移动群智感知中高效可验证的安全真值发现方法

王涛春[1,2], 张晨露[1,2], 蔡松健[1,2], 陈付龙[1,2], 沈慧敏[1,2], 谢 冬[1,2]

[1] 安徽师范大学 计算机与信息学院 芜湖 中国 241002

[2] 安徽师范大学 安徽省医疗大数据智能系统工程研究中心 芜湖 中国 241002

摘要 针对移动群智感知中参与者数据的真值和隐私保护问题, 提出了一种高效可验证的安全真值发现方法 EVSTD, 通过安全迭代更新参与者权值和评估对象真值, 从而得到对象的真实数据。EVSTD 中, 参与者利用本地随机数和协商随机数对敏感数据进行双掩码数据扰动, 使得 EVSTD 不仅能够保证敏感数据的隐私性, 且解决了参与者因延迟发送感知数据而导致的敏感数据泄露问题。同时, EVSTD 利用秘密共享协议解决了参与者掉线或失效的问题, 且通过动态选择 L 邻居节点策略让参与者只与其关联邻居进行通信从而大大降低了参与者的计算和通信开销。此外, 参与者通过计算敏感数据的同态哈希值以用于数据的验证并上传给服务器, 服务器对敏感数据进行聚合和对验证信息进行乘积, 并将计算结果发送给参与者, 参与者再对聚合结果和证明信息进行验证, 验证通过则说明聚合结果正确, 进一步保证了真值发现结果的可信性, 防止服务器对参与者的敏感数据进行篡改, 保证了聚合结果的真实性。实验结果显示所提方法在保证数据隐私的同时获得真实可靠的数据信息, 且能够有效的防止服务器篡改数据和共谋攻击。

关键词 移动群智感知; 真值发现; 数据隐私; 验证; 双掩码

中图法分类号 TP309 DOI 号 10.19363/J.cnki.cn10-1380/tn.2024.03.09

An Efficient and Verifiable Secure Truth Discovery in Mobile Crowdsensing

WANG Taochun[1,2], ZHANG Chenlu[1,2], CAI Songjian[1,2], CHEN Fulong[1,2], SHEN Huimin[1,2], XIE Dong[1,2]

[1]School of Computer and Information, Anhui Normal University, Wuhu 241002, China

[2]Anhui Engineering Research Centers of Medical Big Data Intelligent System, Anhui Normal University, Wuhu 241002, China

Abstract Aiming at the privacy protection of participant's truth data in mobile crowdsensing, we proposed an efficient and verifiable security truth discovery method, named EVSTD. By the security iterations, the participant's weight value and the truth of the evaluated object are updated, so we can obtain the truth of the object. In EVSTD, participants use local seed to generate a local random number, and negotiate seed with associated neighbors to generate a negotiated random number by the key agreement protocol. Participants use local random number and negotiated random number to disturb sensitive data with double masks, which can not only ensure the privacy of sensitive data, but also solve the problem of sensitive data leakage caused by delayed sending of perceived data by participants. At the same time, EVSTD uses secret sharing protocol to solve the problem of disconnection or invalidation of participants, and the strategy of select L-neighbor node dynamically to let participants only communicate with their associated neighbors, thus greatly reducing the computational and communication costs of participants. At the same time, when the participant generates disturbed data, it calculates the homomorphic hash of sensitive data and the data used for verification and uploads them to the server. The server calculates the aggregation result and its proof according to the sensitive data and verification information, and finally sends them to participants. The participant verifies the aggregation results and proof data sent by the server according to the existing verification information. If the verification passed, the aggregation result will be correct, which further guarantees the credibility of the truth discovery results, so as to solve the problem that the cloud server may tamper with the participant's sensitive data, and ensure the reality of the aggregation results in mobile crowdsensing perception. The experimental results show that the proposed method can identify the true and reliable data information while protecting data privacy, and can prevent servers from tampering with data and conspiracy attacks.

Key words mobile crowd sensing; truth discovery; data privacy; verification; double mask

通信作者: 王涛春, 博士, 教授, Email: wangtc@ahnu.edu.cn。

本课题得到国家自然科学基金项目(No.62272006,No.61972438)、安徽省重点研究与开发计划项目(No.2022a05020049)、安徽省自然科学基金项目(No.2108085MF219)、安徽省教育厅高校自然科学研究重点项目(No.2023AH052695)资助。

收稿日期: 2022-05-03; 修改日期: 2022-09-18; 定稿日期: 2023-11-01

1 引言

随着移动智能设备的快速普及, 移动群智感知(Mobile CrowdSensing, MCS)得到越来越多的研究, 并且在实际生活中得到了广泛的应用[1-9]。但是 MCS 面临着参与者的隐私安全问题。例如, 参与者的位置、健康数据以及购物信息等隐私数据的泄露, 这是限制 MCS 应用发展的一个重要因素。

通过收集参与者的感知数据进行分析处理, 能够使得 MCS 应用服务于参与者, 其中数据对 MCS 应用的决策起到决定性的作用, 但是由于外界的环境及恶意节点因素的, 导致得不到真值, 且数据质量的高低直接影响 MCS 的决策和服务质量, 因而找到可靠真实的数据的真值发现方法[10-18]到了很多关注, 当前针对真值发现的隐私保护研究中, 通常采用数据加密[19-21]、数据扰动[22]等方案来保护参与者的隐私数据。但是这些方案都是采用单个或者两个非共谋云服务器进行数据聚合, 云服务器可能存在单点故障问题, 从而影响聚合结果的真实性。

本文设计了一种基于 MCS 的高效可验证的安全真值发现方法 EVSTD。EVSTD 利用双掩码保证数据的隐私性并通过 CRH[14]算法完成真值发现。同时, 所有参与者动态地选择 L 个邻居节点进行信息交互, 从而提高 MCS 网络的通信效率并降低参与者和云服务器的计算开销。由于云服务器可能存在丢弃或篡改聚合结果的行为, 破坏聚合数据的真实性。在 EVSTD 中, 参与者向云服务器发送感知数据和验证数据, 云服务器将感知数据进行聚合得到聚合结果, 同时通过验证材料得到聚合验证材料, 并将聚合结果和聚合验证材料发送给参与者, 参与者通过验证材料验证聚合结果的真实性。实验结果表明, EVSTD 能够实现安全高效的真值发现, 并且具备抵抗云服务器单点故障的能力。

本篇文章的主要贡献如下:

- 本文提出了一种高效可验证的安全真值发现方法(EVSTD)。EVSTD 中, 利用双掩码数据混淆、Diffie–Hellman 密钥协商[24](以下简称 DH 密钥协商)对参与者的隐私数据进行混淆, 从而保护参与者的隐私。并通过 Shamir 秘密共享[23]将参与者用于生成扰动数据的种子进行秘密分享, 从而解决了参与者掉线而导致聚合结果不准确的问题。
- 不同于一般方法需要所有参与者两两之间需要进行信息交互的方案, EVSTD 中参与者动态地选择 L 个邻居节点进行通过 S 进行信息传递从而降低了参与者的交互频率。因此, 在保证感知数据隐私性的情况下, 降低了参与者的计算成本和通信开销。
- 针对云服务器可能存在的丢失部分或伪造感知数据的情况, EVSTD 利用同态哈希函数和双线性配对, 提出了一种真值结果验证方案, 验证云服务器真值结果的真实性。

第 2 部分描述了本文的相关工作; 第 3 部分描述了预备知识; 第 4 部分对 EVSTD 算法进行了细节讨论分析; 第 5 部分通过实验进行安全性以及性能的分析; 第 6 部分进行总结。

2 相关工作

数据的真实性是 MCS 提供高质量服务的基础, 真值发现是获取真实感知数据的有效方法, 是当前的研究热点。Yin 等人[24]通过不同网站显示信息的可靠性, 首次提出了一种基于假设概率的真值发现方法, 该算法思想为一个提供了大量真实信息的网站是可靠的, 从而认定该网站是值得信赖的并由此信赖该网站所提供的信息, 最终迭代得到对象的最终真值。Dong 等人[25]提出了贝叶斯模型和隐马尔可夫模型的真值发现方法, 通过概率的方式, 从变量的密度估计对象的真值, 同时处理参与者提供错误或过时的信息。Li 等人[14]提出了一个具有高准确性且能够处理异构数据的真值发现方法 CRH, 该算法通过迭代更新参与者的权值和观测对象的真值最终让两次迭代的真值差达到收敛条件后停止迭代。Li 为解决参与者在感知过程中出现的长尾问题又提出了一种真值发现方法 CATD[15]。但上述方法均没有考虑感知数据的隐私性。

由于真值发现方法中参与者的感知数据都是明文传输, 存在隐私泄漏的风险。Miao 等人[19]首次提出了基于真值发现的隐私保护方案 PPTD, 通过同态加密技术, 参与者的感知数据和权重信息得到了隐私保护, 但是该方案对参与者的要求较高, 即所有参与者都需要通过 Paillier 同态加密技术对感知数据和权重进行加密, 而很多参与者受限于移动设备, 并不能高效地参与到其中。Miao 等人[26]提出了一种轻量级的真值发现隐私保护方法 L-PPTD, 同时为减少用户负担、提高系统效率, 设计了一种双非共谋服务器来实现一个更为轻量级的真值发现隐私保护方法 L^2-PPTD, 但这种方案的设计前提是两个非共谋服务器, 在实际情况下, 用户的隐私信息并不能得到完全保障, 仍然存在隐私泄漏的风险, 同时 PPTD 和 L^2-PPTD 要求所有参与者在线, 无法解决参与者

掉线的情况。Zheng 等人[21]提出了一种使用乱码电路(GC)和加法同态密码系统来实现可信的真值发现方法 NPPTD, 该方案可以解决所有参与者的掉线问题, 且能够保护参与者的感知数据和权重信息, 但是由于乱码电路的生成和传输产生较大的计算和通信开销, 所以 NPPTD 效率较低。Xu 等人[22]通过 Shamir 秘密共享来解决所有用户必须在线的问题, 提出了具有隐私保护的真值发现方法 EPTD, 但 EPTD 中所有参与者需要两两通过 Diffie-Hellman 密钥交换协议协商公共密钥以及秘密共享, 网络中参与者、服务器的通信和计算开销较大。

3 预备知识

本节首先介绍真值发现方法 CRH 的基本概念, 其次描述了 EVSTD 的网络模型和攻击模型。

3.1 真值发现

真值发现是一种提高数据质量的方法, 它能够对不同数据源提供的冲突信息进行数据整合, 从而获得近似真实的数据。本文采用 CRH 真值发现方法[14]迭代进行权值更新和真值评估阶段。假设有 N 个参与者, 每个参与者收集 M 个对象数据, 第 i 个参与者的权值表示为 W_i, 第 i 个参与者对第 m 个对象的观测值表示为 x_m^i, 对象 m 的真值表示为 x_m^*, 对象 m 的感知数据标准差 std_m, 第 i 个参与者对第 m 个对象的观测与真值的距离计算公式为:

$$d_{ist}\left(x_m^i, x_m^*\right)=\frac{\left(x_m^i - x_m^*\right)^2}{std_m} \tag{1}$$

