隐私保护的轨迹相似度计算方法

于海宁[1]，张宏莉[1]，余翔湛[1]，曲家兴[2]，葛蒙蒙[1,3]
（1. 哈尔滨工业大学网络空间安全学院，黑龙江 哈尔滨 150001；2. 黑龙江省网络空间研究中心，黑龙江 哈尔滨 150001；
3. 南洋理工大学计算机科学与工程学院，新加坡 639798）

摘 要：为解决轨迹外包服务中轨迹相似度计算的隐私泄露问题，提出了一种隐私保护的轨迹相似度计算（pTSC）方法，在该方法中轨迹服务存储来自轨迹拥有者的加密轨迹，接收来自轨迹查询者的加密兴趣轨迹，并支持基于加密的兴趣轨迹和存储轨迹的相似度安全计算，进而避免拥有者的存储轨迹和查询者的兴趣轨迹泄露。为高效地计算密态轨迹的相似度，提出了一个基于最长公共子序列的轨迹相似度安全计算协议，该协议利用类同态加密算法和安全比较协议实现了密态轨迹的最长公共子序列的高效计算。此外，设计了一种密文压缩算法，进一步提升效率。理论分析和实验评估证明了 pTSC 方法的安全性和高效性。

关键词：隐私保护；轨迹相似度；同态加密；安全计算
中图分类号：TP39
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022223

Privacy-preserving trajectory similarity computation method

YU Haining[1], ZHANG Hongli[1], YU Xiangzhan[1], QU Jiaxing[2], GE Mengmeng[1,3]
1. School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China
2. Heilongjiang Province Cyberspace Research Center, Harbin 150001, China
3. School of Computer Science and Engineering, Nangyang Technological University, Singapore 639798

Abstract: To tackle privacy concerns on user information leakage in trajectory outsourcing services, a privacy-preserving trajectory similarity computation (pTSC) method was proposed. A trajectory outsourcing service provider was enabled to store encrypted trajectories from owners, wait for encrypted interested trajectories from requesters, and compute trajectory similarity between an interested trajectory and stored trajectories in ciphertext domain without learning anything about users' trajectories. To compute a trajectory similarity over encrypted trajectories efficiently, a secure trajectory similarity computation protocol with longest common subsequence was proposed, which used somewhat homomorphic encryption and secure comparison protocol to compute the length of longest common subsequence over two encrypted trajectories. Furthermore, a ciphertext compression algorithm was designed to improve efficiency. Theoretical analysis and experimental evaluations show that pTSC method is secure and efficient.

Keywords: privacy-preserving, trajectory similarity, homomorphic encryption, secure computing

0 引言

随着无线通信、普适计算、卫星导航技术的不断发展，带有 GPS 定位功能的智能设备被应用在许多领域[1]。这些设备能够记录相关实体的运动轨迹，进而形成海量的轨迹数据。这些轨迹数据不但记录了个体对象的运动模式、行为特征与规律，例如，经常活动的路线、生活工作的地点以及兴趣爱好和

收稿日期：2022−04−13；**修回日期：**2022−07−12
基金项目：国家自然科学基金资助项目（No.62172123, No.61732022）；黑龙江省自然科学基金资助项目（No.YQ2021F007）；中央引导地方科技发展专项资金资助项目（No.ZY20B11）
Foundation Items: The National Natural Science Foundation of China (No.62172123, No.61732022), Heilongjiang Provincial Natural Science Foundation of China (No.YQ2021F007), The Special Projects for the Central Government to Guide the Development of Local Science and Technology (No.ZY20B11)

健康状况等，而且蕴含了群体对象的泛在移动模式与规律，例如，社会群体活动特征、城市交通拥堵规律、路网拓扑与地点坐标等。针对海量的轨迹数据，人们通过轨迹分析、挖掘等技术手段进行知识发现，并将其运用在各种交通和位置服务应用中，包括交通导航、位置服务推荐、交通指挥、物流配送、车辆监控等。轨迹相似度计算是轨迹分析的基础操作之一，其主要分析不同轨迹之间的位置相似性。提取相似轨迹在出行路径预测、兴趣区域发现、轨迹聚类、个性化路径推荐等领域具有广泛的应用。

云计算外包服务[2]的普及使很多轨迹拥有者选择将轨迹上传到云端轨迹服务存储，以降低轨迹存储和计算成本。同时，轨迹查询者可以向轨迹服务发送关于其兴趣轨迹的相似度计算请求，轨迹服务计算该兴趣轨迹与存储轨迹的相似度，并将最相似轨迹返回给查询者。用户在享受轨迹服务的同时，也面临着严重的隐私泄露风险[3]。轨迹数据中蕴含了大量关于用户的私密信息[4]，例如用户住址、经济和健康状况等。轨迹服务往往会收集轨迹拥有者的上传轨迹和轨迹查询者的兴趣轨迹，并从中挖掘用户画像。

针对上述隐私泄露问题，有研究提出了面向加密轨迹的轨迹相似度安全计算方法，其主要利用同态加密[5]、姚氏混淆电路[6]、安全求交集[7]等密码学工具计算轨迹相似度，从而避免将轨迹泄露给轨迹服务[8-10]。例如，Liu 等[11]利用同态加密和姚氏混淆电路实现了轨迹相似度的安全计算框架，但该框架计算效率有待提升，例如，其计算 2 条长度为 100 的轨迹之间的相似度大约需要 13 min。

基于加密轨迹计算轨迹相似度是一个较复杂的问题，如何有效降低计算和通信开销以适应大规模、长轨迹的相似度计算是亟待解决的问题。本文聚焦大规模长轨迹的相似度安全计算，基于同态加密设计了隐私保护的轨迹相似度计算（pTSC, privacy-preserving trajectory similarity computation）方法，该方法能够在保护用户轨迹隐私的前提下，更高效地计算长轨迹之间的相似度。

本文主要的研究工作如下。

1) 提出了 pTSC 方法，在该方法中轨迹服务存储来自轨迹拥有者的加密上传轨迹，接收来自轨迹查询者的加密兴趣轨迹，并支持基于密态的兴趣轨迹和存储轨迹的相似度安全计算，从而避免拥有者的轨迹和查询者的查询意图泄露。

2) 提出了一个最长公共子序列安全计算（SLCSS）协议，该协议利用类同态加密算法和安全比较协议实现了基于密态轨迹的最长公共子序列计算。此外，该协议还设计了一种密文压缩（简称 Compress）算法，用以降低通信开销。

3) 实现了 pTSC 原型系统，并基于真实的轨迹数据集开展了性能开销的仿真实验，实验结果表明，该方法具有良好的计算和通信性能，且优于现有的计算方法。

1 预备知识

1.1 轨迹相似度计算

定义 1 轨迹。轨迹 tr 是一个 GPS 点的序列，$\mathrm{tr}=(p[0],p[1],\cdots,p[m-1])$，其中，$p[i]=(x[i],y[j])$，$x[i]$ 和 $y[i]$ 分别为经纬度坐标。

最长公共子序列（LCSS, longest common subsequence）[12]用于衡量 2 条轨迹的相似程度，其对轨迹噪声具有较强的容忍度。给定 2 条轨迹 $\mathrm{tr}_1=(p_1[0],\cdots,p_1[n-1])$ 和 $\mathrm{tr}_2=(p_2[0],\cdots,p_2[m-1])$，$\mathrm{Head}(\mathrm{tr}_1)$ 和 $\mathrm{Head}(\mathrm{tr}_2)$ 分别表示 $p_1[0]$ 和 $p_2[0]$，$\mathrm{Rest}(\mathrm{tr}_1)$ 和 $\mathrm{Rest}(\mathrm{tr}_2)$ 分别表示 $(p_1[1],\cdots,p_1[n-1])$ 和 $(p_2[1],\cdots,p_2[n-1])$，那么，tr_1 和 tr_2 之间的最长公共子序列表示为

$$\mathrm{LCSS}(\mathrm{tr}_1,\mathrm{tr}_2)=\begin{cases}0，\ n=0\ 或\ m=0\\ \mathrm{LCSS}(\mathrm{Rest}(\mathrm{tr}_1),\mathrm{Rest}(\mathrm{tr}_2))+1，\\ \quad 0\leqslant \mathrm{dist}(\mathrm{Head}(\mathrm{tr}_1),\mathrm{Head}(\mathrm{tr}_2))<\epsilon\\ \max\{\mathrm{LCSS}(\mathrm{tr}_1,\mathrm{Rest}(\mathrm{tr}_2)),\\ \quad \mathrm{LCSS}(\mathrm{Rest}(\mathrm{tr}_1),\mathrm{tr}_2)\}\ ，其他\end{cases}$$

其中，阈值 ϵ 用于判断 GPS 点是否邻近。LCSS 的计算复杂度为 $\mathcal{O}(nm)$。轨迹 tr_1 和 tr_2 的之间的 LCSS 相似度为

$$\mathrm{sim}_{\mathrm{LCSS}}(\mathrm{tr}_1,\mathrm{tr}_2)=1-\frac{\mathrm{LCSS}(\mathrm{tr}_1,\mathrm{tr}_2)}{\min\{n,m\}}$$

1.2 同态加密算法

类同态加密（SHE, somewhat homomorphic encryption）支持无限次密文加法以及有限次密文乘法。CKKS（Cheon-Kim-Kim-Song）同态加密算法[13]是基于环上错误学习（RLWE, ring learning with error）[14]的 SHE 方案，具有语义安全性。CKKS 同态加密算法概述如下。

