Journal of Parallel and Distributed Computing 136 (2020) 40–51

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Blockchain for secure location verification

Mohammad Reza Nosouhi [a,b,*], Shui Yu [a], Wanlei Zhou [a], Marthie Grobler [b], Habiba Keshtiar [c]

[a] *School of Computer Science, University of Technology Sydney, Australia*
[b] *CSIRO's DATA61, Australia*
[c] *Deakin University, Melbourne, Australia*

ARTICLE INFO

Article history:
Received 29 April 2019
Received in revised form 26 September 2019
Accepted 24 October 2019
Available online 31 October 2019

Keywords:
Distance bounding
Location-based services
Location privacy
Location proof

ABSTRACT

In location-sensitive applications, dishonest users may submit fake location claims to illegally access a service or obtain benefit. To address this issue, a number of location proof mechanisms have been proposed in literature. However, they confront various security and privacy challenges, including Prover–Prover collusions (Terrorist Frauds), Prover–Witness collusions, and location privacy threats. In this paper, we utilize the unique features of the blockchain technology to design a decentralized scheme for location proof generation and verification. In the proposed scheme, a user who needs a location proof (called a *prover*) broadcasts a request to the neighbor devices through a short-range communication interface, e.g. Bluetooth. Those neighbor devices that decide to respond (called *witnesses*) start to authenticate the requesting user. We integrate an incentive mechanism into the proposed scheme to reward such witnesses. Upon successful authentication, a transaction is generated as a location proof and is broadcast onto a peer-to-peer network where it can be picked up by verifiers for final verification. Our security analysis shows that the proposed scheme achieves a reliable performance against Prover–Prover and Prover–Witness collusions. Moreover, our prototype implementation on the Android platform shows that the proposed scheme outperforms other currently deployed location proof schemes.

1. Introduction

In recent years, advancements in smartphone technology and positioning systems have resulted in the emergence of location-based applications and services such as activity-tracking applications, location-based services (LBS), database-driven cognitive radio networks (CRNs), and location-based access control systems. In these services, mobile users' real-time location data is utilized by a location-based service provider (LBSP) to provide users with requested information or access to a resource or service. These applications are fast growing and very popular due to the range of useful services they offer [8,23,29,39].

However, these services and applications are vulnerable to location spoofing attacks since dishonest users are incentivized to lie about their location and submit fake position data [21,28,35,44]. To clarify and highlight the fake location submission issue consider LBSPs like Yelp and Foursquare that may offer some rewards (such as gift vouchers) to users who frequently check-in at specific locations. This creates an incentive for dishonest users to submit fake check-ins by manipulating their GPS data. In a research study, Zhang et al. [43] found that a significant percentage of Foursquare check-ins are fake and submitted by dishonest users to obtain benefit. Furthermore, in database-driven CRNs, malicious users can submit fake locations to the database to access channels which are not available in their location [21,22,28]. In addition, considering location-based access control applications, attackers can gain unauthorized access to a system or resource by submitting fake location claims [7,9,16]. Finally, in activity-tracking applications, insurance companies may offer health insurance plans in which customers are offered discounts if they have a minimum level of physical activity. This creates an incentive for dishonest users to cheat on their location data [1–4,35] (refer to [36] for other examples of location-sensitive applications). Thus far, with these examples, it is clear that preventing fake location submissions in these applications is still an open challenge.

To solve the problem, different location proof mechanisms have been proposed in literature [18,20,26,32,36,37,40,45]. In these mechanisms, a user's physical presence at a specific location is checked and a location proof (LP) is issued. LPs are small sets of meta-data that certify their receiver to a geographical location [36]. Using the received LPs, users can submit a location claim with an LBSP. However, the current LP schemes confront a number of different security and privacy challenges,

[*] Corresponding author at: School of Computer Science, University of Technology Sydney, Australia.
E-mail address: mohammad.r.nosouhi@student.uts.edu.au (M.R. Nosouhi).

https://doi.org/10.1016/j.jpdc.2019.10.007
0743-7315/© 2019 Elsevier Inc. All rights reserved.

including Prover–Prover collusions (Terrorist Frauds), distance frauds, Prover–Witness collusions, and location privacy threats [10–13,40] (refer to Section 4 for more detail). To the best of our knowledge, all the LP schemes proposed so far are vulnerable to at least one of these threats and challenges (refer to Section 2 for more detail).

In this paper, we present a blockchain-based LP generation and verification scheme. A preliminary version of this paper appears as [33]. We utilize the unique features of the blockchain mechanism [15,24,27,34,38] to address the aforementioned challenges. In the proposed scheme, mobile users act as witnesses for a user (prover) that requests an LP in their physical proximity. Witnesses start to authenticate the prover and if the prover is successfully authenticated, a transaction is created based on the proposed blockchain framework. The transaction (that holds the prover's LP data) is then broadcast onto a peer-to-peer network over the internet where it can be picked up by verifiers for further verification. Finally, the verified transaction is stored in a time-stamped public ledger accessible for LBSPs. To preserve users' location privacy, they cryptographically commit to their spatiotemporal data before it is inserted into a transaction. Later, they open the commitment when they submit a location claim with an LBSP.

To prevent distance frauds, all the communications between prover and witness devices are performed through a short-range communication interface like Bluetooth. Moreover, we integrate an incentive mechanism into the proposed scheme whereby witnesses and verifiers are rewarded by a small amount of cryptocurrency. This incetivizes them to collaborate with the system rather than ignoring provers' requests to save power on their device.

In the proposed blockchain framework, transactions are created using a secure and privacy-aware method. They hold provers' commitment to their spatiotemporal data and the information related to witnesses and verifiers' rewards. However, they do not contain any data that identifies a prover. A group of transactions forms a block which is identified by a unique header generated by a cryptographic hash function. Each block contains hash of the previous block, therefore, all blocks are inherently linked. This makes the ledger (that holds users' LPs) immutable and irreversible. Moreover, employing the decentralized blockchain architecture enables us to benefit from the power of consensus, while the conventional LP schemes are performed by a central third-party entity. Our security and privacy analysis shows that the proposed scheme protects location-based applications against fake submissions. In addition, we implemented the proposed scheme on the Android platform and conducted extensive experiments. The results show that the proposed scheme outperforms the current LP schemes. The following are our contributions:

- We propose a blockchain-based, secure, and privacy-aware scheme for LP generation and verification in which mobile users generate LPs for each other. The designed architecture is collusion resistant and preserves users' location privacy.
- We develop an incentive mechanism to encourage mobile users to collaborate with the system and integrate it into the proposed scheme.
- We perform a prototype implementation on the Android platform. The experimental results show that our proposed scheme outperforms the existing decentralized LP schemes.

The rest of our paper is organized as follows. After reviewing the related work in Section 2, we present an overview of the proposed system in Section 3. The design challenges are discussed in Section 4. The proposed architecture and a comprehensive security analysis are presented in Sections 5 and 6. Finally, after presenting the implementation results in Section 7, we conclude the paper.

2. Related work

With the recent advances in the blockchain technology, many new distributed applications have been developed. Huckle et al. [25] present a number of blockchain-based distributed applications. The focus of their work is on utilizing the blockchain and Internet of Things technologies to create secure shared economy applications. Digital Rights Management, AutoPay, and a blockchain-based currency exchange application are some example applications presented in the paper. In [6], the authors present a literature review of the potential applications of blockchain technology in government services. Furthermore, Casino et al. [14] provide a systematic literature review of blockchain applications in different domains. There are also some survey papers that focus on applications of the blockchain technology in specific domains including decentralized digital currencies [38], healthcare systems [31], and IoT [17].

Generally, the LP mechanisms are categorized into two groups depending on the system architecture: *centralized* and *decentralized*. In the centralized schemes, a trusted fixed wireless infrastructure, usually a WiFi access point, is employed to check the proximity of mobile users and generate LPs for them. On the other hand, in the decentralized schemes, this task is done by ordinary mobile users who act as witnesses and issue LPs for each other. This makes their implementation easier and cheaper than the centralized mechanisms. In this section, we review the related literature on each category separately.

2.1. Centralized schemes

In the centralized architectures, central trusted nodes, usually WiFi access points, are employed to generate LPs for mobile users in their neighborhood. These mechanisms are dependent on a large number of wireless access points in order to work properly. One of the earliest research works in this area was presented by Saroiu et al. [36]. In the proposed mechanism, mobile users request an LP from the nearest WiFi access point or cell tower. They can then submit the LP obtained from the infrastructure to a service provider. In addition, the researchers presented a list of different location-based applications of LPs. However, location privacy issues have not been considered in the proposed centralized architecture.

