INTERNATIONAL JOURNAL OF DIGITAL EARTH
2022, VOL. 15, NO. 1, 1018–1037
https://doi.org/10.1080/17538947.2022.2087775

Exploiting European GNSS and Ethereum in location proof systems

Gianluca Lax and Antonia Russo

DIIES Dept., University Mediterranea of Reggio Calabria, Reggio Calabria, Italy

ABSTRACT

Location-Based Services (LBSs) are essential in many application contexts like ride-sharing or navigation apps. There are cases where users could gain an advantage by submitting fake locations. The problem faced in this paper concerns the possibility that the geographic location declared by a user is not the actual location in which the user is placed. Some solutions are based on centralized or distributed verification in the literature, and other solutions are based on witnesses or infrastructure. In this paper, we highlight the limitations of such approaches and propose a new scheme that exploits signals coming from satellites to provide trustworthy location proofs, also respecting users' privacy. The proposed approach is decentralized because location proofs are stored by users in a suitably-encrypted way, and a blockchain is adopted to guarantee data integrity and authenticity. We show that the proposed approach overcomes the state of the art through a detailed analysis.

ARTICLE HISTORY
Received 14 February 2022
Accepted 1 June 2022

KEYWORDS
Localization; location privacy; location-based services; blockchain; Galileo

1. Introduction

Location-based services (LBSs) use the geographical locations of users to provide tailored information in several domains. Car navigation systems are indeed the most popular example of LBSs and are helpful in providing real-time traffic information and assistance for drivers. We can also think about location-based tracking data such as floating car data, geo-referenced social network data, and crowdsourced geographic information. Another well-known example is eHealth, where LBSs are particularly useful in outdoor exercise to monitor fitness activity.

In recent years, emergency and disaster recovery applications such as crime mapping, global epidemic infection, or real-time traffic monitoring, gained the attention of location-based service providers (Hu et al. 2021; H. Huang et al. 2018; Sharma, Rani, and Memon 2020). Indeed, even more in these cases, the precise and real-time user's current location is the primary goal for the service provider to optimize the reliability of the application.

The Global Navigation Satellite System (GNSS) is responding to these urgent and primary needs (Kou et al. 2021). As the location-sensitive applications are increasing, the number of GNSS-enabled devices has also boosted by around 6 billion in the last years, with a majority of smartphones able to calculate and process users' positions in real-time (European GNSS Service Centre 2020; Piedrafita et al. 2018). The European Geostationary Navigation Overlay System (EGNOS) broadcasts GNSS-like signals primarily dedicated to providing integrity information and wide-area corrections but can also be used as extra navigation signals.

CONTACT Antonia Russo antonia.russo@unirc.it DIIES Dept., University Mediterranea of Reggio Calabria, Via Graziella, Località Feo di Vito, Reggio Calabria 89122, Italy

To access a service or obtain a particular benefit, users need to prove to have been in a given place at a given time. For this purpose, a location proof (LP) is used, which is a claim attesting that a user is in a specific physical position at a declared time (Zafar et al. 2021; H. Huang et al. 2018).

However, false location reporting is encouraged by the incentives associated with location proofs. For example, a dishonest user may submit fake location proofs to illegally obtain a benefit or access a service (Nosouhi et al. "Blockchain for Secure" 2020). Furthermore, several vulnerabilities expose users to the risk of violating their privacy or improperly using their position.

Various approaches have already been proposed to tackle these challenges. Some proposals require a suitable *infrastructure*, such as trusted access points, typically Wi-Fi access points (Luo and Hengartner 2010). The drawback of this approach is related to the cost of the infrastructure and the scalability performances.

Other proposals are based on *witnesses*, which are mobile entities temporarily present at a specific location and involved in the generation and verification processes (Nosouhi et al. "Pasport: A Secure and Private" 2020). However, the number of these entities should be adequate for the localization space so that such approaches are not effective for broad areas.

Several strategies have been proposed concerning where to store location proofs: in a central server (Talasila, Curtmola, and Borcea 2010), in a distributed way (Nosouhi et al. 2018), or on the user's device (Wang et al. 2016). These choices are also impacted by the location proof size that must be adequate concerning the storage capacity of devices or equipment. The centralized strategy limits the system scalability, whereas the storage in the user's device is preferable for reducing privacy issues.

Recent research contributions (Amoretti et al. 2018; Nosouhi et al. "Blockchain for Secure" 2020; Nasrulin, Muzammal, and Qu 2018) focus on exploiting the blockchain technology to implement fully decentralized location proof systems. A blockchain is a secure distributed ledger where users can generate immutable transactions shared among users (Swan 2015). This emerging technology represents a future trend in the context of location proof systems (Zafar et al. 2020) as a potential approach for creating, storing, and verifying location proofs.

This paper analyzes this problem and proposes a new solution limited to outdoor localization because it is based on satellite signals. The contributions of our solution to the state of the art can be summarized as follows: (1) the generation of the location proof occurs without the need for witnesses; (2) the proof's reliability verification can be carried out only by authorized verifiers; (3) the user's privacy is guaranteed because the user's position is revealed only if the user reveals the proof.

We rely on the GNSS infrastructure and blockchain technology to generate, store and verify location proofs shared among authorized entities in the proposed scheme. We instantiate our solution in a real-life scenario and demonstrate that it is unfeasible both to generate fake location proofs and to guess the users' position starting from shared public information.

It is worth noting that in this paper, we provide the definition of our proposal and an example of its use in a real-life scenario, but no experimental validation is carried out. We cannot experimentally validate our proposal because the signals we need in our solution have the characteristics that the spreading codes are made public after some time of their use. This type of signal is not yet available, but there is a Call for Tenders (European Commission, D. D. I. and Space 2020) that proposes a type of signal to be transmitted encrypted for Commercial Authentication Service based on semi-assisted spreading code authentication: an initial signal capability is expected by 2023 (Scott 2021). The paper is structured as follows. The following section introduces the background information needed to understand the proposal. In Section 3 the most significant proposals of the state of the art related to the addressed problem are reviewed. Section 4 presents the proposed approach used to allow a user to generate and a-posteriori verify location proofs. The security analysis is presented in Section 5. In Section 6, we instantiate the proposed scheme to a specific scenario and provide technical details about how our solution works. Finally, in Section 7, we draw our conclusions.

2. Background

This section recalls some concepts used in this paper: the Global Navigation Satellite System and blockchain technology.

The Global Navigation Satellite System (GNSS) is made of various constellations of satellites that provide autonomous geo-spatial positioning with global coverage (EU Agency for the Space Programme 2021).

Mainly, GNSS includes four constellations: Europe's Galileo (European Commission 2021), China's BeiDou Navigation Satellite System (BeiDou Navigation Satellite System 2021), the USA's NAVSTAR Global Positioning System (GPS) (National Coordination Office for Space-Based Positioning, Navigation, and Timing 2021), and Russia's Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) (Information and Analysis Center for Positioning, Navigation and Timing, Korolyov, Russia 2021). These constellations are composed of more than 100 GNSS satellites in Medium Earth Orbit (MEO) and continue to provide navigation services with global coverage. In addition, to increase the coverage at a regional level, the Regional Navigation Satellite Systems (RNSS) such as the Indian NavIC (Department of Space, Indian Space Research Organisation 2021) and Japanese QZSS (Government of Japan 2021) have been launched. The number of satellites drastically increased in the last ten years as the GNSS-enabled devices, and most of these are smartphones.

GNSS consists of three segments: (1) space segment, made of satellites and signals emitted by them; (2) control segment, stations on Earth monitoring satellites and bringing necessary corrections and additional parameters; (3) user segment, consisting of devices using satellite's signals.

It is feasible to generate a sphere where the user lies, starting from the known satellite position. A three-dimensional position can be calculated with three satellites. The three spheres intersect at two points: the former is not located close to Earth's surface, and the latter is the true position of the receiver. Anyway, considering three satellites, the receiver clock could determine a consistent error in the position estimation. Therefore, at least four visible GNSS satellites are required to estimate the actual receiver position. The GNSS receivers' energy consumption during the signal and navigation message acquisition phases has always gained attention. Indeed, the low transmission rate of navigation messages broadcast by the GNSS satellites results in long download times (several 10 s of seconds) and, therefore in huge device battery consumption. Various solutions have been proposed to solve this problem. For example, the *snapshot positioning* technique consists in storing a digital copy of a received GNSS signal for a very short time: usually, 2–100 ms compared to 1–10 s for a traditional receiver (European GNSS Agency 2020a). Starting from this recorded digital copy, the receiver can calculate the signal frequency and code phase and then the pseudoranges. The satellite ephemeris data and the collected pseudoranges are then input to compute the receiver's accurate position and time. In addition, the remote processing technique, for example, the outsourcing of position estimation to the cloud can be used to reduce further power consumption (European GNSS Agency 2020b).

