Journal of Systems Architecture 135 (2023) 102802

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Smart contract-based security architecture for collaborative services in municipal smart cities

Shahbaz Siddiqui [a], Sufian Hameed [a], Syed Attique Shah [b,*], Abdul Kareem Khan [a], Adel Aneiba [b]

[a] Department of Computer Science, NUCES, Karachi, 75160, Pakistan
[b] School of Computing and Digital Technology, Birmingham City University, Millennium Point, Birmingham, B4 7XG, United Kingdom

ARTICLE INFO

Keywords:
Blockchain
IoT
Multichain
SDN
Smart city
Smart contracts
Smart municipal city

ABSTRACT

The Internet of Things (IoT) can provide intelligent and effective solutions to various applications with higher accuracy that requires less or no human intervention. Smart Cities are one of the significant applications of the IoT comprising a collection of various services such as intelligent transportation, waste management, smart homes, etc. These heterogeneous services offer a wide range of collaborative applications in smart cities. A smart municipality in a smart city is a concept in which a digital municipal corporation is developed to provide comprehensive local government collaboration services based on digitization and automation aiming towards raising the living standards of citizens. Interoperability between heterogeneous services for collaborative tasks creates challenges for data security and privacy. Ensuring integrity and confidentiality of information is critical, and reliable data is essential to both the government and its citizens. In this paper, we proposed a service security architecture based on authentication and authorization for constrained environments during collaborative tasks for Software Defined Networking (SDN) and smart contract-enabled municipal smart cities. The proposed collaborative service security framework is being tested on the Multichain Blockchain networks. We present a novel method for using smart contracts in multichain blockchains for data security during collaborative tasks in smart city municipal architecture. The proposed security solution is based on the dynamism of smart contracts to govern and control all interactions and transactions securely between different heterogeneous IoT networks. We implemented a supportive use case for collaborative services in an SDN-enabled IoT architecture to evaluate the feasibility of the proposed service security architecture.

Code metadata

Permanent link to reproducible Capsule: https://codeocean.com/capsule/7038031/tree/v1.

1. Introduction

Internet of Things (IoT) is making it possible to envision a world where practically every device around us is connected to others, allowing for improved living standards by providing smart services, intelligent analytics, and reliable communication. The collaboration of devices such as sensors, actuators, vehicles, cameras, etc creates a variety of access networks that automate data collection providing critical information when properly analysed. This information can help for efficiently managing infrastructures and resources available for fulfilling the complex data requirements of different other applications [1]. As various applications such as the automotive sector, medical science, municipal services, and other fields push towards IoT adoption, significant research growth, and technological innovations can recently be seen in the smart cities domain [2]. Many applications have strict real-time requirements for sharing information between devices and everything else. The goal is to make these interrelated systems smart and increase the quality of services offered to citizens and eventually improve their quality of life [3].

Smart cities offer a wide range of heterogeneous services aiming towards the improvement of the living standards of citizens. Smart city services are of significant interest to municipal governments and are being adopted to enhance the quality of life for residents of smart cities and to cut administrative costs by automating processes [4]. Local governments and municipalities can govern some fundamental administrative policies through smart cities aimed at providing

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.

E-mail addresses: shahbaz.siddiqui@nu.edu.pk (S. Siddiqui), sufian.hameed@nu.edu.pk (S. Hameed), syed.shah2@bcu.ac.uk (S.A. Shah), k20-1317@nu.edu.pk (A.K. Khan), Adel.Aneiba@bcu.ac.uk (A. Aneiba).

https://doi.org/10.1016/j.sysarc.2022.102802
Received 30 June 2022; Received in revised form 30 November 2022; Accepted 2 December 2022
Available online 13 December 2022

intelligent water management, intelligent street lighting, and smart waste management services. To make sure that all linked activities (such as monitoring, reporting, and interventions) of the municipal system in a smart city continue to work effectively, these systems need significant repairs to improve their collaborative management processes [5]. The main challenges for collaborative services in smart cities are related to security and privacy, including service provider trust, the reliability of sensed data, and data ownership, caused by the diversified co-operation of various services in a municipal smart system [6].

Blockchain technology providing distributed public ledger (DPL), is based on the concept of immutability and offers to secure the data cryptographically. Using blockchain technology, digital assets may be exchanged and stored without the involvement of a third party [7]. It uses a peer-to-peer architecture with a large number of nodes to reach a consensus on data blocks in an untrusted environment [8]. Smart cities may benefit from the immutability, transparency, and decentralization of transactions made possible by blockchain technology [9]. The most common uses of blockchain technology include the transfer of assets and digital applications through smart contracts, as well as the creation of distributed information records. This technology may eliminate the need for centralized storage of data schemes in different financial institutions such as banks and notaries, as well as government agencies and trade groups, and also provide automation in the process, ultimately saving time [10]. Businesses can benefit from the saved time and resources and give more focus to work on long-term business strategies [11]. A blockchain-based smart city with a distributed system faces a number of problems, such as the large communication costs that come with synchronizing transactions. Frequent peer-to-peer (P2P) node transactions make it hard to keep the state in sync. Blockchain technology could help smart cities by keeping a safe, open, decentralized, and unchangeable record of all transactions generated by different collaborative services [12,13].

Software-defined networking (SDN) is a networking architecture that allows feedback control to be logically centralized. With SDN, decisions are made by the "network brain", which has a view of the whole network, making it easier to optimize the network [14]. In SDN, the data plane elements become highly efficient and programmable packet forwarding devices, while the control plane elements are represented by a single entity called the controller [15]. In SDN, it is much easier to build and deploy applications than in traditional networks [16]. An SDN framework offers an efficient way to deal with blockchain problems because it can make it easier to share resources efficiently and also because it is a logically centralized architecture that helps keep all the authorized nodes in "sync" with each other globally [17]. The SDN architecture also gets rid of redundancy because the global controller does transactions only once, instead of each node doing the same transaction on its own [18,19]. SDN is being used more and more with other technologies such as the Internet of Things and blockchain [14,20]. The combination of SDN networking architecture with blockchain in smart cities allows the network to be managed centrally along with configurable functionalities [21].

1.1. Motivation

Integration of IoT, blockchain, and SDN holds a lot of promise for smart cities. With the ongoing evolution and rapid deployments of smart home and smart city concepts, thousands of disparate services are expected to form many collaborative services for common applications in future smart cities. The fact that these services working together raises a lot of questions and concerns about data sources, sharing of information, data integrity, data confidentiality, and privacy, among other issues, that need to be carefully thought of [33]. Through reviewing the available literature with regards to the research gaps identified and summarized in Table 1, it can be identified that in order to have secure and safe communication between different heterogeneous IoT networks focusing on collaborative tasks in smart cities and or smart municipal systems, there is a high need for,

(a) Scalable infrastructure that can adapt to diverse heterogeneous IoT networks' security requirements in a smart municipal system during the conduction of collaborative tasks.
(b) An adaptive security protocol for collaborative services in smart cities, capable of meeting the dynamic security requirements of numerous heterogeneous IoT services' interoperability in smart municipal system architecture.

The primary reason for merging SDN, IoT, and blockchain is to create an intelligent, easily controlled, and scalable system that can support billions of networked IoT devices. On one hand, due to SDN's centralized management and programmability features, SDN has emerged as a possible option for managing such a large number of IoT devices in the future, and on the other hand, Blockchain technology provides enhanced traceability to the overall deployment of the smart city.

1.2. Contributions

In this study, we present how smart contracts can be used to create a decentralized service security protocol for collaborative tasks in the smart municipal city concept. The main contributions of this paper are as follows,

(a) A novel architecture that describes the registration of IoT devices for heterogeneous networks and the secure communication of multiple heterogeneous networks during collaborative tasks in municipal smart cities.
(b) An implementable dynamic service security protocol deployed through smart contracts along with the integration of key and trust management module based on a combination of Contiki [34,35], Multichain [36], SDNWise [37,38].
(c) A feasibility demonstration of our proposed decentralized service security architecture for heterogeneous IoT devices during collaborative tasks by implementing a proper municipal smart city use-case with the integration of blockchain and SDN.
(d) The proposed solution is evaluated on throughput, access time delay, and the service security protocol execution time in each operation when collaborative services between IoT devices and heterogeneous networks are taking place. The results show the performance superiority of our proposed solution.

The rest of the paper is organized as follows. Section 2 presents the discussion of smart cities' heterogeneous services along with collaborative services and their security challenges in the smart municipal city. Section 3 shows the related work and highlights various security solutions that come under the framework of SDIoT. In Section 4, and 5 we describe the system overview along with the detailed presentation of the proposed architecture based on SDIoT integrated with blockchain. In Section 6 we discuss the use case in detail. In Section 7 we present the testbed and experimentation details. Finally, in the last section we conclude the study and briefly outline the research challenges and future work.

2. Municipal smart cities

The concept of a "Municipal smart city" can be described as a collection of digital services provided by a network of people who are linked together to share information, resources, expertise, and other assets to improve the efficiency of municipal services [39]. These digital services are one of the most crucial components of a smart city since they connect service providers and users, infrastructures, and communities. The concept of a "smart city" on the other hand, can be described as a collection of heterogeneous network services such as smart transport, smart homes, etc. All of these services must be supported under a general architecture of the smart city. According to the International Telecommunication Union Telecommunication Standardization Sector

Table 1
Summary of identified research gaps through available literature

Research questions	Studies	Summary
Is the current IoT-based Smart cities network secure?	[5] [22]	Confidentiality, integrity, authentication, non-repudiation, and availability are just a few of the security issues that plague current smart city network solutions based on IoT. We will need a scalable solution to keep it secure. SDN and blockchain technology can be used to secure IoT networks.
Can SDN solve the scalability issues in IoT-based Smart cities network?	[23] [24] [25]	SDN's centralized control and programmability make it a viable solution for controlling billions of heterogeneous IoT services that are associated with the future smart cities. On the other hand, data security, privacy, and trust in collaborative services require a solution such as blockchain, that cannot be changed or tampered with.
Can blockchain solve the scalability issues in IoT-based smart cities?	[26] [27] [28]	IoT network scalability issues can be solved thanks to blockchain technology's highly distributed emphasis on *peer-to-peer* distribution. It is still a challenge to put it into practise in real-world applications. Simple blockchain implementations, such as those that use *proof-of-work* and do not have lightweight nodes, will always have bad results.
How to trust heterogeneous IoT services during collaborative tasks?	[29] [30]	There are flaws in the trust proposals that are currently available. For example, it is not clear how the data on a trust list would be organized, and it is not clear how the trust values of collaborative heterogeneous IoT services in smart cities would be calculated.
Is there any security solution available for communication between heterogeneous IoT networks for collaborative tasks?	[31] [32]	A variety of security solutions are available for collaborative services in smart cities, such as smart homes, which address different security issues such as confidentiality, integrity, and availability, but there is a lack of security solutions between multiple smart cities networks for collaborative services.

(ITUT) (ITU 2015) [40], the smart and sustainable city (SSC) architecture should be layered consisting of a) sensing layer; b) network layer; and c) application layer. The sensing layer contains all the supportive protocols that are required to implement the data sensing unit for the applications in a smart city, the network layer is responsible to support all the communication technology that will help to execute any use case of a smart city, and the data and support layer consists of application support services, application support server, data service APIs and can also contain the data repositories related to different services.

2.1. Collaborative services in municipal smart cities

Smart cities are distributed computing environments offering a large number of smart services. Smart city architecture allows for the continuous evolution of various services and related ecosystems. The deployment of the smart city involves a close collaboration between different types of heterogeneous service networks such as (Smart homes, Smart transport, etc.) [41,42]. Smart e-government services deployed in the city depend on information systems that integrate all public sub-services and their provision [43]. For example, forecasting for any upcoming climate change, such as the prediction of storms that may severely affect the city's infrastructure and human life in the city, the collaboration of multiple sub-services in the smart city can be useful, as the national disaster service can obtain information from the city's smart weather service to broadcast a climate alert message to the city's connected citizens. A citizen at the same time can report a problem with public lighting and provide feedback on the municipal policy.

