Computers & Security 88 (2020) 101653

Contents lists available at ScienceDirect

Computers & Security

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

PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities

Imran Makhdoom[a],[*], Ian Zhou[a], Mehran Abolhasan[a], Justin Lipman[b], Wei Ni[c]

[a] University of Technology, Sydney, Australia
[b] Food Agility CRC Ltd, University of Technology Sydney, Australia
[c] Data61-CSIRO, Australia

ARTICLE INFO

Article history:
Received 6 May 2019
Revised 14 October 2019
Accepted 18 October 2019
Available online 22 October 2019

Keywords:
Internet of Things
Smart city
Security and privacy
Blockchain
EU GDPR Compliance

ABSTRACT

The ubiquitous use of Internet of Things (IoT) ranges from industrial control systems to e-Health, e-commerce, smart cities, agriculture, supply chain management, smart cars, cyber-physical systems and a lot more. However, the data collected and processed by IoT systems especially the ones with centralized control are vulnerable to availability, integrity, and privacy threats. Hence, we present "PrivySharing," a blockchain-based innovative framework for privacy-preserving and secure IoT data sharing in a smart city environment. The proposed scheme is distinct from existing strategies on many aspects. The data privacy is preserved by dividing the blockchain network into various channels, where every channel comprises a finite number of authorized organizations and processes a specific type of data such as health, smart car, smart energy or financial details. Moreover, access to users' data within a channel is controlled by embedding access control rules in the smart contracts. In addition, data within a channel is further isolated and secured by using private data collection and encryption respectively. Likewise, the REST API that enables clients to interact with the blockchain network has dual security in the form of an API Key and OAuth 2.0. The proposed solution conforms to some of the significant requirements outlined in the European Union General Data Protection Regulation. We also present a system of reward in the form of a digital token named "PrivyCoin" for users sharing their data with stakeholders/third parties. Lastly, the experimental outcomes advocate that a multi-channel blockchain scales well as compared to a single-channel blockchain system.

1. Introduction

There has been an exponential growth in the IoT-based services in the world, especially in telehealth, manufacturing and in urban areas to form smart cities. IoT is expected to connect 30 billion devices by 2020 (Lund et al., 2014). Use of IoT technologies will not only improve the quality of life of people but also contribute to the world economy. IoT is predicted to create about USD 7.1 trillion contributions to the global economy by 2020 (Lund et al., 2014). Concurrently, it is also estimated that by 2030 the global urban population will reach 5 billion. This rapid urbanization demands effective, and optimum use of city resources as well as smart governance and efficient service delivery (Moustaka et al., 2018; Zhang et al., 2017). It is believed that the solution to the rapid urbanization problems lies in creating a smart city that utilizes IoT technologies to monitor the physical world in real-time and provide intelligent services. These services may include eToll, smart parking (Zhang et al., 2017), smart health (remote patient monitoring, health emergency response), and police assistance (for law and order situations, e.g., riots, crime, or security breaches) (Moustaka et al., 2018).

However, at the same time, IoT devices are vulnerable to a vast number of security and privacy attacks (Makhdoom et al., 2019). Although, these threats are known to the manufacturers, unfortunately security in IoT devices is either neglected (due to cost or lack of expertise) or treated as an afterthought (Wurm et al., 2016). Similarly, a smart city network also suffers from numerous security and privacy issues (Bartoli et al., 2011; Moustaka et al., 2018), such as threats to privacy, integrity, and availability of user data, false data injection (Zhang et al., 2017), vulnerability to Sybil Attack (Cui et al., 2018), and single point of failure due to centralized control.

[*] Corresponding author.
E-mail addresses: imran.makhdoom@uts.edu.au (I. Makhdoom), ian.zhou@student.uts.edu.au (I. Zhou), mehran.abolhasan@uts.edu.au (M. Abolhasan), Justin.Lipman@uts.edu.au (J. Lipman), Wei.Ni@data61.csiro.au (W. Ni).

https://doi.org/10.1016/j.cose.2019.101653
0167-4048/© 2019 Elsevier Ltd. All rights reserved.

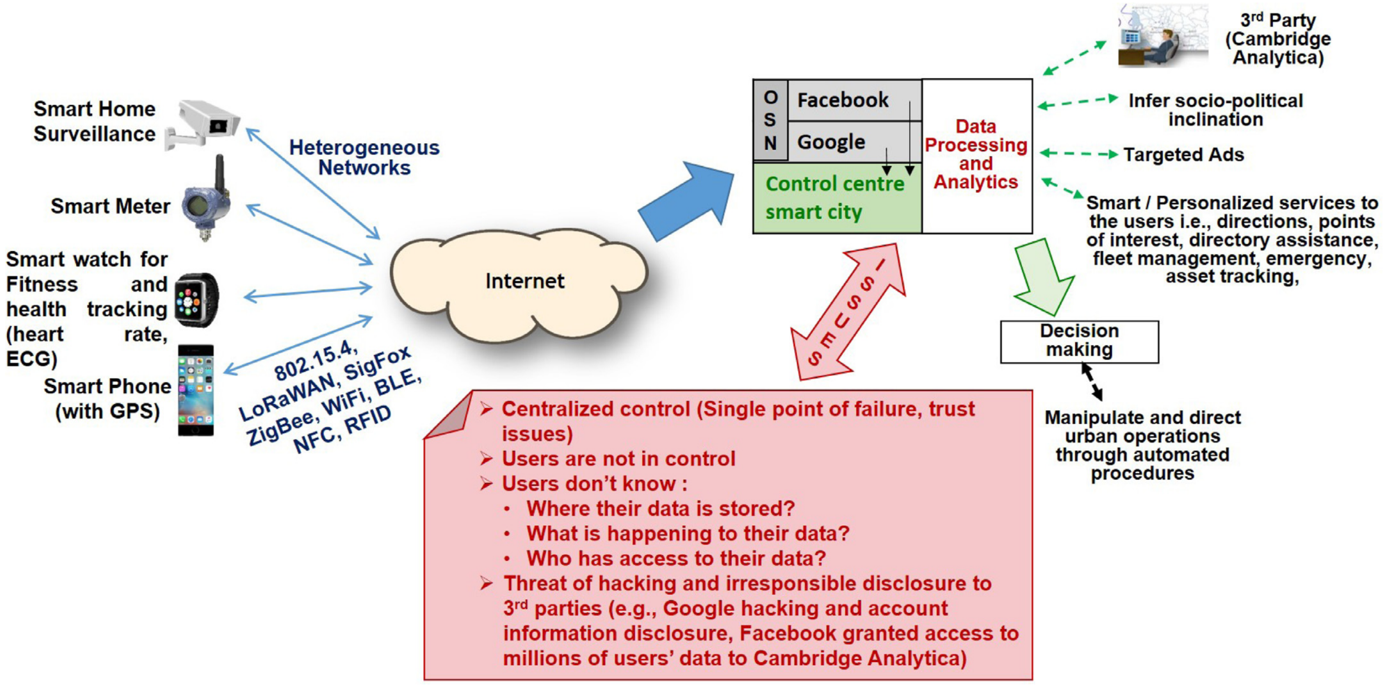

Smart Home Surveillance
Smart Meter
Smart watch for Fitness and health tracking (heart rate, ECG)
Smart Phone (with GPS)
Heterogeneous Networks
802.15.4, LoRaWAN, SigFox ZigBee, WiFi, BLE, NFC, RFID
Internet
OSN
Facebook
Google
Control centre smart city
Data Processing and Analytics
3rd Party (Cambridge Analytica)
Infer socio-political inclination
Targeted Ads
Smart / Personalized services to the users i.e., directions, points of interest, directory assistance, fleet management, emergency, asset tracking,
Decision making
Manipulate and direct urban operations through automated procedures
ISSUES
➢ Centralized control (Single point of failure, trust issues)
➢ Users are not in control
➢ Users don't know :
• Where their data is stored?
• What is happening to their data?
• Who has access to their data?
➢ Threat of hacking and irresponsible disclosure to 3rd parties (e.g., Google hacking and account information disclosure, Facebook granted access to millions of users' data to Cambridge Analytica)

Fig. 1. Issues in the smart city environment.

If we look at Fig. 1, the user data collected by numerous sensors is stored and processed by various OSN (Online Social Networks), smart city control center or various other smart city components such as Intelligent Transportation Systems (ITS), health emergency response, fire and rescue, etc., These components (with mostly centralized control) process user data for the provision of various services to the users and third parties. Although such a centralized control may look effective from the outside, yet it has some significant security concerns.

Centralized control is subject to a single point of failure in case of a cyber-attack or other technical malfunctions (Puthal et al., 2016). Moreover, it also has trust issues, as the users have to put their trust in the entity that is handling their data. Hence, users have no control over their data assets. Further concerns for user data include: Users do not know where their data is stored and what is happening to it. Who has access to it, and is there any unauthorized disclosure to the third parties. The above-mentioned users' concerns are very much real as the disclosure of personal data leakage concerning millions of users by Facebook Inc. (Jason, 2019; Sara and Michael, 2018) and a bug in Google Plus (Sara, 2018) that resulted in the exposure of personal information of approx 500,000 users is a candid example of one of cloud/OSN vulnerabilities.

Moreover, any smart city application is believed to store, process and analyze users' data. Hence, every security solution developed for a smart city environment must comply with the under-mentioned key requirements of European Union General Data Protection Regulation (EU GDPR) (GDPR, 2019) while handling users' data:

- Personal data should be processed only with the consent of the data owner.
- Any technology dependent on user data must preserve user privacy by design.
- The gathering, processing or use of personal data should be in accordance with the instructions based on a mutual contract between the user and the third parties.
- The owner of data has the right to access the information concerning the processing of his data, i.e., which third parties have access to what data and how they use it.
- It is the right of the data owners that their data be erased immediately once it is no longer needed.
- The system should be transparent such that individuals know about the collection and use of their data.

As far as IoT security is concerned, researchers and security analysts are trying to leverage cryptographic security benefits of blockchain to resolve security and privacy issues of IoT. Hence, we believe that a carefully selected blockchain technology with an insightful business network design can resolve most of the data integrity and privacy issues of a smart city.

1.1. Related work

Security researchers around the world are developing and investigating ingenious ways to implement blockchain in the IoT environment. These use cases aim to take advantage of the inherent benefits of the blockchain such as decentralized control, immutability, cryptographic security, fault tolerance, and capability to run smart contracts. Recently, researchers (Michelin et al., 2018) presented a blockchain-based data sharing framework for a smart city environment. The framework called "SpeedyChain" focuses on reducing the TX settlement time for real-time applications such as smart cars and also aims to ensure user privacy. Moreover, it ensures data integrity, tamper-resistance, and non-repudiation that are some of the intrinsic benefits of the blockchain. In another work, Pradip Kumar and Jong Hyuk proposed a Software Defined Networking (SDN) and blockchain-based hybrid network architecture for a smart city (Sharma and Park, 2018). The proposed architecture addresses usual smart city issues including high TX latency, security and privacy, bandwidth bottlenecks, and requirement of high computational resources. In the proposed model the smart city network is divided into a distributed core network comprising resourceful miner nodes and the centralized edge network constituting inept devices. The edge nodes store access policies for locally registered nodes. Authors claim that in addition to reducing

TX latency, and reduced network bandwidth, the proposed model limits the effects of a node compromise to the local area.

Additionally, authors in (Rahman et al., 2019) proposed a smart contract based sharing economy services in a smart city. The proposed model uses Artificial Intelligence (AI) for data analytics and also uses blockchain to store the results. Similarly (Biswas and Muthukkumarasamy, 2016) presented an overview of a blockchain-based security framework for secure communication between smart city entities. Authors claim that the integration of the blockchain with devices in the smart city will provide a shared platform where all the devices would be able to communicate securely. However, the researchers did not disclose some necessary details about the type of blockchain platform, consensus protocol, and TX/block validation techniques adopted in the smart city application.

In another endeavor (Haidar et al., 2017; Kountché et al., 2017), security researchers have proposed solutions to address various user privacy issues in ITS. Nonetheless, they do not cater to the challenges of smart cities such as trustless data sharing among multiple organizations. Similarly, Ali Dorri and Raja Jurdak proposed a secure, private and lightweight architecture of a blockchain-based smart home application (Dorri et al., 2017; 2016). It aims to solve certain blockchain issues such as computational intensiveness, latency in TX confirmation and energy consumption. To reduce computational overhead, and energy consumption each block is mined without any Proof of Work (PoW). Moreover, the latency in TX confirmation is reduced by considering a TX true, whether it is mined in a block or not. In addition, the proposed scheme utilizes cloud storage to ease up the memory requirements for smart home devices. However, there are many security concerns that need further explanation with logical reasoning (Makhdoom et al., 2018a). Likewise, another team of researchers proposed an Ethereum Blockchain (Buterin et al., 2014) based mechanism to manage IoT devices (Huh et al., 2017). Nonetheless, Ethereum Blockchain does not provide data privacy.

In another work, to avoid issues concerning the single point of failure in a centralized system, researchers proposed an Ethereum Blockchain based decentralized, self-managing Vehicular Ad-Hoc Network (VANET) with a challenge-response based authentication (Leiding et al., 2016). However, the proposed scheme does not explain the procedure of consensus and block mining. There is also no discussion about the type of information to be published on the blockchain and the latency in TX confirmation. Above all, Ethereum Blockchain does not provide data privacy and confidentiality.

Correspondingly, Yu Zhang and Jiangtao Wen proposed an Ethereum Blockchain based decentralized electronic business model for the IoT (Zhang and Wen, 2016). However, the proposed solution mostly focused on the working of the e-business model, so there is a lack of discussion on technical aspects such as block mining mechanism, modalities of implementing blockchain on IoT devices, and the methodology of achieving data confidentiality and privacy. Similarly, in another work (Krishnan et al., 2018) authors introduced a blockchain-based security framework for IoT implementations. Nonetheless, the proposed solution focuses on data authentication and secure communication between the sensor devices and the controllers. The researchers make use of the received signal strength (RSSI) of the message sent by a sensor device as a parameter to ensure the randomness of data to avoid replay and data forging attacks by a MITM (Man-in-the-Middle) attacker. Few other researchers have also proposed a blockchain-based approach of exchanging data in the smart city between nontrusted organizations (Qian et al., 2018). In this regard, if a third party queries some data, e.g. a credit report concerning a user, then the executor node gets the input in the form of private data from the respective organization through a local private API. The data is encrypted with an organization's private key and is decrypted once in the executor sandbox using the organization's public key. Hence, the querying party receives only the processed data and does not see the original data itself.

Since the GDPR legislation came in to effect on 25th May 2018, researchers have been working on various aspects of data protection to develop GDPR compliant data protection/processing frameworks. In this endeavor (Truong et al., 2019) proposed a blockchain-based design concept for developing GDPR compliant data management platforms. The solicited framework shares, and revoke the sharing of user data only with the consent of data owner. Moreover, the blockchain-based framework can also endorse the service providers for being correctly following the GDPR policies or not. As per the devised concept, only data owners and data controllers can create, update, and withdraw consent, and only authorized entities can process user data. The proposed mechanism uses blockchain to handle authentication, authorization, and data access control token validation. Whereas, the data is stored in a centralized resource server that is assumed to be a trusted party. Apart from the resource server being a trusted party, the proposed solution does provide some security guarantees; however, it seems to have high communication complexity. As authors also claim that due to increased message overhead, the proposed scheme does not support high performance and scalability since the TX latency significantly increases and throughput decreases with the increase in the number of nodes. Similarly (Faber et al., (2019) recommended a conceptual architecture of a human-centric and GDPR compliant blockchain-based personal data and identity management system (BPDIMS). The authors focus on designing a framework, which is transparent and provides data owners the full control over the usage of their data. The researchers address specific issues concerning data usage, i.e., user consent, transparency of data processing, purging of user consent, reward mechanism for users, data integrity, and confidentiality. However, this work is still at conceptual stages and does not present any technical details or performance evaluation.

Similarly, Ricardo et al. proffered a blockchain-based scheme to facilitate data accountability and provenance tracking (Neisse et al., 2017). Data provenance tracking is achieved by maintaining a list of references to the data provided to the controller. The list is updated whenever some data is sent to the data controller/service provider. Whereas, data accountability is accomplished by specifying restrictions on data usage in smart contracts. The restrictions are defined under the domain of a preventive mechanism, using a security policy language recommended by SecKit (Model-based Security Toolkit). The preventive mechanism denies actions such as allow, deny, modify, or delay the operations concerning data usage to the data controllers. The authors primarily discuss various design choices for the data usage contract models while considering the provision of maximum data provenance information to the data owners in a trusted and privacy friendly-manner. The sample contract models are evaluated based on gas consumption in Ethereum Virtual Machine (EVM). Correspondingly, authors in (Rantos et al., 2018) introduced a consent management platform named ADvoCATE for IoT data processing. ADvoCATE uses Ethereum Blockchain to preserve the integrity of users' consents and related updated versions. The ADvoCATE may be interpreted as a cloud service platform with various components such as blockchain, intelligent policy analyzer, consent notary, and storage. The consents notary ensures that the created consents are up to date and are also protected against unauthorized modification. Whereas, the intelligence component makes use of Fuzzy Cognitive Maps (FCM) methodology to identify any rules/policies that contradict with GDPR requirements concerning the handling of users' data. Moreover, whenever an IoT device is installed, the user gives his consent to the data controller/service provider through a smart contract to access IoT device data. The digital consents duly signed by the data controller and the device owner are stored on the ADvoCATE platform, whereas, the blockchain stores only the hashes of these consents for integrity. However, the proposed platform is still in the development phase and has not been extensively tested or evaluated. The authors only highlight the gas (ether) consumption of smart contracts, and there is no analysis on TX latency, TX throughput, scalability, or communications overhead.

