A Blockchain-based Solution to Fake Check-ins in Location-Based Social Networks

Sara Migliorini
sara.migliorini@univr.it
Department of Computer Science,
University of Verona
Verona, Italy

Mauro Gambini
mauro.gambini@univr.it
Department of Computer Science,
University of Verona
Verona, Italy

Alberto Belussi
alberto.belussi@univr.it
Department of Computer Science,
University of Verona
Verona, Italy

ABSTRACT

Location-Based Social Networks (LBSNs) are an emerging kind of social network in which users can share their position with others and talk about visited places, providing comments and recommendations. Some LBSNs encourage the voluntary submission of place reviews by offering to users some sort of reward for this activity. However, soon or later this possibility has lead to fraudulent behaviours, in which attackers try to perform fake check-ins in order to increase the obtained rewards without actually visiting any place. Several different solutions have been proposed for distinguishing between real and fake check-ins with a certain degree of confidence. In this paper, we propose an alternative solution based on the use of the emerging blockchain technology, where a decentralized service can provide reliable presence claims for users.

CCS CONCEPTS

• **Information systems → Geographic information systems**.

KEYWORDS

Blockchain, Smart Contracts, Proof of Location, Location-Based Social Network, Fake Check-in

ACM Reference Format:
Sara Migliorini, Mauro Gambini, and Alberto Belussi. 2019. A Blockchain-based Solution to Fake Check-ins in Location-Based Social Networks. In *3rd ACM SIGSPATIAL International Workshop on Analytics for Local Events and News (LENS'19), November 5, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3356473.3365191

1 INTRODUCTION

Location-Based Social Networks (*LBSNs*) are a particular kind of social network offering some location-based services to their users, such as the possibility to share their position with others. People can use LBSNs to communicate with their friends and share their experiences through various online activities, such as sharing the attended events, commenting other users' activities, recommending places or uploading media files. Among these, *checking-in* is the fundamental one, it consists in a simple action that informs your friends when and where you are in a given place and can be used to find others who are nearby. Once being checked-in, a user can share photos, write reviews, comment on posts, leave tips and interact with other users. Through these functionalities, users may be enrolled for voluntary advertising and location recommendation, eventually encouraged to do so through rewards of various kind, including economic ones. However, this possibility soon or later leads to fraudulent behaviors: in order to obtain the reward, users can cheat about their real position and try to provide recommendations or reviews about places that they have never visited. This fake check-ins can affect the reputation of the overall system. For instance, Foursquare[1] adopts check-ins to give users new information and award badges, but it had to develop a combination of approaches to prevent attackers from checking-in too often or when they are not physically present in a place.

Several efforts have been made for defining mechanisms able to prevent fake check-ins in LBSNs, some of them are summarized in Sect. 2. The root cause of this vulnerability is called *location cheating*, namely the lack of a proper location verification mechanism. In this paper, we explore an alternative approach for preventing fake check-ins, which is based on the concept of *Proof of Location* (*PoL*) and makes use of the emerging blockchain technology. The goal of PoL is certify the position of an object or agent during a given time interval and in principle such certification can be built around a decentralized consensus mechanism. In this way, the blockchain technology can offer an infrastructure for both providing a certificate about the user position and building an online market for user reviews. To the best of our knowledge this is the first time that a blockchain-based solution for fake check-ins is explored. The proposed solution is made upon three main blocks described in Sect. 3: blockchain, smart contracts and oracles. Sect. 4 describes how they can be integrated to provide a solution for the fake check-in problem.

2 RELATED WORK

In [6] the authors propose the use of honeypots for preventing fake check-ins through the identification of suspicious user behaviours. Such honeypots are essentially fake venues where no one should be present. When a user perform a check-in in one of such locations, he is automatically flagged as a cheater. Unfortunately, such technique requires the continuous creation of a large number of plausible honeypot venues. A different approach is proposed in [1, 9, 11], where the authors focus on the prevention of sybil attacks, in which a user can obtain multiple fake identities and perform several contemporary check-ins in multiple places. Finally, in [4] the authors propose

[1] Foursquare https://foursquare.com/

LENS'19, November 5, 2019, Chicago, IL, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6958-9/19/11…$15.00
https://doi.org/10.1145/3356473.3365191

a solution based on the analysis of the historical user check-ins for finding anomalous or unusual behaviour. Each of these approaches have their peculiar strengths, but none of them can exclude false positives.