There are several case studies where the municipal smart cities' collaborative services can be carried out to benefit from the potential of IoT in general and smart cities in particular. In a municipal electricity system in a smart city, the collaboration of smart electric services and a smart energy grid balances the instantaneous generation and storage capacities, which smooths out the peaks in electricity demand in the smart city [44]. The implementation of healthcare services such as connected, smart devices will improve healthcare management in a smart city. When physical availability of health specialists is not possible, remotely linked health care facilities backed by data analytics systems can be provided to the treatment centres. It is also necessary for public health practitioners to have access to patient's health information at any time and from any location. Several problems exist for implementing healthcare services, including the following: collaboration between healthcare providers, misunderstanding or mistakes in case of insurance circumstances, data privacy issues, and new standards of operation that are expected of the healthcare system [45].

2.2. Security challenges for collaborative services

Smart city heterogeneous networks integrate millions or billions of IoT devices. The cooperating services with the help of a heterogeneous network in the smart city create several challenges and issues [46] related to data sources, characteristics, information sharing, data quality, security, privacy that must be addressed before the deployment some of them highlighted below,

- **Sharing information**: Collaborative services in smart cities during data sharing have data privacy and legislation challenges because data is exchanged between different cross-domain services. There is a high need for a systematic trust management mechanism and data legislation that can ensure privacy of data during sharing and exchanging of information.
- **Privacy**: In a smart city, privacy is one of the most desirable features. Procedures should be defined to protect the confidentiality of data because sensitive data about individuals and the government is gathered and stored in the database to keep the data safe from unauthorized users, viruses, and bugs. Since there is a high risk of attacks in smart applications involving data transmission through different networks, better policies must be maintained. Data Privacy of connected IoT nodes is vital during collaborating services because data is moving from one system to another network.
- **Agreement between IoT networks**: For collaborative service, there is a need for systematic and transparent cooperation between the IoT networks. Different collaborative services needs to sign an agreement for transparency to provide the required services.
- **Audit trails**: Continuous security audit mechanism is required during collaborative IoT services, as there can be loopholes in requirements to make various monitoring and management policies.
- **Security solution**: Each heterogeneous network has its security solution and resource limitation, so when the collaboration of service is needed, IoT networks need to agree with the security solution of each communicating network, which can be complex.

2.3. Security requirement for collaborative services

Heterogeneous smart city networks connect millions or even billions of Internet of Things (IoT) devices. When services in a smart city work together, they raise a slew of questions about data sources, characteristics, information exchange, data quality, safety, and privacy. Here are the security requirements that should be taken into account when IoT networks are considered for collaborative services in municipal smart cities.

- **Adaptive trust**: Secure and controlled collaborative service between heterogeneous IoT networks is established through adaptive trust mechanisms same as in human nature. This mechanism requires to issue a unique identity and behaviour to each device in heterogeneous IoT networks.
- **Incentive Policies**: To ensure trust during collaborative services in heterogeneous IoT networks for the smart city, we required incentive policies for better interaction.
- **Cryptographic algorithms**: Cryptography can be challenging keeping in view the IoT devices depending on the types of heterogeneous connectives, and aiming to reduce the chances of attacks from unauthorized end-users during collaborative services.
- **Data Protection**: Data confidentiality and integrity must be maintained as part of collaborative services during data transfer and storage. Furthermore, backup and recovery of critical data (e.g., configuration data) are critical.
- **Network Security**: Data security within a network is equally critical; therefore, network security is intertwined with data security. Data forging, replay assaults, network infiltration, and other types of network attacks are all possible. As a result, network security is required within the network. To summarize, in collaborative services between heterogeneous IoT networks of smart city decentralized security is a big challenge. All the actions during collaborative service between heterogeneous IoT networks must be recorded and tracked conveniently for audit authority.

3. Literature review

In this section, we review open problems and challenges associated with heterogeneous IoT nodes' security and scalability in smart cities and the security and scalability challenges during collaborative tasks between multiple heterogeneous IoT networks.

Managing an immense number of IoT devices is now a technological challenge due to their extensive use in various areas. Current IoT management systems are mostly on centralized models and their implementations have some drawbacks in situations when we have numerous numbers of IoT Devices [47]. The existing centralized solution, which is used to manage IoT Devices, creates security and privacy issues in the management of IoT Devices. Blockchain is used as a decentralized solution in the management of IoT Devices because traceability is easy in Blockchain. However, it is not trivial to implement Blockchain on IoT Devices due to scalability and high resource costs that occur in the implementation of Blockchain [48].

Sharma et al. in [49] have also proposed blockchain-based Proof of Work for IoT Devices, which is based on Argon2 for the privacy and security of data and to maintain the integrity of data. This Proof of Work is composed of edge and core network which provides support in achieving decentralized and centralized properties of a network. Alphand et al. [50] proposed a modern solution to network authentication for IoT applications known as “IoTChain”. They used multiple servers to hold keys and other records and used Blockchain as an authorization server. Shen et al. [51] suggested a new system for data sharing known as MedChain. This system runs on two decentralized networks. The first one is the storage of P2P, which stores variable data, including data and session descriptions. The second one is the blockchain network that holds data that cannot be modified, such as data digest.

Cirani et al. [52] proposed a security system that includes outsourced immunization and authorization over the transport layer to intelligent devices. The development of their security model was based on the REST architecture and included the OAS integration with API services. This architecture was designed to achieve specific security goals such as CIA, authorization, outsourced authorization; certain limitations were associated with this architecture. It was a computationally heavy architecture in terms of data processing. Rahman et al. [53] also proposed a security system consisting of a designed security architecture aiming towards a stable IoT cloud environment. This ecosystem concentrated on providing transmission encryption, sufficient network configurability and physical system protection.

Kshetri in [54] describes cloud applications reliance as an enormous weakness because their data storing and processing functionalities are centralized. The author encourages the use of the blockchain to defend digital rights using decentralization and access control systems, as data will be accessible to the parties performing transactions on the distributed ledger. Stergiou et al. [55] published a paper outlining the feasibility of a decentralized distributed ledger through a blockchain that would guarantee immunization through digital signatures, cryptographic hashing and permanent data storage during financial transactions and businesses.

Sahay et al. [56] proposed a framework for providing a network-level defence for the communication network component with the help of translating high-level policy language into open flow rules for the SDN controller. The authors discuss the algorithm of OpenFlow control for making low-level policy also consider the mitigation engine, which has a repository of network policy. Kalkan et al. [57] proposed a security architecture for distributed SDN controllers such as Intrusion controller, key controller, and crypto controller. Each SDN controller interacts with the domain controller with access rules. Farooq et al. [58] proposed framework that implemented five different cryptographic schemes of AES, which demand additional resource requirements. The proposed framework is adaptive and finds the solution through a weighted value of resource and throughput numbers.

Malik et al. [59] proposed a framework that manages the supply chain “Trust Chain” via mapping reputation score and trust score of IoT system in the blockchain. The submitted model has three-layer to stabilize and build trust among the supply chain products and commodities. At top block chain, it checks the cooperates product entities reputation and trust, with the help of predefine states and rules of smart contract blockchain deny or permit products. Finally, the observation layer sends sensing data of IoT products managed by the top layer. Huang et al. [60] proposed a decentralized trust-based framework for IoV and also, introduce the concept of blockchain sharding in their proposed framework. S. Hameed et al. [31] the authors proposed scalable SD-IoT decentralized key exchange and trust management for Heterogeneous IoT Networks. Trust and authentication issues are catered through blockchain efficiently due to its decentralized property, tamper-proof, and immutability data construction.

We have different security challenges in centralized security solutions, such as centrality failure, security solution forging, etc. We identified from the available literature that many researchers are working to address these issues in centralized [52–54] and decentralized environments [31,55,59,60]. In a decentralized solution, the literature suggests a security solution is suggested with the core features of Blockchain such as immutability, decentralization, traceability, and verification. In this paper, we used a dynamic security deployment approach with the help of security contract deployment to individual heterogeneous IoT nodes for the collaborative task, which works according to the trust values in the system. The novelty of our proposed security solution is a dynamic decentralized distributed smart security contract execution by the merger of three security features such as authentication, authorization, and trust.

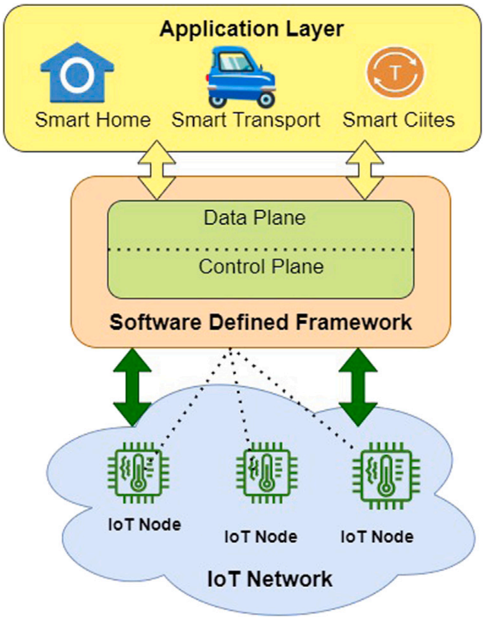


Fig. 1. SDN Integrating IoT networks.

4. System overview

The proposed architecture is the integration of three technologies such as SDIoT, Multichain Blockchain and Smart contract. Our proposed architecture based on smart contract solution will use SDN to implement service security protocols during collaborative service between heterogeneous IoT networks in smart municipal cities. The details about the core technologies involved in the making of the proposed framework are given in subsequent sections.

1. Software-Defined Internet of Things

SD-IoT, or the Software-Defined Internet of Things, has emerged as a new paradigm that uses SDN technology to address the well-known IoT scalability issues. The Internet's current architecture may not be able to handle all of the data generated by the IoT. SDN, on the other hand, is a promising technology that simplifies and scales everything at the controller level.

The challenges of overseeing the IoT can be mitigated with the use of software-defined networking (SDN), a potential technological advance in communication standards [61,62]. Software-defined networking (SDN) separates the data-forwarding elements from the IoT network controller, which simplifies network management. The application plane, the control plane, and the data forwarding plane make up the SDN framework that is integrated with the IoT network. The control plane, also known as the SDN controller, is responsible for keeping track of the overall network and monitoring, prioritizing, and deprioritizing network traffic via programmable APIs. The Data Plane (DP) is responsible for managing data paths and packets according to the control plane policies. The data plan is responsible for taking decisions to forward, drop, or alter packets based on defined policies by the control plane. The application layer is responsible for executing an application [37].

1.1 SDN integrating IoT networks

IoT Integrating an SDN makes it easier to gather data, analyse that data, make decisions, and take action. By utilizing SDN in IoT, network assets may be monitored and managed, and access can be controlled based on factors such as the user's identity, the organization the device is associated with, and the specifics of the implementation. Network control capabilities in the Internet of Things can be improved with the advent of SDWN (Software Defined Wireless Networking). Flexible and scalable demand-based IoT networks are made possible by SDN [63,64]. A typical architecture of integration of the SDN framework with the IoT network is shown in Fig. 1.

