

# Spatial Blockchain: Smart Contract Using Multiple Camera Censuses

Vasanth Iyer<sup>1(⊠)</sup>, Asif Mehmood<sup>2</sup>, Baniya Babu<sup>3</sup>, and Y.B Reddy<sup>1</sup>

- Grambling State University, Grambling, LA 71245, USA iyerv@gram.edu.com, ybreddy@gram.edu
- <sup>2</sup> Air Force Research Labs, Wright-Patterson AFB, Riverside 4543, USA asif.mehmood1.civ@mail.mil
- <sup>3</sup> Department of Computer Science and Information Systems (CS&IS), Bradley University, 1501 W. Bradley Avenue, Peoria, IL 61625, USA bbaniya@fsmail.bradley.edu

Abstract. With many data breaches and spoofing attacks on our networks, it becomes imperative to provide a reliable method for verifying the integrity of the source. Blockchain location-based proof-of-origin is explored for tracking trucks and vehicles. Blockchain applications that support quick authentication with these non-mutable ledger properties: consensus and implemented as smart contracts at the edge. This Blockchain application will now be known as the POWTracker platform, gathering data from multiple cameras. POWTracker is based on an existing GPS-based blockchain ledger and runs on an edge device that uses AI consensus and multiple cameras. By using GPS algorithms, we present a novel mining algorithm that rewards POW miners, providing a trustworthy, verifiable proof-of-location system.

**Keywords:** Spatial vehicle identification  $\cdot$  Smart contract  $\cdot$  Blockchain  $\cdot$  Re-identification algorithm  $\cdot$  AI for edge devices  $\cdot$  Trustworthy AI

#### 1 Introduction

A blockchain is an encrypted, distributed database in which all peers share information. Blockchain technology can be reliably used in spatial applications requiring proof of location due to its decentralization, immutability, and anonymity. Numerous consensus algorithms have been proposed, such as Proof-of-Work (PoW), Proof-of-Stake (PoS), Byzantine Fault-Tolerant (BFT), and Paxos. Because the spatial blockchain uses PoW, we will consider using the Raft algorithm since it is scalable. The spatial blockchain can indeed be a public network, but in the context of re-identification, all participants are known and have smart-contract obligations to act correctly, which is why we use Raft as our primary protocol. Raft is more efficient than other algorithms, and the implementation of the protocol is consistent as it is a distributed implementation.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023 K. Arai (Ed.): FTC 2022, LNNS 560, pp. 55–66, 2023.  $https://doi.org/10.1007/978-3-031-18458-1\_4$  Here are the specific goals of the Proof-Of-Work (POWTracker) application:

- Aim 1: Investigating how Proof-of-Work for spatial algorithms can be integrated into public blockchain: We will examine how linked data [24] can be used, since it is immutable.
- Aim 2: Exploring consensus-based answers for user GPS queries: We will
  examine how a model of queryable GPS can be implemented using cryptographic consensus.
- Aim 3: Investigating mining of the existing POWTracker blocks based on a GPS difficulty level.
- Aim 4: Exploring algorithms that can re-identify the same object using old and new positions and execute a smart contract for proof-of-location.

The Text Discusses the Importance of the Research Question: Current location-based applications rely on a centralized GPS system that is easy to spoof and lacks data integrity. The ability to query and identify using geolocation, and perhaps not only track but re-identify the package, eliminating duplicates at the other end and finer tamper-free details without paying a middleman. POWTracker will use Proof-of-Work consensus to create new truths in the blockchain Hash blocks, which are immutable. The current mobile and smartphone applications use GPS to log their location trajectories [23] into a central server, which are accurate but can be spoofed and manipulated. Geospatialaware [23] queryable protocol uses decentralized blockchain and allows miners to authenticate and verify location information by answering spatial queries. Point of Location (POL) data and background signatures can be included in the location data. There are both visual and non-visual components in the spatial data [4,23]. As soon as miners have completed the POW, a query is made to determine whether the object has reached its destination. The shipper is paid directly by the customer if the object is close to the customer because the smart contract can be programmed to verify spatial verification automatically without the help of a third party.