The second topic we introduce is the blockchain technology. It was first proposed by Nakamoto (2008) with the name of *Bitcoin*. From the beginning, it was discovered and deepened as a disruptive technology (Swan 2015) with a huge potential impact on several digital economy sectors. A blockchain is a distributed ledger able to generate and maintain digital transactions shared among users. Inside the blockchain, users trust the distributed system, without trusting a central authority or a third-party intermediary. Indeed, the blockchain consensus algorithm (Mingxiao et al. 2017) guarantees that every transaction has to be confirmed and validated by blockchain participants. These transactions are stored inside blocks and contain information on the recipient's public address, transaction data, and the cryptographic signature, which guarantees the integrity and authenticity of the transaction itself. Furthermore, every block is linked to the chain by its header where the previous block's hash and a timestamp are embedded.

Generally, the blockchain has some well-known properties, which are immutability, transparency, accountability, and integrity. By exploiting these properties, many applications and improvements are enabled.

A Blockchain network can be *permissionless* or *permissioned* (Helliar et al. 2020). The former, also known as public blockchain, is widely used in cryptocurrencies and financial markets. The latter, also known as private blockchain, has entered the domain of business applications and institutional practices. The main characteristics of permissionless blockchains are anonymity and full transparency of transactions over open-source protocols. In contrast, permissioned blockchains are developed by private entities, and for this reason, the network transparency and participants' privacy are controlled by the organization itself. Hybrid blockchains are used by organizations requiring a private, permission-based system alongside a public permissionless system. This setting allows the organization to control access to data stored in the blockchain. Consortium or federated blockchains are similar to hybrid blockchains but are maintained and shared among multiple organizations.

The introduction of smart contracts, programs executed and distributed over the blockchain infrastructure by making a transaction request, has been proposed in Ethereum (2020), the Blockchain 2.0. Without the need for a third intermediary, a smart contract verifies the occurrence of specific conditions. If these conditions are met, it automatically carries out predetermined actions. Smart contracts are written in Solidity (2020), an object-oriented and high-level language, which makes the creation and the deployment of distributed applications possible (Ethereum dApps 2020).

The two most common blockchain-based digital assets are cryptocurrencies and tokens: cryptocurrencies have their specific blockchain and their name usually derives from the blockchain name (e.g. Ether is the cryptocurrency of Ethereum), whereas tokens allow anyone to create a cryptocurrency without the need to build a blockchain for that cryptocurrency, thus making the process of developing cryptocurrencies faster, simpler, and less expensive.

Besides Bitcoin and Ethereum, there are many other blockchains. *Hyperledger Fabric* is a permissioned blockchain running smart contracts and implementing an open-source private blockchain with a modular architecture (Aggarwal and Kumar 2021). Since it was conceived to enable the creation of private platforms shared among enterprises, the access to the network can be restricted to selected participants (Androulaki et al. 2018).

The blockchain *MultiChain* enables the creation and deployment of private blockchains used inside organizations (Polge, Robert, and Le Traon 2021). Furthermore, fine-grained permissions of users over transactions, assets, blocks, and streams can be set by the system administrator (Greenspan 2015).

Chain Core (Ellervee, Matulevicius, and Mayer 2017) was primarily developed to enable financial services such as the exchange of digital assets among the network participants. It relies on the paradigm of a permissioned blockchain. *Corda* is a distributed ledger platform for performing business and financial agreements among institutions (Brown et al. 2016). Furthermore, both Chain Core and Corda enable the development of smart contracts (Saraf and Sabadra 2018).

IOTA was built to respond to the need for the Internet of Everything. Indeed, it is a permissionless blockchain designed for IoT applications (Silvano and Marcelino 2020). IOTA transactions are free since the platform relies on Tangle, a new data structure based on a directed acyclic graph, which does not need blocks, miners, or any chain (Popov 2018). Even if this platform presents better performance than Ethereum and Hyperledger blockchains, one of its limitations regards the constraints on devices not capable of performing the Proof of Work (Florea 2018).

EOSIO is a public and permissionless blockchain (Eosio 2020). It exploits the Delegated Proof of Stake consensus algorithm to obtain fast transaction time. Nevertheless, the security of EOSIO DApps was questioned (Y. Huang et al. 2020).

FOAM is a decentralized database wholly dedicated to enabling location proofs' creation and verification (Foamspace Corp 2018). It was presented in 2018 and relies on the Ethereum protocol

for sharing decentralized geospatial data and on Low Power Wide Area Networks as radio technology by entities called zones (Victor and Zickau 2018). In addition, however, the system envisions the presence of verifiers that are computers checking zones for fraud, providing proof of accurate data, and obtaining a reward in return.

In Section 4.2, we exploit the discussion about the different blockchains to establish which are more suitable for our purposes.

3. Related work

This section presents the state of the art related to location proof systems and location-based services. Location proof systems provide a digital proof that a user was physically present at the claimed location at a certain time and are used to enable secure location-based services. In recent years, location proof systems have experienced a consistent evolution in technology, architecture, and possible application areas (Zafar et al. 2020). At the same time, the increasing number of location-based services and the pervasiveness of such applications lead to an impressive growth of privacy and security risks for the end-user. Indeed, several studies have already considered implementing location-privacy approaches to identify better and prevent those possible attacks (Wernke et al. 2014; Usama et al. 2019; Mohammadani et al. 2020).

Various protocols have been proposed to tackle the challenges in this field, and they can be divided into two main categories: infrastructure-based and infrastructure-independent location proof systems. The main difference between these two categories is that the former assumes the presence of trusted access points (typically Wi-Fi access points); instead, the latter exploits only users' devices by a short-range communication between near smart devices and does not rely on any added infrastructure. Initially, the first secure location proof systems belonging to each of the two categories relied only on a centralized server. However, the performance and complexity issues with the heavy dependency on trust third parties have contributed to embracing new distributed models. In contrast, distributed location proof systems are characterized by several peers collaborating to generate and validate proofs.

The main actors of a location proof system are the prover, an entity that is willing to obtain location proofs; the witness, a device in proximity to the prover and can create a location proof; and the location authority (LA), which assists the prover in location proof generation.

Location verification through Immediate Neighbors Knowledge (LINK) is proposed by Talasila, Curtmola, and Borcea (2010). The solution is a location authentication protocol based on collaboration among neighbors to verify each other's location claims. The solution does not rely on network and localization infrastructures. Every claim has to be approved by the Location Certification Authority based on users' trust scores and spatio-temporal correlation between the users. The authors measure LINK performances considering environments in which user density is relatively high and, the case of an alone user is not considered.

Luo and Hengartner (2010) consider user privacy a critical design component of VeriPlace, a solution that guarantees users' anonymity from the proof issuer. The Access Point issues an intermediate location proof that certifies the user's presence nearby. Another entity is responsible for issuing final location proofs that also contain the access point's identity. However, these two figures centralize location-proof generation and verification.

SPARSE (Nosouhi et al. 2018) relies on a distributed architecture in which mobile users generate location proofs for each other. No Distance Bounding mechanism is used for secure proximity checking. A time-limited approach is adopted to make the solution resistant to collusion attacks. Witnesses are assumed to be untrusted; instead, the verifier is trusted. Nearby mobile devices communicate through their short-range Bluetooth interface to issue location proofs of location for each other. A mobile device, Certificate Authority, changes its pseudonym periodically to preserve users' location privacy. This mechanism imposes high communication and processing overhead since it needs to change regularly pseudonyms and generate dummy LPs.

A relevant problem includes preserving the privacy of a vehicle in road networks (Memon et al. 2019) and users' sensitive information about their current location and future directions. In the paper of Arain et al. (2018), multiple mix-zones are exploited to enable mobile users to query map services without revealing sensitive location information. Arain et al. (2018) demonstrate that the proposed model assures service usability while effectively preserving map service users' location privacy. Multiple mix zone schemes coupled with a pseudonym change strategy are explored by Memon et al. (2018) to provide trajectory privacy and unlikability for the road network.