The most recent centralized mechanism has been proposed by Javali et al. [26]. In their approach, the unique wireless channel characteristics, i.e., channel state information (CSI), are used to decide on users' proximity to a trusted WiFi access point. They employ an information theoretically secure fuzzy vault scheme for LP verification that is performed by a verifier and a server. However, the proposed mechanism does not preserve users' location privacy. In addition, all the available devices on market may not be capable of measuring CSI information. Furthermore, different entities are employed in the proposed architecture that makes it expensive for implementation. In reality, the costs associated with employing a large number of WiFi access points in different sites around the world are too much for an LBSP.

2.2. Decentralized schemes

In decentralized architectures, ordinary mobile users act as a witness and generate an LP for each other. Since the witnesses are inherently not trusted, a user must send his/her obtained LP to a trusted verifier for further verification. An advantage of this approach is that there is no need to employ trusted access points for LP generation. Thus, these architectures are generally cheaper to implement than the centralized mechanisms. Moreover, they

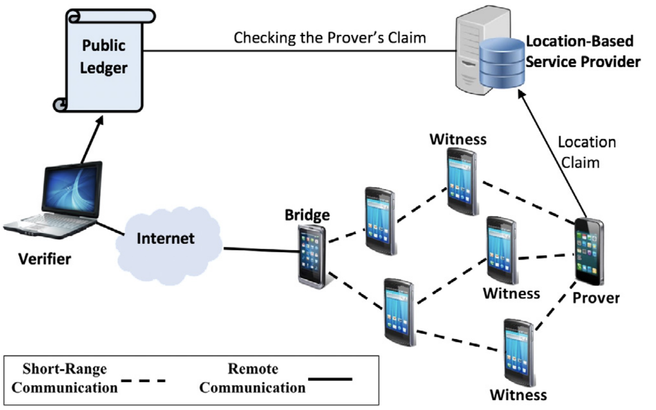


Fig. 1. The proposed decentralized architecture. The prover submits a location claim with a LBSP after his/her LP is stored on the public ledger.

are suitable for use by applications where users are not within proximity of wireless infrastructure.

One of the earliest decentralized architectures proposed in the literature is APPLAUS, introduced by Zhu et al. [45]. In APPLAUS, nearby mobile devices communicate through their short-range Bluetooth interface to issue LPs for each other. To preserve users' location privacy, a mobile device changes its pseudonym periodically. To do this, it first needs to register a set of M public/private key pairs with a trusted Certificate Authority (CA) where the M public keys are the pseudonyms of the user. However, this mechanism imposes high communication and processing overhead since it needs to periodically change pseudonyms and generate dummy LPs. In addition, they have not addressed Prover–Prover collusions in their proposed scheme.

Davis et al. [18] proposed another distributed architecture in which a privacy-preserving alibi (or LP) system is presented. In this mechanism, a user's ID is not revealed during the alibi generation phase until he/she decides to submit his/her alibi to a judge. The authors did not discuss the proposed mechanism's performance against security threats and users collusions.

In LINK, introduced by Talasila et al. [37], a prover device contacts its neighbor devices through a short-range wireless interface to obtain LPs. To verify the issued LPs, a trusted central Location Certification Authority (LCA) is employed. The LCA verifies the users' location claims using three parameters: (1) the spatiotemporal correlation between users, (2) individual users' trust scores, and (3) the history of the trust scores. The main disadvantage of this work is that privacy protection has not been considered in the protocol design.

PROPS, introduced by Gambs et al. [20], is another decentralized architecture that allows users to act as a witness and generate LPs for other users in a private way. However, it does not address Prover–Witness collusions.

STAMP, proposed by Wang et al. [40], presents an entropy-based trust model to address Prover–Witness collusions, but this approach cannot prevent or detect these collusions with 100% certainty. In addition, they have employed the Bussard–Bagga Distance Bounding (DB) protocol [13] to address Terrorist Frauds (Prover–Prover collusions). This protocol is shown to be unsecure and does not provide enough protection against Terrorist Frauds [10–12]. Furthermore, due to the use of a DB protocol, the computation time required by STAMP to generate LPs increases when users adopt a large private key [26].

Table 1 presents a comparison of these LP schemes. To summarize, the previous research works in both the centralized and decentralized architectures require a central trusted authority for LP verification. However, as you will see in this paper, the use of a decentralized blockchain framework enables ordinary users to verify LPs.

3. System overview

In this section, an overview of the proposed architecture is presented. Firstly, we introduce the different entities involved in the system architecture. Then, we present the threat and trust model considered in the system design.

3.1. Entities

Fig. 1 shows the proposed system architecture. There are four types of entities in the proposed architecture:

Prover: A mobile user who needs to have his/her spatiotemporal data stored in the public ledger.

Witness: A mobile user who generates an LP for another user that is in his/her vicinity in order to obtain a reward.

Verifier: The entity who verifies a newly generated transaction, creates a block and adds it to the chain. The verifier does not necessarily need to be located in the prover's proximity and can be located anywhere in the network.

Bridge: The entity who is responsible for broadcasting issued LPs (received from a short-range communication interface such as Bluetooth or WiFi) onto the peer-to-peer network on the internet. Thus, it acts as an interface between the local communications and the internet. It can be either a mobile or a fixed device (such as a WiFi access point).

An LBSP that acts as a third party provides the prover with a facility, reward or access to a service when it ensures that the prover's claimed spatiotemporal data stored in the public ledger corresponds to a specific location and time.

3.2. Threat and trust model

To gain more benefit, dishonest provers might attempt to obtain an LP for locations at which they are not physically located. For this reason, they might either send wrong information to the witnesses or try to manipulate the content of an LP. Moreover, to attain this target, a dishonest prover might collude with other users. In this subsection, we present some assumptions regarding the threat and trust model which we consider in this paper:

- Users never share their private key with each other [26,40].
- It is assumed that the witnesses in the network are untrusted. Hence, they might collude with a remote dishonest prover to generate a fake LP for him/her.
- There is a bridge selection mechanism in which a mobile device (or a Wifi access point if possible) in a specific zone is elected by the other users to act as a bridge for a specific period of time.
- Users execute the proposed scheme by running end-user software that generates a unique public/private key pair for a user as well as a unique cryptocurrency address. It also allows every device to have only one ID (public key). Moreover, bridges use this software to send transactions onto the peer-to-peer network where they can be picked up by verifiers and added into a block.

4. Preliminaries

In this section, we first present an overview of the blockchain technology and review the three different types of blockchains. Following this, we present some of the design challenges that we need to address in this paper.

Table 1
Comparison of LP schemes.

LP scheme	Features	Advantages	Disadvantages
Javali et al. [26]	Centralized architecture, no DB mechanism is used, utilizes channel state information to decide on users proximity	Resistant to P–P collusions	Privacy issue, expensive for implementation
Saroiu et al. [36]	Centralized architecture, access point broadcasts sequence numbers periodically, provers sign the last transmitted Sequence number to request an LP	Resistant to P–P Collusions	Privacy issue
VeriPlace [30]	Distributed architecture, a user needs to get an intermediate LP from a trusted access point to obtain a final LP	Privacy–aware	Needs three types of trusted entities run by separate parties
STAMP [40]	Distributed architecture, an entropy-based trust model is used to address P–W collusions	Supports location Granularity	Vulnerable to P–P collusions (the broken Bussard–Bagga protocol is used)
APPLAUS [45]	Distributed architecture, provers change their pseudonyms periodically	Privacy–aware	High communication and computation overheads
Alibi [18]	Distributed architecture, provers' ID is revealed only when they choose to submit their alibi to a judge	Privacy–aware, lightweight	Vulnerable to collusion attacks
Link [37]	Distributed architecture, a group of local users collaboratively verify a prover's location	Resilient to situations when there is not enough neighbor devices	Privacy issue
SPARSE [32]	Distributed architecture, no DB mechanism is used for secure proximity checking	Privacy–aware, resistant to P–P collusions	Prevents P–W collusions only in crowded scenarios
PROPS [20]	Distributed architecture, group signatures and ZKP are used to make provers anonymous	Privacy–aware, efficient	Vulnerable to P–W collusions

4.1. Blockchain overview

A blockchain system is a tamper-proof and tamper-apparent ledger that is digitally implemented using a distributed approach without requiring a central storage system. It can also be implemented in such a way that no central authority is required to operate and maintain the whole system. The distributed ledger consists of users transactions that are cryptographically signed by them. A group of transactions create a *block*. Thus, the distributed ledger is made of blocks of transactions. Users generate their transactions and broadcast them in the network where they can be read by verifiers for verification. A verifier can be either an ordinary user (in public blockchains) or an authorized user or entity (in private and consortium blockchains). Once a transaction is verified, it is added to a new block. After a new block is issued (added to the ledger), it is computationally infeasible to tamper its transactions [15,38,41]. The reason is that each block contains the hash of its previous block. This links every block to its previous block which results in having a chain of blocks.