Research Gaps and how you will Address them: Ten years ago, Bitcoin [3] was the first blockchain application. Rather than using computationally intensive cryptographic algorithms, this design will explore domain adaptation (proof-origin) for vehicle position. As current vehicle logging algorithms are susceptible to tampering and inaccuracies, we train a domain model that is robust [9] to synthetic fake data identification [1,16] at the point-of-origin. Using blockchain logs, which cannot be altered, the re-identification algorithm uses domain-adapted verification.

# 2 Review of Spatial Blockchain

Blockchain has been incorporated into industry-leading supply chain networks in automotive, banking, healthcare, insurance, media entertainment, retail consumer goods, and travel and transportation. Identification, tickets, and boarding passes can cause a frustrating logistical bottleneck for passengers and carriers. By

providing the ability to create a single identification token (nounce) [26] that can be validated throughout the trip, blockchain systems could simplify boarding, reduce terminal congestion, and reduce the need for multiple travel documents. As shown in Fig. 1, blockchain can also be used for POL, when a supply chain network requires data provenance, as described in the introduction. In order to establish the provenance of each part, blockchain can consolidate details such as manufactures, production dates, and batches. Maintaining immutable records makes it possible to schedule maintenance, provide work order documentation, and prevent fraud. As Blockchain uses hash [25], the integrity of the data is protected from tampering. As illustrated in Fig. 2, we will hash the GPS coordinates consisting of longitude and latitude into a 256-bit unique number using the blockchain concept for POL. The blockchain is built using concepts like linked lists, hash functions, and Peer-to-Peer consensus networks. The next section will discuss each one in detail and how to use multiple cameras [20] for smart-contract consensus. The mining puzzle is adapted to the needs of edge smart contracts in our proposed transportation vehicle framework.



Fig. 1. A simple application of spatial proof-of-location.

# 3 Mining Bitcoin on a Blockchain

[3] is a mathematical Proof-of-Work that solves a complex puzzle to enable new bitcoins to be added to the blockchain ledger. A mining difficulty of [3] is determined by the network of miners competing to solve the hashing puzzle simultaneously using powerful GPUs, and the first miner to solve it wins bitcoins. This is accomplished by taking the current hash or the genesis block's hash and finding a smaller hash. Initially, miners create nonces (numbers used once), which are 32-bit numbers compared to the target hash. After the nonces are hashed, the 256-bit hash is produced, which is a guess because the space of possible hashes is so large. When the generated hash is less than the target hash, the miner is rewarded with bitcoins.

# 4 Methodology - Ledger

As part of this section, we will design the ledger [25] for a spatial blockchain that consists of data structures such as linked data. Through the implementation of a basic linked list [1], the building blocks of a supply chain hypothesis which is resistant to cyberattacks will be introduced, such as GPS and truck routes.

#### 4.1 Non-mutable Spatial Blockchain

In aim 1, we will develop the spatial version of blockchain using the non-immutable property [25]. Equations (1) and (2) shows how vehicle GPS and location be encoded into a blockchain ledger using a nonce. The  $new_{loc}$  contain an encoded hash that serves as the start chain hash for encoding the vehicle's next route as shown in Fig. 2.

$$new_{loc} = hash_{gps-loc} + Truck_{id} + Time_{stamp}$$
 (1)

$$= hash_{gps-loc} + hash_{nonce} \tag{2}$$

The truck route is encoded and its chain is described in Fig. 3, we build a proof-of-location between point A and point B by storing the GPS location provided by the truck during its entire route in an immutable blockchain. The immutable blockchain is created using a hash of each GPS location along the route. The immutable linked-data verifies the same truck's route, giving proof of the truck's location. The next block uses the previous block's hash to add the new GPS location, so that if one of the location values changes, the stored hashes will no longer match the truck's complete route log. An ID for each truck can be used to scale the design to multiple routes. The hash of the truck's GPS location may be the same for each route, but when combined with the ID and timestamp, the nuance is unique. Furthermore, the GPS locations are stored in the blockchain as 256-bit hashes which maintain privacy since blockchain ledgers are publicly visible by design. The blockchain consensus can be attacked internally if one blockchain peer is malicious, and the framework cannot validate actual values as it does not use spatial data from neighbors.