1) 密钥生成：KeyGen(n,q)。输入安全参数正整数 n 和模 q，从多项式环 $R_q=\dfrac{\mathbb{Z}_q[X]}{X^n+1}$ 中选择随机多项式 a、私密多项式 s 和噪声多项式 e，输出私钥 sk=s、公钥 $\mathrm{pk}=(-as+e,a)$。

2) 加密：Enc(p,pk)。输入明文 $p\in R_q$、公钥 $\mathrm{pk}=(b,a)$，输出密文 $\hat{c}=(p+b,a)$。

3) 解密：Dec$(\hat{c},\mathrm{sk})$。输入密文 $\hat{c}=(c_0,c_1)\in R_q^2$、私钥 $\mathrm{sk}=s$，输出明文 $p=c_0+c_1s$。

4) 同态加法：Add$(\hat{c},\hat{c}')$。输入密文 $\hat{c}=(c_0,c_1)$ 和 $\hat{c}'=(c_0',c_1')$，输出密文 (c_0+c_0',c_1+c_1')。加法同态性表示为

$$\mathrm{Dec}(\mathrm{Add}(\hat{c},\hat{c}'),\mathrm{sk})\equiv\mathrm{Dec}(\hat{c},\mathrm{sk})+\mathrm{Dec}(\hat{c}',\mathrm{sk})$$

5) 同态乘法：Mult$(\hat{c},\hat{c}')$。输入密文 $\hat{c}=(c_0,c_1)$ 和 $\hat{c}'=(c_0',c_1')$，输出密文 $(c_0c_0',c_0c_1'+c_0'c_1,c_1c_1')$，乘法同态性表示为

$$\mathrm{Dec}(\mathrm{Mult}(\hat{c},\hat{c}'),\mathrm{sk})\equiv\mathrm{Dec}(\hat{c},\mathrm{sk})\,\mathrm{Dec}(\hat{c}',\mathrm{sk})$$

本文使用⊞、⊟和⊠分别表示密文的同态加法、减法和乘法操作，即 $\hat{c}\boxplus\hat{c}'=\mathrm{Add}(\hat{c},\hat{c}')$、$\hat{c}\boxminus\hat{c}'=\mathrm{Add}(\hat{c},-\hat{c}')$ 和 $\hat{c}\boxtimes\hat{c}'=\mathrm{Mult}(\hat{c},\hat{c}')$。明文与密文之间的同态运算也可同样表示，例如明文 p 和密文 $\hat{c}$ 之间的同态加法可表示为 $p\boxplus\hat{c}$。

CKKS 同态加密算法利用 SIMD 技术[15]可将多个明文编码打包，并加密成一个密文。给定向量 $\boldsymbol{a}=(a_0,a_1,\cdots,a_{n-1})$ 和 $\boldsymbol{b}=(b_0,b_1,\cdots,b_{n-1})$，SIMD 编码操作可以表示为

$$\langle a_0+b_0\,|\cdots|\,a_{n-1}+b_{n-1}\rangle=$$
$$\mathrm{Dec}(\mathrm{Enc}(\langle a_0\,|\cdots|\,a_{n-1}\rangle,\mathrm{pk})\boxplus\mathrm{Enc}(\langle b_0\,|\cdots|\,b_{n-1}\rangle,\mathrm{pk}),\mathrm{sk})$$
$$\langle a_0b_0\,|\cdots|\,a_{n-1}b_{n-1}\rangle=$$
$$\mathrm{Dec}(\mathrm{Enc}(\langle a_0\,|\cdots|\,a_{n-1}\rangle,\mathrm{pk})\boxtimes\mathrm{Enc}(\langle b_0\,|\cdots|\,b_{n-1}\rangle,\mathrm{pk}),\mathrm{sk})$$

如上所述，n 个明文可以打包到一个密文中，且对打包后的密文进行一次同态运算可以完成这 n 对明文的运算。

CKKS 同态加密算法还支持同态循环旋转操作。假设一项密文对应的明文多项式系数为 $m_0,\cdots,m_{n-1}$，那么，可以对密文做向右的 k 步同态循环旋转操作来改变系数在明文多项式中的位置，获得 $m_{\pi_k(0)},\cdots,m_{\pi_k(n-1)}$，其中，$0\leqslant k<n$，$\pi_k(i)=k+i\bmod n$。支持 SIMD 向右的 k 步同态循环旋转操作可表示为

$$\langle a_{\pi_k(0)}\,|\cdots|\,a_{\pi_k(i+n-1)}\rangle=$$
$$\mathrm{Dec}(\mathrm{Rot}_{\pi_k}(\mathrm{Enc}(\langle a_0\,|\cdots|\,a_{n-1}\rangle,\mathrm{pk})),\mathrm{sk})$$

其中，$\mathrm{Rot}_{\pi_k}(\cdot)$ 表示同态循环旋转操作。

1.3　安全比较协议

安全比较协议在不泄露双方私密输入值的前提下获得 2 个私密输入值的大小关系。假设双方各自持有 l bit 的私密输入值，分别为 x 和 y。它们可以执行如下的安全比较协议[16]，以实现 x 和 y 的安全比较。

1) 双方分别将各自的私密输入值转换为二进制表示：$x[l-1]x[l-2]\cdots x[0]$和 $y[l-1]y[l-2]\cdots y[0]$。

2) 当 $i=l-1,\ l-2,\cdots,\ 0$ 时，双方计算 $a[i]$和 $b[i]$如下。

如果 $i=l-1$，则 $a[i]=x[i](1-y[i]),b[i]=y[i](1-x[i])$。

如果 $i<l-1$，则 $a[i]=(1-b[i+1])(a[i+1]+(1-a[i+1])\cdot x[i](1-y[i])),b[i]=(1-a[i+1])(b[i+1]+(1-b[i+1])y[i](1-x[i]))$。

3) 最终比较结果判断如下。

如果 $a[0]=1$，则 $x>y$。

如果 $b[0]=1$，则 $x<y$。

如果 $a[0]=b[0]=0$，则 $x=y$。

2　模型与问题定义

2.1　系统模型

系统模型如图 1 所示，pTSC 主要涉及如下对象。

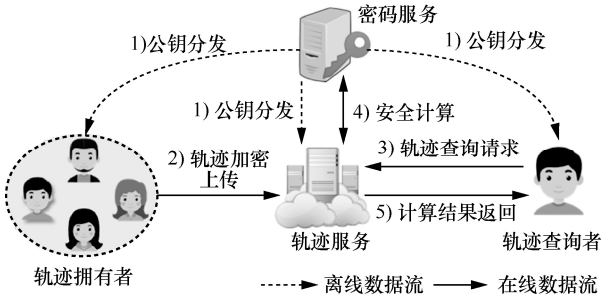


图 1　系统模型

轨迹服务（TS, trajectory service）：管理轨迹数据库，对外提供轨迹外包存储服务，并支持针对存储轨迹的相似度计算服务。

密码服务（CS, crypto service）：对外提供密钥分发和管理服务，并参与轨迹相似度的安全计算。

轨迹拥有者：使用轨迹服务提供的存储服务，以记录其所拥有的历史轨迹数据。

轨迹查询者：使用轨迹服务提供的轨迹相似度计算服务，查询与其兴趣轨迹最近似的轨迹。

上述对象的交互流程概括如下。

1) 密码服务生成一对公私钥，公钥被分发给其他对象，而私钥被密码服务保留。此步骤可以离线执行。

2) 轨迹拥有者加密其轨迹，上传至轨迹服务存储。

3) 轨迹查询者加密其兴趣轨迹，向轨迹服务发起查询请求，获得与兴趣轨迹最近似的存储轨迹。

4) 轨迹服务联合密码服务基于加密轨迹计算相似度，检索出与兴趣轨迹最近似的存储轨迹。

5) 轨迹服务将最近似的存储轨迹及其相似度返回给轨迹查询者。

2.2 威胁模型

pTSC 的威胁模型描述如下。

轨迹服务和密码服务均是半诚实的，即双方将严格地执行协议，但是计算过程中它们会尽可能地根据中间信息和计算结果推测出更多的额外信息。针对半诚实模型的安全协议不但能够实现高效的计算，而且对恶意模型下的安全协议研究具有重要参考价值。

轨迹拥有者和轨迹查询者均是半诚实的，前者会上传其真实的加密轨迹信息，后者会提交其真实的加密查询请求。

轨迹服务、密码服务、轨迹拥有者和轨迹查询者中任意两方不存在共谋关系。此假设被广泛认可，因为轨迹服务和密码服务提供商往往是信誉优质的大型企业，共谋行为会极大地损害声誉。

底层通信网络是安全的，不存在外部敌手窃听或篡改通信内容。

pTSC 面临如下隐私威胁。

1) 轨迹跟踪攻击。轨迹服务或密码服务获取用户的轨迹数据，实施针对目标用户的在线或离线跟踪。

2) 大规模轨迹推理攻击。轨迹服务或密码服务收集大量用户轨迹数据，并进一步分析挖掘额外的用户隐私信息，如家庭住址、经济或健康状况等。

2.3 问题定义

本文关注的问题定义如下。

给定加密的存储轨迹集合 $\mathbf{TR}=\{\hat{\mathrm{tr}}_0,\cdots,\hat{\mathrm{tr}}_{N-1}\}$，以及待查询的加密兴趣轨迹 $\hat{\mathrm{tr}}$，拟从 **TR** 中检索出与兴趣轨迹 $\hat{\mathrm{tr}}$ 最近似的存储轨迹 $\hat{\mathrm{tr}}^*$，表示为

$$\hat{\mathrm{tr}}^*=\underset{0\leqslant k<N}{\arg\min}\{\mathrm{sim}_{\mathrm{LCSS}}(\mathrm{tr},\mathrm{tr}_k)\}$$