In PASPORT (Nosouhi et al. "Pasport: A Secure and Private" 2020), no central trusted entity is required to operate as a witness device. The solution is based on a decentralized architecture designed for ad hoc scenarios in which mobile users can generate LPs for each other. The generation of LPs provides users with added security properties, such as unforgeability and non-transferability of LPs. Furthermore, a privacy-aware distance bounding protocol, P-TREAD, is proposed to further assure privacy-aware proximity checking. Finally, a witness selection mechanism is implemented to address the prover-witness collusions.

The Prover-Witness collusions issue is also faced within STAMP (Wang et al. 2016). The solution is designed for ad-hoc mobile users in a distributed setting and accommodates wireless access points. STAMP ensures the integrity and non-transferability of the location proofs. However, the solution requires a semi-trusted third party, Certificate Authority, responsible for proof verification and trust evaluation. Recent studies on location proof systems are exploring the application of the blockchain technologies for decentralized location proof systems. This technology is exploited to assure location provenance in location proof systems (Zafar et al. 2020). Amoretti et al. (2018) propose a customized proof of stake blockchain where storing proofs of location. Thus, the blockchain participants (i.e. nodes) can verify and store proofs of location, without any centralized supernode that acknowledges the processes. In this solution, communication among neighbors can be implemented through any short-range communication technology. Nosouhi et al. "Blockchain for Secure" (2020) designed a blockchain-based scheme for location proof generation and verification. First, a blockchain transaction is generated as a location proof and is broadcast on the peer-to-peer network. There, the verifiers check it. However, even in this solution, a middle-ware entity is expected, that is the bridge, responsible for broadcasting the issued LPs that have already been issued. Finally, the verified transaction is stored in a time-stamped public ledger accessible for location-based services.

The comparison of the state of the art with our proposal is summarized in Table 1 by analyzing four features that a solution should have: being (i) distributed, (ii) infrastructure-independent, (iii) neighbors-based, and (iv) rewarding-based. Indeed, distributed solutions provide better scalability; infrastructure-based solutions need a cost for the infrastructure; neighbors-based approaches have poor performance under low to medium adoption rates; rewarding introduces a price. Thus, we aim to obtain a solution which is distributed, infrastructure independent, not neighbors-based, and not rewarding-based. As it will be clear after defining our approach, the proposed solution is the only one providing these four features.

Finally, we observe that our solution paper takes origin from the proposal given by Lax and Russo (2021), which exploits Blockchain 1.0 to generate and verify location-aware transactions

Table 1. Comparison with existing solutions.

Approaches	Distributed	Infrastructure independent	Neighbors based	Rewarding based
Talasila, Curtmola, and Borcea (2010)	No	Yes	Yes	No
Luo and Hengartner (2010)	No	No	No	No
Nosouhi et al. (2018); Nosouhi et al. "Pasport: A Secure and Private" (2020) Nosouhi et al. "Blockchain for Secure" (2020)	Yes	Yes	Yes	Yes
Amoretti et al. (2018)	Yes	Yes	Yes	No
Proposed solution	Yes	Yes	No	No

among users and authenticated verifiers. The new architecture uses Blockchain 2.0 to improve the effectiveness and performance of solutions. More precisely, a suitable developed smart contract acts as a trusted third party in the phase of publishing the location proofs on Ethereum and making them available to all blockchain users. Even in the verification phase that is carried out a-posteriori, only the location proofs fulfilling integrity checks are marked as valid by the smart contract. It is worth noting that in every phase of our protocol, privacy mechanisms are implemented to protect location data from unauthorized entities. The reliability of Ethereum, complemented by the high usability derived from its high rate distribution among blockchain users, allows us to state that the proposed solution is the first attempt to exploit Blockchain 2.0 for a location-proof system.

4. Proposed solution

In this section, we present the proposed approach to provide users with the capacity to generate and a-posteriori verifying location proofs. First, we start by giving an overview of the system and define the proposed solution.

4.1. Overview

The general scenario considered in this paper is composed of the following actors:

- a user, an individual who wants to prove a-posteriori to have been in a given position by using *location proofs*;
- a set of *signal sources*, which are entities that transmit a suitable signal and are positioned in such a way that all locations to be monitored are covered redundantly. The reader can assume they coincide with the GNSS satellites: for this reason, in the rest of the paper, we will refer to signal sources also with the term satellites;
- the Ethereum public blockchain;
- a verifier, an entity in charge of verifying the reliability of a location proof.

In this scenario, the reliability of the self-declared user's location should be guaranteed to prevent fraudulent positioning. Moreover, the stored information should be suitably handled to avoid unauthorized parties knowing the user's position. In summary, the goals to reach are:

(1) the user should be able to generate a secure self-declared location proof without the need for witnesses;
(2) verifiers should be able to guarantee the reliability of the proof of location;
(3) the privacy of the proof of location should be assured, that is the position is revealed after the proof is released.

An outline of how our approach works is depicted in Figure 1. The signal sources transmit two types of signals: the first is a plain-text signal used for global positioning; the second is a signal modulated by *direct sequence spreading*, and each specific spreading code changes over time and is kept secret while it is used. The user exploits the first signal to obtain the current position and time (Step 1.a). Furthermore, at the same time, the user stores a digital representation B of the second signal (Step 1.b). The user builds a location proof by suitably encrypting position, time, and B, and this proof is published on the blockchain (Step 2). Observe that the location proof is encrypted before being published to avoid the risk of violating the user's privacy). When the user wants to make available their position at a specific time, the location proof is disclosed in plain text to the verifier (Step 3). The verifier checks the presence of the encrypted location proof on the blockchain (Step 4). Moreover, the spreading codes used by signal sources at the time in which location proof has been built are retrieved and suitably used to check if the position

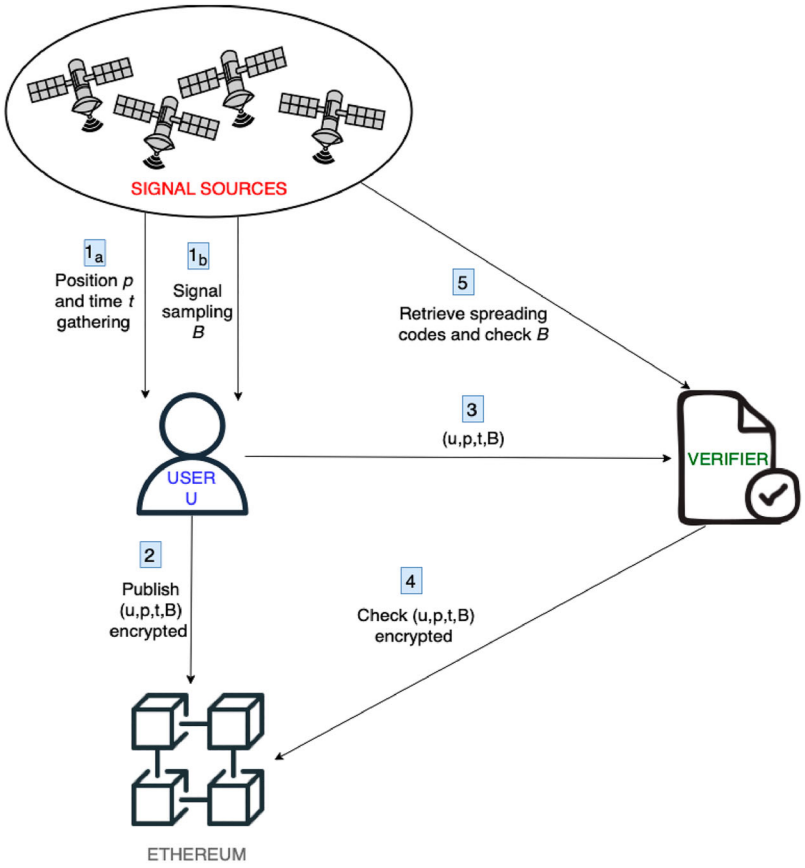

Figure 1. Overview of the solution.

declared by the user is compatible with that extracted by the digital representation B to establish if the declared information by the user is reliable.

4.2. Design choices

Our solution strongly relies on the use of GNSS and the Ethereum blockchain. In this section, we motivate the reasons for these choices.

