DGK 加密算法的密文位长；t、m 和 e 分别表示文献 [12] 中涉及到的对称加密密文位长、RSS 值的比特长度和欧氏距离比特长度；k 为返回的最近邻的个数；Exp 和 Mul 分别表示一次模指数运算和一次模乘法运算。

表 1 可知，文献 [11] 方案中，CS 与 LSP 的频繁交互增加了整体通信开销，并加重了 LSP 的计算负担。文献 [12] 方案中 K 匿名算法虽增强了隐私保护强度，但导致每次定位服务的计算和通信开销均增加了 K 倍；而本文方案采用两种不同的局部敏感哈希算法处理指纹向量，有效降低了计算和通信开销。具体来说，本方案的计算开销主要来源于两部分：与 λ 个 BF 做内积计算产生的开销 λlL；密文坐标计算产生的开销 (k-1)Mul。对于通信开销，明文数据传输开销忽略不计，密文坐标传输仅产生 $2T$ 比特开销。此外，本文方案 LSP 在在线阶段不参与任何计算，进一步减轻了其负担。

表 1 在线阶段计算和通信开销分析

Tab. 1 Analysis of computational and communication overheads in online phase

方案	计算开销			通信开销/bit
	LSP	CS	用户	
文献[11]	mM Exp+$M(N+3)$Mul + kM	M Exp + Mul	$4N$ Exp + $8N$ Mul + kh	$MT + eMT + 2etM$
文献[12]	$3NMK$ Mul	NMK Exp + MK	NK Exp + $2NK$ Mul	$[M(N+1)+N]KT$
本文方案	—	λlL + (k−1) Mul	$2hM$ + Exp	$2T$

4.2 实验分析

本文使用公开数据集 Miskolc[18] 评估所提方案性能，数据集由 3 个楼层共 1 571 条记录构成，其中包含 RP 实际坐标和每个 RP 对 32 个 AP 的采样 RSS。同时与现有文献 [6] 和 [12] 所提方案进行对比。实验采用 Python 语言，PC 运行环境如下：Intel(R) Core(TM) i9-13900HX CPU @2.20 GHz，16GB RAM，64 bit 操作系统。使用数据集第三层作为定位地图并设置实验参数如下：指纹维数 N = 32，也即参与定位的 AP 数量为 32 个；参考点数量 M = 430；式 (3) 中 w = 0.045；布隆过滤器分组数和单组长度分别设置为 L = 10，l = 200；返回的最近邻个数 k = 4，匿名度 K = 5；加密算法的密钥长度为 1 024 bit。

4.2.1 方案开销

3 种方案一个具体实例的计算开销和通信开销如表 2 所示。

表 2 计算和通信开销对比

Tab. 2 Comparison of computational and communication overheads

方案	计算开销/ms			通信开销/bit
	LSP	CS	用户	
文献[6]	12 256.1	—	156.8	462×2 048
文献[12]	6 590.8	37 977.1	144.7	71 110×2 048
本文方案	—	5.4	2.3	2×2 048

文献 [6] 提出的方案每次进行定位服务时，利用同态加密算法在密文域计算用户指纹与指纹库指纹的距离，使得计算开销较大。文献 [12] 在运用同态加密算法的基础上，进一步对用户指纹引入了 K 匿名算法，使得单次定位需完成 K 个匿名指纹的距离匹配，导致时间开销和通信开销进一步增加。而本文方案通过利用 LSH，将密文域指纹距离比较转化为布隆过滤器内积运算，有效降低了定位服务的整体开销，在确保数据隐私得到

充分保护的同时，提升了定位服务的实时性，每次定位服务开销在毫秒内。

图 4 展示了局部定位方法对文献 [6] 和本文方案计算时间开销的影响。从图 4 中可以看出，引入局部定位后文献 [6] 方案的时间开销减少了约 70%。局部定位通过优化指纹匹配过程，减少了大量需要进行同态运算的密文指纹数量，从而缩短了计算时间。同样，本文所提方案由于局部定位的引入减少了精确定位阶段参与内积运算的布隆过滤器数量，使得整体方案时间开销减少了约 60%。