Fig. 2. An example of GPS meter accuracy hash which is unique.

#### 4.2 Spatial Blockchains that are Verifiable

The Eqs. (3) and (4) show a proof of concept for a blockchain [23] with mining incentives. The previous hashing design as Eq. (2) is very precise by using the original GPS. Currently, the proof-of-location [23] does not have a consensus because it relies on single GPS signals. Therefore, the truck can secretly collaborate with the agent and manipulate the location information through illegitimate means.

$$new_{loc} = hash_{Gnss-log} + Truck_{id} + Time_{stamp}$$
 (3)

$$= hash_{nonce} + hash_{submeter-loc} \tag{4}$$

Modern mobile GPS cannot be trusted as they may be susceptible to spoofing attacks, blockchain provides an additional level of security as shown in Eq. (4). For a queryable mobile GPS, we assume that the truck can actually query its location, rather than relying on GPS signals. Figure 4 shows a queryable mobile GPS implementation with two independent stations located near the truck's checkpoint, where it needs proof of location. The communication between the truck and the checkpoint must be encrypted. Our scalable scheme will use public keys. Assuming that all trucks have access to the public keys of the proof-oflocation, the truck can encrypt using  $key_1$  for tower 1 and  $key_2$  for tower 2. An encrypted message is sent to both towers using a single, one-time pad. The onetime pad is then encrypted with  $key_1$  and  $key_2$ . Two towers receive the truck's id along with the GnssLog [11] encrypted with one-time pad. One-time pads are decrypted by the towers using their private keys, which also decrypt messages sent between them. The two-way authentication to query the truck GPS prevents any spoofing suspected in the traditional GPS implementation. Figure 4 shows that the same blockchain ledger from the previous implementation also includes a consensus of POL. As shown in the next section, the enhanced design allows mining that is based on multiple camera consensus, so it is completely safe and hard to spoof easily.

Spatial Blockchain Mining. In the context of spatial blockchains, we propose a mining puzzle. Based on GPS accuracy, we model the complexity of the answer to be mined based on Bitcoin mining. The GPS used in the previous section is based only on the satellite, which is precise to a few meters. A miner will be awarded if the GPS of the current truck can be improved to sub-meter accuracy, as described in the derived term  $hash_{submeter-loc}$  Eq. (4). The accuracy of satellite-based GPS is subject to errors [6,8] due to reflected signals caused by obstacles and buildings, and delays between signals received by using multiple satellites. In addition to using powerful GPUs for Bitcoin mining, spatial miners use advanced Global Navigation Satellite System (GNSS), which can be cross-linked with other ground-based satellites to correct global satellite GPS errors. With Android smartphones, one can view raw GNSS logs [11], and one can estimate the global position and velocity with sub-millimeter accuracy as

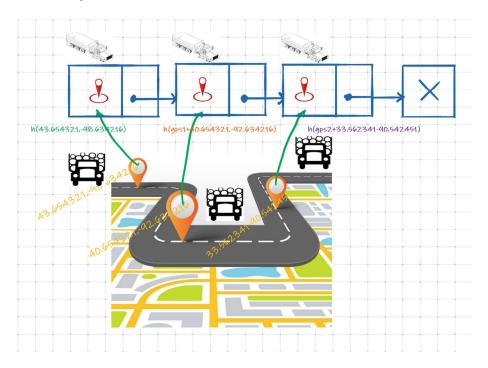


Fig. 3. Design of a non-mutable spatial blockchain.

calculated in Eq. (5). We propose a incentive mining algorithm that must perform the following steps [11] for a potential payoff to build the next chain in the ledger:

- 1: Read data from GnssLogger
- 2: Compute pseudoranges
- 3: Compute least square position and velocity
- 4: Calculate the difference (error)
- 5: Use Kalman filter to predict the sub-meter accuracy.