本文方法的设计目标如下。

高性能。pTSC 方法应该具有低的计算和通信开销，在服务端支持高效的密态长轨迹相似度计算，在客户端支持资源受限设备运行。

隐私保护。轨迹拥有者的存储轨迹以及轨迹查询者的兴趣轨迹不会被泄露给轨迹服务和密码服务。

3 pTSC 方法设计

本文提出的 pTSC 方法可表示为 pTSC=(Init, Upload, Query, SimComp)，如图 2 所示，其中，Init 表示系统初始化，轨迹服务初始化系统参数，密码服务生成一对公钥和私钥(pk, sk)，并公开 pk，保留 sk；Upload 表示轨迹上传存储；Query 表示轨迹查询请求提交；SimComp 表示轨迹相似度安全计算。

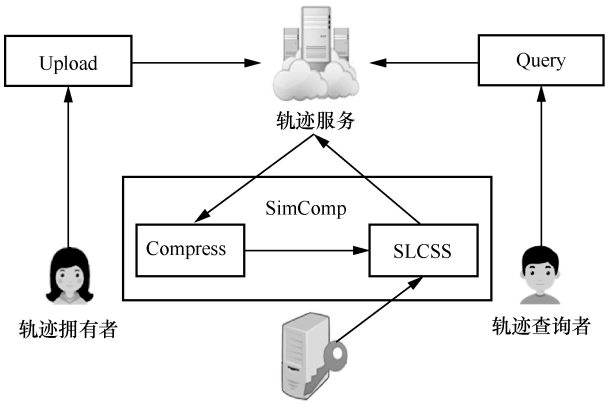


图 2 pTSC 方法

3.1 轨迹上传存储

轨迹拥有者加密自身轨迹，并上传到轨迹服务存储。假设轨迹拥有者 owner 持有轨迹集合 $\{\mathrm{tr}_k=(p_k[0],p_k[1],\cdots,p_k[m_k-1])\}_{0\leqslant k<N}$，其中 $p_k=(x_k[j],y_k[j])$。针对集合中任一存储轨迹 tr_k，拥有者构造明文多项式，并使用 CKKS 同态加密算法加密该多项式，获得加密的存储轨迹 $\hat{\mathrm{tr}}_k=(\hat{p}_k^x,\hat{p}_k^y)$，其中

$$\hat{p}_k^x=\mathrm{Enc}(p_k^x,\mathrm{pk})=\mathrm{Enc}\left(\left\langle p_k^x[0]|\cdots|p_k^x[m_k-1]\right\rangle,\mathrm{pk}\right)$$

$$\hat{p}_k^y=\mathrm{Enc}(p_k^y,\mathrm{pk})=\mathrm{Enc}\left(\left\langle p_k^y[0]|\cdots|p_k^y[m_k-1]\right\rangle,\mathrm{pk}\right)$$

此外，轨迹拥有者构造如下密文用于标识轨迹 tr_k 的长度

$$\hat{l}_k=\mathrm{Enc}(\langle\underbrace{-1|\cdots|-1}_{m_k}|\underbrace{1|\cdots|1}_{n-m_k}\rangle,\mathrm{pk})$$

加密轨迹集合 $\{(\widehat{\mathrm{tr}}_k,\hat{l}_k)\}_{0\leqslant k<N}$ 被发送给轨迹服务进行存储。

3.2　轨迹查询请求提交

轨迹查询者加密查询请求，并提交到轨迹服务查询与其最近似的轨迹。假设轨迹查询者持有兴趣轨迹 tr=(p[0], p[1], ⋯, p [m–1])，其向轨迹服务发起查询请求，以获得与 tr 最相似的轨迹。为此，查询者构造明文多项式，并使用 CKKS 同态加密算法加密该多项式，进而获得加密的兴趣轨迹 $\widehat{\mathrm{tr}}=(\hat{p}^x,\hat{p}^y)$，其中

$$\hat{p}^x=\mathrm{Enc}(p^x,\mathrm{pk})=\mathrm{Enc}\left(\left\langle p^x[0]|\cdots|p^x[\alpha-1]\right\rangle,\mathrm{pk}\right)$$

$$\hat{p}^y=\mathrm{Enc}(p^y,\mathrm{pk})=\mathrm{Enc}\left(\left\langle p^y[0]|\cdots|p^y[\alpha-1]\right\rangle,\mathrm{pk}\right)$$

加密的兴趣轨迹及其长度 $(\widehat{\mathrm{tr}},\alpha)$ 作为查询请求被提交给轨迹服务。

3.3　轨迹相似度安全计算

轨迹服务联合密码服务基于密态轨迹计算相似度，并将检索结果返回给轨迹查询者。假设轨迹服务存储了 M 个拥有者的轨迹，拥有者 owner_i 所属的轨迹集合表示为 $\{(\widehat{\mathrm{tr}}_k,\hat{l}_k)\}_{0\leqslant k<N_i}$，那么，轨迹服务存储的密态轨迹集合可表示为

$$\mathbf{TR}=\left\{\left(\mathrm{owner}_i,\{(\widehat{\mathrm{tr}}_k,\hat{l}_k)\}_{0\leqslant k<N_i}\right)\right\}_{0\leqslant i<M}$$

轨迹查询者能够指定轨迹拥有者的范围，针对目标拥有者的轨迹集合发起查询请求。轨迹服务计算集合中每条密态存储轨迹与兴趣轨迹的相似度，并选择出最近似的存储轨迹作为查询结果。

给定密态查询请求 $(\widehat{\mathrm{tr}},\alpha)$ 和密态存储轨迹 $(\widehat{\mathrm{tr}}_k,\hat{l}_k)$，轨迹服务能够安全地计算二者之间的 LCSS 轨迹相似度，如算法 1 所示。

算法 1　轨迹相似度安全计算

输入　轨迹服务 TS 输入密态查询请求 $(\widehat{\mathrm{tr}},\alpha)$ 以及一条密态存储轨迹 $(\widehat{\mathrm{tr}}_k,\hat{l}_k)$，CS 输入私钥 sk

输出　轨迹 tr 与 tr_k 之间的轨迹相似度 $\mathrm{sim}_{\mathrm{LCSS}}(\mathrm{tr},\mathrm{tr}_k)$

TS

1)　$\hat{\boldsymbol{D}}\leftarrow\varnothing,\hat{\boldsymbol{D}}'\leftarrow\varnothing,\boldsymbol{R}\leftarrow\varnothing$;

2)　$l_{\mathrm{tr}}\leftarrow\langle\underbrace{-1|\cdots|-1}_{\alpha}|\underbrace{1|\cdots|1}_{n-\alpha}\rangle$;

3) for　$i=0,1,\cdots,n-1$　do

4)　　$\widehat{\mathrm{dist}}_i\leftarrow(\mathrm{Rot}_{\pi_i}(\hat{p}^x)\boxminus\hat{p}_k^x)^2\boxplus(\mathrm{Rot}_{\pi_i}(\hat{p}^y)\boxminus\hat{p}_k^y)^2$;

5)　　$\widehat{\mathrm{dist}}_i\leftarrow\widehat{\mathrm{dist}}_i\boxtimes\hat{l}_k\boxtimes\mathrm{Rot}_{\pi_i}(l_{tr})\boxminus\frac{1}{2}\boxtimes(\hat{l}_k\boxplus\langle 1|\cdots|1\rangle)$;

6)　　$\hat{\boldsymbol{D}}\leftarrow\bigcup\widehat{\mathrm{dist}}_i$;

7) end for

8)　$\hat{\boldsymbol{D}}_c\leftarrow\mathrm{Compress}(\hat{\boldsymbol{D}})$;

9) for　$\widehat{\mathrm{dist}}_i\in\hat{\boldsymbol{D}}_c$　do

10)　　生成一组随机数 $\boldsymbol{r}_i=\{r_i[0],\cdots,r_i[n-1]\}$;

11)　　$\hat{\boldsymbol{D}}_c'\leftarrow\bigcup\widehat{\mathrm{dist}}_i\boxplus\langle r_i[0]|\cdots|r_i[n-1]\rangle$;

12)　　$\boldsymbol{R}\leftarrow\bigcup\boldsymbol{r}_i$;

13) end for

14)　发送 $\hat{\boldsymbol{D}}_{c'}$ 至 CS;

CS

15)　$\boldsymbol{D}_{c'}\leftarrow\mathrm{Dec}(\hat{\boldsymbol{D}}_c',\mathrm{sk})$;

TS 和 CS

16)　$\mathrm{LCSS}(\mathrm{tr},\mathrm{tr}_k)\leftarrow\mathrm{SLCSS}(\boldsymbol{D}_c',\mathrm{sk};\boldsymbol{R})$;

TS

17)　$\mathrm{sim}_{\mathrm{LCSS}}(\mathrm{tr},\mathrm{tr}_k)\leftarrow1-\dfrac{\mathrm{LCSS}(\mathrm{tr},\mathrm{tr}_k)}{\alpha}$;

