Quantifying Location Privacy in Permissioned Blockchain-Based Internet of Things (IoT)

Abdur R. Shahid
SCIS, FIU
Miami, FL
ashah044@fiu.edu

Niki Pissinou
SCIS, FIU
Miami, FL
pissinou@fiu.edu

Laurent Njilla
US-AFRL
Rome, NY
laurent.njilla@us.af.mil

Sheila Alemany
SCIS, FIU
Miami, FL
salem010@fiu.edu

Ahmed Imteaj
SCIS, FIU
Miami, FL
aimte001@fiu.edu

Kia Makki
TUA
Miami, FL
kia.makki@gmail.com

Edwin Aguilar
SCIS, FIU
Miami, FL
eagui044@fiu.edu

ABSTRACT

Recently, blockchain has received much attention from the mobility-centric Internet of Things (IoT). It is deemed the key to ensuring the built-in integrity of information and security of immutability by design in the peer-to-peer network (P2P) of mobile devices. In a permissioned blockchain, the authority of the system has control over the identities of its users. Such information can allow an ill-intentioned authority to map identities with their spatiotemporal data, which undermines the location privacy of a mobile user. In this paper, we study the location privacy preservation problem in the context of permissioned blockchain-based IoT systems under three conditions. First, the authority of the blockchain holds the public and private key distribution task in the system. Second, there exists a spatiotemporal correlation between consecutive location-based transactions. Third, users communicate with each other through short-range communication technologies such that it constitutes a proof of location (PoL) on their actual locations. We show that, in a permissioned blockchain with an authority and a presence of a PoL, existing approaches cannot be applied using a plug-and-play approach to protect location privacy. In this context, we propose **BlockPriv**, an obfuscation technique that quantifies, both theoretically and experimentally, the relationship between privacy and utility in order to dynamically protect the privacy of sensitive locations in the permissioned blockchain.

CCS CONCEPTS

• **Security and privacy → Pseudonymity, anonymity and untraceability**; **Privacy protections**; • **Computer systems organization → Peer-to-peer architectures**.

DISTRIBUTION A. Approved for public release; Distribution unlimited. Case Number 88ABW-2019-3145; Dated 26 Jun 2019

MobiQuitous, November 12–14, 2019, Houston, TX, USA

ACM ISBN 978-1-4503-7283-1/19/11…$15.00
https://doi.org/10.1145/3360774.3360800

KEYWORDS

Blockchain, IoT, Location Privacy, Smart Mobile Devices

ACM Reference Format:
Abdur R. Shahid, Niki Pissinou, Laurent Njilla, Sheila Alemany, Ahmed Imteaj, Kia Makki, and Edwin Aguilar. 2019. Quantifying Location Privacy in Permissioned Blockchain-Based Internet of Things (IoT). In *16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), November 12–14, 2019, Houston, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360774.3360800

1 INTRODUCTION

As of this writing, the mobility-centric Internet of Things (IoT) systems utilize a centralized model to handle the vast amount of data generated by IoT devices (e.g., smart vehicles in the Vehicular Ad Hoc Network (VANET) [7], smartphones in Ad Hoc networking-based mobile crowdsensing [10]). Such models are weak in ensuring security and trust and are not capable of handling the fast-paced growth of IoT. Thus, distributed systems are considered to address the problems of IoT systems. Recently, blockchain, a unique distributed technique, has gained tremendous attention from the IoT community. It is a distributed peer-to-peer (P2P) technique for recording digital interactions in a unique way that is designed to be secure, transparent, highly resistant to outages, auditable, and efficient[13, 22]. It provides built-in integrity of information, and security of immutability by design, making it very useful for ensuring trust, security, and transparency in P2P trustless networks. To date, two main categories of blockchain have been studied in a variety of IoT applications: public and permissioned. In a public blockchain, there exists no authority of the blockchain; a node can join and leave the network at any point with random pseudonyms and can also change its public keys at any time instance (e.g., for every transaction). This frequent pseudonym scheme makes the IoT nodes[1] untraceable and provides high privacy. However, in a permissioned blockchain, such a high level of privacy is not easily attainable, as the authority of the blockchain controls the blockchain network with a variety of access controls spanning from control over joining the network to perform consensus mechanisms. Amazon's Quantum Ledger Database (QLDB)[1], J.P. Morgan's Quorum blockchain [21], and Microsoft's Azure blockchain [4] are just a few examples of industry standard permissioned blockchains.

[1] In this paper, we use the terms IoT node, IoT device, and users interchangeably.

Similar to many other fields, permissioned blockchain is also being studied in the IoT of mobile devices. For better understanding, we draw motivation of a permissioned blockchain from CreditCoin, a privacy-preserving blockchain framework for the Vehicular Ad Hoc Network (VANET) [18]. In this framework, the vehicles are required to be registered with the authority. This authority is responsible for generating and providing the vehicles with cryptographic keys, and keep track of the relationship between the vehicles and the provided keys. A set of trace managers at different locations also aids the authority in tracking malicious vehicles/users. In this framework, only road-side units (RSUs) and authorized vehicles are responsible for managing the blockchain. This framework is built around the short-range communication technology-based P2P network of the vehicles. Here, the vehicles make transactions with their peers such that each transaction is signed by each of the peer vehicles by their public keys. As these transactions are made through a short-range communication technology (e.g., Wi-Fi, Bluetooth), they can be treated as a proof-of-location (PoL) for the vehicles' whereabouts in the spatiotemporal domain. In some frameworks, such as the one proposed in [2], the proof of location is explicitly defined in the design. Based on the transaction information, the vehicles generate a rating about each other and forward them to the nearest RSU. The RSUs then compute the overall rating of each vehicle and append the new rating into the blockchain. Similar motivation can also be drawn from the work presented in [28]. Obviously, these frameworks can be integrated into many other mobility-centric IoT scenarios, such as mobile crowdsensing. The RSUs and smart vehicles can be replaced with Wi-Fi access points and low powered mobile devices (e.g., smart phones and smart watches), respectively. In terms of location privacy, these frameworks only guarantee conditional privacy to IoT users. That is, the devices can enjoy privacy from their peers by using the public keys provided by the authority. However, as the authority holds the mapping between real identity and the public keys, the privacy of sensitive locations from a malicious authority cannot be preserved using only a key changing mechanism. A malicious authority can perform a spatiotemporal analysis of the disclosed locations of a user and can reveal sensitive information.

In this paper, we study the location privacy issue in the context of permissioned blockchain, where: (1) the authority of the blockchain holds the public and privacy key distribution task in the system, (2) a transaction can be considered as a proof of location (PoL) for a user's temporal whereabouts, and (3) there is a spatiotemporal correlation between the locations. We make the following key contributions: (1) We first discuss the limitations of existing location privacy-preserving mechanisms under a PoL in the context of permissioned blockchain. (2) We present an effective solution, called **BlockPriv**. As discussed above, in **BlockPriv**, the worst form of privacy leakage is considered. That is, whenever an IoT node makes a transaction with its peers, its location information is known to the malicious blockchain authority and the authority is completely capable of mapping the real identity of a node with its public key pairs. Taking a node's privacy preference for different locations and spatiotemporal correlation between the transactions, **BlockPriv** decides whether or not a node should make a transaction, such that its undisclosed sensitive location's privacy is also preserved with a set of locations. (3) We quantify the trade-off between privacy and utility theoretically and empirically using two factual datasets.