The simple least square is given by the equation:

$$r_u^k = ||x_{sat}^k - x_u|| \tag{5}$$

where  $r_u^k$ =actual distance between truck and the satellite

 $x_{sat}^k = \text{position of the Truck}$ 

 $x_u = \text{position of the truck}$ 

A miner's goal is to minimize the measured and estimated pseudorange for every satellite (assuming at least four are required). Inertial Measurement Unit (IMU) is a device that can measure acceleration and angular velocity, such as from a car. IMU measurements are absolute values. By comparing GNSS values with IMU absolute values, errors can be corrected using Eq. (5) to provide submeter accuracy. As the mining blocks are mined and rewarded if POL is

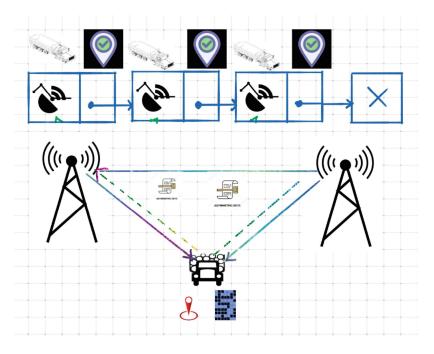


Fig. 4. Queryable GPS using Gnnss for spatial blockchain.

provided at sub-meter accuracy, the sub-meter positioning will be much closer to the truck's position than the initial GPS estimates.

# 5 Smart Contracts for Spatial Blockchains

Previously, we discussed a truck route from Fig. 3 that requires POL in order to ensure that the goods arrive at the final destination without further delays which are imported across borders using supply chain tracking. Our Proof-of-Work for aims 1 to 4 successfully utilized only GPS locations requiring interaction from the truck to the spatial blockchain network, causing some delays at each location along the route. [19] is a way of running Proof-of-Origin on an edge device [21] such as a smart camera. If the process were automated using smart contract, we may not be able to rely on the position alone. A smart contract that generates a response to a transaction can be integrated into an application such as spatial vehicle monitoring using multiple-cameras. Due to the popularity of deep learning and images of trucks from various cameras deployed along the routes, multiple-camera monitoring is becoming increasingly popular. Our deeplearning model will be trained with images from multiple cameras of the same truck which is in transit. As shown in Fig. 5, a typical multi-camera setup is also connected to a smart contract that can serve as an output of a deep learning model [2] to identify the current trucks based on previously seen cameras. We provide POL using visual features [18] by combining Blockchain smart contract and verificable GPS with trusted AI. We can see from the Eqs. (6), (7) and (8) that the smart contract's output is only triggered when features of  $cam_1$  and  $cam_2$  are the same as images of the earlier vehicle seen compared to  $cam_3$ .

$$new_{loc} = cam_{feature-1} + cam_{feature-2} \tag{6}$$

$$= cam_{image-3} \tag{7}$$

$$=$$
 trusted AI smart contract (8)

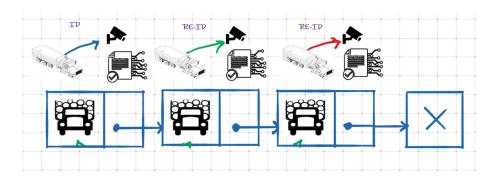
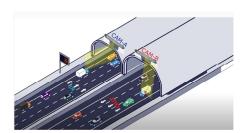
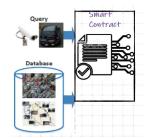


Fig. 5. Smart contract automation running at the edge device.

#### Algorithm for Re-identification 5.1



(a) Re-Identification of Vehicles Entering (b) Re-Identification Algorithm Deployed nel.



(CAM-A) and Leaving (CAM-B) a Tun- in a Edge-Device. Smart Contract Shows Multi-Camera Authentication.

Fig. 6. Vehicle re-identification using multi-camera system.

