Confidential, Decentralized Location-Based Data Services

Benjamin Adams

Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand

Abstract

There are many privacy risks when location data is collected and aggregated. We introduce the notion of using confidential smart contracts for building location-based decentralized applications that are privacy preserving. We describe a spatial library for smart contracts that run on Secret Network, a blockchain network that runs smart contracts in secure enclaves running in trusted execution environments. The library supports not only basic geometric operations but also cloaking and differential privacy mechanisms applied to spatial data stored in the contract.

2012 ACM Subject Classification Security and privacy → Human and societal aspects of security and privacy; Security and privacy → Economics of security and privacy

Keywords and phrases spatial data, privacy, smart contract, differential privacy

Digital Object Identifier 10.4230/LIPIcs.GIScience.2023.12

Category Short Paper

Supplementary Material *Software*: `https://github.com/darwinzer0/secret-data-tools`
archived at `swh:1:dir:aa0055da5ff292a15f6b44e5c54abf79fbfdab04`

1 Introduction

Mobile devices provide a number of opportunities to collect spatial data about human behavior, which can be used for data analytics and as training data for machine learning algorithms [10]. However, location data can also reveal extremely sensitive personal information and can easily be used in a harmful manner that violates an individual's privacy [3, 5]. Location data is not unique in this manner – centralized online platforms, such as large online social networks, have in recent years been critiqued for a wide-variety of ways that private user data is used and aggregated for commercial gain at the expense of user privacy. A recent trend toward building more decentralized applications that allow users greater control over their own data proposes to counteract this practice of gathering private data into centralized silos [8].

In this paper we present an approach to building decentralized location-based applications using confidential smart contracts that execute within hardware-encrypted environments. We demonstrate how we can use confidential smart contracts to implement spatial cloaking and to calculate summary statistics with global differential privacy on private location information without revealing individual information to others, such as a centralized server administrator. In addition, we explain how this model can allow for the creation of privacy-preserving location data marketplaces where data contributors can opt-in and control the amount of data they produce, control the level of spatial granularity, privacy thresholds, etc. on what data is made available, all while providing a mechanism for data producers to be directly paid for their contributions to the data set.

Figure 1 illustrates a sample scenario of computing over location data in a decentralized application that uses a confidential smart contract. Alice and Bob are users of mobile devices that allow them record their spatial location. They each execute a transaction on a confidential smart contract that stores their location (x, y) at a given timestamp on the chain

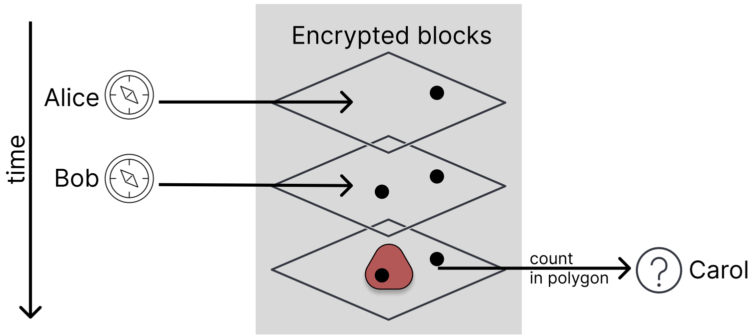

Figure 1 A simple scenario where private location points are stored on a blockchain and a third-party user makes a spatial query on the data without learning personal data.

in an encrypted format. They can query their own data points, but others, such as Carol, are only authorized to query for COUNT values within a given polygon. Permissions to access data are set by the logic of the contract and can be customized to suit any application. Furthermore, the permission system can be built to allow data owners to personalize settings, such as privacy thresholds or even charge for the use of their data before its acquisition.