In a similar endeavor (Kaaniche and Laurent, 2017), presented a blockchain-based data usage auditing architecture that provides the data controllers with unforgeable evidence of users' consent. The researchers claim to provide user anonymity by letting the data owners (which are delegated PKG) create a distinctive public-private key pair for each smart contract they initiate to share data with a service provider or a data processor. Moreover, the authors used hierarchical ID-based encryption to prevent unauthorized disclosure. The data is stored on off-blockchain storage, whereas blockchain smart contracts are used to store the hash of data and data usage policy. Also, there is a specific smart contract between the data owner and every other service provider or data processor. However, the architecture is not supported by any performance evaluation, e.g., TX settlement time, block commit time, or latency. In another work, authors evaluated the potential use of blockchain technology to facilitate the transformation of institution-centric exchange of data to patient-centric, and patient-driven data sharing (Gordon and Catalini, 2018). The researchers recommend that the blockchain can be used to provide transparency over the state of shared data, and related TXs among different stakeholders. In that permissioned blockchains can be more productive in terms of delivering strict access control concerning read-write permissions over users' health data. Authors also believe that the blockchain provides a lower cost of TX verification and data integrity as compared to the traditional systems. It is also accredited that the blockchain can also ensure the availability, swift access, and immutability of health data. Moreover, it can also provide unique identities to all patients. However, authors foresee inevitable glitches in the use of blockchain such as high TX volume of health records, and related massive storage requirements, security, and privacy issues concerning user data.

Though the research work discussed above has undoubtedly made some significant contributions towards blockchain and IoT domain. Nevertheless, there are many open issues such as preserving data privacy in a smart city environment, user-defined fine-grained access control, fast TX settlement, users' right to forget (concerning data deletion), an incentive for users to share their data, and distributed storage of user data without centralized control. Therefore, to fill the respective research gaps, we propose "PrivySharing," a blockchain-based secure and privacy-preserving data sharing framework. The proposed solution aims to protect a smart city environment against most of the data integrity and privacy threats. The experimental results have shown that a carefully designed blockchain solution can ensure user data privacy and integrity in various network settings as per the wishes of the data owner. It also effectively protects against false data injection and Sybil Attacks. Moreover, PrivySharing complies with some of the significant data security and privacy requirements of the European Union General Data Protection Regulation (EU GDPR). The significant contributions of this paper are:

1. Provides protection against most of the external as well as insider attacks threatening user data integrity and privacy in a smart city setting.
2. Compliance with some of the essential requirements of EU GDPR.
3. A blockchain-based solution providing the "right to forget" concerning user data.
4. A scalable (concerning blockchain size), secure, and an efficient (in terms of energy consumption and computational requirements) data sharing framework.
5. User-defined fine-grained access control to user data.
6. Providing a transparent and auditable network operation and simultaneously controlling the exposure of users' private data.
7. Secure client access to the blockchain network through a REST API.
8. A reward system for the users for sharing their data with the stakeholders/third parties.

1.2. Basic terminologies

Before getting involved with the detailed architecture of PrivySharing, it is imperative to understand some terminologies specific to Hyperledger-Fabric:

- **Smart Contract (SC).** A SC is a sort of a digital contract based on certain rules between different organizations in the form of an executable code (Hyperledger-Fabric, 2019b). Blockchain network uses smart contracts not only to encapsulate information but also to automate certain aspects of business TXs. Applications invoke a smart contract to generate TXs that are further recorded on the ledger.
- **Chaincode.** The difference between smart contracts and chaincode is that, a smart contract defines the TX logic that updates the state of a business object contained in the world state. Whereas, a chaincode can be termed as a technical container that may contain multiple related SCs for installation and instantiation. When a chaincode is deployed, all smart contracts within it are made available to the applications (Hyperledger-Fabric, 2019b).
- **Committing Peers.** Every peer node in the Hyperledger-Fabric blockchain is a committing peer. However, a Committing Peer does not have a smart contract installed. It just validates and commits a new block of TXs sent by the Ordering Service (ODS) to its copy of the ledger (Hyperledger-Fabric, 2019a).
- **Endorsing Peers.** These are special committing peers with the capability to run the smart contracts. They prepare, sign and endorse the responses to the TX proposals sent by the clients, in line with the endorsement policy of the respective channel (Ch) (Hyperledger-Fabric, 2019a).
- **Ordering Service (ODS).** It is a collection of some peer nodes that arrange the new TXs in a block and then broadcast that block to all the peers of the concerned Ch (Hyperledger-Fabric, 2019a).
- **Membership Service Provider (MSP).** While Certificate Authorities (CAs) issue X.509 certificates to the network entities, an MSP states that which CAs are accepted by the blockchain network and also determine that which peer nodes are members of which organization. Different MSPs can be used to represent various organizations or multiple groups within an organization. Usually, the MSPs are defined at the network, Ch and local/peer level.

1.3. Organization of the paper

The rest of the paper is organized into four sections. Section 2 presents the detailed architecture, reward mechanism, working, and security analysis of "PrivySharing." Experimental results, some limitations of the proposed solution and a way forward to address these limitations are illustrated in Section 3. Finally, the paper is concluded in Section 4, with a gist of future work.

Table 1
List of assets.

Data types	Assets
Health Data	- Health Alert (Heart rate, blood sugar, blood alcohol, etc.) - Full Health History - Insurance Cover - Health Payment Claims - Type of Disease - Current Disease History
Smart Car Data	- GPS Data - Accident Alert - Damage Assessment - Servicing and Auto Payments
Smart Meter Data	- Line Status - Units Consumed and Bill - Consumption Pattern
Surveillance Data	- Equipment Status and Servicing - Security Breach Alert - CCTV Recording
Financial Transactions	- Income - Expenses - Tax

2. Privysharing: Blockchain-based secure data sharing

By leveraging data integrity and smart contract features of the blockchain, various operations in a smart city environment can be securely and autonomously performed. Moreover, blockchain also protects against the adverse effects of server hacking and falsification/modification of permissions (Cui et al., 2018). No doubt, people in a smart city environment feel safe while sharing their personal information only when they have the assurance that their personal and sensitive data collected by various devices are fully protected and they have control over it (Mazhelis et al., 2016). Such assurance can only be provided by none other than a prudently selected and assiduously designed blockchain technology.

2.1. Smart city scenario

We assume that Alice is living in a smart city where every aspect of her life is being monitored and controlled through numerous sensors and smart devices. The critical aspects include monitoring of key health parameters, smart car operation and service management, smart living operation and service management including smart meters generating data concerning energy consumption, surveillance cameras, and intrusion detection equipment generating security-related data and financial TXs to keep the services running. For better understanding, we have formulated a list of numerous assets (associated with a specific type of data) that Alice owns (as shown in Table 1). Based on these assets, Alice can easily decide about the permissions (shown in Table 2) to be granted to the stakeholders/third-parties in relation to her data assets. Such a distinction among the stakeholders/third-parties further assists Alice to plan and control the access to her data. It is also assumed that all the registered users of the smart city network, whether offline or online, interact with each other through the PrivySharing (blockchain) APIs.

To implement the above mentioned smart city use case we have used Hyperledger-Fabric as the underlying blockchain platform due to its effective data security and privacy preserving capabilities as compared to other blockchain platforms (Makhdoom et al., 2018a; 2018b). Hyperledger-Fabric is a private and a permissioned blockchain that restricts participation in the network to the authorized parties only. The key feature that distinguishes Hyperledger-Fabric from other blockchain technologies is that in Hyperledger the blockchain ledger consists of two distinct but related parts, i.e., a blockchain to log the TXs and a world state (a database such as CouchDB (Anderson et al., 2010), and LevelDB (Dent, 2013)) to keep track of the ledger states.

2.2. Network architecture

As shown in Fig. 2, we have designed a smart city blockchain network comprising eleven organizations and their associated peer nodes. Keeping in view the sharing of different categories of users' data with different stakeholders (shown in Table 2) and the requirement to ensure user data privacy and security, the blockchain network shown in Fig. 3 comprises five different data Chs. Where Ch1 is used for the sharing of users' health data and organization-2 (O2), O3, and O5 are its members. Similarly, Ch2 is for smart transportation data and it comprises O3, O4, O5, and O6. Whereas Ch3 is for smart energy, Ch4 for smart security and Ch5 handles financial data (e.g., income, expenses and taxes). A Ch provides a completely separate communication mechanism between a set of organizations. Moreover, every Ch is independent of the other Chs. Hence, these Chs serve to preserve the privacy of user data by securely sharing a particular type of data with authorized entities only. The network is initiated by organization-1 (O1), i.e., the Ministry of Development and Smart Services and is governed by the policy rules specified in the network configuration (NC). NC also controls access to the smart city network. Later, O1 updates NC and gives administrative (admin) rights to O2, O3, and O4 as well. These organizations can now create consortia and Chs to add more network members. Similarly, every Ch is regulated by the policy rules specified in the respective Channel Configuration (CC). In this setting, Ch1 is under the control of O2 and O5 and is governed by CC1. Correspondingly, Ch2 is regulated by CC2, and so on.

The CC is essential for Ch security, e.g., if the client application (clientApp) wants to access a SC on P1, then P1 consults its copy of CC1 to determine the operations that clientApp can perform. Moreover, there is a separate ledger for every Ch, and all the peer nodes have to maintain a copy of the ledger concerning every Ch they are a member of. Therefore, if a peer, say P4, is a member of three different Chs, then it has to maintain three ledgers. Data in a Ch is isolated from the rest of the network including other Chs. Another important aspect of smart city blockchain network is the ODS, which is common to all the Chs. In this setup, the ODS has four ordering nodes, one each from O1, O2, O3, and O4. Each node in the ODS keeps a record of every Ch created through NC. Regarding CAs, every organization in the network can have its own CA. But there is one Root CA (RCA) in the network to establish the root of trust. As a Proof of Concept (PoC) for PrivySharing, we are using Hyperledger-Fabric RCA to issue X.509 certificates to all the network entities. These certificates serve to authenticate the network entities and to digitally sign the client application TX proposals and smart contract TX responses. A user accesses the network through a clientApp with a specific X.509 ID, using a SC. It is imperative to mention that only the endorsing peers can see the SC logic as they have to run the users' TX proposals to prepare the responses.

To ensure the privacy of critical user data within a Ch, i.e., keeping part of user data private from some organizations within a Ch, we adopted a methodology of “Private Data Collection,” in which the critical private data is sent directly to the authorized organizations/stakeholders only. This data is stored in a private database (a.k.a sideDB) on the authorized nodes. While private information is stored on the authorized nodes, only the hash of this data is processed, i.e., endorsed, ordered, and written to the ledgers of every peer on the Ch. The hash of the data serves as evidence of the TX, and it also helps in the validation of the world state. A vital data security feature here is that the ordering nodes do not see the private data. However, to further increase the level of data privacy/confidentiality, the user has the option to encrypt his pri-

Table 2
Assets, stakeholders, and access rights.

Assets	Users / Stakeholders	Access Rights
Health Data		
Health Alert (blood alcohol, blood sugar, heart rate, etc.)	- Alice	- Read
	- Pri (Primary) Medical Center	- Read and write
	- Police	- Read
Health History	- Alice	- Read
	- Pri Medical Center	- Read
	- Alice and Pri Medical Center	- Modify (Requires consent of both Alice and the medical center)
Insurance Cover	- Alice	- Read
	- Pri Medical Center	- Read
	- Health Insurer	- Read
	- Alice and Health Insurer	- Modify (Requires consent of both, Alice and the insurer)
Health Payment Claims	- Alice	- Read and write
	- Pri Medical Center	- Read
	- Health Insurer	- Read
Type of Disease	- Alice	- Read
	- Pri Medical Center	- Read and write
	- Health Insurer	- Read
	- Ministry of Health	- Read
Current Disease History	- Alice	- Read
	- Pri Medical Center	- Read
	- 2nd Medical Center	- Read
Smart Car Data		
GPS Data	- Alice	- Read
	- Car Service provider	- Read
	- Roads/Transportation Authority (ITS)	- Read
Accident Alert	- Alice	- Read
	- Police	- Read and write
	- Car Insurer	- Read
Damage Assessment	- Alice	- Read
	- Car Insurer	- Read
	- Workshop	- Read and write
Servicing and Auto Payments	- Alice	- Read
	- Smart Parking	- Read and write
	- Security Service Provider	- Read and write
	- RTA	- Read and Write
Smart Meter Data		
Line Status	- Alice	- Read
	- Lineman	- Read
Units Consumed and Bill	- Alice	- Read
	- Finance Manager of the Service Provider	- Read and write
Consumption Pattern	- Alice	- Read
	- Operations Manager of the Service Provider	- Read
Total Energy Consumption	- Ministry of Power	- Read
Surveillance Data		
Equipment Status and Servicing	- Alice	- Read
	- OEM/Service Provider	- Read
Security Breach Alert	- Alice	- Read
	- Police	- Read
CCTV Recording	- Alice	- Read
	- Police	- Read
Total Incidents of Security	- Ministry of Interior	- Read
Financial Transactions		
Income	- Alice	- Read and write
	- Bank	- Read
	- Revenue	- Read
Expenses	- Alice	- Read and write
	- Bank	- Read
Tax	- Alice	- Read
	- Bank	- Read
	- Revenue	- Read and write

vate data such that not even the peers/nodes authorized to view data stored in the private data collection can see the original contents. The data is encrypted using AES-256 bit symmetric encryption key and then stored in the private data collection. Later on, only the authorized users who have access to the decryption key can query the user's private data. Supplementary to the data encryption, there is an additional feature of signed encryption of private data for an increased level of user authentication and data security.

Another important feature of our proposed network architecture is the use of Membership Service Provider (MSP) at various levels such as network, Ch and local/peer. The network MSP

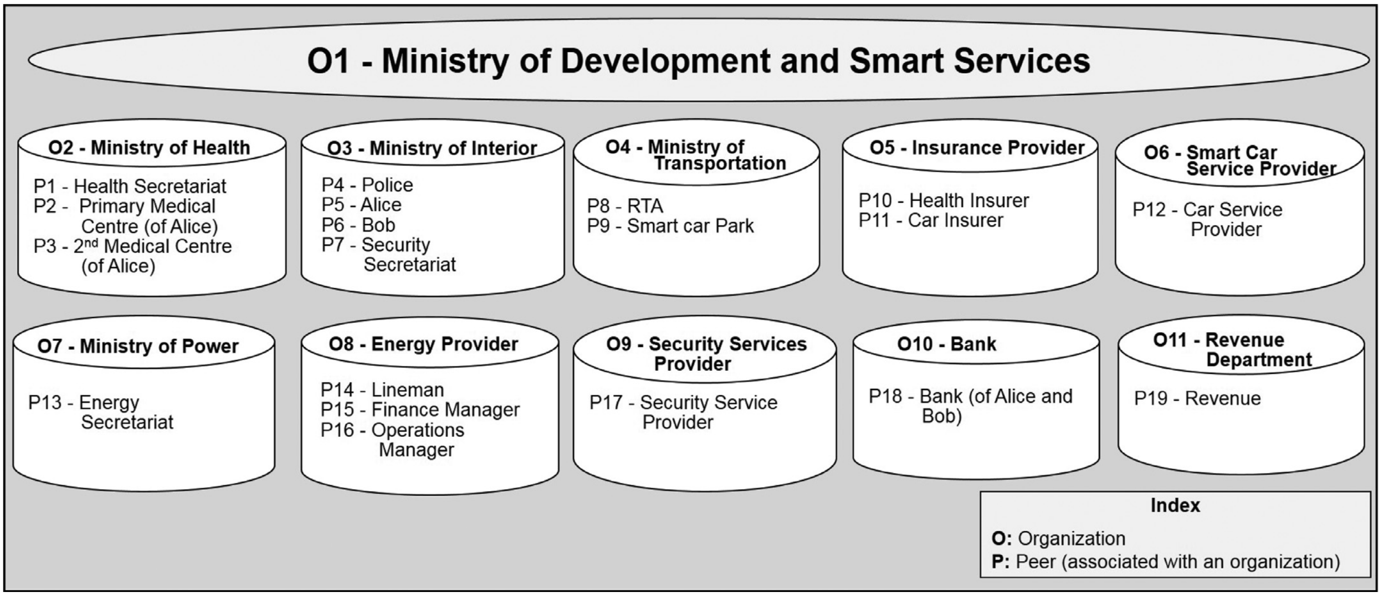

O1 - Ministry of Development and Smart Services
O2 - Ministry of Health
P1 - Health Secretariat
P2 - Primary Medical Centre (of Alice)
P3 - 2nd Medical Centre (of Alice)
O3 - Ministry of Interior
P4 - Police
P5 - Alice
P6 - Bob
P7 - Security Secretariat
O4 - Ministry of Transportation
P8 - RTA
P9 - Smart car Park
O5 - Insurance Provider
P10 - Health Insurer
P11 - Car Insurer
O6 - Smart Car Service Provider
P12 - Car Service Provider
O7 - Ministry of Power
P13 - Energy Secretariat
O8 - Energy Provider
P14 - Lineman
P15 - Finance Manager
P16 - Operations Manager
O9 - Security Services Provider
P17 - Security Service Provider
O10 - Bank
P18 - Bank (of Alice and Bob)
O11 - Revenue Department
P19 - Revenue
Index
O: Organization
P: Peer (associated with an organization)

Fig. 2. Network participants.

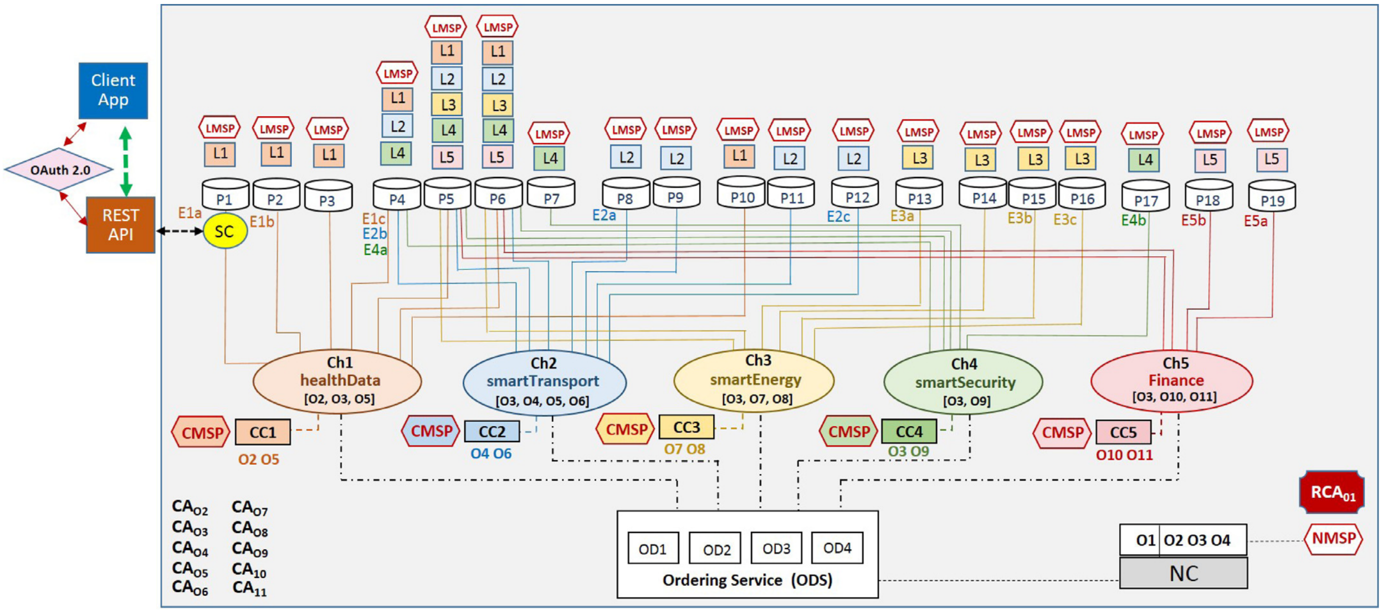


Fig. 3. Smart city blockchain-network architecture.