参与者 i 的权值计算公式为:

$$w_i = \log\left(\frac{\sum_{i=1}^{N}\sum_{m=1}^{M} d_{ist}\left(x_m^i, x_m^*\right)}{\sum_{m=1}^{M} d_{ist}\left(x_m^i, x_m^*\right)}\right) \tag{2}$$

对象 m 的真值计算公式为:

$$x_m^* = \frac{\sum_{i=1}^{N} w_i x_m^i}{\sum_{i=1}^{N} w_i} \tag{3}$$

具体执行过程如算法 1 所示:

算法 1. 真值发现

输入: N 个参与者收集的 M 个对象的感知数据 $\left\{x_m^i\right\}_{i,m=1}^{N,M}$, 阈值 σ

输出: M 个对象的真值 $\left\{x_m^*\right\}_{m=1}^{M}$

初始化所有对象真值 $\left\{x_m^*\right\}_{m=1}^{M}$

1. WHILE (1)
2. FOR i = 1, 2, ..., N do
3. FOR m = 1, 2, ..., M do
4. 计算权值/*根据公式(1)(2)计算权值*/
5. 计算真值/*根据公式(3)计算真值*/
6. END FOR
7. END FOR
8. IF 真值差≤σ CONTINUE; /*如果满足收敛条件停止迭代, 否则继续真值迭代*/
9. END WHILE

算法 1 迭代进行权值更新与真值评估。在权值更新阶段, 首次迭代则真值为初始化真值, 否则真值为上轮迭代的真值。根据公式(1)计算所有参与者的感知数据与所有对象真值的距离, 并将所有参与者的距离进行聚合, 再由公式(2)计算出权值; 在真值评估阶段, 聚合所有参与者的权值以及加权值, 最终由公式(3)计算出对象的真值。迭代进行权值更新与真值评估, 直到真值差≤σ, 得出最终真值。

3.2 网络模型

在 EVSTD 中, MCS 系统网络模型主要包括四类实体: 密钥管理中心, 参与者, 用户, 云服务器。如图 1 所示, 密钥管理中心向参与者分发密钥及参数; 云服务器向参与者发布任务并接收感知数据; 参与者接受任务并收集上传感知数据。

- 密钥管理中心(Key Manager Center, KMC)。KMC 负责为每个参与者分配密钥及用于验证聚合结果的参数, 并且 KMC 与参与者之间是在安全的通信信道进行信息传递。
- 参与者(Participant, P)。P 是感知任务的参与者, 负责接收平台发送的感知任务, 一旦接受任务, 即通过移动设备对感知对象进行数据采集, 处理后上传至云服务器。
- 云服务器(Server, S)。S 负责接收参与者上传的感知数据并进行聚合, 得到聚合结果以及聚合结果证明, 并发送给参与者进行验证。
- 用户(User, U)。数据的使用者, 云服务器通过任务分发获取到精确数据, 用户通过平台获取数据服务。

3.3 攻击类型

在 MCS 系统中, 攻击类型主要分为外部攻击和内部攻击:

1) 外部攻击是指外部攻击者通过窃听无线网络传输的敏感信息来破坏数据的机密性, 在 EVSTD 中, 假设外部攻击者可以窃听整个网络, 即外部攻击者

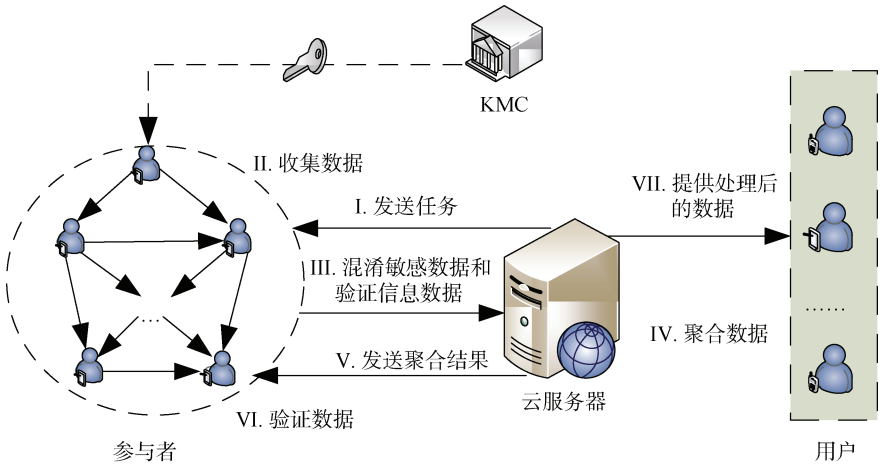


图 1 EVSTD 网络模型

Figure 1 EVSTD network model

可以窃听所有参与者发送的扰动数据、加密切片和服务器的聚合结果。

2) 内部攻击是指攻击者是指 MCS 系统中的参与者、用户或云服务器。一种攻击是参与者、用户或云服务器试图推导其他参与者的敏感信息; 另一种攻击是云服务器可能丢弃部分参与者的感知数据或篡改聚合结果, 导致真值发现的聚合结果不准确。此外, 参与者、用户或云服务器可能会进行共谋攻击来获取其他参与者的敏感信息。在 EVSTD 中, 假设所有的参与者或用户能够严格按照协议执行。

4 EVSTD

本节介绍了一种安全可验证的真值发现方法 EVSTD, 首先介绍 EVSTD 算法思想, 并详细描述了网络拓扑结构、权值更新和真值评估 3 个主要阶段, 最后对 EVSTD 的安全性进行了分析。

4.1 算法思想

在 EVSTD 中, 假设有 N 个参与者接受任务观测 M 个对象, 获取 M 个对象的真值。算法主要包括三个阶段: (1) 动态选择 L 邻居节点阶段。为降低参与者的计算成本和通信开销, EVSTD 通过 KMC 分发秘钥及参数, 参与者随机选择 L 个邻居节点作为出度邻居节点并将选择参与者集合发送给 S。(2) 安全可验证的权值更新阶段。参与者将混淆后的距离数据(隐私数据)及用于验证的计算信息发送至 S, S 对其计算处理并回传给参与者, 参与者验证聚合的距离数据无误后再更新权值, 完成参与者的权值更新。(3) 安全可验证的真值评估阶段, 参与者将混淆后的权值以及加权值发送给 S, S 聚合数据并进行真值计算, 最终参与者对 S 发送的真值信息进行验证, 以确定真值结果的真实性。算法主要流程如下图所示:

流程 1. EVSTD 执行流程

Process 1 EVSTD execution process

· 阶段 1 (L 邻居节点的动态选择):

-可信第三方给所有参与者 N_i 分发两对密钥 $S_i^1=(sk_i^1, pk_i^1)$ 、 $S_i^2=(sk_i^2, pk_i^2)$, (α,β), $K=(K1, K2)$, (λ , η)和 d。

-所有参与者 N_i 将两个公钥发送给 S。

-S 记录收到的公钥的参与者 ID 并组成集合 U_P 并生成用于验证信息 *MRHL* 并发送给 U_P 集合参与者。

-U_P 集合参与者 N_i 选择 L 个邻居节点作为自己的出度邻居为集合 U_i^{Out} ,并将集合 U_i^{Out} 发送给服务器 S。

-服务器 S 发送每个参与者 N_i 的入度邻居节点 U_i^{In} , 关联邻居 U_i 和验证信息 *MRHL*'。

-每个参与者 N_i 进行恶意节点检测, 首先检查接收到的公钥数量是否大于 $4\times L$, 若大于则重新进行选择, 其次验证服务器发送的 *MRHL*' 是否包含于 *MRHL*, 且检查对应的值与是否一致, 一致则 L 邻居节点选择完成。

· 阶段 2 (安全的权值更新阶段):

-U_P 集合中参与者 N_i 将 b_i^1 和 sk_i^1 切分出共享碎片并加密发送给服务器。

-S 记录收到的加密切片的参与者 ID 并组成集合 U_E 并将集合信息与加密切片转发给对应参与者。

-收到加密碎片的参与者 N_i 将对应的切片解密, 并生成的两种随机数加入到距离数据 Dis_i 中得到扰动后的距离数据 $\overline{Dis_i}$, 最终所有参与者 N_i 将 $\overline{Dis_i}$ 发送给服务器。

-S 记录收到混淆数据的参与者 ID 并组成集合 U_D, 并根据参与者发送数据的情况判断参与者是否掉线, 并以此向所有参与者请求对应的秘密数据切片, 抵消添加的随机数, 再对混淆后的距离数据进行聚合, 得到所有参与者敏感数据 Dis_i 聚合结果。

-服务器将聚合结果发送给所有在线参与者, 参与者根据公式(1)和(2)进行权值更新。

·阶段 3 (安全的真值评估更新):

-参与者更新随机数种子为 b_i^2 ,将 b_i^2 切片加密发送给服务器 S 转发给出度邻居节点。

-参与者计算加权值 $W_i \times X_m^i$, 并将两种随机数分别加入到权值、加权值中, 得到混淆的数据 $\overrightarrow{W_i}$ 和 $\overrightarrow{M_i}$, 最终将 $\overrightarrow{W_i}$ 和 $\overrightarrow{M_i}$ 发送给服务器;

-服务器根据参与者发送数据的情况判断参与者是否掉线, 并以此向参与者请求不同的切片数据, 恢复随机数进行抵消, 再对混淆的数据进行聚合, 得到所有参与者权重 W_i 的聚合结果以及加权值 $W_i \times X_m^i$ 的聚合结果, 最终根据公式(3)计算真值。

-服务器判断相邻两次的真值差是否小于阈值, 小于则最后一次迭代的真值为所求结果协议结束, 否则继续迭代上述阶段。

4.2 *L* 邻居节点的动态选择

每个参与者动态地选择 L 个邻居节点进行 DH 密钥协商和 Shamir 秘密共享, 以降低通信开销和计算开销, 同时在部分参与者的退出时保证聚合结果的正确性, 提高了真值的准确性。处理过程如图 2 所示:

Step1: 分发参数

KMC 给每个参与者 N_i 分配两对密钥 $S_i^1=(sk_i^1, pk_i^1)$ 和 $S_i^2=(sk_i^2, pk_i^2)$, 且给所有参与者广播同态哈希函数秘密参数(α,β),伪随机函数的秘密参数 $K=(K1, K2)$, (λ, η)和正整数 d。

