

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368591981>

A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT

Article in Information · February 2023

DOI: 10.3390/info14020129

CITATIONS

32

READS

557

5 authors, including:



Samia Masood Awan

NED University of Engineering and Technology, Karachi

7 PUBLICATIONS 60 CITATIONS

SEE PROFILE



Muhammad Ajmal Azad

University of Porto

58 PUBLICATIONS 1,359 CITATIONS

SEE PROFILE



Junaid Arshad

Birmingham City University

97 PUBLICATIONS 2,667 CITATIONS

SEE PROFILE



Urooj Waheed



University of Karachi

14 PUBLICATIONS 69 CITATIONS

SEE PROFILE

Article

A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT

Samia Masood Awan ¹, Muhammad Ajmal Azad ^{2,*}, Junaid Arshad ², Urooj Waheed ³ and Tahir Sharif ⁴

¹ Department of Computer Science, NED University of Engineering & Technology, Karachi 74200, Pakistan

² School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7BD, UK

³ Department of Computer Science, DHA Suffa University, Karachi 74200, Pakistan

⁴ College of Science and Engineering, University of Derby, Derby DE22 1GB, UK

* Correspondence: muhammadajmal.azad@bcu.ac.uk

Abstract: The connected or smart environment is the integration of smart devices (sensors, IoT devices, or actuator) into the Internet of Things (IoT) paradigm, in which a large number of devices are connected, monitoring the physical environment and processes and transmitting into the centralized database for advanced analytics and analysis. This integrated and connected setup allows greater levels of automation of smart systems than is possible with just the Internet. While delivering services to the different processes and application within connected smart systems, these IoT devices perform an impeccably large number of device-to-device communications that allow them to access the selected subsets of device information and data. The sensitive and private nature of these data renders the smart infrastructure vulnerable to copious attacks which threat agents exploit for cyberattacks which not only affect critical services but probably bring threat to people's lives. Hence, advanced measures need to be taken for securing smart environments, such as dynamic access control, advanced network screening, and monitoring behavioural anomalies. In this paper, we have discussed the essential cyberthreats and vulnerabilities in smart environments and proposed ZAIB (Zero-Trust and ABAC for IoT using Blockchain), a novel secure framework that monitors and facilitates device-to-device communications with different levels of access-controlled mechanisms based on environmental parameters and device behaviour. It is protected by zero-trust architecture and provides dynamic behavioural analysis of IoT devices by calculating device trust levels for each request. ZAIB enforces variable policies specifically generated for each scenario by using attribute-based access control (ABAC). We have used blockchain to ensure anonymous device and user registrations and immutable activity logs. All the attributes, trust level histories, and data generated by IoT devices are protected using IPFS. Finally, a security evaluation shows that ZAIB satisfies the needs of active defence and end-to-end security enforcement of data, users, and services involved in a smart grid network.

Keywords: smart cities; cyber security; Internet of Things; cyber-physical systems; zero-trust; ABAC; blockchain; IPFS



Citation: Awan, S.M.; Azad, M.A.; Arshad, J.; Waheed, U.; Sharif, T. A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT. *Information* **2023**, *14*, 129. <https://doi.org/10.3390/info14020129>

Academic Editors: Spyros Panagiotakis and Evangelos K. Markakis

Received: 12 January 2023

Revised: 9 February 2023

Accepted: 10 February 2023

Published: 16 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Industrial Revolution, benefiting from advancements within artificial intelligence, 5G, the Internet of Things, and blockchain technology has introduced a massive surge in technology inclusion, expansion, innovation, and research. Such paradigm shifts have highlighted the need for machine-to-machine and machine-to-human interactions where a huge amount of data transfer occurs during the process of communication devices setting up an Internet of Things network [1]. Alongside the need to transfer data at high speeds with low latency, the security of such systems is crucial due to applications dealing with sensitive user data or critical national infrastructure [2].

These types of massive communication handling require fool-proof security because whether the data come from home users or the data are being dealt with by any commercial

company, such as smart industries or smart grid stations, a security breach can risk multiple human lives or the unavailability of resources offered by critical cyber-physical systems [2]. Smart grids have long been a crucial component of energy networks, incorporating a variety of instruments, such as IoT devices, sensors and linked gadgets, that monitor and analyse the physical processes. It has aided in the optimization of energy production, distribution, consumption, and storage. In 2007, the sophisticated attack on Iran's nuclear power plant disturbed the distribution and development of the country's nuclear energy resources [3,4]. On 25 December 2015, in the midst of a civil war, an electrical power grid in Ivano-Frankivsk was hit by the cyber attacker that left 80,000 people without electricity and affected many critical services [5]. Hence, security is the most crucial aspect of these cyber-physical systems nowadays. With the increase in the number of technologies coming out, the security risks associated with them are also increasing exponentially. It is impossible for security systems to achieve 100% efficiency, and even military-grade technologies are somewhat vulnerable when they are attached to the Internet [6–8].

Hackers have several ways to compromise systems if traditional boundary security measures are deployed. Detecting an intrusion in such a setup becomes increasingly challenging if an attacker successfully breaches that parameter layer of defence. In contrast to these trust-based systems, authentication provides a way to present the credentials that the user or machine is the legitimate user of the network. The traditional authentication and authorization system might not be directly deployed in the IoT network because of resource limitations and the dynamic nature of the network. The network requires a dynamic policy-based system that enforces policies in real-time considering the user's constraints as well as the dynamic nature of the network [9]. Within this setup, a zero-trust (ZT) model applies some kind of policy decision to authorise every action of a user or device. Every attempt to access data or resources is verified by the organisation, hence common modern attacks make it very difficult for intruders to impersonate or masquerade as an authenticated device or authorized user. ZT promotes a host-based monitoring approach where every host or owner device gets to set the criteria required to access it. Fine-grained data access control allows the host to dictate the intended audience and makes sure that the data cannot be accessed by any undesired user. Hence, ZT opens up new ways for security-enabled collaboration opportunities between organizations. On the implementation side, only regular updates of new technologies and methodologies with the pace of research and innovation, such as ZT, ABAC, and blockchain, can support a system to become less vulnerable against intruder attacks.

With the changing environment of networks and the new ways of communication among devices, a lot of effort is required to manage network security in real-time in a dynamic environment. The best practices of cybersecurity are becoming obsolete with the passage of time and new approaches are coming into the realization stage with every passing day. The issues associated with network security are no longer general, and the same policy and standards for each network cannot be replicated for every network. Therefore, aggressive encounter measures are required that not only support the network as a gatekeeper but also secure the system from malicious activities [10]. Access and authentication policies should be uniform at one end but must also be dynamic to reduce the vulnerabilities within the network in real-time. The Internet of Things deals with machine-machine and machine-human interactions over the Internet and blockchain is a distributed ledger primarily available for tamper-proof, hack-proof, and immutable recording of transactions into the ledger [11]. The combination of the IoT and blockchain-based networks somehow sorts out the problems associated with the domain. Similarly, access control generally implemented through MAC address, IP, and other tags is not sufficient. A modern and evolved approach is required to deal with network security [12]. In this paper, we have introduced a method to make security as efficient as possible compared to conventional ways. We proposed a system that ensures the security of IoT devices and users through the use of emerging concepts of zero-trust architecture, attribute-based access control, blockchain, and IPFS. The system will be used to sustain a

network and communication as efficiently as possible to reduce real-time attacks through the implementation of real-time monitoring, dynamic policy generation mechanisms, and interminable monitoring of the various aspects of network security and communication.

