# Privacy Protection in 5G Positioning and Location-based Services Based on SGX

ZHENG YAN, The State Key Lab of Integrated Services Networks, School of Cyber Engineering, Xidian University, China and The Department of Communications and Networking, Aalto University, Finland
XINREN QIAN, The State Key Lab of Integrated Services Networks, School of Cyber Engineering, Xidian University, China
SHUSHU LIU, The Department of Communications and Networking, Aalto University, Finland
ROBERT DENG, The School of Information Systems, Singapore Management University, Singapore

As the sensitivity of position, the privacy protection in both 5G positioning and its further application in location-based services (LBSs) has been paid special attention and studied. Solutions based on k-anonymity, homomorphic encryption, and secure multi-party computation have been proposed. However, these solutions either require a trusted third party or incur heavy overheads. Besides, there still lacks an integrated solution that can protect privacy for both positioning and LBS provision. Based on Intel SGX, this article proposes a novel light-weight scheme that can protect privacy in both 5G positioning and its further applications in LBS provision in an integrated way. Through secret sharing, the proposed scheme can also support multiple location-based service providers without frequent key exchange. We seriously analyze the security of our scheme. Based on scheme implementation, its efficiency is proved through the performance evaluation conducted over a real-world database.



CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; *Hardware attacks and countermeasures*; *Key management;*

Additional Key Words and Phrases: 5G positioning, location-based service, secret sharing

**ACM Reference format:**
Zheng Yan, Xinren Qian, Shushu Liu, and Robert Deng. 2022. Privacy Protection in 5G Positioning and Location-based Services Based on SGX. *ACM Trans. Sen. Netw.* 18, 3, Article 41 (August 2022), 19 pages.
https://doi.org/10.1145/3512892

The work is supported in part by the Academy of Finland under Grants 308087, 335262, and 345072; in part by the National Natural Science Foundation of China under Grant 62072351; in part by the ZheJiang Lab; in part by the Shaanxi Innovation Team Project under Grant 2018TD-007; and in part by the 111 Project under Grant B16037.
Authors' addresses: Z. Yan (corresponding author), The State Key Lab of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, China, 710071 and The Department of Communications and Networking, Aalto University, Konemiehentie 2, Espoo, Finland, 02150; email: zheng.yan@aalto.fi; X. Qian, The State Key Lab of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, China, 710071; email: 18151213250@stu.xidian.edu.cn; S. Liu, The Department of Communications and Networking, Aalto University, Konemiehentie 2, Espoo, Finland, 02150; email: liu.shushu@aalto.fi; R. Deng, The School of Information Systems, Singapore Management University, Victoria Street 81, Singapore, 188065; email: robertdeng@smu.edu.sg.

1550-4859/2022/08-ART41 $15.00
https://doi.org/10.1145/3512892

## 1 INTRODUCTION

5G positioning and its further application in **location-based services (LBSs)** have been widely studied due to their great benefits to people's life. The representative services include **Points of Interest (POI)** recommendation, navigation, taxi-hailing, and so on. However, individual position is sensitive and can be used to breach private information such as home address, religion, or even health status. It is essential to enhance privacy when designing a reliable 5G positioning service chain.

The study of the 5G positioning service chain can be divided into two main parts: positioning and LBS provision. 5G positioning focuses on the process of positioning, which includes positional signal measurement and position calculation [24]. Normally, three or more base stations are required to measure the signal originating from a positioning request device. These measurements are then communicated to a central infrastructure (known as fusion center) where they are combined to offer a position estimation with high accuracy and reliability by applying machine learning methods for truth discovery, attack detection, and tracing [24]. The estimated position is returned to the requesting device for further usage. However, an **LBS provider (LBSP)** receives the location of **user equipment (UE)** and provides UE with tailored information or services based on that location. For example, with the help of LBSs, UE can get answers to various location-based queries, such as the nearest **automated teller machine (ATM)**, restaurant, or retail store to its current geographical position. LBS has gained a wide range of promising applications [37, 38, 47, 48], such as extended reality, trouble shooting, and location-based training.

The privacy concern for the 5G positioning service chain resides in the trustworthiness of the involved infrastructures and communication channels. It includes two main aspects: UE position privacy and LBSP data privacy. This results from the sensitivity of position as the possibility of inferring sensitive information related to the position's owner. For example, a person's home address can be inferred if the location is in a residential area or his health status can be inferred if the location is a hospital for some specific disease treatment. Hence, the estimated position should be kept secret from unauthorized parties. However, an LBSP typically holds a POI database or valuable datasets, based on which location-based queries are processed and location-based services can be offered. The POI database and the datasets owned by the LBSP are their valuable assets, as building such a database and datasets generally requires consuming many resources and is by no means a trivial task. Therefore, the privacy of an LBSP should also be protected; that is, its data should not be disclosed without fees.

Existing solutions to the above problems are also divided into two categories according to their research focus. For the privacy protection in positioning process, the solutions are mainly based on k-anonymity [19, 46], homomorphic encryption [23, 42], and secure multi-party computation like garbled circuits [16]. For example, Pei et al. [33] implemented two secure and privacy-preserving 3D positioning schemes. Hussain and Koushanfar [14] proposed vehicle positioning solutions with Garbled Circuit and Beaver Micali Rogaway. For preserving privacy in location-based queries, methods such as access control [2, 45], mix zone [3, 11], $k$-anonymity [1, 8, 28], and dummy location [18, 20, 44] have been adopted to prevent LBSP from learning the exact position of UE while preserving the privacy in location-based services.

Though a variety of approaches can be found to provide privacy preservation in the 5G positioning service chain, the practicality of these solutions is still in doubt. First, solutions like k-anonymity, mix zone, or dummy locations always introduce a trusted third party that maintains all UE position information. When a UE starts a query, a cloaked area, instead of the exact position of the UE, is generated by the third party and sent to the LBSP for subsequent processing. This kind of method is vulnerable to the misbehavior of the third party. Second, solutions based on

cryptography such as homomorphic encryption and secure multi-party computation suffer from high computational complexity and are not efficient enough to support real-time positioning. They cannot well support various ways of data processing (e.g., data truth discovery and attack tracing based on machine learning [24]), which is needed during positioning. In particular, there are not any existing solutions that can provide seamless privacy protection of both UE position and LBSP data assets in 5G positioning and LBSs.

Based on Intel **Software Guard Extensions (SGX)**, this article proposes a novel lightweight scheme that can protect privacy in both 5G positioning and its further applications in LBSs in an integrated way. The scheme contains two parts. The first part is a privacy preservation scheme at a **fusion center (FC)** for processing the positioning data provided by multiple edge devices (i.e., base stations) based on UE control. The second part is a common framework for preserving both UE location privacy and LBSP data resource privacy when UE is being served by multiple LBSPs by applying secret sharing.

**Privacy preservation in UE positioning.** When applying the FC to aggregate the positioning data provided by multiple edge devices to offer accurate and reliable position information to UE, the edge devices use a key ($sk'$) provided by UE ahead of time to protect positioning data before sending them to the FC. At the FC, a **trusted execution environment (TEE)**, Enclave E1, is attested by UE to process the received UE positioning data. E1 decrypts the positioning data with the $sk'$ sent by UE through a secure channel between UE and E1, processes the data at E1 to get UE position information, which is shared with UE at real time through the secure channel. The E1 hosted by the FC also protects the UE position information with another key ($sk$) specified by the UE for flexibly providing multiple LBSs.

**Privacy preservation in LBSs.** UE can authorize multiple LBSPs to offer services. It first divides $sk$ into two parts ($sk_a$, $sk_b$). The first part ($sk_a$) is provided to another UE attested TEE (Enclave E2) at the FC. Another part ($sk_b$) is further divided into a number of $N$ shares ($sk_b^1, \ldots, sk_b^N$), each of which is issued to one of a number of $N$ LBS providers, respectively. With the key share, the LBSP can contact E2 to access the UE position for providing its LBS to UE. Concretely, (1) the LBSP attests E2 as trusted; (2) the LBSP sends its key share $sk_b^i$ to E2 for validity verification with regard to its authority for LBS provision to UE. Any $sk_b^i$ together with $sk_a$ can be combined to recover $sk$, thus allow E2 to access UE position, which also verifies the eligibility of the LBSP; (3) meanwhile, the LBSP issues its own key $lbsk$ to E2 to allow E2 to access to its resources (e.g., database, service information); (4) E2 accesses UE position and the LBS provider's resources and provides tailored LBS information to UE; (5) E2 removes the $lbsk$ and $sk$ from its temporal memory after above operations. In this way, neither the FC nor the LBSPs can access UE position and the LBS information tailored for the UE. Meanwhile, the privacy of LBSP's resources can also be preserved.