Step2: 发送用户集合、真值及验证信息

每个参与者N_i将两个公钥(pk_i^1, pk_i^2)发送给S, S记录接收到公钥参与者的ID, 记为集合 U_P, 且对 U_P 集合中所有参与者 N_i 的两个公钥构造 Merkle[27]得到 Merkle 根哈希 MRH=SHA256(SHA256(pk_i^1) +SHA256 (pk_i^2)), 记所有参与者的 Merkle 树根哈希组成的集合为 *MRHL*, 最后将集合 U_P、*MRHL*、所有对象的真值 $\{x_m^*\}_{m=1}^M$ (如果是第一次迭代, 所有对象的真值则由系统初始化; 否则为上一轮迭代的真值)广播给 U_P 集合中的参与者。

Step3 : L 个邻居节点选择

U_P 集合中每个参与者 N_i 随机选择 L 个参与者节点记为出度邻居节点, 则选择 N_i 为出度邻居节点的参与者集合为参与者 N_i 的入度邻居节点集合, 记为 U_i^{In}, 最终将出度邻居节点集合U_i^{Out} 发送给 S。S 收到后, 记 $U_i=U_i^{\text{Out}} \cup U_i^{\text{In}}$ 为参与者 N_i 的关联邻居节点集合, 并向每个参与者 N_i 发送其入度邻居节点集合U_i^{In} 以及关联邻居节点的公钥, 同时给每个 N_i 发送关联邻居节点的公钥组成的 Merkle 树的根哈希集合 *MRHL*'。

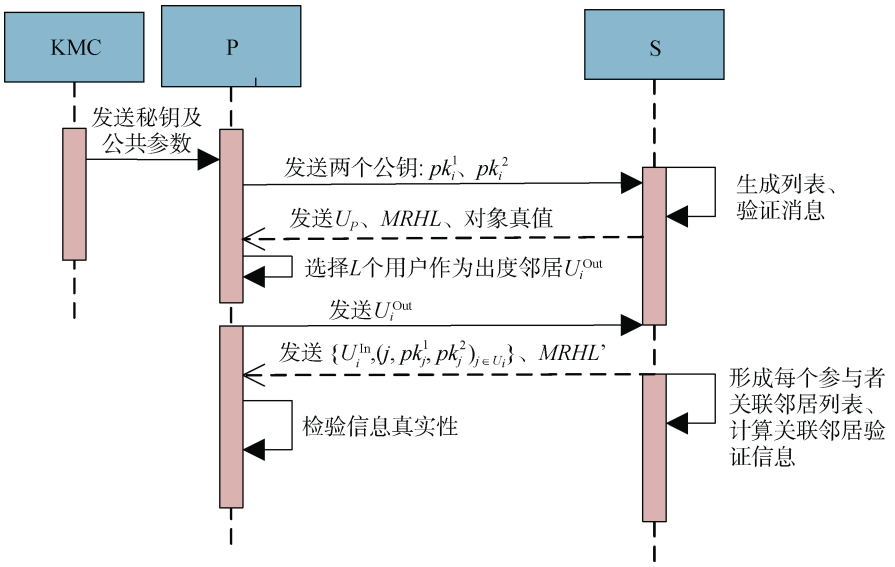


图 2 *L* 邻居节点的动态选择

Figure 2 Dynamically select *L* neighbor nodes

Step4: 恶意节点检测

参与者 N_i 检查 S 发送的公钥数量是否大于 $4\times L$, 大于则说明云服务器加入了恶意节点, 协议结束, 否则检查收到的 *MRHL*'是否包含于且对应的值否与 *MRHL* 相同, 相同则 L 邻居节点完成选择, 否则云服务器篡改了信息, 协议结束。

算法 2: L 邻居节点的动态选择

输入: 所有参与者的公钥 pk_i^1, $pk_i^2 (i \in N)$

输出: 所有参与者的出度邻居 U_i^{Out}、U_i^{In} 和关联邻居 $U_i (i \in N)$

1. $L=\lfloor \log N \rfloor$; /* L 为参与者所要选择的邻居节点数*/
2. FOR i=1, 2, 3, …, N do /*参与者 N_i 将公钥发送给 S*/
3. pk_i^1, $pk_i^2 \rightarrow$S;
4. $i \rightarrow U_P$; /* S 收集发送公钥的参与者 ID, 并组成参与者集合 U_P */
5. END FOR
6. FOR i = 1, 2, 3, …, .N do
7. $(U_P, \{x_m^*\}_{m=1}^M, MRHL) \rightarrow N_i$
8. U_i^{Out}=Pseudorandom (U_P, L); /*参与者选择 L 个出度邻居节点集合*/
9. $U_i^{\text{Out}} \rightarrow$S; /* 参与者将选择的出度邻居节点集合发送给 S*/
10. /*S 发送每个参与者的入度邻居集合、关联邻居公钥和验证信息发送给参与者 */
11. $(U_i^{\text{In}}$、$(pk_i^1$、$pk_i^2)_{i \in U_i}$、$MRHL') \rightarrow N_i$
12. /* 检查 $MRHL'$是否包含于且对应的值与 $MRHL$ 是否相同*/
13. check $MRHL' \subseteq MRHL$
14. $U_i=U_i^{\text{Out}} \cup U_i^{\text{In}}$ /* 记 U_i 为参与 N_i 的关联邻居节点集合*/
15. END FOR

4.3 安全可验证的权值更新

完成 L 邻居节点的动态选择后, EVSTD 采用 CRH 真值发现算法[14]进行权值更新: 所有参与者将扰动后的观测距离数据(隐私数据)及验证数据信息发送给 S, S 将所有参与者的观测距离数据及验证信息进行安全聚合得到所有参与者的距离和以及聚合结果证明并发送给所有参与者进行验证, 所有参与者验证通过后, 根据公式(2)进行权值更新, 之后进入真值评估阶段, 如图 3 所示:

Step1: 生成并发送加密切片

(1) U_P 集合中每个参与者 N_i 本地生成一个随机数种子 b_i^1, 通过 Shamir 算法将 b_i^1 和 sk_i^1 切分出秘密分享的切片 $Y_i^{b_i^1}=\{y_{i,1}^{b_i^1}, y_{i,2}^{b_i^1}, ..., y_{i,L}^{b_i^1}\}=S_{slice}(t, L, b_i^1)$ 和 $Y_i^s = \{y_{i,1}^s, y_{i,2}^s, ..., y_{i,L}^s\}=S_{slice}(t, L, sk_i^1)$ 用于后续发送给 L 个出度邻居节点, 并用私钥 sk_i^2 与出度邻居节点 N_j ($j \in U_i^{Out}$)的公钥 PK_j^2 进行 DH 协商, 协商密钥为 $Q_{i,j}=K_c(sk_i^2, pk_j^2)$, 将切片加密为 $E_{i,j}=Enc(Q_{i,j}, (i \| j \| Y_i^{b_i^1} \| Y_i^s)$, 随后将$(j,E_{i,j})$发送给 S。

(2) S 验证收到的加密切片的参与者数量是否大于 Shamir 秘密共享恢复的阈值 t, 如果小于则本轮处理结束, 否则记录接收到加密切片的参与者的 ID, 记为集合 $U_E (U_E \subseteq U_P)$, 将收到的加密切片 $E_{i,j}$ 转发给在集合 U_E 中参与者 N_i 对应的出度邻居节点 N_j。

Step2: 解密切片

收到 S 转发的加密切片的 N_i 的出度邻居节点 N_j 与参与者 $N_i (i \in U_E)$的公钥 PK_i^2 进行 DH 协商得到密钥 $Q_{i,j}=K_c(sk_j^2, pk_i^2)$, 之后解密 $E_{i,j}$ 得到 $Y_i^{b_i^1}$ 和 Y_i^s 的共享加密切片。

Step3 :生成扰动数据

(1) U_E 集合中的参与者 N_i 用私钥 sk_i^1 与其在 U_E 集合的关联邻居节点 N_j 的公钥 pk_j^1 进行 DH 协商, 得到密钥 $r_{i,j}=K_c(sk_i^1, pk_j^1)$。

(2) U_E 集合中每个参与者 N_i 根据公式计算所有对象的真值 x_m^* 与观测值 x_m^i 的距离和 Dis_i($Dis_i = \log(\sum_{m=1}^{M} d_{ist}(x_m^i, x_m^*))$ 为要保护的参与者 N_i 的隐私数据), 并根据伪随机函数生成 $\overrightarrow{b_i^1}=PRG(b_i^1)$、$\overrightarrow{m_{i,j}}=PRG(r_{i,j})$ 与隐私数据 Dis_i 混合形成双掩码的扰动数据 $\overrightarrow{Dis_i}$ (其中对于 ID 小于 N_i 的参与者 N_j, 参与者 N_i 则在敏感数据后减去与关联邻居 N_j 协商的随机数 $\overrightarrow{m_{i,j}}$, 反之在敏感数据后加上与关联邻居协商的随机数 $\overrightarrow{m_{i,j}}$):

$$\overrightarrow{Dis_i}=Dis_i+\overrightarrow{b_i^1}-\sum_{0<j<i\ (j\in U_i)}\overrightarrow{m_{i,j}}+\sum_{i<j\leqslant N\ (j\in U_i)}\overrightarrow{m_{i,j}} \quad (4)$$

Step4: 生成验证数据及签名

(1) U_E 集合中每个参与者 N_i 生成用于验证的数据向量 $\boldsymbol{V}_i=(A_i, B_i, P_i, Q_i)$, 验证数据具体计算如下:

$$HF(Dis_i)=(A_i, B_i)=(g^{HF_{a,b}(Dis_i)}, h^{HF_{a,b}(Dis_i)})$$

$$PRG_{K1}(i)=(\lambda \times i, \eta \times i); PRG_{K2}(\tau)=(\sigma, \delta)$$ (τ 为迭代次数)

$$PRG_K(i, \tau)=(C_i, D_i)=(g^{\lambda\times i\times\sigma+\eta\times i\times\delta}, h^{\lambda\times i\times\sigma+\eta\times i\times\delta})$$

$$P_i=(C_iA_i^{-1})^{1/d}=(g^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(Dis_i)})^{1/d}$$

$$Q_i=(D_iB_i^{-1})^{1/d}=(h^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(Dis_i)})^{1/d}$$