Approach

A novel approach is required to deal with the above-mentioned challenges, the advanced technologies and techniques will play a vital role to get this job done. For example, the use of blockchain technology can secure the data and allow only recognized participants to join the network [13]. Similarly, a dynamic policy mechanism is required to create, authenticate, and recognize participants (IoT devices) in the non-trusted environment [14]. Each node/participant must authenticate first, before interacting with the system or the participants associated with the network. The non-trusted environment will reduce the chances of hackers exploiting the network by masquerading, man-in-the-middle attacks, and brute force attacks, which are the most commonly used techniques to attack the network [15]. The creation of blockchain wallets for each participant (each IoT device) at the time of registration of the new device will help to recognize and record the device details better in an automated way using smart contract technology, and IPFS secures all the information about the devices and the data generated by these IoT devices for any further pre-processing [16]. The ZTA is associated in such a way that the entire system has multiple divisions, and each division has its categories and priorities. Each division may be called a “Zone”, and must have a PEP that enforces all policies within the network zone, and also routes all device communication requests through to the PDP where decisions are made to accept or reject the request, and when accepted it will create encrypted channels to entertain the interactions [17]. These PDPs are further connected to the PE that generates dynamic access policies. Due to the nature of IoT, devices are multivariant; therefore, the policy-making must be designed as per the variable attributes of the object and the subject of a certain request, hence the attributed-based access control should be introduced on top of ZTA [15].

The rest of the paper is organized as follows. Section 2 discusses the background of zero-trust architecture and the challenges for designing such architecture for IoT networks. Section 3 discusses related work in this domain. Section 4 provides the system architecture of zero-trust-based access control. Section 5 defines policies and Section 6 defines device attributes and management. Section 8 evaluates the approach and Section 9 concludes the paper.

2. Background

Zero-trust architecture (ZTA) is a practical implementation of the concept; trust is nothing but a vulnerability when it comes to network security [18,19]. The notion of zero trust is centred on network segmentation into “Microcores and Perimeters” (MCAP). Instead of having a trusted domain built by a perimeter around the network, ZT suggests everything, and everyone is untrusted even within the network perimeters [17]. Hence, it promotes the “never trust and always verify” principle even within the enterprise network.

ZTA divides the entire network into microcores and sets perimeters around each core. ZTA implementation includes a policy engine (PE) which generates access control policies, a policy administrator (PA) that evaluates a request to access an enterprise resource by applying the policies generated by PE, and a policy enforcement point (PEP) that enforces these policies by accepting or rejecting the received request as per the decision made by the PA. The core components of ZTA are shown in Figure 1.

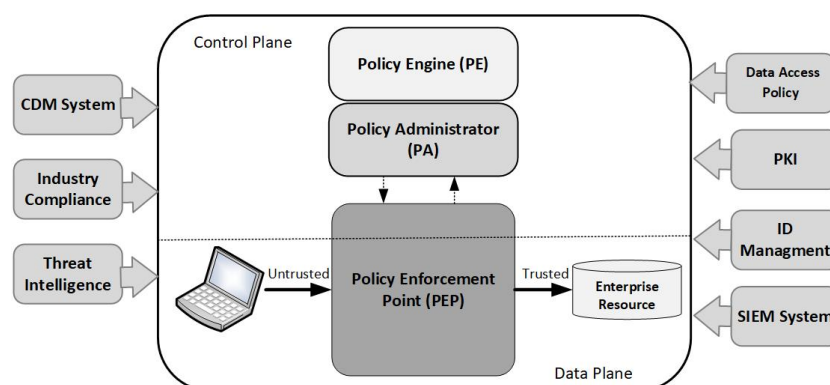


Figure 1. Core zero trust logical components.

Setting perimeter-based security is neither efficient nor possible for IoT networks as they are distributed in nature. Therefore, ZT is the perfect solution for all IoT security problems [1]. ZT provides a complete scheme for guaranteeing users access across amalgam infrastructures and networks through smartphones, computers, cloud applications, and other IoT devices [11].

2.1. Challenges

Network security, which deals with the high-volume data traffic from multiple sources, such as millions of IoT devices, is the top priority for any enterprise. The conventional ways of securing networks, such as firewalls, access control, etc., are becoming obsolete day by day and new innovative methods are required to deal with network topology, IoT device management, and overall the entire process of end-to-end communication, data storage devices, and data management [6]. The challenges presented in IoT networks are real-time monitoring, data handling, data storage, access management, trust-based or trust-less criteria to deal with the participant nodes and their behaviour, accessibility, roles, modes, and parameters and factors compromising security in real-time monitoring and management [2].

Some challenges that need to be addressed to enhance system security are as follows:

- **Authentication** refers to the verification of user credentials. Robust authentication mechanisms are required to identify valid users from ill-intended impersonators who try to gain illegal access to IoT devices and their data. Therefore, all the users and IoT devices should be registered, and their baseline behaviours need to be analyzed for instant detection of any behaviour anomalies such as impersonation or masquerade attacks due to the illicit use of valid user credentials.
- **Authorization** is another key aspect of IoT networks. The device owners have comprehensive rights and complete authority over the data generated by it, hence the type of access (read/write) to any device and its data can be granted or revoked based on the criteria set by the owner. Successful implementation of generic access control policies that evolve dynamically based on the current scenarios is one of the key challenges for huge IoT infrastructures.
- **Confidentiality** can be defined as the protection of system resources against unauthorized access. The degree of authorization required to access devices and data in an IoT network needs to be set intelligently in order to maintain the confidentiality of the classified information. Smart cities have every aspect of human life being connected and controlled with IoT devices. A data breach may result in life-threatening situations as sensitive information, such as the daily schedules or healthcare records of citizens, needs to remain concealed for their safety and well-being.
- **Privacy** means having full control or decision-making authority on how the user's data can be collected and used. Users have the right to protect their personal information, such as their daily schedules and medical and financial records, from being revealed

without their consent. Hence upholding the privacy of data generated by handheld gadgets, surveillance devices, or home IoT networks is one of the most important goals for any smart city infrastructure. The proposed framework needs to address all of these challenges and provide efficient solutions for them.

2.2. Attribute-Based Access Control

In attribute-based access control (ABAC), any access request is approved based on the attributes of the subject and the object. The identities, roles, functions, and other complex features of a subject are all posited as its attributes [17]. The attributes of the requester (subject) and the attributes of the requested (object) are both combined to form an access control policy to fulfil the security demand of the object's owner. Specified access policies are set by the object owners to determine whether the subject has appropriate privileges for the requested access based on these attributes [20]. Figure 2 shows the policy creation logic and components of ABAC.

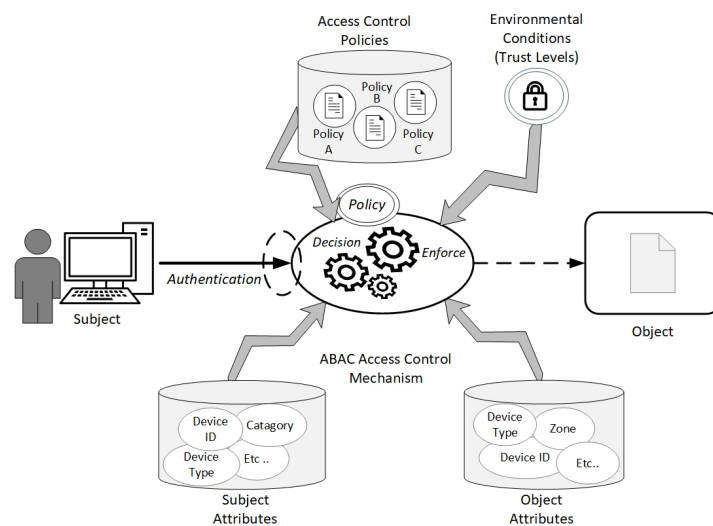


Figure 2. ABAC logical components.

Traditional access control models, such as discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC), are completely centralized. Based on the distributed, decentralized, and dynamic architectures of IoT networks, attribute-based access control (ABAC) is regarded as the most suitable approach for IoT scenarios. ABAC provides the strong dynamicity, scalability, and flexibility required for IoT environments to implement fine-grained control over the access requests for every device [21].

2.3. Interplanetary File System

Another vital component of our proposed system is the InterPlanetary File System (IPFS). It is a file system where data are stored in the form of uniquely identifiable blocks by multiple peers following a distributed approach [22]. The IPFS maintains all versions of a file as separate individual blocks and cryptographic hashes are assigned to each block as a unique identifier, which means no two different blocks in the system can have the same cryptographic hash. While searching for some content, it is located and accessed by its assigned hash value [16]. In a huge IoT infrastructure with millions of devices, the amount of data generated by these devices cannot be stored on-chain and large off-chain storage reserves are required. IPFS provides a distributed and secure solution to all storage issues as it has such a massive data storage capacity [23].