(NMSP) defines, who all are the members of the network and who out of them have the admin rights. Additionally, an NMSP also defines that which RCAs/CAs are trusted. On the other hand, the Ch MSPs (CMSP) outline admin and participatory rights at the Ch level. All the peers and the ODS share a common CMSP to correctly authenticate and verify the authorizations of the Ch members. A use case for the CMSP is that, e.g., an admin of an organization wants to instantiate a SC on Ch1, then by looking at the CMSP, the other Ch members can verify that whether that admin is a part of a specific organization or not and whether he is authorized to instantiate the SC on Ch1 or not.

Similarly, a local MSP (LMSP) is defined for every client-node/peer. The LMSP associates a peer with its organization. It also defines the permissions for that peer and allows it to authenticate itself in its TXs on the Ch. Here a question may arise that, what is the difference between CC and a CMSP. A CC contains the policies that govern that Ch, i.e., which organizations can regulate the Ch and add new members. Whereas, a CMSP establishes the linkage between the nodes and their respective organizations, and what roles a node can play within a Ch, i.e., can it instantiate a SC on a Ch? Concerning decentralization aspect; the use of a dedicated trusted CA, a blockchain admin, and a business network admin by every organization in the blockchain network provides some degree of decentralization as compared to all the admin rights resting with a single organization.

Another question may arise that what advantages do we get by using multiple Chs for different data types as compared to a single Ch blockchain network to share all the types of data. There are two aspects to this selection; one is scalability, and second is increased privacy of user data. From the scalability point of view, if there is only one Ch for all types of data, then it means that the users will have to store the ledger comprising all those TXs

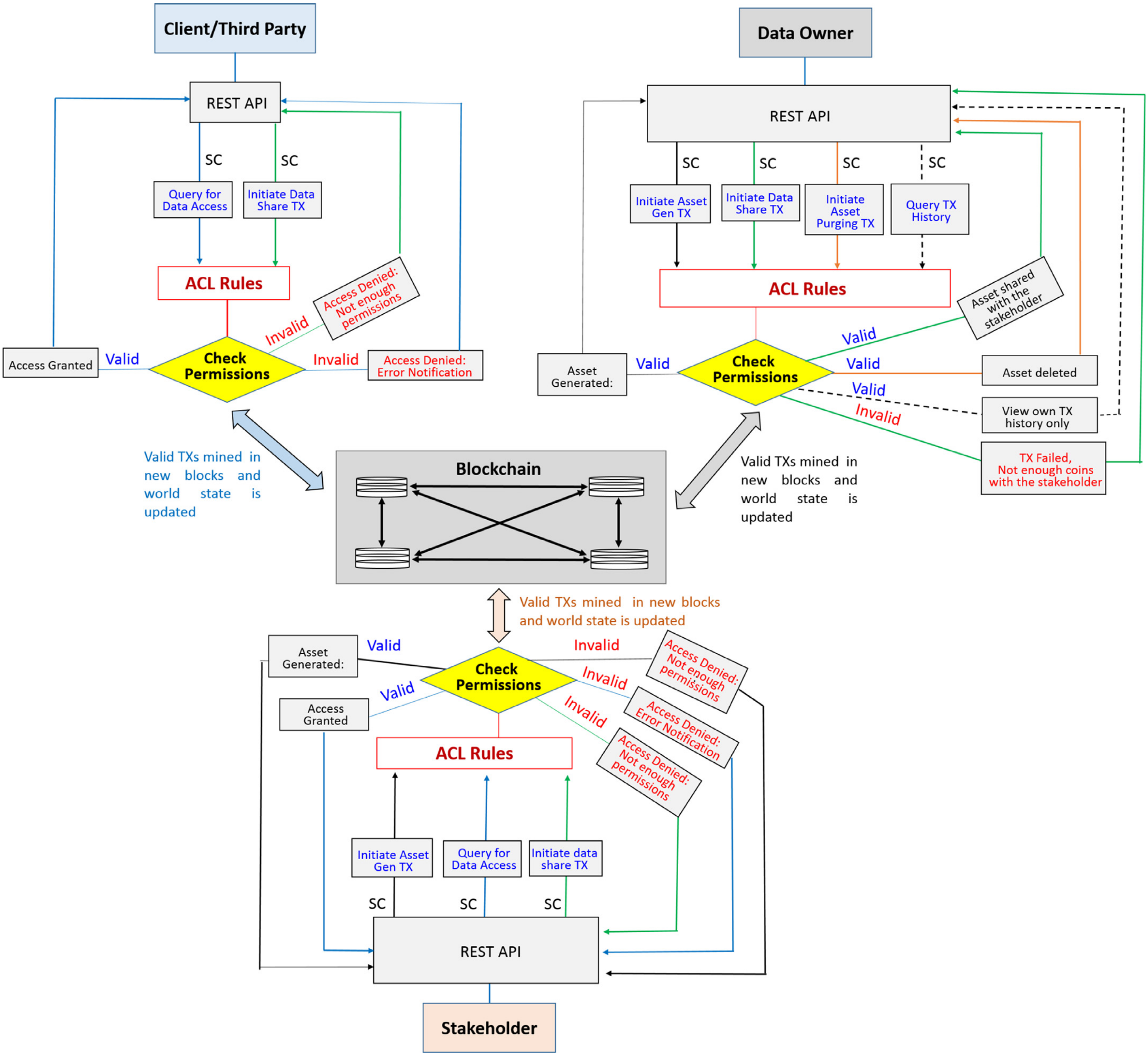


Fig. 4. Smart contract TXs.

that are not even related to them. Hence, the ledger size will increase rapidly, thus putting more strain on storage resources of all the users/peers. Whereas, in the case of “PrivySharing”, the users will maintain a ledger that stores only that data which concerns all the users of that particular Ch. Moreover, the experimental results (Section 3) have validated that the multi-Ch blockchain network scales well as compared to a single-Ch blockchain. As far as the privacy of user data is concerned, a data specific Ch shared only by some of the stakeholders provides more privacy than a single Ch comprising all the stakeholders sharing multiple data types. Although, use of multiple data specific Chs seems scalable as compared to a single Ch, yet the requirement for users to maintain a ledger each for every Ch, in which they participate, may still crave for ample storage resources.

PrivySharing framework has been designed, developed, and tested based on the agile blockchain application development guidelines proposed by (Marchesi et al., 2018). The said guidelines helped in a systematic design, development, and testing of PrivySharing network architecture, SC functionality, and efficacy of ACL rules. Moreover, influenced by these guidelines, Fig. 4 highlights different TXs initiated by various actors operating in the smart city network. Every TX and its associated decision/response based on ACL rules are depicted by the same colored line. E.g., a client/third party can only query for some user data asset. If it is authorized to access the data, the query will be successful. Otherwise, there will be an access denied error message. Both the query and respective response are shown by blue lines. Similarly, the data share TX is sketched in green color. As per PrivySharing business model the client/third party should not be allowed to submit a data sharing TX, hence, if a client still initiates a TX to share data asset of some user, then he gets a “access denied: not enough permissions” error message. TXs concerning data owner and stakeholder are also projected accordingly.

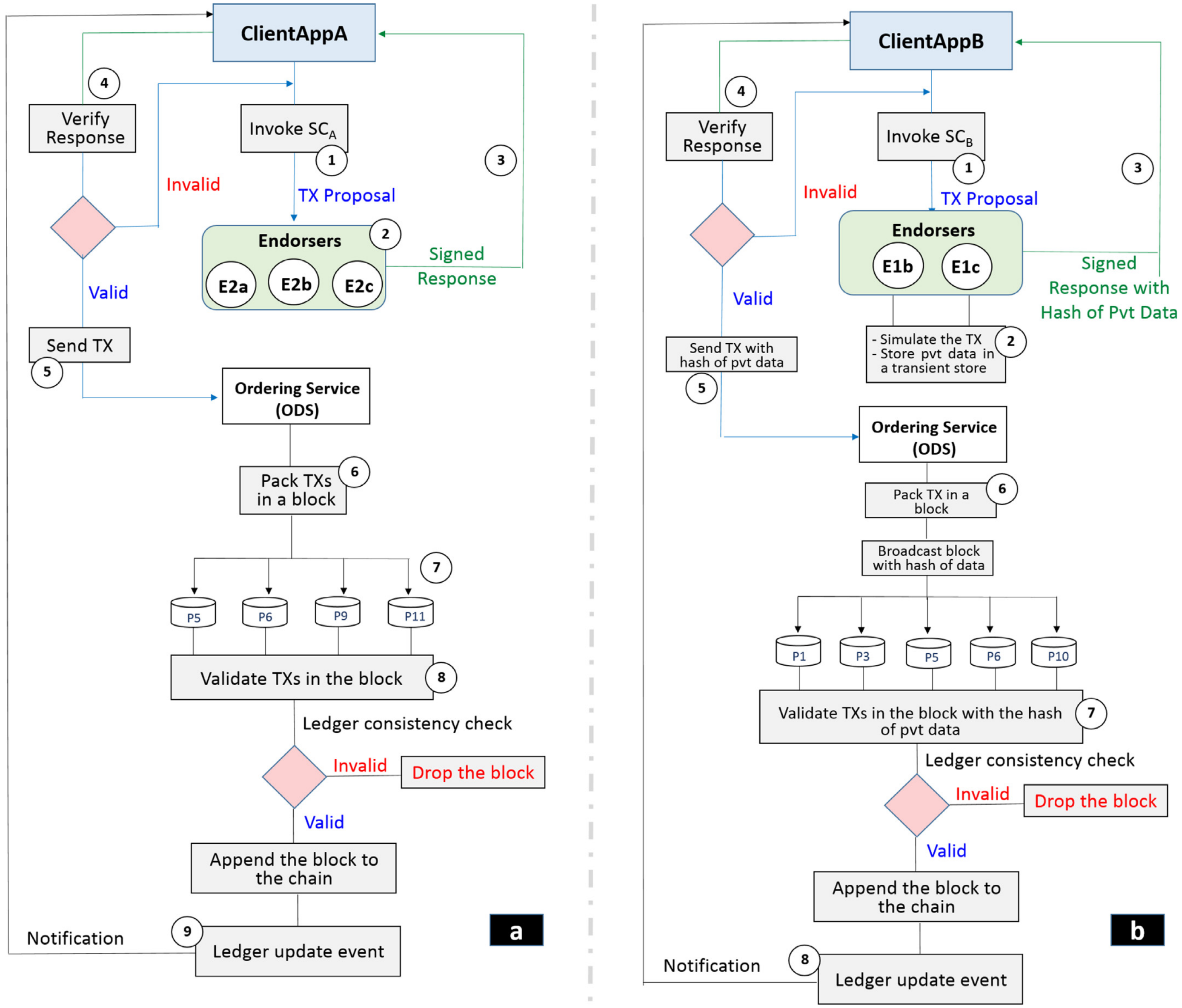


Fig. 5. (a) Plain TX flow, and (b) Private data TX flow.

2.3. *Smart city blockchain - Plain TX flow*

There are two types of TXs; one is plain TX that can be viewed by all the Ch members and the other one is private data TX that is to be shared only with some selected peers in a Ch. In this regard, e.g., a plain TX that is required to update Alice's car's current location state on Ch2 is initiated by the ClientAppA installed in Alice's smart car. This TX (as shown in Fig. 5(a)) is processed in the following steps:

Step-1. ClientAppA invokes the SC_A and sends a TX proposal containing the current location of Alice's car to the pre-defined endorsers as per SC_A endorsement policy on Ch2. In this case, the endorsers are E2a (RTA), E2b (Police) and E2c (Car Service Provider). A TX will be approved if it is endorsed by a minimum two out of the three prescribed endorsers.

Step-2

2.1. Three endorsers E2a, E2b, and E2c, invoke SC_A with the proposal.

2.2. SC_A generates a query or update proposal response. The endorsers, E2a and E2b endorse the proposal for correctness.

Step-3. E2a and E2b both send a signed (endorsed) TX proposal response along with RW (read, write) set back to the ClientAppA. At this stage, the endorsing peers do not apply the proposed update to their copy of the ledger.

Step-4. ClientAppA verifies that the response received from at least two endorsers is the same, i.e., deterministic. However, there is a possibility that the results were generated at different times on different peers with ledgers at different states. Hence, the peers can return different TX responses for the same TX proposal. In this case, an application can simply request a more up-to-date proposal response. Another less likely possibility is that the SC might be non-deterministic, e.g., while getting forex (foreign exchange rates) data from some websites, the TX responses can be different, as forex rates may differ at different times. Therefore, inconsistent results cannot be accepted by the application and applied to the ledger.

Step-5. Once the ClientAppA verifies the endorsers' responses, it sends the TX to the ODS.

Step-6. ODS then groups the received TXs in a block. The sequence of TXs in a block is not necessarily the same as the order

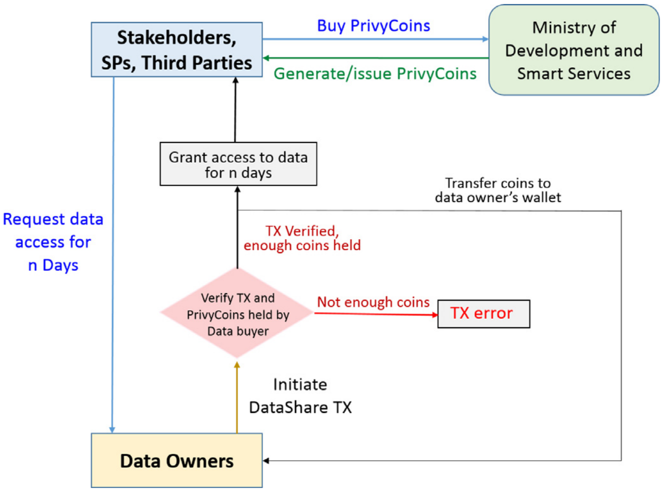
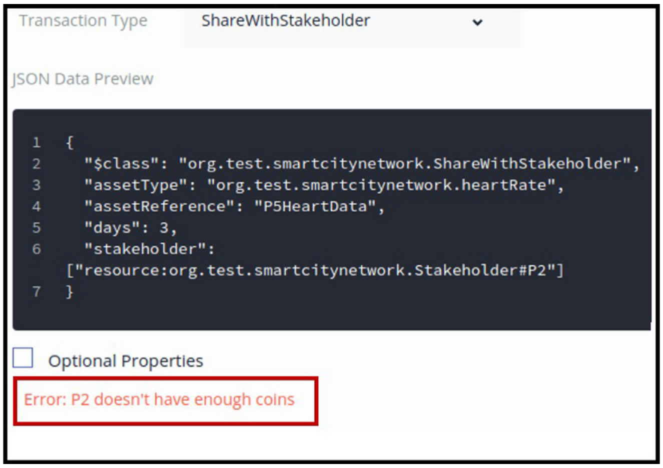

Stakeholders, SPs, Third Parties
Buy PrivyCoins
Ministry of Development and Smart Services
Generate/issue PrivyCoins
Grant access to data for n days
Transfer coins to data owner's wallet
Request data access for n Days
TX Verified, enough coins held
Verify TX and PrivyCoins held by Data buyer
Not enough coins
TX error
Initiate DataShare TX
Data Owners

Fig. 6. Reward mechanism based on PrivyCoins.



Transaction Type ShareWithStakeholder

JSON Data Preview

```
{
  "$class": "org.test.smartcitynetwork.ShareWithStakeholder",
  "assetType": "org.test.smartcitynetwork.heartRate",
  "assetReference": "P5HeartData",
  "days": 3,
  "stakeholder":
["resource:org.test.smartcitynetwork.Stakeholder#P2"]
}
```

☐ Optional Properties

Error: P2 doesn't have enough coins

Fig. 7. Error for not having enough coins.

of arrival of the TXs at the orderers. However, the generated blocks are final, and there are no forks. Moreover, the orderers do not host the ledger and the SCs, and they are also not concerned about the value of the TX rather they just package the TXs into the blocks.

Step-7. ODS Broadcasts the next proposed block to all the peers on the Ch2.

Step-8. All the committing peers validate each and every TX in a block (in the same sequence as they appear in the block) to ensure that it is correctly endorsed by all the required endorsers before it is applied to the ledger. Once a TX is verified correctly, the peers perform a ledger consistency check to establish that the current state of the ledger is compatible with the state of the ledger when the proposed update was generated. World state is updated based on the validated TXs. It is to be noted that, the failed TXs are not applied to the ledger, but they are retained for audit purpose. Moreover, TX validation in Step-8 does not require running of SCs. This is done only by the endorsers. Hence, SCs are installed only on the endorsers. This keeps the logic of the SCs confidential to the endorsing organizations only. Moreover, peers also mark each TX in each block as valid or invalid. Finally, a new block is appended to the hash chain stored in the ledger L2, maintained by all the peers in their file system.

Step-9. Ledger update event is generated, and the ClientAppA is notified.