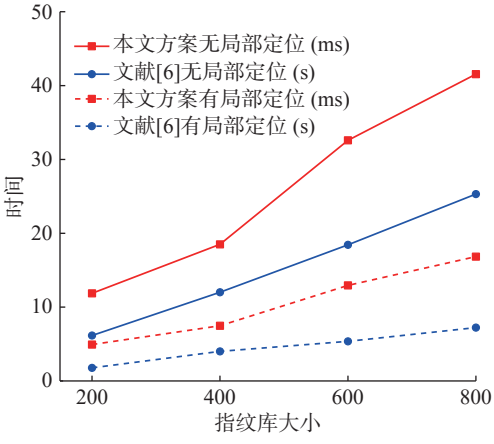

图 4　局部定位对计算开销的影响

Fig. 4　Effect of local positioning on computational overhead

4.2.2　方案定位精度

局部定位可有效提升定位的整体实时性，尤其本方案也可以与结合，缩小最近邻的搜索空间。但局部定位依赖用户指纹签名能否被映射至包含最多真实最近邻的哈希桶。图 5 展示了局部定位在 50 个测试点中引入的定位偏差，其中箭头起点表示 k-NN 定位结果，终点指向经过本文局部定位后 k-NN 定位结果，箭头方向和长度反映了局部定位引入的偏差。可以看出，50 个测试点中 60% 的定位偏差在 0.5 m 以内，且仅有 10% 的测试点定位偏差超过 2 m。通过进一步分析，产生该偏移的主要原因在于 MinHashLSH 的相似度阈值判定机制，当两组指纹的 Jaccard 相似度略低于设定阈值时，即使它们在欧氏空间中距离较近，也可能被划分至不同的哈希桶，导致真实近邻未被检索，从而引入定位误差。同时，LSH 本身具有概率性特征，不同哈希函数组对指纹进行映射时，可能对阈值边界的指纹造成不稳定分类。

图 6 给出了不同 MinHashLSH 阈值下哈希桶内候选布隆过滤器的数量变化趋势。当所设阈值较

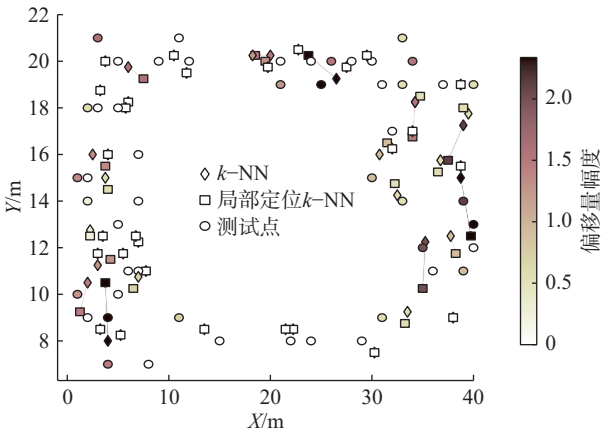

图 5　50 个测试点的局部定位定位偏移

Fig. 5　Local positioning deviation of 50 test points

低时，候选布隆过滤器数量接近全指纹库规模，此时局部定位无法有效缩减检索空间。随着阈值升高，哈希桶内指纹相似度更高，候选布隆过滤器数量显著减少。图 7 给出了不同阈值对定位精度的影响，可以看出随着阈值提升，定位误差仅微弱增加。这是因为在低相似度时，候选布隆过滤器较多，有更多的真实近邻被纳入候选。随着阈值提升，检索空间大幅缩减，部分真实近邻因相似度略低于阈值被排除，导致定位精度降低。总体而言，当阈值选定为 0.95 时，平均候选布隆过滤器数量仅 60 左右，大幅缩减了后续精确定位所需的检索空间，且保证了整体定位精度。