In our proposed framework, SDN is used for routing, flow table management, trust management, and key management to access collaborative smart city services within the smart city and multiple smart city interactions. To address the SDN controller's single point of failure issue, we integrate it with the Local-chain blockchain as a decentralized application controller, in which end-to-end message flow of the system is through the controller. The Trust management module is responsible for calculating the trust of the heterogeneous IoT nodes by verifying the security parameters in the received message. To provide CIA to the application message at the controller layer, we implement a key management module to provide an ECC cryptographic security suite.

2. Multichain blockchain

Multichain is an open-source blockchain platform. One of the main goals of blockchain technology is to make things less centralized. Blockchain automatically stores its data in multiple copies in different places around the world. This makes the data very accessible and makes it harder to change chained data [65].

With the introduction of smart filters in MultiChain 2.0, custom logic may now be implemented in a blockchain for the purpose of authenticating transactions and data. Smart Filters are comparable to "smart contracts" that other blockchain platforms offer. There are two types of smart filters, such as transaction filters and stream filters that support multichain blockchain. In transaction filters, the inputs, outputs, and metadata of on-chain transactions are examined by a transaction filter, which verifies the transactions. Transactions that fail to pass the filter are rejected by all nodes in the network. In a stream filter, individual items written to a multi-chain stream are checked against their key, publisher, and on-chain or off-chain data, in JSON, text, or binary format [66].

3. Smart contract

A smart contract is a computer programme (code) that implements the agreed-upon business logic among the network members. In blockchain-based applications, users can access and interact with data. This code is available to everyone on the network (i.e. orderers, peers). As with transactions, these smart contracts are included in blocks that are added by members of a blockchain, just as they are. The next block that is created after a transaction updates a smart contract's state contains a record of the change. Similar to the immutability mechanisms used in transactions, smart contracts can have an immutability mechanism as well. Since the consent of the majority of participants is required to override the terms of a smart contract, it is impossible for one party to unilaterally alter its terms. Smart contracts can be used to automate the business processes.

5. Proposed security service architecture

The proposed architecture for collaborative services between IoT devices of heterogeneous networks is based on three layers: the perception layer, the controller layer, and the application layer, as shown in Fig. 2. In the following subsections, we will describe the overall architecture providing a brief discussion of the key entities of the proposed framework for collaborative tasks between municipal smart city services.

The perception layer is responsible for sensing the intelligent environment's parameters and providing them to the controller layer. At the controller layer, we used the SDN-Wise architecture, composed of the SD-IoT controller. Besides the default module in the SD-IoT controller, we also have the key management and trust module responsible for providing security features to the incoming and outgoing traffic during

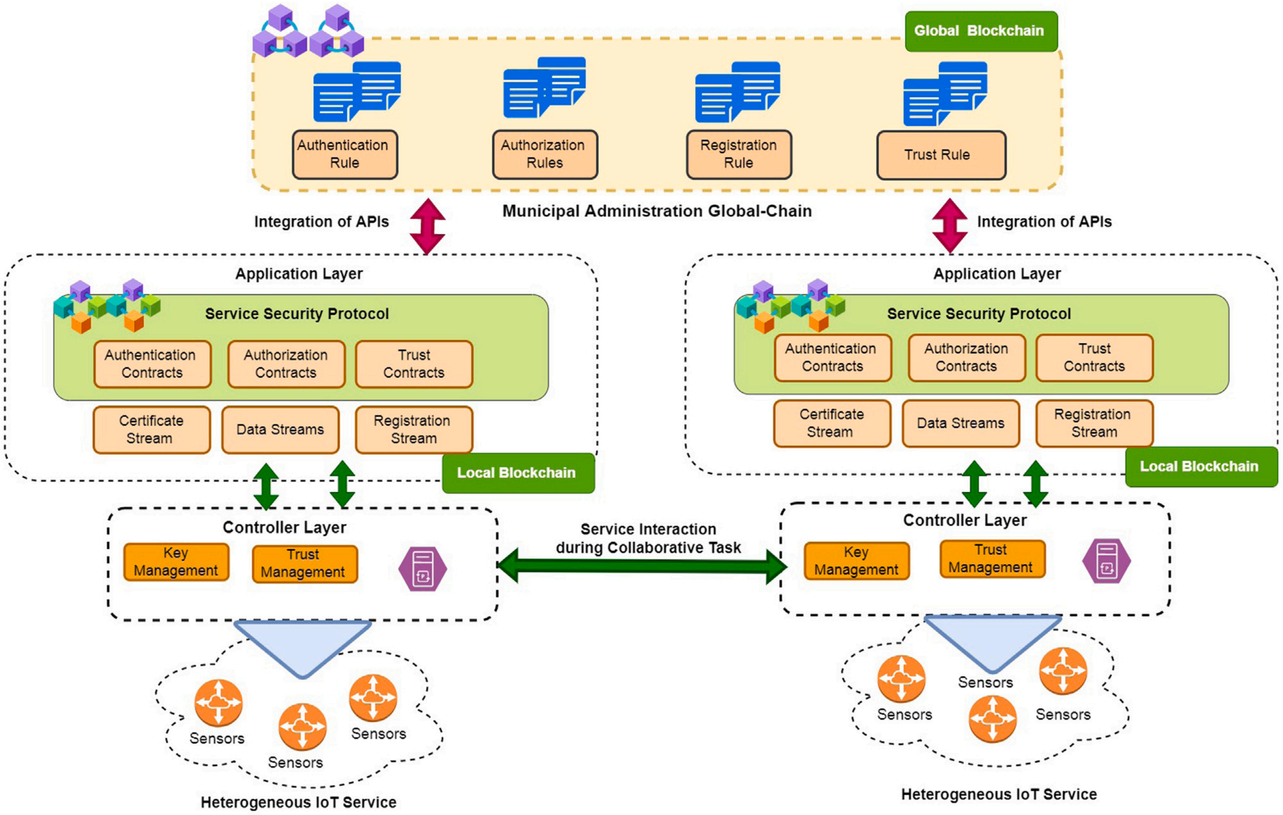


Fig. 2. Proposed security architecture for municipal smart city collaborative services.

collaborative tasks. At the application layer, we have a group of decentralized smart applications. In our proposed framework, we used two types of blockchain such as public and private. The public blockchain acts as an authoritative unit of municipal government, such as the National Disaster Management Authority. The public unit is responsible for providing security policies to the decentralized application running on the local chain. The local blockchain is responsible for executing the security policies through service security protocol execution with the help of smart contracts.

5.1. ECC cryptographic suite

Elliptic curve cryptography (ECC) is public key cryptography created in 1985 by Neal Koblitz and Victor Miller [67]. Elliptic curve cryptography is based on the difficulty of solving elliptic curve number problems [68]. The mathematical algebraic equation of ECC is defined in Eq. (1) [69]. In Eq. (1) a and b are the constant variables. Each value of a and b produces a distinct elliptic curve [70]. The elliptic curve includes all points (x, y) that satisfy the above equation. The public key is a point on the curve, while the private key is a random integer. The public key is calculated by multiplying the private key by the curve's generating point [71].

$$y^2 = x^3 + ax + b \qquad \textbf{where} \qquad 4a^3 + 27b^2 \neq 0 \tag{1}$$

In addition to the curve parameters a and b, some additional parameters provide security attributes during communication between two parties called the domain parameter. Generally, the protocols implementing the ECC specify the domain parameters, one of the domain parameters is the Prime Field, which is discussed in detail in the next subsection.

5.1.1. Domain parameter for prime field F_p

The domain parameters for the prime field are **P, a, b, and G,** where P is the prime number defined for the finite prime field, a and b are the constant variables defining the new curve equation as shown in the Eq. (2) [72]. Here, G is the generator point x_g, y_g, a point on the elliptic curve chosen for cryptographic operations.

$$y^2 mod\, P = (x^3 + ax + b)\ mod\, P \tag{2}$$

To facilitate identification, domain parameters over F_p have been given nicknames. Each name starts with sec, which stands for 'Standards for Efficient Cryptography', followed by a p, which stands for parameters over F_p, and a number, which stands for the length in bits of the field size p. The features of the proposed elliptic curve domain parameters over F_p are summarized in . The names of the elliptic curve domain parameters are provided in the column labelled "parameters". The estimated number of bits of security that the settings provide is shown in the "strength" column. The length of the field order is given in bits in the column titled "size". The "RSA/DSA" column provides an estimate of the size of an RSA or DSA modulus at similar strength.

The major advantage of ECC in our proposed framework is improved performance to tackle the delay-sensitive application. ECC uses much smaller key sizes than RSA and Diffie–Hellman, but it still provides the same security level as the RSA and Diffie–Hellman cryptographic suite. ECC cryptographic suite can be applied for various cryptographic mechanisms such as secret key exchange, digital signatures, and public key encryption [73] (see Table 2).

5.2. Key management

In the controller layer, the key management module is responsible for generating public and private key pairs for IoT devices and a symmetric key for SDN controller legitimacy of heterogeneous networks for collaborative tasks. All IoT devices have in the framework

Table 2
Properties of recommended elliptic curve domain parameters over F_p [72].

Parameters	Strength	Size	RSA/DSA
Secp128r1	64	128	704
Secp160r2	80	160	1024
Secp192r2	96	192	1536
Secp224k1	112	224	2048

uses lightweight 128-bit ECC asymmetric keys, symmetric key AES128-bit Fernet cipher for SDN controller legitimacy. The key management module have also certificate authority through openssl library for generating certificate to IoT devices. Algorithm-1 shows the steps involving to generate the keys and certificate.

After generating a key-pair of IoT devices, symmetric key for SDN controllers, the key management module will need a mechanism to securely transfer the keys to the IoT devices for heterogeneous network to other heterogeneous network through controller. This can be done according to the following sequential steps:

(1) IoT devices of heterogeneous network request the SDN controller to join the other network.
(2) The key management module in the SDN controller-1 sends a self-signed certificate to the SDN controller-2 of other heterogeneous networks.
(3) SDN controller-2 of heterogeneous network will verify the signature on the certificate using the public key of the certificate authority. We assume the certificate authority is already trusted for both SDN controllers and IoT nodes.
(4) SDN controller-2 of heterogeneous network will create a session key, encrypt it with the SDN controller-2 AES-128 symmetric key, and send it to the SDN controller-1.
(5) The SDN controller will generate the key-pair through Key Management module, encrypt the keypair with the session key, and send it back to the IoT devices of heterogeneous networks of both controller.
(6) IoT device of heterogeneous network will decrypt the key-pair using the session keys that received from the controllers of heterogeneous networks.

Algorithm 1 Keys and Certificate Generation

```
Input: Node ID
Output: Generate KeyPairs, Self Signed Certificate, AES Symetric Key
Steps:
SDNkey = Fernetgeneratekey() //for creating a self-signed certificate for the session key//
Certificatekey1=openssl.generateKey(ECC128.private)
Certificatekey2=openssl.generateKey(ECC128.public)
Send Certificate to Session_Key
while (Node count is less than 0) do
    String Private_key, Public_Key
    Privat_key=openssl.generateKey (ECC128.private)
    Public_key=openssl.generateKey (ECC128.public)
    Node.count_PrivateKey=Private_key
    Node.count_PublicKey=Public_key
    Generate Certificate for IoT Nodes
    count=count+1
end while
```

5.3. Trust management

To make an accurate judgement of an entity's trustworthiness, we require an adaptive trust same as in human nature that continuously updates its trust value on some parameters. The trust management module in the SDN controller is responsible to manage all the trust values having a method of retrieving the previous trust and calculating the new trust value. We are using the Boa et al. [74] trust equation, which uses honesty, cooperativeness, and community-interest trust properties.

$$T_{ij}(t) = (1-\alpha)T_{ij}(t-\ \Delta T) + \alpha D_{ij}(t) \tag{3}$$

Here, $T_{ij}(t)$ is a trust assessment during the collaborative task of IoT devices) of heterogeneous services (i), proceeding network (j), at the time (t). The $T_{ij}(t)$ range is between [0,1] and normalizes these values to trusted, semi-trusted, and untrusted as [1, 0.5, and 0]. A parameter (α) represents a trust factor range from 0 to 1, the higher trust factor assessment will rely more on direct observations. The term ($D_{ij}(t)$) represents the direct assessment of trust values between IoT nodes of heterogeneous services. We set (α) value as 1 in our proposed framework which means that the trust value is dependent on the value of direct assessment. Here, by direct assessment, we mean observation of controllers and IoT nodes of a heterogeneous IoT network based on the success of security operations. In Algorithm-2, we use α parameter in order to build the trust value based on direct, indirect, or both assessments. We initially set the trust values for each IoT node to zero, which builds up to 1 throughout the communication. We do not consider the case of an indirect assessment of the trust values in the proposed framework. We will consider it in the future for the use-case attacking scenario. We set the direct assessment value to 0.001 based on security verification through security contracts, which means that each IoT node of a heterogeneous IoT network required 1000 iterations to build the trust up to 1.