The rest of the paper is organized as follows. Related works are discussed in section 2. The overview of the system and its design goals are presented in section 3. Then, the proposed **BlockPriv** approach is detailed in section 4. Important security, privacy, and utility aspects of **BlockPriv** are analyzed in section 5. A discussion of the experimental analysis is covered by section 6. Finally, the paper is concluded in section 7. Important notations used in the paper are presented in Table 1.

2 RELATED WORK

Location privacy preservation is a comparatively well studied problem in centralized architecture-centric IoT systems. Several classes of mechanisms have been proposed to mitigate the privacy leakage, such as (1) pseudonym, (2) location perturbations, and (3) spatial obfuscation. The goal of these mechanisms is to apply them to a node's actual location before releasing it to the central authority. For instance, in the case of a pseudonym, before revealing the location, the mechanism changes the ID of a node to make it untraceable [29]. These approaches depend on a trusted third party (TTP) to carry out the steps of changing pseudonym. This is similar to the mixing approach [8] used in blockchain to improve privacy by exchanging the public key of a mobile node with a random public key such that the probability of linking multiple transactions is reduced. However, in a permissioned version of blockchain such an approach will not work.

Perturbation mechanisms, such as differential privacy-based geo-indistinguishability [3], add statistical noise to a node's real location before it is shared with the system. Obviously, under a PoL, such mechanisms have limited impact [19]. On the other hand, spatial obfuscation reduces the precision of the actual location information before releasing it to the authority of the system. This is done by either infusing more locations[17] or replacing the actual location with a realistic larger region[14]. Similar to location perturbation, location obfuscation works only at a limited scale under the PoL. In a nutshell, the existing privacy-preserving mechanisms, designed for centralized IoT systems, cannot be applied in a plug-and-play way to the problem that we are trying to solve here.

In the scope of blockchain, the frequent change of public keys is the most explored solution to preserve privacy[11, 22, 25, 30]. It was first proposed by Nakamoto [22], the creator of Bitcoin. Motivated by Bitcoin's solution, Dorri et al. [11] also suggested to use a fresh unique public key to prevent linkage attacks while communicating with other nodes in their proposed Lightweight Scalable Blockchain (LSB) architecture for smart-vehicle ecosystems. In blockchain-based centralized proof-of-location (PoL) generation, Brambilla et al. [9] also proposed changing the public keys frequently to preserve a node's sensitive location privacy while generating proof of locations. Michelin et al. [19] proposed a privacy-preserving blockchain-based SpeedyChain framework for a vehicular network scenario. Similar to most of the other works in this context, SpeedyChain considers the fixed positioned roadside infrastructure units (RSIs) as the key to maintaining the blockchain. Unlike Bitcoin or Ethereum-like blockchains, here, for each vehicle there exists exactly one block in the blockchain. In order to maintain privacy,

this framework proposes the timely change of the public key of each vehicle. However, these frameworks do not fit completely into the scenario considered in this paper, where the authority of the blockchain controls the private and public key distributions to the mobile nodes in the system.

The idea of a permissioned blockchain is primarily stemed from the evidence of misuse of freedom in public blockchains for illegal activities. For instance, almost half of the bitcoin transactions are estimated to be related to illegal drug sales, ransomware, and other malicious activities[12]. Hence, the deanonymization of the blockchain users has gained significant attention from both the law enforcement and the security and privacy communities. In fact, it is found that changing the public keys in order to nullify a linking attack in a public blockchain is not quite as bulletproof as it was expected [6, 16]. Research efforts show that it is possible to map the public keys of Bitcoin users to their unique identities (e.g., IP addresses) [6, 16]. For instance, Koshy et al.[16] were able to deanonymize 1162 addresses by analyzing transaction relaying patterns. Biryukov et al. [6] proposed a deanonymizing algorithm by exploiting only the input and output transactions of mixing services and identified a relationship between the input and output addresses at a very high accuracy. Recently, Roulin et al. [23] applied decision tree algorithms on smart home devices' data (e.g., smart things, nest smoke alarm) by utilizing off-chain information to classify IoT devices for understanding a user's activity pattern. While the work is done in the context of smart home, it can be adapted for the mobility of the IoT devices. All these deanonymization works highlight that simply changing the public keys frequently is not the ultimate solution to providing privacy in the blockchain, even in a public version.

Moving forward, our work is focused on an authority-based permissioned blockchain where privacy is tougher to achieve by default. It is closely related to the work proposed by Li et al. [18] in the context of a vehicular network. Using their proposed framework, it is possible to achieve only conditional privacy, as the trace manager can track anyone at any time, if necessary. Similarly, Yang et al. [28] presented a blockchain-based decentralized trust management framework for vehicles where each vehicle is registered with the system using its VIN number. Thus, only conditional privacy can be attained with this framework. Likewise, Sharma et al. [24] proposed a permissioned blockchain by incorporating traceability features while maintaining privacy in the Internet of Vehicles (IoV). However, they used a server for vehicle registration, which would store all vehicle IDs in an encrypted scheme; the central authority can track any vehicle when needed.

To achieve complete location privacy, Yang et al. [27] proposed an obfuscation approach to protect location privacy in a private blockchain for crowdsensing applications. In this work, a worker submits an obfuscated region to the system to protect their exact location's privacy. However, in the case of P2P communication of the nodes, this type of approach cannot be applied without the collaboration of the nodes. Jia et al. [15] designed a blockchain-based incentive mechanism for crowdsensing applications with a focus on preserving the location privacy of the users. In their framework, a confusion layer was proposed, in which a user's location is encoded in such a way that it can be confused with other $k-1$ users' locations. While this could be a solution to protect location privacy, it requires the honest collaboration of other users.

In contrast to all these works, we intend to design a location privacy-preserving obfuscation mechanism that does not require collaboration from other users and can provide complete privacy in permissioned-blockchain under the presence of PoL.

Table 1: Notations and Their Description

Notation	Description
MU	Mobile user or mobile node
$\mathcal{N}_x$	Privacy parameter for a location l_x
$\mathcal{P}(l_h)$	Privacy level achieved for location l_h
$Pr^t_{MU}(l)$	MU's probability of being at location l at time t
l^s	A sensitive location
S	Set of all sensitive locations of a MU
$\mathcal{U}(l)$	Loss of utility for location l
T_r	A trajectory
n	Total number of sensitive locations in a T_r
δt	Time difference
$\mathcal{L}_a$	Set of all locations reachable to/from location l_a
$\Phi(a, b)$	Required time to reach from location l_a to l_b
$\|X\|$	Size/number of elements in a set X
α	% of location types selected as sensitive
r	Privacy region radius

3 SYSTEM OVERVIEW AND DESIGN GOALS

In this section, we present the details of the system model and the behavior and attack strategies of the malicious entities in the system. We then formulate the central problem of this paper and state the goals we set out to achieve in the design of its solution.