It is important to note that prior to appending a block, a version check is performed to ensure that the states being updated are the same that were read during SC execution. It protects against double spending and other data integrity threats. The above mentioned TX workflow mediated by the orderers is called “Consensus”, as all the peers reach on an agreement about the content and the order of the TXs.

2.4. Smart city blockchain - Private data TX flow

As per smart city network settings shown in Fig. 3, if a wearable blood alcohol monitoring device on Alice generates an alert to be seen only by her primary medical center and the local police for immediate response. In such a case, it is required to keep such a TX private which should not be seen by other members on Ch1 except P2, P4, and P5. Such a private data TX (as shown in Fig. 5(b)) is processed in the following steps:

Step-1. The clientAppB submits a proposal request to invoke a SC function (RW private data) to the endorsing peers E1b (Primary Medical Center) and E1c (Police), which are part of the authorized organizations of the collection (defined by the private data dissemination policy on health alert). The private data concerning health alert on blood alcohol level is sent in a transient field of the proposal.

Step-2. E1b and E1c simulate the TX and store the private data in a transient data store (temporary storage local to them). The endorsing nodes also distribute the private data based on the collection policy to authorized peers via gossip. But in this case, we only have three peers, i.e., P2(E1b), P4(E1c) and P5.

Step-3. E1b and E1c send the proposal response back to the clientAppB with public data, including a hash of the private data key and value (Blood alcohol level). No private data is sent back to the clientAppB in plaintext.

Step-4. The clientAppB verifies that the RW sets received from E1b and E1c are same.

Step-5. The clientAppB submits the TX with a hash of the private data to the ODS.

Step-6. The ODS packs the TX in the latest block. The block with the hashed value is distributed to all the peers on Ch1.

Step-7. All the peers on the channel validate TX with the hash of the private data in a consistent way, without knowing the actual private data.

Step-8. Ledger update event is generated, and the clientAppB is notified.

2.5. Reward mechanism

PrivySharing incentivizes the users to share their data with other users, stakeholders, or third parties by rewarding them with a local digital token named “PrivyCoin”, as exhibited in Fig. 6. PrivyCoin is just like an asset in the smart city network that is issued only by the network admin (Ministry of Development and Smart Services) against the payment in terms of fiat currency. The secure execution of such a TX is not covered in this paper. However, it is envisaged that the stakeholders can pay the ministry through any secure payment app and then receive the coins in their wallet, just like any other cryptocurrency/token. PrivyCoin is primarily used for trading or getting access to the data assets. After acquiring PrivyCoins, the stakeholder forwards the request for data access along with asset ID and the duration of access (in terms of days). Currently, in PrivySharing, the third parties/stakeholders pay one PrivyCoin to a user to get access to a data asset for one day (24 hours). Hence, if a stakeholder wants to get access to two data assets of a user for five days, he has to pay ten PrivyCoins to the user. Upon receiving the request to share data, it is only the prerogative of the data owner to initiate the data sharing TX. The data owner gets the incentive as soon as the data sharing TX is commit-

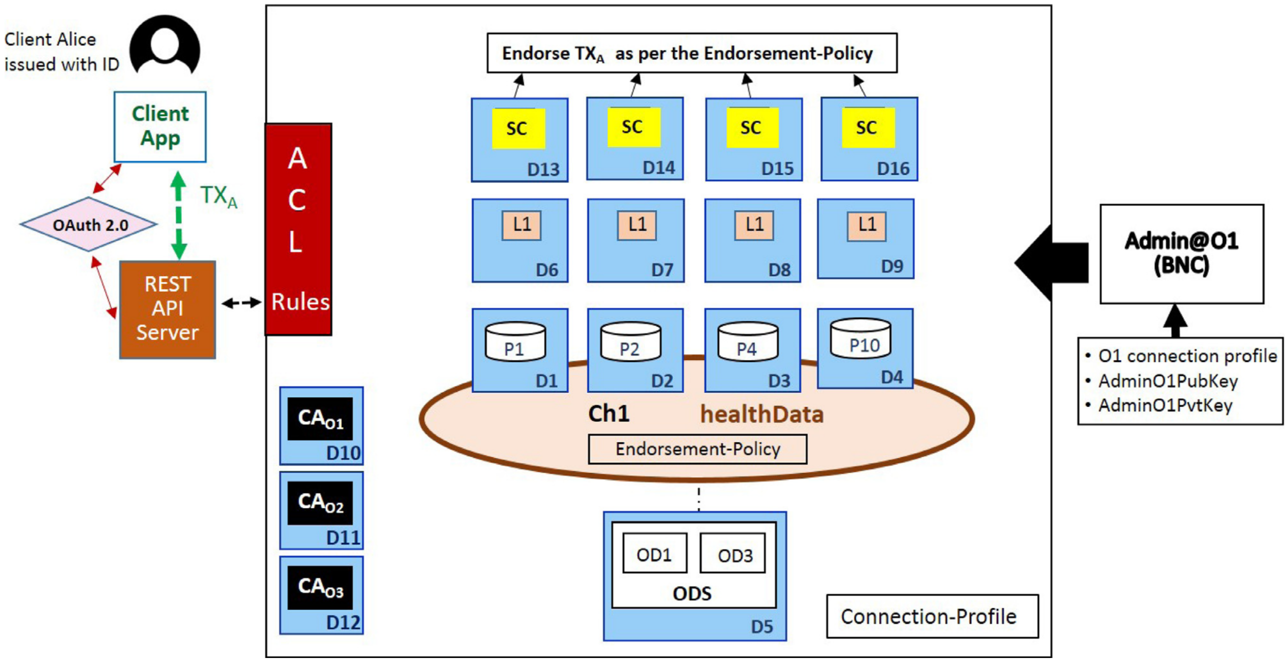


Fig. 8. Elements of PrivySharing network security.

ted. In this context, if a stakeholder does not have requisite coins in his account, the TX will fail (shown in Fig. 7). The pseudocode for the reward-based data sharing TX is illustrated in Algorithm 1. This algorithm can be summarized into four steps. Firstly, the data asset to be shared is obtained from the asset registry. Whereas, the input data structure of the data sharing *TX* contains the asset type (e.g., Heart Rate, Blood Sugar, etc.), the asset reference (ID of the asset), the time duration of sharing (e.g., three days) and a list of stakeholders (e.g., P2, P4). Then, the algorithm checks whether the asset has already been shared with the stakeholders or not. After that, stakeholders pay PrivyCoins to the data owner. Finally, the asset status is updated, and an event is emitted to notify the related parties, i.e., user and the stakeholders.

Algorithm 1 Reward-based *data sharing* with the *stakeholders*.

```
Input: ShareWithStakeholder($tx$)
  $asset \leftarrow assetRegistry.get(tx.assetReference)$ {STEP-1: Retrieving the asset from asset registry}
  {STEP-2: Check, whether an asset is already shared with the stakeholder or not}
  for all $stakeholder$ In $tx.stakeholders$ do
      if $asset.stakeholdersWithAccess$ is not Empty then
          $stakeholderId \leftarrow stakeholder.operatorId$
          if $stakeholderId$ exists in $asset.stakeholdersWithAccess$ then
              MESSAGE: Data already shared.
              Jump to the next $stakeholder$
          else
              push $stakeholderId$ into $asset.stakeholdersWithAccess$
          end if
      else
          $asset.stakeholdersWithAccess \leftarrow [stakeholderId]$
      end if
      {STEP-3: Stakeholders pay coins to the asset owner}
      $coins \leftarrow Coinsbelongtostakeholder$
      if $coins.length < tx.days$ then
          return ERROR: $stakeholder$ does not have enough coins
      else
          for $j = 0$ to $tx.days - 1$ do
              $coins[j].owner \leftarrow asset.owner$
              Update coin status
          end for
      end if
      {STEP-4: Event generation}
      Emit event of sharing
  end for
  Update $asset$ status
  return Sharing Success
```

3. Security analysis

The security, being the core objective of this work has been assessed at every level of the network operation. The key aspects shown in Fig. 8 are illustrated as under.

When the blockchain network is first created, all the peers and orderer organizations are issued with certificates from respective RCA, or other trusted CAs. Then, a connection profile is created for all the network entities including Chs, ODS, organizations, peers, and CAs. The connection profile defines the complete blockchain network setup. E.g., for a Ch, it defines the Ch name, its associated ODS and peers. It also defines which peers are the endorsing peers for that particular Ch. For an organization, it defines the namespace, MSP ID, member peers, and the respective CA. The peers' profile includes the namespace, URL including the port number, and the Transport Layer Security (TLS) certificate for its principal organization. The key point here is that no other peer (with the intention of endorsing the TXs on a Ch) can join the network if it is not defined in the connection profile. It is clarified that by peers, we mean committing, endorsing or ODS peer nodes that maintain the blockchain network. Whereas, the users/clients access the blockchain network through REST API or clientApps. The smart city blockchain network entities including ODS, peers, CAs, ledgers, and SCs run in separate docker containers (symbolize by blue boxes numbered from D1 to D15 in Fig. 8. This separation minimizes the effects of a container compromise, i.e., if one container's security is breached the other containers remain unaffected.

To deploy the business network model (PrivySharing in this case) that comprises asset definitions, TX and event logic, and ACL rules on the blockchain, the admin of responsible organization (O1 in this scenario) requires a Business Network Card (BNC). The BNC is created using the connection profile of the organization and the valid public and private key for that admin issued by the authorized CA, as defined in the connection profile. The TXs initiated by the clientApps on a specific Ch are endorsed as per the endorsement policy defined for the respective Ch before the start of the business network. The endorsement policy may include, e.g., what all peers (with endorsing ability) are required to endorse a TX on a Ch concerning health data. Similarly, a TX is considered valid, only if the response of all the required endorsing peers is the same. Hence, only a valid TX will update the world state. Another vital security feature of PrivySharing is that before the start of the business network on the blockchain, business network admins have to be defined and issued with the certificates (Public and Private key pairs) by the respective CAs. These certificates are later used to create the BNCs for the said admins to access the business network. Without a valid BNC, no one can add participants (clients/peers) for an organization. Moreover, every new client/peer added under an organization is also issued with an ID by the respective CA with the approval of the business network admin. These IDs are further used to control access to the users' profile and assets as per the ACL rules defined for the specific Ch.

As far as privacy of user data is concerned, the use of data specific Chs, private data collection, and data encryption does provide some degree of data privacy. However, even if a user's IoT device data is encrypted, still a passive network attacker can infer a pattern of user's activities. The same has been demonstrated by the researchers in (Apthorpe et al., 2017). The authors exhibited that an adversary capable of monitoring the network traffic between a smart home gateway device and the internet can determine the type of IoT devices being used inside a smart home, based on DNS queries. Also, the attacker can analyze the metadata of the network traffic and observe variations in the IoT data send/receive rates. Hence, based on these abrupt changes in data rate/packet size, the adversary can deduce vital information about user's behavior and daily routine. Although, the conventional IoT classification methods do not apply to the blockchain, as the TXs in blockchain contain public keys instead of IP addresses, and are broadcast to the network. Nevertheless, to avert the effects of malicious network traffic monitoring measures such as incorporation of VPN tunneling or obfuscating and shaping all smart home network traffic can be taken to mask variations that encode real-world behavior of the device owner.

Correspondingly, in blockchain-based IoT systems, the combination of device classification and user deanonymization can infer private information about a user to an adversary. Although, in PrivySharing, the IDs of all the members of the network are known and there is also a provision that each user can be issued with multiple cryptographic IDs (Public-Private key pairs) (Hyperledger-Fabric, 2019). Hence, users can use a different ID to communicate with every stakeholder. Such an arrangement seems robust against linking attacks (Dorri et al., 2019). However, blockchain researchers in (Roulin et al., 2018) established the possibility of IoT devices classification by analyzing IoT device data stored on the blockchain by applying Machine Learning (ML) algorithms. Unlike in (Apthorpe et al., 2017), an adversary is assumed to have access only to the data stored on the blockchain rather than the network traffic (Roulin et al., 2018). The attack methodology identifies the IoT devices based on different patterns of timestamp differences in successive TXs of each type of device. However, researchers also proposed combinations of various methods of timestamp obfuscation to avoid device classification. These techniques include: introducing a random delay in the TXs of a device, combining multiple data packets of a specific device into a single TX, and lastly, merging ledgers of numerous devices.

3.1. ACL rules

PrivySharing has embedded user-defined ACL rules in the data sharing chaincodes to protect user data. The graphical illustration of the access control process based on some of the ACL rules is shown in Fig. 9. These rules enforce that the data asset owners have access to their assets only, i.e., no user can see data assets of any other user, and only the data owners can initiate a TX to share their data assets with other users/stakeholders. Similarly, a data owner has the right to revoke the sharing of his assets, and he can also delete his assets when no longer required without affecting the TX history stored on the blockchain. Moreover, as all the TXs are recorded on the blockchain, hence, to increase privacy, a data owner can see the TX history concerning his own assets only. Additionally, valid users can read and update their profiles only, and other users/stakeholders cannot see each other's profile. Users can also delegate the stakeholders to create assets on their behalf. E.g., Alice (P5) delegates her primary medical center (P2) to create a health data asset for her. Accordingly, the stakeholders can only see the data assets that are shared with them or created by them. Lastly, all the users/stakeholders can view their coins only. The pseudocode of the data asset unsharing and asset deletion is accordingly shown as Algorithms 2 and 3, respectively.

Algorithm 2 Unsharing data assets with the stakeholders.

```
Input: UnshareWithStakeholder($tx$)
  $asset \leftarrow assetRegistry.get(tx.assetReference)${COMMENT: Retrieving the asset from asset registry}
  {COMMENT: Removing the stakeholders}
  for all $stakeholder$ In $tx.stakeholders$ do
      if $asset.stakeholdersWithAccess$ is not Empty then
          $stakeholderId \leftarrow stakeholder.operatorId$
          if $stakeholderId$ exists in $asset.stakeholdersWithAccess$ then
              Remove $stakeholder$ from $asset.stakeholdersWithAccess$
          else
              MESSAGE: $Asset$ is not shared with the $stakeholder$
          end if
      else
          MESSAGE: $Stakeholder$ has no access to any record.
      end if
      {COMMENT: Emitting an event of unsharing asset}
      Emit event of unsharing
  end for
  Update $asset$ status
  return  Unsharing Success
```

Algorithm 3 Deleting a data asset.

```
Input: DeleteAsset($tx$)
  $asset \leftarrow assetRegistry.get(tx.assetReference)${COMMENT: Retrieving the asset from asset registry}
  {COMMENT: Removing the asset from asset registry}
  Delete $asset$
  {COMMENT: Emitting an event of Deleting}
  Emit event of asset deletion
  return  Deleting Success
```

3.2. Security of REST API and Dapp

Access to the REST API is secured using the API key which is required to launch the REST API. In addition to the API Key, OAuth-2.0 authorization protocol (Hardt, 2012) is also employed to authorize access to PrivySharing REST server instance, and allow the end-users/clients to interact with the PrivySharing business network deployed on the blockchain. The mechanism of OAuth-based REST API security protocol is shown in Fig. 10. In step-1, the client/user/third-party App sends an authorization request to the PrivySharing business network admin from O1 that also acts as the resource owner. The resource owner then replies with the authorization grant. In step-3, the client sends an authorization token request containing the authorization grant received from the resource owner in step-2 to the authorization server. After validating the authorization grant, the authorization server issues an

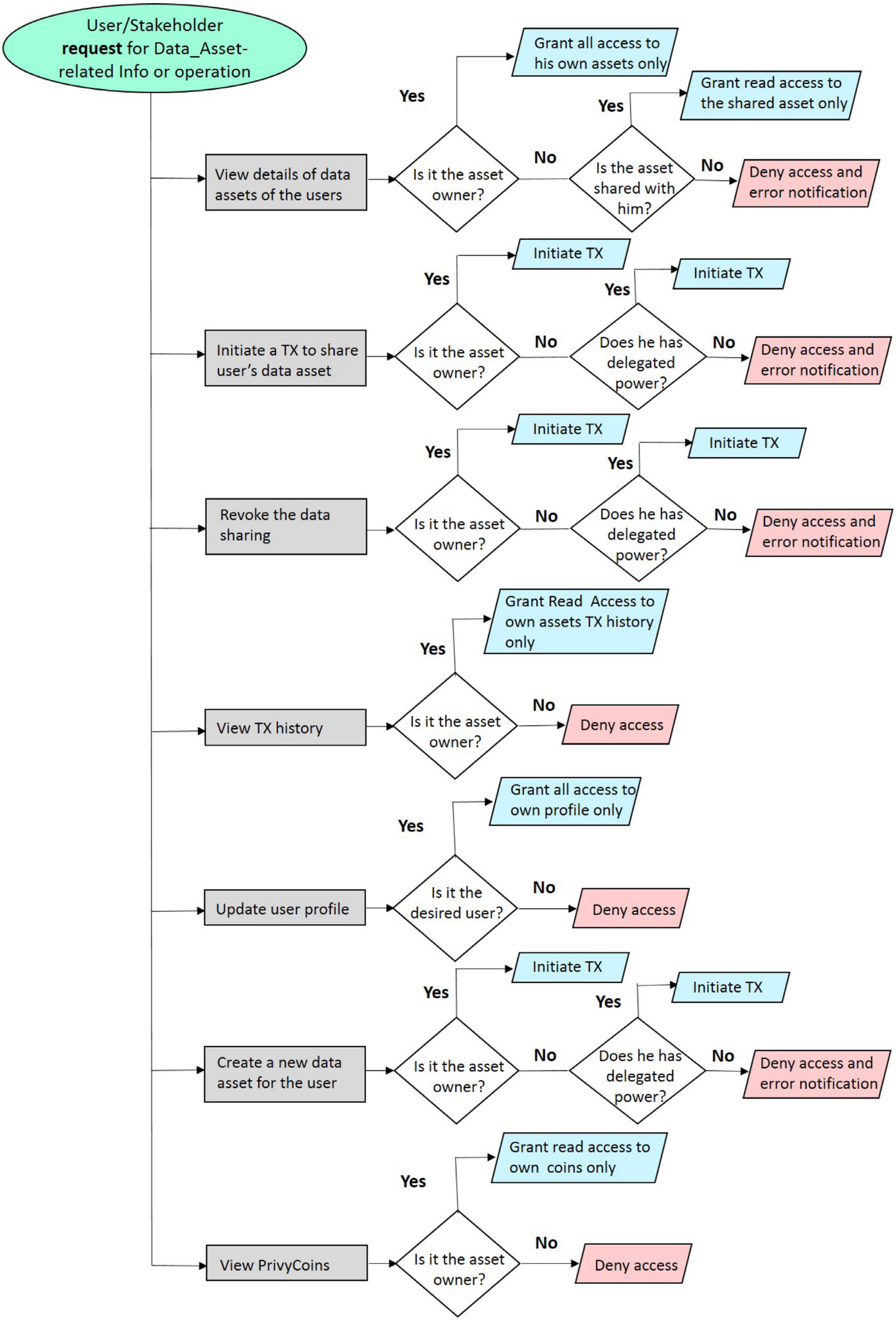

User/Stakeholder **request** for Data_Asset-related Info or operation
View details of data assets of the users
Is it the asset owner?
Yes
Grant all access to his own assets only
No
Is the asset shared with him?
Yes
Grant read access to the shared asset only
No
Deny access and error notification
Initiate a TX to share user's data asset
Is it the asset owner?
Yes
Initiate TX
No
Does he has delegated power?
Yes
Initiate TX
No
Deny access and error notification
Revoke the data sharing
Is it the asset owner?
Yes
Initiate TX
No
Does he has delegated power?
Yes
Initiate TX
No
Deny access and error notification
View TX history
Is it the asset owner?
Yes
Grant Read Access to own assets TX history only
No
Deny access
Update user profile
Is it the desired user?
Yes
Grant all access to own profile only
No
Deny access
Create a new data asset for the user
Is it the asset owner?
Yes
Initiate TX
No
Does he has delegated power?
Yes
Initiate TX
No
Deny access and error notification
View PrivyCoins
Is it the asset owner?
Yes
Grant read access to own coins only
No
Deny access

Fig. 9. ACL rules.