Algorithm 2 Trust Management

```
Input: Message from IoT Nodes/Node
Output: Add Trust Values to IoT nodes
Steps:
Float Trust=0
Float Node Trust=0
Set ($\alpha$)=1
while (Node.count is less than 0) do
    KnowSharekey=Hash(SDN Pre-shared Key)
    Decrypt message extracts the SDN Pre-shared Key
    PreSharekey=SDN Pre-shared Key
    if Hash(PreSharekey)==KnowSharekey then
        Node Trust=Node Trust+Trust
        x=(1-($\alpha$))
        Trust=x * Trust+($\alpha$)*(0.001)
    end if
end while
```

5.4. Multichain as local-Blockchain

We used the concept of local blockchain by using multichain blockchain as a decentralized distributed service running on a multichain blockchain in the proposed framework. In multichain, we have published a subscriber stream module used as a general-purpose append-only database, including timestamping, notarization, and immutability.

MultiChain has its own key management module that makes private and public key pairs and stores them away from the node, such as in another decentralized application. The createkeypairs API command is used by the key management module to create these key pairs. We used this module in order to bind the legitimacy of private blockchain with the SDN controller.

The key management module of Multichain securely transfer the their keys to the SDN Controllers of heterogeneous networks. This can be done according to the following sequential steps:

(1) SDN Controllers of heterogeneous network request to the blockchain to join the blockchain network.

(2) The key management module in the multichain sends a self-signed certificate to SDN Controllers of heterogeneous networks.
(3) SDN Controllers of heterogeneous network will verify the signature on the certificate using the public key of the certificate authority. We already discuss the assumption of certificate authority in key management subsection.
(4) The SDN controllers of heterogeneous network will create a session key, encrypt it with the SDN controller's AES-128 bit key, and send it to the blockchain network.
(5) Eventually Blockchain network and SDN controller communicate with each other through this session key.

We used three types of data streams i.e., certificate stream, registration stream, data stream. Certificate stream store the certificate for IoT devices of heterogeneous network which contain the public addresses of IoT nodes signed by the SDN controller. Registration stream responsible to store the registration status of each IoT devices of heterogeneous network. The data stream contains the data logs of distributed applications during collaborative tasks. In order to maintain the security of stream data all the content of stream is signed by the private key of private blockchain.

5.5. Service security protocol

A security protocol is a series of actions performed by two or more communicating entities in order to achieve some mutually desirable goal that makes use of cryptographic techniques and allows the communicating entities to achieve a security goal. A service security protocol is essentially a secure communication protocol for IoT nodes' communication within the network or multiple heterogeneous networks for collaborative tasks. We have three smart contracts that cover the three security goals such as,

(a) Authenticity contracts
(b) Trust contracts
(c) Authorization contracts

The authenticity contract is responsible to verify the authenticity of the IoT device of heterogeneous service and the SDN controller's as per the defined policies for authentication in the global chain. After verification of authenticity, the trust contract is responsible to fetch the trust index of IoT devices of heterogeneous services from the controller in order to update the access list to access the heterogeneous services during collaborative task and return the updated list to the SDN controller. Algorithm-3 presents the pseudocode for this process of service security protocol with smart contract.

Algorithm 3 Service Security Protocol with Smart Contract

```
Input: Message From SDN Controller, Node
Output: List of Trusted Nodes
Steps:
KnowShareKey=Hash (SDN PreSharekey)
Decrypt message extracts the SDN PreSharekey
PreSharekey=SDN PreSharekey
NodeTrustedArray[]=Initialize the empty list
if Hash(PreSharekey)==KnowSharekey then
    Fetch Trusts values from SDN Controller
    Authorize (Node, Trust values) return NodeTrustedArray
else
    Not Allowed
end if
while (Node.count is less than 0) do
    if (node.thresholdTrust is > 0.5) then
        NodeTrustedArray.append(Node)
    end if
end while
```

5.6. Multichain as global-blockchain

An authorized organization's security policies are a formalized set of rules for ensuring that users with access to data and information assets follow rules and guidelines related to data security. These rules are in place to keep confidential information safe from unauthorized access, disclosure, modification, or destruction. The policy manual of an organization is a useful tool for preventing security threats and figuring out how to deal with them [75]. In Global Blockchain Multichain, we present a concept of a decentralized network of authorized entities that pushes security policies. Local chain IoT devices on a heterogeneous network must adhere to these security policies in order to communicate with other IoT devices on the same network for collaborative tasks. We used the python sabac library in order to implement the policy definition point in the global blockchain. Algorithm-4 defines the implementation of the four policies. In the global chain, we have four sets of policies such as,

(a) Authentication policy
(b) Authorization policy
(c) Registration policy
(d) Trust rule

Algorithm 4 Global Blockchain Policy Definition Point

```
Steps:
Authentication Rule=Request of authentication verification must have SDN KEY
Trust Rule=IoT node's trust value must be 0.001 or above
AuthorizationRule=IoT node's trust value must be above 0.5
RegistrationRule=Request of registration verification must have SDN key=0
```

5.7. Registration of network to another heterogeneous network

Fig. 3 depicts the architecture and design for the registration process of IoT devices in one network to another heterogeneous network. Each step is further elaborated as following,

(1) At the start of the system flow IoT devices providing heterogeneous networks needs to register themselves with other heterogeneous networks by requesting it from the SDN controller first.
(2) This request is sent to the key management module by the SDN controller-1. The module then creates the public and private key pairs for IoT nodes and certificates for other networks with different types of nodes. The certificate has both the public key and the address of the SDN controller-2 that is being asked to communicate. The SDN controller-1 only needs to sign the certificate that was made by taking the certificate's hash and signing it with its own private key. The certificate must then be updated with the digital signature and the message encrypted with the SDN controller-2 public key.
(3) In Step 3, the SDN controller-2 verifies the authenticity of the SDN controller-1 by comparing the hash values of the pre-shared key of the SDN controller-1 with the private key of the SDN controller-2.
(4) The key management module of SDN controller-2 calls this the request contract, which sends a request to the globalchain blockchain to get the registration policy. It also generates the public and private keys for the IoT nodes in a network with different types of devices. Eq. (4) shows the request message signed by the SDN-2 controller and encrypts it with the global blockchain public key.

$$Message = \left[(M) + (Cert_{pr}) + (PubKey)_{h}\right]_{pb} \tag{4}$$

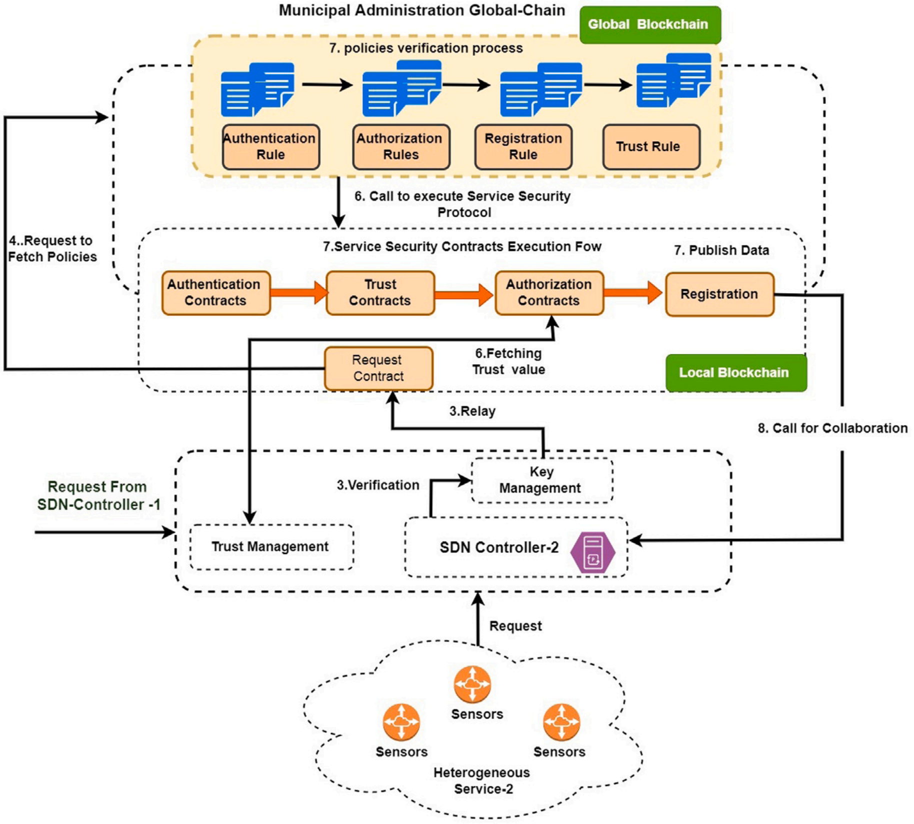


Fig. 3. Registration of network1 to another heterogeneous network2.

$$\begin{cases} Cert_{pri} = \text{SDN signing the Certificate} \\ (Message)_{pub} = \text{Encryption With SDN Public key} \\ (M) = RegisterString \end{cases}$$

(5) By comparing the hash values of the pre-shared key of the SDN controller-2 with the private key of the global blockchain, the global blockchain first confirms the legitimacy of the SDN-2 controller.
(6) The registration policy is sent to the local blockchain after the SDN-2 controller's legitimacy has been successfully verified. It is signed by the global blockchain's private key and encrypted using the local blockchain's public key.
(7) Check the SDN controller's legitimacy first in the service security protocol. Obtain the trust values of IoT nodes from the SDN controller and pass them to the authorization contract to update the access service list if the authenticity is confirmed.
(8) Finally, the local chain registration stream updates the heterogeneous network's registration status on other networks.