3. Related Works

With the vastly used cloud applications and IoT networks, traditional network security approaches such as building a wall between trusted and untrusted devices and the trusted local networks do not work anymore. The need for secure and smart access control where no trusted networks or devices exist has been fulfilled by ZTA. Various variants of ZT have been proposed and implemented by researchers to satisfy their network's unique security demands. In [24], Pedro Assunacao discussed a ZTA that eliminates static credentials, applies multi-factor authentication, and maintains a log of all devices and network traffic.

In [25], authors suggested context-based ZT access control to overcome the challenge of a secure and heterogeneous Moodle application. This framework is an application of a model that provides access control to an e-Learning platform called Moodle. Through the implementation of the zero-trust model, a positive performance on the webserver has been seen. However, to evaluate the non-functional performance of the zero-trust model, additional tests need to be carried out.

With the scalability of IoT networks, there is a need for trust management controls that could safeguard the systems against malicious attacks. To overcome this, a centralized validation mechanism is required. The authors in [11] have presented an IoT-based zero-trust model which enables a novel hierarchical mining concept. According to the authors, IoT infrastructure is a zero-trust model, and cannot be trusted. To overcome this, a blockchain-based middleware, Amatista, has been introduced. The mining platform integrates distributed validation authorities for the IoT at different levels of trust. Firstly, context-based mining has been introduced. Secondly, publish/subscribe provisioning of data has been introduced. Amatista has been evaluated using IoT sensors and edge networks, and it has been concluded that the system can handle not only the infrastructure but the transactions as well.

In [19], authors have suggested a similar mechanism in the context of smart cities. The authors have applied the network classification to extend the idea of zero trust. As per ZTA, the framework divides the system into separate MCAPs for web access, mobile access, and database, and a blockchain node is attached to each MCAP for request authentication. The IoTs are also divided into eight different categories based on their risk analysis calculated on three factors, i.e. network capability, risk score, and data risk. All the MCAPs are connected by a segmented gateway. Every time an access request is generated, the blockchain node attached to the targeted MCAP verifies it using a smart contract, and access is granted if the request is considered genuine and is verified. Although the system has no implementation in the real-life IoT network, the authors claim to address multiple security concerns including the risk-based MCAPs for IoT devices. However, the IoT data transferred over the network are not secured.

Chen and Qiao in [1] implemented a smart healthcare system for 5G networks where they divided their entire system into four dimensions, i.e., object, subject, environment, and behaviour. Fine-grained access policies are being defined by using machine learning and deep learning that use real-time threat and trust levels generated for all IoMT devices and users. In Fabric-IoT [21], the authors have implemented hyper-ledger Fabric-based access control for IoT using smart gateways where every IoT device sends its encrypted data to a URL. The framework comprises three smart contracts: one creates, modifies, and deletes policies, the second one assigns URLs to IoT devices, and the third one enforces access control. Nevertheless, to achieve better performance, the system's scalability needs to be improved.

In [26], the researchers implemented attribute-based access control on IoT sensor data by using a rule-based proactive engine which helps to generate new rules and policies, monitors the environment, and helps the PDP decide on what to do in case of any sudden changes by creating a behaviour baseline saving all the previous transactions in the PIP database. However, details on how these policies will be implemented need further description. In [27], a secure IoT system using ABAC is implemented by IPFS and smart contracts. All transactions generated by IoT devices along with all the policies are saved in

the IPFS database in the form of hashed blocks. When a user sends a request to access any IoT device's data, the PDP requests attributes from PIP and policy rules from PAP, matches the two, and decides whether to grant this access or deny it.

In [28], to preserve the privacy of e-health data authors have proposed using blockchain with non-interactive zero-knowledge proof-based key authentication to manage the device authentication process for millions of medical things joining the network. While the process ensures that no unauthorized device joins the e-health network, it does not deploy any dynamic mechanism to identify intrusions caused by device compromise after they have completed their authentication process.

The authors in [29,30] have presented a blockchain-based data provenance system which uses the Merkel chain blockchain property to maintain a chain of custody for data. While the system maintains complete data access logs, it does not provide any trust management to handle the dynamics of machine-to-machine communication. Analysing a baseline machine behaviour can help to revoke access whenever a machine misbehaves. Through a complete analysis of related works as mentioned in Table 1, we discovered that most of the already proposed architectures either lack proper dynamic policy generation, which allows the systems to automatically create new policies for previously unseen situations, or they lack a completely decentralized architecture where any IoT device can get its request instantly processed by any available node without any delay or a centralized authority being involved in the process [31]. Even if we use secure and anonymous device authentication using zero-knowledge proofs, it does not guarantee that the device will remain uncompromised [32]. In this paper, we have proposed an architecture that is truly decentralized, completely dynamic, and will process every request based on the current environment and the behaviour of the IoT device instead of old-school role-based or identity-based authentication for processing access requests. The Internet of Things (IoT) has introduced a smart lifestyle. Home appliances, power plants, vehicles, and healthcare, we are aiming to automate the entire metropolitan infrastructure [33]. With all our systems connected with the IoT, the most crucial concern is if we can trust the current IoT authentication and access control infrastructure after connecting it to millions of “things” [34]. The current systems are not trustworthy enough to have our lives depend on them. The following design is inspired by ZTA and helps to curtail the mistrust of the current systems.

Table 1. Analysis of existing approaches to zero trust for IoT.

Paper ID	IoT Domain	Utilized Techniques	Contribution	Limitations
[1]	IoT in Healthcare	5G, Zero Trust, Attribute-based Access Control	The system uses trust assessment and risk level of objects to dynamically grant access based on attributes and performs traffic monitoring, load matching, access control, and auditing by using ML and DL.	Specific to healthcare scenario and therefore focused on access to resources rather than communication requests.
[11]	Hierarchical Management in IoT	Blockchain	Introduced a novel hierarchical mining the concept of using twotier miners for contextbased validation.	Other hierarchies of validation should be introduced as consensus on twotiers is time expensive. Authors should also include more specialized smart contracts for IoT.
[19]	Securing IoT devices using Zero Trust and blockchain	Zero Trust, Blockchain	The proposed framework divides the system into separate MCAPs and it uses risk factors to categorize IoT devices into different zones. Blockchain nodes are attached to each MCAP for request authentication.	All data is to be stored in blockchain where transaction per second rate is very slow and the management server is a single centralized server that defies the decentralized nature of the proposed model.
[21]	Fabric-IoT: A blockchain-based Access Control System in IoT	Hyper-Ledger Fabric, ABAC	Using smart gateways, Fabric IoT uses a hyper-ledger-based approach to implement ABAC.	Scalability is the biggest limitation for fabric-IoT along with minimal support for IoT application integration.

Table 1. Cont.

Paper ID	IoT Domain	Utilized Techniques	Contribution	Limitations
[25]	Context-based Access Control and Trust Scores in Zero Trust Campus Networks	Zero Trust	Secures the Moodle Application for university-wide open and heterogeneous research network using zero trust	It lacks policies for the policy engine as well as for trust score metrics.
[35]	FairAccess: a new Blockchain-based access control framework for the Internet of Things	Bitcoin-based Blockchain and OrBAC	Secures the IoT devices by using identity-based and permission-based access control policies	The approach does not analyse the dynamic IoT device behaviours and hence is not ideal for evolving scenarios of machine-to-machine communications.
[36]	CES Blocks—A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT	Consortium Blockchain, ABAC	Secures the IoT devices using ABAC and records all attributes and requests as blockchain transactions by using a simple hash and signature protocol	It lacks creation of new policies along with calculation for device trust scores.
[26]	Context-aware and Attribute-based Access Control Applying Proactive Computing to IoT System	ABAC access control, rule-based proactive engine	Implemented ABAC on IoT sensor data using rule-based Proactive Engine which helps to generate new rules and policies, monitors the environment, and helps the PDP to decide what to do in case of any sudden changes by creating a behaviour baseline, saving all the previous transactions in the PIP database.	The paper only discusses the data received from IoT sensors and actions to be initiated based on this data but does not mention how users' access requests will be entertained.
[27]	IoT architecture based on ABAC smart contract	ABAC, IPFS	A secure IoT system using ABAC is implemented by using IPFS and smart contracts. All transactions generated by IoT devices are saved in the IPFS database along with all the policies in the form of hashed blocks based on which it is decided whether to grant access or deny it.	Static ABAC policies do not consider environmental or behavioural attributes while granting any access control request.
[37]	Securing Home IoT Environments with Attribute-Based Access Control	ABAC access control, NIST NGAC	The proposed framework suggests securing IT devices by using ABAC policies by defining attributes for Subject, Object, and Network.	Uses a set of predefined policies and no new smart dynamic policies can be made by the system at run-time to counter a new undefined scenario.
[38]	BlockShare: A Blockchain-Empowered System for Privacy-Preserving Verifiable Data Sharing	Blockchain, Zero-Knowledge Proof	Uses a newly defined data structure to store all e-health records for sharing.	While the approach emphasises anonymous data sharing, it does not consider access control and hence is not suitable for D2D communication.
[28]	Preserving Privacy in Mobile Health Systems Using Non-Interactive Zero-Knowledge Proof	Blockchain, ABE, Non-Interactive Zero-Knowledge Proof, IPFS	An authentication scheme that is lightweight enough to run on e-Health devices with minimal resources to provide a secure device authentication mechanism.	It does not detect a compromised device once it has completed the secure device authentication process.