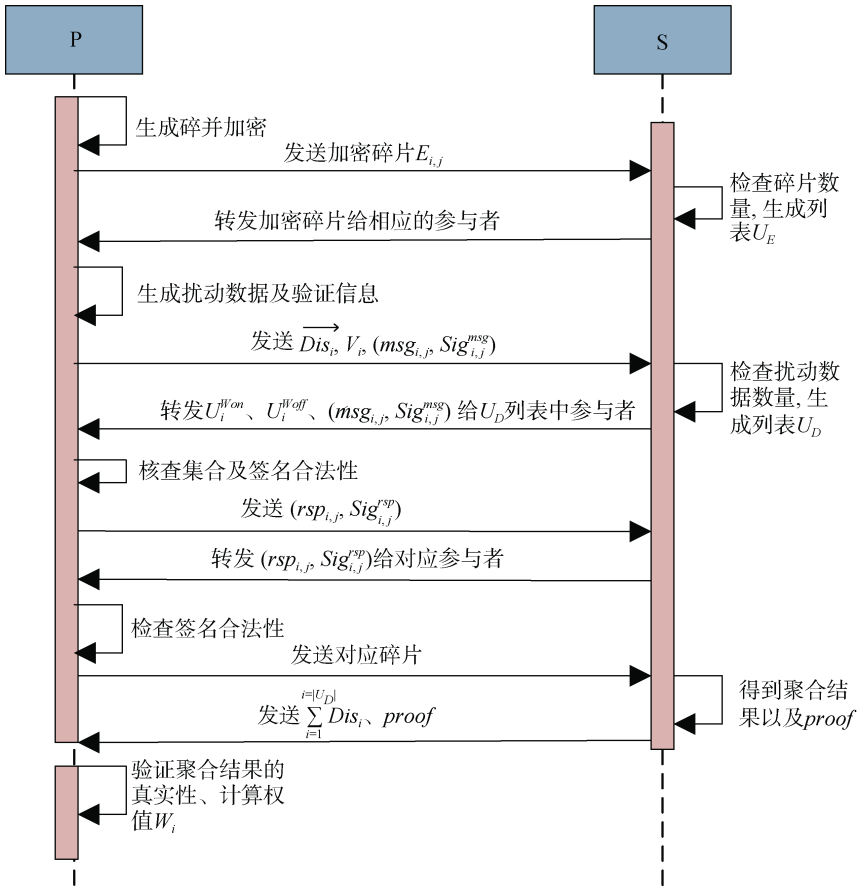


图 3　安全可验证的权值更新

Figure 3　Security and verifiable weight update

(2) 每个参与者 N_i 在本地生成一个消息 $msg_{i,j}=(i, j(j\in U_i\subseteq U_E), "involved")$，该消息用于表明参与者 N_i 已经将与关联邻居协商的随机数加入到混淆数据中，再对该消息进行签名，得到 $Sig_{i,j}^{msg}$。

(3) 每个参与者 N_i 将扰动数据 $\overrightarrow{Dis_i}$、数据向量 $V_i=(A_i, B_i, P_i, Q_i)$、$(msg_{i,j}, Sig_{i,j}^{msg})$一同发送给服务器 S。

Step 5: 计算数据

(1) 为了保证真值结果的准确性，S 验证收到的数据的参与者数是否大于 $\mu\times N$(此处 μ=0.2)，条件不成立则本轮处理结束，成立则记录所有收到数据参与者的 ID，记为集合 $U_D(U_D\subseteq U_E)$(U_E 集合中部分参与者可能因为掉线而未发送数据，即 U_D 为在线参与者集合，在 U_E 集合但不在 U_D 集合为掉线参与者)。此外，S 统计 U_D 集合中所有参与者 N_i 的在线入度邻居节点集合 U_i^{Won} 以及掉线入度邻居节点集合 U_i^{Woff}，并将 U_i^{Won}、U_i^{Woff} 和$(msg_{i,j}, Sig_{i,j}^{msg})$发送给 U_D 集合所有的参与者 N_i。

(2) 每个参与者 N_i 对参与者 N_j 发送一个确认 ack 消息 $rsp_{i,j}=(i, j, "ack")$，使用自己的私钥 sk_i^2 对该消息进行签名得到 $Sig_{i,j}^{rsp}$，将 $(rsp_{i,j}, Sig_{i,j}^{rsp})$ 发送给云服务器 S, S 将消息转发给对应的参与者 N_j。

(3) 所有收到云服务器 S 发来 ack 消息的参与者都会检查其中的签名是否合法，一旦有误立刻停止协议，否则集合 U_D 中所有参与者 N_i 发送其所有在线的入度邻居节点 N_j 的切片 $y_{j,i}^{b_i^1}(N_j\in U_i^{Won})$和掉线的入度邻居节点 N_j 的切片 $y_{j,i}^{s}(N_j\in U_i^{Woff})$给 S。

(4) S 通过接收到的切片 $y_{j,i}^{b_i^1}(N_j\in U_i^{Won})$来恢复参与者在线参与者 N_i 的 b_i^1 及 $\overrightarrow{b_i^1}$。

(5) S 通过接收到的切片 $y_{j,i}^{s}(N_j\in U_i^{Woff})$来恢复参与者掉线参与者 N_i 的 sk_i^1，并通过秘密协商恢复掉线参与者与其他在线参与者协商的 $r_{i,j}$，最终恢复出掉线参与者与其他在线参与者协商的 $\overrightarrow{m_{i,j}}$。

(6) S 得到聚合结果($U_E\backslash U_D$表示在 U_E 集合但不在 U_D 集合)并发送给参与者 N_i，见公式(5):

$$\sum_{i=1}^{i=|U_D|} Dis_i = \sum_{i=1}^{i=|U_D|}(\overrightarrow{Dis_i} - \overrightarrow{b_i^1} + \sum_{0<j<i(j\in U_i\cap(U_E\setminus U_D))} \overrightarrow{m_{i,j}} - \sum_{i<j\leqslant N(j\in U_i\cap(U_E\setminus U_D))} \overrightarrow{m_{i,j}}) = \sum_{i=1}^{i=|U_D|}\sum_{m=1}^{M} d_{ist}(x_m^i, x_m^*) \quad (5)$$

(7) S 对所有的数据向量 $\boldsymbol{V}_i = (A_i, B_i, P_i, Q_i)$ 做乘积运算得到 *proof* 并发送给每个参与者:

$$proof = \{A = \prod_{i=1}^{i=|U_D|} A_i, B = \prod_{i=1}^{i=|U_D|} B_i, P = \prod_{i=1}^{i=|U_D|} P_i, Q = \prod_{i=1}^{i=|U_D|} Q_i\}$$

Step6 : 验证聚合结果并进行权值更新

(1) U_D 集合所有参与者 N_i 计算 $\xi = \sum_{i=1}^{i=|U_D|}(\lambda\times i\times\sigma + \eta\times i\times\delta)$ 以及 $\gamma = e(g,h)^{\xi}$。

(2) 每个参与者 N_i 在两个双线性群 *G1* 和 *G2* 上对 S 发送的 $\sum_{i=1}^{i=|U_D|} Dis_i$ 同态哈希计算如公式(6):

$$HF(\sum_{i=1}^{i=|U_D|} Dis_i) = (A',\ B') = (g^{HF_{\alpha,\beta}(\sum_{i=1}^{i=|U_D|} Dis_i)}, h^{HF_{\alpha,\beta}(\sum_{i=1}^{i=|U_D|} Dis_i)}) \quad (6)$$

(3) 通过 DDH 猜想[28]、l-BDHI 假设[29]以及双线性配对的性质[30], 参与者顺序验证 $\gamma = e(A,h)$ $e(P,h)^d$, $e(P,h) = e(g,Q)$ 和 $e(A,h) = e(g,B)$ 是否成立, 如果不成立则 S 的聚合结果不正确(篡改或丢弃部分数据), 协议结束。之后再利用同态哈希函数性质[29] $(g^{HF_{ab}(X_1)+HF_{ab}(X_2)}, h^{HF_{ab}(X_1)+HF_{ab}(X_2)}) \rightarrow HF(X_1+X_2)$ 来验证 $(A,B) = (A',B')$ 是否成立, 如果不成立则协议结束, 否则所有参与者 N_i 根据公式(3)计算权值:

$$W_i = \log(\sum_{i=1}^{i=|U_D|}\sum_{m=1}^{M} d_{ist}(x_m^i, x_m^*)) - \log(\sum_{m=1}^{M} d_{ist}(x_m^i, x_m^*)) \quad (7)$$

算法 3.安全可验证的权值更新:

输入: 所有参与者 N_i 的公钥 pk_i^1, $pk_i^2 (i\in N)$, N_i 的入度邻居节点 U_i^{In} 及关联邻居节点 $U_i (i\in N)$, 聚合次数 τ, 阈值 σ, N_i 对所有对象的观测值 $x_m^i (i\in N, m\in M)$

输出: 所有参与者的权值 $W_i (i\in N)$

1. FOR i=1, 2, 3, ⋯, N do
/*参与者本地生成随机数 b_i^1 */
2. b_i^1=Pseudorandom(i);
3. /* 将 b_i^1、sk_i^1 切片加密后发送给 S 转发给相应出度邻居 N_j */
4. $(Q_{i,j}(\mathrm{S}_{\mathrm{slice}}(t, L, b_i^1)), Q_{i,j}(\mathrm{S}_{\mathrm{slice}}(t, L, sk_i^1)))\rightarrow U_j$
5. END FOR
6. /* 如果为首次迭代, 则真值为随机值, 否则为上一轮的真值*/
7. If τ=0 x_m^*=math.random();
8. FOR i=1, 2, 3, ⋯, N do
9. $\overrightarrow{b_i^1} = PRG(b_i^1)$, $\overrightarrow{m_{i,j}} = PRG(r_{i,j})$; /* 根据 b_i^1 和 $r_{i,j}$ 分别生成两个随机数*/
10. FOR m = 1, 2, 3, ⋯, M *do*
11. $N_i \rightarrow x_m^i$ /* 参与者 N_i 获取所有对象的观测值*/
12. N_i 计算 V_i=(A_i, B_i, P_i, Q_i), ($msg_{i,j}$, $Sig_{i,j}^{msg}$)
13. /*将混淆后的数据及用于验证的数据发送给服务器*/
14. $(\overrightarrow{Dis_i}, V_i, (msg_{i,j}, Sig_{i,j}^{msg}))\rightarrow$S;
15. END FOR
16. END FOR
17. S 生成 U_D
/*U_D 是成功发送混淆数据到服务器的参与者集合*/
18. FOR i=1, 2, 3, ⋯, N do
19. $(U_i^{Won}, U_i^{Woff}, (rsp_{i,j}, Sig_{i,j}^{rsp}))\rightarrow N_i$ /*发送所有 N_i 的在线和掉线入度邻居*/
20. $(Y_i^{b_i^1}, Y_i^s)\rightarrow U_i^{Out}$ /* 所有 N_i 发送对应切片给出度邻居节点*/
21. END FOR
22. S 计算 *proof*, $\sum_{i=1}^{i=|U_D|} Dis_i$
23. FOR i=1, 2, 3, ⋯, N DO
24. /* S 发送聚合结果及其证明给所有参与者, 参与者对聚合结果进行验证*/
25. N_i verify ($\sum_{i=1}^{i=|U_D|} Dis_i$, *proof*)→ pass
26. IF pass = 0 BREAK; /*验证不通过, 终止本