5.8. Communication between heterogeneous Network1 to Network2

The architecture and design for the communication process between multiple IoT devices of heterogeneous networks involved in the communication is further elaborated in the following steps,

(1) In our use case, the collaborative task between IoT nodes of heterogeneous networks is based on automation, which means that the condition of the collaborative task is triggered under certain conditions. Therefore, in the communication workflow, the IoT nodes of heterogeneous networks first publish their data on the local blockchain of the other network. For this reason, IoT nodes need to request it first by requesting it from the SDN controller-2 to SDN controller-1 through a request message. The request message includes the message payload, signed certificate from SDN controller-1 containing the public address of the global blockchain and the hash of the pre-shared key of the SDN controller-1 as shown in Eq. (5),

$$Message = \left[(M + SData) + (Cert_{pr}) + (PKey)_h\right]_{pb} \tag{5}$$

$$\begin{cases} Cert_{pri} = \text{SDN signing the Certificate} \\ (Message)_{pub} = \text{Encryption With SDN Public key} \\ (M + SDATA) = Collaborative + SensorsValues \end{cases}$$

(2) In the next step, the SDN controller-2 verifies the authenticity of the SDN controller-1 through comparing the hash values of the pre-shared key of the SDN controller-1 with the private key of the SDN controller-2.
(3) The key management module of SDN controller-2 then calls the request contract, which is responsible for sending a request to fetch the data write access policy from the globalchain blockchain. It also generates the public and private keys for the IoT nodes of heterogeneous networks. Eq. (5) shows the request message signed by the SDN-2 controller and encrypts it with a global blockchain public key
(4) Global blockchain first verifies the legitimacy of the SDN-2 controller by comparing the hash values of the pre-shared key of the SDN controller-2 with the private key of the global blockchain.
(5) After successful verification of the legitimacy of the SDN-2 controller the Data write access policy is sent to the local blockchain which is signed by the private key of the global blockchain and encrypted with the public key of the local blockchain.
(6) In the service security protocol, first check the authenticity of the SDN controller. If the authenticity is verified, then fetch the trust values of IoT nodes from the SDN controller and pass them to the authorization contract in order to update the access service list.

(7) At last, the data status of the heterogeneous network on other network are updated on the local chain registration stream.

6. Use-case discussion

As a use case, we consider the Disaster Management Authority (DMA) for any municipal smart city deployment. DMA is an organization that monitors disasters and develops strategies to reduce the danger posed by them. DMA generally works around many levels of alert systems, such as drought alerts, flood alerts, and others, and preparations for various levels of destruction [76,77]. We are currently concentrating on air quality and environmental alerts in our use case. We have developed a DMA admin portal that is responsible for fetching the sensor values from the environment. For that, we use the open weather API. Sensors will receive data from the environment, such as latitude, longitude, location, air quality, etc. The sensor data will be displayed on the DMA dashboard. An air quality index value above 300 is hazardous and can cause serious respiratory problems. As soon as the air quality index exceeds 300, notification requests will be generated to smart citizens over heterogeneous IoT networks. A smart citizen component is a government authority that holds information on citizens that including their names, addresses, phone numbers, and other details. In our use-case, the user of the smart citizen component is a mobile user that will receive an alert message whenever the air quality index exceeds 300.

In the next section, we are going to discuss the use case in detail with the architectural depictions of the whole interaction during the collaborative task. It will demonstrate how heterogeneous networks communicate with another heterogeneous network from registration to the final communication processes.

7. Testbed and implementation

The open-source COOJA network simulator [78], which is based on the Contiki operating system, is used to replicate our use case. The COOJA simulator is well-known within the WSN (Wireless Sensor Networks) research community [79]. It is used in simulations of wireless sensor networks and examines the performance of numerous devices interacting in wireless sensor networks. In addition, the Cooja simulator allows us to simulate the connection of Contiki OS-based IoT devices to a wireless sensor network utilizing network-level protocols such as CoAP and 6LowPAN. It permits simulation and hardware-level emulation of the Contiki node network. Contiki motes are Internet of Things devices driven by the Contiki operating system [80].

In order to execute the SDIoT architecture, the COOJA simulator was added to SDN-WISE. SDN-WISE is an SDN solution for infrastructureless and wireless networks. It enables wireless devices to communicate on SDN-based networks using the IEEE 802.15.4 physical and MAC layers. In its network, there are different nodes and sinks. Sinks are wired to the network infrastructure, whereas nodes operate wirelessly [81]. All IoT node data packets are transferred to the controller using sinks. In addition, it has a forwarding layer that processes incoming packets according to the inflow tables. The control plane is responsible for updating the forwarding layer's flow tables. Together, they offer us a virtual SD-IoT network on which to conduct and assess our experiments. MultiChain is ultimately used as the blockchain alternative [82]. MultiChain is a Bitcoin fork that can be utilized to construct a permissioned network. MultiChain is platform-independent. However, as we implemented COOJA simulator and SDN-WISE on Linux and our SDN controller is likewise on Linux, and we also installed MultiChain on Linux.

For the test bed, we have installed the COOJA simulator, SDN-WISE, MultiChain, and MultiChain PHP API on two separate Linux PCs for executing the use-case. We are configuring multichain as private and global chains. The sensor nodes were deployed on IEEE 802.15.4 boards with 8 KB RAM and 256 KB flash memory. The detailed hardware and software setup configuration is shown in Table 3.

Table 3
Configuration of the two deployed physical machines.

Physical machines 1 and 2	
CPU (Processor)	Intel® 6th Gen Intel® Core™ i7 (6700)
Memory	16GBDDR
Chipset	Intel® H110 Chipset
Hardrive	512GB Solid State Drive SATA

The proposed solution is evaluated on throughput, access time delay, and the service security protocol execution time in each operation when collaborative services between IoT devices and heterogeneous networks are taking place. The throughput will indicate the number of bytes of data successfully transmitted over the network per second. The operation's duration will be determined by the access delay time, service security protocol execution time, and the running time efficiency of the proposed algorithm. The evaluation method measures the throughput and delays in access time for the following four processes that are compatible with the proposed solution,

(a) Fetching policies from the global Blockchain.
(b) Key submission from network-1 to local Blockchain of network-2.
(c) Send encrypted collaborative messages between networks.
(d) Decryption of publishes data from network-1 to network-2.

We are interested in measuring the throughput, access time delay, and service security execution running time of the upper operations to check the scalability of the global blockchain and local blockchain with our proposed architecture. We first increased the number of requests per node with delays of 600 ms, 120 ms, 60 ms, 30 ms, and 10 ms in each operation when heterogeneous IoT network-1 communicates with other heterogeneous IoT networks-2 having a setup of 50 IoT nodes each time.

To process IoT node transactions during collaborative tasks, the memory pool of both blockchains plays a vital role that is responsible for managing the request and response flow. Both blockchains use a platform called multichain which has its own memory pool. The performance of the proposed framework is majorly dependent upon the working of the memory pool of the multichain blockchain to process transactions of IoT nodes of heterogeneous networks during communication, both when they first arrive (in the memory pool) of the multichain blockchain from the SDN controller and when the transactions are confirmed within a block.

7.1. Fetching policies from the global blockchain

Our solution begins with fetching policies from the global blockchain. A policy is a rule that specifies who may access a particular resource. A single policy can specify authorization using binary or non-binary options. A binary decision consists of "yes" or "no". In order to verify the global blockchain's authenticity, the fetching policies must be verified through the services' security protocol. The high-level graphical presentation of the fetching operation is shown in Fig. 4. It shows the flow of how IoT nodes send requests to fetch the policy in each heterogeneous service.

As the number of requests increases, each request's throughput goes up, and the time it takes to process the fetch operation between requests goes down. Table 4 presents the results of this operation. We also noticed that the throughput of the operation goes down when the amount of delay is 10 ms to send 5000 fetching requests from 50 IoT nodes, which means a total load of requests in one service is 250 000 requests per node, and the time it takes to process the fetch operation between requests goes up as shown in the graphical representation of fetching policies operation in Fig. 7. Service security execution time during policy fetching is shown in Table 5.

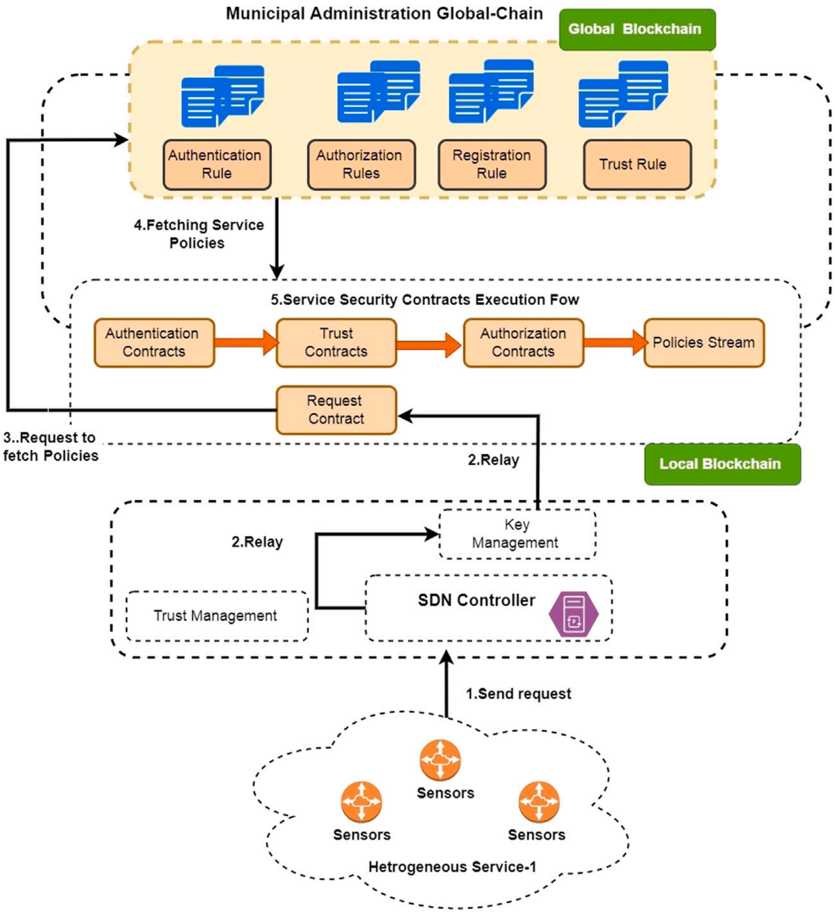


Fig. 4. Fetching policies operation.

Table 4
Fetching policies request from global Blockchain.

Number of requests	Throughput fetching policies (requests/s)	Access delay (ms)
100	1.63	631
500	7.43	137.57
1000	13.67	73.44
2000	24.1	41.47
3000	36.55	27.44
5000	22.06	61.47

Table 6
Key submission operation.

Number of requests	Throughput key submission (requests/s)	Access delay (ms)
100	1.2	667
500	6.25	175.08
1000	11.27	110.91
2000	19.43	91.45
3000	32.02	68.01
5000	18.86	151.45

Table 5
Service security execution time during policies fetching.

Number of requests	Service security execution time (ms)
100	0.9
500	0.74
1000	0.52
2000	0.32
3000	0.2
5000	1.74

7.2. Key submission from Network-1 to local blockchain of Network-2

The second focus in our solution is the submission of the key to the local blockchain of the collaborative IoT network that is requested for collaborative tasks as shown in Fig. 5. This request is sent to the key management module by the SDN controller-1. The module generates public and private key pairs and certificates for other heterogeneous networks. The public key and the address of the requested communicating SDN controller-2 are both included in the certificate. The process is to simply take the certificate's hash and sign it with the SDN controller-1's private key. The requested message is encrypted with the SDN controller-2 public key after all the steps of verification at the SDN-2 controller. The submission of keys of IoT nodes of heterogeneous network-1 follow the same steps, that is, fetch the policy first, then execute the service security protocol to publish the key to the stream of the local blockchain of heterogeneous network-2. Table 6 presents the results of this operation. We can notice the same pattern of result as in the previous operation with a difference of the throughput values as compared with the previous goes down. This is due to execution of operations is after fetching operation. The service security execution time during key submission is shown in Table 7.

The access delay of the operation shows the same pattern but the difference of minor increased in the value of delay operation as shown in Fig. 9. The execution time of the service security protocol is decrease when the submission key request is increased as shown in Table 9. We also noticed at key submission operation throughput compared with fetching operation goes down. The service security execution time is little bit increase as compared with fetching operation as shown in Fig. 8.