4. System Architecture

The framework contains a consortium blockchain network and a network of IoT devices whose attributes serve as policy components for policy creation and implementation processes. A block architecture of the proposed ZAIB system is shown in Figure 3.

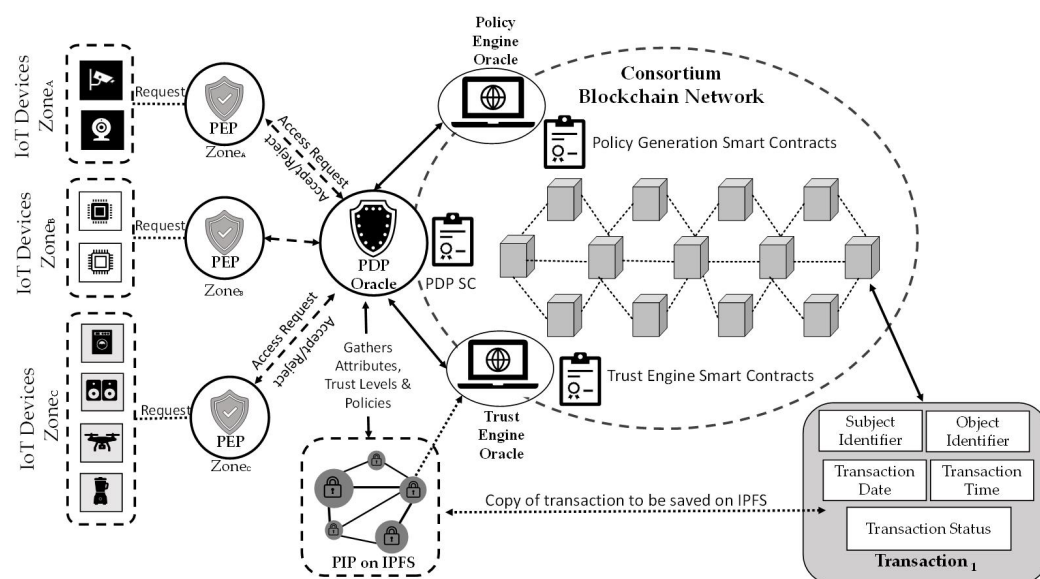


Figure 3. Architecture of the ZAIB framework.

A blockchain component is added to ZAIB for facilitating different IoT devices to communicate freely, securely, and anonymously on the network. To ensure device and data security, ABAC access control mechanisms are being implemented through smart contracts for device communication management. The policy engine (PE) oracle receives requests to make new policies and triggers the policy engine smart contract (PESC) to make new access policies for ABAC. To save IoT device attributes for ABAC, the data generated by them, all the policies generated by the policy engine (PE), and trust-level histories for device behaviour analysis, we are using IPFS. IPFS stores large-sized files easily, hence small-block-size issues are resolved. IPFS provides secure storage due to automatic resource mapping and hashed data; it is also connected with smart contracts, hence the authenticity of all information stored on the IPFS can be checked by comparing it with the transactions stored on the blockchain ledger. To implement zero-trust architecture, a trust engine oracle triggers the trust calculation smart contract which calculates the trust level of different devices considering several factors from their behaviour history stored in the ledger. Lastly, PDP smart contracts approve or reject IoT device requests for device-to-device communication.

4.1. Device Registration on Blockchain

Blockchain is a vital component of the system as it provides anonymous and secure D2D communication using smart contracts and its immutable distributed ledger [39]. Cryptographic key pairs provide the security feature in blockchain wallets. On registration of every new IoT device, an account is assigned to it to call contracts or initiate transactions. The system architecture of registration of new IoT devices is shown in Figure 4. The device's authentication and transaction anonymity are ensured due to the blockchain wallet [40]. The PBFT consensus algorithm is selected, as the frequency of requests is very high, and consensus needs to be reached very quickly. All the device attributes are saved in PIP which is implemented as an IPFS storage, and device management smart contracts are installed on the device.

Whenever an IoT device generates a communication request, the PEP acts as a gateway to pass it along to the PDP oracle, which triggers the PDP smart contract, hence recording this request as a transaction on the ledger. The PDP smart contract checks if any policies regarding such a request exist and decides to accept or reject the request based on these policies and the decision is recorded as a transaction. If it is found that such a request does not exist, it generates a request to create a new policy and triggers the PESC; this transaction is also recorded on the chain. Whenever a request is processed, the trust level smart contract is triggered and the new trust value for the IoT devices is saved both as a

transaction on the chain and also in the PIP. A hashed link of all the data, trust levels, and policies stored in a block on the IPFS-based PIP is also stored in the chain which later on can be used for data validation.

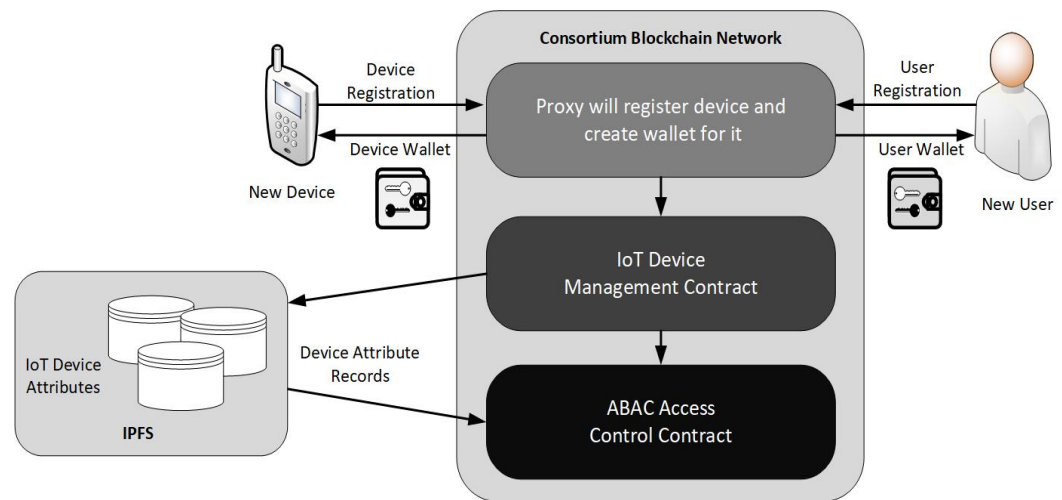


Figure 4. New IoT device registrations.

4.2. Hashed Storage of IoT Data Using IPFS

In our proposed system, the IPFS is responsible for storing the attributes of all connected IoT devices, smart contracts, access policies, trust level history for all the connected devices, and the data generated by our IoT devices in a very secure manner. Data generated by IoT devices, even the audio, video, and images can be encrypted and stored in blocks [30]. At any instance, the authenticity of the policies or trust levels stored in the IPFS can be checked by comparing the IPFS hashed blocks with the on-chain transaction to ensure that data or policies on IPFS have never been tampered with or corrupted. Figure 5 presents how data is stored in our blockchain system.