2 Confidential smart contracts

Blockchains are essentially decentralized append-only databases [11]. They record transactional information as blocks of data that are generated by a peer-to-peer network of miners or validator nodes. The network operates using a consensus mechanism that ensures the history of the data (i.e., the chain) is immutable. In other words a bad actor cannot corrupt the chain and claim a transaction that did not occur and, for example, double spend some amount of currency recorded by the chain. The most well-known public blockchain, Bitcoin, uses a proof-of-work consensus mechanism which has an energy footprint that scales with the size of the network, but many other networks, including most that run smart contracts (described below), have long since shifted to proof-of-stake, which is far more energy efficient. By design all the data that is stored on a blockchain is visible, so it is a public database that allows one to examine every state of the chain, which means all transactions are by default public as well.

Smart contracts, first developed for the Ethereum network, are programs which can be run on blockchain networks [2]. Each node in the network runs a virtual machine that can execute code written in a turing-complete language. Ethereum uses the Solidity language running in the Ethereum Virtual Machine (EVM), while a number of newer chains run contracts that are written in general-purpose programming languages (e.g., Rust) and which are compiled to WebAssembly. Smart contracts are deterministic and modify the state of the data on the chain based on cryptographically-signed input messages that are sent to the network by client applications (often called decentralized apps or dapps). Smart contracts are referred to as trustless applications, because the logic of the contract is fixed and the consensus mechanism of the blockchain network ensures that messages sent to the chain will be interpreted in a fixed way based on the logic built into the program. This provides the ability to develop automated programs that allow users to conditionally transact on information without requiring a trusted third-party to verify that conditions have been met. Smart contracts have vastly increased the utility of blockchains leading to a wide-variety of applications in decentralized finance, digital art, and supply chain management.

Despite these many new kinds of dapps, the fact that all data and transactions that occur on most blockchains is completely transparent means that they are unsuitable for applications that require keeping information confidential, e.g. for user privacy. Some privacy blockchains utilize non-interactive zero knowledge proofs to provide transactional privacy built in, where proof of transactions from one account to another is recorded on the chain without revealing the amount of the transaction or which accounts were included [9]. However, the utility of zero knowledge proofs is limited to situations where there are only two parties involved. In other words, they are not capable of answering questions about data points that are individually private to a large number of different parties (e.g., calculating aggregate statistics).

However, there are a few blockchain networks actively developing more general-purpose confidential smart contract frameworks, where the internal state of smart contract execution as well as the data on chain is encrypted [14]. With confidential smart contracts you can create secure systems involving multiple clients that protect individually-supplied information while also computing over that information to provide outputs that are usable by other parties. Among three proposed methods: homomorphic encryption, secure multi-party computation (MPC), and trusted execution environments (TEEs), only the last (TEEs) is implemented in a working public blockchain. Fully homomorphic encryption is simply too slow to be a practical solution in current blockchain networks and secure MPC has also not been successfully implemented in a live network (although research remains active). TEEs however have been successfully integrated into live public blockchains since 2020 (with Secret Network[1] and more recently the Oasis Network[2]).

TEE-based smart contracts rely on using specific hardware chips where code executions can occur within a protected encrypted enclave [7]. For example, all the nodes in Secret Network are running on a set of Intel SGX chipsets. This means that the trust in the encryption of the network is based on trust that the hardware is secure. The upshot is that even the people who are running the computer nodes in the network cannot inspect the program state or data being used while the smart contract is executing, and all data written to the blocks are encrypted. In this paper we explore implementations of spatial algorithms using smart contracts running on Secret Network, at the moment the most mature network for developing confidential smart contracts. On Secret Network, contracts are written in Rust and compiled to WebAssembly before being uploaded to the chain. To date, most applications running on Secret Network are in the area of decentralized finance – this paper presents the first exploration into developing privacy-preserving location-based applications using the network.