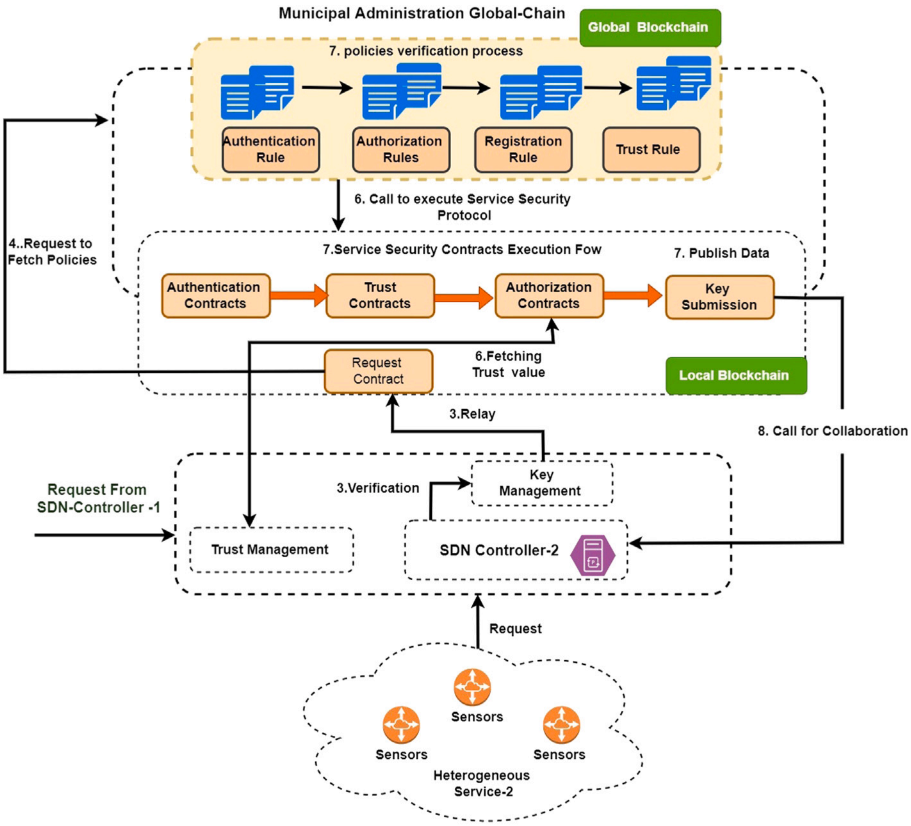


Fig. 5. Key submission operation.

Table 7
Service security execution time during key submission.

Number of requests	Service security execution time (ms)
100	1.93
500	1.58
1000	1.2
2000	0.95
3000	0.467
5000	2.56

Table 8
End-to-end send encrypted messages.

Number of requests	Throughput send encrypted message (requests/s)	Access delay (ms)
100	0.98	695
500	5.2	197.53
1000	8.96	142.44
2000	17.5	111.49
3000	27.02	87.41
5000	16.5	167.52

7.3. End-to-end message sending for collaborative service

The third operation in our solution is to encrypt and send messages during collaborative tasks between two heterogeneous services as shown in Fig. 6. In our use case, we have two heterogeneous services in which one is responsible for sensing sensor values such as the value of latitude, longitude, and air index value. The other service is responsible for providing alert messages to their connected IoT nodes when a collaborative call is generated from service 1. The work flow of end-to-end message sending for collaborative tasks of a defined use-case is given below.

1. The IoT node of heterogeneous service-1 sends a message to the SDN controller by sending a sense value along with an SDN preshared key in a message encrypted with the public key of the SDN controller.
2. The SDN controller first verifies the authenticity of the legitimate IoT node message by decrypting the message with its private key and comparing the hash value of the preshared key with the help of key management module. After verification, the key management module sends the request to the request contract from the local blockchain
3. The request contract forward the request to execute the service security protocol
4. The service security protocol verifies the message's authenticity using the SDN preshared key, then extracts the trust values of the IoT nodes from the controller and updates the authorized access list for data publishing of particular IoT nodes.
5. When the triggered condition occurs, such as the air index value crossing the threshold value then Collaborative service request will called through SDN controller to the requested service SDN controller
6. Here, we assume that both SDN controllers knows public keys of each other. Therefore when collaborative call is generated after triggered condition the Service-2 SDN controller first verify the authenticity of service-1 SDN controller by decrypting the message with its private key and comparing the hash value of the pre-shared key of SDN-1 controller with the help of key management module. After verification, the key management

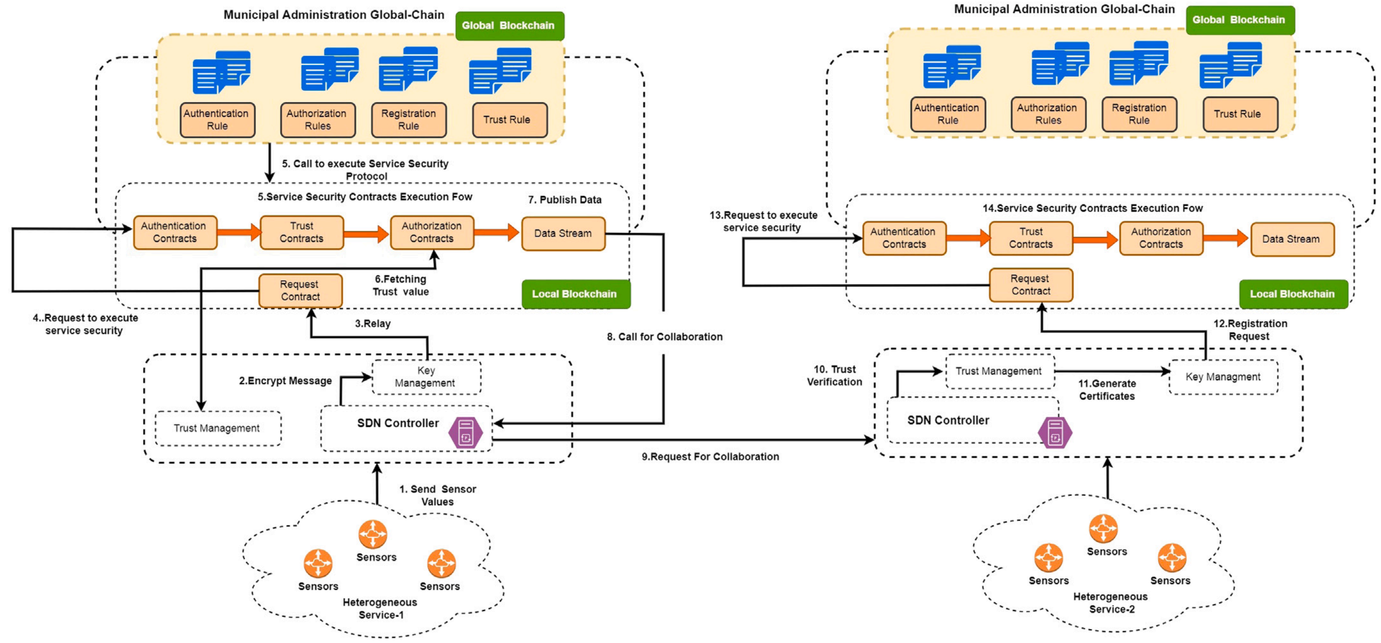


Fig. 6. End-to-end encrypted message sending procedure.

Table 9
Service security execution time during End-to-End message sending.

Number of requests	Service security execution time (ms)
100	2.3
500	1.94
1000	1.57
2000	1.24
3000	0.89
5000	3.0

Table 10
Decryption of publish data from network-1 to network-2.

Number of requests	Throughput decryption message (requests/s)	Access delay (ms)
100	1.52	645.6
500	7.1	142.56
1000	12.15	93.91
2000	23.51	56.45
3000	34.02	35.01
5000	20.08	57.45

Table 11
Service security execution time during decryption of publish data.

Number of requests	Service security execution time (ms)
100	1.2
500	0.945
1000	0.72
2000	0.63
3000	0.56
5000	0.9

module sends the request to the request contract from the local blockchain.

7. The request contract forward the request to execute the service security protocol.
8. The service security protocol verifies the message's authenticity using the SDN pre-shared key, then extracts the trust values of the IoT nodes from the controller and updates the authorized access list for data publishing of particular IoT nodes.

As the number of requests increases, each request's throughput goes up, and the time it takes to process the key submission operation between requests goes down, shown in Table 8. From the result it is also deduce that throughput of the end-to-end message is less as compared with other operation because of the encryption process involve during communication from SDN controller-1 to SDN-2 controller.

On the other hand, the execution time of the service security protocol shows a decrease when the load of requests is increased, but it goes up when the amount of delay is 10 ms to send 5000 fetching requests from 50 IoT nodes, as shown in Table 9.

7.4. Decryption of publish data from Network-1 to Network-2

The last operation in our solution is the access of collaborative message from the stream of service-2 local blockchain published by service-1. The IoT nodes of heterogeneous services-2 send requests to the SDN controller to access the publish data. The SDN controller first verifies the legitimacy of IoT nodes of heterogeneous service-2 by comparing the hash value of SDN preshared key after decryption of the request message with the help of key management module. The key management module then forwards the request to the service security module in order to access the data. By using the same method of evaluation as before the throughput of the operation is much more better than the previous case because of publish and subscriber model for data access. The heterogeneous network-1 is only responsible to post or publish the data on data stream of heterogeneous network-2 when the alarming condition occur. The user of that network only access the data through subscription of this alert message. Table 10 shows the throughput reading on request basis to access the data. The service security execution time during decryption of publish data is shown in Table 11.

It is clear from Fig. 7 that the throughput of the decryption operation of publishing data from heterogeneous network-1 to network-2 is a little bit similar to operation one, which is a fetching operation. Due to the publish subscriber model, on the other hand, the execution time of the service security protocol shows a decrease when the load of requests

Table 12
Comparative result analysis of End-to-end send encrypted messages with ECC key length.

Number of requests	Throughput (requests/s) ECC-128 key length	Throughput (requests/s) ECC-192 key length	Throughput (requests/s) ECC-521 key length
100	0.93	0.75	0.53
500	5.2	4.58	3.21
1000	8.96	5.92	4.21
2000	17.26	14.48	11.21
3000	27.16	18.12	10.01
5000	16.02	12.1	8.01

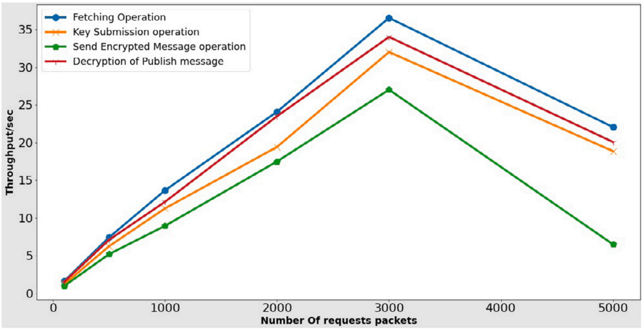


Fig. 7. Throughput of all operations when changing the number of request messages.

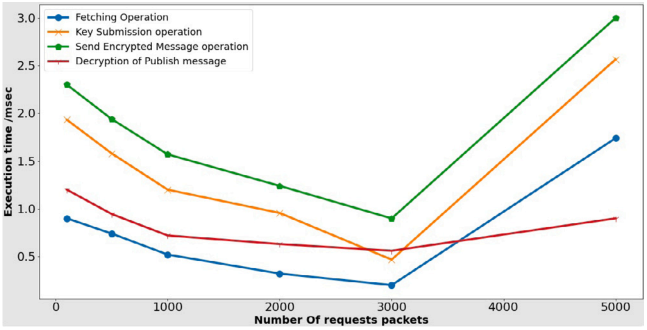


Fig. 8. Running time of service security execution of all operation when changing the number of request messages.

is increased, but it goes up when the amount of delay is 10 ms to send 5000 fetching requests from 50 IoT nodes as shown in Table 9. The graphical representation of service security protocol of this operation when changing the number of request message is shown in Fig. 8. The result pattern of the access delay time of the operation represents that when the number of request goes up to 5000 the access delay time is increase as shown in Fig. 9.