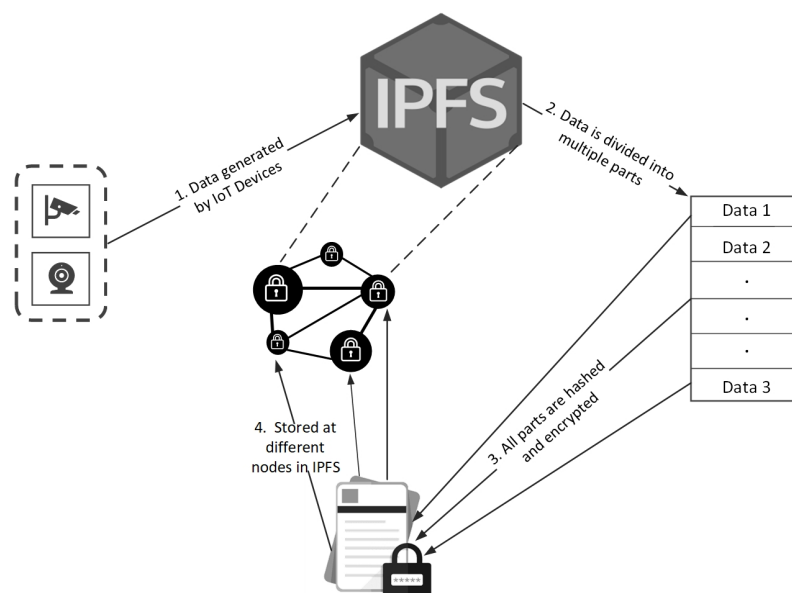


Figure 5. Storage of IoT data.

4.3. Zero-Trust Architecture

Since there are no trusted devices, trusted systems, or trusted users, all access and device-to-device communication requests need to be monitored and granted only when they are tested as valid access requests [15]. The integration of ZT in the IoT network

and all its connected devices to provide complete and utmost security requires the key features of ZTA, such as micro-core, perimeters, and trust calculations, to be added to the infrastructure [17].

4.3.1. Zone Division

To implement ZTA, our entire IoT network is divided into different micro-cores, called “Zones”. On any network, IoT devices can be categorized into zones based on their physical location, device categories, and priorities [41]. For example, if ZT is applied to a smart home, various similar category devices can be grouped to form different zones, for example, all kitchen appliances, which may include microwaves, refrigerators, coffee makers, juicers, blenders, etc., can be assigned a separate zone. Similarly, a home surveillance and security devices zone may contain all the cameras, smoke detectors, and smart locks. When applied to a huge smart edifice such as a smart city, separate zones can be identified in every division of the metropolitan infrastructure [42].

Each zone has its own policy enforcement point (PEP) that receives every communication request from all the devices and routes it to the connected policy decision point (PDP) which makes all policy decisions and accepts or rejects them based on the policies defined by the policy engine (PE). If a request is accepted, the PEP creates an encrypted channel to facilitate IoT device interactions.

4.3.2. Policy Enforcement

Each policy decision point oracle (PDPO) has multiple PEPs attached to it. To make decisions for all requests submitted by the PEPs, the PDPO reads the policies and device attributes from the PIP and the current trust levels of each device from the TE. It makes continuous dynamic decisions to accept or reject the request by putting the acquired data depicting the run-time status of the system, the network, and its involved devices into the policy. If no suitable policy is found to review the present request, it demands PEO to generate a new policy for the current scenario as shown in Figure 6.

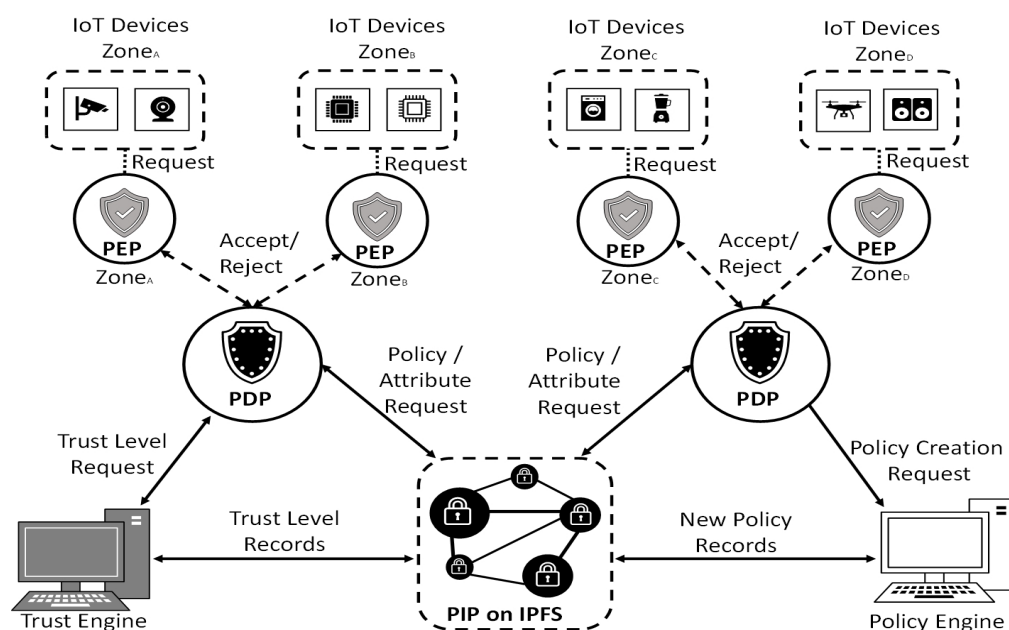


Figure 6. Zero-trust architecture for IoT devices.

The policy engine (PEO) generates new policies when the PESC is triggered, these policies are based on the set of policy frameworks provided by the network administrator dynamically. A detailed description of the policy creation process of these policies and the basic guidelines to be followed during this policy generation is shown in Section 5.

4.4. Trust Engine

One very important component of the system is the trust engine (TE) that calculates trust levels for IoT devices on the network [25]. The TEO is connected to the PDPO to provide updated trust levels of subject (S) and object (O) IoT devices for policy evaluation as shown in Figure 7.

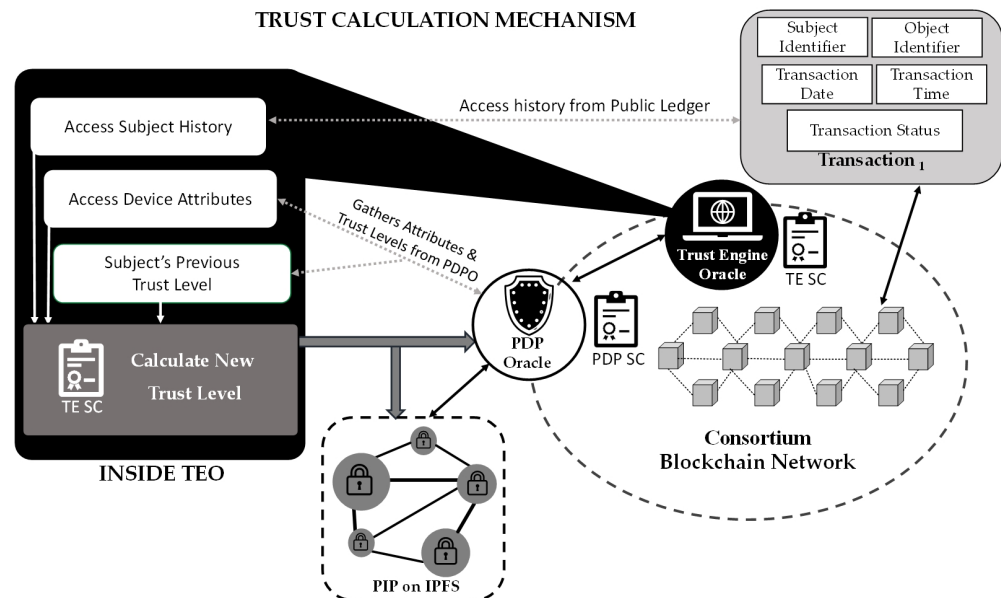


Figure 7. Trust calculation mechanism.