算法 1 具体步骤介绍如下。

1) 轨迹服务 TS 计算 2 条加密轨迹 $\widehat{\mathrm{tr}}=(\hat{p}[0],\cdots,\hat{p}[\alpha-1])$ 和 $\widehat{\mathrm{tr}}_k=(\hat{p}_k[0],\cdots,\hat{p}_k[m_k-1])$ 中任意一对 GPS 点之间的欧氏距离平方值（步骤 3)～步骤 7)），共需计算 αn 个欧氏距离平方值。轨迹服务利用 SIMD 同态循环旋转操作分别连续向右旋转密态轨迹 $\widehat{\mathrm{tr}}$ 的 GPS 点的经纬度坐标序列 $\hat{p}^x$ 和 $\hat{p}^y$（步骤 4)）。每次旋转，轨迹服务至多可以计算 α 个欧氏距离平方值。为解密后区分明文多项式系数中的有效距离值和无效值，轨迹服务利用 CKKS 的同态运算将无效系数置为负值（步骤 5)），而有效距离值为正值。2 条轨迹中 GPS 点之间欧氏距离计算示例如图 3 所示，其中，$\alpha=3$, m=6, n=8。图 3 中，轨迹服务对密态轨迹 $\widehat{\mathrm{tr}}$ 执行了 7 次同态向右循环旋转一步操作和 8 次欧氏距离平方值计算，并获得了 8 个距离密文，其中包括 18 个无效系数。针对每个距离密文，轨迹服务执行 2 次同态乘法和一次同态加法将无效系数置为负值。

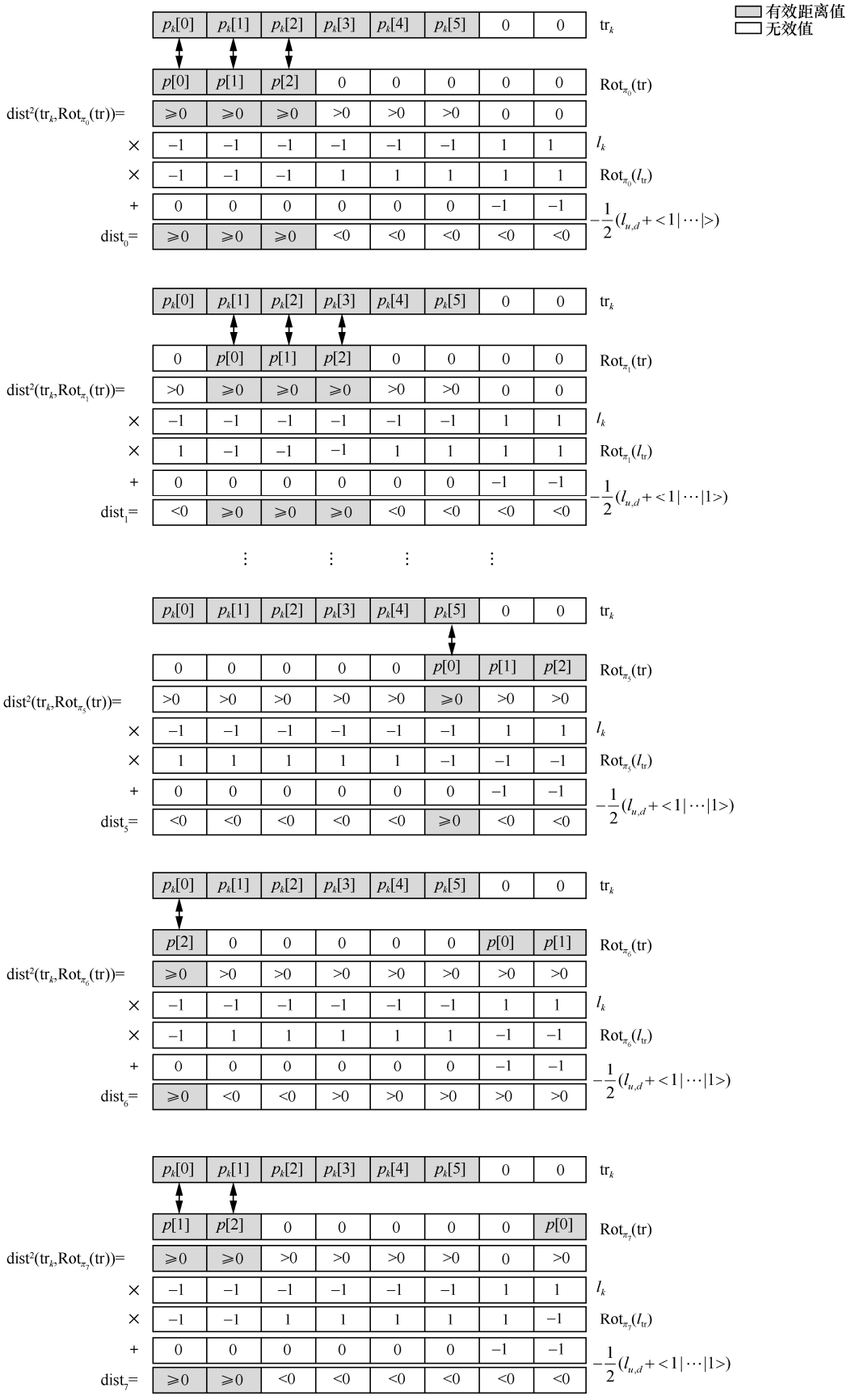


图 3　2 条轨迹中 GPS 点之间欧氏距离计算示例

2) 当轨迹较短时（即$\left\lfloor \frac{n}{\alpha} \right\rfloor > 1$），轨迹服务计算出的多个距离密文可以被压缩，从而有效降低通信开销，密文压缩示例如图 4 所示。密文压缩算法如算法 2 所示，轨迹服务将待压缩距离密文中的无效系数置为 0（步骤 6)～步骤 7)、步骤 10)～步骤 11)、步骤 14)～步骤 15)），然后，利用同态循环旋转操作消除有效系数的位置重叠（步骤 8)、步骤 12)、步骤 16)）。最后，利用同态加法操作将$\left\lfloor \frac{n}{\alpha} \right\rfloor$个距离密文压缩为一个密文（步骤 17)）。上述密文压缩算法的压缩率为$1-\left(\frac{1}{\left\lfloor \frac{n}{\alpha} \right\rfloor}\right)$。

3) 轨迹服务基于压缩后的距离密文，联合密码服务执行安全计算协议，进而计算出轨迹 tr 与 tr_k 的最长公共子序列 $\mathrm{LCSS}(\mathrm{tr},\mathrm{tr}_k)$，同时避免轨迹隐私泄露给轨迹服务或密码服务。具体地，针对每一个压缩距离密文，轨迹服务首先选择 n 个 $\ell-1$ bit 的随机整数（算法 1 步骤 10)），然后利用同态加法分别将这些随机数加到压缩密文对应的明文多项式系数上，以达到保护有效距离值的目的（步骤 11)）。轨迹服务将添加过随机数的距离密文集合 $\hat{\boldsymbol{D}}_c'$ 发送给密码服务。密码服务解密 $\hat{\boldsymbol{D}}_c'$，并解码明文多项式获得一个添加了随机数据的距离平方值的集合 $\boldsymbol{D}_c'$。

4) 轨迹服务持有随机数集合 $\boldsymbol{R}=\{r_0,\cdots,r_{\beta-1}\}$，$\boldsymbol{\epsilon}=\{r_0+\epsilon^2,\cdots,r_{\beta-1}+\epsilon^2\}$，其中 $\beta=n\left\lfloor \frac{n}{\left\lfloor \frac{n}{\alpha} \right\rfloor} \right\rfloor$。密码服务持有添加了随机数据的距离平方值的集合 $\boldsymbol{D}_c''=\{\mathrm{dist}_0'',\cdots,\mathrm{dist}_{\beta-1}''\}$。基于各自持有的集合，轨迹服务和密码服务执行 SLCSS 协议，计算$[0,\epsilon^2]$范围内的距离平方值的数量，如算法 3 所示。具体地，轨迹服务和密码服务利用CKKS同态加密算法的同态性实现 1.3 节所述安全比较协议，并通过 SIMD 打包技术实现比较操作的高并发处理。如算法 3 所示，密码服务首先将集合 $\boldsymbol{D}_c''=\{\mathrm{dist}_0'',\cdots,\mathrm{dist}_{\beta-1}''\}$ 中每个元素转换为二进制表示 $\mathrm{dist}_i''=\mathrm{dist}_i''[\ell-1]\cdots\mathrm{dist}_i''[0]$。然后，集合中元素的同一比特被打包加密到同一密文（步骤 2)）。最后，密码服务获得 ℓ 个密文，并将其发送至轨迹服务。轨迹服务也采用类似的方法分别对随机数集合 $\boldsymbol{R}$ 和 $\boldsymbol{\epsilon}$ 中的元素进行二进制转换，并按位打包加密（步骤 6)～步骤 9)）。采用 1.3 节的安全比较协议，轨迹服务分别安全比较 dist_i'' 和 r_i，以及 dist_i'' 和 $r_i+\epsilon^2$，其中，$0\leqslant i<\beta$，进而获取索引集合 $\{i\mid 0\leqslant \mathrm{dist}_i''-r_i<\epsilon^2\}$（步骤 10)～步骤23))。比较的结果集合存储在2个密文 $\hat{b}_0$ 和 $\hat{d}_0$ 中。如果 $\mathrm{dist}_i''-r_i\geqslant 0$，则 $b_0[i]=0$；如果 $\mathrm{dist}_i''-r_i<\epsilon^2$，则 $d_0[i]=1$，也就是说，如果 $0\leqslant \mathrm{dist}_i''-r_i<\epsilon^2$，则 $(1-b_0[i])d_0[i]=1$。基于此，轨迹服务计算比较结果密文 $\widehat{\mathrm{rlt}}$（步骤 24)），并将其发送至密码服务。密码服务解密结果密文，并将其返回给轨迹服务。基于比较结果，轨迹服务能够直接计算出轨迹 tr 与 tr_k 之间的 LCSS 值（步骤 28)），进而计算出轨迹 tr 与 tr_k 之间的相似度 $\mathrm{sim}_{\mathrm{LCSS}}(\mathrm{tr},\mathrm{tr}_k)$（算法 1 步骤 17)）。