8. Evaluation

The selection of key length in cryptographic suites for IoT devices during collaborative services is one of the important parameters that can determine the details of how much the security parameters have improved by the proposed solution. We conducted a new experiment for the case of sending encrypted collaborative messages between networks. The main theme for this experiment is to show that the selection of cryptographic key length is an important variable in order to tackle the delay in sensitive applications and the execution of dynamic smart security protocols. We show how important the key length is to the performance of the system and how well service security protocols work by comparing the performance of the system with three different key lengths for the ECC cryptography suite: 128 bits, 192 bits, and 521 bits.

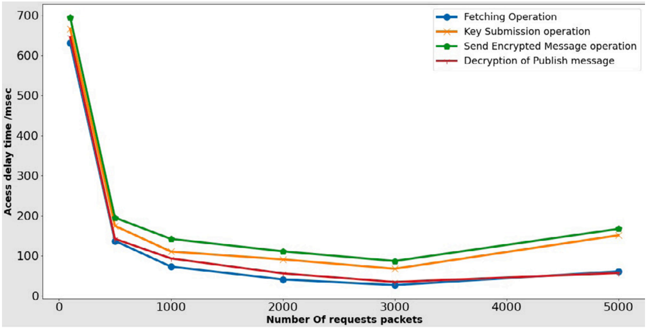


Fig. 9. Access delay of all operation when changing the number of request messages.

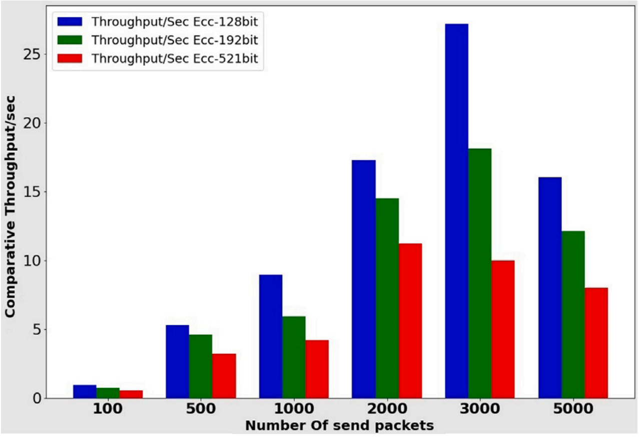


Fig. 10. Comparative result of End-to-end encrypted message by changing key length.

Table 12 shows the comparative throughput result against different ECC key sizes of the proposed framework at the particular operation. We notice from the throughput patterns shown in Fig. 10 that when the number of sent messages with each ECC key length increases, the system's throughput goes down. This is because of the encryption of messages during the End-to-End send operation. The larger the key length, the greater the encryption time. The importance of key length for delay sensitive application during the particular operation can also be noted in these experiments.

9. Conclusion

As the population continues to rise and the concept of smart cities is becoming a reality, monitoring and managing the effects of urbanization on the environment, improving citizens' daily lives, and providing enhanced efficiency for government authorities demand increasingly innovative strategies. In this paper, the proposed architecture supports the communication framework of heterogeneous IoT networks to communicate with each other during collaborative tasks. Our solution focuses on the implementation of adaptive security mechanisms.

The proposed architecture describes the registration, communication, and dynamic service security protocol execution between IoT devices in heterogeneous networks in a municipal smart city concept. Our evaluations proved that the proposed solution is scalable when the number of requests is increased during the communication of two different IoT networks for collaborative tasks. We also present a unique concept of integrating global blockchain and local blockchain in order to implement a municipal smart city environment. The strength of our proposed architecture is shown by how well the service security algorithms work in real life. The current solution can be extended in the following directions as future work,

a. Global-chain dynamic policy engine implementation for enhanced security mechanisms.
b. Implementation of other added modules such as incentive schemes and mechanisms in local chains for smart citizens focusing on various collaborative applications.

CRediT authorship contribution statement

Shahbaz Siddiqui: Conceptualization, Methodology, Software, Writing-original draft preparation. **Sufian Hameed:** Conceptualization, Methodology, Formal analysis, Supervision. **Syed Attique Shah:** Formal analysis, Methodology, Writing-original draft preparation, Validation. **Abdul Kareem Khan:** Implementation, Software. **Adel Aneiba:** Editing, Validation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Syed Attique Shah reports financial support, equipment, drugs, or supplies, and writing assistance were provided by Birmingham City University. Syed Attique Shah reports a relationship with Birmingham City University that includes: employment.

Data availability

Data will be made available on request.

References

[1] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things, IEEE Internet Things J. 1 (2) (2014) 112–121.

[2] J.H. Lee, M.G. Hancock, M.-C. Hu, Towards an effective framework for building smart cities: Lessons from Seoul and San Francisco, Technol. Forecast. Soc. Change 89 (2014) 80–99.

[3] J. Curzon, A. Almehmadi, K. El-Khatib, A survey of privacy enhancing technologies for smart cities, Pervasive Mob. Comput. 55 (2019) 76–95.

[4] P. Bellini, P. Nesi, G. Pantaleo, IoT-enabled smart cities: A review of concepts, frameworks and key technologies, Appl. Sci. 12 (3) (2022) 1607.

[5] F. Al-Turjman, H. Zahmatkesh, R. Shahroze, An overview of security and privacy in smart cities' IoT communications, Trans. Emerg. Telecommun. Technol. 33 (3) (2022) e3677.

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

[7] Z. Khan, Z. Pervez, A.G. Abbasi, Towards a secure service provisioning framework in a smart city environment, Future Gener. Comput. Syst. 77 (2017) 112–135.

[8] A. Al Mahfuj Shaan, T. Nausheen, A.B. Haque, Blockchain for smart city: Opportunities and future research directions, in: International Conference on Digital Technologies and Applications, Springer, 2022, pp. 267–275.

[9] A.A. Monrat, O. Schelén, K. Andersson, A survey of blockchain from the perspectives of applications, challenges, and opportunities, IEEE Access 7 (2019) 117134–117151.

[10] A. Buldas, D. Draheim, M. Gault, R. Laanoja, T. Nagumo, M. Saarepera, S.A. Shah, J. Simm, J. Steiner, T. Tammet, A. Truu, An ultra-scalable blockchain platform for universal asset tokenization: Design and implementation, IEEE Access 10 (2022) 77284–77322.

[11] R. Chaudhary, A. Jindal, G.S. Aujla, S. Aggarwal, N. Kumar, K.-K.R. Choo, BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system, Comput. Secur. 85 (2019) 288–299.

[12] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, J.J. Kishigami, Blockchain contract: A complete consensus using blockchain, in: 2015 IEEE 4th Global Conference on Consumer Electronics, GCCE, IEEE, 2015, pp. 577–578.

[13] E. Portmann, Rezension blockchain: Blueprint for a new economy, 2018, http://dx.doi.org/10.1365/s40702-018-00468-4.

[14] S. Siddiqui, S. Hameed, S.A. Shah, I. Ahmad, A. Aneiba, D. Draheim, S. Dustdar, Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects, IEEE Access 10 (2022) 70850–70901.

[15] G.S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, R. Buyya, Blocksdn: Blockchain-as-a-service for software defined networking in smart city applications, IEEE Network 34 (2) (2020) 83–91.

[16] M. Khalid, S. Hameed, A. Qadir, S.A. Shah, D. Draheim, Towards SDN-based smart contract solution for IoT access control, Comput. Commun. 198 (2023) 1–31.

[17] M.J. Islam, A. Rahman, S. Kabir, M.R. Karim, U.K. Acharjee, M.K. Nasir, S.S. Band, M. Sookhak, S. Wu, Blockchain-SDN-based energy-aware and distributed secure architecture for IoT in smart cities, IEEE Internet Things J. 9 (5) (2021) 3850–3864.

[18] K. Gai, J. Guo, L. Zhu, S. Yu, Blockchain meets cloud computing: A survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2009–2030.

[19] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, Q. Zhang, K.-K.R. Choo, An energy-efficient SDN controller architecture for IoT networks with blockchain-based security, IEEE Trans. Serv. Comput. 13 (4) (2020) 625–638.

[20] T. Alharbi, Deployment of blockchain technology in software defined networks: A survey, IEEE Access 8 (2020) 9146–9156.

[21] J. Xie, H. Tang, T. Huang, F.R. Yu, R. Xie, J. Liu, Y. Liu, A survey of blockchain technology applied to smart cities: Research issues and challenges, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2794–2830.

[22] B. Bhushan, A. Khamparia, K.M. Sagayam, S.K. Sharma, M.A. Ahad, N.C. Debnath, Blockchain for smart cities: A review of architectures, integration trends and future research directions, Sustainable Cities Soc. 61 (2020) 102360.

[23] K.H. Manguri, S.M. Omer, SDN for IoT environment: A survey and research challenges, in: ITM Web of Conferences, vol. 42, EDP Sciences, 2022, p. 01005.

[24] S. Rathore, B.W. Kwon, J.H. Park, BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network, J. Netw. Comput. Appl. 143 (2019) 167–177.

[25] C. Esposito, M. Ficco, B.B. Gupta, Blockchain-based authentication and authorization for smart city applications, Inf. Process. Manage. 58 (2) (2021) 102468.

[26] A. Abdelmaboud, A.I.A. Ahmed, M. Abaker, T.A.E. Eisa, H. Albasheer, S.A. Ghorashi, F.K. Karim, Blockchain for IoT applications: Taxonomy, platforms, recent advances, challenges and future research directions, Electronics 11 (4) (2022) 630.

[27] M. Asif, Z. Aziz, M. Bin Ahmad, A. Khalid, H.A. Waris, A. Gilani, Blockchain-based authentication and trust management mechanism for smart cities, Sensors 22 (7) (2022) 2604.

[28] M.A. Dar, A. Askar, S.A. Bhat, Blockchain based secure data exchange between cloud networks and smart hand-held devices for use in smart cities, in: 2022 International Conference on Artificial Intelligence in Information and Communication, ICAIIC, IEEE, 2022, pp. 457–460.

[29] A. Altaf, H. Abbas, F. Iqbal, M.M.Z.M. Khan, A. Rauf, T. Kanwal, Mitigating service-oriented attacks using context-based trust for smart cities in IoT networks, J. Syst. Archit. 115 (2021) 102028.

[30] B. Wang, M. Li, X. Jin, C. Guo, A reliable IoT edge computing trust management mechanism for smart cities, IEEE Access 8 (2020) 46373–46399.

[31] S. Hameed, S.A. Shah, Q.S. Saeed, S. Siddiqui, I. Ali, A. Vedeshin, D. Draheim, A scalable key and trust management solution for IoT sensors using SDN and blockchain technology, IEEE Sens. J. 21 (6) (2021) 8716–8733.

[32] S. Naoui, M.E. Elhdhili, L.A. Saidane, Lightweight enhanced collaborative key management scheme for smart home application, in: 2017 International Conference on High Performance Computing & Simulation, HPCS, IEEE, 2017, pp. 777–784.

[33] L. Cui, G. Xie, Y. Qu, L. Gao, Y. Yang, Security and privacy in smart cities: Challenges and opportunities, IEEE Access 6 (2018) 46134–46145.

[34] F. Österlind, A Sensor Network Simulator for the Contiki OS, Swedish Institute of Computer Science, 2006.