Global Navigation Satellite Systems are widely exploited in various applications requiring certified geo-referencing and transaction data timestamping. In some cases, GNSS signals need an authentication layer to fight already known attacks such as jamming or spoofing and to deliver reliable applications for commercial use. The GNSS community is striving at making location more robust, both at the receiver, as well as at the signal level. Indeed, the European Space Agency highly recommended the implementation of authentication for the Galileo Open Service (Fernandez-Hernandez, Vecchione, and Díaz-Pulido 2018) by two main services: (i) a Navigation Message Authentication, consisting of the digital signature of Open Service's navigation data to ensure the data authenticity; (ii) a Commercial Authentication Service (CAS) based on the authentication of the Commercial Service signal to protect Galileo signals from spoofing, for commercial applications. Thanks to these services, the European Galileo system enables applications supporting high precision and signal authentication, an important building block for achieving location security. Particularly, the E6 Galileo Commercial Authentication Service includes an authentication feature

thanks to two encrypted signals on the E6 band (1260–1300 MHz), made of a data component (E6-B), allowing the transmission of 448 bits per second, and a pilot component (E6-C). This capability allows not only to authenticate the information encoded in the signal but also to authenticate the signal time of arrival, at least against certain threats and with a certain confidence level (European GNSS Service Centre 2022)

Galileo is a good candidate to play the role of the signal source. Spreading codes of both components of E6 signal can be either encrypted or disseminated in plain. The first option provides spreading code authentication for receivers having the encryption keys. This way, an unpredictable bit-stream generated through a secret key replaces spreading codes, making the signal indistinguishable from the noise for unauthorized receivers. We exploit this type of signal in our proposal: for this reason, we adopt the E6-C Commercial Authentication Service in our solution.

The second choice we made regards the use of the Ethereum blockchain. As seen in Section 2, Blockchain is used in many application contexts, such as financial services, industry 4.0, smart city, healthcare. Permissionless blockchains provide anonymity and full transparency of transactions over open-source protocols and can be accessed by everyone: for these reasons, any permissionless blockchain could be adopted in our solution.

Among the permissionless blockchains, the most used are Ethereum, IOTA, and EOSIO. IOTA is built on Tangle, a directed acyclic graph, and uses proof-of-work for authenticating transactions (Popov 2018). EOSIO uses the delegated proof-of-stake consensus algorithm to reduce transaction fees and increase the transaction rate (Eosio 2020). Among these platforms, in the following, we refer to Ethereum for showing how our proposal works because it is the largest platform for implementing smart contracts.

4.3. Proposal definition

This section is devoted to the formalization of our proposal. We start by introducing the notation used in this paper, which is also summarized in Table 2:

- let U be the user;
- id is an identifier of the user U;
- let S_i be the ith *signal source* in a set of *signal sources*;
- let BC be the Ethereum public blockchain;
- let V be the verifier.
- lp is the location proof generated by U;
- B is the digital representation of a signal;
- SD_i is the seed used by the ith signal source to generate the spreading codes;
- p and t are a position and a timestamp, respectively;
- r denotes a random bit string;

Table 2. Notations used in the rest of the paper.

U	User
id	User's identifier
S_i	The ith signal source
BC	Ethereum Blockchain
V	Verifier
lp	Location proof
B	Digital representation of the signal
SD_i	Seed of a PRNG used by the ith signal source
t, p	timestamp and position
r	Random
$PRNG$	Pseudo-random number generator
H	Cryptographic hash function

- *PRNG* denotes a pseudo-random number generator;
- *H* is a cryptographic hash function;

For the sake of presentation, we do not explore known problems such as satellite clocks, orbit errors, ionospheric and tropospheric delay, multipath, which have been solved for GNSS (NovAtel Inc. 2021). The entities presented in our scenario cooperate as described in the next phases.

(1) *Setup*. This phase is the preliminary step required for the actors to use the proposed schema. First of all, every user exploits the GNSS signal to synchronize their time with that of signal sources in such a way that the synchronization error is less than a given threshold depending on the requested precision of the position. It is well known that users' clocks are usually imprecise with respect to the atomic time provided by GNSS satellites so it is necessary to use four satellites instead of three to estimate the receiver clock's drift from the GNSS scale, as well as the user's three geographical coordinates. Moreover, each device also uses a second synchronization method based on a secure Web time transfer protocol (such as Network Time Security over the Network Time Protocol (Franke et al. 2018) or Roughtime (Malhotra, Langley, and Ladd 2020)). As discussed in Section 5, this secondary method contrasts a specific attack.

Since users exploit Ethereum, in this phase, every user registers an account on Ethereum (specifically, an *External Owned Account*), which is controlled by a pair of public and private keys and identified by a blockchain address. Moreover, the verifier associates each user with their blockchain address.

Finally, in this phase, a suitable smart contract is developed in Ethereum: in Figure 2, we sketch a basic code of the smart contract implementing our solution (this code will be discussed in the following phases). We used (Solidity 2020), an object-oriented and high-level language created to develop the smart contracts in Ethereum.

(2) *Signal Transmission*. This section describes the signal and the data transmitted by signal sources. The *i*th signal source utilizes a secret random seed SD_i to initialize a pseudo-random number generator used to generate a stream of random codes (i.e. $PRNG(SD_i)$). The seed is

```
1  // SPDX-License-Identifier: GPL-3.0
2  pragma solidity ^0.8.7;
3
4  contract LocationProof {
5
6      mapping (bytes32 => address) lpUser;
7      event Publish (address prover, bytes32 lp);
8      event VerifiedProof (address prover, bytes32 lp, uint time);
9
10     function publishLP (bytes32 location_proof) public returns (bool success) {
11         lpUser[location_proof] = msg.sender;
12         emit Publish (msg.sender, location_proof);
13         return true;
14     }
15
16     function verifyLP (string memory position, uint timestamp, bytes32 sample,
           uint random) public returns (uint result) {
17         bytes32  digest = keccak256(abi.encodePacked(msg.sender, position,
               timestamp, sample, random));
18         if (lpUser [digest] == msg.sender){
19         emit VerifiedProof (msg.sender, digest, block.timestamp);
20         return block.timestamp;
21         }
22     }
23 }
```

Figure 2. Sketch of the smart contract.

regenerated after a given amount of time (e.g. each hour). After a seed expires, the signal source publishes its value on a suitable repository along with the starting and ending time when it has been used (this aspect will be discussed in Section 6).

Concerning signal transmission, every signal source transmits data on a carrier of frequency L by direct sequence spreading. The spreading process is done by directly combining the data with the random codes generated by the pseudo-random number generator. The resulting chip signal is transmitted in L band. Observe that the generated signal appears as noise to users because they do not know the spreading codes used.

(3) *Location Proof Generation*. When a user needs to demonstrate to be in a given place at a given time, Algorithm 1 is executed.

The user is provided with a device with GNSS-enabled functionalities (e.g. a smartphone). By this device, u calculates their position and the current time by exploiting the GNSS positioning. Let p and t be the obtained position and time. Thanks to the GNSS-enabled functionalities, the device can receive, amplify, and band-pass filter the signals sent from signal sources described in the previous section. Then, the signal is sampled and converted to obtain a digital representation by the standard hardware illustrated in Figure 3.

In this figure, the Radio Frequency (RF) front-end is a device that converts high RF signal to low RF signal. Indeed, it down-converts and filters RF signals to an intermediate frequency compatible with an analog-to-digital (A/D) converter. The local oscillator supports this process by generating a sinusoidal signal characterized by a frequency such that the receiver can generate the correct resulting frequency. Then, the analog signal is converted by the A/D converter into a digital representation. The user stores the digital representation of the signal starting from the instant t. Let B be this digital representation (how B is used will be clear after reading the next phase). This process is modeled by Line 1 of Algorithm 1.

It is important to observe that the recording of GNSS digital data is an effective methodology that has been recently proposed for several purposes, such as structure health monitoring, ionosphere indices generation, configurable tracking scheme, satellite anomaly monitoring, man-made vulnerability (see Navarro et al. 2019 for details). We highlight that even though the idea of storing digitized intermediate frequency data is recent, the purpose for which we propose its use is new and original.

Now, the user generates a random r (Line 2) and calculates the *location proof* $lp = H(id\|p\|t\|H(B)\|r)$, where id is an identifier of the user (e.g. name and surname), as reported in Line 3 of Algorithm 1.