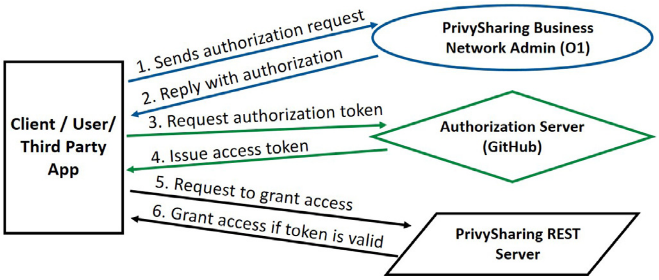

Client / User/ Third Party App
1. Sends authorization request
2. Reply with authorization
3. Request authorization token
4. Issue access token
5. Request to grant access
6. Grant access if token is valid
PrivySharing Business Network Admin (O1)
Authorization Server (GitHub)
PrivySharing REST Server

Fig. 10. PrivySharing REST server OAuth protocol.

access token to the client. The client then requests the PrivySharing REST Server to grant access by presenting the access token. Finally, in step-6, if the token is valid, the client is granted access to call the PrivySharing REST API operations. Currently, there are more than three hundred options for the client REST Server authentication strategies including SAML, LDAP, GitHub and a blend of OSN such as Facebook and Google. For this PoC, we have used Passport-GitHub strategy to authenticate the users. The detailed procedure of enabling OAuth for PrivySharing REST Server is depicted in Fig. 11.

Furthermore, due to the distributed nature of the SCs, the integrity of any business network deployed on the blockchain is guaranteed. Similarly, it also protects against hacking of servers, where, the attackers can change the policy rules, escalate access rights, etc. Correspondingly, protection against application and web vulnerabilities can also be guaranteed with high probability, as any change in the smart contract requires installing and instantiating a new version of the contract on all the endorsing peers. However, it cannot be done discretely. Additionally, due to a distinction between blockchain and the world state, an auditable log of TXs and events is maintained without compromising the privacy of the users' data.

3.3. Restricted access to user data assets via multiple Chs

In addition to restricting access to users' data assets through ACL rules within a Ch, the use of data specific Chs is also helpful in preserving users' data privacy. Through our PoC, we have validated that every Ch in PrivySharing smart city network is independent of other Chs with associated Ch members. As shown in Fig. 12, when P13 from O7 (not a member of Ch1), tries to query a user's heartRate data, he gets an access denied error because he is not authorized to access any data asset propagated on Ch1. As PrivySharing is a permissioned consortium blockchain, all the network members are duly registered and authenticated before joining the network. However, even if an unauthorized node gets added to the system through a corrupt network admin, the ACL rules prohibit the intruder from unauthorized access to users' data assets.

Moreover, Table 3 shows the methodology we adopted to achieve the security objectives derived from smart city threat environment and EU GDPR requirements. However, one of these objectives, i.e., IoT device integrity check has not been addressed in this paper.

4. Experimental results

To validate the security effectiveness and measure the performance efficiency of the proposed solution, we designed, developed and set up a three-Ch smart city data sharing scenario for the sharing of health, smart energy, and financial data. The experimental setting, as shown in Fig. 13, comprises six organizations and twelve peers. However, for a production environment, the minimum nodes required to establish a blockchain network primarily depends upon the type of consensus protocol being used for ordering service. Moreover, other contributing factors may include the type of blockchain application and the degree of decentralization required. Hence, there may be multiple Chs, more than two organizations with their peers and CAs, and numerous stakeholders participating in the ordering service. Currently, Kafka is the recommended consensus protocol for the production environment. Moreover, Kafka-based ordering service is a combination of a Kafka cluster and Zookeeper ensemble. To establish a Kafka cluster and Zookeeper ensemble, there should be a set of a minimum of four Kafka and three Zookeeper nodes to achieve fault tolerance. As a PoC, we deployed the business network model of PrivySharing on Hyperledger Fabric ver 1.4 and validated various security and performance attributes. It is also verified that access to users' data assets are effectively regulated by numerous ACL rules. To measure key performance indicators of PrivySharing, we used Hyperledger Caliper, a blockchain benchmark tool. The experiments were performed on a machine with Intel Core i7 2.9 GHz CPU, 8 GB RAM, and Ubuntu 18.04 operating system.

4.1. Validation of ACL rules

The validity of the ACL rules was checked on both, the Hyperledger Composer-Playground and the REST API. E.g., As shown in Fig. 14(a) and (b), to compare the access rights we have created a user with admin rights that can view assets (blood alcohol level) of all the users, i.e., P5 and P6 in this case. Whereas, the user P5 with ID Pid5 can only see his assets. Moreover, Fig. 14(c) and (d) show that initially, a user P4 with id Pid4 cannot see any asset, as no asset is currently shared with him. However, once user P5 shares his blood alcohol level with P4, he can then see P5's blood-alcohol level. Similarly, only P5 can initiate a TX to share its assets. Whereas, if P4 tries to share the asset of P5 with any other entity then he will get an error (as shown in Fig. 15) as he currently does not have the right to initiate a data sharing TX. As far as the purging of a data asset is concerned, as shown in Fig. 16(a), a data asset say P5's blood sugar can be deleted. However, Fig. 16(b) manifests that the historical record (TX history) of a deleted asset remains immutable in the blockchain. Sequel to this, the TX history concerning the data assets can only be viewed by respective users only. As shown in Fig. 16(c) and d, only P5 (Alice) can view the record of her data sharing TXs. Whereas, any other user, say P6 (Bob) cannot see Alice's TX history. However, even if a blockchain admin is allowed to view the transaction history of all the nodes for accountability, the admin still cannot see the value of the data asset being shared.

4.2. Performance efficiency

Though, a detailed comparison of performance efficiency of Hyperledger Fabric with some of its counterparts is already presented in (Makhdoom et al., 2018a) and (Pongnumkul et al., 2017). However, as per the experimental settings for phase-1 (as shown in Fig. 13), we measured the time taken to commit various types of TXs in the preview of PrivySharing. The average commit time has been measured for three different TXs based on ten iterations. The TXs include; plain text (PlainText) TX, private data (PvtData) TX, and encrypted private data (EncPvtData) TX. These TXs are analyzed in two different consensus environments, i.e., SOLO and Kafka.

It is evident from Fig. 17 that all types of TXs irrespective of the employment methodology take less than 490 (milliseconds)ms to commit in a new block. However, there is a clear pattern that the EncPvtData TXs for both asset generation and sharing take

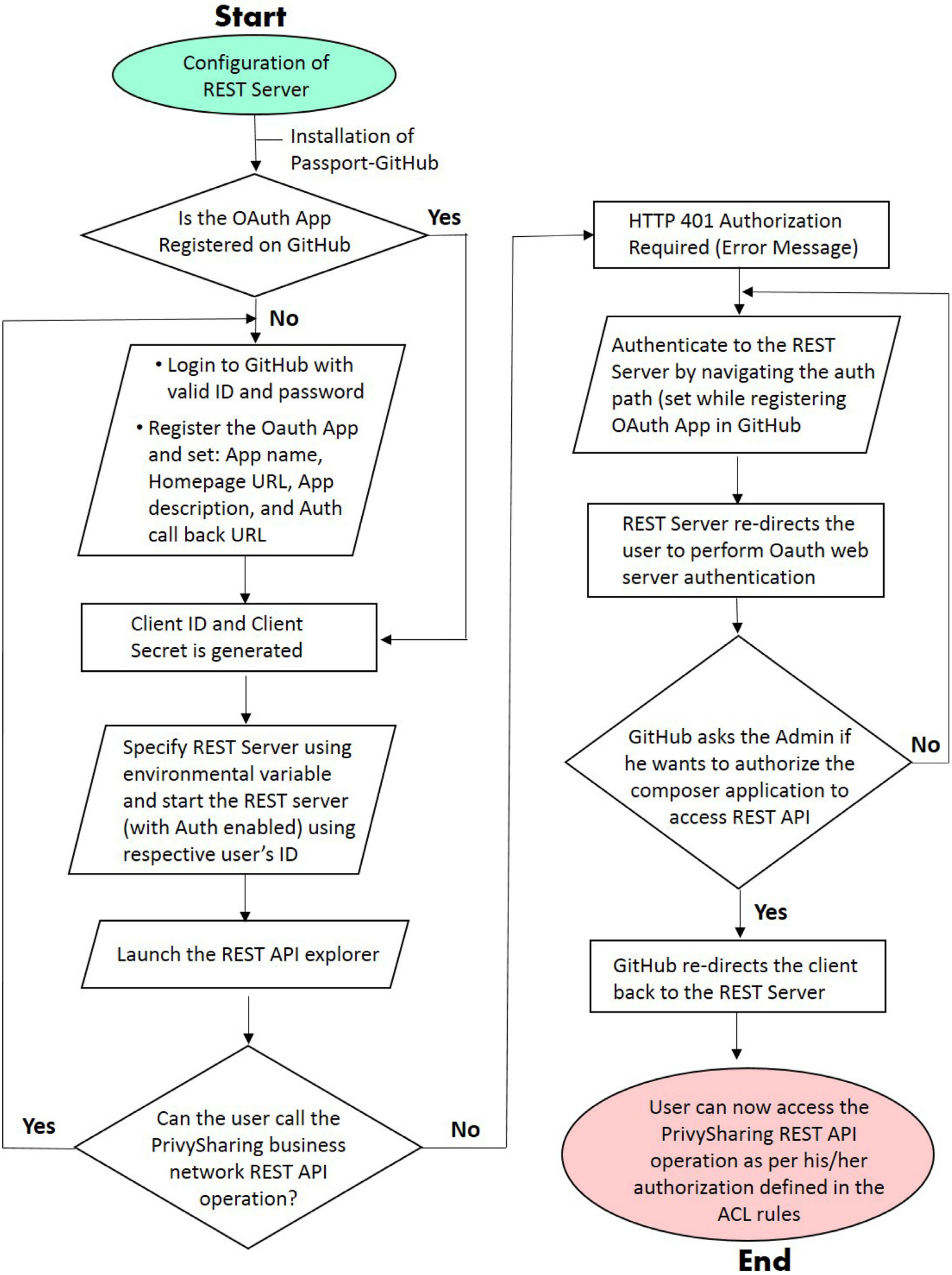
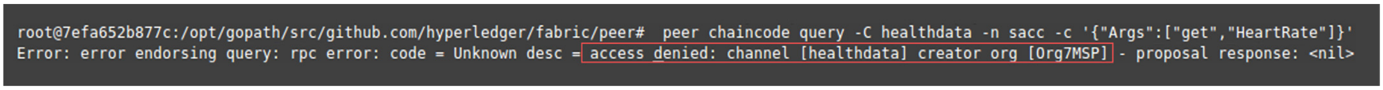

Start
Configuration of REST Server
Installation of Passport-GitHub
Is the OAuth App Registered on GitHub
Yes
No
• Login to GitHub with valid ID and password
• Register the Oauth App and set: App name, Homepage URL, App description, and Auth call back URL
Client ID and Client Secret is generated
Specify REST Server using environmental variable and start the REST server (with Auth enabled) using respective user's ID
Launch the REST API explorer
Can the user call the PrivySharing business network REST API operation?
Yes
No
HTTP 401 Authorization Required (Error Message)
Authenticate to the REST Server by navigating the auth path (set while registering OAuth App in GitHub
REST Server re-directs the user to perform Oauth web server authentication
GitHub asks the Admin if he wants to authorize the composer application to access REST API
No
Yes
GitHub re-directs the client back to the REST Server
User can now access the PrivySharing REST API operation as per his/her authorization defined in the ACL rules
End

Fig. 11. PrivySharing REST server OAuth flowchart.



```
root@7efa652b877c:/opt/gopath/src/github.com/hyperledger/fabric/peer#  peer chaincode query -C healthdata -n sacc -c '{"Args":["get","HeartRate"]}'
Error: error endorsing query: rpc error: code = Unknown desc = access denied: channel [healthdata] creator org [Org7MSP] - proposal response: <nil>
```

Fig. 12. Access denied for out of Ch data query.

Table 3
Methodology to achieve PrivySharing objectives.

Ser	Factors Deriving the Objectives	Objectives	Methodology
Threats to User/Data Security in a Smart City Environment			
1.	User privacy (ID disclosure)	Reduce the possibility of users' real world ID disclosure	PKI (X.509 Certificates) based multiple IDs for users
2.	User data privacy	Data confidentiality at rest and in transit, prevent over data collection, controlled access to data as defined by the data owner	Data encryption, use of SSL/TLS for data security in transit, user-defined ACL rules, use of multiple Chs and private data collection within a Ch
3.	Single point of failure (from physical as well as trust point of view)	Distributed data storage and decentralized control	Hyperledger-Fabric Blockchain
4.	False injection of data	Prevent data injection by unauthorized users	ID management, authentication and participation of only authorized nodes in the network. Moreover, TX initiation rights given to data owners or the parties given delegated powers by the data owners
5.	Vulnerability to Sybil Attack	Prevent Sybil Attack	User ID management and TX initiation by authorized entities only as per ACL rules
6.	Lack of common security framework for heterogeneous IoT devices with different communication protocols and diverse hardware parameters	Provide a common platform to store data transmitted/received from the heterogeneous sensors, irrespective of their diverse hardware and communication technologies	Hyperledger-Fabric Blockchain
7.	Threats to data integrity (data forgery and manipulation)	Preserve user data integrity	User authentication and restricted privileges to update user data, and blockchain's inherent data integrity protection
8.	Threats to smart city applications	Protect applications against the escalation of privileges and alteration attacks	Use of smart contracts based DApps
10.	Scalability	Contain the size of the blockchain	Use of blockchain to store TX logs only, whereas a world state is used to store user data updated states
11.	TX Latency and Throughput	More TX throughput with less latency	Use of multi-Ch blockchain as compared to a single-Ch blockchain
Essential GDPR Requirements for User Data Security			
1.	Personal data to be processed only with data owner's consent	The data owner is in complete control of his data, transparency of the complete process, visibility of all security and data access control changes	Chaincode-based user data access control rules, maintaining, and disseminating TX log on the need to know basis (Only a data owner or an authorized entity can see the TX log of a specific asset) and data sharing TX can only be initiated by the data owner
2.	Privacy by design	By default user data should be inaccessible to all, except those who are specifically allowed by the data owner	Access control rules deny everyone to see other's profile and assets unless explicitly shared by the data owner
3.	Commissioned data processing (i.e., data collection and processing as per the contract between the data owner and other parties)	A contract-based user data sharing that should conform to the contractual obligations	Business logic is transformed into Smart Contracts for secure and efficient data sharing as per contractual obligations
4.	Data owner should have access to all the information concerning his data (i.e., where is it stored, who has access to it, and for how long)	A transparent system, where data owner has complete visibility of the process and should be able to see and control the access to his data	User-defined data access control, and TX log management
5.	Right to forget, i.e., user data to be erased when no longer required	The system should allow user data deletion after a specific time, when the contract between the user and a third party expires, or when data is outdated or no longer required. Hence, there should be some distinction between TX log maintenance and user data storage. Such that even if user data is deleted, we are still able to verify the integrity of the past data	The world state is distinct from the blockchain. Hence, data/asset owner can delete user data from the world state without affecting TX log history
6.	Transparency	The system should be transparent, i.e., log all the activities concerning users' data (when and who modified the access control policies for data and updated the data itself)	TX log management and event notification

more time to commit than the PvtData and PlainText TXs. Moreover, the time taken by an asset sharing TX is lower than the asset generation/creation TX in almost all three cases. Similarly, Fig. 18 highlights the average (avg) time taken for state validation, block commit, and state commit for asset generation and asset sharing TXs with SOLO and Kafka consensus both. It can be ascertained that the time taken for block commit (represented by rust strip) in all three cases, i.e., EncPvtData, PvtData, and PlainText TXs, does not show many variations. However, the state commit time (expressed in the grey strip) significantly reduces for the PlainText TXs with SOLO and Kafka consensus in both cases, i.e., asset generation and asset sharing TXs. Similarly, the overall TX commit time for a plain text TX is lower than the EncPvtData and PvtData TXs.