Instance Re-Identification [2,18] is a vital computer vision task, which targets remote monitoring using drones for wildlife protection, vehicle re-identification, etc. Smart contracts can be used with the blockchain ledger previously designed with GPS. To re-identify the truck at different locations, the smart contract uses a deep learning model [17], which has been trained domain adapted multiple cameras in the trunk's route. Figure 6(b) shows the deep learning framework [18]. Re-identification is an artificial intelligence technique that finds a particular vehicle, asset, or person from a large collection of videos. Like the expensive 3D-annotation task, the re-identification task consumes a lot of video-watching time to find the vehicle, or person. Two or more cameras recording the same vehicle or person will require re-identification algorithms. Using our example, if we have three cameras, then a query is generated for the current instance of the truck when proof-of-location is requested, similar to linked data in Fig. 5. The query image is submitted to the re-id trained model for visual authentication, as shown in Fig. 6(b). The query of the current instance of the truck includes the previous instances (like the blockchain ledger). Therefore, we can confirm that the current instance is part of the truck routes, enabling a consensus for POL. AI and blockchain are used to make networked consensus that is trustworthy and programmable through smart contracts that are difficult to guess.

# 6 Preliminary Results

The models for aims 1 to 4 have been designed, and the final smart contract has been deployed to an edge device [2,19] the code can be found at the GitHub repository. The results of the testing of aims 3 and 4 are shown below.

#### 6.1 Computing Spatial Blockchain Proof-of-Work

Using the truck's GNSS logs, we, as miners, develop a spatial POW difficulty based on GPS accuracy to a sub-meter level. We use the dataset provided by Google, which is formatted in GNSS [11] format. (Details are available here: https://github.com/google/gps-measurement-tools: GNSS log data are used: https://www.kaggle.com/google/android-smartphones-high-accuracy-datasets MATLAB code is used for baseline position correction. Table 1 shows a regular GPS reading with the Least Square method. We can get a more accurate GNSS reading by correcting the estimated satellite distance, position, and signal-to-noise errors with a more precise sub-meter GPS measurement as calculated in Eq. (5). There are a number of external variables such as ionospheric conditions, a challenging optimization problem, and the equivalent of finding a hash smaller than the target in traditional blockchain consensus [22].

Table 1. Results of linear regression to estimate satellite distances precisely.

Proof-of-work using GPS accuracy				
GPS (meter)	Gnss (sub-meter)		Algorithm	
(37.42357595 =	(37.42361372		Simple	
122.09413204)	122.0936959)		Square	Esti-
			mation	

### 6.2 Implementing Smart Contract Using Multi-camera Consensus

The re-id algorithm uses visual clues instead of GPS to compute POW as described in the earlier sections, combining aims 1 to 4 into a smart contract running on a edge device. We train a multi-camera dataset [10] for vehicles using a deep learning MOBILENetv2 [19] model using a cosine pair-wise loss function [12] to separate fine grained vehicle classes [13]. The results of the training loss [14] are shown in Fig. 7. The ROC accuracy achieves 80% with a very low false-positive rate. When a truck instance image is provided as illustrated in Fig. 6, the results have very high similarities for other cameras [15]. An example query using the pre-trained model [5] for the VERI-Wild dataset is shown for the current truck instance in Fig. 6(b). The results are ranked (in our case for three cameras), showing a 99% match with one false positive. As discussed before, a smart contract can be deployed to provide proof-of-location using visual clues [7] alone running on the edge at near real-time speeds eliminating trucking delays.

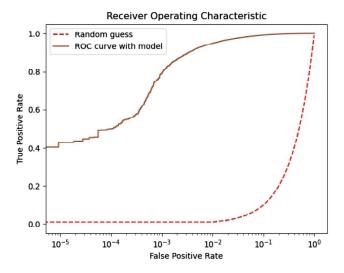


Fig. 7. Triplet training loss for veri-wild dataset

**Acknowledgments.** We would like to thank IBM and National Fintech Center, Morgan State for providing the blockchain discussions for this project. The NSF Award#2101181 funded security and privacy work aim 1, 2, and 3.