Then, the user calls the function `publishLP` of the smart contract (see Lines 10–14 in Figure 2) and passes the location proof as a parameter. The smart contract stores the location

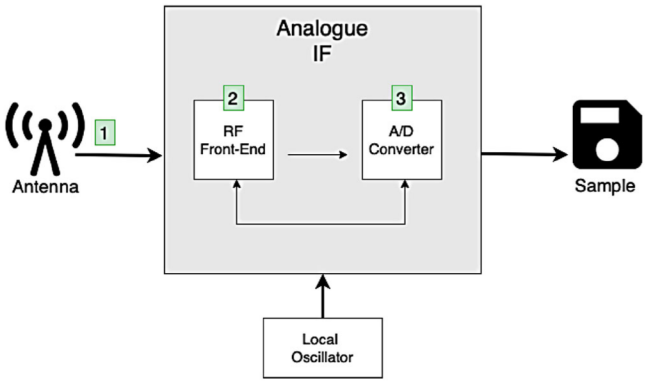

Figure 3. Hardware used for location proof generation (Lisi 2020).

proof *lp* in the mapping `lpUser` that associates the location proof with the Ethereum address of the user.

Finally, the user stores the values p, t, B, and r used to generate the location proof locally (Line 5 of Algorithm 1).

(4) *Location Proof Verification*. This phase is run when the user needs to prove to have been in a given place at a given time for which a location proof *lp* has been generated. The definition of this phase is reported in Algorithm 2.

In this phase, the user discloses the data used to generate *lp*, which are *id*, p, t, B, and r and calls the function `verifyLP` (Lines 16–22 reported in Figure 2) with parameters *id*, p, t, $H(B)$, r.

The smart contract exploits the keccak256 function (Dinur, Dunkelman, and Shamir 2012), a cryptographic function built into the solidity programing language that takes a bit string in input and returns a unique 32 byte hash (digest). The function `verifyLP` checks if this digest is found in the mapping `lpUser` in correspondence of the user's address. If this digest has not been found, then the function returns `null` and the verification fails. Otherwise, the function returns the timestamp $t*$ of the transaction called by the user to publish the location proof.

Now, the verifier checks that this timestamp is equal to the location proof timestamp declared by the user (Lines 2–7 of Algorithm 2). After this, the verification process goes on outside the blockchain.

For each signal source S_i, the verifier retrieves the seed SD_i used at the timestamp t of the location proof generation as described by Line 9 of Algorithm 2 (recall that each satellite publishes this seed after a given amount of time). By calculating the stream of spreading codes as $PRNG(SD_i)$, the verifier can despread the sampled signal B (Line 10) (if any) and obtain the transmission signal timestamp ST_c and satellite position SP_c (Lines 12 and 13 – this process will be better illustrated in Figure 6).

This process must succeed for at least four satellites (see Section 2) and at least four time delays must be obtained, otherwise, the verification process fails (Lines 19–21). At this point, the verifier knows the position and timestamp in which at least four satellites sent the signal and the timestamp in which the user declares to have received it, so that the position of the user can be calculated by the standard GPS trilateration as shown in Figure 4.

At the end, this phase returns the calculated position p_u of the user at the timestamp t.

5. Security analysis

This section is devoted to the analysis of the security of our proposal, which should guarantee the following two properties:

(1) it is unfeasible for an adversary to generate fake location proof;
(2) it is unfeasible for an adversary to know the user's position from a location proof.

Our threat model assumes that:

(1) secrets and random numbers cannot be guessed;
(2) the security properties of cryptographic primitives (hash function and encryption) cannot be violated;
(3) signal sources run the scheme correctly;
(4) blockchain transactions cannot be tampered with.

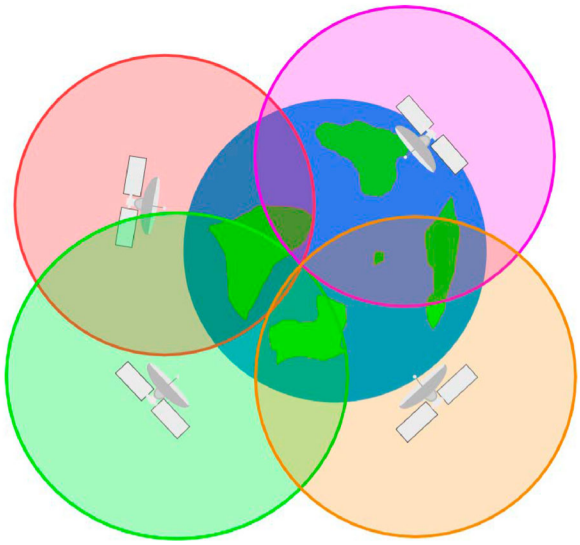

Figure 4. GPS trilateration.

The first attack we analyze aims to generate a fake location proof. In Algorithm 1, Line 3, the location proof is generated from p, t, B, and r. The adversary could generate position p, timestamp t, and the random value r, whereas B is obtained by collecting data sent from signal sources. However, each signal source generates a signal that is modulated by direct sequence spreading, and spreading codes are unknown to the adversary while they are used. For this reason, B cannot be generated by the adversary and must be collected by sampling the signal received in a specific place and time. Suppose the adversary uses old values of B (i.e. a signal collected in a past time). In that case, the timestamp recovered by the verifier from B (Lines 12–13 of Algorithm 2) refers to an old timestamp, and the calculated delay of the transmission between the signal source and the user is higher than the actual one. Then, the calculated distance between the user and each signal source is higher than the actual one, and the position of the user is calculated from the verifier by the standard GPS trilateration returned by the intersection of the spheres (Line 22 of Algorithm 2) can either be empty or a point, not on the surface of the Earth.

The second attack we consider is carried out by GPS spoofing, which consists in feeding the receiver false information so that it computes an erroneous time and location (Humphreys et al. 2008). In general, spoofed signals cause a time bias in the victim's receiver clock, and this type of attack is a serious threat to any time-dependent system. Indeed, an adversary could take control of autonomous vehicles (e.g. drones) that rely on GPS positioning. As discussed in the previous attack, the adversary cannot generate the signal sent from a signal source because spreading codes cannot be guessed. Thus, the idea of the adversary is to collect source signals received in a given location and to broadcast the collected signals with a higher signal strength than the true GPS signal. In this way, the receiver believes that the fake signal is actually the true signal from space and then calculates an erroneous position based on this false signal. We observe that this issue is not specific to our scenario but is a problem well-known in the literature. There are several proposals to prevent spoofing that are based on hardware or software systems, such as obscuring antennas, adding sensors to detect characteristics of signal interference, using more signal types (for example, see Liang et al. 2019, 2022). The last type of solution (i.e. using more signal types) is exploited in our proposal to contrast signal spoofing in addition to standard techniques. Indeed, as seen in the setup phase, each device also uses a second time synchronization method that is exploited to detect the time bias generated by spoofing attacks. Specifically, in this attack, the

adversary must collect the signals in a specific location, transmit them to another location, and then broadcast them to the victim. Observe that these three operations require some time and, in particular, the transmission of the signals to another location takes a time interval depending on the data transfer rate. To contrast this attack, in the setup phase, the device also uses a second-time synchronization method that is not based on GPS. In this way, when the time that the device receives from the GPS signal is older than the time given from the secondary synchronization method and this difference is higher than a given threshold, the received GPS signal is considered counterfeit and is ignored (details on how to set this threshold are given in Section Setup of 6).

In the next attack we analyze, an adversary captures the signals and then publishes them with the id of another user (Line 4 of Algorithm 2). The first operation carried out by the verifier is calling the function `verifyLP` with parameters id, p, t, $H(B)$, r (Line 1 of Algorithm 2). This function checks the blockchain address of the caller (`msg.sender` in Line 18 of Figure 2): recall that 'the verifier associates each user with their blockchain address' in the setup step. Consequently, the attacker should call the function `verifyLP` by using the blockchain address (and, then, the secret key) of the victim, thus violating Assumption 1 of our threat model (i.e. secrets and random numbers cannot be guessed). Concerning the second security property related to user's privacy, we observe that only $lp = H(id\|p\|t\|H(B)\|r)$ is published on the Blockchain (Line 4 of Algorithm 1). The adversary aims to guess p and t by knowing B, which is publicly available from the signal sources. As we assume that the adversary cannot invert hashes, the only possibility is generating several p, t, and r and calculating the digest trying to obtain lp. However, since we assumed that the value r was randomly generated by the user so that it cannot be guessed, also this attack fails.