3 Storing spatial data on chain

Developing on blockchain networks has limitations not found in normal programming environments. An important difference is that smart contracts cannot use floating point mathematical operations because they are not deterministic (different platforms implementing the IEEE 754 standard can output different results based on rounding), therefore there is a risk that different nodes in a network will be unable to come to consensus on floating point data written to chain. As a result to build a contract for operating on spatial data, e.g. points, lines, and polygons with x and y coordinates in a projected coordinate system,

[1] https://scrt.network/
[2] https://oasisprotocol.org/

there are two options: 1) store locations on an integer grid using 128 bit or 256 bit integer values for the x and y coordinates, or 2) using a fixed-point representation where numbers are represented as rational numbers using software operations on integer data structures. In the first case, operations will be fast and with 128-bit and greater sized integers meaning that units can be expressed in micro-meters or smaller, which for all intents and purposes allows the same level of geographic precision as any floating point representation. However, the kinds of operations that can be executed will be limited. In the second case, we can utilize fixed-point math libraries with functions such as *sqrt* and transcendental functions, including *exp*, *sin*, *cos*, etc.

Despite not being able to do some mathematical operations, many common spatial queries are still possible on data represented on an integer grid. For example, point in polygon, line intersection, convex hull, range searching, and nearest neighbor are all possible. This is due that fact you can calculate length squared as the dot product of a vector with itself and calculate signed area for two vectors and infer turn relationships based on the sign. Some operations however do require a fixed-point representation, such as great circle distance or applying differential privacy techniques to return fuzzy statistics on the data set stored in the contract.

The other main limitation for storing data on the chain is that doing so can be expensive. Smart contracts meter the amount of computation and data written and read from the chain and charge a fee (i.e., a gas requirement) to manage the computational load on the network. Read-only queries are free, but anything that writes data to a new block on the chain will cost something as a factor of these. Paying a small upfront gas fee for storing personal location data on the chain might be acceptable to some users if it means they maintain control over it and in fact if they are able to directly commoditize their own data while also having access to location-based services.

4 Spatial library for confidential smart contracts

We have developed a `secret-data-tools` spatial package for creating spatial applications on Secret Network. Code is written in Rust and available on Github (see Supplementary Material). For both integer-grid and fixed-point the library provides a set of geometric primitive structs written in Rust for Points, LineSegments, and Polygons. It also includes a set of basic geometric query operations, such as point in polygon, that work for both integer and fixed-point representations.

Spatial cloaking is a method for masking location data points into a wider geographic region (or some minimum size) and in a manner that maintains a certain level of k-anonymity [6, 12]. Implementing spatial cloaking of data with the library functions is rather trivial. A contract can mask data into grid cells or other regions when producing answers to queries. Furthermore, because authorization of data access can be customized to any use case, users can e.g., choose to provide more granular data to specific individuals, categories of individuals, or applications.

For point data stored in fixed-point representation, the library provides an implementation of global ϵ-differential privacy using the Laplace mechanism [4]. Differential privacy adds noise to the result of a function, e.g. COUNT or AVERAGE, such that the result satisfies the constraint set by the privacy budget parameter, ϵ. Composed with the basic geometry operations, the library provides the capability to perform queries such as returning a fuzzy count of the number of spatial observations that fall within a polygon boundary, without revealing any information about individual data points.

Services that use differential privacy on location data, e.g. collected by mobile phone apps, will often utilize *local* differential privacy. With local differential privacy, noise is added to each individual data point prior to collection in a centralized database. This helps to maintain individual privacy, however, because noise is added to each observation, rather than only the final result, the overall accuracy of the data is degraded more quickly. However, using confidential smart contracts, because the data values are only visible to the contract itself when running in the protected enclave and no outside observer can view them, we can store direct observation values and implement global differential privacy without needing to trust a central data administrator.

A characteristic of differential privacy is that each query made on the database erodes the privacy budget. This happens because multiple queries on the same data can reveal the true value eventually, so for data stored on a blockchain we need to add additional limitations on the number of times that a query can be performed. Therefore, all differential privacy queries are implemented not as read-only queries but rather as read-write transactions that not only provide the answer but also update the remaining privacy budget on chain. The query will fail if not enough privacy budget remains for the query.

5 Toward building decentralized location-based applications

The `secret-data-tools` spatial package is a toolkit for building decentralized location-based applications on Secret Network. Using this library enables a number of different possibilities for data sharing and services. Allowing data contributors to set their own thresholds for acceptable data sharing can lead to fine-grained control over location data. Data sources need not be from individual, personal devices either. Other location-based data, such as from object tracking or transportation nodes, might require confidentiality for business processes.