In the second phase of the experiment, we measured various performance indicators of PrivySharing using Hyperledger Caliper as per the settings shown in Table 4. For the initial test, we ran thirty rounds of the experiment for both one-Ch and three-Ch scenarios with Kafka ordering service (consensus). There were six peers and six clients operating in the one-Ch and two peers and

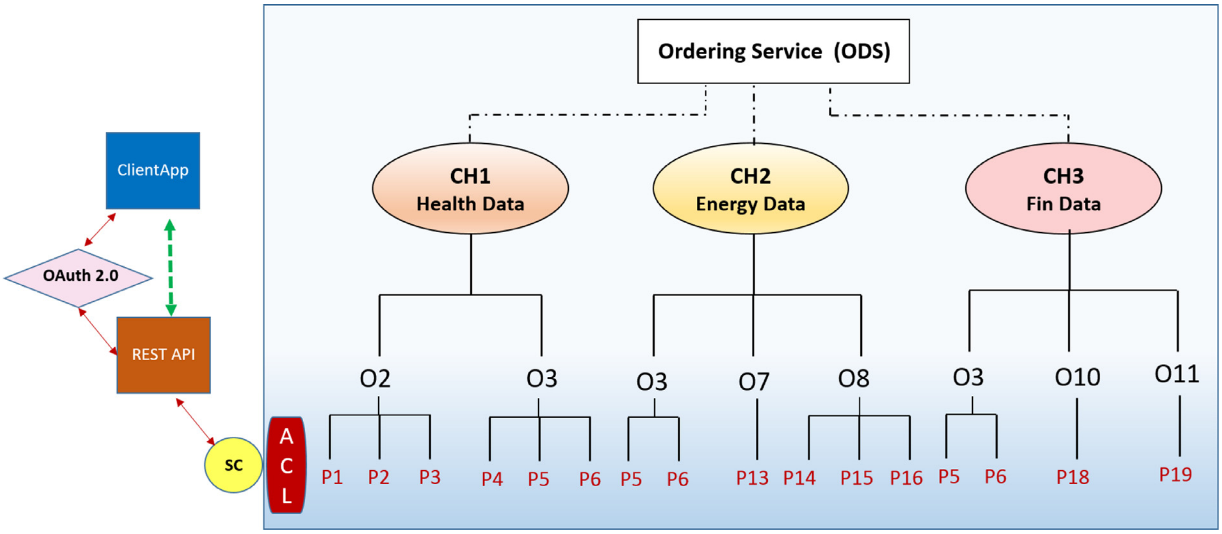
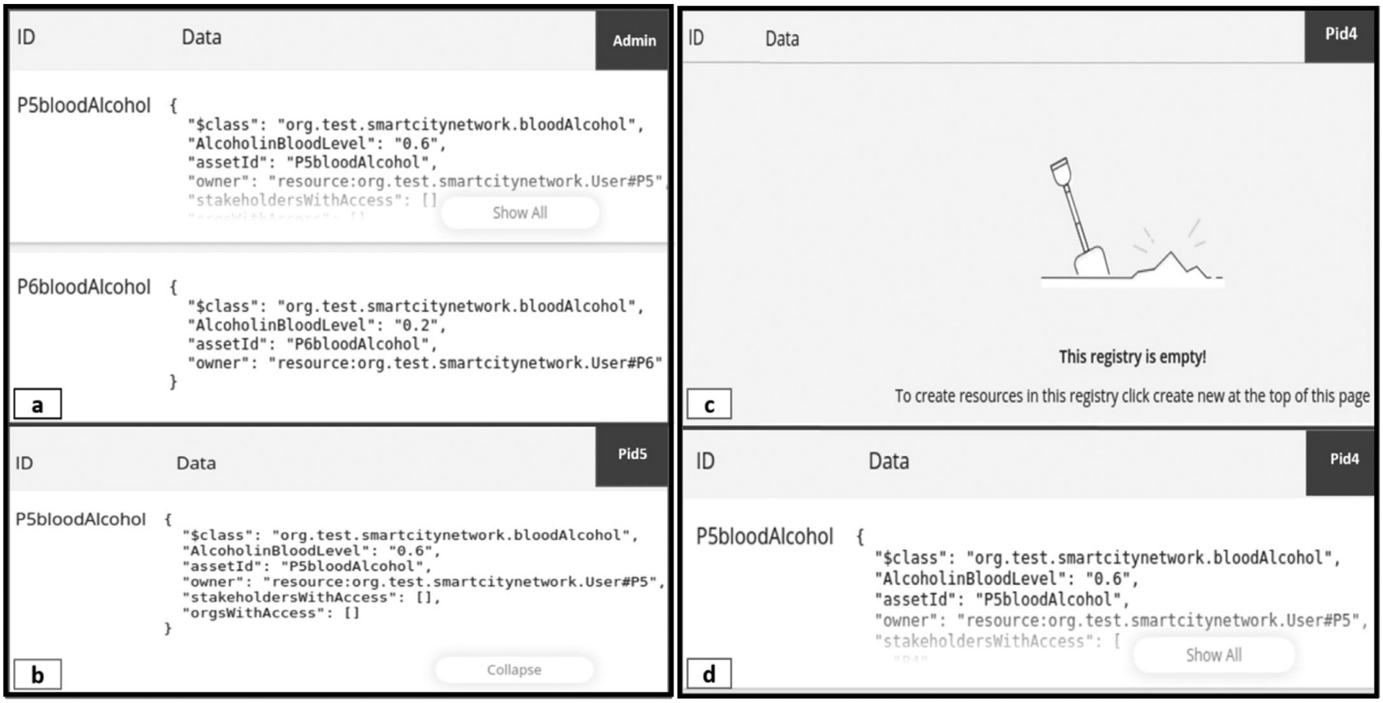


Fig. 13. Experimental Settings Phase-1.



a

```
P5bloodAlcohol  {
    "$class": "org.test.smartcitynetwork.bloodAlcohol",
    "AlcoholinBloodLevel": "0.6",
    "assetId": "P5bloodAlcohol",
    "owner": "resource:org.test.smartcitynetwork.User#P5",
    "stakeholdersWithAccess": []
    "orgsWithAccess": []
```

```
P6bloodAlcohol  {
    "$class": "org.test.smartcitynetwork.bloodAlcohol",
    "AlcoholinBloodLevel": "0.2",
    "assetId": "P6bloodAlcohol",
    "owner": "resource:org.test.smartcitynetwork.User#P6"
  }
```

b

```
P5bloodAlcohol  {
    "$class": "org.test.smartcitynetwork.bloodAlcohol",
    "AlcoholinBloodLevel": "0.6",
    "assetId": "P5bloodAlcohol",
    "owner": "resource:org.test.smartcitynetwork.User#P5",
    "stakeholdersWithAccess": [],
    "orgsWithAccess": []
  }
```

c

This registry is empty!

To create resources in this registry click create new at the top of this page

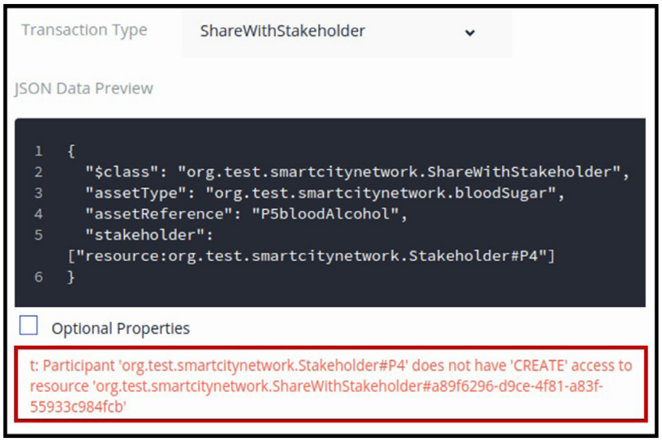
d

```
P5bloodAlcohol  {
    "$class": "org.test.smartcitynetwork.bloodAlcohol",
    "AlcoholinBloodLevel": "0.6",
    "assetId": "P5bloodAlcohol",
    "owner": "resource:org.test.smartcitynetwork.User#P5",
    "stakeholdersWithAccess": [
      "P4"
```

Fig. 14. Validation of assets access control.

Table 4
Experimental settings Phase-2.

Parameters	Settings for One CH Scenario	Settings for Three CHs Scenario
Number of Chs	01	03
Number of Input TXs	300	300
TX Send Rate	50 tps	50 tps
Number of Member Organizations	6	6
Peers Per Ch	6	2
Total Peers	6	6
Number of Orderer Nodes	4 Kafka Nodes 3 Zookeeper Nodes	4 Kafka Nodes 3 Zookeeper Nodes
Number of Clients	6	6
Number of Experiment Rounds	30	30



Transaction Type ShareWithStakeholder

JSON Data Preview

```
{
  "$class": "org.test.smartcitynetwork.ShareWithStakeholder",
  "assetType": "org.test.smartcitynetwork.bloodSugar",
  "assetReference": "P5bloodAlcohol",
  "stakeholder":
["resource:org.test.smartcitynetwork.Stakeholder#P4"]
}
```

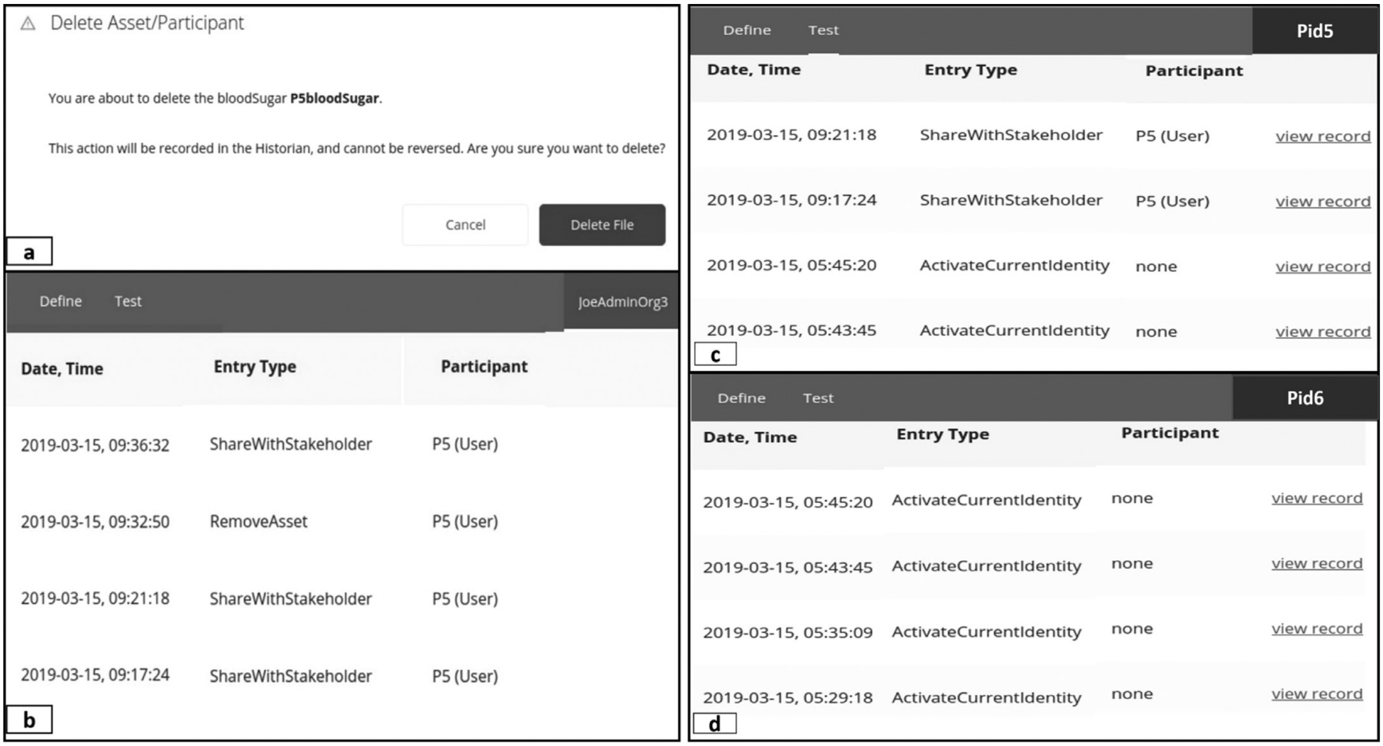
□ Optional Properties

t: Participant 'org.test.smartcitynetwork.Stakeholder#P4' does not have 'CREATE' access to resource 'org.test.smartcitynetwork.ShareWithStakeholder#a89f6296-d9ce-4f81-a83f-55933c984fcb'

Fig. 15. Validation of TX initiation rights.

two clients per Ch in the three-Ch scenario. Total three hundred TXs were input to the system at the rate of fifty tps (Transactions per second), in both scenarios. The highlight of this experiment as shown in Fig. 19 is that the three Ch scenario has demonstrated efficient performance contrary to the single Ch scenario, with an avg throughput of 42.4 tps and avg latency of 1.54 s at the TX Send Rate of 50 tps. After this primitive comparison, we also determined the p-values (Hypothesis Testing, 2019; Salkind, 2010), for both the scenarios to substantiate our findings. In that, we first applied independent two-sample T-test on latency measurements to determine the p-value to accept or reject the null hypothesis, i.e., “The average latency of the one-Ch network is equal to the average latency of the three-Ch network.” Whereas, the alternative hypothesis is; “The average latency of the one-Ch network is greater than the average latency of the three-Ch network.” The p-value resulted from the first test on system latency was 8.62×10^{-31}, which is less than 0.05. The result suggests the rejection of the null hypothesis in favor of the alternative hypothesis. Therefore, it is more probable that the average latency of the one-Ch network is higher than the average latency of the three-Ch system. Later, The second two-sample T-test was performed over throughput values. The null hypothesis in this case was; “The average throughput of the one-Ch network is equal to the average throughput of the three-Ch network.” Whereas, the alternative hypothesis states that “The average throughput of the one-Ch network is less than the average throughput of the three-Ch network.” The p-value emanated from this proceeding was 1.23×10^{-28}, which is smaller than 0.05. Hence, the result asserts the rejection of the null hypothesis in favor of the alternative hypothesis. Therefore, it is much likely that the average throughput of the one-Ch network is smaller than the average throughput of the three-Ch system. Hence, based upon the p-values, it can be concluded that the one-Ch network has inferior performance in terms of high latency and low throughput as compared to the three-Ch network.

In the third phase of the performance testing, we mapped the correlation between different performance indicators for the three-Ch network. TX Send Rate was pitched against network latency and throughput, as per the test settings shown in Table 5. The experiment was run for ten rounds with varying TX Send Rate in each round. Although we had set specific TX Send Rate for the test case, however, the actual Send Rate that was executed by the system came out to be different. There were two peers, and two clients in each Ch to process and submit the TXs, respectively. Fig. 20(a), interprets the relationship between TX Send Rate and network latency. The avg latency increases uniformly until the TX Send Rate reaches around 106 tps. After that, the latency starts fluctuating between 3 and 4 s. Correspondingly, Fig. 20(b) also highlights a similar trend, in which the network throughput rises with the increase in the TX Send Rate. However, once TX Send Rate reaches 106, the throughput waffles between 50 and 56 tps. We



Delete Asset/Participant

You are about to delete the bloodSugar **P5bloodSugar**.

This action will be recorded in the Historian, and cannot be reversed. Are you sure you want to delete?

Date, Time	Entry Type	Participant
2019-03-15, 09:36:32	ShareWithStakeholder	P5 (User)
2019-03-15, 09:32:50	RemoveAsset	P5 (User)
2019-03-15, 09:21:18	ShareWithStakeholder	P5 (User)
2019-03-15, 09:17:24	ShareWithStakeholder	P5 (User)

Date, Time	Entry Type	Participant	
2019-03-15, 09:21:18	ShareWithStakeholder	P5 (User)	view record
2019-03-15, 09:17:24	ShareWithStakeholder	P5 (User)	view record
2019-03-15, 05:45:20	ActivateCurrentIdentity	none	view record
2019-03-15, 05:43:45	ActivateCurrentIdentity	none	view record

Date, Time	Entry Type	Participant	
2019-03-15, 05:45:20	ActivateCurrentIdentity	none	view record
2019-03-15, 05:43:45	ActivateCurrentIdentity	none	view record
2019-03-15, 05:35:09	ActivateCurrentIdentity	none	view record
2019-03-15, 05:29:18	ActivateCurrentIdentity	none	view record

Fig. 16. Historical record of purged data asset and visibility of TX history.

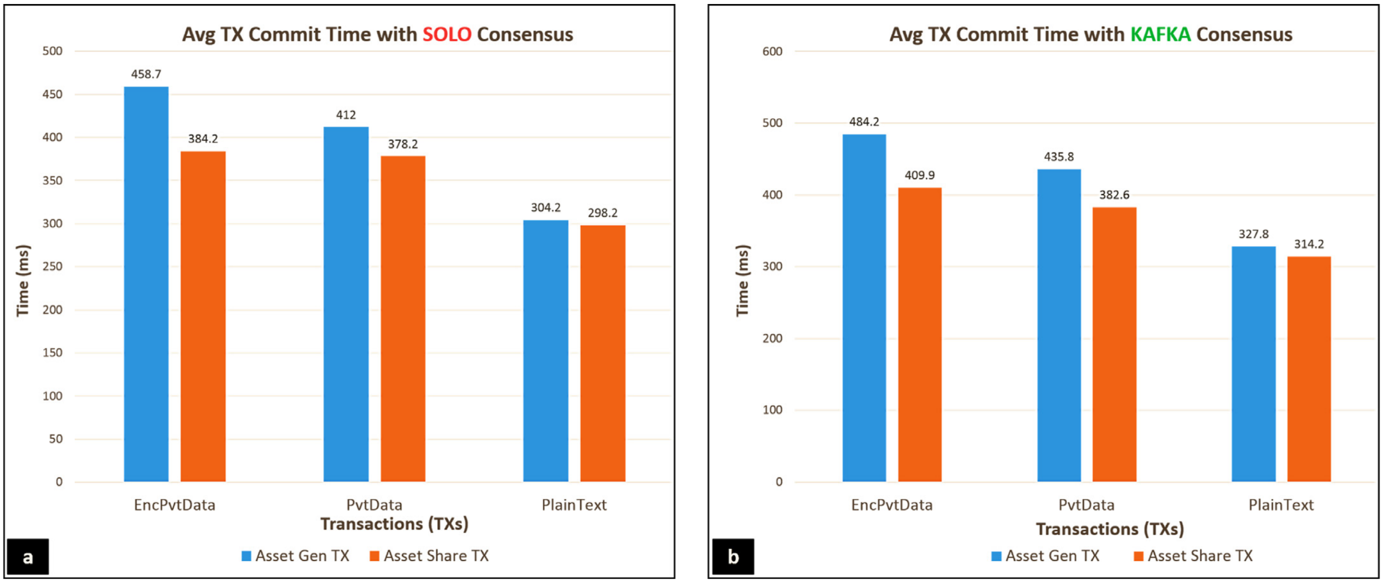


Fig. 17. Avg TX Commit Time.