轨迹服务重复上述过程，计算出轨迹 tr 与 **TR** 中所有轨迹之间的相似度，进而选择出与 tr 最近似的轨迹 tr^* 返回给轨迹查询者。

算法 2　密文压缩算法

输入　$\hat{\boldsymbol{D}}=\{\widehat{\mathrm{dist}}_0,\cdots,\widehat{\mathrm{dist}}_{n-1}\}$

输出　压缩后的密文集合 $\hat{\boldsymbol{D}}_c$

1)　$n_c\leftarrow\left\lfloor \frac{n}{\alpha} \right\rfloor$;

2)　if　$n_c>1$　then

3)　　for　$i=0,1,\cdots,n-1$　do

4)　　　$\mathrm{idx}\leftarrow i \bmod n_c$;

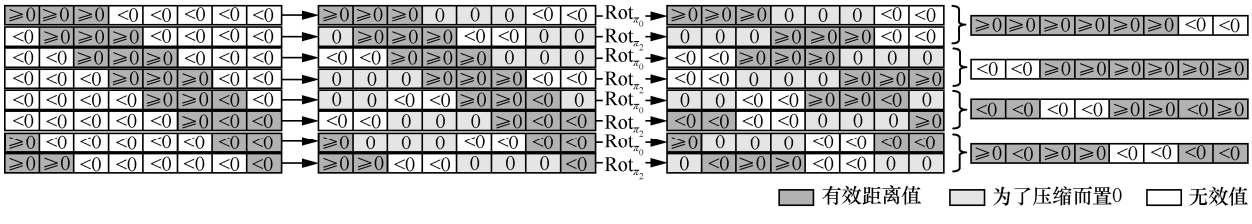


图 4　密文压缩示例

5) if idx=0 then
6) zeropad $\leftarrow \langle a_0 \mid \cdots \mid a_{n-1} \rangle$，其中，如果 $j \in \{i+\alpha \bmod n, \cdots, i+n_c\alpha-1 \bmod n\}$，那么 $a_j = 0$，否则 $a_j = 1$;
7) $\widehat{\text{dist}}_i \leftarrow \widehat{\text{dist}}_i \boxtimes \text{zeropad}$;
8) $\widehat{\text{dist}}_i \leftarrow \text{Rot}_{\pi_{(i \bmod n_c)(\alpha-1)}}(\widehat{\text{dist}}_i)$;
9) else if $0 < \text{idx} < n_c - 1$ then
10) zeropad $\leftarrow \langle a_0 \mid \cdots \mid a_{n-1} \rangle$，其中，如果 $j \in \{i-(n_c-\text{idx})\alpha+n \bmod n, \cdots, i-1 \bmod n\} \cup \{i+\alpha \bmod n, \cdots, i+(n_c-\text{idx}+1)\alpha-1 \bmod n\}$ 那么 $a_j = 0$，否则 $a_j = 1$;
11) $\widehat{\text{dist}}_i \leftarrow \widehat{\text{dist}}_i \boxtimes \text{zeropad}$;
12) $\widehat{\text{dist}}_i \leftarrow \text{Rot}_{\pi_{(i \bmod n_c)(\alpha-1)}}(\widehat{\text{dist}}_i)$;
13) else if $\text{idx} = n_c - 1$ then
14) zeropad $\leftarrow \langle a_0 \mid \cdots \mid a_{n-1} \rangle$，其中，如果 $j \in \{i-(n_c-1)\alpha+n \bmod n, \cdots, i-1 \bmod n\}$，那么 $a_j = 0$，否则 $a_j = 1$;
15) $\widehat{\text{dist}}_i \leftarrow \widehat{\text{dist}}_i \boxtimes \text{zeropad}$;
16) $\widehat{\text{dist}}_i \leftarrow \text{Rot}_{\pi_{(i \bmod n_c)(\alpha-1)}}(\widehat{\text{dist}}_i)$;
17) $\hat{\boldsymbol{D}}_c \leftarrow \cup \widehat{\text{dist}}_{i-2} \boxplus \widehat{\text{dist}}_{i-1} \boxplus \widehat{\text{dist}}_i$;
18) end if
19) end for
20) return $\hat{\boldsymbol{D}}_c$;
21) end if
22) return $\hat{\boldsymbol{D}}$;

算法 3 最长公共子序列安全计算（SLCSS）

输入 TS 输入 $\boldsymbol{R} = \{r_0, \cdots, r_{\beta-1}\}$，$\boldsymbol{\epsilon} = \{r_0+\epsilon^2, \cdots, r_{\beta-1}+\epsilon^2\}$，其中 $\beta = n\left\lfloor \frac{n}{\lfloor \frac{n}{\alpha} \rfloor} \right\rfloor$；CS 输入 $\boldsymbol{D}'_c = \{\text{dist}'_0, \cdots, \text{dist}'_{\beta-1}\}$ 和私钥 sk

输出 最长公共子序列 $\text{LCSS}(\text{tr}, \text{tr}_k)$

CS
1) for $i = \ell-1, \cdots, 0$ do
2) $\widehat{\boldsymbol{\text{dist}}}'_i \leftarrow \text{Enc}(\langle \text{dist}'_0[i] \mid \cdots \mid \text{dist}'_{\beta-1}[i] \rangle, \text{pk})$;
3) end for
4) 发送 $\{\widehat{\boldsymbol{\text{dist}}}'_i\}_{0 \leqslant i < \ell}$ 至 TS;

TS
5) $\textbf{one} \leftarrow \langle 1 \mid \cdots \mid 1 \rangle$;
6) for $i = \ell-1, \cdots, 0$ do
7) $\boldsymbol{r}_i \leftarrow \langle r_0[i] \mid \cdots \mid r_{\beta-1}[i] \rangle$;
8) $\boldsymbol{\epsilon}_i \leftarrow \langle (r_0+\epsilon^2)[i] \mid \cdots \mid (r_{\beta-1}+\epsilon^2)[i] \rangle$;
9) end for
10) for $i = \ell-1, \cdots, 0$ do
11) if $i = \ell-1$ then
12) $\hat{a}_i \leftarrow \widehat{\boldsymbol{\text{dist}}}'_i \boxtimes (\textbf{one} - \boldsymbol{r}_i)$;
13) $\hat{b}_i \leftarrow \boldsymbol{r}_i \boxtimes (\textbf{one} \boxminus \widehat{\boldsymbol{\text{dist}}}'_i)$;
14) $\hat{c}_i \leftarrow \widehat{\boldsymbol{\text{dist}}}'_i \boxtimes (\textbf{one} - \boldsymbol{\epsilon}_i)$;
15) $\hat{d}_i \leftarrow \boldsymbol{\epsilon}_i \boxtimes (\textbf{one} \boxminus \widehat{\boldsymbol{\text{dist}}}'_i)$;
16) end if
17) if $i < \ell-1$ then
18) $\hat{a}_i \leftarrow (\textbf{one} \boxminus \hat{b}_{i+1}) \boxtimes (\hat{a}_{i+1} \boxplus (\textbf{one} \boxminus \hat{a}_{i+1}) \boxtimes \widehat{\boldsymbol{\text{dist}}}'_i \boxtimes (\textbf{one} - \boldsymbol{r}_i))$;
19) $\hat{b}_i \leftarrow (\textbf{one} \boxminus \hat{a}_{i+1}) \boxtimes (\hat{b}_{i+1} \boxplus (\textbf{one} \boxminus \hat{b}_{i+1}) \boxtimes \boldsymbol{r}_i \boxtimes (\textbf{one} \boxminus \widehat{\boldsymbol{\text{dist}}}'_i))$;
20) $\hat{c}_i \leftarrow (\textbf{one} \boxminus \hat{d}_{i+1}) \boxtimes (\hat{c}_{i+1} \boxplus (\textbf{one} \boxminus \hat{c}_{i+1}) \boxtimes \widehat{\boldsymbol{\text{dist}}}'_i \boxtimes (\textbf{one} - \boldsymbol{\epsilon}_i))$;
21) $\hat{d}_i \leftarrow (\textbf{one} \boxminus \hat{c}_{i+1}) \boxtimes (\hat{d}_{i+1} \boxplus (\textbf{one} \boxminus \hat{d}_{i+1}) \boxtimes \boldsymbol{\epsilon}_i \boxtimes (\textbf{one} \boxminus \widehat{\boldsymbol{\text{dist}}}'_i))$;
22) end if
23) end for
24) $\widehat{\text{rlt}} \leftarrow (\textbf{one} \boxminus \hat{b}_0) \boxtimes \hat{d}_0$;
25) 发送 $\widehat{\text{rlt}}$ 至 CS;

CS
26) $\text{rlt} \leftarrow \text{Dec}(\widehat{\text{rlt}}, \text{sk})$;
27) 发送 rlt 至 TS;