#### References

- Sheng, H., et al.: Near-online tracking with co-occurrence constraints in blockchainbased edge computing. IEEE Internet Things J. 8(4), 2193–2207 (2021). https:// doi.org/10.1109/JIOT.2020.3035415
- Iyer, V., Mehmood, A.: Multi-Object On-Line Tracking as n Ill-Posed Problem: Ensemble Deep Learning at the Edge for Spatial Re-Identification, Computing Conference, UK (2022)
- 3. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
- Iyer, V., Aved, A., Howlett, T.B., Carlo, J.T.: Autoencoder Versus Pre-trained CNN networks: Deep-features Applied to Accelerate Computationally Expensive Object Detection in Real-time Video Streams, SPIE (2018)
- 5. Iyer, V., et al.: Fast Multi-modal reuse: co-occurrence pre-trained deep learning models. In: SPIE 2019
- Iyer, V., Iyengar, S.S., Pissinou, N.: Ensemble Stream Model for Data-cleaning in Sensor Networks, AI Matters (2015)
- 7. Iyer, V., Mehmood, A.: Metadata learning of non-visual features: co-occurrence overlap function for rectangular regions and ground truth data. In: SPIE 2020
- Iyer, V.S.: Sachin, Virtual Sensor Tracking using Byzantine Fault Tolerance and Predictive outlier Model for Complex Tasks Recognition, SPIE Defense + Security (2015)
- 9. Richard, R., Brooks, S., Iyengar, S.: Robust distributed computing and sensing algorithm. Computer **29**(6), 53–60 (1996). https://doi.org/10.1109/2.507632
- Lou, Y., Bai, Y., Liu, J., Wang, S., Duan, L.: VERI-Wild: a large dataset and a new method for vehicle re-identification in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
- van Diggelen, F., Khider, M., Raw, A.: GNSS Measurement Datasets for Precise Positioning (2020)
- 12. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., Jin, R.: Softtriple loss: deep metric learning without triplet sampling (2020)
- 13. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13) (2013)
- 14. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks (2017)
- 15. Zulch, P., Distasio, M., Cushman, T., Wilson, B., Hart, B., Blasch, E.: Escape data collection for multi-modal data fusion research. In. IEEE Aerospace Conference 2019, pp. 1–10 (2019)
- Wojke, N., Bewley, A., Paulus, D.: Simple Online and Realtime Tracking with a Deep Association Metric (2017)
- 17. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks, CoRR abs/1604.01802
- 18. He, L., Liao, X., Liu, W., Liu, X., Cheng, P., Mei, T.: Fastreid: a pytorch toolbox for general instance re-identification, arXiv preprint arXiv:2006.02631
- 19. luxonis, OAK-D: Stereo camera with edge ai, stereo Camera with Edge AI capabilites from Luxonis and OpenCV (2020)

- 20. luxonis, DepthAI: Embedded machine learning and computer vision api, software available from luxonis.com (2020)
- Demidovskij, A., Tugaryov, A., Kashchikhin, A., Suvorov, A., Tarkan, Y., Mikhail, F., Yury, G.: OpenVINO deep learning workbench: towards analytical platform for neural networks inference optimization. J. Phys. Conf. Ser. 1828(1), 012012 (2021)
- 22. Khaled Salah, 1, (Senior Member, Ieee), Habib Ur Rehman, M., Nizamuddin, N., Ala Al-Fuqaha, Blockchain For Ai: Review And Open Research Challenges, IEEE Access (2018)
- 23. Kamel Boulos, M.N., Wilson, J.T., Clauson, K.A.: Geospatial blockchain: promises, challenges, and scenarios in health and healthcare. Int. J. Health Geographics (2018)
- 24. Linked List. https://en.wikipedia.org/wiki/Linked\_list
- 25. Hash Chain. https://en.wikipedia.org/wiki/Hash\_chain4.MerkleTree
- 26. SHA-256 Calculator. https://www.xorbin.com/tools/sha256-hash-calculator