Finally, we observe that the solution's reliability, and therefore, the reliability of the generated location proof, is based on the features of the blockchain. The Ethereum network counts many nodes that work to keep alive the network, ensuring the reliability of the blockchain-based solutions.

6. Use case

There are many cases in which proving to be in a given place can be useful: Think of a football fan who wants to show that he has attended the champions league final or a war correspondent who has to prove to be on the battlefield. To show how our solution works, we instantiate the presented model to a real-life scenario. We consider a user who participates in a lottery intended for shop visitors. In case of a win, the visitor should prove afterward to an authorized party (i.e. the verifier), such as the retailer, to have been in that place at the lottery time. Among all scenarios, we chose this one because it clearly points out the proof generation time and the proof verification time, which helps the reader follow the protocol. In the chosen scenario, we assume that used signals are available in the proximity of the shop.

The instantiation of the procedures carried out by the actors is described in the following.

(1) *Setup*. Visitors must be equipped with a Galileo-enabled device. In this phase, they have to synchronize their time with the Galileo signal to obtain the precise time at the moment of location-proof submission.

Moreover, the device also uses a secondary synchronization method based on a secure Web time transfer protocol. Although this secondary synchronization method is less precise than Galileo (indeed, the synchronization error is up to 300 ms O'Driscol, Keating, and Caparra 2020), this redundancy is helpful to protect against attacks exploiting fake Galileo signals discussed in Section 5. Specifically, if the time that the device receives from the GPS signal is older than the time given from the secondary synchronization method and this difference is higher than the synchronization error, the received GPS signal is considered counterfeit and is ignored. Time is also periodically synchronized to correct clock drift.

Moreover, in this phase, the visitor creates an Ethereum account (typically, with the help of a

suitable platform). Then, a pair of private and public keys are generated, and the Ethereum address is generated as a digest of the public key.

(2) *Signal Transmission*. We adopt the Galileo E6-C Commercial Authentication Service (CAS) (Ávila Rodríguez 2011) transmitted in *L* band centered at 1278.750 MHz. Each satellite utilizes a specific 5115-bit spreading code to generate data with a chipping rate 5115 chips per second (these are standard parameters (Ávila Rodríguez 2011)). Codes are generated to fulfill properties of randomness as well as possible.

(3) *Location Proof Generation*. In this phase, the visitor enters the store and starts the procedure of location proof generation.

Figure 5 shows an example of how our solution works on the user's device. The user can locate and publish the location proofs when needed.

First of all, the visitor calculates their position p and the current time t by exploiting the Galileo positioning. Then, by the visitor's device, the signals sent from satellites are sampled for a short time that depends on the receiver's antenna, sky visibility, and the surrounding environment's conditions. Typically, a value of 0.1 s is enough for most situations (see snapshot positioning discussed in Section 2). The obtained sample B has a size of about 1 hundred kilobytes.

Thanks to their limited size, location proofs can be saved locally through the application installed on the user's device.

Each time the user locates, a copy of the location proof is saved locally on the device, and a blockchain transaction starts to call `publishLP (lp)` of the smart contract, where *lp* is the

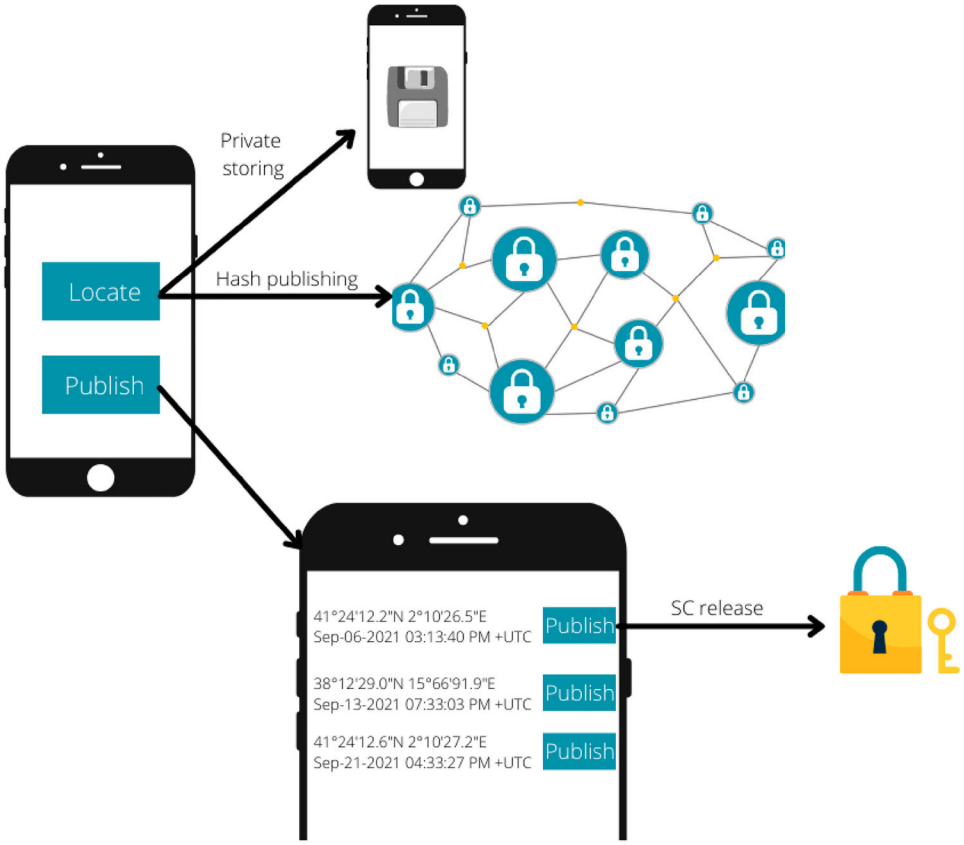

Figure 5. Example of the user's app.

location proof computed as described in Section 4. This transaction calls the function that enables the publication of the location proof hash on Ethereum.

(4) *Location Proof Verification*. Assume that the visitor wins the lottery and needs to prove their position in time. For this purpose, the user selects one of the saved location proof and the device carries out the verification phase. The function `verifyLP` is called with parameters *id*, *p*, *t*, *H*(*B*), *r*.

The retailer searches for the transaction hash over Ethereum and verifies the authenticity of the visitor's Ethereum account. Then, the verifier ensures that the declared timestamp is acceptable, that is, the timestamp is during the lottery validity time. At this stage, the retailer obtains the spreading codes used by the satellites in correspondence with the timestamp *t* declared by the visitor. Recall that after a predetermined time (e.g. a few seconds after they expire), the satellite publishes the seed *SD* used to generate spreading codes. About how the retailer obtains the spreading code, a simple and effective solution is that they are made available at the website of the European GNSS Service Centre (i.e. https://www.euspa.europa.eu/): the use of the HTTPS protocol guarantees codes authentication.

The retailer calculates *PRNG*(*SD*) and obtains the spreading codes used by the satellite at the time t_x declared by the user in the location proof (first line in Figure 6). Now, the retailer compares and aligns the bits sequence generated by the satellite with the sample *B* included in the location proof. This task is common to many digital systems, and there is extensive literature on it (for example, see Sarwate 1997). Figure 6 briefly explains this process: therein, rectangles represent bit data. In contrast, squares represent chips (remember that the Galileo E6-C CCAS utilizes 5115-bit spreading code to generate data with a chipping rate 5115 chips per second).

The first line of Figure 6 represents the spreading codes recovered by *PRNG*(*SD*), which appear random (this is denoted by the symbol '?' in the squares). Now, the verifier attempts to align these codes with the ones extracted by *B*. In the figure, we disclosed the codes starting this alignment '1 0 1' (for the sake of presentation, we showed only the beginning of this alignment). From this alignment, the verifier can calculate the transmission delay, which is 69.5 ms in the example considered in Figure 6: indeed, the user has stored in *B* 69.5 ms later what the satellite has sent at the timestamp t_x. This process is repeated for all the satellites according to Algorithm 2.

In the end, the retailer can calculate and check the visitor's position and time by the standard methods used in Galileo, as described in Section 4. This concludes the generation and verification of location proofs.