We test the performance of our scheme based on a prototype implementation. UE positioning can be completed within 1 ms with 0.5 KB data transfer. And for LBS provision, the whole process can be finished within 80 ms with 3 KB data transfer in the case of 10 LBSPs involved. The result shows that our proposed scheme preserves the privacy of both UE position and LSBP data with relatively high efficiency. In summary, the contributions of this article are listed below.

- We design a lightweight scheme that preserves both UE position privacy and LBSP data resource privacy based on Intel SGX. It is the first scheme to offer mutual privacy preservation for both UE and LBSP and can be easily integrated into current LBS systems.
- The scheme can support multiple LBSPs at the same time by applying secret sharing to allow UE to initiate LBS by providing its partial share of the secret to the SGX enclave, which can also easily authenticate LBSP based on its held secret share with the control of UE.

- We implement the proposed scheme and prove its efficiency through performance evaluation conducted over a real-world database.

The rest of the article is organized as follows: Section 2 reviews the related work regarding 5G positioning, LBS provision, and SGX-enabled data protection. Section 3 introduces the background knowledge of this article. The system model and security model of the scheme are defined in Section 4. Detailed description of scheme design is presented in Section 5, followed by security analysis in Section 6. Our experimental results are reported and analyzed in Section 7. Eventually, we end the article with a conclusion drawn in Section 8.

## 2 RELATED WORK

This section reviews the related work regarding privacy-preserving 5G positioning and LBSs. We also highlight the discussion on SGX data processing.

### 2.1 Privacy-preserving 5G Positioning

Privacy preservation in 5G network has attracted enormous attention. The research has been focusing on a variety of topics like network slicing [7], heterogeneous network [41], fog/edge computation [49], drone-enabled communication [40], and so on. Herein, we focus on discussing the related work in 5G positioning, where private information of involved participants is protected.

The related work can be divided according to the methods and scenarios of positioning. Konstantinidis et al. [19] proposed a scheme based on bloom filter and k-anonymity techniques for indoor positioning based on **Received Signal Strength (RSS)**. The localization query is first mapped into a Bloom filter vector and further obfuscated with another $k-1$ vectors before sending it to a computation server. Although privacy can be protected under k-anonymity, it requires a trusted third party for obfuscation. Similar work was also presented by Sazdar et al. in Reference [34] based on a bloom filter. Also, for indoor positioning, Li et al. [23] implemented a privacy-preserving solution for fingerprint-based localization. By leveraging the homomorphic property of the Paillier cryptosystem, computation servers can compute the Euclidean distance between query and database in a form of ciphertext and respond with the nearest location. In this way, the privacy of the query can be protected. However, this solution is discovered with disclosure of the server's database when the client is not totally "honest" and is allowed to send fabricated queries to the server, as spotted by Yang and Järvinen in Reference [42]. They further improved the scheme with two solutions based on fully homomorphic encryption and garbled circuits, respectively. But the high computational cost introduced by cryptographic protocols retards its application in practice. Another work in solving privacy leakage in indoor positioning was presented by Schauer, Dorfmeister, and Wirth in Reference [35]. To prevent user privacy leakage in probe requests when accessing localization services, they adapted the communication protocol between access points and UE from active IEEE 802.11 scan to passive scan, so the UE only listens to the periodical signals sent by the access points. This solution requires an update to current infrastructures. Additionally, by abandoning active communications from UE to access points, the solution suffers from accuracy decrease. Liu et al. [26] implemented a verifiable indoor positioning scheme for edge computing scenario.

Apart from indoor positioning, related work has also been presented in **vehicular ad hoc networks (VANETs)** and image-based positioning. Based on a one-pass authentication key agreement protocol, Pei et al. [33] implemented two secure and privacy-preserving 3D positioning schemes. Hussain and Koushanfar [14] also proposed secure vehicle positioning methods with garbled circuits and Beaver Micali Rogaway. Shu et al. [39] focused on studying privacy-preserving positioning based on triangulation by formulating localization as **least-square-error (LSE)** estimation and implemented its secure computation by matrix multiplication and homomorphic encryption.

Based on the homomorphic addition property of Paillier encryption, Jiang et al. [17] implemented location calculation based on encrypted distance information so the distance can be protected. Liu et al. [27] solved the location privacy protection in D2D cooperative communication.

Obviously, the above proposed schemes apply different methods from ours for 5G positioning privacy protection.

### 2.2 Privacy-preserving LBSs

The public concern on privacy has stimulated lots of research efforts in privacy-preserving LBSs.

Information access control [2, 45] was first proposed to protect location information gathered by location tracking systems. It requires that the location of the user is gathered and relies on LBSP to restrict access to stored location information through rule-based policies. But it is vulnerable when the third party that maintains all user locations misbehaves.

Mix zone [3, 11], *k*-anonymity [1, 8, 28], and dummy location [18, 20, 44] solve this problem by hiding user locations in some big zone or many records so LBSP cannot locate the exact position of a user. Both mix zone and k-anonymity use a trusted third party to assign a user with cloaked query, so the user can query LBSP without revealing its exact location. In this situation, k-anonymity is affected heavily by the distribution and density of users, which are out of control, and the balance of privacy and precision is another difficult problem that needs to be solved. Dummy location is completed by sending many random locations along with the user's query. Although dummy location does not rely on any third party, the LBSP can still infer the user's location in a coarse level, which leads to weak privacy.

In contrast, schemes based on cryptography can provide stronger privacy protection than the above methods. Based on **private information retrieval (PIR)**, Paulet et al. [32] allowed the user to retrieve data from a POI database without disclosing query indexes to LBSPs. Similar work was presented by Ghinita et al. by applying computational PIR [12]. To improve performance, Paulet et al. [32] proposed a scheme by integrating PIR and oblivious transfer. A scheme based on Paillier homomorphic encryption was also presented by Yi et al. [43] for preserving the location privacy of users in LBS. To protect the location privacy and LBSP's data privacy, Liu et al. [25] designed a framework based on oblivious transfer. However, oblivious transfer introduces high communication cost.

As summary, the above solutions are either dependent on a trusted third party or heavy computation cost. Thus, they are not suitable for the practical use case.

### 2.3 SGX-enabled Data Protection

Intel SGX is a security solution promising strong and practical security guarantees for trusted computing. It has been integrated into many security-aware systems for processing sensitive data. For example, Gremaud et al. [13] built an SGX-based middleware for IoT data processing in the cloud. Schuster et al. [36] and Le et al. [22] designed SGX-based secure data analytics framework VC3 and SGX-PySpark for MapReduce and other query computations. Chen et al. [7] proposed QShield to achieve secure and efficient SQL-query based on SGX with multi-user query control. With regard to 5G positioning, Choi et al. [15] mentioned an idea to process localization measurement in attestable SGX containers, but without implementation. And for LBS provision, Kulkarni et al. [21] implemented a simple location-based service using SGX to protect user location and evaluated its performance in terms of efficiency and effectiveness.

Different from the above two works, we consider how to preserve privacy in both 5G positioning and LBS provision. Apart from user location privacy in positioning at FC and LBS provision, we also protect data privacy for LBSPs. Our scheme is more holistic and advanced compared with the existing works.

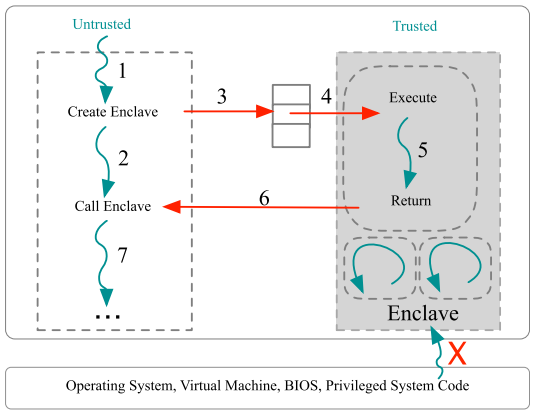



Fig. 1. SGX system diagram.