3.1 Blockchain System Model

We consider a permissioned blockchain, where its authority also acts as the certificate authority to provide the public and private key pairs to the mobile nodes. The mobile nodes are registered with the system and communicate with each other using the preassigned key pairs. Communication between the nodes takes place using a short-rage communication technology. The nodes can request the authority for new key pairs at any point of time. The blockchain is managed by preassigned mobile edge computing devices (e.g., RSU, Wi-Fi access points, and so on), distributed over a large region. These devices constitute the blockchain nodes and are connected with each other in a P2P network over the internet. The transactions among the IoT nodes are broadcasted to the blockchain nodes in the blockchain network. The blockchain nodes aggregate and insert the new transactions into the blockchain through a consensus mechanism (e.g. practical byzantine fault tolerance, proof-of-stake) in a timely fashion (e.g. every 30 minutes). We consider a blockchain architecture similar to the one presented in CreditCoin [18] without considering the rewarding phase. We assume that the mobile nodes have internet capability to compute the time to reach one location from another with the help of a traffic information provider in real time, e.g., Google Maps. We also assume that the information between the traffic information service provider and a node is

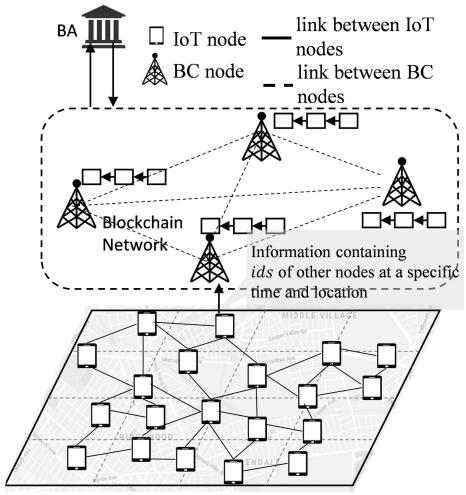


Figure 1: System model of permissioned-blockchain where BC and BA refer to blockchain and blockchain authority, respectively. The BA also acts as certificate authority and trace manager. The mobile IoT nodes are connected with each other in a P2P network using a short-range communication technology. They make transactions with each other and send information on the transactions (e.g., rating about other mobile nodes at a specific location and time) to the nearest blockchain node. Here, each grid refers to a specific location.

anonymous and the provider is independent from the blockchain authority.

3.2 Malicious Entities and Attack Strategies

3.2.1 Malicious Entities. In the system, we consider the authority of the blockchain as the malicious entity. It follows the honest-but-curious adversary model in the system. That is, it tries to predict a target node's sensitive spatiotemporal information without violating any protocol of the system or dismantling the way blockchain works. Furthermore, it is not going to hack into the device of a target node. We also consider that, in order to compute the time reachability information, the authority also uses a traffic information service provider. From this point on, we refer to the authority as an attacker. It is important to note that some of the mobile nodes can be malicious. However, as we mentioned earlier in the system model, the mobile nodes can change their public keys at any point of time; the malicious mobile nodes cannot track a target node from their transactions without colluding with the authority. This is a fundamental privacy feature of blockchains. Thus, we focus on the attack strategies of the blockchain authority.

3.2.2 Attacker's Goal and Strategies. The goal of an attacker is to understand a mobile node's presence at different locations in the temporal domain. In order to do so, it utilizes the time-reachability-based spatiotemporal correlation between a node's disclosed locations in the blockchain as its fundamental strategy. Let the random variable O^t_{MU} represents the actual location of a mobile node MU at time t. Given a node's locations l_i, l_j at time t_a, t_b respectively, the node's probability of being at a location l_h at a discrete time t_q $(t_a < t_q < t_b)$ is

$$Pr^q_{MU}(l_h) = \mathbb{P}\mathrm{r}(O^q_{MU} = l_h | O^a_{MU} = l_i, O^b_{MU} = l_j) \tag{1}$$

The attacker computes $Pr^q_{MU}(l_h)$ using the time reachability correlation as follows.

$$Pr^q_{MU}(l_h) = \begin{cases} 1 & \text{If } l_h \text{ is reachable to and from } l_i \\ & \text{and } l_j \text{ in } (t_b - t_a) \text{ time} \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

Obviously, it is possible to have multiple locations with $Pr^q_{MU}(l_h) = 1$. Thus, the ultimate goal of the attacker is to minimize the number of such locations. That is,

$$\text{minimize} \left(\sum Pr^q_{MU}(l_h) \right) \tag{3}$$

This forms the core of an attacker's strategy. Based on this, we consider mainly the following attacks that can be exploited by the attacker to infer a target node's location information.

(1) *Collusion with malicious mobile nodes*: Malicious nodes collude with the authority and provide it with the location information of a target node for profit.

(2) *Map matching attack*: The authority employs the map information to understand spatially reachable and unreachable location information. A spatially unreachable location refers to a location that cannot be reached at any time using a map service (e.g., the middle of a lake). Thus, $Pr^{\infty}_{MU}(l) = 0$.

(3) *Time-reachability-based path reconstruction attack*: In order to reconstruct the actual path between two revealed locations, the authority can use the time reachability information to construct the valid paths that can be traveled between the two locations within a time limit.

We also analyze the impact of transaction dropping attack on location privacy. Note that, the scope of this paper encompasses the analysis of location privacy invading attacks from a user's point of view and thus different blockchain related attacks, such as DDoS, Sybil, 51% attack, and eclipse attack are not covered here.

3.3 Problem Formulation and Design Goals

It is clear that there is an important trade-off between location privacy and utilization of the system. The problem lies with the short-range communication technology-based transactions between the mobile nodes that form proof of locations (PoL) for the nodes. Thus, in order to protect a sensitive location's privacy, a mobile node must remain silent in the network: that is, it must not make any transaction in the network. This leads to the question of how long in both spatial and temporal domains a node must remain silent to protect a sensitive location's privacy. Remaining silent infinitely results in a location privacy of 100%, but a system utilization of 0%. In other words, an indefinite silence will incur a 100% *loss of utility*. Hence, the goal of this work is to formulate, design, implement, and evaluate a location privacy-preserving mechanism, called **BlockPriv**, for mobile nodes in the context of permissioned blockchain by solving the following problem:

$$\text{minimize } \{\mathcal{P}^{-1}(l^s), \mathcal{U}(l^s)\} \tag{4}$$

Here, $\mathcal{P}(l^s)$ and $\mathcal{U}(l^s)$ refer to the achieved privacy for sensitive location l^s and the loss of utility due to privacy preservation for l^s, respectively.

To summarize, in the design of the **BlockPriv** mechanism, we intend to achieve the following goals: (1) achieve privacy without collaborating with any other entity in the system, and (2) achieve a quantifiable balance between privacy and utility.

4 THE BLOCKPRIV APPROACH

For the sake of clarity and to maintain coherence with the blockchain concept, we first discuss the public key changing technique adapted in **BlockPriv**. In our scheme, we adapt the temporal public key changing concept proposed by Michelin et al. [19]. Here, at a fixed time interval t^{key}, a mobile node will change its public key in order to nullify the possibility of spatiotemporal linkage attack from malicious nodes. Note that, in our problem, public key changing can only provide privacy to a mobile node against its peers, not against the authority that distributes the keys. Also, this scheme is vulnerable against colluding attack between the authority and malicious mobile nodes, which is one of the focus of our work.