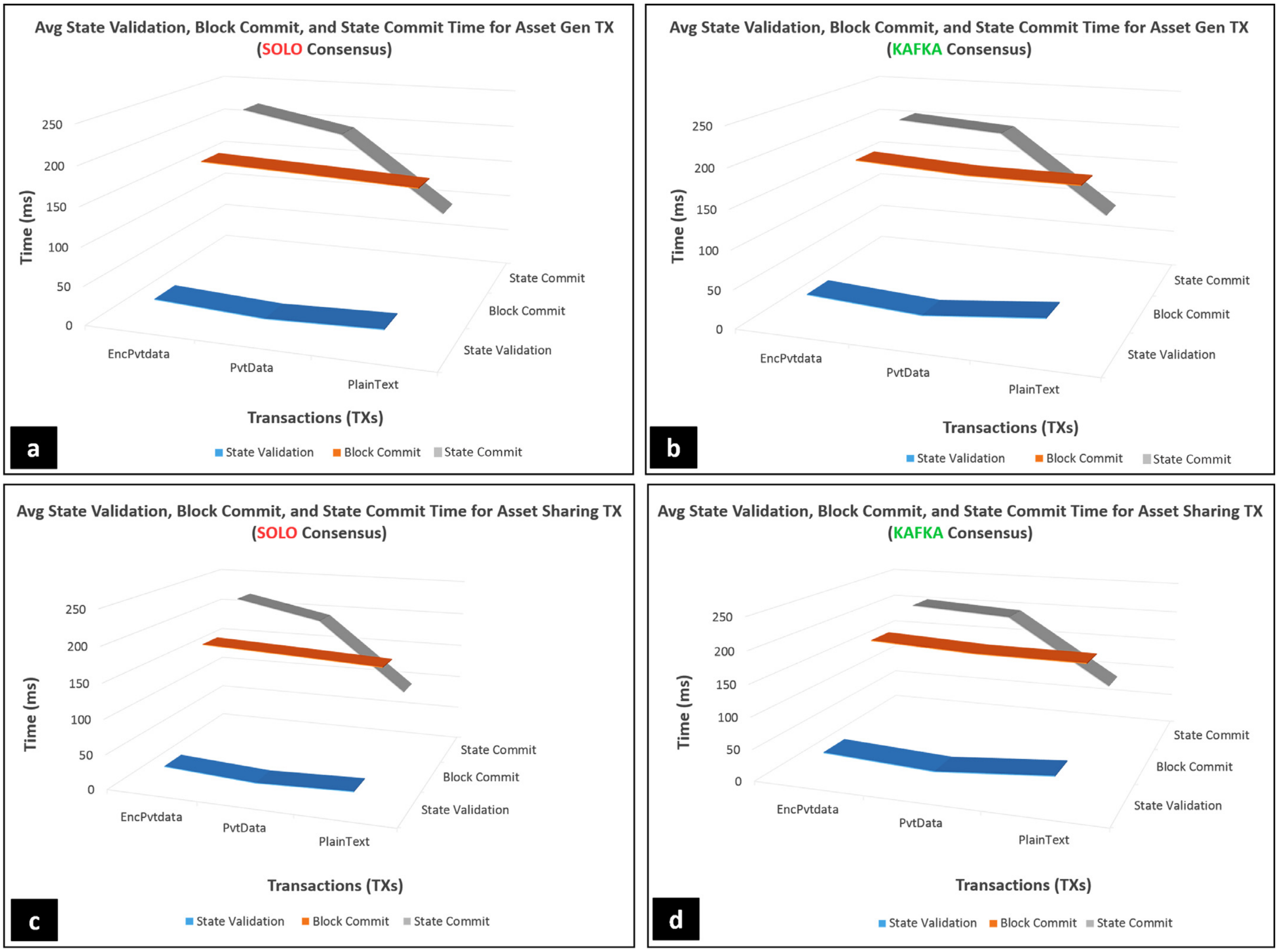


Fig. 18. Comparison of State validation, Block commit, and State commit avg time.

Table 5
Experimental settings Phase-3.

Parameter	Settings
Number of Chs	03
Number of Input TXs	300
TX Send Rate (configured) for Ten Rounds (tps)	25, 50, 75, 100, 125, 150, 175, 200, 225, 250
TX Send Rate (actual) for Ten Rounds (tps)	24.4, 47.4, 70.9, 89.6, 106.4, 116.3, 145.8, 154.3, 198.2, 199.1
Number of Member Organizations	6
Peers Per Ch	2
Total Peers	6
Number of Orderer Nodes	4 Kafka Nodes 3 Zookeeper Nodes
Number of Clients Per Ch	2
Total Clients	6
Number of Experiment Rounds	10

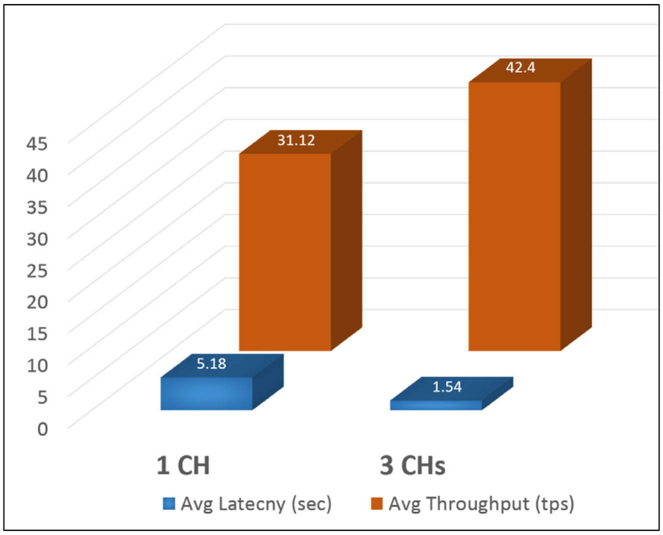


Fig. 19. Comparison of Avg Latency and Avg Throughput in One-Ch and Three-Ch Scenario.

believe that such a result is induced by the small number of orderer nodes, which could not handle more than 200 tps. Likewise, the latency in TX confirmation increases with the rise in TX Send Rate.

Later, we also studied the correlation between an increase in the number of peers and avg latency, and throughput respectively at varying TX send rates (as shown in Fig. 21(a)–(c)). For this test, there were six clients, and the number of peers varied from 6 to 24 in an increment of 6. It is observed that the throughput is mostly consistent with the send rate until the number of peers goes beyond 18. It can also be seen in Fig. 21(c) that the throughput decreases notably as the number of peers reaches 24. Similarly, the latency also increases with the increase in the number of peers. Such a behavior can be attributed to the number of endorsing and orderer nodes in the network that have to endorse and pack the TXs in the blocks and broadcast new blocks, respectively. Moreover, it can also be accredited to the fact that for this experiment all the peers were run on a single machine in a constrained environment. Hence, once in distributed setting, each peer is expected to perform much better. It is also believed that the TX throughput can be scaled by load balancing TX endorsement across a pool of endorsers (Ferris, 2019).

The experimental results uphold the idea of a multi-Ch blockchain network, as the same has demonstrated more throughput and less latency than the one-Ch system. The network latency and throughput in Hyperledger-Fabric depend upon numerous factors, such as, application design, fabric network architecture, specifications of endorsement policies, complexities of ACL rules, application/chaincode language, number of endorsers and ordering nodes, the batch timeout, and the physical or the virtual network infrastructure (Ferris, 2019). Hence, a meticulously designed and laid out blockchain network and application can yield higher TX throughput with less latency. E.g., FabCoin built on top of Hyperledger-Fabric can achieve a throughput of over 3560 tps with Kafka ordering service (Androulaki et al., 2018).

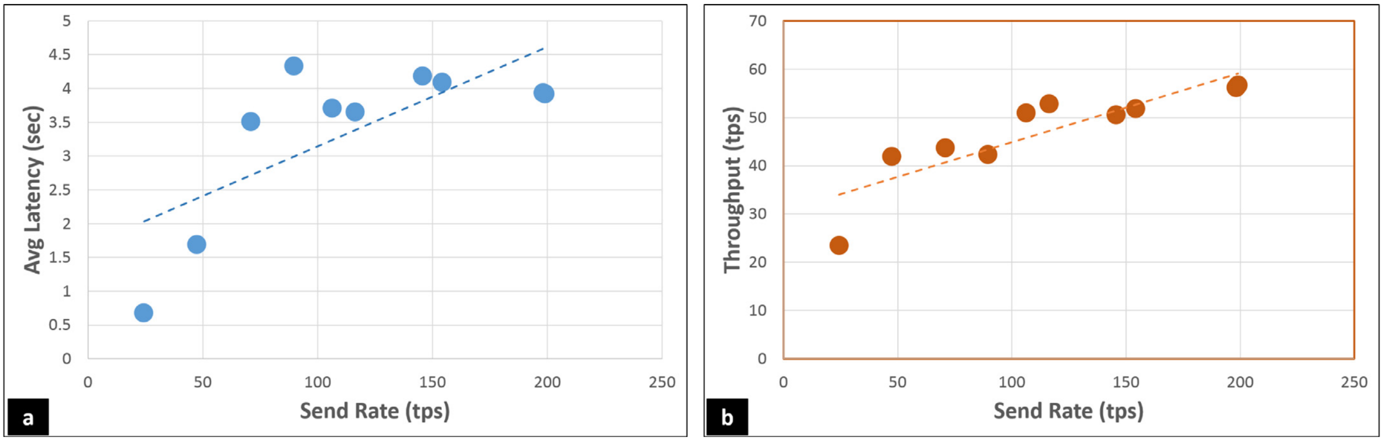


Fig. 20. a) Correlation between TX send rate and latency. b) Relation between TX send rate and network throughput.

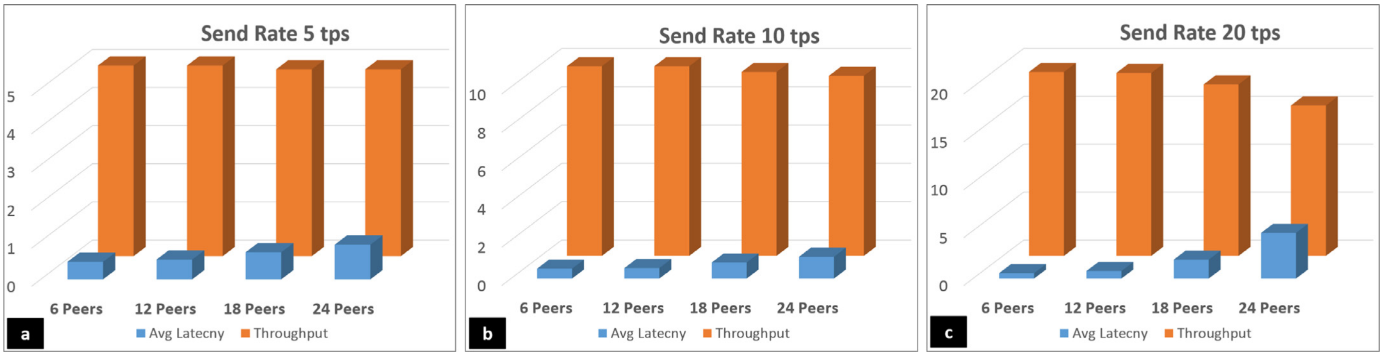


Fig. 21. Correlation between number of peers and network throughput at the send rate of (a) 5 tps, (b) 10 tps, and (c) 20 tps.

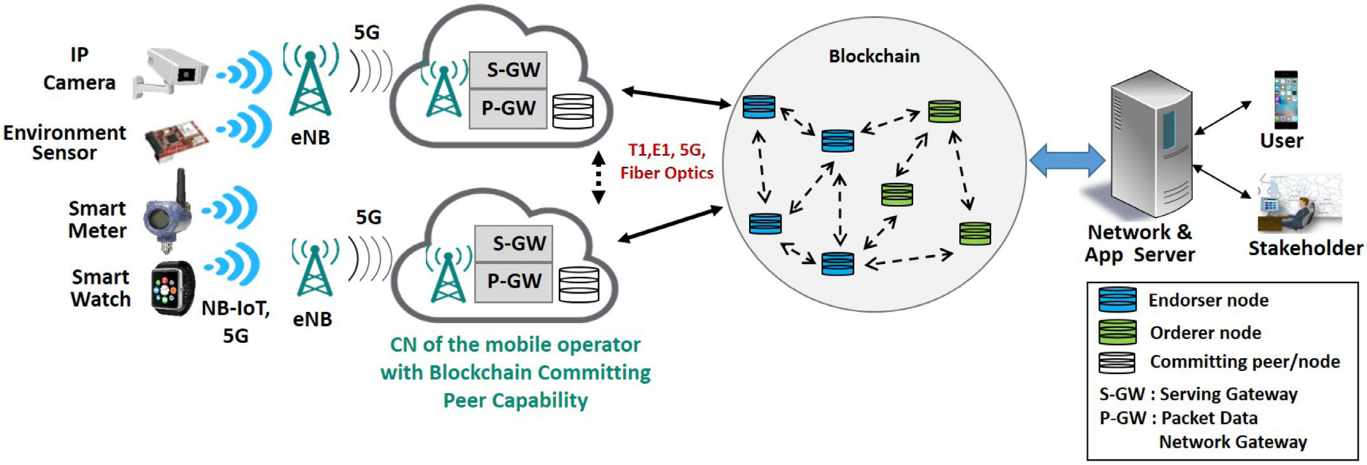


Fig. 22. Integration of Blockchain with MEC.

4.3. Limitations and open challenges

4.3.1. Multiple ledger storage by the peers

The use of multiple data specific Chs is presumed to be scalable than a single Ch. However, since committing peers have to maintain numerous ledgers, there may be a massive resource requirement for such nodes in a vast smart city network.

4.3.2. IoT device integrity

Electronic equipment, once connected to the internet is vulnerable to cyber-attacks. Resultantly, a hacker can hack into the electronic device and install malware, or modify the software or hardware components to alter the legitimate operation of the device (Dunn, 2019; Kumar et al., 2016; Sadeghi et al., 2015). Moreover, if an electronic device is physically compromised, the attacker can also change the hardware components, i.e., extend device memory, increase RAM, increase or decrease processor speed, change network configuration, activate or deactivate unauthorized ports or interfaces (JTAG, UART etc), and change I/O (input/output) pins configuration (Hernandez et al., 2014; Rostami et al., 2014; Wurm et al., 2016). Moreover, an electronic device malfunction can be caused by an unintentional or unprovoked technical fault, hardware or software failure, and a human error. Existing methods of device integrity check are based on code or memory attestation. These methods do not protect against physical compromise of the device, and modification of hardware or software components (Makhdoom et al., 2019). Contrarily, IoT data being the essential element to provide various seamless services in a smart city environment necessitate that the device initially generating and processing that data should be credible, i.e., only a legitimate and clean device should be able to input data to the blockchain. Whereas, currently, there is no plausible mechanism to test the integrity of the IoT devices at run time.

4.4. A way forward to address the limitations

4.4.1. Alternative to ledger storage by the peers

The concept of integrating edge computing into the mobile network architecture is not new (Hu et al., 2015). Thereafter, researchers are exploring the idea of using MEC (Mobile Edge Computing) as a gateway for IoT devices to achieve low latency, data aggregation, processing, and real-time application response (Abdelwahab et al., 2016; Salman et al., 2015; Sun and Ansari, 2016). The deployment models of MEC range from SCC (Small Cell Cloud) (FP7 European Project, 2012; Lobillo et al., 2014) to MMC (Mobile Micro Cloud) (Wang et al., 2013), MobiScud (Fast Moving Personal Cloud) (Wang et al., 2015), FMC (Follow Me Cloud) and etc. In all these MEC concepts, the first point of contact between the UE (User Equipment) and the mobile network is SCeNB (Small Cell evolve NodeB) or eNB (evolve NodeB). However, depending upon the MEC architecture the computational and storage resources are located (can be in hardware or virtual form) at SCeNB/eNB for SCC and MCC, and at distributed CN (Core Network) in the case of FMC. However, FMC with decentralized control and distributed architecture is the preferred choice over SCC and MCC (Mach and Becvar, 2017). We believe that based on the edge computing concept we can integrate blockchain with MEC to relieve end nodes from maintaining many ledgers. In this context, the SCeNB/eNB or CN nodes (in case of FMC architecture) can be harnessed with a suitable blockchain platform to facilitate fast TX settlement and provision of swift data processing and analytics services. Moreover, the end nodes can send queries for data (authorized to them) to the MEC nodes. To realize this concept, we propose a solution based on the FMC model as shown in Fig. 22. As of today, almost every inch of a populated area has cellular coverage, and most of the latest IoT devices also support NB-IoT technology. NB-IoT is a sub LTE frequency band, and in the near future, all the telcos (telecommunications companies) will be able to provide NB-IoT services. Moreover, the launch of 5G mobile network technology is also imminent. Hence, IoT devices can send sensor data to the MEC nodes via NB-IoT/5G. The MEC nodes being resourceful in terms of infrastructure, computational power, storage, and energy can also act as a blockchain committing peer. In this way, we can utilize the existing infrastructure of MEC/cellular networks without incurring excessive costs. The MEC node can then communicate with the endorsing nodes/peers using backhaul network (5G, E1, T1, fiber optics, satellite, etc.) and existing infrastructure at any distance. The inherent communication security of fiber optics, NB-IoT (Makhdoom et al., 2019), 4G, and 5G (Ahmad et al., 2018; Ferrag et al., 2018) technology will also add another layer of security over the blockchain P-2-P communication.

Turning a MEC node into a blockchain committing peer will be safe from the data security point of view, as the committing peers do not install and run the SCs. Hence, the SC TX logic will not be visible to them. Moreover, to incentivize the cellular companies for their services, they can be paid some TX fee as a reward in terms of the local digital token, e.g., PrivyCoin. Another advantage of integrating blockchain with MEC model will be ease in mobility management (e.g., handover) of end nodes/user devices if they move throughout the network.