FOAM [3], Platin [10] and XYO [8] are three active projects that provide a PoL service for blockchain platforms. The aim of these projects is to build a location layer for smart contracts that is based on a protocol through which nodes can provide to customers a proof of their presence in a certain position at a given time. These projects are moving the first step towards the integration between the geo-spatial and the blockchain technologies. The main differences between them reside on the underlying mechanisms used for acquiring spatial locations and the way the consensus about locations is achieved. FOAM is an hardware-based solution for replacing GPS which is based on low power wide area networks over unlicensed radio spectrum; each node inside the FOAM network provides geo-triangulations combined with a verified timestamp. Conversely, Platin uses a smartphone app to record and distribute location data acquired using different sensors, such as GPS, Bluetooth, WiFi. Finally, XYO operates on an ecosystem of devices and technologies to determine where physical objects are.

3 BLOCKCHAIN-BASED SMART CONTRACTS

Blockchain. Currently, several variants of blockchain exist, these variants are often classified as distributed public ledgers. In its original form [5] a *blockchain* is essentially a temporally ordered list of permanent data blocks. The head of the list is called *genesis block* and includes some evidence about its release date, while every other block is generated at fairly regular basis and it contains a cryptographic message digest, or briefly a *hash*, of its predecessor, creating a chain of references. Each block also includes a *proof of work*, namely an evidence that a certain amount of work has been spent for producing it. This proof is obtained by repeatedly applying a cryptographic hash function to a block, varying its content at each iteration by using a different nonce, until the cryptographic problem has been solved, namely when one of the desired hashes is found. These operations are part of the *mining* process that is simultaneously performed by several competitive network agents called *miners*. Altering a given block requires the recomputation of the hashes of all its successors in a limited amount of time. Since such operation could be quite expensive, the probability of observing a block replaced by another one decreases over time as new ones are added in front of it. A block referring to a given one is said to *confirm* it and after a certain number of confirmations, a block is considered practically immutable.

At the core of the blockchain technology there is a decentralized emergent consensus protocol that enables a group of agents to reach an agreement about a global state by accepting data transmitted across an open byzantine Peer-to-Peer (P2P) network. The consensus can be considered *emergent*, because there is not specific point in time in which it is explicitly reached. The network is said to be *open* and *byzantine*, because agents can be self-interested, they can enter and leave the system without authentication or secure connections and they can act strategically against the P2P protocol. Following the protocol, each agent can independently validate both transactions and blocks and reach a consensus about the blockchain state in an autonomous way. The Bitcoin protocol exploits this consensus mechanism to solve the double spending problem without the need for a trusted central authority [5]. In Bitcoin each block contains a list of transactions representing a transfer of tokens from source to destination addresses. The blockchain can be extended in several ways, a powerful generalization is captured by the notion of smart contract.

Smart Contracts. The original concept of *smart contract* can be traced back to the work of N. Szabo [7]. In recent years, this concept has been reintroduced, and in some way reinvented, by the blockchain technology. Platforms like Ethereum [2] provide not only a support for describing transactions about tokens, but they can run general-purpose scripts, commonly recognized as smart contracts. A smart contract encodes a set of public functions that are executed by the platform when a certain event occurs, for instance when a transaction is scheduled or a message received. A smart contract is deployed on the Ethereum platform through a special transaction, called *contract creation*, which returns a unique contract address that unambiguously identify it. A smart contract remains dormant until a transaction towards its address triggers its execution, either directly or indirectly as part of a chain of contract calls. When the function of a contract is triggered, the contained instructions can control the related token balance, its persistent internal storage and the invocation of other contracts.