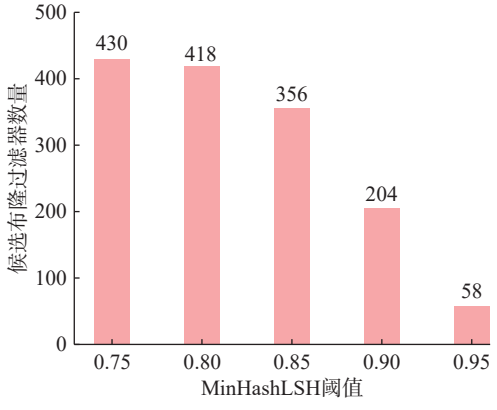

图 6　MinHashLSH 阈值对候选布隆过滤器的影响

Fig. 6　Effect of MinHashLSH threshold on candidate BF

本文精确定位阶段造成定位误差的因素是为保证指纹信号的隐私，将信号间基于欧氏距离的匹配用局部敏感哈希构建的布隆过滤器的内积大小来实现。图 8 展示了不同布隆过滤器分组数 L 对定位误差的影响。结果表明，采用多组局部敏感哈希独立映射布隆过滤器可以有效降低误差。这是因为随着 L 的增加，以更大概率实现了距离保

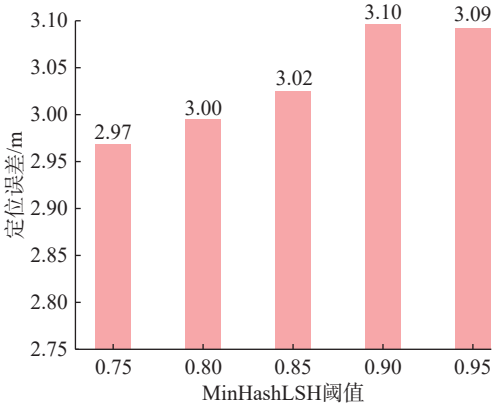

图 7 MinHashLSH 阈值对定位误差影响

Fig. 7 Effect of MinHashLSH threshold on positioning errors

全。图 9 通过定位误差的累积分布函数(cumulative distribution function, CDF)，展示了针对 50 个测试点分别采用本文所提的内积定位方法和 *k*-NN 定位的误差情况，其中 L=10。结果显示，两种方法的定位精度无明显差异。

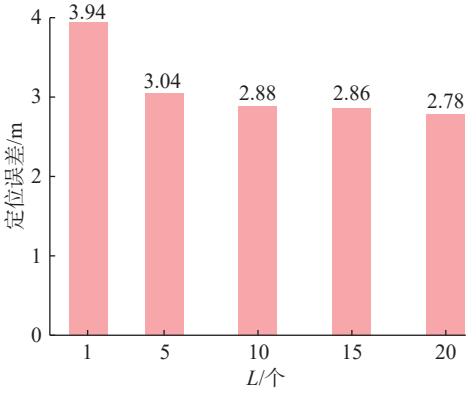

图 8 BF 分组数对定位误差影响

Fig. 8 Effect of BF grouping number on positioning errors

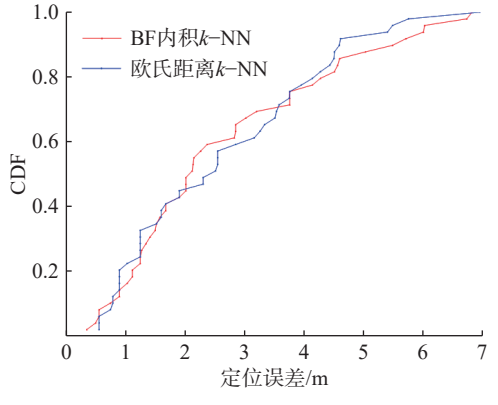

图 9 内积与欧氏距离定位误差

Fig. 9 Positioning errors using inner product vs. Euclidean distance

图 10 比较了本文方案、文献 [6] 和文献 [12] 3 种方案定位误差的 CDF。从图中可以看出，本文方案与文献 [6] 方案的定位精度相近，两者的 CDF 曲线在大部分误差范围内几乎重合。而由于文献 [12] 在指纹匹配时采用 Sφrensen 距离，其定位精度略高于欧氏距离。总体来看，本文方案与其他两种方案的定位精度差异不大，接近 60% 的定位误差均小于 3 m。