Generally, blockchains can be divided into the following three categories based on their permission model, which determines who can operate and maintain the system (e.g., generate a new block) [41].

- Public blockchains: In a public blockchain system, any user can verify transactions and generate a new block. This type of blockchain systems is also called permissionless blockchain in which the system allows anyone to join the network with both read and write permissions. Bitcoin is an example of a public blockchain.
- Private blockchains: In private blockchains, only specific users or entities can verify transactions and generate a new block. In other words, the system is controlled by an organization that manages access permissions. Thus, not all the users have access to the detail of the transactions stored in the ledger.
- Consortium blockchains: These systems are actually private blockchains that are employed by a group of organizations. A consortium blockchain is considered as a semi-decentralized blockchain that usually adopts a hybrid access method. Quorum and Corda are two examples of consortium blockchains.

It is proposed that the LP scheme proposed in this paper is implemented as a public blockchain system. This has several advantages including complete decentralization (independent of a trusted third party), full transparency of the public ledger, more security (due to use of incentivized validation that results in more miners contributing to validations), and self-sustainability. However, depending on the application scenario, the proposed scheme can be implemented as a private or consortium blockchain as well.

4.2. Design challenges

Blockchain technology has created a great opportunity to design decentralized systems for different applications. However, some novel features of a blockchain-based architecture introduce a number of design challenges for our work. For example, recording users' location data in a public ledger contradicts their location privacy. Moreover, regardless of the blockchain architecture, there are different security and privacy challenges for LP generation and verification that must be addressed. In this section, these design challenges are presented in three different categories,

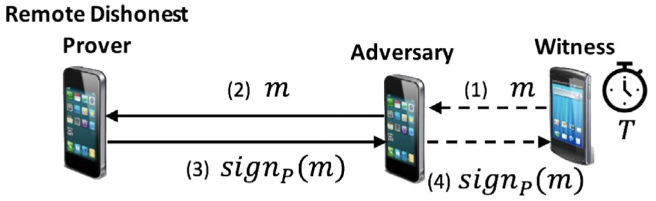


Fig. 2. An example of P–P collusions. To obtain a fake LP, the dishonest prover colludes with an adversary who is in a desired location. The adopted time-limited approach prevents the adversary from relaying the random sequence m to the remote prover and obtain his signature.

i.e., *Security*, *Privacy*, and *Application-related* challenges. We also present a countermeasure for each design challenge which is used in the proposed scheme to address the challenge.

4.2.1. Security challenges

Prover–Prover (P–P) collusions: In P–P collusions (also known as Terrorist Frauds), a distant malicious prover colludes with an adversary who is located in the vicinity of an honest witness. During the attack, the adversary pretends to be the distant prover and submits an LP request with the witness on behalf of the malicious prover.

To detect P–P collusions, we adopt a time-limited approach in which a witness generates a random number m and sends it to the prover through a short-range communication interface. Then, only a short period of time T is given to the prover to sign m and send it back to the witness. If the witness receives the response after the period T, it rejects the prover's request to generate an LP. Thus, in the case of a P–P collusion, an adversary does not have enough time to relay m to the remote dishonest prover and obtain his signature (see Fig. 2). Note that this approach assumes that users never share their private key with each other. Therefore, the adversary cannot sign m on behalf of the remote prover. Refer to Section 6 for more details on how a P–P collusion is detected using this technique.

Prover–Witness (P–W) collusions: In this collusion scenario, a dishonest witness colludes with a distant malicious prover and issues a fake LP for him. P–W collusions are the most difficult challenges to address in this area of research. To the best of our knowledge, no reliable and effective solution has been offered in the literature so far to address these attacks. In this paper, we adopt a novel mechanism to address P–W collusions in which an attacker (a dishonest witness who wants to generate a fake LP for a remote dishonest prover) is forced to change his attack to a P–P collusion attack. Therefore, the P–W collusion is detected since the proposed scheme detects P–P collusions through the presented time-limited approach (more detail is provided in Section 6).

4.2.2. Privacy challenges

To design an LP system using the blockchain architecture, one possible approach is that we record users' plaintext spatiotemporal data in a public ledger. However, this approach contradicts users' location privacy because their spatiotemporal data is shown publicly. In addition, it has been shown that even if users hide their real identity (which is common in blockchain systems) it is still possible to identify a user by analyzing the history of his/her spatiotemporal data [29,40,42]. Thus, adopting a pseudonym by users cannot guarantee their location privacy.

To address this challenge, we adopt a novel approach in which users commit to their spatiotemporal data before they create a transaction. These commitments are then added to the transaction and will be stored in the public ledger after verification:

$$C_{(P,ST)} = Commit(ST, r), \tag{1}$$

Table 2
List of notations.

Notation	Description
$\|$	Concatenation symbol
$Sign_u(m)$	Signature of user u on message m
$H(m)$	One-way hash function
ST	The prover's spatiotemporal data
$C_{(P,ST)}$	The prover's commitment to ST
LP	Location proof
Tx	Transaction

where $ST = (Loc, Time)$ is the spatiotemporal data of the user (prover) and r is a random nonce generated by him/her for the commitment to ST.

When a user submits a location claim with a LBSP, he/she opens the commitment by sending r to the LBSP. Therefore, if the user loses this r, his/her location claim will not be confirmed by the LBSP. This is similar to a user wanting to spend his/her Bitcoin money (in Bitcoin, users cannot spend the money in their wallet if their private key is lost). Therefore, it is infeasible to obtain the history of a user's spatiotemporal data by analyzing his/her transactions stored in the public ledger.

4.2.3. Application-related challenges

In this subsection, we present two critical challenges that may result in negative impacts on the performance of LP schemes if not addressed.

Challenge 1: A big challenge for our decentralized scheme lies in how we can convince mobile users to act as a witness since they generally tend to reject requests to generate LPs if there is not enough incentive for them (for example, to save on battery consumption).

To address this problem, we integrate an incentive mechanism into our proposed LP scheme to reward users who collaborate with the system with a specific amount of cryptocurrency. The LBSPs can make this currency valuable by exchanging them for rewards, badges and benefits that they are currently providing to their users (see [43] and [36] for more details and examples). Moreover, other businesses such as insurance companies and government agencies that might utilize LPs of their customers can contribute to make the currency more valuable. This creates the necessary incentive for mobile users and verifiers to collaborate with the system.

Challenge 2: Speed is another challenge for an LP system that needs to be addressed. As far as we know, the majority of the LP schemes which have been proposed in literature so far utilize a DB protocol to check the proximity of a prover to a witness. This not only requires some hardware changes on mobile devices, but also makes the LP generation process slow when users adopt a long private key [26] since a prover device must respond to m challenge messages that a witness sends to it, where m is the size of the prover's private key. This process is called *fast–bit–exchange* in literature.

In this paper, we do not adopt a DB protocol to check the proximity of a prover to a witness. Instead, the time-limited mechanism (discussed in the security challenges) enables the witness to check whether the prover is really located in its vicinity or not. This makes the LP generation process much faster than the current solutions (see Section 7 for a comparison).

5. The proposed architecture

The proposed scheme is executed in three stages, discussed next. Fig. 3 shows the message exchange diagram of the proposed scheme (refer to Table 2 for a list of the notations that we use in

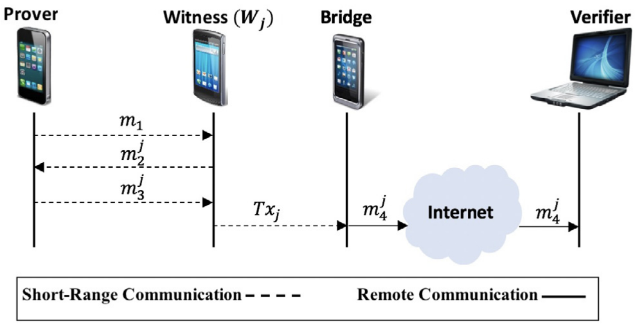


Fig. 3. Message exchange diagram of the proposed scheme.