Oracle. Blockchains are mainly designed to be self-contained, namely the execution of a transaction cannot depend on external data. The rationale behind this choice is to increase the predictability of the overall system and reduce the attack surface. If the state transaction logic depends on external connections, each node could derive a different global state and there would be no way for the network to achieve a decentralized consensus. Unfortunately, useful smart contracts often require useful data, like results of sport competitions, which should be used in a deterministic way. A common adopted solution is to delegate an external service for retrieving the data and storing them inside the blockckain for later use. This kind of service is called *oracle* and can be implemented in several ways depending on how the external data are collected and made available to other system components. More specifically, an oracle is responsible for periodically querying the environment to retrieve new data and submitting a transaction towards an *Oracle Smart Contract* (*OSC*) through which data are stored on-chain. An OSC is a particular kind of smart contract that usually contains a whitelist of recognized oracle addresses from which it can receive transactions that in turn can trigger one of its functions. The captured data can be stored on-chain in different ways depending on the chosen blockchain platform. A smart contract can directly access the stored data or subscribe itself to receive notifications. If data are not encrypted, external nodes can access them by reading the blockchain content without paying fees.

Oracles are a critical aspect of any advanced blockchain infrastructure, because their execution may be subject to external manipulation, particularly when they are implemented as centralized services. A better solution is represented by decentralized oracles composed of a network of independent data provides together with their own consensus mechanism.

4 BLOCKCHAIN-BASED LBSN CHECK-INS

In the previous section we have introduced the basic building blocks used to develop our solution for the fake check-in problem. In particular, we have discussed how smart contract can interact with the environment by means of oracles that store external data on-chain. Geographical positions together with time are probably the most important pieces of information to make smart contracts effective in the physical world. For instance, a smart contract may require that a performed activity is valid only if it is executed by an agent which is verifiably at a certain location, or that two or more agents can fulfill the contract only when they are nearby to each other. This paper considers the contract established between a LBSN service provider and its users which automatically pays the latter when they submit a review about a visited place.

The term *Proof of Location* (*PoL*) has been revived in the last years in conjunction with the blockchain technology. The goal of PoL is to reach a consensus on whether an object or agent is verifiably at a certain point in space and time. A *presence claim* is a digital certificate that endorses such localization. In principle, it can be built by means of oracles and used during a smart contract execution. For example, the FOAM project mentioned in Sect. 2 is able to provide a PoL mechanism for contracts that run inside the Ethereum platform. In the following, we assume the existence of an oracle implemented as a decentralized network of independent cells that cover the area of interest.

The proposed solution is exemplified in Fig. 1, it takes advantages of the blockchain technology for both certifying the location of users and automatically paying a reward for their work. The solution involves four main actors: the PoL Infrastructure (PLI), the Smart Contract Infrastructure (SCI), the LBSN Service Provider (LSP), and the LBSN Application Layer (LAL) implemented for instance as a library for a smartphone application. In our solution, we encode inside a User Smart Contract (USC) part of the license agreement between the service provider and the end-user. The USC will be responsible to automatically reward a user for her work as soon as a review is submitted for a place she actually visited.

The PLI is responsible for providing presence claims and storing them inside the SCI blockchain by means of an Oracle Smart Contract (OSC). The OSC is initially deployed by the PLI to offer its services and it is connected to an SCI account (0xXYZ) to collect the received payments. A presence claim request is performed by invoking a particular OSC function (step 2.6) which temporarily stores such request into a data structure, called here request queue. The PLI continuously monitors the request queue to check if someone needs a presence claim and this can be done by polling the OSC data structure at regular intervals without making transactions. When a new request has been received, the PLI produces the required presence claim that in turn will be stored inside the blockchain by calling a function of the OSC (step 2.7). Such function updates the request queue accordingly and permanently stores a localization event in the event log data structure. The event object acts as a presence claim encoding few essential properties such as the user identity (e.g., her SCI address), the localization details, and some error codes in case of failure.

The LSP initially deploys a unique Localization Smart Contract (LSC), then it will deploy an USC for each new registration. The user starts her interaction with the LSBN through a sing up activity (step 1.1) which includes the generation of a private key (step 1.2) and the associated public address (0x123) for the SCI (step 1.3). The private key is stored locally (step 1.4) and managed by the end-user, while the registration data and the public address are stored remotely under the control of the LSP (step 1.5). After the sign in (step 1.6), the user can access the LSBN services. In particular, when the system detect (steps 2.1 and 2.2) that she is near a recognized Point of Interest (POI), a notification is shown on her device together with a check-in suggestion (step 2.3). The user is encouraged to perform the check-in, because such action unlocks further LSBN functionalities, like finding nearby friends and in particular submitting reviews in the near future. Once the user agrees to perform the check-in (step 2.4), the LAL submits a transaction to the LSC (step 2.5) containing as payload the GPS location together with the user public address and the device identifier. Such identifier is used by the PLI to detect the device and produce the presence claim, while the user public address will be part of the presence claim stored in the event log.