## 3 PRELIMINARIES

This section briefly introduces the background knowledge of the technologies adopted in the article: Intel SGX and bilinear maps.

### 3.1 Intel SGX

Intel SGX is a promising hardware-assisted trusted computing technology [15]. It provides memory isolation, which enables a host to set up a protected execution environment, called enclave, so that the code and data run inside it are resilient to attacks from privileged software, including OS kernel and **Virtual Machine (VM)** hypervisor. A typical SGX system consists of two modules as presented in Figure 1. The untrusted module executes security uncritical codes and the trusted module executes security-critical codes inside the enclave. The two modules communicate via two specifically designed calling functions: Ecall and Ocall. Ecall is a trusted function that enters an enclave, while Ocall is an untrusted function that leaves an enclave. Thus, Ocall for I/O operations should be carefully applied during the enclave algorithm design. Intel SGX also offers two auxiliary functionalities: remote attestation and storage sealing. The former makes a distant entity capable of verifying the authenticity of an enclave, checking the integrity of desired code running inside it and meanwhile establishing a secure communication channel with the enclave. The latter allows to securely store enclave data in an untrusted storage outside the enclave for future recovery in case of server shutdown, system failure, and/or power outage. For a more detailed technical analysis of Intel SGX, please refer to Reference [9].

**Side-channel attack** SGX is vulnerable to access pattern based side-channel attacks. These attacks make use of information like page fault, cache, branch prediction, and speculative execution [5, 29] to predict the operations inside SGX. **Oblivious RAM (ORAM)** is an effective countermeasure to defend against these attacks. It provides cryptographically proven protection against access pattern-based side-channel attacks. The detailed implementation of ORAM schemes can be are in References [30, 31]. For detailed review on SGX attacks and countermeasures, please refer to Reference [10].

### 3.2 Bilinear Maps

Let $G_1$ and $G_2$ be two multiplicative cyclic groups of prime order $p$, and $g$ be a generator of $G_1$. An efficiently computable bilinear map $e : G_1 \times G_1 \rightarrow G_2$ defined over the two groups satisfies the following properties:

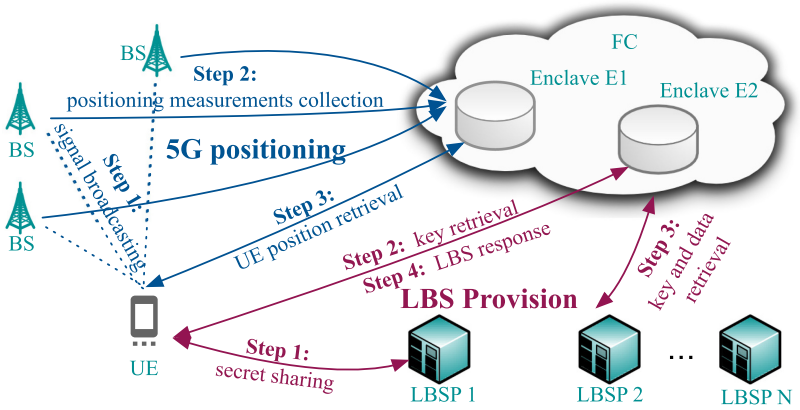



Fig. 2. System model.

- Bilinearity: for all $a, b \in Z_p$, there exists $e(g^a; g^b) = e(g; g)^{ab}$;
- Non-degeneracy: $e(g; g) \neq 1$ where 1 is a unit in $G_2$;
- Computability: for any $u, v \in G_1$, there exists an efficient algorithm to compute $e(u; v)$.

Decisional **Bilinear Diffie-Hellman (BDH)** Assumption: The security of our secret sharing scheme is based on the Decisional BDH assumption. Basically, let $a, b, c, z$ be chosen randomly from $Z_p$, there exists no probabilistic polynomial time algorithm $BB$ that can distinguish the tuple $(A = g^a; B = g^b; C = g^c; e(g; g)^{abc})$ from the tuple $(A = g^a; B = g^b; C = g^c; e(g; g)^z)$ with more than a negligible advantage $\epsilon$, where the probability is computed over a randomly chosen generator $g$, the randomly chosen $a, b, c, z \in Z_p$, and the random bits consumed by $BB$. For more details, please refer to Reference [4].

## 4 PROBLEM STATEMENT

This section describes the system model and the security model of our proposed scheme about privacy-preserving 5G positioning and LBSs.

### 4.1 System Model

The system model for 5G positioning and its further application in LBSs is shown in Figure 2. The system contains four different entities: **base station (BS)**, UE, **fusion center (FC)**, and LBSP. Note that a base station is an access point with device authentication functionality, which may have one or multiple antennas. Specially, roadside units or other signal receivers can also serve the role of the base station for the purpose of positioning.

UE initializes a positioning query by sending wireless signals to surrounding BSs. Once receiving the signals, BS extracts positioning measurements such as **Time of Arrival (ToA)** and **Direction of Arrival (DoA)**. Theoretically, the UE position can be estimated through these measurements. However, the position from one BS maybe not be accurate, as these measurements could be impacted by interference, attacks, interruption, and so on. To eliminate the influence from noise data, positioning calibration with multiple BSs are required. In practice, UE positioning data collected by multiple BSs are sent to an FC for further processing like clustering to get accurate position information of UE by eliminating noises and unreal data [24]. The final position result is sent back to UE through BS downlink transmission. Multiple LBSPs could exist in the system. They offer a diversity of LBSs, such as restaurant or gas station searching based on UE's requests, tourist guideline, and location-based trouble shooting. To make full use of resources, LBSPs are intended to

outsource their services to a powerful computation center, which can also be the FC for simplicity. LBS-related computation is processed in the FC.

The privacy concern on the above system includes two aspects: UE position privacy and LBSP data privacy. This results from the sensitivity of position as the possibility of inferring sensitive information related to the user of UE. For example, the user's home address can be inferred if his/her UE's location is in a residential area or his/her health status can be inferred if the location is a hospital for some specific disease treatment. Hence, the estimated position should be kept secret from unauthorized parties. However, an LBSP typically holds a POI database or valuable datasets, based on which location-based queries are processed and location-based services can be offered. The datasets owned by the LBSP are their valuable assets, as building a POI database and collecting location-related datasets generally requires consuming many resources and is by no means a trivial task. Therefore, the privacy of an LBSP should also be protected; that is, its data should not be disclosed without authorization and control. The formal privacy definitions of UE and LBSP are below.

*Definition 4.1 (UE's privacy in Positioning and LBS Provision).* Owing to sensitivity, UE hides its position $UE_p$ from all other system entities in the process of positioning and LBS provision.

*Definition 4.2 (LBSP's Privacy in LBS Provision).* The POI database held by LBSP is a valuable asset that should be protected in the process of LBS provision.

To achieve the data privacy, both UE positioning and LBS provision are processed with enclaves held by FC. We denote them as E1 and E2 separately. E1 and E2 have different functionalities for different purposes. E1 is for UE positioning by cooperating with BSs, and E2 is for LBS by cooperating with LBSPs. Although both E1 and E2 are built on Intel SGX, they are two independent modules that have different settings of running space and other parameters. They can even be applied on different servers held by FC. Besides, the enclaves are created only if needed and destroyed immediately after its service is completed.

### 4.2 Security Model

Based on the above system model, we present the security model for all the system entities.

FC could be a cloud server with sufficient storage and computation resources. But FC cannot be fully trusted by UE and LBSP, although it normally performs according to system design. This is because FC could be hacked or intruded or suffer from internal attacks. Thus, the data stored in the FC could be leaked without essential protection, e.g., encryption.

We suppose the TEE applied in this article can be fully trusted by applying some essential countermeasures to overcome its vulnerabilities (e.g., side-channel attacks [6]). FC can host several TEEs, e.g., Intel SGX enclaves, to process data in a secure way and with execution code integrity, which can be attested by TEE clients.

We suppose BSs are semi-trusted. It receives UE signals and connects UE to a mobile core network. Predictably, UE is in the coverage range of BS, however, the accurate position of UE is hard to gain due to signal interruption, environmental reasons, and attacks. Besides, we suppose that BSs do not collude with FC due to business conflicts between mobile operators and FC.