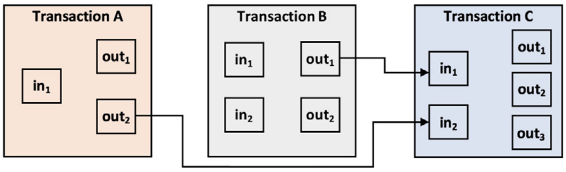


Fig. 4. In the proposed scheme, each transaction can have different inputs and outputs. Every input comes from one of the outputs of an unspent transaction (indicates the source of the amount of cryptocurrency that the prover wants to spend in this transaction). For example, in transaction C, the prover wants to spend the cryptocurrency that he has already obtained in Transaction A and B (output indices 2 and 1, respectively). The first and second outputs contain the identity of the witness and bridge, respectively, and the third output can be assigned to the prover (indicates that the difference between $in_1 + in_2$ and $out_1 + out_2$ remains in the prover's wallet). Moreover, the difference between the sum of all the inputs and the sum of all the outputs is transferred to the verifier's wallet as the transaction fee.

this paper). We explain the scheme step by step based on the computations and operations that each entity performs.

(1) LP Request Submission

Prover. In the first stage, the prover generates the following message m_1 and broadcasts it to the surrounding witness devices through a predefined short-range communications interface:

$$m_1 = \bar{m}_1 \parallel r \ ,$$

where $\bar{m}_1 = ID_P \| H(Prev\_Tx) \| Index \| Rew \| C_{(P,ST)}$.

In $\bar{m}_1$, ID_P is the prover's ID (public key), $H(Prev\_Tx)$ is the hash of the prover's previous unspent transactions containing the outputs that the prover wants to spend now (see Figs. 4 and 5), *Index* specifies the index of that output (it indicates how the prover has received this cryptocurrency that he/she wants to spend), and $C_{(P,ST)}$ is the prover's commitment to his/her spatiotemporal data (see Eq. (1)).

As you see, the prover sends the random nonce r to the witness because the witness must be able to open the commitment $C_{(P,ST)}$ to ensure that it is the same with the current location. This prevents dishonest provers from committing to a different location data and obtaining an LP for it.

Rew in the above message is the amount of reward that the prover is willing to pay to a witness and a bridge. This reward cannot be less than a predefined minimum amount Rew_{min}. The minimum reward amount is set to provide additional protection against P–W collusions where a malicious witness is forced to change his attack to a P–P collusion (see Section 6 for more details). Without the predefined minimum amount, the malicious witness can broadcast a m_1 message on behalf of a remote dishonest prover and add a very low reward amount to the *Rew* field to decrease the incentive of other witnesses to issue an LP for this request. Thus, in this case, the attack can proceed with a higher chance of success.

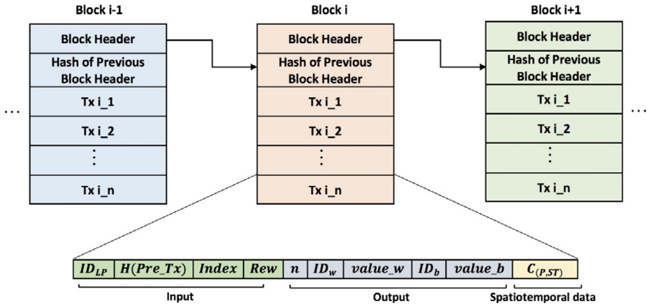


Fig. 5. Block creation and transaction structure in the proposed scheme. Each transaction consists of three sections: input, output, and the spatiotemporal data. The input section can include different unspent transactions. For simplicity, users' signatures have not been shown in the picture.

(2) Tx Generation and Submission

Witness. Upon receiving m_1, a witness (let us say $w_j, j = 1, 2, \ldots, J$ where J is the number of witnesses that reply to the prover's request) opens the commitment $C_{(P,ST)}$ to check whether the spatiotemporal data inserted by the prover matches the current location and time or not. If they are the same, w_j sends the following message m_2^j to the prover and starts a timer. This is done after the prover device acknowledges that it is ready to receive the challenge message m_2^j. This prevents any disorder in execution of the mechanism since J different witnesses want to send their challenges message to the prover:

$$m_2^j = LP_j \parallel Sign_{w_j}(LP_j), \quad j = 1, 2, 3, \ldots, J$$

where $LP_j = \bar{m}_1 \| n_j \| ID_{w_j} \parallel ID_b$ in which n_j is a random number generated by w_j ID_{w_j} and ID_b are the identity of the witness and selected bridge, respectively. Note that each device knows the ID of the currently serving bridge that has been elected by users.

Prover. When the prover device receives each m_2^j, it immediately signs it using its private key and sends the following message m_3^j to w_j:

$$m_3^j = m_2^j \| Sign_P(m_2^j) \| \quad j = 1, 2, 3, \ldots, J$$

By signing m_2^j (which is an LP), the prover consents that he/she rewards *Rew* to the witness, selected bridge, and verifier. For simplicity, we assume their share is equal, however this assumption can be removed by adding two fields *value_w* and *value_b* next to the ID_{w_j} and ID_b respectively, to specify their share (see Figs. 4 and 5). The difference between *Rew* and $value\_w + value\_b$ is considered as the verifier's reward (same as the payment mechanism in Bitcoin).

Witness. Upon receiving m_3^j, witness w_j stops the timer and checks to see whether it was received in the predefined period of time T. T is a system parameter and must be carefully designed such that it only provides the prover with an opportunity to sign m_2^j, and send it back to w_j (in Section 7, we propose some practical values for T based on our experimental results). Thus, with a carefully selected T, in case of a P–P collusion, an adversary does not have such an opportunity to relay m_2^j to a remote dishonest prover and receive his signature.

If m_3 is received in time, w_j creates the following transaction Tx_j and broadcasts it through its short-range communication interface to be delivered to the selected bridge. Other mobile users that are located between the witness and bridge discard the transaction if they have not already received the prover's request message m_1. Otherwise, they broadcast it such that it finally reaches the bridge. As you will see in the next section, this technique enables the proposed scheme to prevent P–W collusions.

$$Tx_j = m_3^j \parallel Sign_{w_j}(m_3^j), \quad j = 1, 2, \ldots, J$$

Note that w_j does not have to check the prover's signature on m_2^j since a verifier (that has more power and computational resources) can do it later in the *Tx Verification* phase. You will further see in the *Tx Verification* phase that a verifier randomly selects the *Tx* issued by one of the *J* witnesses to add to the chain.

If m_3^j is not received in time, or the signature on m_2^j is not correct, w_j broadcasts the following transaction Tx_j onto the peer-to-peer network on the internet:

$$Tx_j = m_3^j \| Nack \| Sign_{w_j}(m_3^j \parallel Nack)$$

Thus, in case of a P–P collusion, the *Nack* transaction informs the verifiers on the internet that a collusion is taking place.

The selected bridge. Upon receiving $Tx_j, j = 1, 2, \ldots, J$, the selected bridge device checks it to make sure its ID has been correctly inserted. It also discards the additional copies of a transaction that might be received. Then it signs Tx_j using its private key and broadcasts the following result onto the peer-to-peer network using its internet interface:

$$m_4^j = Tx_j \parallel Sign_b(Tx_j)$$

(3) Tx Verification

A *Tx* can be verified by any verifier in the network. A verifier plays a similar role to a miner in Bitcoin. Upon receiving $m_4^j, j = 1, 2, \ldots, J$, a verifier starts to perform the following checks:

- The prover signature on m_2^j matches the prover's public key, i.e. ID_P.
- The witnesses' signatures on LP_j and Tx_j messages match their ID.
- The bridge signature on m_4^j is correct.
- *Rew* is equal to or less than the output value of the prover's unspent transactions (indicated by $H(Prev\_Tx)$ and *Index*).
- At least J_1 non–*Nack* transactions received for this LP that pass the above checks.

Moreover, if the output value in the prover's previous transaction (indicated by *index* in Fig. 5) is greater than *Rew*, the difference is considered as the last output of the current *Tx* and goes to the prover's wallet. Thus, in this case, the prover can use this *Tx* later as an unspent transaction.

If all the above checks are passed, the verifier randomly selects one of the *J* transactions and adds it to the current block that he/she is creating (see Fig. 5). Blocks are identified by a unique header created by a cryptographic hash function. Each block is linked to the previous blocks since the hash of the previous block is stored in every block. This makes the ledger immutable and irreversible.

Regarding the consensus algorithm, we adopt the Proof of Stake (PoS) approach instead of the Proof of Work (PoW) method. In PoS, a random selection process is used by the network to determine which node is the generator of the next block. This selection process is performed by considering a combination of different parameters, e.g., the wealth of each node, the age of stakes, and different randomization factors [38]. Those nodes who want to participate in the block generation process must lock a specific amount of their coins into the network. This is considered as their bond or stake that determines their chances to be selected as the next block validator (larger stakes result in higher chances). Some randomization techniques are also integrated into the selection mechanism to randomize it and prevent the wealthiest nodes to take control of the network. Dishonest validator nodes lose their stake if they do not perform the protocol honestly (i.e., not to verify a fraudulent transaction).