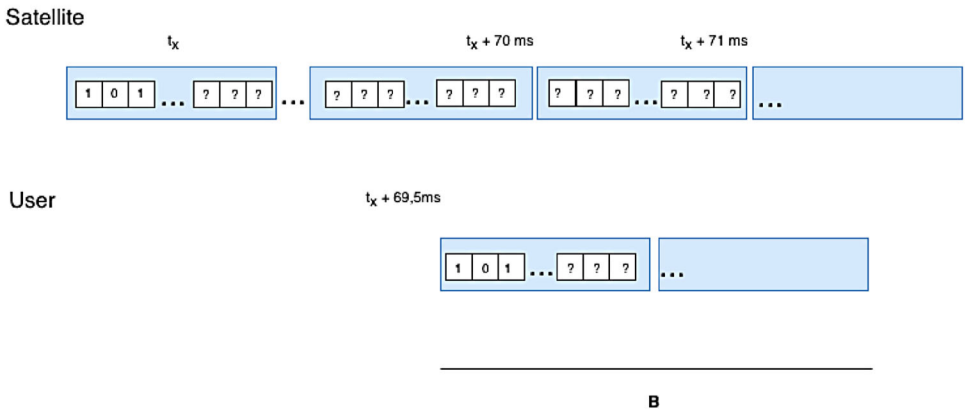

Figure 6. Computation of signal transmission delay.

Now, we discuss the cost analysis of our solution. On the Ethereum network, gas refers to the unit that measures the computational effort required to execute specific operations. A fee, paid in Ether, is required to conduct a transaction successfully on Ethereum. Gas prices are denoted in gwei, which is a denomination of ETH, and each gwei is equal to 10^{-9} ETH (Ethereum 2021).

Appendix G of the Ethereum Yellow Paper (Wood 2014) gives a broad overview of how these gas prices are calculated. Indeed, each abstract operation that a transaction may affect corresponds to a tuple of scalar values related to their relative costs, in gas. We calculate the contract deployment costs, which is 226280 gwei (in May 2022, this is about 0.57$). The call to the function `publishLP` (Lines 10–14) of the smart contract reported in Figure 2 costs 45414 gwei (in May 2022, this is about 0.11$). Instead the call to the function `verifyLP` (Lines 16–22 reported in Figure 2) costs 26964 gwei (in May 2022, this is about 0.07$).

In the considered scenario, if we suppose a weekly lottery, a regular visitor who participates in all the lotteries submits about 4 location proofs for a month. Thus, the total cost spent by the visitor is about 50 cents for a month and about 6$ for an annual subscription.

7. Conclusion

In this paper, we propose a solution allowing users to self-declare their physical position in time without any witnesses. The resulting schema benefits from the features of the Ethereum blockchain. Indeed, by exploiting a suitable smart contract, users can declare, store, and a-posteriori verify their location proofs in a reliable, secure, and transparent way. Notably, any authorized verifier can check the validity of such proofs. Furthermore, the integration of Galileo Commercial Authentication Service in our protocol assures the integrity and reliability of the received signals.

The limitation of this study is related to the experimental validation, which cannot be done at this moment because the signals used in our solution are not yet available. However, by a Call for Tenders of the European Commission, this type of signal is expected to be available in 2023. Thus, our next step will be the implementation of the system and its experimental validation.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statement

Data sharing not applicable – no new data generated.

ORCID

Gianluca Lax http://orcid.org/0000-0002-5226-0870
Antonia Russo http://orcid.org/0000-0002-3038-8574

References

Aggarwal S., and N. Kumar. 2021. "Hyperledger." In *Advances in Computers*, 323–343, Vol. 121. Amsterdam: Elsevier.

Amoretti M., G. Brambilla, F. Medioli, and F. Zanichelli. 2018. "Blockchain-Based Proof of Location." In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 146–153. IEEE.

Androulaki E., A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, et al. 2018. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains." In *Proceedings of the Thirteenth EuroSys Conference*, Porto, Portugal, 1–15.

Arain Q. A., I. Memon, Z. Deng, M. H. Memon, F. A. Mangi, and A. Zubedi. 2018. "Location Monitoring Approach: Multiple Mix-zones with Location Privacy Protection Based on Traffic Flow Over Road Networks." *Multimedia Tools and Applications* 77 (5): 5563–5607.

Ávila Rodríguez J. A. 2011. "Galileo Signal Plan." Accessed 15 July 2021. https://gssc.esa.int/navipedia/index.php/Galileo_Signal_Plan.
BeiDou Navigation Satellite System. 2021. Accessed 15 July 2021. http://en.beidou.gov.cn.
Brown R. G., J. Carlyle, I. Grigg, and M. Hearn. 2016. "Corda: An Introduction." *R3 CEV, August* 1 (15): 14.
Department of Space, Indian Space Research Organisation. 2021. "Indian Regional Navigation Satellite System (IRNSS): Navic." Accessed 15 July 2021. https://www.isro.gov.in/irnss-programme.
Dinur I., O. Dunkelman, and A. Shamir. 2012. "New Attacks on Keccak-224 and Keccak-256." In *International Workshop on Fast Software Encryption*, 442–461. Springer.
Ellervee A., R. Matulevicius, and N. Mayer. 2017. "A Comprehensive Reference Model for Blockchain-Based Distributed Ledger Technology." In *ER Forum/Demos*, Valencia, Spain, 306–319.
Eosio. 2020. "Eosio Blockchain Official Website." https://eos.io.
Ethereum. 2020. Accessed 13 January 2021. https://www.ethereum.org.
Ethereum. 2021. "Gas and Fees." https://ethereum.org/en/developers/docs/gas/.
Ethereum dApps. 2020. "Explore Decentralized Applications." Accessed 13 January 2021. https://www.stateofthedapps.com.
EU Agency for the Space Programme. 2021. "What is GNSS?" Accessed 15 July 2021. https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss.
European Commission. 2021. "Galileo." Accessed 15 July 2021. https://ec.europa.eu/growth/sectors/space/galileo_en.
European Commission, D. D. I. and Space. 2020. "Test Platform on Galileo HAS/CAS/OSNMA." https://etendering.ted.europa.eu/document/document-file-download.html?docFileId=82511.
European GNSS Agency. 2020a. "GNSS User Technology Report." Accessed 15 July 2021. https://www.euspa.europa.eu/simplecount_pdf/tracker?file=uploads/technology_report_2020.pdf.
European GNSS Agency. 2020b. "Power-Efficient Positioning for the Internet of Things: Merging GNSS with Low-Power Connectivity Solutions." Accessed 15-July-2021. https://www.euspa.europa.eu/newsroom/news/power-efficient-positioning-iot.
European GNSS Service Centre. 2020. "GNSS a Key Element of All-Purpose, User-Driven Positioning Solutions." Accessed 15 July 2021. https://www.gsc-europa.eu/news/gnss-a-key-element-of-all-purpose-user-driven-positioning-solutions.
European GNSS Service Centre. 2022. "Paving the Way to New Galileo Accuracy and Authentication Services." Accessed 1 February 2022. https://www.gsc-europa.eu/news/paving-the-way-to-new-galileo-accuracy-and-authentication-services-galileo-e6-bc-codes-now.
Fernandez-Hernandez I., G. Vecchione, and F. Díaz-Pulido. 2018. "Galileo Authentication: A Programme and Policy Perspective." In *69th International Astronautical Congress*, Adelaide, Australia.
Florea B. C. 2018. "Blockchain and Internet of Things Data Provider for Smart Applications." In *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, 1–4. IEEE.
Foamspace Corp. 2018. "Foam Whitepaper." Accessed 10 April 2022. https://foam.space/publicAssets/FOAM_Whitepaper.pdf.
Franke D., D. Sibold, K. Teichel, M. Dansarie, and R. Sundblad. 2018. "Network Time Security for the Network Time Protocol." In *Internet Draft, draft-ietf-ntp-using-nts-for-ntp-11*. IETF 100, Singapore, Singapore.
Government of Japan. 2021. "Quasi-Zenith Satellite System (QZSS)." Accessed 15 July 2021. https://qzss.go.jp/en.
Greenspan G. 2015. *Multichain Private Blockchain-White Paper*, Vol. 1, 1–17. *http://www.multichain.com/download/MultiChain-White-Paper.pdf*.
Helliar C. V., L. Crawford, L. Rocca, C. Teodori, and M. Veneziani. 2020. "Permissionless and Permissioned Blockchain Diffusion." *International Journal of Information Management* 54: Article ID 102136.
Hu T., S. Wang, B. She, M. Zhang, X. Huang, Y. Cui, J. Khuri, et al. 2021. "Human Mobility Data in the Covid-19 Pandemic: Characteristics, Applications, and Challenges." *International Journal of Digital Earth* 14 (9): 1126–1147.
Huang H., G. Gartner, J. M. Krisp, M. Raubal, and N. Van de Weghe. 2018. "Location Based Services: Ongoing Evolution and Research Agenda." *Journal of Location Based Services* 12 (2): 63–93.
Huang Y., H. Wang, L. Wu, G. Tyson, X. Luo, R. Zhang, X. Liu, G. Huang, and X. Jiang. 2020. "Understanding (Mis)Behavior on the EOSIO Blockchain." *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4 (2): 1–28.
Humphreys T. E., B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner. 2008. "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer." In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, Savannah, Georgia, 2314–2325.
Information and Analysis Center for Positioning, Navigation and Timing, Korolyov, Russia. 2021. "Glonass." Accessed 15 July 2021. https://www.glonass-iac.ru/en.
Kou R., B. Yang, Z. Dong, F. Liang, and S. Yang. 2021. "Mapping the Spatio-temporal Visibility of Global Navigation Satellites in the Urban Road Areas Based on Panoramic Imagery." *International Journal of Digital Earth* 14 (7): 807–820.