协议, 退出*/

27. IF pass = 1 N_i 计算 W_i;
/*验证通过, 参与者 N_i 根据公式(2)计算权值*/

28. END FOR

4.4 安全可验证的真值评估

权值更新后进行真值评估并且所有参与者通过对服务器 S 的权值和与加权值和的结果进行验证, 最终验证比对参与者计算的真值与服务器发送的真值是否一致来对真值结果实验可验证: 所有参与者对权值以及加权值混淆后再发送给云服务器, 云服务器得到所有参与者的权值及加权值的和进行数据聚合, 并生成聚合结果证明, 最后计算出对象的真值。云服务器将权值和、加权值的和, 以及聚合结果证明和真值发送给参与者进行验证, 参与者验证通过后, 完成本轮真值更新, 如达到收敛条件, 协议完成; 否则进入下一轮真值发现。

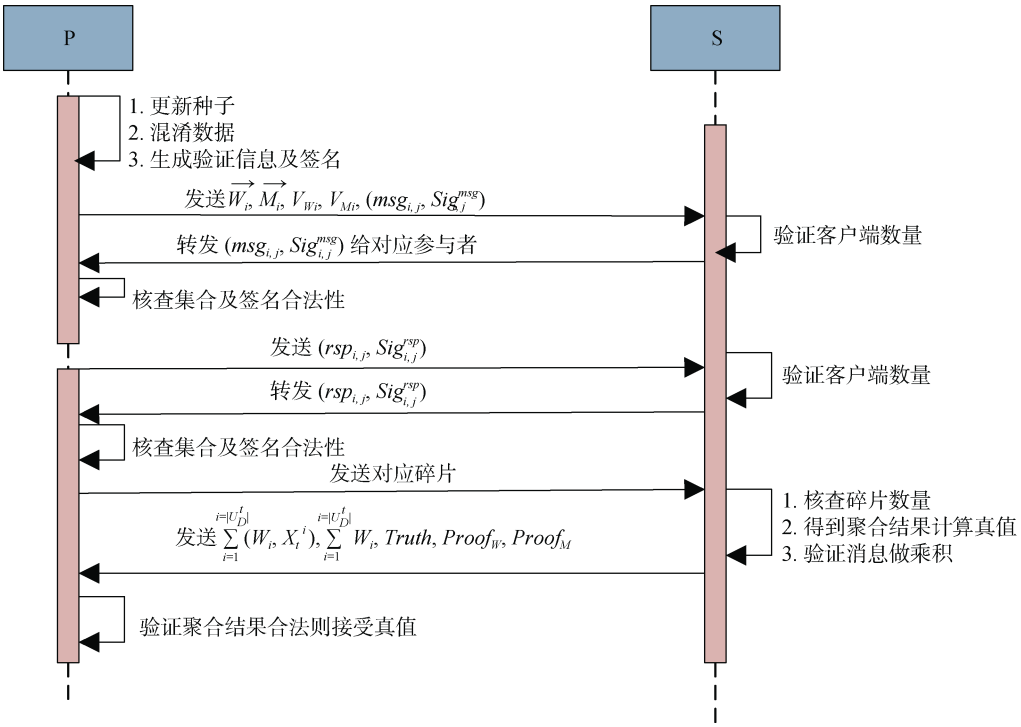


图 4 安全可验证的真值更新

Figure 4 Security and verifiable truth update

Step1: 更新切片数据

(1) U_D 集合参与者 N_i 更新随机数种子 b_i^1 为 b_i^2 并进行切分加密发送给 S, 具体如下: $Y_i^{b_i^2} = \{y_{i,1}^{b_i^2}, y_{i,2}^{b_i^2}, ..., y_{i,L}^{b_i^2}\} = S_{slice}(t, L, b_i^2)$ $ET_{i,j} = Enc(Q_{i,j}, (i \| j \| Y_i^{b_i^2}))$。

(2) S 检查参与者切片数量是否大于阈值 t, 并记录发送切片参与者的 ID 为集合 U_E^t ($U_E^t \subseteq U_D$)。S 将更新的切片转发给集合 U_E^t 中对应的出度邻居节点 N_j, 出度邻居节点解密切片, 得到 $Y_i^{b_i^2}$ 的共享切片。

Step2: 生成扰动数据

U_E^t 集合参与者 N_i 计算权值 W_i 与对象观测值的乘积 $\overrightarrow{M_i}$, 并 N 利用伪随机函数生成 $\overrightarrow{b_i^2} = PRG(b_i^2)$ 将 W_i 和 $W_i \times X_m^i$ 进行扰动。

$$\overrightarrow{W_i} = W_i + \overrightarrow{b_i^2} - \sum_{0<j<i(j\in U_i)} \overrightarrow{m_{i,j}} + \sum_{i<j\leqslant N(j\in U_i)} \overrightarrow{m_{i,j}} \quad (8)$$

$$\overrightarrow{M_i} = W_i \times X_m^i + \overrightarrow{b_i^2} - \sum_{0<j<i(j\in U_i)} \overrightarrow{m_{i,j}} + \sum_{i<j\leqslant N(j\in U_i)} \overrightarrow{m_{i,j}} \quad (9)$$

Step3: 生成用于验证的数据及签名信息

(1) 同理 4.3 节 step4(1), U_E^t 集合参与者 N_i 计算 $V_{W_i} = (A_{W_i}, B_{W_i}, P_{W_i}, Q_{W_i})$ 和 $V_{M_i} = (A_{M_i}, B_{M_i}, P_{M_i}, Q_{M_i})$, 具体计算如下:

$$HF(W_i) = (A_{W_i}, B_{W_i}) = (g^{HF_{\alpha,\beta}(W_i)}, h^{HF_{\alpha,\beta}(W_i)})$$

$$PRG_{K1}(i) = (\lambda \times i, \eta \times i);\ PRG_{K2}(\tau) = (\sigma, \delta)$$

$$P_{W_i} = (C_{W_i} A_{W_i}^{-1})^{1/d} = (g^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(W_i)})^{1/d}$$

$$Q_{W_i} = (D_{W_i} B_{W_i}^{-1})^{1/d} = (h^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(W_i)})^{1/d}$$

$$HF(M_i) = (A_{M_i}, B_{M_i}) = (g^{HF_{\alpha,\beta}(M_i)}, h^{HF_{\alpha,\beta}(M_i)})$$

$$PRG_{K1}(i)=(\lambda\times i,\eta\times i);\ PRG_{K2}(\tau)=(\sigma,\delta)$$

$$P_{M_i}=(C_{M_i}A_{M_i}^{-1})^{1/d}=(g^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(M_i)})^{1/d}$$

$$Q_{M_i}=(D_{M_i}B_{M_i}^{-1})^{1/d}=(h^{\lambda\times i\times\sigma+\eta\times i\times\delta-HF_{\alpha,\beta}(M_i)})^{1/d}$$

生成消息 $msg_{i,j}=(i,j(j\in U_i\subseteq U_E^t),"involved")$ 该消息用于表明参与者 N_i 已经将与关联邻居协商的随机数加入到混淆数据中, 并对该消息进行签名, 得到 $Sig_{i,j}^{msg}$。

(2) 每个参与者 N_i 将扰动数据 $\overrightarrow{W_i}$, $\overrightarrow{M_i}$ 和数据向量(V_{W_i}, V_{M_i}, ($msg_{i,j}$, $Sig_{i,j}^{msg}$))一同发送给服务器 S。

Step4: 聚合数据

(1) S 记录所有收到发送混淆数据的参与者的 ID, 记为集合 U_D^t, 同理 4.3 节 step5(1)~(6), S 恢复 $\overrightarrow{b_i^2}$ 和 $\overrightarrow{m_{i,j}}$ 并得到权值和加权值的聚合结果, 并根据公式(3)得到对象 m 的真值并发送给 S 所有参与者, 具体计算如下:

$$\sum_{i=1}^{i=|U_D^t|}W_i=\sum_{i=1}^{i=|U_D^t|}(\overrightarrow{W_i}-\overrightarrow{b_i^2}+\sum_{0<j<i(j\in U_i\cap(U_E^t\setminus U_D^t))}\overrightarrow{m_{i,j}}-\sum_{i<j\leqslant N(j\in U_i\cap(U_E^t\setminus U_D^t))}\overrightarrow{m_{i,j}}) \tag{10}$$

$$\sum_{i=1}^{i=|U_D^t|}W_i\times X_m^i=\sum_{i=1}^{i=|U_D^t|}(\overrightarrow{M_i}-\overrightarrow{b_i^2}+\sum_{0<j<i(j\in U_i\cap(U_E^t\setminus U_D^t))}\overrightarrow{m_{i,j}}-\sum_{i<j\leqslant N(j\in U_i\cap(U_E^t\setminus U_D^t))}\overrightarrow{m_{i,j}}) \tag{11}$$

$$Truth_m=\frac{\sum_{i=1}^{i=|U_D^t|}W_i\times X_m^i}{\sum_{i=1}^{i=|U_D^t|}W_i} \tag{12}$$

(2) 同理 4.3 节 step5(7), S 将参与者 N_i 发送的用于验证的信息 $V_{W_i}=(A_{W_i},B_{W_i},P_{W_i},Q_{W_i})$ 和 $V_{M_i}=(A_{M_i},B_{M_i},P_{M_i},Q_{M_i})$ 做乘积生成 $proof_w$、$proof_M$。具体计算如下:

$$proof_w=\{A_w=\prod_{i=1}^{i=|U_D^t|}A_{W_i},B_w=\prod_{i=1}^{i=|U_D^t|}B_{W_i},P_w=\prod_{i=1}^{i=|U_D^t|}P_{W_i},Q_w=\prod_{i=1}^{i=|U_D^t|}Q_{W_i}\}$$

$$proof_M=\{A_M=\prod_{i=1}^{i=|U_D^t|}A_{M_i},B_M=\prod_{i=1}^{i=|U_D^t|}B_{M_i},P_M=\prod_{i=1}^{i=|U_D^t|}P_{M_i},Q_M=\prod_{i=1}^{i=|U_D^t|}Q_{M_i}\}$$