At this point, we present the formal definition of location privacy and utility from the perspective of a mobile node. The definition of privacy can be derived from the formulation of the attacker's objective, defined by equation 3, as follows.

$$\mathcal{P}(l^s) = \text{maximize}\left(\sum Pr^q_{MU}(l_h)\right) \tag{5}$$

Let us consider: a node's last revealed location in the blockchain is l_i at time t_a, and it was at a sensitive location l^s_h at time t_q. It should reveal its location, also known as making a transaction, at an insensitive location l_j at time t_b ($t_a < t_q < t_b$) if and only if

$$\mathcal{P}(l^s_h) = \left(\sum Pr^q_{MU}(l^s_h)\right) \geq \mathcal{N}_h \tag{6}$$

To explain, a node should reveal its location l_j at time t_b in the network to the authority when there exists at least $\mathcal{N}_q$ number of locations, including l^s_h, which are both reachable from and to l_i and l_j in ($\delta t = t_b - t_a$) time. Here, $\mathcal{N}_h$ is a user defined privacy parameter for location l^s_h. This formulation is applicable only for a single sensitive location. It is also possible that, after l^s_h, the node was also at another sensitive location l^s_p at time t_r ($t_a < t_q < t_r < t_b$) such that, after $\delta t = t_b - t_a$ time, $\mathcal{P}(l^s_h) \geq \mathcal{N}_h$, but $\mathcal{P}(l^s_p)) < \mathcal{N}_p$. In such a case, the node should not make any transaction at location l_j at time t_b. Formally, if there are m number of sensitive locations visited by a node between time t_a and t_b, then it will make a transaction with its peers at an insensitive location at time t_b in the network if and only if

$$\mathcal{P}(l^s_i) = \left(\sum Pr^{q_i}_{MU}(l^s_i)\right) \geq \mathcal{N}_i; \quad \forall i = 1, \ldots m \tag{7}$$

Note that, from t_a to t_b, the node was continuously silent in the network. We call it single or 1 round silence to maintain privacy of the m number of sensitive locations. If a trajectory T_r contains n number of sensitive locations, then the average privacy of each sensitive location in that trajectory is defined as

$$\mathcal{P}(T_r) = \frac{1}{n}\sum_i \mathcal{P}(l^s_i), \quad i = 1, \ldots n \tag{8}$$

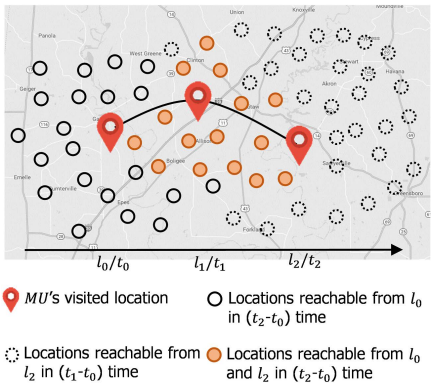


Figure 2: Illustrated BlockPriv: The curve refers to a mobile node (MU)'s actual path between l_0, l_1, and l_2 locations at times t_0, t_1, and t_2, respectively. The location l_1 is privacy sensitive for the MU. Thus, it remained silent at location l_1. It will make a blockchain transaction at l_2 at time t_2 only when the number of locations reachable from both l_0 and l_2 in $t_2 - t_0$ time, meets the privacy requirement for l_1.

From the formulation of privacy, we can also define the loss of utility due to the application of privacy preservation. Let us consider: at i-th round silence, the node opted not to make any transaction at $\mathcal{P}(l^s_h)$ number of locations. In our definitions, this number is the loss of utility of **BlockPriv**. If a node maintained k rounds of silence to preserve privacy of a trajectory T_r with n number of sensitive locations, then the average loss of utility for each sensitive location is

$$\mathcal{U}(T_r) = \frac{1}{n}\sum_{i=1}^{i=k} \mathcal{U}_i \tag{9}$$

This allows us to reconstruct the multi-objective optimization problem, presented in equation 4, as a single objective optimization problem as follows:

$$\begin{aligned} &\text{minimize } \mathcal{U}(T_r) \\ &s.t. \quad \mathcal{P}(l^s_i) \geq \mathcal{N}_i; \forall l^s_i \in T_r \end{aligned} \tag{10}$$

Now we present in detail the mechanism of **BlockPriv** to solve this problem.

In this mechanism, the mobile nodes are responsible for labeling their sensitive locations and assigning level of privacy to each of them. The nodes utilize radius r to specify the level of privacy for a sensitive location as $\mathcal{N} = \pi r^2$. Let us consider: a node MU made a transaction in the network at time t_a at location l_i. Then, it moved to a privacy sensitive location l^s_h at time t_q and did not make any transactions. Then, after every Δt time at location l_j, different from both l_i and l^s_h, it checks the number of locations that are reachable to and from l_i and l_j. Let current time and location be t_b and l_j, respectively. The node first computes the set of all the locations $\mathcal{L}_i$ that are reachable from l_i in $\delta t = t_b - t_a$ time. Next, it computes the set of all the locations $\mathcal{L}_j$ from which location l_j is reachable. Then, $\mathcal{L} = \mathcal{L}_i \cap \mathcal{L}_j$ forms the set of all locations from which both l_i and l_j is reachable in δt time. In other words, each of the location in $\mathcal{L}$ creates a valid 1-hop route from l_i to l_j in δt time. That is, based

on the time reachability information, the node can move from l_i to any location $l_l \in \mathcal{L}$ and then move to l_j in δt time. Thus,

$$\mathcal{L} = \{\forall l | (\Phi(l_i, l) + \Phi(l, l_j)) \leq \delta t\} \tag{11}$$

Here, $\Phi(a, b)$ refers to the time to get from location a to b. The size of $\mathcal{L}$ defines the privacy level achieved for sensitive location l_h^s in δt time. That is, $\mathcal{P}(l_h^s) = |\mathcal{L}|$. The node will make a transaction at time t_j at location t_b only when $|\mathcal{L}| \geq \mathcal{N}_h$. If there is a total m number of sensitive locations visited by the node in δt time, according to equation 7, it will make a transaction at time t_j and location l_b if and only if

$$|\mathcal{L}| \geq \mathcal{N}_i; \quad \forall i = 1, \ldots m \tag{12}$$

It is understandable that, in the case when all the sensitive locations have the same level of privacy, comparing $\mathcal{L}$ with the level of privacy of the latest sensitive location is enough to check whether the condition in equation 7 is valid. However, for sensitive locations with different levels of privacy, the *MU* is required to check whether all the previous sensitive locations' levels of privacy are met before making any transaction.