Lax G., and A. Russo. 2021. "A-Posteriori Privacy-Preserving Tracing Mechanism Exploiting Satellites to Prevent Fraudulent Positioning." In *72nd International Astronautical Congress (IAC)*. International Astronautical Federation (IAF).

Liang C., M. Miao, J. Ma, H. Yan, Q. Zhang, and X. Li. 2022. "Detection of Global Positioning System Spoofing Attack on Unmanned Aerial Vehicle System." *Concurrency and Computation: Practice and Experience* 34 (7): e5925.

Liang C., M. Miao, J. Ma, H. Yan, Q. Zhang, X. Li, and T. Li. 2019. "Detection of GPS Spoofing Attack on Unmanned Aerial Vehicle System." In *International Conference on Machine Learning for Cyber Security*, 123–139. Springer.

Lisi M. 2020. "Gnss User Technology Report 2020." *GEOmedia* 24 (5): 1–108.

Luo W., and U. Hengartner. 2010. "Veriplace: A Privacy-Aware Location Proof Architecture." In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA, USA, 23–32.

Malhotra A., A. Langley, and W. Ladd. January 2020. "Roughtime." Internet-Draft draft-roughtime-aanchal-04, Internet Engineering Task Force. Work in Progress.

Memon I., L. Chen, Q. A. Arain, H. Memon, and G. Chen. 2018. "Pseudonym Changing Strategy with Multiple Mix Zones for Trajectory Privacy Protection in Road Networks." *International Journal of Communication Systems* 31 (1): e3437.

Memon I., H. T. Mirza, Q. A. Arain, and H. Memon. 2019. "Multiple Mix Zones De-correlation Trajectory Privacy Model for Road Network." *Telecommunication Systems* 70 (4): 557–582.

Mingxiao D., M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun. 2017. "A Review on Consensus Algorithm of Blockchain." In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2567–2572. IEEE.

Mohammadani K. H., K. A. Memon, I. Memon, N. N. Hussaini, and H. Fazal. 2020. "Preamble Time-division Multiple Access Fixed Slot Assignment Protocol for Secure Mobile Ad Hoc Networks." *International Journal of Distributed Sensor Networks* 16 (5): 1–18. doi:10.1177/1550147720921624.

Nakamoto S. 2008. "Bitcoin: A Peer-To-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.

Nasrulin B., M. Muzammal, and Q. Qu. 2018. "A Robust Spatio-Temporal Verification Protocol for Blockchain." In *International Conference on Web Information Systems Engineering*, 52–67. Springer.

National Coordination Office for Space-Based Positioning, Navigation, and Timing. 2021. "GPS: The Global Positioning System." Accessed 15 July 2021. https://www.gps.gov.

Navarro V., R. Dittrich, K. Skaburskas, Y. Ying, M. Begin, and F. Pérez-López. 2019. "Big Data GNSS for Intermediate Frequency Recording Stations." In *Big Data from Space (BIDs 2019) Conference*. Munich, Germany.

Nosouhi M. R., K. Sood, S. Yu, M. Grobler, and J. Zhang. 2020. "Pasport: A Secure and Private Location Proof Generation and Verification Framework." *IEEE Transactions on Computational Social Systems*7 (2): 293–307.

Nosouhi M. R., S. Yu, M. Grobler, Y. Xiang, and Z. Zhu. 2018. "Sparse: Privacy-Aware and Collusion Resistant Location Proof Generation and Verification." In *2018 IEEE Global Communications Conference (GLOBECOM)*, 1–6. IEEE.

Nosouhi M. R., S. Yu, W. Zhou, M. Grobler, and H. Keshtiar. 2020. "Blockchain for Secure Location Verification." *Journal of Parallel and Distributed Computing* 136: 40–51.

NovAtel Inc. 2021. "GNSS Error Sources." Accessed 15 July 2021. https://novatel.com/an-introduction-to-gnss/chapter-4-gnsserror-sources/error-sources.

O'Driscol C., S. Keating, and G. Caparra. 2020. "A Performance Assessment of Secure Wireless Two-Way Time Transfer." In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 3938–3951. [Online].

Piedrafita R., R. Béjar, R. Blasco, A. Marco, and F. J. Zarazaga-Soria. 2018. "The Digital 'connected' Earth: Open Technology for Providing Location-based Services on Degraded Communication Environments." *International Journal of Digital Earth* 11 (8): 761–782.

Polge J., J. Robert, and Y. Le Traon. 2021. "Permissioned Blockchain Frameworks in the Industry: A Comparison." *Ict Express* 7 (2): 229–233.

Popov S. 2018. "The Tangle." *White Paper* 1 (3): 1–28.

Saraf C., and S. Sabadra. 2018. "Blockchain Platforms: A Compendium." In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 1–6. IEEE.

Sarwate D. V. 1997. "Acquisition of Direct-Sequence Spread-Spectrum Signals." In *Wireless Communications*, 121–145. Springer.

Scott L. 2021. "GPS & Galileo Civil Signal Authentication." https://www.gps.gov/governance/advisory/meetings/2021-12/scott.pdf.

Sharma R., S. Rani, and I. Memon. 2020. "A Smart Approach for Fire Prediction Under Uncertain Conditions Using Machine Learning." *Multimedia Tools and Applications* 79 (37): 28155–28168.

Silvano W. F., and R. Marcelino. 2020. "Iota Tangle: A Cryptocurrency to Communicate Internet-of-things Data." *Future Generation Computer Systems* 112: 307–319.

Solidity. 2020. Accessed 13 January 2021. https://solidity.readthedocs.io/en/v0.5.8.

Swan M. 2015. *Blockchain: Blueprint for a New Economy*. Boston, USA: O'Reilly Media, Inc.

Talasila M., R. Curtmola, and C. Borcea. 2010. "Link: Location Verification Through Immediate Neighbors Knowledge." In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, 210–223. Springer.

Usama M., O. Rehman, I. Memon, and S. Rizvi. 2019. "An Efficient Construction of Key-dependent Substitution Box Based on Chaotic Sine Map." *International Journal of Distributed Sensor Networks*15 (12): 1–9. doi:10.1177/1550147719895957.

Victor F., and S. Zickau. 2018. "Geofences on the Blockchain: Enabling Decentralized Location-Based Services." In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 97–104. IEEE.

Wang X., A. Pande, J. Zhu, and P. Mohapatra. 2016. "Stamp: Enabling Privacy-preserving Location Proofs for Mobile Users." *IEEE/ACM Transactions on Networking* 24 (6): 3276–3289.

Wernke M., P. Skvortsov, F. Dürr, and K. Rothermel. 2014. "A Classification of Location Privacy Attacks and Approaches." *Personal and Ubiquitous Computing* 18 (1): 163–175.

Wood G. 2014. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." *Ethereum Project Yellow Paper* 151 (2014): 1–32.

Zafar F., A. Khan, A. Anjum, C. Maple, and M. A. Shah. 2020. "Location Proof Systems for Smart Internet of Things: Requirements, Taxonomy, and Comparative Analysis." *Electronics* 9 (11): 1776.

Zafar F., A. Khan, S. U. R. Malik, M. Ahmed, C. Maple, and A. Anjum. 2021. "Mobchain: Three-way Collusion Resistance in Witness-oriented Location Proof Systems Using Distributed Consensus." *Sensors*21 (15): 5096.