TS
28) return $\text{LCSS}(\text{rlt})$;

4 理论分析

4.1 复杂度分析

本文利用在线计算开销和通信开销分析 pTSC 的复杂度。计算复杂度主要关注开销较大的操作，例如，加解密、密文同态运算等，而忽视明文参与的相关运算。通信复杂度主要关注密文传输的开销。

在客户端，轨迹拥有者对每条待上传的轨迹需要执行 2 次加密操作，用以加密轨迹的经纬度坐标序列，同时向轨迹服务上传 3 个密文。轨迹查询者每次查询需要执行 2 次加密操作，同时向轨迹服务发送 2 个密文。

在服务端，针对每次密态轨迹相似度计算，轨迹服务需要执行 $2n$ 次密文同态乘法，用以计算 2 条轨迹 GPS 点之间的距离平方值。轨迹服务执行 $\left\lfloor \frac{n}{\alpha} \right\rfloor - 1$ 次同态循环旋转操作将 n 个距离密文压缩为 $\beta = n\left\lfloor \frac{n}{\left\lfloor \frac{n}{\alpha} \right\rfloor} \right\rfloor$ 个密文。压缩后的密文被发送给密码服务，密码服务执行 β 次解密操作，并执行 SLCSS 协议。针对 SLCSS，轨迹服务和密码服务联合执行 2β 次安全比较。受益于 SIMD 打包技术，针对这些安全比较，轨迹服务仅需执行 $8\ell - 7$ 次同态乘法，密码服务执行 ℓ 次加密和一次解密。同时，轨迹服务和密码服务之间需要传输 $\ell + 1$ 个密文。

CKKS 同态加密算法的多项式模度是影响 pTSC 的主要指标，其值取 2 的整数次幂。多项式模度取值越大，方案安全性越高，但也会使密文尺寸变大，导致加密、解密、同态加法、同态乘法、同态旋转等操作效率降低。

4.2　安全性分析

由于CKKS同态加密算法满足语义安全，因此，轨迹服务无法从密态存储轨迹 $\widehat{\mathrm{tr}}_k$ 和兴趣轨迹 $\widehat{\mathrm{tr}}$ 中获取轨迹拥有者和轨迹查询者的轨迹信息。此外，密码服务无法从添加了随机数据的距离平方值的集合 $\boldsymbol{D}'_c = \{\mathrm{dist}'_0, \cdots, \mathrm{dist}'_{\beta-1}\}$ 中获取轨迹信息，这是因为 κ bit 的距离平方值被加上了 ℓ bit 的随机数，满足 $2^{\kappa-\ell+1}$ 的统计安全。

定义 2　服务端协议安全性。假设 TS 和 CS 之间的协议由 TS 计算并输出 $F_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$，CS 计算并输出 $F_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$，其中 $\widehat{\boldsymbol{D}}_c$ 和 sk 分别为 TS 和 CS 的输入。令 $\mathrm{View}_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$ 和 $\mathrm{View}_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$ 分别表示 TS 和 CS 执行协议时的视角。也就是说，如果 $(\widehat{\boldsymbol{D}}_c, r_{\mathrm{TS}})$ 和 $(\mathrm{sk}, r_{\mathrm{CS}})$ 分别表示 TS 和 CS 的输入及随机数，则

$$\mathrm{View}_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}) = (\widehat{\boldsymbol{D}}_c, r_{\mathrm{TS}}, m_1, \cdots, m_t)$$

$$\mathrm{View}_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}) = (\mathrm{sk}, r_{\mathrm{CS}}, m_1, \cdots, m_t)$$

其中，m_i 表示 TS 和 CS 之间传递的消息。令 $O_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$ 和 $O_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})$ 分别表示 TS 和 CS 的输出。那么，如果存在多项式仿真者 S_{TS} 和 S_{CS} 满足

$$(S_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c, F_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})), F_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk})) \equiv (\mathrm{View}_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}), O_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}))$$

$$(F_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}), S_{\mathrm{TS}}(\mathrm{sk}, F_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}))) \equiv (O_{\mathrm{TS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}), \mathrm{View}_{\mathrm{CS}}(\widehat{\boldsymbol{D}}_c;\mathrm{sk}))$$

则称服务端协议在半诚实攻击者存在的条件下是安全的。

定理 1　pTSC 的服务端协议在半诚实攻击者存在的条件下具有安全性。

证明　针对如下 2 个场景构建多项式仿真者。

1) 假设 CS 被攻击者攻陷。构建一个仿真者 S_{CS} 来仿真 CS 的视角 $\mathrm{View}_{\mathrm{CS}}$。当 TS 依次盲化集合 $\widehat{\boldsymbol{D}}_c$ 中的密文得到 $\hat{\boldsymbol{D}}'_c$，且拟发送其至 CS 时，S_{CS} 随机生成一个与 $\hat{\boldsymbol{D}}'_c$ 大小相同的集合，集合中每个元素由加密 ℓ bit 的随机数获得。然后，S_{CS} 发送该集合至 CS。当收到这些密文后，S_{CS} 解密获得一组 ℓ bit 的随机数。不存在多项式攻击者能够区分这些随机数与 $\hat{\boldsymbol{D}}'_c$ 中的元素，因为 $\hat{\boldsymbol{D}}'_c$ 中的元素是由 $\widehat{\boldsymbol{D}}_c$ 中的元素加上均匀分布的随机数获得的。此外，在 SLCSS 协议中，CS 将收到加密的比较结果。当 TS 拟发送比较结果的打包密文 $\widehat{\mathrm{rlt}}$ 至 CS 时，其包括的比较结果为 $\mathbf{rlt} = (\mathrm{rlt}_0, \cdots, \mathrm{rlt}_{\beta-1})$。$S_{\mathrm{CS}}$ 生成 β 个 1 bit 的随机数 $\mathbf{rlt}' = (\mathrm{rlt}'_0, \cdots, \mathrm{rlt}'_{\beta-1})$，将其打包加密获得密文 $\widehat{\mathrm{rlt}}' = \mathrm{Enc}(\langle \mathrm{rlt}'_0, \cdots, \mathrm{rlt}'_{\beta-1} \rangle, \mathrm{pk})$，并发送 $\widehat{\mathrm{rlt}}'$ 至 CS。CS 解密 $\widehat{\mathrm{rlt}}'$ 获得集合 $\mathbf{rlt}'$。不存在多项式攻击者能够区分 $\mathbf{rlt}$ 与 $\mathbf{rlt}'$，因为 $\mathbf{rlt}$ 与 $\mathbf{rlt}'$ 服从相同分布。综上所述，S_{CS} 和 $\mathrm{View}_{\mathrm{CS}}$ 是不可区分的。

2) 假设 TS 被攻击者攻陷。构建一个仿真者 S_{TS} 来仿真CS的视角 $\mathrm{View}_{\mathrm{TS}}$。当CS拟发送 $\{\widehat{\mathbf{dist}}'_i\}_{0 \leqslant i < \ell}$ 至 TS 时，S_{TS} 加密 ℓ 个 β bit 的随机数，并将密文发送给 TS。不存在多项式攻击者能够区分这 ℓ 个密文与 $\{\widehat{\mathbf{dist}}'_i\}_{0 \leqslant i < \ell}$。当 CS 拟发送比较结果 rlt 至 TS 时，S_{TS} 打包 β 个 1 bit 的随机数，并发送至 TS。不存在多项式攻击者能够区分打包的随机数与 rlt，因为它们服从相同分布。综上所述，S_{TS} 和 $\mathrm{View}_{\mathrm{TS}}$ 是不可区分的。

上述分析证明了 pTSC 的服务端协议满足定义 2 的安全性，其在半诚实攻击者存在的条件下具有安全性。证毕。

pTSC 能够解决 2.2 节描述的隐私威胁，具体分析如下。

轨迹跟踪攻击。轨迹服务或密码服务需要获取用户轨迹中的部分 GPS 点信息来发起此攻击。在 pTSC 中，轨迹服务无法从 CKKS 密文中获取轨迹中任何 GPS 点信息，甚至无法获取存储轨迹的长度。密码服务同样无法获得任何轨迹信息。

大规模轨迹推理攻击。轨迹服务或密码服务可以通过获取用户轨迹中的部分 GPS 点信息发起此攻击，也可以引入一些攻击背景知识通过关联分析多条轨迹来发起此攻击，例如分析轨迹长度、轨迹重叠来关联已知的热点线路。CKKS 同态加密算法满足语义安全，因此，加密 2 条相同的轨迹会得到 2 个不可区分的密文。同时，轨迹服务也无法获取存储轨迹的长度。因此，轨迹服务无法对用户轨迹开展有效的关联分析。