PoS has several advantages over PoW and adopting PoS as the consensus algorithm makes the block generation faster and more energy-efficient. For example, it is a much greener consensus mechanism since block creators need to consume a lot of energy in the PoW approach. Unlike the PoW approach, in PoS the block generators do not need to compete to solve difficult energy-consuming puzzles. Therefore, the time required for block generation only depends on the computational power of the block generator, i.e., the block generator only needs to verify the block's transactions and computes header of the block. In PoW, however, miners have to spend additional time to solve difficult puzzles as well. In addition, PoS provides more security regarding the 51% attack [38].

Similar to the Bitcoin setup, we propose two types of nodes in the network, i.e., *lightweight* and *full nodes*. Full nodes can store a copy of the entire ledger (all the blocks and transactions). Thus, there can be many backups of the public ledger in the network. Location–based service providers have the required incentive to play the role of a full node in the network since they benefit from the system a lot. Moreover, they have more storage resources to store a copy of the entire ledger. On the other hand, lightweight nodes (i.e., ordinary users) does not have to store the whole blockchain. These nodes can access and explore the ledger using the access services offered by the full nodes. These services are actually similar to the Simplified Payment Verification (SPV) service offered in Bitcoin [5]. Thus, lightweight nodes can create or verify a transaction without having to download the entire ledger.

6. Security and privacy analysis

In this section the security and privacy of the proposed scheme are discussed.

Location Privacy: In the proposed scheme, a prover commits to his/her spatiotemporal data before requesting an LP. Thus, instead of the plaintext spatiotemporal data, this commitment is stored in the public ledger. This makes the prover's location data publicly inaccessible. However, as discussed in Section 4.2, when the prover wants to submit a location claim with a LBSP, he/she opens the commitment by sending the random nonce r to the service provider. In the following, we provide a detailed analysis on the proposed technique to show how it preserves users' location privacy.

We make use of unforgeable cryptographic commitments to keep submitted locations private in the public ledger. To perform this technique, users commit to their spatiotemporal data before they create a transaction:

$$C_{(P,ST)} = Commit(ST, r) = g^r v^{ST},$$

where $ST = (Loc; Time)$ is the spatiotemporal data of the prover and $r \in \{0, 1, \ldots, q-1\}$ is a random nonce generated by the prover for commitment to ST. g and v are selected from the subgroup of G of order q in Z_p^*, where q and p are pre-determined values such that q divides $p-1$.

As discussed in Section 5, $C(P, ST)$ is added to a transaction and will be stored in the public ledger after verification. When the prover wants to submit a location claim with a LBSP, he/she opens the commitment by sending r and ST to the service provider who can confirm $C(P, ST) = g^r v^{ST}$. Using the above commitments, the proposed technique achieves two essential security properties:

1. **Binding Property:** A prover cannot change the submitted location anymore. Thus, when the commitment is opened at a later stage by an LBSP, the revealed location data is

really what the prover has already committed to. In other words, the prover cannot find another nonce $\bar{r}$ such that it results in the same commitment value $C(P, ST)$ considering a different spatiotemporal data $\bar{ST} \neq ST$.

Proof: If a prover P commits to ST and can later open the commitment as $\bar{ST} \neq ST$, then we have:

$$g^r v^{ST} = g^{\bar{r}} v^{\bar{ST}},$$

or equivalently:

$$\log_g(v) = \frac{r - \bar{r}}{\bar{ST} - ST}$$

This means that the prover could calculate the discrete logarithm $\log_g(v)$ that contradicts the fact that discrete logarithms are computationally infeasible to calculate [19].

2. **Hiding Property:** When a user's commitment is added to the public ledger, it is infeasible to open it unless the prover shares the random nonce r.

   Proof: Similar to the encryption with one-time pad, v^{ST} and consequently ST, are perfectly hidden in $g^r v^{ST}$ because g^r is a random element of G [19]. Thus, it is computationally infeasible for an adversary to obtain ST without the knowledge of r.

Therefore, the provers' private spatiotemporal data are not revealed to the public even though they are stored in a public ledger. However, the provers can open their commitments by sharing the random nonce r with the related service provider.

Resistance to Distance Frauds: This attack is performed by a single attacker, i.e. a malicious prover who is far from a desired location L [11,12]. During the attack, the malicious prover attempts to convince the honest witnesses (located at L) that his physical distance to them is less than what it really is. In the proposed architecture, mobile users perform the proposed scheme by running an application that communicates with other devices through a short-range communication interface. In other words, witness devices only listen for any incoming LP requests using their short-range communication interface. This makes it impossible for a distant malicious prover to perform a distance fraud.

Resistance to Terrorist Frauds (P–P collusions): The detection of P–P collusions is commonly performed by witnesses and a verifier. When a witness device receives a prover's LP request, it performs the presented time-limited mechanism in which the prover is given a short period of time T to sign message m_2^j (generated by the witness $w_j, j = 1, 2, \ldots, J$) and send it back to the witness. In case of a P–P collusion, an adversary who is conducting the attack has only two options to choose from, since he does not have the prover's private key:

(a) The adversary relays m_2^j to the remote dishonest prover to obtain his signature. This process takes a period of time $T' = 2t_c + t_p$ where t_c and t_p are the communication and processing times, respectively (see Fig. 2). In this case, the witness receives the response after $T'' = T + 2t_c$ approximately. Hence, the attack is detected by the witness because T'' is definitely greater than T. In Section 7, we propose some practical values for T based on our implementation results.

(b) The adversary signs m_2^j himself (using his own private key). In this case, the verifier who checks the signatures will detect this in the Tx verification phase. Note that checking the signature could also be done by the witnesses. However, it is more efficient to make a verifier responsible for this duty because it has more power and computational resources than a witness device (as signature checking requires heavy computations to be performed).

Moreover, utilizing the random number n_j makes m_2^j a random message. This prevents the adversary from guessing m_2^j and relaying it to the prover in advanced to have his signature ready to send back to the witness.

Resistance to P–W collusions: The proposed architecture has been designed in such a way that a malicious witness who is going to perform a P–W attack is forced to change his attack to a P–P collusion. To clarify this, let us consider a malicious witness W who colludes with a dishonest prover P to generate a fake LP for him. For this reason, W creates a fake transaction Tx_W and based on his situation performs one of the following:

(1) W broadcasts the fake transaction through its short-range communication interface for the elected bridge to receive and broadcast it onto the peer-to-peer network on the internet. The local mobile users (including the bridge) who receive Tx_W broadcast it only if they have already received an LP request (message m_1) with the same ID_P. Since the prover's ID_P is not published in the P–W collusion scenario, the fake transaction Tx_W is prevented from being published to the potential verifiers on the internet. Consequently, Tx_W is not added to the public ledger. This forces the witness to locally publish the ID_P by broadcasting an LP request message m_1. In this case, we can say that the attack is changed to a P–P collusion in which an adversary (W in this scenario) broadcasts an LP request on behalf of a remote prover. As we discussed in the previous analysis, the proposed scheme is resistant to P–P collusions. Therefore, P–W collusions are prevented by the scheme as well.

(2) W broadcasts the fake transaction Tx_W onto the peer-to-peer network himself (instead of broadcasting it to be received by a legitimate bridge). In this case, W does not have to be located at the desired location. To be successful, W needs to insert another ID along with its associated signature into the transaction as a bridge ID and signature, respectively. Since each mobile device may only have one ID, three mobile devices must be involved in the attack at this stage (P, W, and a bridge). However, in the verification phase, verifiers need to receive at least J_1 non–*Nack* transactions related to this LP (this is one of the requirements that a verifier checks before adding a transaction to a block). Hence, in this case, a malicious prover needs to collude with $J_1 + 1$ mobile devices to conduct a successful attack (totally $J_1 + 2$ mobile devices must be involved in the attack). This can be more expensive for the malicious prover than the reward or benefit that the LBSP provides specifically when J_1 is a large number. Note that users usually need to have an LP for crowded public places. Thus, there is no concern about the number of available witnesses. Specifically, by integrating the incentive mechanism into the proposed scheme, mobile users have enough incentive to respond to other users' LP requests. Therefore, a large J_1 can be adopted by the system to make the attack non-economic for malicious provers. Moreover, LBSPs can look at the history of LPs that have been issued for a prover to determine if a specific group of witnesses always issue LPs for this prover.

To make the scheme more resistant against P–W collusions, verifiers can adopt a random J_1 in a specific range. Therefore, the dishonest prover does not know what collusion group size he must adopt, making it more challenging to conduct an attack.