A user can make a review for a POI (steps 3.1-3.3) through the LAL which is responsible for two things: sending the complete review to the service provider (step 3.5) and performing a transaction containing an hash of the review towards the USC (step 3.6). The actual review is not stored inside the blockchain for two main reasons: on one hand, the blockchain storage is expensive and cannot contain large media files; on the other hand, the LSP maintains the control over the uploaded files, so it can remove inappropriate content. Nevertheless, storing the review hash on-chain within the contract is enough to prove its origin and authenticity. The transaction performed towards the USC invokes the contract function responsible for paying the reward. In particular, such function checks the existence of a presence claim in the event log (step 3.7). In case such control completes successfully, a certain amount of tokens is automatically transferred (step 3.8) from the service provider account (0xABC) to the user account (0x123). The LSP can perform some off-line controls on the submitted reviews for evaluating their quality and eventually taking appropriate actions. For instance, it can adjust the amount of tokens to pay on the basis of the review quality, or it can close the contract in case of inappropriate contents (Destroy USC in Fig. 1).

A blockchain infrastructure does not run transactions and smart contracts for free, but it requires the payment of a certain amount of tokens for every operation. As a consequence, economic aspects are an integral part of the proposed solution. When the user joins the LBSN for the first time, she is rewarded with a small amount of tokens that covers the initial transaction costs. The user has to pay for both performing a check-in and submitting a review for a visited POI. The check-in transaction cost is not related to the cost of a presence claim that is paid by the LSC. A POI review matching the related presence claim is rewarded with an amount of tokens that covers the expenses and make a net profit for the end-user. Paying some transaction fees to both check-in and submit a review would discourage cheating behaviours, for instance asking a presence claim for a POI never visited. Overall, loosing tokens for checking-in should be a good incentive to make the related review.

In the proposed solution, the end-user has full control over her SCI account (0x123), hence she may withdraw the entire amount