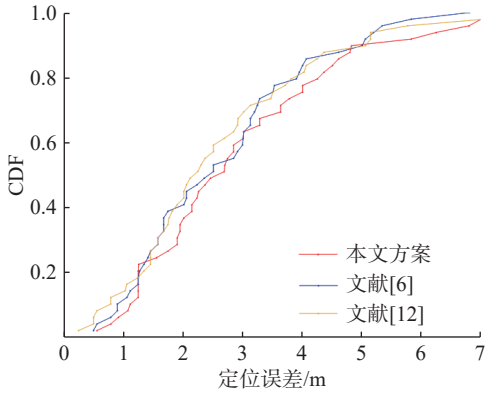

图 10 3 种方案定位误差对比

Fig. 10 Positioning error comparison of three schemes

指纹维度，即参与定位的 AP 数量同样是影响定位精度的关键因素之一，通常 AP 数量越多，指纹的位置区分度越明显。图 11 对比了本文方案、文献 [6] 与文献 [12] 方案在不同 AP 数量下的平均定位误差。可以看出，当 AP 数量变化时，各个定位算法都呈现较明显的定位精度下降，但本文采用的对指纹特征进行局部哈希映射的方式在实现指纹数据隐私保护的同时，并没有引入额外的定位精度损失。

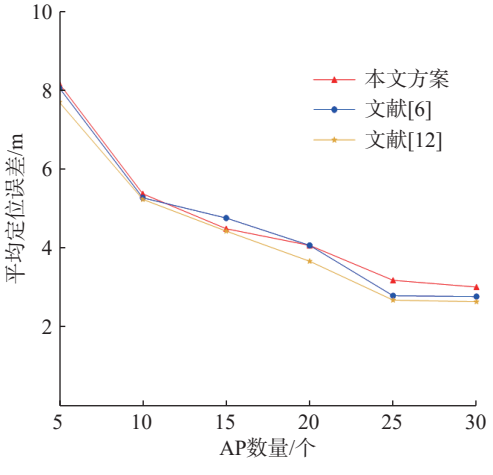

图 11 不同 AP 数量对定位误差的影响

Fig. 11 Effect of the number of access points on positioning errors

图 12 呈现了本文方案、文献 [6] 与文献 [12] 方案 20% 的有效 RSS 值发生±15 dBm 的瞬时突变时的定位误差 CDF 特性。从实验结果分析可知，

RSS 信号强度的波动变化使得指纹信号匹配时的错误最近邻更容易发生，从而引起相应的定位精度的变化，这符合指纹定位的特性。本文所提的方案采用布隆过滤器后，内积匹配仅关注二进制位分布的一致性，不受信号具体幅度波动的直接影响。由此本文方案对各类动态信号干扰均具有较强的鲁棒性。

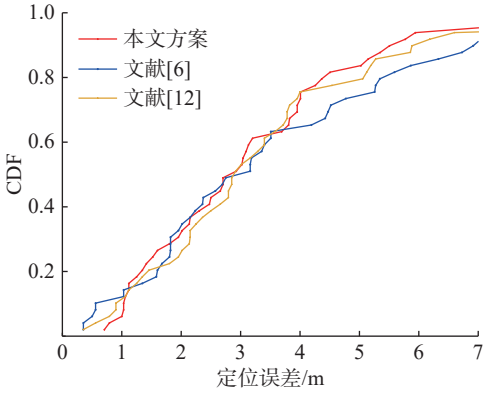

图 12　RSS 波动对定位误差的影响

Fig. 12　Effect of RSS fluctuations on positioning errors

5　结　论

提出了一种云环境下轻量级的室内 Wi-Fi 指纹定位隐私保护方案，有效解决了外包云数据的隐私安全与定位实时性问题。所提方案利用局部敏感哈希为离线指纹构造了保持信号欧氏距离关系的指纹特征布隆过滤器，将指纹定位匹配时的距离计算转化为二进制向量的内积计算，在不泄露指纹明文信息的情况下由云端完成定位；为进一步降低在地图大数据中的计算开销，构造了基于 MinHashLSH 的离线指纹哈希桶，可实现对用户的局部定位。理论分析和实验结果表明，本文所提方案在将定位服务外包时有效保护了用户和 LSP 资源信息的隐私，同时在保证定位精度的前提下以较低的计算开销和通信开销完成了用户的位置估计。此外，方案对室内动态信号干扰具有较强鲁棒性，“局部–精确”的定位设计使其能高效适配 Wi-Fi 覆盖密集的大型室内空间及大规模指纹库场景。但方案也存在一定局限性，如在 AP 部署稀疏的室内环境中，由于指纹向量的位置区分度不足，会导致定位精度下降。

参考文献：

[1] Zhu Y P, Qiu Y, Wang J Y, et al. Protecting position privacy in range-based crowdsourcing cooperative localization[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(1): 1136−1150.