We suppose UE is a trusted system user. UE wants to get its accurate position information. It trusts BSs to connect it to the mobile core network through mutual authentication. UE would like LBSPs to offer LBSs, but does not want them to get its position during service retrieval. Since the UE position is sensitive and can be used to breach private information such as home address, religion, or even health status, UE intends to hide its accurate position UEp from others in both positioning and LBS provision.

Table 1. Notations

| Symbol | Description |
|---|---|
| $sk'$ | The key used to protect positioning data sent from BSs |
| $sk$ | The key used to protect the position of UE |
| $sk_a$ | One part of $sk$ |
| $sk_b$ | Another part of $sk$ |
| $sk_b^1, \ldots, sk_b^N$ | The shares of $sk_b$, which are issued to different LBSPs, respectively |
| $psu$ | The pseudonym of UE |
| $UE_p$ | The position information of UE |
| $lbsk_i$ | The key used to protect LBSP $i's$ resources |
| $UE_{lbs}$ | The LBS response according to UE's position |

We suppose LBSPs are semi-trusted. They offer their LBSs to UE but may also be interested in its position information. However, the resources (e.g., maps, LBS-related datasets) held by LBSP are valuable assets owing to their business value. LBSPs do not want other parties to get their resources during LBS provision. Thus, they do not fully trust FC, since FC could gain and disclose the valuable resources of LBSPs and impact their business.

We suppose the communications between all system entities are protected by applying existing secure communication protocols.

## 5 SCHEME DESIGN

Our scheme design focuses on the computation and communication security between all the entities involved. This refers to privacy preservation in UE positioning and privacy preservation in LBSs offered by multiple LBSPs. The description of each part is presented as follows, and the notations used in this article are described in Table 1.

### 5.1 Privacy Preservation in UE positioning

The scheme for privacy preservation in UE positioning is shown in Figure 3. UE issues BSs a key $sk'$ and its pseudonym $psu$ during core network access. $sk'$ and $psu$ are registered at the core network ahead of time. When applying the FC to aggregate the positioning data provided by multiple BSs to offer accurate and reliable position information to UE, the BS (edge devices) uses $sk'$ provided by UE to protect positioning data (e.g., ToA and DoA) before sending them to the FC. At the FC, an enclave E1 is created and attested by UE first to process the received UE positioning data as trusted by UE. If the attestation is positive, then a secure channel is established between UE and E1. Then, UE sends E1 $(sk, sk', psu)$ through the secure channel. E1 decrypts the positioning data from BS with the $sk'$, processes the data in E1 to get UE position information $UE_p$, which is shared with UE in real-time through the secure channel. The E1 hosted by the FC also protects the $UE_p$ with another key $(sk)$ specified by the UE for flexibly providing multiple LBSs.

### 5.2 Secret Sharing

To support the service from multiple LBSPs, we design a crypto primitive named **secret sharing (SS)** based on bilinear maps, which distributes the full decryption capability among involved participants, i.e., the enclave E2 and one of LBSPs. With our scheme, neither the enclave nor the LBSP has the full capability to recover $UE_p$ outside the scope of a position query. Besides, the exposure risks of $sk$ to the enclave can be eliminated as it is distributed among multiple participants. Meanwhile, it inherently supports authentication between E2 and LBSP for LBS provision. Such a secret sharing scheme can be defined over a tree-based access structure $T$ as shown in Figure 4.

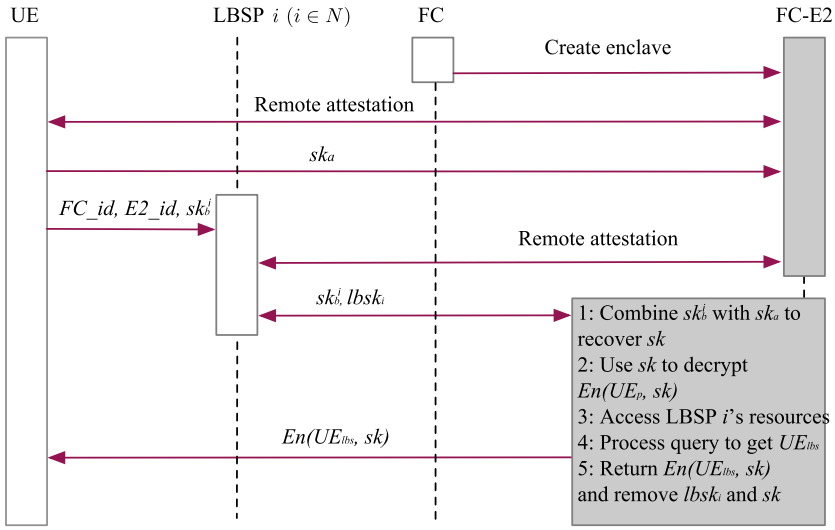



Fig. 3. Privacy preservation in positioning.

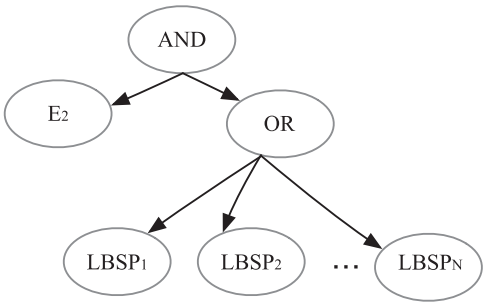



Fig. 4. Tree structure of secret sharing.

Specifically, the root node of the access tree has an *AND* gate: its left child represents E2; its right child has an *OR* gate with $N$ children, each representing an LBSP. We notice that this construction has been widely used in cryptography, like attribute-based encryption.

Suppose we have two cyclic groups $G_1$ and $G_2$, both with prime $p$. Let $g$ be the generator of $G_1$ and $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map over $G_1$ and $G_2$. The scheme SS consists of following algorithms as shown in Scheme 1:

*5.2.1 Setup.* This algorithm takes a security parameter $\lambda$ and a non-zero positive integer $M$ as inputs. It first defines a universe of participants $U = 0, 1, 2, \ldots, N$, where $0, 1, \ldots, N$ denote the enclave E2 and a number of $N$ LBSPs, respectively. For each participant $i$ in $U$, it uniformly chooses a number $t_i$ from $Z_p$. It then picks up a number $y$ from $Z_p$. Finally, this algorithm outputs a public key $pk : g^{t_0}, g^{t_1}, \ldots, g^{t_N}, Y = e(g; g)^y$ and a master private key $msk : t_0, t_1, \ldots, t_N, y$.

*5.2.2 Shares Generation (ShareGen).* This algorithm takes as inputs the predefined tree-based access structure $T$ and the master secret key $msk$. It computes key shares for all involved participants in $U$ as follows: The algorithm first defines a polynomial $q_t(x)$ for each node $t$ in $T$. Specifically, the degree $d_t$ of each polynomial is derived from the threshold value $k_t$ of the current node, i.e., $d_t = k_t - 1$. Suppose that the node has an AND gate, then $k_t$ equals to the number of the node's

**ALGORITHM 1:** Secret Sharing

**Algorithm 1: Setup**
**Input:** security parameter $\lambda$, non-zero positive integer $M$
**Output:** public key $pk$, master private key $msk$

1: Let $g$ be the generator of $G_1$ and $e : G_1 \times G_2 \rightarrow G_2$;
2: For enclave E2 and each LBSP ($i$), choose a random number $t_i$ from $Z_p$;
3: Choose a random number $y$ from $Z_p$;
4: Output a public key $pk$ as $g^{t_0}, \ldots, g^{t_N}, Y = e(g, g)^y$;
5: Output a master private key $msk$ as $t_0, t_1, \ldots, t_N, y$;

**Algorithm 2: Shares Generation**
**Input:** access structure $T$, master private key $msk$
**Output:** key share $sk_a$, key shares $sk_b^i (i = 1, \ldots, N)$

1: Define polynomial for tree-based access structure $T$ as

$$q_t(x) = \begin{cases} ax + y; & t \text{ is an AND gate (root)} \\ a + y; & t \text{ is an enclave node } (i = 0) \\ 2a + y; & t \text{ is an OR gate} \\ 2a + y; & t \text{ is an LBSP node } (i \in 1, \ldots, N) \end{cases}$$