Possible Inputs for Trust Score Calculations

The most important feature for trust calculation is behaviour analysis. This behaviour analysis is carried out by the TESC by accessing the request history of devices from the PIP [43]. The device access history helps to determine the baseline behaviour for each device which is then saved periodically in IPFS storage [1]. A new trust score is generated for each device by comparing its current behaviour with its baseline behaviour [25]. A drastic change in behaviour results in a decrease in its trust level, whereas a persistent behaviour increases the trust level of the device, as shown in Figure 8.

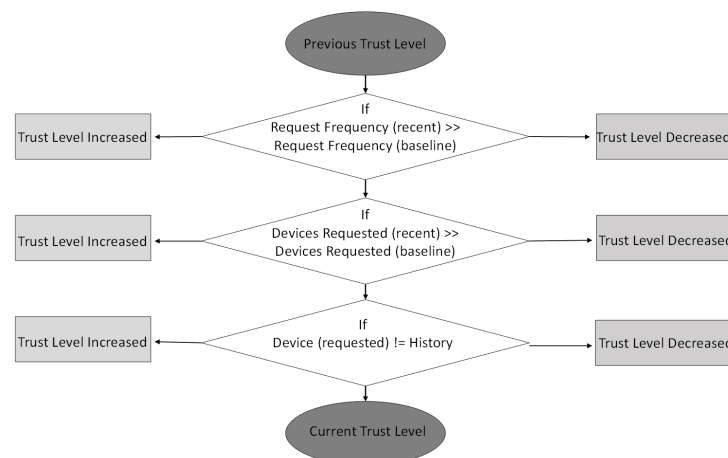


Figure 8. Suggested trust level calculations.

4.5. Access Control Model for Device-to-Device Communication

The decision-making policies for granting or denying a certain device-to-device communication request are generated using the ABAC model based on the features discussed below.

4.5.1. Attributes of IoT Network

Due to the multivariate nature of possible IoT interactions, attributes play a vital role in the decision-making process. Policies are designed based on these variable attributes of both the object and subject of a certain interaction request [26]. This section defines and describes the attributes for object and subject IoTs, the basic entities for ABAC. The next phases define how these attributes inhabit the policy information point (PIP) and how the policy engine (PE) uses these attributes to create new policies.

Every ABAC request must have a subject (S) that initiates the request, an object (O) that is the device the subject wants to commence communication with, a nature (N) which represents the nature of communication, and an environment (E) which represents the network at the time of generation of the request [37]. Based on these factors, the ABAC request format can be represented as

$$\text{IoT Access Request} = \langle \text{Subject}(S), \text{Object}(O), \text{AccessType}(A), \text{Environment}(E) \rangle$$

A received request can be allowed or denied based on the ABAC policies for the combination of attributes of the subject (S), object (O), the access type (A) of the communication request, and the environment (E). We described various key attributes that can be examined by ABAC policies.

Device Attributes: An IoT device can be a subject if it initiates the data request from another device or it wants to transmit data to another device. The object is the device the subject wants to communicate with [44]. Conventional device attributes are

- *DeviceIdentifier*: =The unique blockchain wallet id assigned to every IoT device.
- *DeviceType*: =The devices can be categorized into different types, such as smart TV, cameras, drones, sensory devices, and smart vehicles.
- *DeviceAge*: =The number of days since the device was first registered on the IoT network.
- *DevicePriority*: =Devices may be assigned different priority levels depending on the sensitive nature of their data and the security clearance level required to access them.
- *DeviceTrustLevel*: =Device trust level is to be calculated by the TESC based on the device's previous behaviour and its request pattern on the IoT Network.
- *DeviceCategory*: =Devices can be categorized as entertainment, healthcare, controllers, surveillance, monitoring, diagnostic, etc. A certain category of devices can be allowed to communicate with each other or with devices from other categories.
- *DeviceZone*: =Every device is assigned a zone or group once it registers on the network. Before a certain age and trust level is achieved, a device can only communicate with the devices in its zone.
- *DeviceLocation*: =The physical location of the device can also be stored as some of the policies can depend on the proximity of the devices.
- *DeviceStatus*: =Once a write connection is established with a monitoring device, the object device enters a locked status for all other write requests.
- *NetworkIdentifier*: =For some devices that might need the network identifier, this attribute will be a combination of sub-fields such as IP address and subnet mask.

Subject (S) is an IoT device requesting to initiate communication with an IoT device object (O), both IoT devices will have all these attributes assigned to them. Based on the values assigned to these attributes of subject (S) and object (O), different policies will be generated to accept or reject a generated communication request.

Access Type (A): The operations permitted on an IoT device are accessing the data from the devices or sending new control messages to the devices. The sending of new control messages can also be called a "Write" operation while accessing the device data can be termed as a "read" operation [45]. The nature of access demanded by a user is mentioned in this access type (A) field. The attributes for Access_Type are:

1. read: =data size
2. read_all: =data size

3. write: =size of message
4. write_all: =size of message

Environment (E): The environment for any communication is the network itself; these are the external parameters for both the subject (S) and the object (O). An example of environmental attributes that need to be recorded are date and network time, assuming that the standard synchronization policies such as network time protocol (NTP) are in place [44].

4.5.2. Attribute-Based Access Control Policy Model

Numerous kinds of data can be generated by IoT devices. For example, cameras capture videos, a microphone captures sound, and sensors capture humidity, temperature, and light. All of these features are very important and if someone with malicious intent gets access, it might even end up putting lives at risk [21]. To ensure that only trusted devices can communicate with IoT devices on the network, ABAC access policies are implemented. The establishment of a connection between two devices is shown in Figure 9.

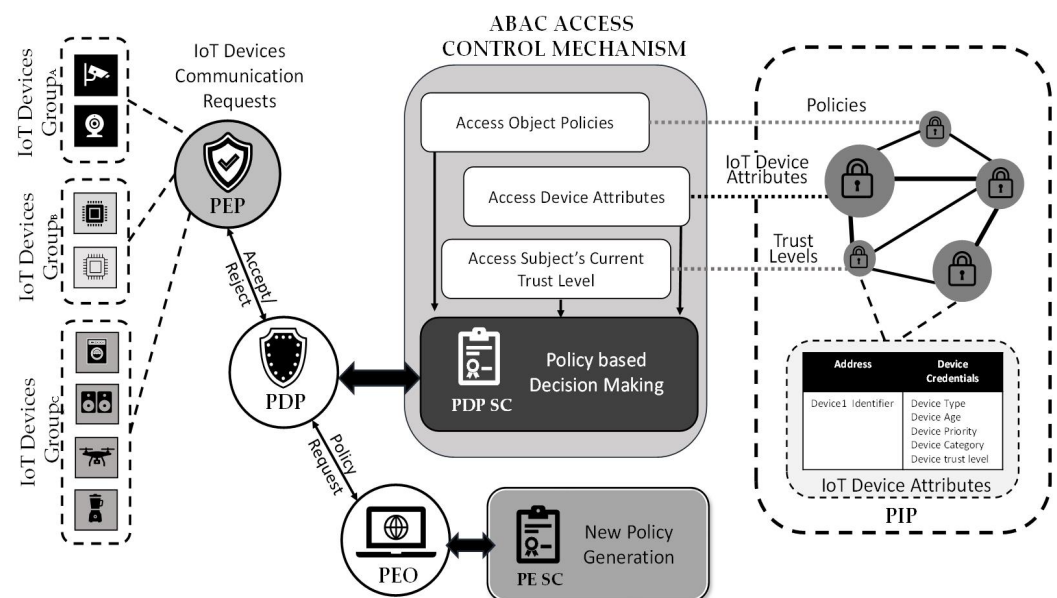


Figure 9. ABAC access policy implementation in ZAIB.

A brief description is also shown in steps 1 to 5.

1. Subject (S) requests to initiate communication with object (O).
2. The request is received by the smart gateway.
3. The request is forwarded to PDP.
4. PDP requests PIP for attributes of both subject (S) and object (O).
5. Based on device type, category, priority, and current trust levels (provided by the trust engine), the policy engine decides to accept/reject the request.
6. The PDP enforces the decision made by PE and, if access is granted, establishes a secure encrypted channel for safe D2D communication.