4.4.2. Secure IoT device integration

The first element in IoT device security measures is device enrolment, in which only approved devices should be allowed to communicate with the blockchain and call smart contract methods (Makhdoom et al., 2018a). Secondly, all the unnecessary ports on the device should be blocked such as JTAG and UART since any open port can be used by an adversary to access the device and make malicious changes. Finally, most of the commercially available IoT devices such as wireless sensors, do not have a secure execution environment amid low costs. Therefore, the device integrity check should frequently be performed to ensure its legitimacy.

5. Conclusions and future work

User data generated by today's smart devices ranging from smart watches to smart cars, smart homes, auto-pay systems, ITS, etc., are vulnerable to privacy and security threats. Moreover, users also reserve the right to manage and control access to the data they own. Therefore, in this paper we introduced “PrivySharing”, an innovative blockchain-based secure and privacy-preserving data sharing mechanism for smart cities. The proposed strategy ensures that personal/critical user data is kept confidential, securely processed and is exposed to the stakeholders on the need to know basis as per user-defined ACL rules embedded in smart contracts. Moreover, the data owners are rewarded for sharing their data with the stakeholders/third parties. PrivySharing also complies with some of the fundamental EU GDPR requirements, such as data asset sharing, accessibility and purging with data owner's consent. In addition, the experimental results verified that a multi-Ch blockchain solution scales better than a single Ch blockchain system.

Though we have presented all the details of the proposed network architecture and security mechanism, however, as a PoC for this paper, we implemented a part of it. In the future, we aim to extend this work and incorporate the concept of the fog nodes based on existing mobile BTS stations and also devise a mechanism for secure integration of IoT devices with the blockchain network.

Declaration of Competing Interest

The authors declare that they do not have any financial or non-financial conflict of interests.

Acknowledgements

We would like to acknowledge Food Agility CRC Ltd for the award of a postgraduate scholarship. Food Agility CRC Ltd is a Co-operative Research Centre funded under the Commonwealth Government CRC program. The CRC Program supports industry-led collaboration between industry, researchers and the community.

References

Abdelwahab, S., Hamdaoui, B., Guizani, M., Znati, T., 2016. Replisom: disciplined tiny memory replication for massive IoT devices in LTE edge cloud. IoT J. 3 (3), 327–338.

Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M., Gurtov, A., 2018. Overview of 5G security challenges and solutions. Commun. Standards Mag. 2 (1), 36–43.

Anderson, J.C., Lehnardt, J., Slater, N., 2010. CouchDB: The Definitive Guide: Time to Relax. O'Reilly Media, Inc.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al., 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the 13th EuroSys Conference. ACM, pp. 1–15.

Apthorpe, N., Reisman, D., Feamster, N., 2017. A smart home is no castle: privacy vulnerabilities of encrypted iot traffic. arXiv:1705.06805, 1–6.

Bartoli, A., Hernández-Serrano, J., Soriano, M., Dohler, M., Kountouris, A., Barthel, D., 2011. Security and Privacy in your Smart City. In: Proceedings of the Barcelona Smart Cities Congress, 292, pp. 1–6.

Biswas, K., Muthukkumarasamy, V., 2016. Securing Smart Cities using Blockchain Technology. In: Proceedings of the 14th International Conference on Smart City High Performance Computing and Communications. IEEE, pp. 1392–1393.

Buterin, V., et al., 2014. A Next-generation Smart Contract and Decentralized Application Platform. Whitepaper.

Cui, L., Xie, G., Qu, Y., Gao, L., Yang, Y., 2018. Security and privacy in smart cities: challenges and opportunities. IEEE Access 6, 46134–46145.

Dent, A., 2013. Getting Started with LevelDB. Packt Publishing Ltd.

Dorri, A., Kanhere, S., Jurdak, R., Gauravaram, P., 2017. Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. In: Proceedings of the 2nd Workshop on Security, Privacy, and Trust in the Internet of Things (PERCOM). IEEE, pp. 1–6.

Dorri, A., Kanhere, S. S., Jurdak, R., 2016. Blockchain in internet of things: challenges and solutions. arXiv:1608.05187.

Dorri, A., Kanhere, S.S., Jurdak, R., 2019. Mof-bc: a memory optimized and flexible blockchain for large scale networks. Future Gener. Comput. Syst. 92, 357–373.

Dunn, J. E., 2016. Krebs DDoS aftermath: industry in shock at size, depth and complexity of attack. https://www.computerworld.com/article/3427227/krebs-ddos-aftermath--industry-in-shock-at-size--depth-and-complexity-of-attack.html.

Faber, B., Michelet, G.C., Weidmann, N., Mukkamala, R.R., Vatrapu, R., 2019. BPDIMS: a blockchain-based personal data and identity management system. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 6855–6864.

Ferrag, M.A., Maglaras, L., Argyriou, A., Kosmanos, D., Janicke, H., 2018. Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes. J. Netw. Comput. Appl. 101, 55–82.

Ferris, C., 2019. Hyperledger Fabric performance and scale. https://www.ibm.com/blogs/blockchain/2019/01/answering-your-questions-on-hyperledger-fabricperformance-and-scale/.

FP7 European Project, 2012. Distributed Computing, Storage and Radio Resource Allocation Over Cooperative Femtocells (TROPIC), http://www.ict-tropic.eu/.

GDPR, 2018. General Data Protection Regulation, https://gdpr-info.eu/.

Gordon, W.J., Catalini, C., 2018. Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Comput. Struct. Biotechnol. J. 16, 224–230.

Haidar, F., Kaiser, A., Lonc, B., 2017. On the Performance Evaluation of Vehicular PKI Protocol for V2X Communications Security. In: Proceedings of the 86th Vehicular Technology Conference (VTC-Fall). IEEE, pp. 1–5.

Hardt, D., 2012. The OAuth 2.0 Authorization Framework. RFC 6749. RFC Editor. http://www.rfc-editor.org/rfc/rfc6749.txt.

Hernandez, G., Arias, O., Buentello, D., Jin, Y., 2014. Smart nest thermostat: a smart spy in your home. Black Hat USA (2014).

Hu, Y.C., Patel, M., Sabella, D., Sprecher, N., Young, V., 2015. Mobile edge computing-A key technology towards 5G. ETSI white paper 11 (11), 1–16.

Huh, S., Cho, S., Kim, S., 2017. Managing IoT Devices using Blockchain Platform. In: Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT). IEEE, pp. 464–467.

Hyperledger-Fabric, 2019a. Blockchain Network, https://hyperledger-fabric.readthedocs.io/en/release-1.4/network/network.html.

Hyperledger-Fabric, 2019b. Smart Contracts and Chaincode, https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract/smartcontract.html.

Hyperledger-Fabric, 2019c. Identity, https://hyperledger-fabric.readthedocs.io/en/release-1.4/identity/identity.html.

Hypothesis Testing (P-Value Approach). 2019. https://onlinecourses.science.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing/pvalue-approach.

Jason, S., 2019. Hundreds of millions of Facebook user records were exposed on Amazon cloud server, (Last accessed 19 April 2019). https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/.

Kaaniche, N., Laurent, M., 2017. A blockchain-based data usage auditing architecture with enhanced privacy and availability. In: Proceedings of the 16th International Symposium on Network Computing and Applications (NCA). IEEE, pp. 1–5.

Kountché, D.A., Bonnin, J.-M., Labiod, H., 2017. The Problem of Privacy in Cooperative Intelligent Transportation Systems (C-ITS). In: Proceedings of the Computer Communications Workshops (INFOCOM WKSHPS). IEEE, pp. 482–486.

Krishnan, K.N., Jenu, R., Joseph, T., Silpa, M., 2018. Blockchain Based Security Framework for IoT Implementations. In: Proceedings of the International CET Conference on Control, Communication, and Computing (IC4). IEEE, pp. 425–429.

Kumar, S.A., Vealey, T., Srivastava, H., 2016. Security in internet of things: Challenges, solutions and future directions. In: Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 5772–5781.

Leiding, B., Memarmoshrefi, P., Hogrefe, D., 2016. Self-managed and Blockchain-based Vehicular Ad-hoc Networks. In: Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. ACM, pp. 137–140.

Lobillo, F., Becvar, Z., Puente, M.A., Mach, P., Presti, F.L., Gambetti, F., Goldhamer, M., Vidal, J., Widiawan, A.K., Calvanesse, E., 2014. An architecture for mobile computation offloading on cloud-enabled LTE small cells. In: Proceedings of the Wireless Communications and Networking Conference Workshops (WCNCW). IEEE, pp. 1–6.

Lund, D., MacGillivray, C., Turner, V., Morales, M., 2014. Worldwide and regional internet of things (IoT) 2014–2020 forecast: A Virtuous circle of proven value and demand. International Data Corporation (IDC), Tech. Rep, 2014..

Mach, P., Becvar, Z., 2017. Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. 19 (3), 1628–1656.

Makhdoom, I., Abolhasan, M., Abbas, H., Ni, W., 2018a. Blockchain'S adoption in IoT: the challenges, and a way forward. J. Netw. Comput. Appl. 125, 251–279.

Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R.P., Ni, W., 2019. Anatomy of threats to the internet of things. IEEE Commun. Surv. Tutor. 21 (2), 1636–1675.

Makhdoom, I., Abolhasan, M., Ni, W., 2018b. Blockchain for IoT: The Challenges and a Way Forward. In: Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT. INSTICC. SciTePress, pp. 428–439.

Marchesi, M., Marchesi, L., Tonelli, R., 2018. An agile software engineering method to design blockchain applications. In: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia. ACM, pp. 1–8.

Mazhelis, O., Hämäläinen, A., Asp, T., Tyrväinen, P., 2016. Towards Enabling Privacy Preserving Smart City Apps. In: Proceedings of the International Smart Cities Conference (ISC2). IEEE, pp. 1–7.

Michelin, R.A., Dorri, A., Steger, M., Lunardi, R.C., Kanhere, S.S., Jurdak, R., Zorzo, A.F., 2018. SpeedyChain: A Framework for Decoupling Data from Blockchain for Smart Cities. In: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. ACM, pp. 145–154.

Moustaka, V., Theodosiou, Z., Vakali, A., Kounoudes, A., 2018. Smart cities at risk!: privacy and security borderlines from social networking in cities. Athena 357, 905–910.

Neisse, R., Steri, G., Nai-Fovino, I., 2017. A blockchain-based approach for data accountability and provenance tracking. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ACM, pp. 1–10.

Pongnumkul, S., Siripanpornchana, C., Thajchayapong, S., 2017. Performance analysis of private blockchain platforms in varying workloads. In: Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, pp. 1–6.

Puthal, D., Nepal, S., Ranjan, R., Chen, J., 2016. Threats to networking cloud and edge datacenters in the internet of things. IEEE Cloud Comput. 3 (3), 64–71.

Qian, Y., Liu, Z., Yang, J., Wang, Q., 2018. A Method of Exchanging Data in Smart City by Blockchain. In: Proceedings of the 16th International Conference on Smart City. IEEE, pp. 1344–1349.

Rahman, M.A., Rashid, M.M., Hossain, M.S., Hassanain, E., Alhamid, M.F., Guizani, M., 2019. Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city. IEEE Access 7, 18611–18621.

Rantos, K., Drosatos, G., Demertzis, K., Ilioudis, C., Papanikolaou, A., Kritsas, A., 2018. ADvoCATE: A Consent Management Platform for Personal Data Processing in the IoT Using Blockchain Technology. In: Proceedings of the International Conference on Security for Information Technology and Communications. Springer, pp. 300–313.

Rostami, M., Koushanfar, F., Karri, R., 2014. A primer on hardware security: models, methods, and metrics. Proc. IEEE 102 (8), 1283–1295.

Roulin, C., Dorri, A., Jurdak, R., Kanhere, S., 2018. On the activity privacy of blockchain for iot. arXiv:1812.08970, 1–8.

Sadeghi, A.-R., Wachsmann, C., Waidner, M., 2015. Security and privacy challenges in industrial Internet of Things. In: Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6.

Salkind, N.J., 2010. Student's t-Test. Encycl. Res. Des..

Salman, O., Elhajj, I., Kayssi, A., Chehab, A., 2015. Edge computing enabling the Internet of Things. In: Proceedings of the 2nd World Forum on Internet of Things (WF-IoT). IEEE, pp. 603–608.

Sara, S., 2018. A Google bug exposed the information of up to 500,000 users. (Last accessed 30 Dec 2018). https://www.cnbc.com/2018/10/08/google-bug-exposed-the-information-of-up-to-500000-users.html.

Sara, S., Michael, N., 2018. Facebook has been worried about data leaks like this since it went public in 2012, (Last accessed 11 September 2018). https://www.cnbc.com/2018/04/12/facebook-warned-of-data-breaches-years-ago-when-it-went-public-in-2012.html.

Sharma, P.K., Park, J.H., 2018. Blockchain based hybrid network architecture for the smart city. Future Gener. Comput. Syst. 86, 650–655.

Sun, X., Ansari, N., 2016. Edgeiot: mobile edge computing for the internet of things. IEEE Commun. Mag. 54 (12), 22–29.

Truong, N.B., Sun, K., Lee, G.M., Guo, Y., 2019. GDPR-Compliant Personal data management: a blockchain-based solution. IEEE Trans. Inf. Forensics Secur. 1–13.

Wang, K., Shen, M., Cho, J., Banerjee, A., Van der Merwe, J., Webb, K., 2015. Mobiscud: a fast moving personal cloud in the mobile network. In: Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges. ACM, pp. 19–24.

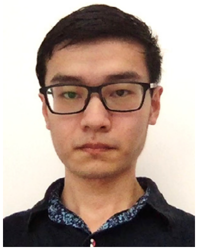
Wang, S., Tu, G.-H., Ganti, R., He, T., Leung, K., Tripp, H., Warr, K., Zafer, M., 2013. Mobile micro-cloud: application classification, mapping, and deployment,. In: Proceedings of the Annual Fall Meeting of ITA (AMITA), pp. 1–7.

Wurm, J., Hoang, K., Arias, O., Sadeghi, A.-R., Jin, Y., 2016. Security Analysis on Consumer and Industrial IoT Devices. In: Proceedings of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, pp. 519–524.

Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., Shen, X.S., 2017. Security and privacy in smart city applications: challenges and solutions. IEEE Commun. Mag. 55 (1), 122–129.

Zhang, Y., Wen, J., 2016. The IoT electronic business model: using blockchain technology for the internet of things. Peer-to-Peer Netw. Appl. 1–12.

Imran Makhdoom (S'18) Imran Makhdoom completed his B.E (Telecommunications Engineering) and Masters in Information Security from National University of Sciences and Technology, Pakistan in 2004 and 2015 respectively. Currently, he is doing research in the area of blockchain and IoT security at the University of Technology Sydney. He has published his research in some high ranking journals. He has also won numerous blockchain hackathons and research showcases. Imran has also made a valuable contribution in IoT security and has filed a patent with IP Australia concerning an innovative method of IoT device integrity check. Prior to this, from 2004 until 2009, he worked at Special Communications Organization on various networking projects based on HF/VHF/UHF wireless radios and CISCO proprietary equipment. In 2009, he accepted the position of Senior Project Manager in a Government Satellite Communications Organization and served till 2014. From 2014–2016 he also worked on various Information security related assignments and carried out security audits of various government organizations. He is an EC-Council certified Secure Computer User.



Ian Zhou is a Ph.D. candidate at the University of Technology Sydney. His research interest include blockchain, IoT security, and AI. Currently, he is doing research on "Multi Sensor-based Traffic incident prediction, detection and response through machine learning." Prior to this, he received the Masters degree in business administration from the University of Technology Sydney in 2018.

Mehran Abolhasan (S'01-M'03-SM'11) received the B.E. degree in computer engineering and the Ph.D. degree in telecommunications from the University of Wollongong in 1999 and 2003, respectively. He is currently an Associate Professor and the Deputy Head of the School of Electrical and Data Engineering, University of Technology Sydney. He has authored over 120 international publications and has won over $3 million in research funding. His current research interests are software-defined networking, IoT, wireless mesh, wireless body area networks, cooperative networks, 5G networks and beyond, and sensor networks.

Justin Lipman (S'94-M'04-SM'12) is an Associate Professor at the University of Technology Sydney focused on research and industry engagement for the Internet of Things, Industrial IoT, Intelligent Transport, Smart Cities and Food Agriculture. He received his Ph.D. Telecommunications and BE Computer Engineering from the University of Wollongong, Australia in 2003 and 1999 respectively. From 2004 to 2017, he was based in Shanghai, China and held a number of senior management and technical leadership roles at Intel and Alcatel leading research and innovation, product architecture and IP generation. Dr. Lipman has consulted for a number of startups and co-founded two startups. He is an IEEE senior member, with over 40 peer reviewed publications, more than 20 USPTO patents awarded and a further 20 USPTO patent submissions under review. Dr. Lipman is a committee member in Standards Australia contributing to Australian IoT standards. His research interests are in all things adaptive, connected, distributed and ubiquitous.

Wei Ni (M'09-SM'15) received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is currently a Team Leader with CSIRO, Sydney, Australia, and an Adjunct Professor with the University of Technology Sydney. He was a Post-Doctoral Research Fellow with Shanghai Jiaotong University from 2005 to 2008, the Deputy Project Manager of the Bell Labs R&I Center, Alcatel/Alcatel-Lucent from 2005 to 2008, and a Senior Researcher with Devices Research and Development, Nokia from 2008 to 2009. He also holds adjunct positions with the University of New South Wales and Macquarie University. His research interests include stochastic optimization, game theory, graph theory, as well as their applications to network and security. He has been serving as the Vice Chair of IEEE NSW VTS Chapter and Editor of IEEE Transactions on Wireless Communications since 2018, the Secretary of IEEE NSW VTS Chapter from 2015 to 2018, the Track Chair for VTC-Spring 2017, the Track Co-Chair for IEEE VTC-Spring 2016, and the Publication Chair for BodyNet 2015. He also served as the Student Travel Grant Chair for WPMC 2014, a Program Committee Member of CHINACOM 2014, and a TPC Member of IEEE ICC'14, ICCC'15, EICE'14, and WCNC'10.