Non-Transferability: In the proposed scheme, a transaction added to a block can only be used by its owner, i.e. the prover. The reason is that it is signed by the prover, hence, if a dishonest prover provides another user with his random nonce r, the user cannot claim an LP with the LBSP using this Tx. Therefore, the prover signature makes a Tx non-transferrable.

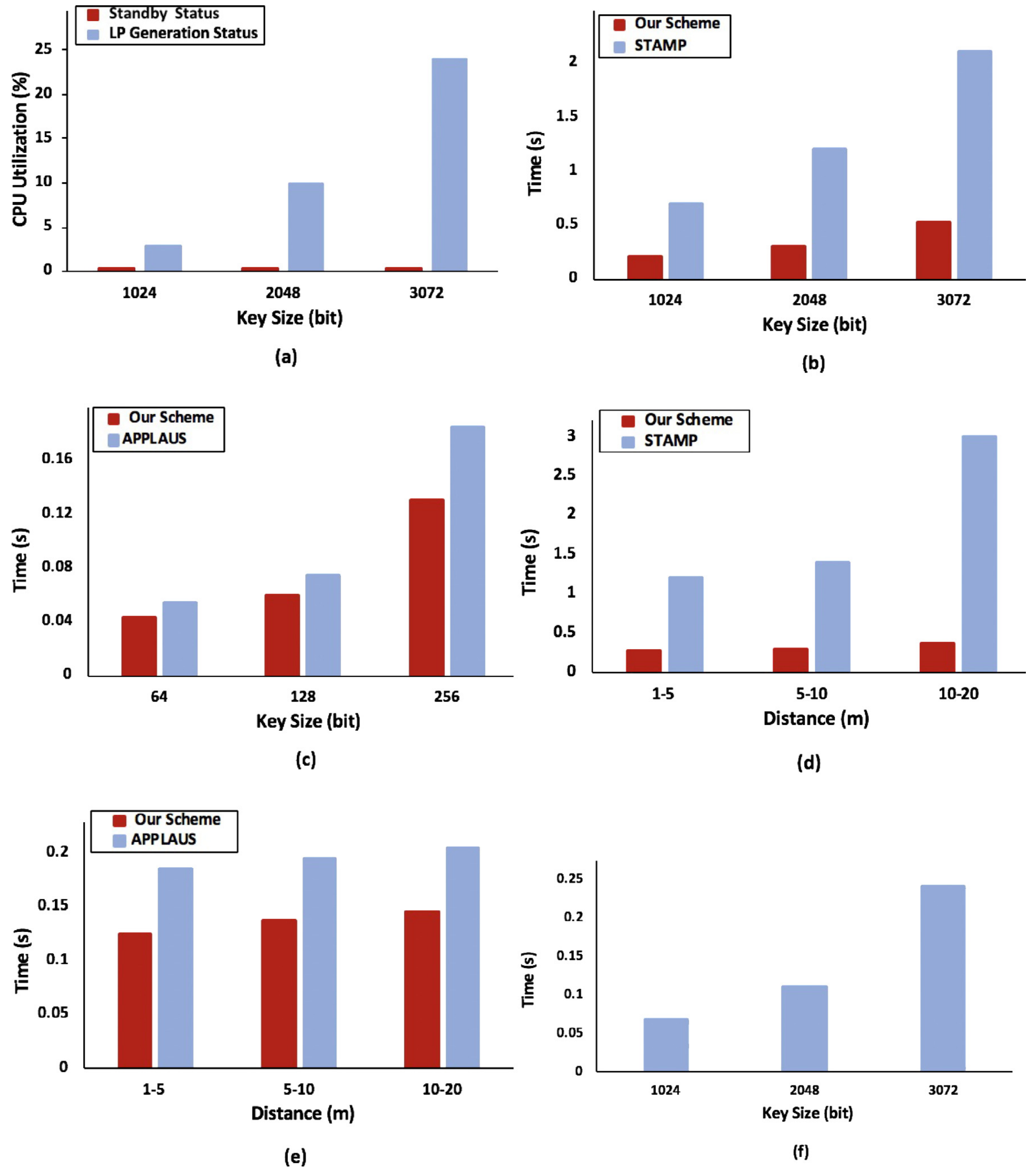


Fig. 6. (a) CPU usage for different key sizes. (b) and (c) Time required for LP generation in our scheme, STAMP, and APPLAUS under different key sizes. In APPLAUS [45], the authors have not implemented their scheme for key sizes larger than 256. (d) and (e) Time required for LP generation over different physical distances. The shown measurements are for the key sizes 2048 (for (d)) and 256 (for (e)). As you see, in our scheme, the level of dependency on physical distance is lower because Bluetooth communications have a smaller share of the amount of time required to generate an LP. (f) Time required for Tx generation after a witness receives message m_3.

7. Implementation results

To study the feasibility of the proposed scheme, we implemented a Java prototype of the proposed scheme on the Android platform. Our experiments were performed on two Android mobile devices: (1) a LG G4–H818P equipped with a Hexa-Core 1.8 GHz processor, 3 GB of RAM, and running Android OS 5.1, acting as a prover, and (2) a Sony Xperia Z1 equipped with a Quad-Core 2.2 GHz processor, 2 GB of RAM, with Android OS 4.4.4, acting as a witness.

We adopted Bluetooth as the communication interface between the mobile devices and conducted the tests in both indoor and outdoor environments. Each measurement shown in this section has been obtained by averaging the results of 10 independent tests. We used RSA key pairs for encryption and SHA1 as the one–way hash function to compute users' signatures. Since the Tx verification phase is performed by verifiers that use desktop or laptop computers with a high level of storage and computational power, we just focus our experiments on *Request Submission*, *Proximity Checking*, and *Tx Generation* phases that are performed

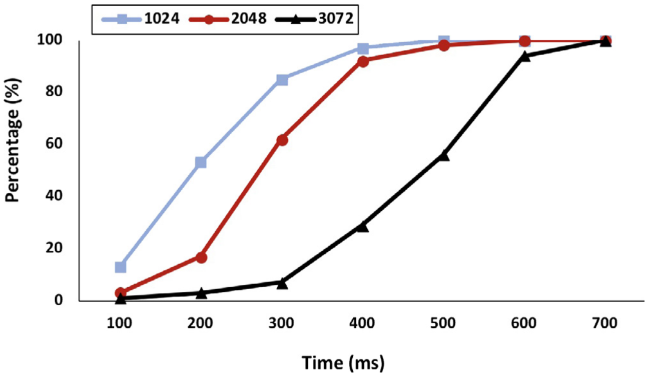


Fig. 7. The percentage of LP requests that successfully pass the P–P collusion detection test for different values of T. As you see, for key sizes 1024, 2048, and 3072, T can be set to 400, 500, and 700 ms, respectively, because for T greater than these values almost all the LP requests successfully pass the time-limited P–P collusion detection test. A specific tolerance can also be considered to support devices with a slower CPU speed.

by mobile devices with limited resources. The implemented code occupies only 64 kB of data memory. Moreover, during the application runtime, less than 1% of the available memory is used. We also recorded the CPU utilization of the code by installing a monitoring application that reports the amount of CPU usage of the processes running on the device. As you see in Fig. 6(a), the CPU usage for a user in the standby mode is almost 0.5% and independent of the key size. However, due to heavy computations required for signature and commitment calculations in the LP generation phases, the average CPU usage increases to 3%, 10%, and 24% for key sizes 1024, 2048, and 3072, respectively.

We measured the amount of time that the proposed scheme requires to generate an LP after the prover device broadcasts m_1. We compared the results to the decentralized schemes STAMP [40] and APPLAUS [45]. Fig. 6(b) and (c) show the results for different key sizes (in APPLAUS, the authors have not implemented their scheme for key sizes larger than 256). As you see, our proposed scheme is faster than STAMP by an order of magnitude. The reason is that we have not adopted a DB protocol to check the prover's proximity to the witness while in STAMP, the Bussard–Bagga DB protocol is used to perform this job. As discussed in Section 4.2, adopting a DB protocol makes an LP generation scheme slow specifically when users select a long private key.

To evaluate the impact of physical distance between mobile users on LP generation, we conduct our experiments for different distances and compare the results to the performance of STAMP and APPLAUS (see Fig. 6(d) and (e), respectively). Compared to them, our scheme's Bluetooth communications have a smaller share in the amount of time required to generate an LP (the number of Bluetooth communications is lower in our scheme since it does not run a DB protocol). Thus, the level of dependency on physical distance is much lower in our scheme. Note that distance only affects the Bluetooth communication latency and does not change the amount of time required for computations performed in mobile devices.