Step5: 验证聚合结果并进行权值更新

(1) 同理 4.3 节 step6(1)～(3)参与者 N_i 计算 $\xi=\sum_{i=1}^{i=|U_D^t|}(\lambda\times i\times\sigma+\eta\times i\times\delta)$ 以及 $\gamma=e(g,h)^{\xi}$, 并在两个双线性群 G1 和 G2 上对 $\sum_{i=1}^{i=|U_D^t|}W_i$ 进行同态哈希, 得到 A'_{iw} 和 B'_{iw}, 参与者依次验证 $\gamma_{iw}=e(A_{iw},h)\times e(P_{iw},h)^d$, $e(P_{iw},h)=e(g,Q_{iw})$, $e(A_{iw},h)=e(g,B_{iw})$ 和 $(A_{iw},B_{iw})=(A'_{iw},B'_{iw})$ 是否成立, 成立则证明服务器聚合结果无误, 接受聚合结果 $\sum_{i=1}^{i=|U_D^t|}W_i$; 如果不成立, 则终止协议。

(2) 同理 4.3 节 step6(2)~(3), 参与者 N_i 在本地将 $\sum_{i=1}^{i=|U_D^t|}(W_i\times X_m^i)$ 做同态哈希, 在两个双线性群 G1 和 G2 上分别得到 A'_{iD} 和 B'_{iD}, 进一步验证是否 $\gamma_{iD}=e(A_{iD},h)\times e(P_{iD},h)^d$, $e(P_{iD},h)=e(g,Q_{iD})$, $e(A_{iD},h)=e(g,B_{iD})$, $(A_{iD},B_{iD})=(A'_{iw},B'_{iD})$, 如果有验证不通过, 立即中止协议, 否则所有的参与者 N_i 接受 S 的聚合结果 $\sum_{i=1}^{i=|U_D^t|}(W_i\times X_m^i)$, 并根据公式(3)计算真值, 将最终计算的真值和 S 发送的真值进行对比判断是否一致, 一致则证明真值结果无误, 不一致则立即终止协议。

此时, 本轮真值发现的迭代已经完成, S 将本轮迭代的真值与上一轮迭代的真值进行误差比较, 判断是否达到收敛条件。如果达到收敛条件, 则中止协议; 否则继续进行下一轮真值发现的迭代。

5 性能分析及实验

为了评估 EVSTD 的性能, 本节通过真实的数据源[31]对 EVSTD、PPTD 和 EPTD 方案进行对比, 主要包括准确度、收敛性、计算开销和通信开销等指标。为了更准确地验证实验结果, 每种实验均测试 10 次, 然后取平均值。

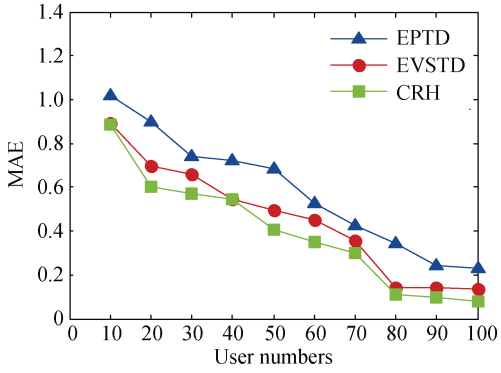
5.1 准确度

本实验通过计算绝对误差的平均数(MAE)和相对误差的平均数(RMSE)来衡量 CRH、EPTD 和 EVSTD 算法的精确度。RMSE 和 MAE 计算为公式(13)和(14)。

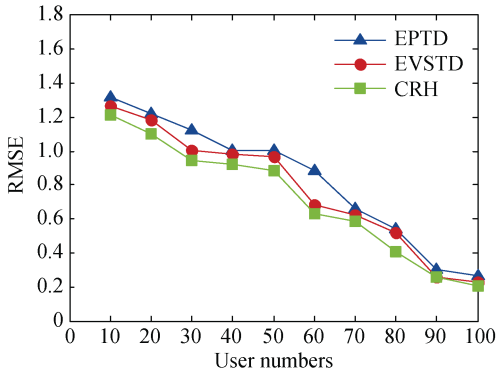
$$MAE=\frac{\sum_{i=1}^{N}\sum_{m=1}^{M}(x_m^*-x_m^i)}{N\times M} \tag{13}$$

$$RMSE=\sqrt[2]{\frac{\sum_{i=1}^{N}\sum_{m=1}^{M}(x_m^*-x_m^i)^2}{M\times N}} \tag{14}$$

图 5 给出了观测对象个数 M=30 时, 不同参与者数量情况下 CRH、EPTD 和 EVSTD 的误差率。由于感知数据需要以整数的形式进行计算, 因此在计算 MAE 和 RMSE 时, 需引进参数 L 对数据以四舍五入法取近似值, 设置 L=10^6。由于 EVSTD 和 EPTD 都增加扰动数据, 所以 CRH 方法准确度最高, 但 CRH 没有考虑数据隐私性。EPTD 方案中, 每个参与者都需要加入 N–1 个混淆数, 而 EVSTD 只需要加入 log$N$ 数量级的混淆数, 所以 EVSTD 的 MAE 和 RMSE 都好于 EPTD。



(a) 绝对误差MAE
(a) Absolute error MAE



(b) 相对误差RMSE
(b) Relative error RMSE

图 5 不同参与者数量下的绝对误差和相对误差对比
Figure 5 Comparison of absolute error and relative error under different number of participants

5.2 收敛性

真值发现的收敛性取决于真值变化小于阈值时迭代的次数。为了验证 EVSTD 的收敛性, 本实验通过设置 5 个不同的初始化真值 x_m^* 来进行验证。其他参数设置为用户数 N=50, 阈值 σ=0.001。图 6 实验结果可知, 在不同初始真值的情况下, EVSTD 在 2 次迭代后都能得到相对平稳的真值, 收敛速度快。因此, 在 EVSTD 算法中, 初始化的真值对 EVSTD 的真值发现结果基本没有影响, 可忽略。

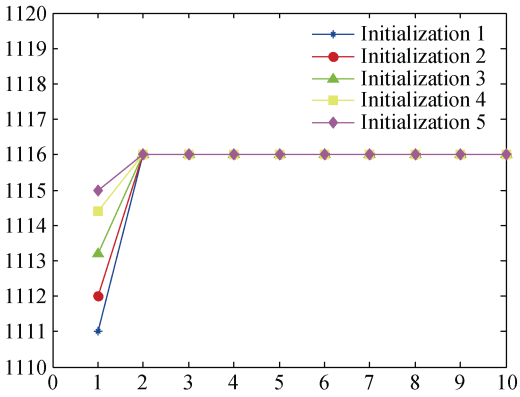


图 6 收敛性比较
Figure 6 Convergence comparison

5.3 通信开销

EVSTD 中, 通信开销主要包括安全的权值更新阶段和安全的真值评估阶段。安全的权值更新阶段, 每个参与者向 log$N$ 个出度邻居发送密钥切片和加密种子切片, 给服务器发送混淆数据和验证信息。服务器端需要向 N 个参与者发送密钥切片和加密种子切片、聚合结果和验证信息。假设切片数据位长为 a, 数据位长 b, 验证信息位长 c, 所以每轮安全权值更新阶段参与者的通信开销为 $2\times a\times\log N+b+c$, 服务器的通信开销为 $2\times a\times N\times\log N+(b+c)\times N$。同理, EPTD 中, 每个参与者需要向 N–1 个参与者发送密钥切片和加密种子切片, 给服务器发送混淆数据, 即参与者通信开销为 $2\times a\times N+b$。服务器端需要向 N 个参与者发送密钥切片和加密种子切片、聚合结果, 即服务器的通信开销为 $2\times a\times N^2+b\times N$。安全的真值评估阶段, EVSTD 中, 每个参与者需要向 log$N$ 阶个出度邻居发送加密种子切片, 给服务器发送混淆的权值及验证信息、混淆的加权值及验证信息, 参与者的通信开销为 $a\times\log N+2\times b+2\times c$。服务器端向 N 个参与者转发长度为加密种子切片、聚合结果数据和验证信息, 服务器的通信开销为 $a\times N\times\log N+(2\times b+2\times c)\times N$; 在 EPTD 中, 每个参与者需要向 N–1 个参与者发送加密种子切片, 向服务器发送混淆数据, 参与者通信开销为 $a\times N+2\times b$。服务器向 N 个参与者发送加密种子切片和聚合结果, 服务器的通信开销为 $a\times N^2+2\times b\times N$。假设迭代次数为 d, EVSTD 和 EPTD 的通信开销如表 1 所示。由表 1 可知 EVSTD 的通信开销小于 EPTD。

表 1 通信开销

Table 1 Communication overhead

开销/方案	EVSTD		EPTD	
	参与者	服务器	参与者	服务器
权值更新阶段	$(2\times a\times\log N+b+c)\times d$	$2\times a\times N\times\log N+(b+c)\times N$	$(2\times a\times N+b)\times d$	$2\times a\times N^2+b\times N$
真值评估阶段	$(a\times\log N+2\times b+2\times c)\times d$	$a\times N\times\log N+(2\times b+2\times c)\times N$	$(a\times N+2\times b)\times d$	$a\times N^2+2\times b\times N$
总的通信开销	$3\times(a\times\log N+b+c)\times d$	$3\times(a\times N\times\log N+(b+c)\times N)$	$3\times(a\times N+b)\times d$	$3\times(a\times N^2+b\times N)$

5.4 计算开销

在相同的软硬件环境下, 对不同数量的对象进行真值发现, 并通过运行时间来评估在权值更新和真值评估阶段的计算开销, 同时对权值更新和真值评估阶段的加解密计算进行详细的分析说明。如图 7 所示, EVSTD 的总运行时间随着观测对象的增加增加。

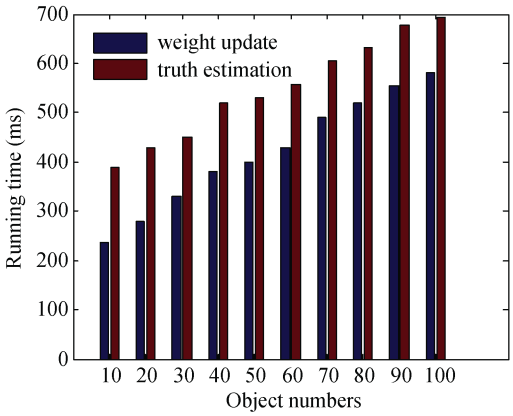
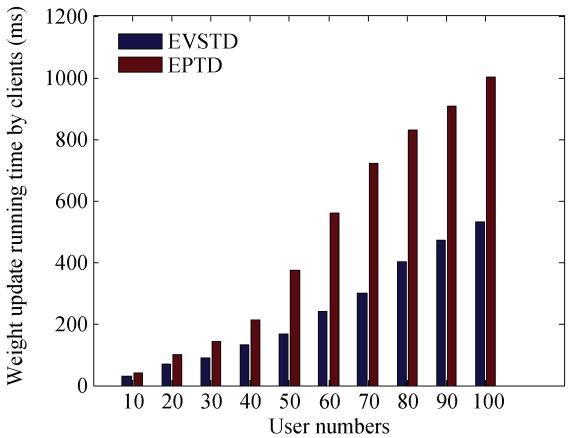