Based on the steps discussed previously, the ABAC device access policies can be defined as mentioned in Algorithm 1:

Algorithm 1 An algorithm for policy

Require: $Policy = Subject_{attributes}, Object_{attributes},$
Require: $Subject_{attributes} =$ Device Identifier, Device Type, Device Age, Device Priority, Device Category, Device Zone.
Require: $Object_{attributes} =$ Device Identifier, Device Type, Device Status, Device Priority, Device Category, Device Zone.
Require: $Environment_{attributes} =$ Date, Time
Require: $TrustLevels =$ Subject Trust Level, Object Trust Level, Network Trust Level
if $Permission == 0$ **then**
 $AccessGranted$
else if $Permission == 0$ **then**
 $AccessDenied$

5. Policy Creation Framework

To generate automated policies, some ground rules have been set that help the PE in its policy-making activities. The policies required for the efficient and secure operation of an IoT infrastructure can be characterized as follows:

5.1. Device Acceptance Policies

Whenever a new device joins the network, it needs to be characterized and the current devices on the network need to be protected from it until it is verified as a trusted and safe device [46]. These policies should be independent of the specific device features for them to be applied to all kinds of devices. After running the diagnostics for the security state of the device and registration of the device, it will take some time to communicate with a few devices slowly and gradually as it ages and the baseline behaviour remains consistent to build up the trust level, only then can the new device request to communicate with highly trusted devices. To secure the network against a new device, some generic policies should be defined. A few sample protective policies for limiting access are provided in this section.

Sample Policy 1: *A new IoT device cannot request communications with more than a certain number of devices in a specific acceptance time.*

The new IoT device is the subject (S), another IoT device is the object (O), and the generation of object access requests by the subject is the desired action. The environment time and the registration time of the device help to calculate the age of the device on our network. A specific time interval is set as device acceptance time during which a new device can only access a limited number of devices. This policy ensures that the new device does not try to access and communicate with all devices on the network before it gains a certain trust, and that its security status is checked. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Age }
- Object: { Device Type, Device Identifier }
- Environment: { Date, Time }

Now, such a policy is very generic and essentially captures the security essence.

Sample Policy 2: *a new IoT can only communicate with devices in its zone until it reaches a specific age.*

The new IoT device is the subject, another IoT device is the object, and the generation of object access requests by the subject is the desired action. The purpose behind setting such a policy is to ensure that the new device does not try to send broadcast messages to devices across all different zones. This policy also helps in the development of a baseline behaviour of the devices and helps limit access to all zones until a certain age and trust level is achieved by this new device. The attribute fields critical for the mentioned access policy are

- Subject { Device Identifier, Device Age, Device Zone }
- Object { Device Identifier, Device Zone }
- Environment { Date, Time }

5.2. Device Access Policies

Access to every IoT device cannot be granted to every other IoT device if it generates a request. The access limitation policies make sure that only valid requests get accepted while all other requests get rejected. Attributes such as device type, device category, and device priority of both subject and object are considered while creating these access policies [37]. A device access request is accepted only when the subject has a certain priority, and trust level that matches the object, and the device category allows the kind of access type of the generated request. Some of the sample access policies are defined as

Sample Policy 3: An IoT device can only communicate with another IoT device if it matches the priority combined with the trust level required to access that device.

Here an IoT device is the subject, another IoT device is the object, and the acceptance of object access requests by the subject is the desired action. This policy allows access to an IoT device if and only if the subject has a combined value of its priority and current trust level greater than or equal to that of the object device. As the write access is granted to a monitoring device, the device status of the object is set to Lock. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Priority, Device Trust Level }
- Object: { Device Identifier, Device Priority, Device Trust Level }
- Access_Type: { read }
- Environment: { Date, Time }

Sample Policy 4: only monitoring type devices can not send control data to any other device.

Here, one IoT device is the subject, another IoT device is the object, and transmission of a control message is the desired action. We do not want any device to be able to change the settings of another IoT device unless it is an authorized and trusted monitoring device. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Priority, Device Trust Level }
- Object: { Device Identifier, Device Status, Device Priority, Device Trust Level }
- Access_Type: { write }
- Environment: { Date, Time }

Sample Policy 5: an IoT device can receive control data from only one monitoring device at a certain instance of time.

Here, an IoT device is the subject, another IoT device is the object, and transmission of a control message is the desired action. While a monitoring device has established a connection with an IoT device and is sending some control instructions, hence changing the other device's settings, no other device should be allowed write access to such an object. We do not want multiple devices to be able to change the settings of another IoT device simultaneously. Hence, the device status of the object is checked whenever a written request is received, and the request is set as pending until the device is set free and its status is turned back to unlock. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Priority, Device Trust Level }
- Object: { Device Identifier, Device Status, Device Priority, Device Trust Level }
- Access_Type: { write }
- Environment: { Date, Time }

5.3. Device Access Limitation Policies

Countermeasures need to be taken to avoid the possibility of any flooding attack. No device should be allowed to access all the devices available on the network simultaneously. In ZTA, the traffic is monitored and zones are mentioned; hence, a rogue device that tries to initiate broadcast requests is sent to a quarantine zone where the device is reset, hence setting its age back to zero and a full scan of the device's security status is performed to detect the cause of such malicious activities. To achieve this goal, the trust level of a device is decreased as it initiates any broadcast request. To enforce a guaranteed rejection of random access requests, some rules can be set to create regional broadcast boundaries as follows:

Sample Policy 6: *A monitoring device can send control data to multiple IoT devices at a certain instance of time if they all belong to the same zone.*

Here a monitoring IoT device is the subject, a set of multiple IoT devices in a certain zone is the object, and transmission of a control message is the desired action. Monitoring devices can establish multiple concurrent write connections with other IoT devices and send some control instructions if and only if they all belong to the same zone. We do not want a change of control settings for multiple devices in multiple zones to occur simultaneously. Hence, the device zone of the object is checked whenever a “write-all” request is received. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Priority, Device Trust Level }
- Object: { Device Identifier, Device Status, Device Priority, Device Trust Level, Device Zone }
- Access_Type: { write_all }
- Environment: { Date, Time }

Sample Policy 7: *Only a controlling/monitoring device can initiate connections to all devices in a zone simultaneously.*

Here, a monitoring IoT device is the subject, a set of all the IoT devices in a zone is the object, and transmission of a control message is the desired action. To ensure the security of our IoT network, only supervising devices are allowed to have concurrent access to all devices in a certain zone. This guarantees that no rogue device is allowed access to multiple devices. If any non-controlling device initiates a “write” request or a “read_all” request, its trust level is depleted, and it is quarantined until a complete security clearance. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Status, Device Priority, Device Trust Level, Device Zone }
- Access_Type: { read_all / write_all }
- Environment: { Date, Time }

Sample Policy 8: *Broadcast messages cannot be sent across the network by any device.*

Here the subject IoT device tries to transmit a write_all control message to all the devices connected across the IoT network. To safeguard our IoT network from flooding attacks, broadcasting messages across the entire network is strictly prohibited. This behaviour is considered malicious and such a device is quarantined instantly. The attribute fields critical for the mentioned access policy are

- Subject: { Device Identifier, Device Priority, Device Trust Level }
- Access_Type: { write_all }
- Environment: { Date, Time }

Table 2 summarises the aforementioned sample policies. Numerous other rules may be developed in the same way to handle various events depending on the requirements of the system.

Table 2. Description of all sample ABAC policies.

Policy	Description
Policy 1	A new IoT device cannot request communications with more than a certain number of devices in a specific acceptance time.
Policy 2	A new IoT can only communicate with devices in its zone until it reaches a specific age.
Policy 3	Any IoT device can only communicate with another IoT device if it matches the priority combined with the trust level required to access that device.
Policy 4	Only monitoring-type devices can send control data to any other device.
Policy 5	An IoT device can receive control data from only one monitoring device at a certain instance of time.
Policy 6	A monitoring device can send control data to multiple IoT devices at a certain instance of time if they all belong to the same zone.
Policy 7	Only a controlling/monitoring device can initiate connections to all devices in a zone simultaneously.
Policy 8	Broadcast messages cannot be sent across the network by any device.