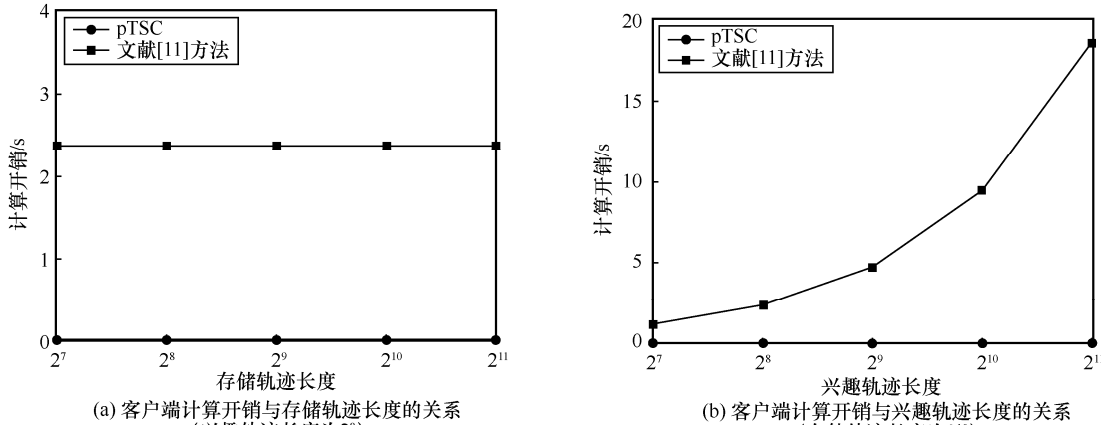
5 实验分析

本文采用微软亚洲研究院 Geolife 项目提供的 GPS 轨迹数据集开展实验，其由 178 位用户从 2007 年 4 月到 2011 年 10 月收集的 17 621 条轨迹组成。本文使用 SEAL 库提供的 CKKS 同态加密算法实现了 pTSC 的原型，其中，CKKS 同态加密算法的多项式模度设置为 n=4 096，距离有效值设置为 $\kappa=16$ bit，屏蔽有效值的随机数设置为 $\ell=32$ bit。为了验证本文方法的有效性，将其与文献[11]方法对比。本文实验环境如下：Ubuntu 18.04 LTS，英特尔 i7-10700 处理器 2.9 GHz，16 GB 内存。在此实验环境下，CKKS 同态加密算法的开销如表 1 所示。

表 1 CKKS 同态加密算法的开销

多项式模度	加密时间/ms	解密时间/ms	同态加法时间/ms	同态乘法时间/ms	同态旋转时间/ms	密文尺寸/KB
1 024	0.159	0.007	0.002	0.016	—	16.6
2 048	0.293	0.013	0.004	0.031	—	33.056
4 096	1.224	0.045	0.011	0.108	0.656	65.936
8 192	3.209	0.147	0.037	0.380	2.871	394.865
16 384	11.05	0.583	0.140	1.485	16.761	789.617

客户端计算开销与存储轨迹长度和兴趣轨迹长度的关系如图 5 所示。从图 5 可以看出，pTSC 在客户端的计算开销非常低，且不会受到存储轨迹长度和兴趣轨迹长度的影响。pTSC 在客户端的计算开销明显少于文献[11]方法。此外，当兴趣轨迹长度增长时，文献[11]方法在客户端的计算开销随之增加，这意味着当兴趣轨迹较长时，此方法会给客户端带来较大的计算开销。

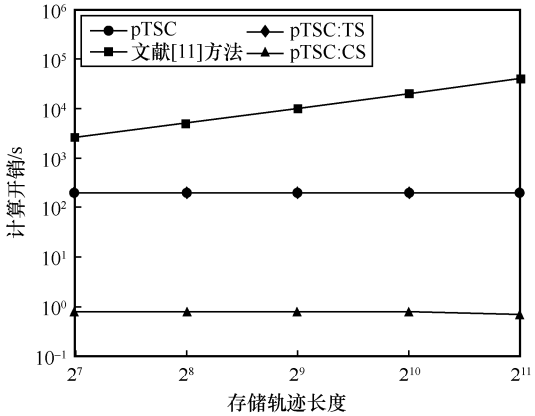


(a) 客户端计算开销与存储轨迹长度的关系（兴趣轨迹长度为2^8）

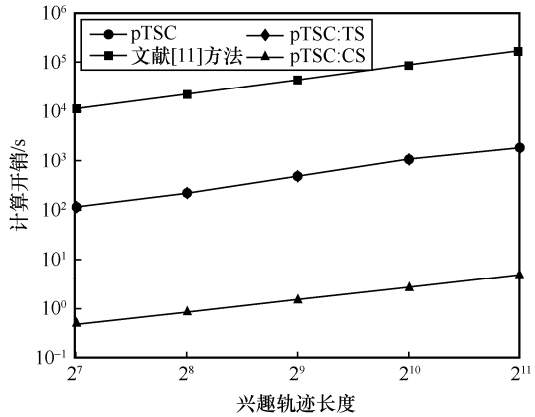
(b) 客户端计算开销与兴趣轨迹长度的关系（存储轨迹长度为2^{10}）

图 5 客户端计算开销

服务端计算开销与存储轨迹长度和兴趣轨迹长度的关系如图 6 所示。从图 6 可以看出，pTSC 在服务端的计算开销不受存储轨迹长度的影响，这是因为存储轨迹长度是加密的，短存储轨迹也不会减少计算开销。然而，当兴趣轨迹长度增长时，pTSC 服务端的计算开销线性增加，这是因为兴趣轨迹长度是非加密的，长兴趣轨迹会引发更多的 GPS 点之间的距离的计算与比较。此外，服务端轨迹服务承担了大部分计算开销，而密码服务的计算开销非常低。文献[11]方法在服务端的计算开销较高，当存储轨迹长度或兴趣轨迹长度增长时，计算开销增加明显，这导致此方法对长轨迹的相似度计算性能差，影响其可实践性。pTSC 在服务端的计算开销相比于文献[11]方法降低了 2 个数量级。



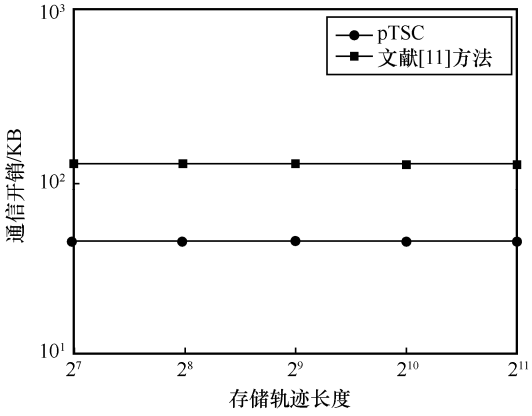
(a) 服务端计算开销与存储轨迹长度的关系（兴趣轨迹长度为2^8）



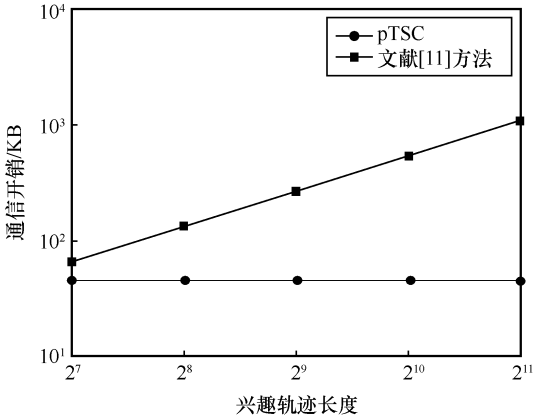
(b) 服务端计算开销与兴趣轨迹长度的关系（存储轨迹长度为2^{10}）

图 6　服务端计算开销

客户端与服务端之间的通信开销与存储轨迹长度和兴趣轨迹长度的关系如图 7 所示。从图 7 可以看出，pTSC 中服务端和客户端之间的通信开销很低，且不受存储轨迹长度和兴趣轨迹长度影响。然而，文献[11]方法的通信开销随着兴趣轨迹长度的增长而增大，这会增加客户端资源受限设备的负担。



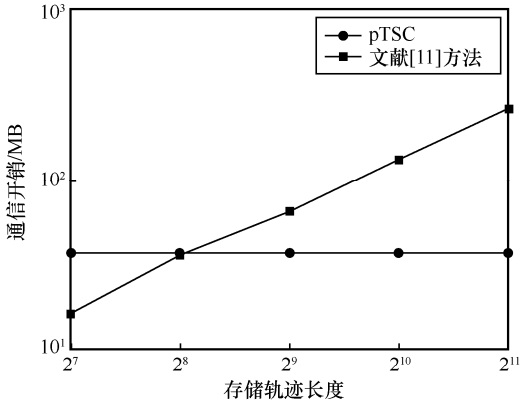
(a) 客户端与服务端之间的通信开销与存储轨迹长度的关系（兴趣轨迹长度为2^8）



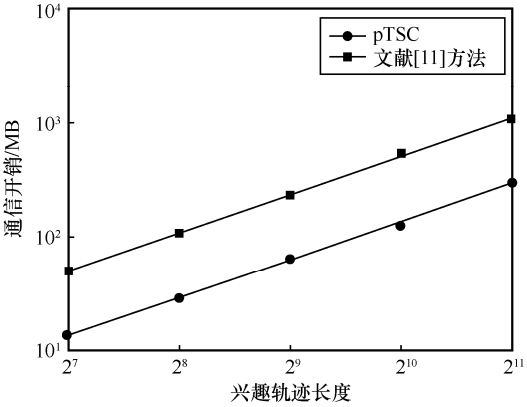
(b) 客户端与服务端之间的通信开销与兴趣轨迹长度的关系（存储轨迹长度为2^{10}）

图 7　客户端与服务端之间的通信开销

轨迹服务与密码服务之间的通信开销与存储轨迹长度和兴趣轨迹长度的关系如图 8 所示。从图 8(a)可以看出，pTSC 的服务端通信开销较低，约为 37.5 MB，且不受存储轨迹长度影响。当存储轨迹长度较小时，文献[11]方法的服务端通信开销较低，但随着存储轨迹长度的增长，其通信开销线性增加，且远高于 pTSC 方法。从图 8(b)中可以看出，pTSC 的服务端通信开销随着兴趣轨迹长度增长而线性增加，当兴趣轨迹长度由 2^7 增长到 2^{11} 时，通信开销由 18.8 MB 增加到 294.3 MB；文献[11]方法在服务端的开销由 65.9 MB 快速增加到 1054.7 MB。综上所述，pTSC 在服务端的通信性能明显优于文献[11]方法。