2: Generate key share $sk_a = g^{\frac{a+y}{t_0}}$ for enclave E2;
3: Generate key share $sk_b^i = g^{\frac{2a+y}{t_i}}$ $(i = 1, \ldots, N)$ for each LBSP $i$;

**Algorithm 3: Position Encryption**
**Input:** UE's position $UE_p$, public key $pk$
**Output:** encrypted UE's position $ct_p$

1: Choose a random number $s$ from $Z_p$;
2: Encrypt $UE_p$ as $ct_p = UE_p \times Y^s; E_i = g^{s \times t_i}$;

**Algorithm 4: Position Decryption**
**Input:** ciphertext $ct_p$, key share $sk_a$, key shares $sk_b^i (i = 1, \ldots, N)$
**Output:** UE's position $UE_p$

1: Perform a bottom-up node decryption on access structure $T$ as

$$F_t = \begin{cases} e(E_i, sk_a); i = 0 \\ e(E_i, sk_b^i); i \in 1, \ldots, N \end{cases}$$

children; suppose that the node has an OR gate or it is a leaf node, then $k_t$ equals to 1. Besides, for the root node $r$, it sets $q_r(0) = y$; for other nodes $t (t \neq r)$, it sets $q_t(0) = q_{parent(t)}(index(t))$, where $parent(t)$ and $index(t)$ represent the parent node of $t$, the index in all children of $t$'s parent node, respectively. Furthermore, all other polynomial parameters are chosen from $Z_p$ randomly.

More concretely, we have:

$$q_t(x) = \begin{cases} ax + y; & t \text{ is an AND gate (root)} \\ a + y; & t \text{ is an enclave node } (i = 0) \\ 2a + y; & t \text{ is an OR gate} \\ 2a + y; & t \text{ is an LBSP node } (i \in 1, \ldots, N), \end{cases}$$

where $a$ is a random number in $Z_p$. Once these polynomials have been decided, the algorithm generates a key share for each participant $i$ by computing $sk^i = g^{\frac{q_{t=i}(0)}{t_i}}$, concretely, $sk_a = g^{\frac{a+y}{t_0}}$ and $sk_b^i = g^{\frac{2a+y}{t_i}} (i = 1, \ldots, N)$.

*5.2.3 Position Encryption (PosEnc).* This algorithm takes as inputs a UE position $UE_p$ and the public key $pk$. It generates a corresponding ciphertext $ct_p$ that can be distributed among participants $i|i \in U$ and only the enclave joining together with one of LBSPs that holds $sk_b^i$ can recover $UE_p$. To do so, the algorithm first chooses a random number $s$ from $Z_p$ and then encrypts $UE_p$ in $G_2$ as $ct_p = UE_p \times Y^s; E_i = g^{s \times t_i} (i \in U)$.

*5.2.4 Position Decryption (PosDec).* This algorithm takes as inputs the ciphertext $ct_p$, the key share $sk_a$ of the enclave, and the key share $sk_b^i (i \in 1, \ldots, N)$ of one of LBSPs. To reconstruct the $UE_p$, it performs a bottom-up node decryption process. In the case where $t$ is a leaf node, it computes

$$F_t = \begin{cases} e(E_i, sk_a); i = 0 \\ e(E_i, sk_b^i); i \in 1, \ldots, N. \end{cases}$$

From the above algorithms, we can see that

$$\begin{aligned} ct_p &= UE_p \times Y^s; E_i = g^{s \times t_i} (i \in U) \\ &= UE_p \times e(g, g)^{ys}; E_i = g^{s \times t_i} (i \in U). \end{aligned} \tag{1}$$

Thus,

$$F_t = \begin{cases} e(g^{s \times t_0}, g^{\frac{a+y}{t_0}}) = e(g, g)^{s(a+y)} \\ e(g^{s \times t_i}, g^{\frac{2a+y}{t_i}}) = e(g, g)^{s(2a+y)}. \end{cases}$$

We can easily get that $\frac{e(g,g)^{2s(a+y)}}{e(g,g)^{s(2a+y)}} = e(g, g)^{ys} = Y^s$, then $\frac{UE_p \times Y^s}{Y^s} = UE_p$.

### 5.3 Privacy Preservation in LBS Provision

UE can request multiple LBSPs to provide their services. It first divides $sk$ into two parts $(sk_a, sk_b)$ by using the above described SS scheme. The first part $(sk_a)$ is provided to another UE attested TEE (Enclave E2) at the FC, i.e., E2 is attested by UE as trustworthy. The other part $(sk_b)$ is further divided into a number of $N$ shares as $sk_b^1, \ldots, sk_b^N$, each of which is issued to one of $N$ LBSPs to give it right to offer LBS, respectively. With the key share $sk_b^i$, any LBSP $i$ can contact E2 to query the UE position for providing its service. The privacy preservation in LBS provision is illustrated in Figure 5. Concretely,

(1) FC creates a trustworthy enclave space E2 for the processing of LBS provision;
(2) UE attests E2 as trusted and builds up a secure channel with E2;
(3) UE sends its key share $sk_a$ to E2 through the secure channel for further LBS provision;

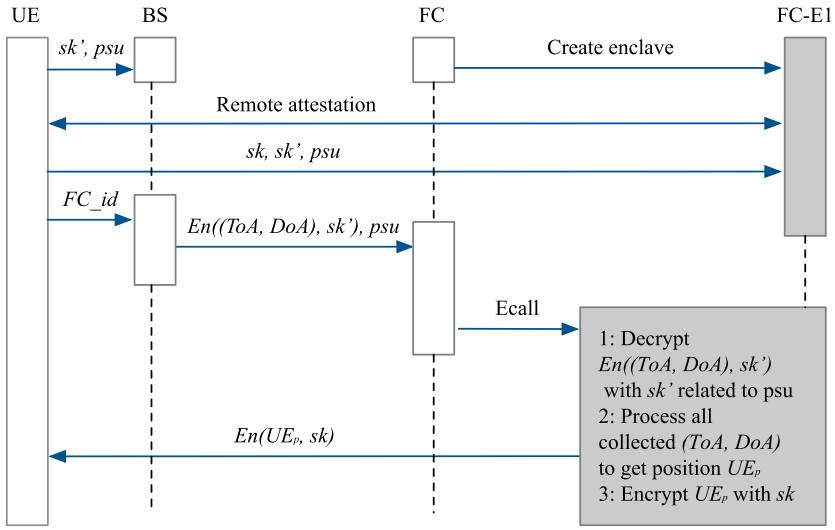



Fig. 5. Privacy preservation in LBS provision.

(4) UE sends any target LBSP $i$ with an LBS request consisting of FC id, E2 id, and key share $sk_b^i$;
(5) Based on the id of FC and the id of E2, the LBSP $i$ attests E2 as trusted and builds up a secure channel with E2;
(6) LBSP $i$ sends its key share $sk_b^i$ to E2 for validity verification concerning its eligibility for LBS provision to UE. Any $sk_b^i$ together with $sk_a$ can be combined to recover $sk$, thus allow E2 to access UE position, which verifies the eligibility of the LBSP $i$ to offer its service to UE;
(7) LBSP $i$ issues its own key $lbsk_i$ to E2 to allow E2 to access its resources, e.g., database, service information;
(8) E2 accesses $UE_p$ and the LBSP $i's$ resources and then provides tailored LBS information $UE_{lbs}$ to UE;
(9) E2 encrypts the $UE_{lbs}$ with key ($sk$), which is specified by the UE in the positioning process and sends the encrypted information back to UE;
(10) E2 removes $lbsk_i$ and $sk$ from its temporal memory after the above operations for security purposes. In this way, neither the FC nor the LBSP can access the UE position and the LBS information tailored for the UE. Meanwhile, LBSP resources can also be preserved from disclosing to FC and UE.

Note that our scheme supports UE query over multiple LBSPs. UE can query any one LBSP for its services at any time by setting up the enclave E2. Multiple enclaves can be created at the same time when FC has enough computation power. The benefit of our design is that when UE starts a new query to a new LBSP, the solution is still working without the need of new key distribution.

## 6 SECURITY ANALYSIS

Security is proved through simulation-based methods. In the simulation-based method, a real world and a simulation world are both defined. The attackers can compute whatever based on the given ciphertext in the real world. While they can only operate without the ciphertext in the simulation world. The scheme is secure on the condition that the view in the real world is indistinguishable from that in a simulation world. The theorem of our proposed scheme and the proof is as follows.