For a single sensitive location l^s, the maximum loss of utility $\mathcal{U}_{max}(l^s)$ is bounded by the value of its privacy parameter $\mathcal{N}$. The higher the value of $\mathcal{N}$, the higher the $\mathcal{U}_{max}(l^s)$. More specifically, $\mathcal{U}_{max}(l^s) \leq \mathcal{L}$. Certainly, from equation 10, we do not want any "extra" loss in utility of the blockchain. Let t_a be the last time a node's location was revealed in the blockchain. After that, at every Δt ($\Delta t \in \mathbb{Z}_{\geq 0}$) time, it computes $\mathcal{L}$ and checks whether it meets the privacy requirement of a set of sensitive locations. That is, after checking $\mathcal{L}$ at time $(t_a + x \times \Delta t)$, it will check $\mathcal{L}$ at time $(t_a + (x+1) \times \Delta t)$. Here, $x \in \mathbb{Z}_{\geq 0}$. Let, t', where $(t_a + x \times \Delta t) < t' < (t_a + (x+1) \times \Delta t)$ is the time when $\mathcal{L} \simeq \mathcal{N}$. Then, computing $\mathcal{L}$ at $(t_a + (x+1) \times \Delta t)$ time will certainly impose some extra loss of utilities. Thus, $\mathcal{U}_{max}(l^s) \leq \mathcal{N} + \mathcal{U}'$. Here, $\mathcal{U}'$ refers to the set of insensitive locations at which the *MU* opted not to make any transaction between time t' and $(t_a + (x+1) \times \Delta t)$. With the higher value of Δt, the value of $\mathcal{U}'$ will be higher. Thus, Δt should remain as small as possible. However, for resource-constrained mobile nodes, a small Δt means very frequent computation of the time reachability, which affects the energy of the device. Thus, the compromise between the capability of the device and loss of utility is an issue that needs to be examined: we leave it for our future work. The detail of **BlockPriv** is presented in Algorithm 1.

Algorithm 1: BlockPriv

Input: Current location l_{cur}, current time t_{cur}, last revealed location in the blockchain l_{prev} and time t_{prev}, list of sensitive locations S, list of level of privacy for the sensitive locations $\mathcal{N}$, previous time of key change t_{prev}^{key}, key expiration time t^{key}

Output: Decision on making transactions.

```
if (t_cur − t_prev^key) ≥ t^key then
    Request new key pair from the authority.
    t_prev^key = t_cur
if l_cur is a sensitive location then
    Append l_cur to S and do not make any transaction.
else
    δt ← t_cur − t_prev
    L_prev ← select all the locations that are reachable from
      l_prev in δt time
    L_cur ← select all the locations from which l_cur is
      reachable in δt time
    L ← L_prev ∩ L_cur
    for (i = 1; i ≤ |S|; i++) do
        if |L| ≥ N(l_i^s ∈ S) then
            Delete l_i^s from S
    if S ≠ ∅ then
        Do not make any transactions in the network.
    else
        Free to make transactions.
```

5 SCHEME ANALYSIS

In this section, we present an analysis of the important privacy, utility, and security aspects of **BlockPriv**.

5.1 Privacy Analysis

5.1.1 Privacy Bound.

Lemma 5.1. *If there are multiple numbers of sensitive locations between two revealed insensitive locations, then each of the sensitive locations achieves a privacy level of* $(\max \mathcal{N})$.

Proof. Let us suppose that a mobile node *MU* has visited m number of sensitive locations between l_{prev} and l_{cur} in $\delta t = (t_{cur} - t_{prev})$ time. According to equation 12, it will make a transaction at location l_{cur} and time t_{cur} only when all of the sensitive locations' privacy requirements are met. That is, a new transaction will take place only when the length of the set $\mathcal{L} \geq (\max \mathcal{N} = \max\{\mathcal{N}_1, \ldots \mathcal{N}_m\})$. Thus, even if a sensitive location's privacy requirement is much lower than $(\max \mathcal{N})$, the achieved privacy for i-th sensitive location l_i^s in the set is $\mathcal{P}(l_i^s) = |\mathcal{L}| \geq (\max \mathcal{N})$. □

5.1.2 Obfuscating Paths.

Lemma 5.2. *If there are any sensitive locations between two revealed insensitive locations l_i and l_j, then, at a minimum, there are* $(\max \mathcal{N})$ *number of* 1*-hop obfuscating paths between the two revealed locations.*

Proof. Equation 11 implies that each location in the set $\mathcal{L}$ is reachable to and from l_{priv} and l_{cur} in δt time. Thus, from the point of reachability, each i-th location in $\mathcal{L}$ forms a 1-hop path between l_{priv} and l_{cur} in δt time. As a result, each path formed by each sensitive location $l_i^s \in \mathcal{L}$ is obfuscated with $(|\mathcal{L}| - 1)$ number of different other paths in δt time. □

5.2 Utility Analysis

5.2.1 Loss of Utility Bound.

Lemma 5.3. *If there are multiple numbers of sensitive locations between two revealed insensitive locations, then the maximum loss of utility $\mathcal{U}_{max}(l^s)$ in **BlockPriv** to preserve privacy of a sensitive location l^s is proportional to* $(\max \mathcal{N})$.

Proof. Lemma 5.1 states that whatever the expected level of privacy assigned to a specific sensitive location, the achieved privacy is bounded by the location with highest level of privacy $\max \mathcal{N}$. Thus, the maximum loss of utility for every sensitive location l^s between the two revealed insensitive locations is $\mathcal{U}_{max}(l^s) \leq (\max \mathcal{N}) + \mathcal{U}'$. □

5.3 Security Analysis

We analyze the efficacy of **BlockPriv** against different location privacy invading strategies by a malicious authority of the blockchain system list in subsection 3.2. We also briefly discuss the interesting impact of the transaction dropping attack on location privacy.

5.3.1 Collusion Attack.

Definition 5.4. A collusion with malicious mobile nodes is successful if the authority of the blockchain can find a new set of locations $\mathcal{L}^*$ about a MU's sensitive location l_i^s such that

$$|\mathcal{L} \cap \mathcal{L}^*| < \mathcal{N}_i. \tag{13}$$

Lemma 5.5. *A combination of time reachability information and collusion with other malicious nodes will not leak privacy of a target mobile node.*

Proof. In **BlockPriv**, a mobile node remains silent in the spatial and temporal domains in order to preserve privacy against an untrusted authority of the blockchain. Thus, even if the authority colludes with some mobile nodes, it will not be able to construct a new set $\mathcal{L}^*$ beyond $\mathcal{L}$ that would satisfy equation 13. In other words, its understanding about a targeted node's whereabouts will not be made any finer than $\mathcal{L}$ by colluding with other nodes. In fact, collusion with mobile nodes to track a target node is a costly approach. The target node changes its public keys frequently and to keep tracking it, the authority needs to update the colluding nodes at the same rate. The only way a colluding attack will be successful is if a malicious node physically tracks a target node. However, our work concentrates on providing security against software-based privacy invading techniques, not on physical observations. □

5.3.2 Map Matching Attack.

Definition 5.6. For a sensitive location l^s, a map matching attack is considered to be successful if a attacker can find a set of locations $\mathcal{L}^*$ from $\mathcal{L}$ such that, $(\mathcal{L}^* \subset \mathcal{L}^*)$, $(|\mathcal{L}^*| > 0)$, and $Pr^{\infty}_{MU}(l_i) = 0; \forall l_i \in \mathcal{L}^*$.