(a) 轨迹服务与密码服务之间的通信开销与存储轨迹长度的关系（兴趣轨迹长度为2^8）



(b) 轨迹服务与密码服务之间的通信开销与兴趣轨迹长度的关系（存储轨迹长度为2^{10}）

图 8 轨迹服务与密码服务之间的通信开销

6 相关工作

轨迹相似度计算是轨迹分析领域的基础操作之一，在明文下的计算方法已较为成熟，具体如下。1) 全局匹配算法，如 DTW（dynamic time warping）、PDTW（piecewise dynamic time warping）等；2) 部分匹配算法，如 LCSS、EDR（edit distance on real sequence）、ERP（edit distance with real penalty）等。为避免相似度计算时的轨迹泄露，一些隐私保护的轨迹相似度计算方法被提出，其通常基于同态加密[5]、姚氏混淆电路[6]、安全求交集[7]等密码学工具实现。PrivatePool[8]利用同态加密算法和安全求交集运算判断轨迹 GPS 点之间的邻近程度，进而推断出轨迹间的重叠部分。TOPPool[9]在 PrivatePool 基础上考虑了轨迹的时间属性，并优化了安全求交集运算，以提升效率。SRide[10]利用同态加密和两方安全等价测试来灵活地计算轨迹之前的重叠。类似的隐私保护的方法还包括文献[17-19]提出的方法等。上述轨迹安全计算方法主要用于解决共乘安全规划问题，通常这类问题涉及的轨迹长度较短。当轨迹长度较长时，上述方法将面临开销过高的问题。针对更通用轨迹相似度安全计算，Zhu 等[20]面向加密的时间序列提出了 DTW 的两方安全计算协议。许华杰等[21]综合方向、速度、空间、时间方面的差异度量进行相似度计算。Teng 等[22]提出了一种基于双向相似度测量的安全轨迹相似性计算（SBD）方法，并使用签名匹配过滤不同的轨迹，以降低开销。Liu 等[11]利用同态加密和姚氏混淆电路设计了 DTW、LCSS 和 EDR 的安全计算框架。基于上述方法，Teng 等[23]构建了加密轨迹搜索平台 SeTS^3，该平台支持 DTW、LCSS 和 SBD 的安全计算。虽然上述方法在性能上取得了较大的提升，但仍难以应对大规模的长轨迹安全计算，例如，针对 2 条长度为 100 的轨迹，文献[11]方法计算一次 LCSS 相似度大约需要 13 min。相比于已有方法，pTSC 具有更低的计算和通信开销。

7 结束语

为了解决轨迹外包服务中轨迹相似度计算的隐私泄露问题，本文提出了 pTSC 方法。在该方法中，轨迹拥有者使用 CKKS 同态加密算法加密轨迹，并上传到轨迹服务外包存储；轨迹查询者也使用 CKKS 同态加密算法加密兴趣轨迹，并向轨迹服务发起关于兴趣轨迹的相似度查询；轨迹服务基于密态兴趣轨迹和存储轨迹计算 LCSS 相似度。此外，本文还设计了一个密文压缩算法和密文安全比较协议，极大地提升了 pTSC 的效率。理论分析和实验结果表明，pTSC 具有安全性和高效性，且明显优于现有轨迹安全计算方法。

参考文献：

[1] 许佳捷, 郑凯, 池明旻, 等. 轨迹大数据: 数据、应用与技术现状[J]. 通信学报, 2015, 36(12): 97-105.
XU J J, ZHENG K, CHI M M, et al. Trajectory big data: data, applications and techniques[J]. Journal on Communications, 2015, 36(12): 97-105.

[2] SHAN Z H, REN K, BLANTON M, et al. Practical secure computation outsourcing[J]. ACM Computing Surveys, 2019, 51(2): 1-40.

[3] 李凤华, 李晖, 贾焰, 等. 隐私计算研究范畴及发展趋势[J]. 通信学报, 2016, 37(4): 1-11.
LI F H, LI H, JIA Y, et al. Privacy computing: concept, connotation and its research trend[J]. Journal on Communications, 2016, 37(4): 1-11.

[4] 万盛, 李凤华, 牛犇, 等. 位置隐私保护技术研究进展[J]. 通信学报, 2016, 37(12): 124-141.
WAN S, LI F H, NIU B, et al. Research progress on location priva-

cy-preserving techniques[J]. Journal on Communications, 2016, 37(12): 124-141.

[5] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]//Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999: 223-238.

[6] HUANG Y, EVANS D, K J, et al. Faster secure {two-party} computation using garbled circuits[C]//Proceedings of 20th USENIX Security Symposium. Berkeley: USENIX Association, 2011: 35.

[7] FREEDMAN M J, NISSIM K, PINKAS B. Efficient private matching and set intersection[C]//Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004: 1-19.

[8] HALLGREN P, ORLANDI C, SABELFELD A. PrivatePool: privacy-preserving ridesharing[C]//Proceedings of IEEE 30th Computer Security Foundations Symposium. Piscataway: IEEE Press, 2017: 276-291.

[9] PAGNIN E, GUNNARSSON G, TALEBI P, et al. TOPPool: time-aware optimized privacy-preserving ridesharing[J]. Proceedings on Privacy Enhancing Technologies, 2019, 2019(4): 93-111.

[10] AÏVODJI U M, HUGUENIN K, HUGUET M J, et al. SRide: a privacy-preserving ridesharing system[C]//Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks. New York: ACM Press, 2018: 40-50.

[11] LIU A, ZHENGY K, LIZ L, et al. Efficient secure similarity computation on encrypted trajectory data[C]//Proceedings of IEEE 31st International Conference on Data Engineering. Piscataway: IEEE Press, 2015: 66-77.

[12] SU H, LIU S C, ZHENG B L, et al. A survey of trajectory distance measures and performance evaluation[J]. The VLDB Journal, 2020, 29(1): 3-32.

[13] CHEON J H, KIM A, KIM M, et al. Homomorphic encryption for arithmetic of approximate numbers[C]//Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2017: 409-437.

[14] LYUBASHEVSKY V, PEIKERT C, REGEV O. On ideal lattices and learning with errors over rings[J]. Journal of the ACM, 2013, 60(6): 1-35.

[15] SMART N P, VERCAUTEREN F. Fully homomorphic SIMD operations[J]. Designs, Codes and Cryptography, 2014, 71(1): 57-81.

[16] QI Y N, ATALLAH M J. Efficient privacy-preserving k-nearest neighbor search[C]//Proceedings of The 28th International Conference on Distributed Computing Systems. Piscataway: IEEE Press, 2008: 311-319.

[17] HE Y Y, NI J B, WANG X Y, et al. Privacy-preserving partner selection for ride-sharing services[J]. IEEE Transactions on Vehicular Technology, 2018, 67(7): 5994-6005.

[18] NARAYANAN A, THIAGARAJAN N, LAKHANI M, et al. Location privacy via private proximity testing[C]//Proceedings of Network and Distributed System Security. Piscataway: IEEE Press, 2011: 11.

[19] SALDAMLI G, CHOW R, JIN HX, et al. Private proximity testing with an untrusted server[C]//Proceedings of the sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks. NewYork: ACM Press, 2013: 113-118.

[20] ZHU H, MENG X, KOLLIOS G. Privacy preserving similarity evaluation of time series data[C]//Proceedings of Extending Database Technology. Berlin: Springer, 2014: 499-510.

[21] 许华杰, 吴青华, 胡小明. 基于轨迹多特性的隐私保护算法[J]. 计算机科学, 2019, 46(1): 190-195.
XU H J, WU Q H, HU X M. Privacy protection algorithm based on multi-characteristics of trajectory[J]. Computer Science, 2019, 46(1): 190-195.

[22] TENG Y P, SHI Z, ZHAO F Y, et al. Signature-based secure trajectory similarity search[C]//Proceedings of IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications. Piscataway: IEEE Press, 2021: 196-206.

[23] TENG Y P, ZHAO F Y, LIU J, et al. SeTS3: a secure trajectory similarity search system[C]//Proceedings of Database Systems for Advanced Applications. Cham: Springer International Publishing, 2022: 522-526.

[作者简介]

于海宁（1983− ），男，黑龙江萝北人，博士，哈尔滨工业大学副研究员、硕士生导师，主要研究方向为数据安全、隐私计算、应用密码学等。

张宏莉（1973− ），女，吉林榆树人，博士，哈尔滨工业大学教授、博士生导师，主要研究方向为网络与信息安全、网络测量与建模、网络计算、并行处理等。

余翔湛（1973− ），男，黑龙江哈尔滨人，博士，哈尔滨工业大学教授、博士生导师，主要研究方向为信息安全、网络流量分析等。

曲家兴（1979− ），男，黑龙江哈尔滨人，博士，黑龙江省网络空间研究中心教授级高级工程师，主要研究方向为网络安全、网络舆情分析等。

葛蒙蒙（1994− ），男，安徽亳州人，南洋理工大学博士生，主要研究方向为加密网络流量行为表征、加密网络应用识别、加密网页流量分类和网络流量入侵检测等。