### 6.1 Privacy Preservation in Positioning

Theorem 6.1 (UE's Privacy in Positioning). *Based on the security of symmetric encryption and enclave technology, UE hides its position $UE_p$ from all other participants in the process of positioning.*

Proof. According to the security model given in Section 4, we define polynomial-time simulators $Sim_{BS}$ and $Sim_{FC}$ against the attacking opponents $A_{BS}$ and $A_{FC}$, respectively. We prove that the view generated by the simulator is statistically close to what is viewed in the real world.

Suppose that $View_{BS} = (ToA, DoA), sk'$ is the view observed by BS in the real world. Meanwhile, simulator $Sim_{BS}$ works as follows: It generates a random position measurement $(ToA', DoA')$ and encrypts it with a random symmetric encryption key $sk''$, during which the view observed by attacker $A_{BS}$ is $View'_{BS}$. As the indistinguishability of symmetric encryption, it is easy to conclude that $View_{BS}$ and $View'_{BS}$ are statistically the same with each other. Thus, the privacy of UE is preserved against BS.

For simulator $Sim_{FC}$, it generates a random value set $r_1, r_2$ and sends it to attacker $A_{FC}$. Thus the view of attacker $A_{FC}$ can be denoted as $View'_{FC} = r_1, r_2$. As the view observed by FC is $View_{FC} = En((ToA, DoA), sk'), psu$, which is indistinguishable to $View'_{FC}$ according to the security of symmetric encryption and enclave technology, the privacy of UE is preserved. □

It is noteworthy that BS receives UE signals and connects UE to a mobile core network. Predictably, UE is in the coverage range of BS, however, the accurate position of UE is hard to gain due to signal interruption, environmental reasons, and attacks. Besides, we suppose that BSs do not collude with FC due to business conflicts between mobile operators and FC. Thus, the privacy of UE is still preserved here.

### 6.2 Privacy Preservation in LBS Provision

Theorem 6.2 (UE's Privacy in LBS Provision). *Based on the decisional BDH assumption and enclave technology, UE hides its position $UE_p$ from other participants in the process of LBS.*

Proof. Similarly, we define polynomial-time simulators $Sim_{LBSP_i}$ and $Sim_{FC}$ against the attacking opponents $A_{LBSP_i}$ and $A_{FC}$.

According to the protocol, $View_{LBSP_i} = \{FC_{id}, E2_{id}, sk_b^i\}$ is the view observed by $LBSP_i$. For $Sim_{LBSP_i}$, it generates a random value $r$ and sends $FC_{id}, E2_{id}, r$ to $A_{LBSP_i}$. Based on decisional BDH assumption, $sk_b^i$ is indistinguishable to random value $r$. Thus, $View_{LBSP_i}$ is indistinguishable to $View'_{LBSP_i} = \{FC_{id}, E2_{id}, r\}$. The privacy of UE is preserved against $LBSP_i$.

For $Sim_{FC}$, it generates a random set $\{r_1, r_2, r_3, r_4\}$ and sends it to $A_{FC}$. Thus, the view of $A_{FC}$ is $View'_{FC} = \{r_1, r_2, r_3, r_4\}$. As the randomness of enclave encryption function, $View'_{FC}$ is statistically close to the view of FC, which is $View_{FC} = \{En(sk_a), En(POI), En(sk_b^i), En(UE_p)\}$. Thus, the privacy of UE is preserved against FC. □

Theorem 6.3 (LBSP's Privacy in LBS Provision). *Based on the decisional BDH assumption and enclave technology, LBSP hides its POI database from other participants in the process of LBS.*

Proof. We define polynomial-time simulators $Sim_{UE}$ and $Sim_{FC}$ against the attacking opponents $A_{UE}$ and $A_{FC}$.

We suppose there is a random POI database $P$. According to the protocol, the view of UE in the real world is $View_{UE} = UE_{lbs}$, which is the LBS response corresponding to the position at $UE_p$. For simulator $Sim_{UE}$, it generates a random query $UE'_p$ and processes the LBS over database $P$. The view during this process is denoted as $View'_{UE} = UE_{lbs'}$. Since $P$ is a random database, the distribution of $View'_{UE}$ is statistically close to or the same as $View_{UE}$. Thus, the privacy of LBSP is preserved against UE.

Table 2. Benchmarks of Enclave Operations

| Enclave Tasks | Time | Enclave Tasks | Time |
|---|---|---|---|
| create | 1.45 s | remote attestation | 27.6 ms |
| entry (Ecall) | 0.013 ms | exit (Ocall) | 0.009 ms |

Table 3. Example of Simulated Dataset

| Position | $BS_1$ | $BS_2$ | $\cdots$ | $BS_N$ |
|---|---|---|---|---|
| $(x_1, y_1)$ | $(ToA, DoA)$ | $(ToA, DoA)$ | $\cdots$ | – |
| $(x_2, y_2)$ | $(ToA, DoA)$ | $(ToA, DoA)$ | $\cdots$ | $(ToA, DoA)$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $(x_l, y_l)$ | – | $(ToA, DoA)$ | $\cdots$ | $(ToA, DoA)$ |

$(ToA, DoA)$ is the time of arrival and direction of arrival between UE and each BS; "–" means no signal collected when UE is out of range of BS

The proof of FC is omitted, as it is similar to Theorem 6.2. □

In summary, the proposed scheme provides privacy protection for both UE positioning and LBS provision.

## 7 EVALUATION AND RESULTS

This section performs evaluation on the proposed scheme in terms of privacy-preserving positioning, secret sharing, and privacy-preserving LBS Provision regarding their computation costs and communication costs. We also compare the costs with our scheme to the baremetal one without applying our scheme.

To evaluate the scheme performance, we implement it with GCC and cmake 3.10.2 on a desktop running a Ubuntu-18.04 operating system with 2.20 GHz Intel(R) Core(TM) i-5200U. We also execute Intel SGX with SDK-2.11 and the enclave page cache of SGX is set to the maximum available size of 128 MB.

### 7.1 Benchmarking SGX Overhead

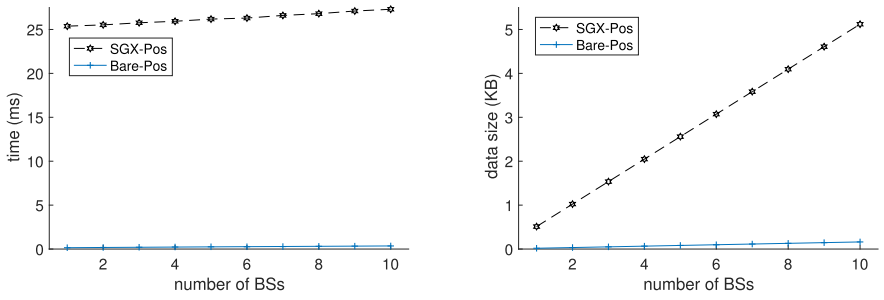
To quantify the overhead involved due to the SGX, we benchmark the latency of basic enclave operations. Table 2 lists the running time of enclave creation, remote attestation, enclave entry, and exit (ECall and OCall). All the results are derived after taking into account the average and variance over 100 runs.

The enclave creation and remote attestation are one-time costs involved during the phase of initialization. Every new client has to bear the initiation cost of retrieving the measurement quote and generating the SGX public key pair. The other tasks, ECall and OCall, are recurring costs and are decided according to the program design.

### 7.2 Performance of Privacy-preserving Positioning

Here, we focus on the computation and communication cost incurred by SGX and compare it (SGX-Pos) with the baremetal implementation (Bare-Pos) of the same application.

We adopt the positioning method based on ToA and DoA. Thus, the ToA and DoA signals between UE and each BS at different positions are recorded as the raw data for further experiments. The example of the collected data is listed in Table 3. $(x, y)$ is an integer pair with 32 bits each. It represents the latitude and longitude of UE's position. $(ToA, DoA)$ is also an integer pair with 8 bytes each, which represents the time of arrival and direction of arrival between UE and each




(a) Computation cost (b) Communication cost

Fig. 6. Overhead of privacy-preserving positioning.