We also examined the computational time required for the witness device to generate a transaction. The results are shown in Fig. 6(f) for different key sizes. As you see, key size has a negative impact on Tx generation latency because the only heavy operation that the witness device needs to perform in the Tx generation process is signature computation.

Finally, to obtain the optimum value for parameter T, we changed it from 100 to 700 ms by steps 100 ms (see Fig. 7) and recorded the percentage of LP requests that passed the P–P collusion detection test. We found that T can be approximately set to 400, 500, and 700 ms for the key sizes of 1024, 2048, and 3072, respectively. These are close to the maximum amount of time that an ordinary device requires to respond to the challenge message sent by a witness device located in its vicinity. If a small value is selected for T, slow mobile devices will fail to sign the challenge message and send it back to the witness in the given time T. Therefore, a specific tolerance can be considered for mobile devices with a slower CPU speed. However, increasing T can provide a malicious prover with the opportunity to successfully conduct a P–P collusion attack.

8. Summary and future work

This paper proposed a blockchain-based, secure, and privacy-aware architecture for LP generation and verification. The target of the proposed scheme is to prevent dishonest mobile users from submitting fake check-ins and location claims with LBSPs. It relies on the collaboration of mobile devices that generate an LP for other mobile devices. We also integrated an incentive mechanism into the proposed scheme to reward mobile users who collaborate with the system. The main strengths of the proposed architecture are the following: (1) It does not require a central trusted authority to operate due to employing the blockchain decentralized architecture. (2) It has reliable performance against P–P and P–W collusions to which the majority of the current schemes are vulnerable. (3) Our prototype implementation shows that the LP generation process in the proposed scheme is faster than the existing schemes due to employing a faster mechanism for proximity checking. (4) It preserves users' location privacy as they commit to their spatiotemporal data before it is published and added to the public ledger. Thus, it is not possible to infer a user's location data by exploring the public ledger.

In the future, we intend to integrate a payment mechanism into the scheme by which users can pay their transaction fees (rewards) through the common existing cryptocurrencies such as Bitcoin and Ethereum. Moreover, utilizing the concept of smart contracts in the scheme design can be another research direction related to this work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] http://www.garcard.com/nikeplus.php.
[2] https://www.higi.com.
[3] http://forbes.com/sites/stevenbertoni/2014/12/08/oscarhealth-using-misfit-wearables-to-reward-fit-customers.
[4] http://technologyreview.com/news/516176/healthinsurers-app-helps-users-track-themselves.
[5] https://en.bitcoinwiki.org/wiki/Simplified_Payment_Verification.
[6] A. Alketbi, Q. Nasir, M.A. Talib, Blockchain for government services–use cases, security benefits and challenges, in: Learning and Technology Conference (L & T), IEEE, 2018, pp. 112–119, http://dx.doi.org/10.1109/LT.2018.8368494.
[7] E. Androulaki, C. Soriente, L. Malisa, S. Capkun, Enforcing location and time–based access control on cloud–stored data, in: Proceedings of IEEE 34th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2014, pp. 637–648, http://dx.doi.org/10.1109/ICDCS.2014.71.
[8] P. Asuquo, H. Cruickshank, J. Morley, C.P. Ogah, A. Lei, W. Hathal, S. Bao, Z. Sun, Security and privacy in location–based services for vehicular and mobile communications: An overview, challenges and countermeasures, IEEE Internet Things J. 5 (2018) 4778–4802, http://dx.doi.org/10.1109/JIOT.2018.2820039.
[9] Y. Baseri, A. Hafid, S. Cherkaoui, K–anonymous location–based fine–grained access control for mobile cloud, in: IEEE Annual Consumer Communications & Networking Conference, 2016, pp. 720–725, http://dx.doi.org/10.1109/CCNC.2016.7444868.

[10] A. Bay, I. Boureanu, A. Mitrokotsa, I. Spulber, S. Vaudenay, The Bussard–Bagga and other distance–bounding protocols under attacks, in: International Conference on Information Security and Cryptology, Inscrypt, Springer, 2012, http://dx.doi.org/10.1007/978-3-642-38519-3_23.

[11] I. Boureanu, A. Mitrokotsa, S. Vaudenay, Practical and provably secure distance bounding, J. Comput. Secur. (2015) 229–257.

[12] I. Boureanu, S. Vaudenay, Challenges in distance bounding, in: IEEE Security & Privacy, vol. 13, IEEE, 2015, pp. 41–48, http://dx.doi.org/10.1109/MSP.2015.2.

[13] L. Bussard, W. Bagga, Distance–bounding proof of knowledge to avoid real–time attacks, in: Security and Privacy in the Age of Ubiquitous Computing, Springer, 2005, http://dx.doi.org/10.1007/0-387-25660-1_15.

[14] F. Casino, T. k Dasaklis, C. Patsakis, A systematic literature review of blockchain–based applications: current status, classification and open issues, Telemat. Inform. 36 (2018) 55–81, http://dx.doi.org/10.1016/j.tele.2018.11.006.

[15] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, R. Du, CertChain: Public and efficient certificate audit based on blockchain for TLS connections, in: IEEE INFOCOM, IEEE, 2018, http://dx.doi.org/10.1109/INFOCOM.2018.8486344.

[16] A. van Cleeff, W. Pieters, R. Wieringa, Benefits of location–based access control: A literature study, in: IEEE/ACM Int'L Conference on Green Computing and Communications, IEEE, 2010, http://dx.doi.org/10.1109/GreenCom-CPSCom.2010.148.

[17] M. Conoscenti, A. Vetro, J.C.D. Martin, Blockchain for the internet of things: A systematic literature review, in: IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), IEEE, 2016, pp. 1–6, http://dx.doi.org/10.1109/AICCSA.2016.7945805.

[18] B. Davis, H. Chen, M. Franklin, Privacy preserving alibi systems, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ACM, 2012, pp. 34–35, http://dx.doi.org/10.1145/2414456.2414475.

[19] H. Delfs, H. Knebl, Introduction to Cryptography, Principles and Applications, third ed., Springer, 2012, http://dx.doi.org/10.1007/3-540-49244-5, (book).

[20] S. Gambs, M.O. Killijian, M. Roy, M. Traore, PROPS: A privacy–preserving location proof system, in: IEEE 33rd International Symposium on Reliable Distributed Systems, IEEE, 2014, http://dx.doi.org/10.1109/SRDS.2014.37.

[21] Z. Gao, H. Zhu, Y. Liu, M. Li, Z. Cao, Location privacy in database–driven cognitive radio networks: Attacks and countermeasures, in: Proceedings of IEEE INFOCOM, IEEE, 2013, pp. 2751–2759, http://dx.doi.org/10.1109/INFCOM.2013.6567084.

[22] M. Grissa, A. Yavuz, B. Hamdaoui, Location privacy preservation in database–driven wireless cognitive networks through encrypted probabilistic data structures, IEEE Trans. Cogn. Commun. Netw. 3 (2017) 255–266, http://dx.doi.org/10.1109/TCCN.2017.2702163.

[23] R. Gupta, U.P. Rao, An exploration to location–based service and its privacy preserving techniques: A survey, Wirel. Pers. Commun. 96 (2017) 1973–2007.

[24] H. Halpin, M. Piekarska, Introduction to security and privacy on the blockchain, in: IEEE European Symposium on Security and Privacy Workshops, IEEE, 2017, http://dx.doi.org/10.1109/EuroSPW.2017.43.

[25] S. Huckle, R. Bhattacharya, M. White, N. Beloff, Internet of things, blockchain and shared economy applications, Procedia Comput. Sci. 98 (2016) 461–466, http://dx.doi.org/10.1016/j.procs.2016.09.074.

[26] C. Javali, G. Revadigar, K.B. Rasmussen, W. Hu, S. Jha, I am Alice, I was in wonderland: Secure location proof generation and verification protocol, in: IEEE 41st Conference on Local Computer Networks (LCN), IEEE, 2016, http://dx.doi.org/10.1109/LCN.2016.126.

[27] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cyptography and privacy–preserving smart contracts, in: IEEE Security & Privacy, IEEE, 2016, pp. 839–858, http://dx.doi.org/10.1109/SP.2016.55.

[28] Y. Li, L. Zhou, H. Zhu, L. Sun, Privacy–preserving location proof for securing large–scale database–driven cognitive radio networks, IEEE Internet Things J. 3 (2016) 563–571, http://dx.doi.org/10.1109/JIOT.2015.2481926.

[29] H. Liu, X. Li, H. Li, J. Ma, X. Ma, Spatiotemporal correlation–aware dummy–based privacy protection scheme for location–based services, in: Proceedings of IEEE INFOCOM, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8056978.