If a user wishes to contribute their location-based information then they will have to pay to put the data on the chain. The amount paid depends on how much data and the parameters of the network – a small amount of data (e.g., an individual point observation) will cost a fraction of a penny, but larger amounts of data will quickly add up. Contracts can operate data marketplaces that require payment from data readers before releasing data, which can be directly paid to data contributors without the need for an intermediary. Furthermore, the privacy parameters of spatial queries (e.g., the size of masking regions or privacy budget) can be made to be user-settable.

A larger, practical concern is that providing a direct incentive for data sharing will also likely incentivize people to upload false information, given that GPS data can be easily spoofed. The use of confidential smart contracts for spatial data analysis is particularly well-suited to be paired with *proof-of-location* systems [1, 13]. New proof-of-location technologies in development, such as FOAM[3] which uses networked LoRA devices to record location-based events, by necessity will require privacy-preserving mechanisms built-in prior to wide adoption. Currently they do not have that capacity, however. Confidential smart contracts provide one possible solution to incorporating privacy in proof-of-location systems, while at the same time proof-of-location can ensure fair decentralized marketplaces for spatial data.

Although we have primarily focused on examples of uploading individual data points, location-based data need not be stored as individual observations. Various methods of rolling up data are possible, which can be more efficient and result in lower gas charges for data contributors. In addition, there is the option of storing encrypted location data off the chain,

[3] `https://foam.space/`

and storing only the decryption key on chain. In such a model there would be very little cost to the user, however many of the advantages of trustless computation on individual spatial data points from multiple contributors will be lost.

6 Conclusion

This paper presented a new approach for building decentralized programs that allow users to privately share location-based data using confidential smart contracts. We introduced an open-source library for Secret Network-based smart contracts, which includes basic geometry operations and can support spatial data cloaking and differentially private queries. There is more research that is required to fully evaluate efficacy of such tools to support different types of privacy-preserving location-based applications and data sharing platforms. In addition, a security analysis of potential side-channel attacks both in terms of the underlying blockchain technology, as well as based on inference from other data is warranted.

References

1 Michele Amoretti, Giacomo Brambilla, Francesco Medioli, and Francesco Zanichelli. Blockchain-based proof of location. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 146–153. IEEE, 2018.

2 Vitalik Buterin. A next-generation smart contract and decentralized application platform. Technical report, Ethereum Foundation, 2014.

3 Matt Duckham and Lars Kulik. Location privacy and location-aware computing. In *Dynamic and mobile GIS*, pages 63–80. CRC press, 2006.

4 Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

5 Hongbo Jiang, Jie Li, Ping Zhao, Fanzi Zeng, Zhu Xiao, and Arun Iyengar. Location privacy-preserving mechanisms in location-based services: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(1):1–36, 2021.

6 Mei-Po Kwan, Irene Casas, and Ben Schmitz. Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(2):15–28, 2004.

7 Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. Innovative instructions and software model for isolated execution. *Hasp@ isca*, 10(1), 2013.

8 Christian Meurisch, Bekir Bayrak, and Max Mühlhäuser. Privacy-preserving AI services through data decentralization. In *Proceedings of The Web Conference 2020*, pages 190–200, 2020.

9 Xiaoqiang Sun, F Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205, 2021.

10 Eran Toch, Boaz Lerner, Eyal Ben-Zion, and Irad Ben-Gal. Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, 58:501–523, 2019.

11 Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. Technical report, National Institute of Standards and Technology, 2018.

12 Chengyang Zhang and Yan Huang. Cloaking locations for anonymous location based services: a hybrid approach. *GeoInformatica*, 13(2):159–182, 2009.

13 Pengxiang Zhao, Jesus Rodrigo Cedeno Jimenez, Maria Antonia Brovelli, and Ali Mansourian. Towards geospatial blockchain: A review of research on blockchain technology applied to geospatial data. *AGILE: GIScience Series*, 3:71, 2022.

14 Guy Zyskind, Oz Nathan, and Alex Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE, 2015.