Lemma 5.7. ***BlockPriv** is resilient against map matching attack.*

Proof. The mobile node calculates the time reachability information using a real-time map service provider and thus each location l, selected to form $\mathcal{L}$, is spatially reachable. That is, $\mathcal{L} = \{\forall l \in \mathcal{L} | Pr^{\infty}_{MU}(l) = 1\}$. Thus, $\mathcal{L}^* = \emptyset$. □

5.3.3 Time Reachability-Based Path Reconstruction Attack.

Definition 5.8. A time reachability-based path reconstruction attack on **BlockPriv** is said to be successful if, for a sensitive location l^s, the authority can find fewer than $\mathcal{N}$ number of paths between two revealed locations for a mobile node.

Lemma 5.9. ***BlockPriv**, is resilient against time reachability-based path reconstruction attack.*

Proof. According to equation 11, every location $l_i \in \mathcal{L}$, including every sensitive location, is reachable from previously revealed location l_{prev} to l_{cur} in δt time. Thus, according to lemma 5.2, there are at least $\max \mathcal{N}$ number of 1-hop obfuscating paths from l_{prev} to l_{cur} for l_i.

We can now generalize the analysis for multi-hop paths. Let the actual path be: $l_{prev} \rightarrow l_1^s \rightarrow l_2^s \rightarrow l_{cur}$ and the temporal sequence of this path be: $t_{prev} \rightarrow t_1 \rightarrow t_2 \rightarrow t_{cur}$. Hence, $\delta t = \Phi(l_{prev}, l_1^s) + \Phi(l_1^s, l_2^s) + \Phi(l_2^s, l_{cur})$. Assume that, using **BlockPriv**, we got $\mathcal{L}$, where $\{l_1^s, l_2^s\} \in \mathcal{L}$. For the sake of argument, let us consider, for every location $l \in \mathcal{L}'$ ($\mathcal{L}' = \mathcal{L} \setminus \{l_1^s, l_2^s\}$), there exists no multi-hop path. In such a case, if somehow it is known that the node visited multiple locations between l_{prev} and l_{cur}, then the attacker can exclude all the single hop paths and is able to reconstruct the actual path: $l_{prev} \rightarrow l_1^s \rightarrow l_2^s \rightarrow l_{cur}$. However, in **BlockPriv**, the node remains silent in the network, such that every location in $\mathcal{L}$ exhibits similar probability of being the node's whereabouts under the time reachability condition. Also, such a special case can occur only when $Pr^{\infty}_{MU}(l) = 0; \quad \forall l \in \mathcal{L}'$. This case falls into the category of a map matching attack and lemma 5.7 proves that **BlockPriv** is resilient against such an attack. Hence, time reachability information cannot help a malicious authority to reconstruct the actual path.

□

5.3.4 Transaction Dropping Attack.

In this attack, a mobile node MU_i attempts to drop the transactions between itself and another node MU_j for a specific intention (e.g. preventing the other node from gaining reward out of ill intention or to protect its instance location privacy). There are two cases to consider here. First, MU_j passes the transaction information to the nearest blockchain node and thus MU_i's location information is revealed. In such a case, MU_i's attempt to protect location privacy will fail. Second, if MU_j also drop the transaction, then both the nodes' location information will remain undisclosed in the blockchain.

5.3.5 Security Limitations.

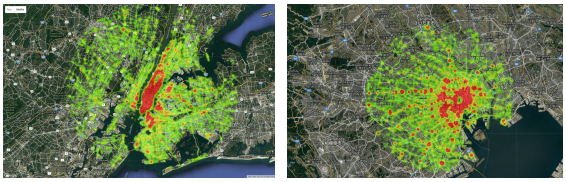
We are also interested in exploring the following security limitations of **BlockPriv** in the future extension of the work.

(1) *Off-Chain Information-Based Attack.* The attacker can combine off-chain information (e.g., information about the hours of operation of a business) with the map matching attack to devise a better inference model.

(2) *Probabilistic Inference Attack from On-Chain Information.* The attacker can personalize the mobility of the node from the information available on the chain using machine learning algorithms (e.g., Markov chains[20]). Such a model can be exploited to improve the path reconstruction attack.

6 EXPERIMENTAL EVALUATION

In this section, we describe the details regarding the experimental evaluation of **BlockPriv**. To properly understand the efficiency and efficacy of our approach, we implemented two cases: locations with (1) similar privacy parameter and (2) different privacy parameters. These two versions will be referred to as **sim-BlockPriv** and **diff-BlockPriv**, respectively.



(a) New York City (NYC) (b) Tokyo (TKY)

Figure 3: Locations in (a) New York City (NYC) and (b) Tokyo (TKY) datasets. Green markers symbolize the locations. The red colors represent the high density regions.

Table 2: Dataset Statistics

Dataset	#Transactions*	#Locations	#Types	#Nodes*
NYC	227428	38333	400	1083
TKY	573703	61858	385	2293

*Originally called "Checkins" and "Users". In this context, we renamed the variables "Transactions" and "Nodes", respectively.

Table 3: Simulation Setup Parameters

Parameter	Value(s)
r	{500, 1000, 1500, 2000} meters
γ	{5,10,15,20}
v	30 miles per hour
α	{2, 4, 6, 8, 10}
n	100

6.1 Experimental Settings

6.1.1 Dataset Description. In this paper, we consider the case of making frequent transactions in the network. Hence, we selected Foursquare's New York City (NYC) and Tokyo (TKY) datasets [26] to test the approach with factual data. These datasets contain the check-in information of nodes, in terms of location and time. The number of transactions, locations, location types, and nodes of the datasets are presented in Table 2 and a visualization of the locations in the datasets are depicted in Figure 3.

6.1.2 Simulation Setup. The datasets do not contain any mark on the privacy sensitive locations of the mobile nodes. Thus, we mark α% of the location types as sensitive locations for all the nodes. The different values of the parameters, including privacy level for a sensitive location r, used in the experiment, are shown in Table 3. For each combination of the parameters, we ran the simulation on both datasets for n number of nodes. As there is a correlation between the number of transactions and the impact of privacy on utility, we selected 100 nodes with the highest number of transactions. We justify this claim through comparing the result with 100 nodes with least number of transactions. Next, since the datasets do not contain continuous location information, we set a speed (v) for each node to simulate its reachability-based mobility. By nature of mobility, there are cases when a node cannot reach a new location, l_{new}, from a previous location, l_{prev} in a certain time, in the dataset with speed v. In these cases, we continue adding a small value to v (e.g. $v/5$) until it can reach l_{new}. In diff-BlockPriv,

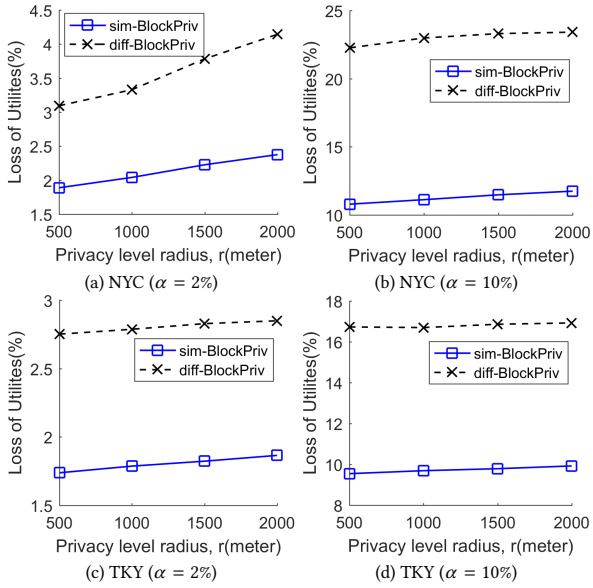


Figure 4: Average loss of utilities versus privacy level in sim-BlockPriv and diff-BlockPriv.