[2] Zhuang Y, Zhang C Y, Huai J Z, et al. Bluetooth localization technology: principles, applications, and future trends[J]. IEEE Internet of Things Journal, 2022, 9(23): 23506−23524.

[3] Torres-Sospedra J, Montoliu R, Trilles S, et al. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems[J]. Expert Systems with Applications, 2015, 42(23): 9263−9278.

[4] Bellovin S M, Hutchins R M, Jebara T, et al. When enough is enough: location tracking, mosaic theory, and machine learning[J]. New York University Journal of Law & Liberty, 2014, 8: 555−628.

[5] Li H, Sun L M, Zhu H J, et al. Achieving privacy preservation in WiFi fingerprint-based localization[C]// Proceedings of the 2014 IEEE International Conference on Computer Communications. Toronto: IEEE, 2014: 2337–2345.

[6] Wang X T, Qiu Y, Hu J M, et al. Secure two-party computation for fingerprinting-based indoor localization [C]//Proceedings of the 2023 IEEE/CIC International Conference on Communications in China. Dalian: IEEE, 2023: 1–5.

[7] Wang Y F, Huang M J, Jin Q, et al. DP3: a differential privacy-based privacy-preserving indoor localization mechanism[J]. IEEE Communications Letters, 2018, 22(12): 2547−2550.

[8] Zhao P, Jiang H B, Lui J C S, et al. P^3-LOC: a privacy-preserving paradigm-driven framework for indoor localization[J]. IEEE/ACM Transactions on Networking, 2018, 26(6): 2856−2869.

[9] 乐燕芬, 雷若兰, 姚恒. 安全高效的多用户室内指纹定位方案 [J]. 西安电子科技大学学报, 2025, 52(3): 123−133.

[10] Järvinen K, Leppäkoski H, Lohan E S, et al. PILOT: practical privacy-preserving indoor localization using OuTsourcing[C]//Proceedings of the 2019 IEEE European Symposium on Security and Privacy. Stockholm: IEEE, 2019: 448–463.

[11] Eshun S N, Palmieri P. A cryptographic protocol for efficient mutual location privacy through outsourcing in indoor Wi-Fi localization[J]. IEEE Transactions on

Information Forensics and Security, 2024, 19: 4086−4099.

[12] 张应辉, 张思睿, 赵秋霞, 等. 基于 Wi-Fi 指纹且计算外包的室内定位隐私保护方案 [J]. 通信学报, 2024, 45(2): 31−39.

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

[14] Datar M, Immorlica N, Indyk P, et al. Locality-sensitive hashing scheme based on p-stable distributions[C]// Proceedings of the Twentieth Annual Symposium on Computational Geometry. Brooklyn: ACM, 2004: 253–262.

[15] Broder A Z, Charikar M, Frieze A M, et al. Min-wise independent permutations[C]//Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC). Dallas: ACM, 1998: 327–336.

[16] An J, Wang Z X, He X, et al. Know where you are: a practical privacy-preserving semi-supervised indoor positioning via edge-crowdsensing[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4875−4887.

[17] Damgård I, Geisler M, Krøigaard M. Efficient and secure comparison for on-line auctions[C]//Proceedings of the 12th Australasian Conference on Information Security and Privacy. Townsville: Springer, 2007: 416–430.

[18] Tóth Z, Tamás J. Miskolc IIS hybrid IPS: dataset for hybrid indoor positioning[C]//Proceedings of the 26th International Conference Radioelektronika (RADIOELEKTRONIKA). Kosice: IEEE, 2016: 408–412.

（编辑：董 伟）