[30] W. Luo, U. Hengartner, VeriPlace: A privacy–aware location proof architecture, in: Proceedings of ACM GIS, ACM, 2010, pp. 23–32, http://dx.doi.org/10.1145/1869790.1869797.

[31] M. Mettler, Blockchain technology in healthcare: The revolution starts here, in: IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, 2016, pp. 1–3, http://dx.doi.org/10.1109/HealthCom.2016.7749510.

[32] M.R. Nosouhi, S. Yu, M. Grobler, Y. Xiang, Z. Zhu, SPARSE: Privacy-aware and collusion resistant location proof generation and verification, in: IEEE GLOBECOM, IEEE, 2018, http://dx.doi.org/10.1109/GLOCOM.2018.8647933.

[33] M.R. Nosouhi, S. Yu, M. Grobler, Q. Zhu, Y. Xiang, Blockchain–based location proof generation and verification, IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (2019).

[34] N. Papadis, S. Borst, A. Walid, M. Grissa, L. Tassiulas, Stochastic models and wide–area network measurements for blockchain design and analysis, in: IEEE INFOCOM, IEEE, 2018, pp. 2546–2554, http://dx.doi.org/10.1109/INFOCOM.2018.8485982.

[35] A. Pham, K. Huguenin, I. Bilogrevic, I. Dacosta, J.P. Hubaux, Secure Run: Cheat–proof and private summaries for location–based activities, IEEE Trans. Mob. Comput. 15 (2016) 2109–2123, http://dx.doi.org/10.1109/TMC.2015.2483498.

[36] S. Saroiu, A. Wolman, Enabling new mobile applications with location proofs, in: Proceedings of ACM HotMobile, ACM, 2009, http://dx.doi.org/10.1145/1514411.1514414.

[37] M. Talasila, R. Curtmola, C. Borcea, Link: Location verification through immediate neighbors knowledge, in: Proceedings of International Conference on Mobile and Ubiquitous Systems, Springer, 2012, pp. 210–223, http://dx.doi.org/10.1007/978-3-642-29154-8_18.

[38] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, IEEE Commun. Surv. Tutor. 18 (2016) 2084–2123, http://dx.doi.org/10.1109/COMST.2016.2535718.

[39] Q.D. Vo, P. De, A survey of fingerprint–based outdoor localization, IEEE Commun. Surv. Tutor. 18 (2016) 491–506, http://dx.doi.org/10.1109/COMST.2015.2448632.

[40] X. Wang, A. Pande, J. Zhu, P. Mohapatra, STAMP: Enabling privacy–preserving location proofs for mobile users, IEEE/ACM Trans. Netw. 24 (2016) 3276–3289, http://dx.doi.org/10.1109/TNET.2016.2515119.

[41] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, nist draft nistir 8202, 2018, available on https://csrc.nist.gov.

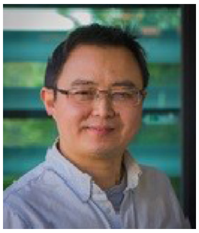
[42] S. Yu, Big Privacy: Challenges and opportunities of privacy study in the age of big data, IEEE Access 4 (2016) 2751–2763, http://dx.doi.org/10.1109/ACCESS.2016.2577036.

[43] Z. Zhang, L. Zhou, X. Zhao, G. Wang, Y. Su, M. Metzger, H. Zheng, B.Y. Zhao, On the validity of geosocial mobility traces, in: Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets), ACM, 2013, http://dx.doi.org/10.1145/2535771.2535786.

[44] Y. Zheng, M. Li, W. Lou, Y.T. Hou, Location based handshake and private proximity test with location tags, IEEE Trans. Dependable Secure Comput. 14 (2017) 406–419, http://dx.doi.org/10.1109/TDSC.2015.2472529.

[45] Z. Zhu, G. Cao, APPLAUS: A privacy–preserving location proof updating system for location–based services, in: Proceedings of IEEE INFOCOM, IEEE, 2011, pp. 1889–1897, http://dx.doi.org/10.1109/INFCOM.2011.5934991.

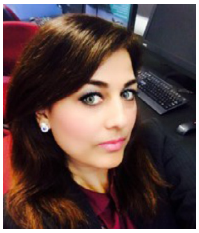
Mohammad Reza Nosouhi received the master's degree from Isfahan University of Technology, Iran. He has worked for more than 10 years in ICT industry in Iran. He is currently pursuing the Ph.D. degree with the School of Computer Science, University of Technology Sydney. Mohammad is also a research scholar with CSIRO's DATA61. He has been teaching in Deakin University and Deakin College since 2017 as a tutor and lecturer, respectively. His research interests include data security and privacy, applied cryptography and AI applications in security.



Shui Yu is currently a Full Professor with the School of Software, University of Technology Sydney, Australia. He initiated the research field of networking for big data in 2013. His h-index is 32. He has published two monographs and over 200 technical papers, including top journals and top conferences, such as the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, the IEEE TRANSACTIONS ON NETWORKING, and in INFOCOM, and edited two books. He actively serves his research communities in various roles. Dr. Yu has served over 70 international conferences as a member of the organizing committee. He is a member of the American Association for the Advancement of Science and the Association for Computing Machinery, and a Distinguished Lecturer of the IEEE Communication Society. He has served as the Publication Chair for the IEEE Globecom 2015 and the IEEE INFOCOM 2016 and 2017, the TPC Chair for the IEEE BigDataService 2015, the General Chair for the ACSW 2017, and the Vice Chair of the Technical Committee on Big Data, IEEE Communication Society. He is currently serving the editorial boards for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the IEEE Communications Magazine, the IEEE INTERNET OF THINGS JOURNAL, the IEEE COMMUNICATIONS LETTERS, the IEEE ACCESS, and the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS.

Wanlei Zhou received the B.Eng. and M.Eng. degrees in computer science and engineering from the Harbin Institute of Technology, Harbin, China, in 1982 and 1984, respectively, the Ph.D. degree in computer science and engineering from The Australian National University, Canberra, ACT, Australia, in 1991, and the D.Sc. degree from Deakin University, Geelong, VIC, Australia, in 2002. He was a Lecturer with the University of Electronic Science and Technology of China, Chengdu, China, Monash University, Melbourne, VIC, Australia, and the National University of Singapore, Singapore, and a System Programmer with HP, Andover, MA, USA. He was an Alfred Deakin Professor and the chair of Information Technology, School of information Technology, Deakin University. He is currently a professor and the chair of School of Software Engineering, University of Technology Sydney. He was the Head of the School of Information Technology twice from 2002 to 2006 and from 2009 to 2015 and an Associate Dean of the Faculty of Science and Technology, Deakin University, from 2006 to 2008. He has published over 300 papers in refereed international journals and refereed international conferences proceedings. His current research interests include distributed systems, network security, bioinformatics, and e-learning. Dr. Zhou was the chair of many international conferences and has been invited to deliver keynote address in many international conferences.

Marthie Grobler is passionate about making cybersecurity more accessible for people in the pathway of the fourth industrial revolution. Her research, management and consulting experience span multiple continents, national and state government departments, and a variety of domains linked with the digital domain. Over the last 12 years, Marthie has worked with organizations, government agencies and education institutions around the world. From applying governance models in the digital space to teaching children in rural areas about technology, her active involvement in her work distils clients with a thorough and effective approach to maximize their cybersecurity knowledge toward practical application. She is passionate about enhancing people's ability to use connected technology more powerfully in an ever-connected world. Her focus is on translating cybersecurity skills to a more digestible format that can easily be adopted by technology users. Marthie currently holds a position as Senior Research Scientist at CSIRO, Data61 in Melbourne, Australia where she drives the research groups work on cybersecurity governance, policies and awareness (see https://goo.gl/JVQq96 for more detail). She is also looking after the groups extended group of cybersecurity Ph.D. scholarship students.



Habiba Keshtiar holds a master's degree in education, a Bachelor of Biomedical Science and a post Graduate Diploma of Teaching from Deakin University. She has also taught a variety of mathematics subjects and Statistics at both Foundation and Diploma levels at different colleges and universities in Victoria over 10 years.

Habiba has been the unit coordinator for Business Analytics and Mathematics at Deakin College. She currently lectures Business Analytics at Deakin College. She has co-authored many textbooks in Analytics and Statistics. Habiba also participate in a research project to help shape the forthcoming Australian and New Zealand edition of Business Statistics. She reviewed and provided feedback regarding first drafts of a newly published Mathematics/ Statistics textbook for McGraw-Hill Education publisher in 2015.