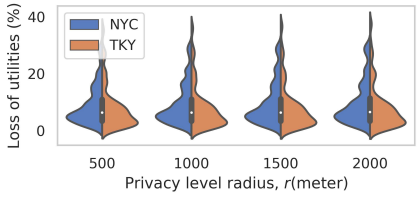


Figure 5: Distribution of loss of utilities in sim-BlockPriv, regarding different privacy levels.

the difference in the privacy level for different sensitive locations is set by drawing a random number from the range $\{r - (r \times \gamma\%),\ r + (r \times \gamma\%)\}$.

6.2 Experiment Results

In the experiment, we examine the loss of utility of sim-BlockPriv and diff-BlockPriv. In particular, we examine the following two relationships, fundamental to the design of a privacy-preserving mechanism: (1) loss of utility versus privacy level, and (2) loss of utility versus number of sensitive locations.

6.2.1 Utility versus Privacy Level. We first examine the relationship between the loss of utility and privacy (in term of radius r in meters). For example, Figures 4(a-d) visually show this relationship for both sim-BlockPriv and diff-BlockPriv when there are a few number of sensitive locations(α = 2%) and a significant number of sensitive locations(α = 10%). Each data point in a figure refers to the average of the 100 users of a specific city. From these figures, we can make several important occlusions. First, we can draw a clear comparison between sim-BlockPriv and diff-BlockPriv, regarding the impact of privacy level r on the loss of utilities. From the city level view, for the same value of r, sim-BlockPriv imposes less utility loss than

Table 4: Pearson's Correlation Values

Dataset	Statistics	U-P	U-S
NYC	Minimum	0.75	0.44
	Average	0.94	0.92
	Maximum	1.00	0.99
TKY	Minimum	0.75	0.74
	Average	0.95	0.95
	Maximum	1.00	0.99

U-P: Loss of Utility vs. privacy level

U-S: Loss of Utility vs. sensitive location types

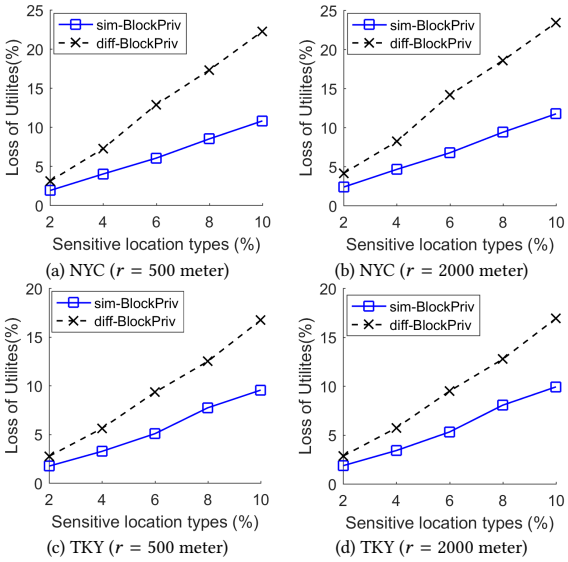


Figure 6: Average loss of utility versus number of sensitive location types (α) in sim-BlockPriv and diff-BlockPriv.

diff-BlockPriv due to the privacy level randomness associated in diff-BlockPriv.

Second, there is an almost linear correlation between the loss of utility and privacy level, regardless of the number of sensitive location types (α) in the dataset. We observe a similar upward trend of loss of utility against the increase in the privacy level for $\alpha = 2\%$ and $\alpha = 10\%$ in both of the datasets. The distribution of loss of utility in Figure 5 further improves the resolution of this linearity. If we look into the exact numeral values, presented in Table 4, the average Pearson's correlation values [5] are 0.94 and 0.95 for the NYC and TKY datasets, respectively. Such linear correlation and lower loss of utility give sim-BlockPriv an upper hand in designing a user-centric privacy scale, which we intend to explore in our extension of this work.

6.2.2 Utility versus Number of Sensitive Location Types. We then analyze the correlation between loss of utility and number of sensitive location types (α). While the analysis of the relationship between utility and privacy level show that the sim-BlockPriv charges less utility loss than diff-BlockPriv, the correlation between utility and number of sensitive location types further signifies the superiority

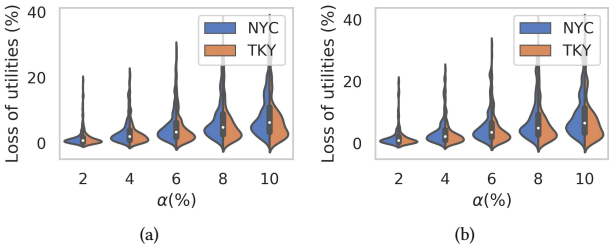


Figure 7: sim-BlockPriv: comparison between the distribution of loss of utility for different numbers of sensitive location types (α) for r =(a) 500 meter, and (b) 2000 meter.

of sim-BlockPriv. Figures 6(a-d) present the average loss of utility for different values of α. We found that, regardless of the value of privacy level r, there is a linear correlation between utility and α. For the same value of r, the higher the value of the α, the higher the loss of utility. However, the increase of loss of utility is slightly sharper in diff-BlockPriv than in sim-BlockPriv. This sharpness is due to the effect of both the increase in the number of sensitive location types and the randomness in the privacy level. As we already know that sim-BlockPriv is better than diff-BlockPriv, we only present the distribution of loss of utility in sim-BlockPriv in Figure 7. For the same reason, we skipped the depiction of impact of different γ in diff-BlockPriv. Similar to the average values in Figure 6, the distributions of the loss of utility exhibit a linear correlation. More accurately, the average correlation is 0.92 and 0.95 in the NYC and TKY datasets, respectively. As we mentioned earlier, such a linear correlation can play important role to make **BlockPriv** usable for privacy-preserving applications.

6.2.3 User Level Correlation Analysis. Figure 8 depicts the correlation values for loss of utility versus privacy level (U-P) and loss of utility versus number of sensitive location types (U-S) for 100 users; Table 4 presents different statistics (min, average, and max) on these values. It is observed that in the NYC dataset, 75% of the nodes have 0.9 correlation for both U-P and U-S. In the case of the TKY dataset, these numbers are 82% and 84%, respectively. Note that, these statistics are generated by considering the 100 nodes with the greatest number of transactions in the datasets. We found that, when the number of transaction is fewer, the loss of utility is significantly less. For instance, in both datasets, the 100 nodes with fewest number of transactions achieved minimum 30% less loss of utility than the 100 nodes with highest number of transactions.