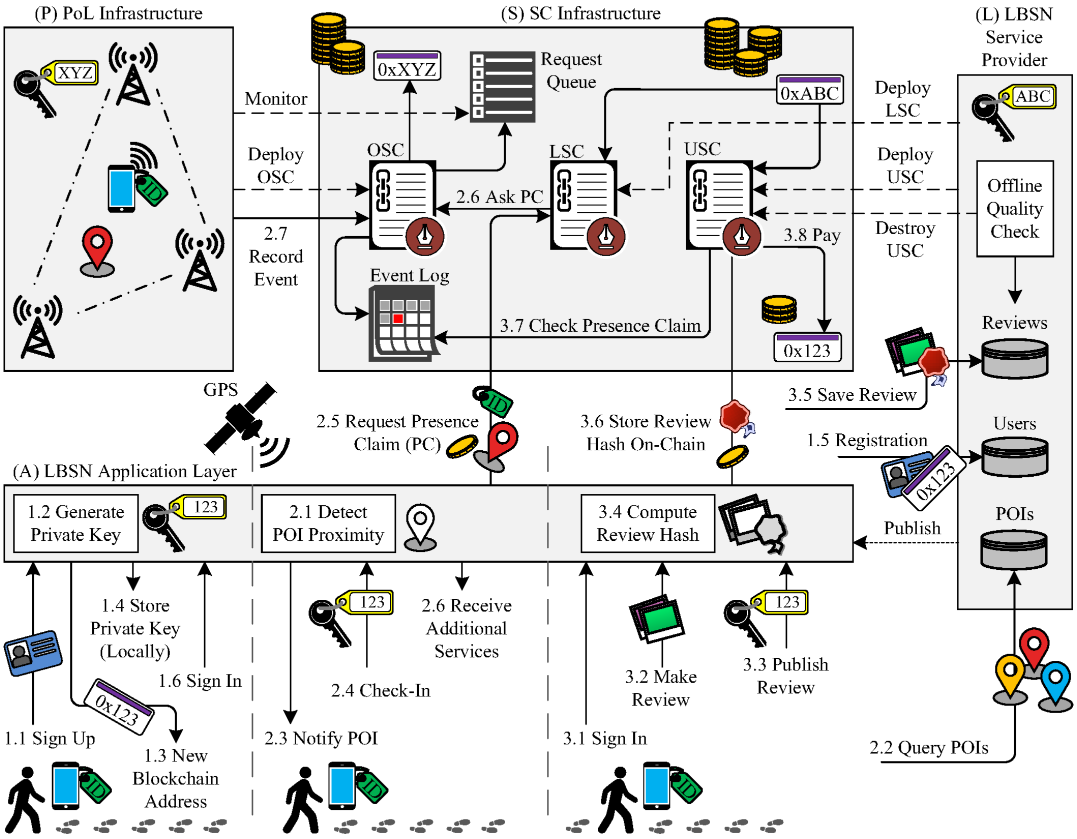

Figure 1: The solution proposed for the fake check-in problem in LBSNs.

of tokens if desired. This freedom can be limited by changing the USC logic and storing the funds in the contract address, instead of using an externally controlled account. This is the recommended approach if the registration process is not strong enough. Otherwise, an attacker can plunder all the tokens stored in the LSP account (0xABC) by automating both the creation of fake user accounts and the withdrawal of the related initial deposits.

5 CONCLUSION

This paper proposes a way to prevent fake check-ins in Location-Based Social Networks (LBNSs) through the use of a blockchain infrastructure. In particular, we investigate how smart contracts and a decentralized oracle system can be used to prevent cheating behaviors when users provide recommendations about visited places in change of an economic reward. At this regards, many solutions have been proposed in literature, but none of them can exclude false positives and they are generally loosely integrated with the reward mechanism. To the best of our knowledge, the proposed solution is the first attempt to address the fake check-in problem with a blockchain infrastructure. Many questions are still open, for instance our solution does not considered scalability issues about real-time guarantees. As an example, the processing of a presence claim can be indefinitely delayed by third-party transactions on the same blockchain. Nevertheless, the given architecture is open to several improvements and optimizations. For instance, some LBSN services can also be implemented in a decentralized way, increasing resilience and scalability of the overall system.

REFERENCES

[1] M. Al-Qurishi et al. 2017. Sybil Defense Techniques in Online Social Networks: A Survey. *IEEE Access* 5 (2017), 1200–1219.
[2] V. Buterin. 2014. A Next-generation Smart Contract and Decentralized Application Platform. http://github.com/ethereum/wiki/wiki/White-Paper.
[3] Foamspace Corp. 2018. FOAM Whitepaper. https://foam.space/publicAssets/FOAM_Whitepaper.pdf.
[4] H. Gao, J Tang, and H. Liu. 2012. Exploring Social-Historical Ties on Location-Based Social Networks. In *6th Int. AAAI Conference on Weblogs and Social Media (ICWSM)*. 114–121.
[5] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. http://www.bitcoin.org/bitcoin.pdf.
[6] K. Pelechrinis, P. Krishnamurthy, and K. Zhang. 2012. Gaming the Game: Honeypot Venues Against Cheaters in Location-based Social Networks. *CoRR* abs/1210.4517 (2012).
[7] N. Szabo. 1997. Formalizing and Securing Relationships on Public Networks. *First Monday* 2, 9 (1997).
[8] A. Trouw, M. Levin, and S. Scheper. 2018. The XY Oracle Network: The Proof-of-Origin Based Crypographic Location Network. https://docs.xyo.network/XYO-White-Paper.pdf.
[9] W. Wei, F. Xu, C. C. Tan, and Q. Li. 2013. SybilDefender: A Defense Mechanism for Sybil Attacks in Large Social Networks. *IEEE Transactions on Parallel and Distributed Systems* 24, 12 (2013), 2492–2502.
[10] L. Wolberger and A. Mason. 2019. Platin. Proof of Location Blockchain, White Paper. https://platin.io/assets/whitepaper/Platin_Whitepaper_v3.01.pdf.
[11] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. 2008. SybilGuard: Defending Against Sybil Attacks via Social Networks. *IEEE/ACM Trans. on Networking* 16, 3 (2008), 576–589.