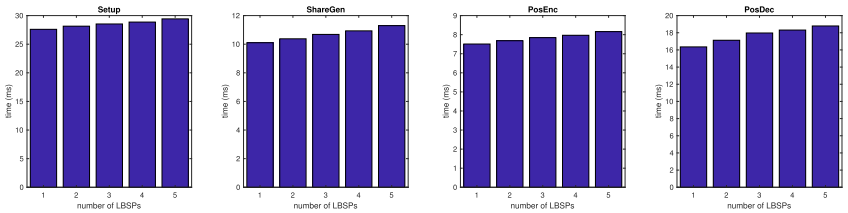



Fig. 7. Performance of secret sharing.

BS. "–" means no signal collected when UE is out of the coverage of BS. We suppose that UE can only reach BS within a distance of 200 m. These signals are encrypted and sent to enclave FC-E1 for position estimation. The overhead covers the time from a positioning query to its response.

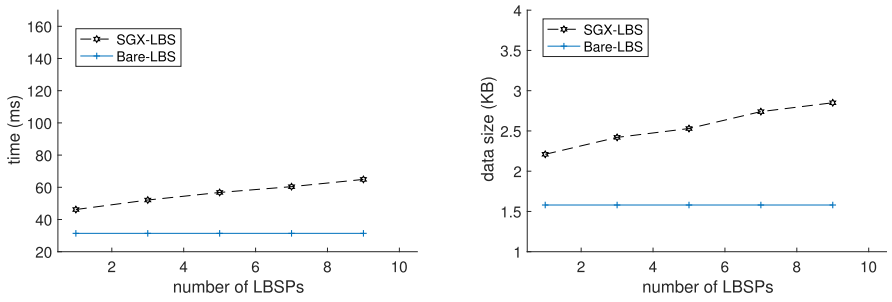
Figure 6 presents the computation and communication cost of the positioning as the number of BS increases. For SGX-Pos, the computation cost remains at the same level (around 25 ms) with the increase of the number of BSs, while the communication cost increases linearly. A majority of this additional computation cost results from transferring the ciphertext and decryption key from BS and UE to the enclave. More specifically, ECalls have to be initiated to execute positioning estimation. Additional overhead is contributed by the encryption and decryption of position signals. However, these costs are marginal and do not lead to noticeable service delays. Although it seems a big overhead when compared with Bare-Pos, which shows only 1 ms time delay and 0.5 KB data transfer, the time delay and data transfer required by SGX-Pos is not a burden to the current network. Thus, the increased overhead is acceptable as a compromise for high-level security protection.

### 7.3 Performance of Privacy-preserving LBS Provision

We experimented over a public POI database collected in Guizhou, China.[1] The LBS provision is to return the nearest neighbor POI according to the position specified in the request. The performance was measured in two phases: secret sharing and LBS query processing. The results of the two phases are described as follows:

*7.3.1 Secret Sharing (offline).* Secret sharing is an offline process that is responsible for key generation, key distribution, position encryption and decryption. It is a one-time cost during the service initialization phase. We measured the computation cost of the four specific operations with regards to the number of LBSPs. As shown in Figure 7, all operations grow linearly with the increase of the number of LBSPs. Among four operations, Setup takes the most computation

[1] https://datacatalog.worldbank.org/dataset/china-ghuizou-poi.




(a) Computation cost (b) Communication cost

Fig. 8. Overhead of privacy-preserving LBS.

cost and PosDec is the second. ShareGen and PosEnc are the third and fourth, respectively. For secret sharing with 5 LBSPs, the running time is no more than 30 ms for the Setup, 12 ms for the ShareGen, 9 ms for the PosEnc, and 20 ms for the PosDec. We can see that the whole process of SS is in a millisecond level, and the computational cost introduced by SS is acceptable considering the limited number of LBSPs.

*7.3.2 LBS Query Processing (online).* In contrast, LBS query processing is an online process that takes UE's position as input and returns the nearest neighbor from the POI database. The overhead of LBS query processing in terms of the number of LBSPs is presented in Figure 8. Similarly, we compared the SGX-based scheme (SGX-LBS) to the baremetal implementation (Bare-LBS) of the same application. As we can see, both the computation and communication costs of SGX-LBS increases with the increase of the number of LBSPs. This is mostly caused by the remote attestation between LBSPs and the enclave E2. Additional overheads like position decryption and query processing are considered marginal and do not lead to noticeable service delays. For Bare-LBS, the cost is steady, as the processing of LBS is irrelevant to the number of LBSPs. For LBS provision, the whole process can be finished within 80 ms with 3 KB data transfer in the case of 10 LBSPs involved. We can see that the cost introduced by privacy preservation in LBS provision is not high, especially when attestation on E2 has been performed already at initiation.

## 8 CONCLUSION

Based on Intel SGX, this article proposed a privacy-preserving scheme for both 5G positioning and its further application in LBS provision. The scheme has a number of advantages. First, it preserves both UE position privacy and LBSP data privacy. The UE position is not disclosed to FC during position calculation, nor leaked to any LBSPs during LBS provision, thanks to the trustworthiness of TEE offered by the enclaves E1 and E2. In addition, LBSP's data assets are not disclosed to FC even though with E2 access. Thus, LBSP data privacy can also be protected. Second, our scheme simultaneously supports multiple LBSPs by applying secret sharing, which allows UE to initiate LBS by providing a part of the secret share. Holding another part of the secret share issued by UE, the LBSP can be authenticated by the E2 at FC directly. By gaining the two parts of the share provided by UE and LBSP, E2 can obtain UE position to offer LBS to UE. In this way, we realize LBSP authentication at E2 with UE control. Third, the scheme offers a generic framework to preserve both positioning privacy and LBS privacy regarding UE position and LBSP data. It can be easily integrated into current LBS systems to offer LBS at the edge. Security analysis and performance evaluation based on scheme implementation over a real-world database show its sound security and acceptable efficiency, although mutual privacy preservation introduces some extra computation and communication costs.

## REFERENCES


[1] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. 2008. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceedings of the 17th International Conference on World Wide Web*. 237–246.

[2] Alastair R. Beresford and Frank Stajano. 2003. Location privacy in pervasive computing. *IEEE Pervas. Comput.* 2, 1 (2003), 46–55.

[3] Claudio Bettini, X. Sean Wang, and Sushil Jajodia. 2005. Protecting privacy against location-based personal identification. In *Proceedings of the Workshop on Secure Data Management*. Springer, 185–199.

[4] Dan Boneh and Matthew Franklin. 2003. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* 32, 3 (2003), 586–615.

[5] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. 2017. Software grand exposure: SGX cache attacks are practical. In *Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT'17)*.

[6] Yaxing Chen, Qinghua Zheng, Zheng Yan, and Dan Liu. 2020. QShield: Protecting outsourced cloud data queries with multi-user access control based on SGX. *IEEE Trans. Parallel Distrib. Syst.* 32, 2 (2020), 485–499.

[7] Xiangle Cheng, Yulei Wu, Geyong Min, Albert Y. Zomaya, and Xuming Fang. 2020. Safeguard network slicing in 5G: A learning augmented optimization approach. *IEEE J. Select. Areas Commun.* 38, 7 (2020), 1600–1613.

[8] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. 2006. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*. 171–178.

[9] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *IACR Cryptol. ePrint Arch.* 2016, 86 (2016), 1–118.

[10] Shufan Fei, Zheng Yan, Wenxiu Ding, and Haomeng Xie. 2021. Security vulnerabilities of SGX and countermeasures: A survey. *ACM Comput. Surv.* 54, 6 (2021), 1–36.

[11] Bugra Gedik and Ling Liu. 2005. Location privacy in mobile systems: A personalized anonymization model. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. IEEE, 620–629.

[12] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. 2008. Private queries in location based services: Anonymizers are not necessary. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 121–132.

[13] Pascal Gremaud, Arnaud Durand, and Jacques Pasquier. 2019. Privacy-preserving IoT cloud data processing using SGX. In *Proceedings of the 9th International Conference on the Internet of Things*. 1–4.

[14] Siam Umar Hussain and Farinaz Koushanfar. 2018. P3: Privacy preserving positioning for smart automotive systems. *ACM Trans. Des. Automat. Electron. Syst.* 23, 6 (2018), 1–19.