图 7 对象个数不同的运行时间比较

Figure 7 Running time comparison of different objects

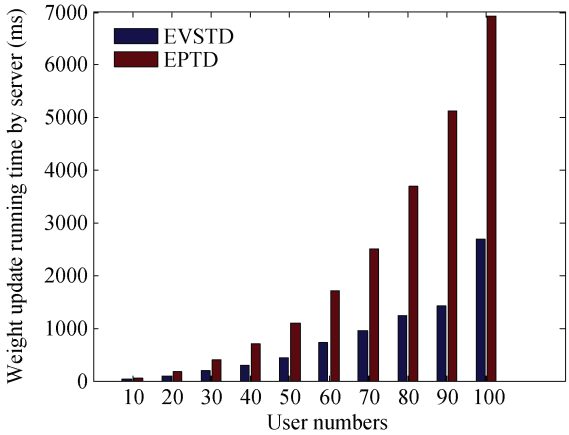
EVSTD 中, 计算开销主要包括安全的权值更新阶段和安全的真值评估阶段。安全的权值更新阶段, 每个参与者分别进行 log$N$ 次加解密运算运算, 混淆数据和验证信息。服务器计算聚合结果及验证信息证乘积。假设加解密的计算复杂度为 e, 验证信息的计算复杂度为 v, 混淆数据的计算复杂度为 f, 聚合结果的计算复杂度为 h, 聚合结果证明的计算复杂度为 v', 所以每轮安全权值更新阶段参与者的计算开销为 $2\times e\times\log N+f+v$, 服务器的计算开销为 $h+v'$。在 EPTD 中, 每个参与者分别进行 N 次加解密运算运算, 计算混淆数据。服务器计算聚合结果。假设聚合结果的计算复杂度为 h'。则每轮安全真值评估阶段参与者的计算开销为 $2\times e\times N+f$, 服务器的计算开销为 h'。EVSTD 和 EPTD 在权值更新阶段的计算开销如图 8(a)和(b)所示。由图 8 可知 EVSTD 的在权值更新阶段计算开销小于 EPTD。

真值评估阶段由于需要安全聚合权值和加权值, 计算开销是权值更新阶段 2 倍。即在 EVSTD 中, 每轮安全真值评估阶段参与者的计算开销为 $2\times(2\times e\times\log N+f+v)$, 服务器的计算开销为 $2\times(h+v')$。在 EPTD 中, 每轮安全真值评估阶段参与者的计算开销为 $2\times(2\times e\times N+f)$, 服务器的计算开销为 $2\times h'$。EVSTD 和 EPTD 在真值评估阶段的计算开销如图 9(a)和(b)所示。由图 9 可知 EVSTD 的在真值评估阶段计算开销小于 EPTD。



(a) 参与者权值更新阶段运行时间

(a) Running time of participant weight update

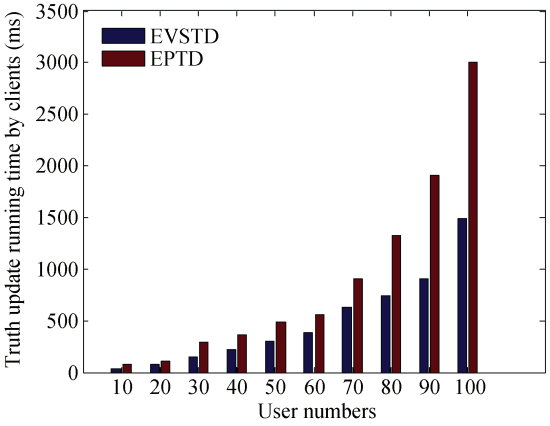


(b) 服务器端权值更新阶段运行时间

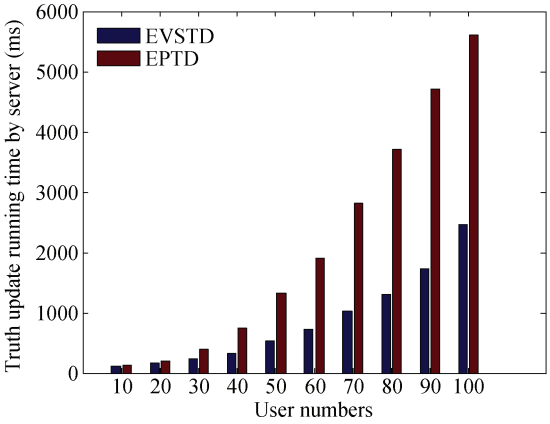
(b) Running time of weight update on Server

图 8 权值更新阶段运行时间比较

Figure 8 Running time comparison of weight update stage



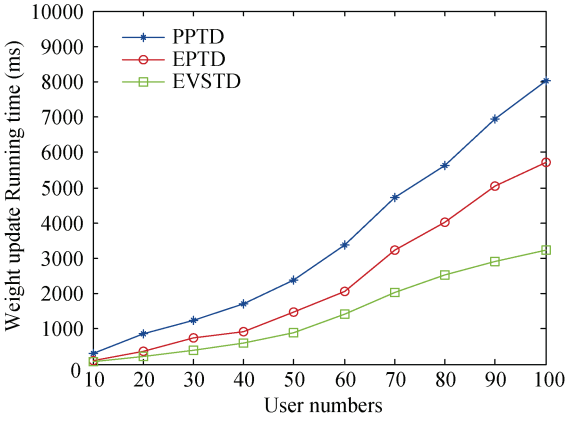
(a) 参与者真值评估阶段运行时间
(a) Running time of participant truth evaluation



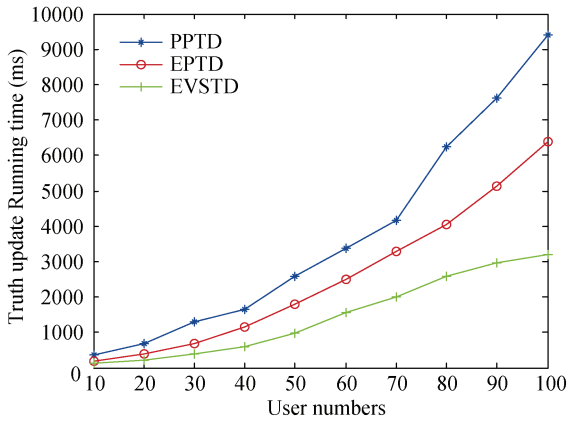
(b) 服务器端真值评估阶段运行时间
(b) Server truth evaluation stage running time

图 9　真值评估阶段运行时间比较

Figure 9　Running time comparison of truth evaluation stage



(a) 权值更新阶段
(a) Weight update stage



(b) 真值评估阶段
(b) Truth update stage

图 10　不同用户数量下的运行时间比对比

Figure 10　Comparison of runtime ratio under different number of users

为了进一步验证 EVSTD 在计算开销方面的性能, 图 10 给出了在参与者数量不断增加的情况下, PPTD、EPTD 和 EVSTD 权值更新和真值评估运行时间的对比。由于 PPTD 使用 Paillier 密码系统进行同态加密, 涉及到大尺寸的密文计算。EPTD 需要网络中所有参与者两两进行密钥协商及加解密运算, 大大增加了参与者与服务器的计算开销。EVSTD 由于进行 L 邻居节点动态选择, 从而大大降低了网络的计算开销。

5.5　安全性分析

EVSTD 是一种高效可验证的安全真值发现方法, 通过动态选择 L 邻居节点进行秘密数据分享保证算法的高效, 利用数据混淆、密钥协商和秘密共享来保护真值发现中参与者的隐私数据不被窃听及篡改保证信息的安全性。本节将对 EVSTD 算法的安全性进行分析, 在 MCS 系统中, 主要的安全威胁来自云服务器以及参与者本身。本节从外部攻击和内部攻击以及验证的正确性来说明 EVSTD 的安全性。

(1) 外部攻击: 外部攻击主要为窃听攻击, 在权值更新阶段, 假设所有的攻击者都可以进行网络监听。参与者发送的数据主要包括隐私数据和切片数据, 隐私数据是添加了扰动数据后发送。在权值更新阶段, 参与者 N_i 发送的扰动数据为 $\overrightarrow{Dis_i}=Dis_i+\overrightarrow{b_i^l}-\sum_{0<j<i\ (j\in U_i)}\overrightarrow{m_{i,j}}+\sum_{i<j\leqslant N\ (j\in U_i)}\overrightarrow{m_{i,j}}$, 其中 Dis_i 是参与者 N_i 的隐私数据, 所以攻击者不能通过截获的混淆后数据推导出隐私数据; 同时, 参与者发送加密的切片数据 $E_{i,j}=Enc(Q_{i,j},(i\|j\|Y_i^{b_i^1}\|Y_i^s))$, 攻击者只能得到切片的密文而不能获得原文数据, 即使攻击者能够破解密文切片数据 $E_{i,j}$, 则至少破解 t 个切片才能得到参与者私钥 sk_i^1 和本地随机数种子 b_i^1。此外, 服务器只能获得所有参与者发送的扰动数据 $\overrightarrow{Dis_i}$, 且最终聚合结果是通过扰动数据进行聚合得到, 所以攻击者不能通过窃听服务器推导出每个参与者的

隐私数据。真值评估阶段同理, 因此, EVSTD 能够防外部攻击。

(2) 内部攻击: 内部攻击主要分为共谋攻击和篡改攻击。假设 MCS 系统中存在多个恶意的节点, 且进行共谋攻击。由 EVSTD 算法可知, 在权值更新阶段, 当攻击者想要得到某个参与者的隐私数据, 必须得到该参与者私钥 sk_i^1 和本地随机数种子 b_i^1, 根据 Shamir 秘密共享算法, 私钥 sk_i^1 和随机数种子 b_i^1 分别至少需要有 t 个切片才能恢复, 所以至少需要有 t 个恶意节点被该参与者选择选为出度邻居节点, 这些恶意节点共谋才能推导出该参与者的隐私数据。当某个参与者掉线, 则增加其关联邻居节点通信开销, 且更容易成为攻击者攻击目标。所以在 EVSTD 中, 每个参与者选择 L 个出度邻居节点, 且其入度邻居节点不超过 L 个, 即每个参与者的关联邻居节点不超过 $2\times L$ 个($4\times L$ 个公钥)。参与者在进行 L 邻居节点选择后收到云服务器发送其关联邻居节点集合, 参与者检查收到的公钥数量是否大于 $4\times L$ 个, 如果大于, 则存在多个恶意节点试图共同选择该参与者作为出度邻居节点的可能, 为了降低该参与者的隐私泄露风险, 其入度邻居节点重新进行选择, 所以攻击者很难通过足够多的恶意节点共谋获得某参与者的数据。EVSTD 中, 服务器不能获得单个参与者的隐私数据, 且在聚合隐私数据后, 服务器需要提交聚合结果证明 *Proof*。参与者通过 *Proof*、双线性对以及同态哈希算法的性质验证聚合结果的正确性。若验证不通过, 则说明服务器篡改聚合结果。真值评估阶段同理, 所以 EVSTD 能够抗内部攻击。