7 CONCLUSION AND FUTURE WORK

In this paper, we introduce a user-centric obfuscation technique called **BlockPriv**, to preserve location privacy in permissioned blockchain-based IoT systems. As part of this work, we consider that a user cannot falsify its location and an untrusted authority can correlate locations by considering spatiotemporal constraints to predict unrevealed sensitive locations of a user. We quantify the relationship between the notion of privacy and utility of the system in **BlockPriv**. We analyze two variations of **BlockPriv**, sim-BlockPriv and diff-BlockPriv, where the first has the same privacy level for all the sensitive locations, and the second has a

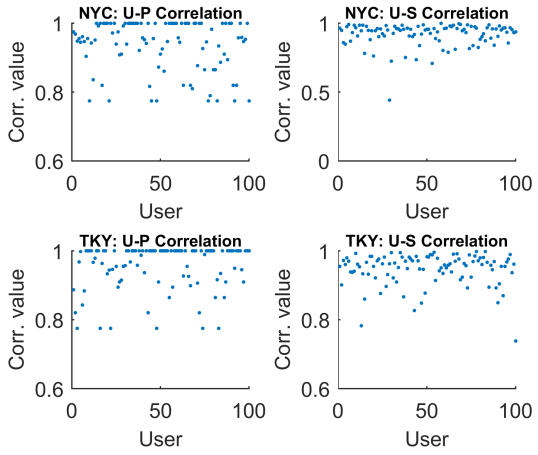


Figure 8: sim-BlockPriv: correlation values (Corr. value) of loss of utility versus privacy level (U-P) and loss of utility versus number of sensitive location types (U-S) for 100 users in NYC and TKY datasets.

different privacy level for different sensitive locations. We show that there is a linear correlation between loss of utility and privacy level in sim-BlockPriv. Such linearity can be exploited to define a usable privacy scale. In the extended version of this work, we intend to employ a more rigorous model to simulate the mobility of the nodes. Our future work also includes, improving the technique by considering different probabilistic attack models based on a combination of off-chain and on-chain information, adapting the approach for the case of continuous transactions in the network, and defining a soft privacy margin to further reduce the loss of utility.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation grant number CNS-1851890 for the REU site, U.S. Air Force Research Laboratory (AFRL) grant FA8750-17-S-7003 and the Dissertation Year Fellowship support provided by FIU's Graduate School. The content of this paper does not necessarily reflect the position or the policy of the US AFRL, NSF, or FIU and no official endorsement should be inferred.

REFERENCES

[1] Amazon. 2019. Amazon QLDB. https://aws.amazon.com/qldb/
[2] Michele Amoretti, Giacomo Brambilla, Francesco Medioli, and Francesco Zanichelli. 2018. Blockchain-Based Proof of Location. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 146–153.
[3] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13)*. ACM, New York, NY, USA, 901–914. https://doi.org/10.1145/2508859.2516735
[4] Microsoft Azure. [n.d.]. Blockchain. https://azure.microsoft.com/en-us/solutions/blockchain/
[5] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.
[6] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 15–29.
[7] Salim Bitam, Abdelhamid Mellouk, and Sherali Zeadally. 2015. VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wireless Communications* 22, 1 (2015), 96–102.
[8] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. 2014. Mixcoin: Anonymity for Bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*. Springer, 486–504.
[9] Giacomo Brambilla, Michele Amoretti, and Francesco Zanichelli. 2016. Using Blockchain for Peer-to-Peer Proof-of-Location. *arXiv preprint arXiv:1607.00174* (2016).
[10] Stefano Chessa, Antonio Corradi, Luca Foschini, and Michele Girolami. 2016. Empowering mobile crowdsensing through social and ad hoc networking. *IEEE Communications Magazine* 54, 7 (2016), 108–114.
[11] Ali Dorri, Marco Steger, Salil S Kanhere, and Raja Jurdak. 2017. Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine* 55, 12 (2017), 119–125.
[12] Sean Foley, Jonathan R Karlsen, and Tālis J Putniņš. 2019. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies* 32, 5 (2019), 1798–1853.
[13] Pedro Franco. 2014. *Understanding Bitcoin: Cryptography, engineering and economics*. John Wiley & Sons.
[14] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. 2016. Protecting against velocity-based, proximity-based, and external event attacks in location-centric social networks. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 2, 2 (2016), 8.
[15] Bing Jia, Tao Zhou, Wuyungerile Li, Zhenchang Liu, and Jiantao Zhang. 2018. A Blockchain-Based Location Privacy Protection Incentive Mechanism in Crowd Sensing Networks. *Sensors* 18, 11 (2018), 3894.
[16] Philip Koshy, Diana Koshy, and Patrick McDaniel. 2014. An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*. Springer, 469–485.
[17] Fenghua Li, Yahong Chen, Ben Niu, Yuanyuan He, Kui Geng, and Jin Cao. 2018. Achieving Personalized k-Anonymity against Long-Term Observation in Location-Based Services. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.
[18] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang. 2018. CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 19, 7 (July 2018), 2204–2220. https://doi.org/10.1109/TITS.2017.2777990
[19] Regio A. Michelin, Ali Dorri, Marco Steger, Roben C. Lunardi, Salil S. Kanhere, Raja Jurdak, and Avelino F. Zorzo. 2018. SpeedyChain: A Framework for Decoupling Data from Blockchain for Smart Cities. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. ACM, New York, NY, USA, 145–154. https://doi.org/10.1145/3286978.3287019
[20] Z. Montazeri, A. Houmansadr, and H. Pishro-Nik. 2017. Achieving Perfect Location Privacy in Wireless Devices Using Anonymization. *IEEE Transactions on Information Forensics and Security* 12, 11 (Nov 2017), 2683–2698. https://doi.org/10.1109/TIFS.2017.2713341
[21] J.P Morgan. 2019. Quorum. https://www.goquorum.com/
[22] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *bitcoin.org* (2008).
[23] Clemence Roulin, Ali Dorri, Raja Jurdak, and Salil Kanhere. 2018. On the Activity Privacy of Blockchain for IoT. *arXiv preprint arXiv:1812.08970* (2018).
[24] Rohit Sharma and Suchetana Chakraborty. 2018. BlockAPP: Using Blockchain for Authentication and Privacy Preservation in IoV. In *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 1–6.
[25] Madhusudan Singh and Shiho Kim. 2017. Blockchain Based Intelligent Vehicle Data sharing Framework. *arXiv preprint arXiv:1708.09721* (2017).
[26] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2015. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.
[27] Mengmeng Yang, Tianqing Zhu, Kaitai Liang, Wanlei Zhou, and Robert H Deng. 2019. A blockchain-based location privacy-preserving crowdsensing system. *Future Generation Computer Systems* 94 (2019), 408–418.
[28] Zhe Yang, Kan Yang, Lei Lei, Kan Zheng, and Victor CM Leung. 2018. Blockchain-based Decentralized Trust Management in Vehicular Networks. *IEEE Internet of Things Journal* (2018).
[29] Bidi Ying, Dimitrios Makrakis, and Zhengzhou Hou. 2015. Motivation for protecting selfish vehicles' location privacy in vehicular networks. *IEEE Transactions on Vehicular Technology* 64, 12 (2015), 5631–5641.
[30] Guy Zyskind, Oz Nathan, et al. 2015. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 180–184.