[35] Y.B. Zikria, M.K. Afzal, F. Ishmanov, S.W. Kim, H. Yu, A survey on routing protocols supported by the Contiki Internet of things operating system, Future Gener. Comput. Syst. 82 (2018) 200–219.

[36] G. Wood, Polkadot: Vision for a heterogeneous multi-chain framework, White Paper 21 (2016) 2327–4662.

[37] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks, in: 2015 IEEE Conference on Computer Communications, INFOCOM, IEEE, 2015, pp. 513–521.

[38] R. Taş, Ö.Ö. Tanrıöver, Building a decentralized application on the ethereum blockchain, in: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT, IEEE, 2019, pp. 1–4.

[39] L. Berntzen, M.R. Johannessen, The role of citizen participation in municipal smart city projects: Lessons learned from Norway, in: Smarter As the New Urban Agenda, Springer, 2016, pp. 299–314.

[40] L. Anthopoulos, Defining smart city architecture for sustainability, in: Proceedings of 14th Electronic Government and 7th Electronic Participation Conference, IFIP2015, 2015, pp. 140–147.

[41] M. Strohbach, H. Ziekow, V. Gazis, N. Akiva, Towards a big data analytics framework for IoT and smart city applications, in: Modeling and Processing for Next-Generation Big-Data Technologies, Springer, 2015, pp. 257–282.

[42] M.M. Rathore, S. Attique Shah, A. Awad, D. Shukla, S. Vimal, A. Paul, A cyber-physical system and graph-based approach for transportation management in smart cities, Sustainability 13 (14) (2021) 7606.

[43] A. AlEnezi, Z. AlMeraj, P. Manuel, Challenges of IoT based smart-government development, in: 2018 21st Saudi Computer Society National Computer Conference, NCC, IEEE, 2018, pp. 1–6.

[44] S. Tanwar, S. Tyagi, S. Kumar, The role of internet of things and smart grid for the development of a smart city, in: Intelligent Communication and Computational Technologies, Springer, 2018, pp. 23–33.

[45] D.J. Cook, G. Duncan, G. Sprint, R.L. Fritz, Using smart city technology to make healthcare smarter, Proc. IEEE 106 (4) (2018) 708–722.

[46] A. Bartoli, J. Hernández-Serrano, M. Soriano, M. Dohler, A. Kountouris, D. Barthel, Security and privacy in your smart city, in: Proceedings of the Barcelona Smart Cities Congress, vol. 292, (no. 6) 2011.

[47] O. Novo, Scalable access management in IoT using blockchain: A performance evaluation, IEEE Internet Things J. 6 (3) (2018) 4694–4701.

[48] Y. Jiang, C. Wang, Y. Wang, L. Gao, A cross-chain solution to integrating multiple blockchains for IoT data management, Sensors 19 (9) (2019) 2042.

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

[50] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, F. Zanichelli, IoTChain: A blockchain security architecture for the internet of things, in: 2018 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2018, pp. 1–6.

[51] B. Shen, J. Guo, Y. Yang, MedChain: Efficient healthcare data sharing via blockchain, Appl. Sci. 9 (6) (2019) 1207.

[52] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, G. Ferrari, Iot-oas: An oauth-based authorization service architecture for secure services in IoT scenarios, IEEE Sens. J. 15 (2) (2014) 1224–1234.

[53] A.F.A. Rahman, M. Daud, M.Z. Mohamad, Securing sensor to cloud ecosystem using internet of things (IoT) security framework, in: Proceedings of the International Conference on Internet of Things and Cloud Computing, 2016, pp. 1–5.

[54] N. Kshetri, Can blockchain strengthen the internet of things? IT Prof. 19 (4) (2017) 68–72.

[55] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B.B. Gupta, Efficient IoT-based sensor BIG data collection–processing and analysis in smart buildings, Future Gener. Comput. Syst. 82 (2018) 349–357.

[56] R. Sahay, W. Meng, D.S. Estay, C.D. Jensen, M.B. Barfod, CyberShip-IoT: A dynamic and adaptive SDN-based security policy enforcement framework for ships, Future Gener. Comput. Syst. 100 (2019) 736–750.

[57] K. Kalkan, S. Zeadally, Securing internet of things with software defined networking, IEEE Commun. Mag. 56 (9) (2017) 186–192.

[58] U. Farooq, N.U. Hasan, I. Baig, Securing internet of things (IoT) through an adaptive framework, in: 2019 16th International Multi-Conference on Systems, Signals & Devices, SSD, IEEE, 2019, pp. 387–392.

[59] S. Malik, V. Dedeoglu, S.S. Kanhere, R. Jurdak, Trustchain: Trust management in blockchain and IoT supported supply chains, in: 2019 IEEE International Conference on Blockchain, Blockchain, IEEE, 2019, pp. 184–193.

[60] P.K. Singh, R. Singh, S.K. Nandi, K.Z. Ghafoor, D.B. Rawat, S. Nandi, Blockchain-based adaptive trust management in internet of vehicles using smart contract, IEEE Trans. Intell. Transp. Syst. (2020).

[61] P.K. Sharma, S. Singh, Y.-S. Jeong, J.H. Park, Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks, IEEE Commun. Mag. 55 (9) (2017) 78–85.

[62] A.L. Aliyu, A. Aneiba, M. Patwary, P. Bull, A trust management framework for software defined network (SDN) controller and network applications, Comput. Netw. 181 (2020) 107421.

[63] J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed, S.A. Shah, A time-efficient approach toward DDoS attack detection in IoT network using SDN, IEEE Internet Things J. 9 (5) (2022) 3612–3630.

[64] O. Salman, I. Elhajj, A. Chehab, A. Kayssi, IoT survey: An SDN and fog computing perspective, Comput. Netw. 143 (2018) 221–246.

[65] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: A survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.

[66] U. Majeed, L.U. Khan, I. Yaqoob, S.A. Kazmi, K. Salah, C.S. Hong, Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges, J. Netw. Comput. Appl. 181 (2021) 103007.

[67] J. Lopez, R. Dahab, An overview of elliptic curve cryptography, 2000.

[68] N. Koblitz, The uneasy relationship between mathematics and cryptography, Notices Amer. Math. Soc. 54 (8) (2007) 972–979.

[69] A. Cilardo, L. Coppolino, N. Mazzocca, L. Romano, Elliptic curve cryptography engineering, Proc. IEEE 94 (2) (2006) 395–406.

[70] M.Y. Malik, Efficient implementation of elliptic curve cryptography using low-power digital signal processor, in: 2010 the 12th International Conference on Advanced Communication Technology, Vol. 2, ICACT, IEEE, 2010, pp. 1464–1468.

[71] M. Qu, Sec 2: Recommended Elliptic Curve Domain Parameters, Certicom Res., Mississauga, ON, Canada, Tech. Rep. SEC2-Ver-0.6, Citeseer, 1999.

[72] C. Sajeev, G.J.A. Jose, Elliptic curve cryptography enabled security for wireless communication, Int. J. Comput. Sci. Eng. 2 (06) (2010) 2187–2189.

[73] Y. Seo, Practical implementations of ECC in the blockchain, Anal. Appl. Math. (2017) 43.

[74] F. Bao, R. Chen, J. Guo, Scalable, adaptive and survivable trust management for community of interest based internet of things systems, in: 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems, ISADS, IEEE, 2013, pp. 1–7.

[75] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, J. Al-Jaroodi, Applications of big data to smart cities, J. Internet Serv. Appl. 6 (1) (2015) 1–15.

[76] S.A. Shah, D.Z. Seker, S. Hameed, D. Draheim, The rising role of big data analytics and IoT in disaster management: Recent advances, taxonomy and prospects, IEEE Access 7 (2019) 54595–54614.

[77] S.A. Shah, D.Z. Seker, M.M. Rathore, S. Hameed, S.B. Yahia, D. Draheim, Towards disaster resilient smart cities: Can internet of things and big data analytics be the game changers? IEEE Access 7 (2019) 91885–91903.

[78] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with Cooja, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, IEEE, 2006, pp. 641–648.

[79] A. Velinov, A. Mileva, Running and testing applications for contiki OS using cooja simulator, 2016.

[80] Sharad, E.N. Kaur, I.K. Aulakh, Evaluation and implementation of cluster head selection in WSN using Contiki/Cooja simulator, J. Stat. Manag. Syst. 23 (2) (2020) 407–418.

[81] S.S.G. Shiny, K. Murugan, TSDN-WISE: Automatic threshold-based low control-flow communication protocol for SDWSN, IEEE Sens. J. 21 (17) (2021) 19560–19569.

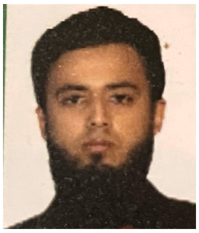
[82] E. Fernando, H. Meyliana, E. Abdurachman, Blockchain technology for tracing drug with a multichain platform: Simulation method, Adv. Sci. Technol. Eng. Syst. 6 (2021) 765–769.

Shahbaz Siddiqui received M.S. Degree in Telecommunication from Hamdard University Karachi, Pakistan. He is pursuing his Ph.D. in Computer Sciences from the National University of Computer and Emerging Sciences, Karachi. He currently works as an assistant professor at the Department of Computer Science at the National University of Computer and Emerging Sciences in Karachi Pakistan. His research interests include the Internet of Things, SDN, and blockchain.

Sufian Hameed received the Ph.D. degree in networks and information security from University of Göttingen, Germany. He works as Associate Professor at Department of Computer Science at National University of Computer and Emerging Sciences, Pakistan. He also leads the IT Security Labs at NUCES. The research lab studies and teaches security problems and solutions for different types of information and communication paradigms. His research area includes network security, web security, mobile security and secure architectures and protocols for Cloud and IoTs.

Syed Attique Shah received the Ph.D. degree from the Institute of Informatics, Istanbul Technical University, Istanbul, Turkey. During his Ph.D., he studied as a Visiting Scholar at the University of Tokyo, Japan, the National Chiao Tung University, Taiwan, and the Tallinn University of Technology, Estonia, where he completed the major content of his thesis. He has worked as an Associate Professor and the Chairperson at the Department of Computer Science, BUITEMS, Quetta, Pakistan. He was also engaged as a Lecturer at the Data Systems Group, Institute of Computer Science, University of Tartu, Estonia. Currently, he is

working as a Lecturer in Smart Computer Systems, at the School of Computing and Digital Technology, Birmingham City University, United Kingdom. He is a Senior Member, IEEE. His research interests include big data analytics, the Internet of Things, machine learning, network security, and information management.



Abdul Kareem Khan received B.S. Degree in Computer Science from Fast National University of computer science Karachi, Pakistan. He is pursuing his M.S. in Computer Sciences from the National University of Computer and Emerging Sciences, Karachi. He currently works as an Senior iOS Engineer at Sixlogs Technologies. His research interests include the Internet of Things, SDN, and blockchain.

Adel Aneiba received his Ph.D. degree in the field of mobile computing and distributed systems from Staffordshire University, in 2008. He has worked as a Senior ICT Consultant for international organizations, for 10 years, including UNESCO and several governmental organizations for many years, and has participated in managing mega ICT projects mainly on data centre designing and development, and reengineering business processes. He is currently an Associate Professor in the area of computer networks and the Internet of Things (IoT) with Birmingham City University. He is the research lead for Cyber–physical Systems (CPS) Research Group. He is supervising several Ph.D. students on various research topics, such as the IoT, SDN, resources allocations, and optimization in 5G and Blockchain applications in the smart cities domain. His current research interests include the IoT, computer networks, evaluation, and optimization, and blockchain. He is a member of the Association for Computing Machinery (ACM) and the Institute of Engineering and Technology (IET) and is a Fellow of the Higher Education Academy (HEA) and a member of many technical committees for scientific academic conferences and journals.