[15] Prerit Jain, Soham Jayesh Desai, Ming-Wei Shih, Taesoo Kim, Seong Min Kim, Jae-Hyuk Lee, Changho Choi, Youjung Shin, Brent Byunghoon Kang, and Dongsu Han. 2016. OpenSGX: An open platform for SGX research. In *Proceedings of the Network and Distributed System Security Symposium*, Vol. 16. 21–24.

[16] Kimmo Järvinen, Helena Leppäkoski, Elena-Simona Lohan, Philipp Richter, Thomas Schneider, Oleksandr Tkachenko, and Zheng Yang. 2019. PILOT: Practical privacy-preserving indoor localization using outsourcing. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P'19)*. IEEE, 448–463.

[17] Han Jiang, Hao Wang, Zhihua Zheng, and Qiuliang Xu. 2019. Privacy preserved wireless sensor location protocols based on mobile edge computing. *Comput. Secur.* 84 (2019), 393–401.

[18] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. 2005. An anonymous communication technique using dummies for location-based services. In *Proceedings of the International Conference on Pervasive Services*. IEEE, 88–97.

[19] Andreas Konstantinidis, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, Paschalis Mpeis, Nikos Pelekis, and Yannis Theodoridis. 2015. Privacy-preserving indoor localization on smartphones. *IEEE Trans. Knowl. Data Eng.* 27, 11 (2015), 3042–3055.

[20] John Krumm. 2009. A survey of computational location privacy. *Person. Ubiq. Comput.* 13, 6 (2009), 391–399.

[21] Vaibhav Kulkarni, Bertil Chapuis, and Benoît Garbinato. 2017. Privacy-preserving location-based services by using Intel SGX. In *Proceedings of the 1st International Workshop on Human-centered Sensing, Networking, and Systems*. 13–18.

[22] Do Le Quoc, Franz Gregor, Jatinder Singh, and Christof Fetzer. 2019. SGX-PySpark: Secure distributed data analytics. In *Proceedings of the World Wide Web Conference*. 3564–3563.

[23] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. 2014. Achieving privacy preservation in WiFi fingerprint-based localization. In *Proceedings of the IEEE Conference on Computer Communications*. IEEE, 2337–2345.

[24] Yilin Li, Shushu Liu, Zheng Yan, and Robert H. Deng. 2021. Secure 5G positioning with truth discovery, attack detection and tracing. *IEEE J. Internet Things* (2021). DOI: 10.1109/JIOT.2021.3088852

[25] Shushu Liu, An Liu, Zheng Yan, and Wei Feng. 2019. Efficient LBS queries with mutual privacy preservation in IoV. *Vehic. Commun.* 16 (2019), 62–71.

[26] Shushu Liu and Zheng Yan. 2020. Verifiable edge computing for indoor positioning. In *Proceedings of the IEEE International Conference on Communications (ICC'20)*. IEEE, 1–6.

[27] Shushu Liu and Zheng Yan. 2021. Privacy-preserving D2D cooperative location verification. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'21)*. IEEE.
[28] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. 2006. The new Casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases*. 763–774.
[29] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. 2020. A survey of published attacks on Intel SGX. *arXiv preprint arXiv:2006.13598* (2020).
[30] Hyunyoung Oh, Adil Ahmad, Seonghyun Park, Byoungyoung Lee, and Yunheung Paek. 2020. Trustore: Side-channel resistant storage for SGX using Intel hybrid CPU-FPGA. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1903–1918.
[31] Oleksii Oleksenko, Bohdan Trach, Robert Krahn, Mark Silberstein, and Christof Fetzer. 2018. Varys: Protecting SGX enclaves from practical side-channel attacks. In *Proceedings of the Usenix Annual Technical Conference (USENIXATC'18)*. 227–240.
[32] Russell Paulet, Md Golam Kaosar, Xun Yi, and Elisa Bertino. 2013. Privacy-preserving and content-protecting location based queries. *IEEE Trans. Knowl. Data Eng.* 26, 5 (2013), 1200–1210.
[33] Qianwen Pei, Burong Kang, Lei Zhang, Kim-Kwang Raymond Choo, Yuanfei Zhang, and Yinxia Sun. 2018. Secure and privacy-preserving 3D vehicle positioning schemes for vehicular ad hoc network. *EURASIP J. Wirel. Commun. Netw.* 1 (2018), 1–12.
[34] Amir Mahdi Sazdar, Nasim Alikhani, Seyed Ali Ghorashi, and Ahmad Khonsari. 2021. Privacy preserving in indoor fingerprint localization and radio map expansion. *Peer-to-Peer Netw. Applic.* 14, 1 (2021), 121–134.
[35] Lorenz Schauer, Florian Dorfmeister, and Florian Wirth. 2016. Analyzing passive Wi-Fi fingerprinting for privacy-preserving indoor-positioning. In *Proceedings of the International Conference on Localization and GNSS (ICL-GNSS'16)*. IEEE, 1–6.
[36] Felix Schuster, Manuel Costa, Cédric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, and Mark Russinovich. 2015. VC3: Trustworthy data analytics in the cloud using SGX. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 38–54.
[37] Shuo Shang, Ruogu Ding, Kai Zheng, Christian S. Jensen, Panos Kalnis, and Xiaofang Zhou. 2014. Personalized trajectory matching in spatial networks. *VLDB J.* 23, 3 (2014), 449–468.
[38] Shuo Shang, Bo Yuan, Ke Deng, Kexin Xie, Kai Zheng, and Xiaofang Zhou. 2012. PNN query processing on compressed trajectories. *GeoInformatica* 16, 3 (2012), 467–496.
[39] Tao Shu, Yingying Chen, Jie Yang, and Albert Williams. 2014. Multi-lateral privacy-preserving localization in pervasive environments. In *Proceedings of the IEEE Conference on Computer Communications*. IEEE, 2319–2327.
[40] Yulei Wu, Hong-Ning Dai, Hao Wang, and Kim-Kwang Raymond Choo. 2021. Blockchain-based privacy preservation for 5G-enabled drone communications. *IEEE Netw.* 35, 1 (2021), 50–56.
[41] Yulei Wu, Yuxiang Ma, Hong-Ning Dai, and Hao Wang. 2021. Deep learning for privacy preservation in autonomous moving platforms enhanced 5G heterogeneous networks. *Comput. Netw.* 185 (2021), 107743.
[42] Zheng Yang and Kimmo Järvinen. 2018. The death and rebirth of privacy-preserving WiFi fingerprint localization with Paillier encryption. In *Proceedings of the IEEE Conference on Computer Communications*. IEEE, 1223–1231.
[43] Xun Yi, Russell Paulet, Elisa Bertino, and Vijay Varadharajan. 2014. Practical k nearest neighbor queries with location privacy. In *Proceedings of the IEEE 30th International Conference on Data Engineering*. IEEE, 640–651.
[44] Man Lung Yiu, Christian S. Jensen, Xuegang Huang, and Hua Lu. 2008. SpaceTwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Proceedings of the IEEE 24th International Conference on Data Engineering*. IEEE, 366–375.
[45] Mahmoud Youssef, Vijayalakshmi Atluri, and Nabil R. Adam. 2005. Preserving mobile customer privacy: An access control system for moving objects and customer profiles. In *Proceedings of the 6th International Conference on Mobile Data Management*. 67–76.
[46] Ping Zhao, Hongbo Jiang, John C. S. Lui, Chen Wang, Fanzi Zeng, Fu Xiao, and Zhetao Li. 2018. P3-LOC: A privacy-preserving paradigm-driven framework for indoor localization. *IEEE/ACM Trans. Netw.* 26, 6 (2018), 2856–2869.
[47] Kai Zheng, Yu Zheng, Nicholas J. Yuan, Shuo Shang, and Xiaofang Zhou. 2013. Online discovery of gathering patterns over trajectories. *IEEE Trans. Knowl. Data Eng.* 26, 8 (2013), 1974–1988.
[48] Kai Zheng, Xiaofang Zhou, Pui Cheong Fung, and Kexin Xie. 2012. Spatial query processing for fuzzy objects. *VLDB J.* 21, 5 (2012), 729–751.
[49] Rang Zhou, Xiaosong Zhang, Xiaofen Wang, Guowu Yang, Hao Wang, and Yulei Wu. 2019. Privacy-preserving data search with fine-grained dynamic search right management in fog-assisted Internet of Things. *Inf. Sci.* 491 (2019), 251–264.

Received September 2021; revised November 2021; accepted January 2022