5.6 方案比较

Li 等人[14]提出的方案解决了传统真值发现方法中每个参与者可靠度相同而使得最终计算结果不准确的问题, 通过迭代更新参与者的可靠度(权值), 并迭代评估对象的真值直到达到收敛条件为止。但该方案忽略了每个参与者的隐私保护问题。Miao 等人[19]提出的方案解决了在保证真值结果准确性的同时利用基于 Paillier 同态加密来保证参与者的隐私, 但是该方案中所有用户都要进行加解密运算通信开销过大。在 Li 等人[14]方案的基础上, Xu 等人[22]利用秘密数据分享实现了数据的隐私保护, 该方案需要参与者两两进行通信, 网络的通信开销和计算开销较大, 且到服务器安全性没有考虑。

本文方案为保证在计算观测对象的真值的基础上高效保护参与者的隐私数据, 在对隐私数据进行双掩码保护的过程中, 让每个参与者动态选择 L 邻居节点进行秘密切片及秘钥的切片分享, 并在之后的秘密数据的恢复过程中只需邻居节点参与恢复, 大大降低了网络的通信开销和计算开销。最后利用双线性对计算结果进行验证, 保证了最终结果没有被篡改, 相比上述三个方案, 本方案在安全性及效率上做了大量优化。如表 2 所示。

表 2 方案比较

Table 2 Scheme comparison

	准确性	隐私性	用户动态性	高效性	验证性
CRH[14]	√	×	×	×	×
PPTD[19]	√	√	√	×	×
EPTD[22]	√	√	√	×	×
EVSTD	√	√	√	√	√

6 总结

本文针对移动群智感知中具有隐私保护的真值发现方法面临的计算开销及通信开销大和服务器节点作恶问题提出了一个安全高效可验证的真值发现方法 EVSTD, 利用 L 邻居节点的动态选择降低整个网络的通信开销及计算开销, 并且每个参与者可以对云服务器的聚合结果可验证, 从而保证最终真值的可靠性。最终将 EVSTD 与 PPTD, EPTD 相比表现出了较好的性能优势, 大幅降低了真值发现过程中的计算开销和运行时间。在未来的工作中, 还将进一步探索如何更加有效的降低服务器端的通信开销。

参考文献

[1] Ren K, Wang Q, Wang C, et al. The Security of Autonomous Driving: Threats, Defenses, and Future Directions[J]. *Proceedings of the IEEE*, 2020, 108(2): 357-372.

[2] Pius Owoh N, Mahinderjit Singh M. SenseCrypt: A Security Framework for Mobile Crowd Sensing Applications[J]. *Sensors*, 2020, 20(11): 3280.

[3] Yin H C, Yu Z W, Wang L, et al. ISIATasker: Task Allocation for Instant-Sensing%DF%9DInstant-Actuation Mobile Crowdsensing[J]. *IEEE Internet of Things Journal*, 2022, 9(5): 3158-3173.

[4] Zhang X C, Lu R X, Shao J, et al. FedSky: An Efficient and Privacy-Preserving Scheme for Federated Mobile Crowdsensing[J]. *IEEE Internet of Things Journal*, 2022, 9(7): 5344-5356.

[5] Ji S G, Zheng Y, Wang Z Y, et al. Look-Ahead Search and Voting-Based Dynamic Participant Recruitment in Mobile Crowd Sensing[J]. *Chinese Journal of Computers*, 2021, 44(10): 1998-2015.
(纪圣塘, 郑宇, 王诏远, 等. 基于前向搜索和投票的移动群智感知动态用户招募方法[J]. *计算机学报*, 2021, 44(10): 1998-2015.)

[6] Huang H J, Yang J, Huang H, et al. Deep Learning for Su-

per-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(9): 8549-8560.

[7] Zhao H G, Ma X F, Shi C. Information Interaction in Wireless Sensor Networks Based on Socially Aware Computing[C]. *China Conference Wireless Sensor Networks*, 2014: 71-81.

[8] Yan H, Peng G J, Luo Y, et al. Survey on Smart Home Attack and Defense Methods[J]. *Journal of Cyber Security*, 2021, 6(4): 1-27. (严寒, 彭国军, 罗元, 等. 智能家居攻击与防御方法综述[J]. *信息安全学报*, 2021, 6(4): 1-27.)

[9] Zhang X L, Yang Z, Liu Y H. Vehicle-Based Bi-Objective Crowdsourcing[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2018, 19(10): 3420-3428.

[10] Chen J X, Liu Y N, Xiang Y, et al. RPPTD: Robust Privacy-Preserving Truth Discovery Scheme[J]. *IEEE Systems Journal*, 2022, 16(3): 4525-4531.

[11] Ouyang R W, Kaplan L M, Toniolo A, et al. Parallel and Streaming Truth Discovery in Large-Scale Quantitative Crowdsourcing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2016, 27(10): 2984-2997.

[12] Jin H M, Su L, Nahrstedt K. Theseus: Incentivizing Truth Discovery in Mobile Crowd Sensing Systems[C]. *The 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2017: 1-10.

[13] Zhang D Y, Badilla J, Zhang Y, et al. Towards Reliable Missing Truth Discovery in Online Social Media Sensing Applications[C]. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2018: 143-150.

[14] Li Q, Li Y L, Gao J, et al. Resolving Conflicts in Heterogeneous Data by Truth Discovery and Source Reliability Estimation[C]. *The 2014 ACM SIGMOD International Conference on Management of Data*, 2014: 1187-1198.

[15] Li Q, Li Y L, Gao J, et al. A Confidence-Aware Approach for Truth Discovery on Long-Tail Data[J]. *Proceedings of the VLDB Endowment*, 2014, 8(4): 425-436.

[16] Yang Y, Bai Q, Liu Q. A Probabilistic Model for Truth Discovery with Object Correlations[J]. *Knowledge-Based Systems*, 2019, 165: 360-373.

[17] Zhang H, Shen F, Jiang S H, et al. Ensemble Weighted Soft Voting Truth Inference Method for Crowdsourcing[J]. *Journal of Tsinghua University* (*Science and Technology*), 2022, 62(2): 347-354. (张桦, 沈菲, 蒋世豪, 等. 集成加权软投票的众包真值推理方法[J]. *清华大学学报(自然科学版)*, 2022, 62(2): 347-354.)

[18] Xiao H P, Gao J, Li Q, et al. Towards Confidence Interval Estimation in Truth Discovery[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2019, 31(3): 575-588.

[19] Miao C L, Jiang W J, Su L, et al. Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems[C]. *The 13th ACM Conference on Embedded Networked Sensor Systems*, 2015: 183-196.

[20] Miao Chenglin, Jiang Wenjun, Su Lu, et al. Privacy-Preserving Truth Discovery in Crowd Sensing Systems[J]. *ACM Transactions on Sensor Networks*, 2019, 15(1):9-32.

[21] Zheng Y F, Duan H Y, Wang C. Learning the Truth Privately and Confidently: Encrypted Confidence-Aware Truth Discovery in Mobile Crowdsensing[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(10): 2475-2489.

[22] Xu G W, Li H W, Liu S, et al. Efficient and Privacy-Preserving Truth Discovery in Mobile Crowd Sensing Systems[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(4): 3854-3865.

[23] Shamir A. How to Share a Secret[J]. *Communications of the ACM*, 1979, 22(11): 612-613.

[24] Yin X X, Han J W, Yu P S. Truth Discovery with Multiple Conflicting Information Providers on the Web[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2008, 20(6): 796-808.

[25] Dong X L, Berti-Equille L, Srivastava D. Truth Discovery and Copying Detection in a Dynamic World[J]. *Proceedings of the VLDB Endowment*, 2009, 2(1): 562-573.

[26] Miao C L, Su L, Jiang W J, et al. A Lightweight Privacy- Preserving Truth Discovery Framework for Mobile Crowd Sensing Systems[C]. *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017: 1-9.

[27] Li H W, Lu R X, Zhou L, et al. An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid[J]. *IEEE Systems Journal*, 2014, 8(2): 655-663.

[28] Fiore D, Gennaro R, Pastro V. Efficiently Verifiable Computation on Encrypted Data[C]. *The 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014: 844-855.

[29] Boneh D, Boyen X, Goh E J. Hierarchical Identity Based Encryption with Constant Size Ciphertext[M]. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 440-456.

[30] Jiang Y, Zhao B W, Tang S H, et al. A Verifiable and Privacy-Preserving Multidimensional Data Aggregation Scheme in Mobile Crowdsensing[J]. *Transactions on Emerging Telecommunications Technologies*, 2021, 32(5): e4008.

[31] Li X A, Dong X L, Lyons K, et al. Truth Finding on the Deep Web[J]. *Proceedings of the VLDB Endowment*, 2012, 6(2): 97-108.

王涛春 于 2016 年在南京航空航天大学计算机应用技术专业获得博士学位。现任安徽师范大学教授。主要研究领域为隐私保护、物联网与群智感知等。Email: wangtc@ahnu.edu.cn

张晨露 于 2022 年在安徽师范大学软件工程专业获得硕士学位。主要研究领域为隐私保护、群智感知等。Email: 987903568@qq.com

蔡松健 于 2022 年在安徽师范大学软件工程专业获得硕士学位。主要研究领域为区块链、区块链跨链。Email: 1071285741@qq.com

陈付龙 于 2011 年在西北工业大学获得博士学位。现任安徽师范大学教授。主要研究方向嵌入式与普计算、信息物理融合系统、高性能计算机体系结构、物联网安全等。Email: long005@ahnu.edu.cn

沈慧敏 于 2023 年在安徽师范大学计算机科学与技术获得硕士学位。主要研究领域为区块链。Email: shenhuimin@ahnu.edu.cn

谢冬 于 2017 年在北京邮电大学密码学专业获得博士学位。现任安徽师范大学副教授。主要研究领域为应用密码学、图像安全。Email: xiedong@ahnu.edu.cn