6. Attribute Management Framework

The attribute management framework is a fundamental component of our system, which extracts and stores all the required attributes of every IoT device connected to the network by working persistently with the policy information point (PIP). The attribute management framework consists of several modules responsible for the compilation and preservation of the attributes of the overall system. In this section, we have discussed the modules that will be useful in extracting the attributes from the respective entities.

6.1. Device Attribute Management

ZAIB's entire ABAC mechanism works on device attributes, hence obtaining and maintaining a proper and updated storage of these attributes is one of the most important aspects of the ZAIB framework. ZAIB requires every IoT device to get registered as soon as it joins the network. To extract device attributes, a device fingerprinting mechanism will vigorously fingerprint different devices and record them. These attributes will be associated with the device wallet ID assigned to each device and hence will be stored in PIP's device database maintained on the chain for active devices and stored on the IPFS for all non-active devices that ever joined the network. The basic fingerprinting techniques defined in [47,48] represent different ways of identifying different device-related attributes, for example in [49] the authors suggest that a TCP port scan reveals enough information to help classify an IoT device. None of the mentioned approaches can individually satisfy our needs but a combination of a few approaches depending on the numerous network-level header fields, payload classification, and other cyber-physical features of devices can help to identify a device successfully.

6.2. User Attribute Management

Our smart citizens and network administrators will be the ones controlling our cyber city's one or more IoT devices. The trust engine determines the extent of access assigned to each user based on the device attributes, user attributes, and additional behaviour attributes such as the device and user's access histories and trust levels. We will be counting on the fingerprinting mechanism to distinguish between data and control packets.

6.3. Network Traffic Attribute Management

Each network device will request the PEP to access any other device on the network, which, when processed according to the policies, will result in being granted or denied. Hence, each transaction will be recorded in the public ledger. The network traffic attributes comprise the packet header fields and the traffic flow statistics. For extracting the packet header fields, we will use the packet capture module and extract the necessary fields. Within the same module, we will incorporate scripts that will appropriately record the necessary flow metrics and record them in a flow monitor module.

7. ZAIB Workflow and Scenario

To explain the working of ZAIB, a complete system workflow is presented as a common use case example of access request of an IoT device, such as a weight sensor, to another IoT device, such as an industrial conveyor belt motor

7.1. ZAIB Workflow

Every new user or device needs to be registered on the network to gain a blockchain wallet with public and private key pairs. This makes the entire communication anonymous and hence completely secured. Since all the communications are encrypted, this further improves the security. The entire workflow of the system is defined in the steps below:

1. After registration, a new device becomes a part of the IoT network and it can request to access any device on the network.
2. Once the request is made, it is received by the PEP from where it is forwarded to the PDPO.
3. The PDPO collects the attributes and trust levels from the PIP and requests the PIP to check if any policy regarding the access of the object by the subject exists.
4. If the policy exists, the PDP SC is triggered that implements the policy and accepts or rejects the request.
5. If such a policy is not found, a request for policy generation is sent to the PEO.
6. After receiving the request, the PEO triggers PE SC that generates the policy based on the role of the subject, its trust level, the type and category of the device, along with the trust level, type, and category required to access the object.
7. Once the policy is generated, the PEP enforces it.
8. If access is allowed, PEP generates an encrypted channel to facilitate secure communication between subject and object. If it is denied, the PEP informs the subject about the request rejection.
9. Every transaction is recorded in the PIP as it is used for determining device trust level and identifying behaviour anomalies.
10. The request and the decision taken on that request are both stored in the distributed ledger as transactions, creating an immutable history of all device activities on the IoT network. Any alteration in PIP can easily be detected by matching its records with the ledger transactions.
11. The TE SC is triggered every time a transaction is accepted or denied and it updates the device trust level based on this new transaction and the device's previous behaviour.

An overall workflow of the system is shown in Figure 10.

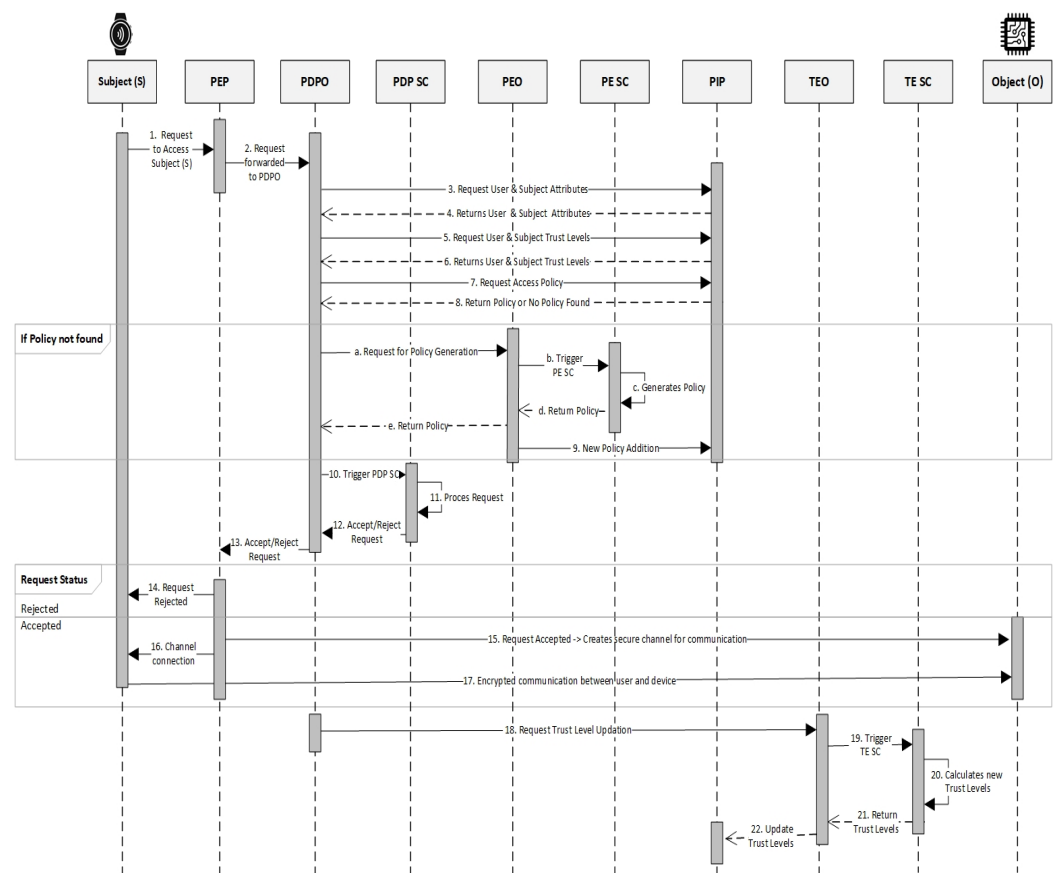


Figure 10. Overall workflow of ZAIB.

7.2. Working Scenario

As a “proof of concept” ZAIB, the proposed framework is a generic access control framework that can be applied to several IoT application systems, including those for smart homes, transportation, smart industries, and health. The following scenario is taken into consideration as a typical use case to exemplify the user experience and practicality of the provided framework. In this scenario, the object is a smart security camera set outside the main door while the subject requesting a picture and approval is a door lock, which will unlock automatically whenever the house owners arrive at the door. The smart security camera can be built using a Raspberry Pi 2 board with its dedicated camera while the lock will also consist of a Raspberry Pi 2 that will unlock whenever the facial image received from the camera matches one of the owner’s images recorded in the SD card. Both the Raspberry Pi will be connected to WiFi to provide remote access. The door lock might take several different actions depending on the privileges as stated by the policies, whereas the camera serves as a resource whose access requests need to be managed. In the current scenario, the camera will take a snapshot and save it on the Raspberry Pi SD card. As per rules set by the access control policies, the lock will request to access the screenshot of the person at the door to compare it to its presorted data and give the requester remote access using a request transaction through our PEP. Hence, the access request will consist of the following four major components:

- * Subject: the smart door lock;
- * Object: the smart security camera at the door;
- * Access_Type: request to take an image and read it;
- * Environment: current date, current time.

An initial implementation of the proposed protocol is demonstrated in Figure 11.

After registration, both devices become a part of the IoT network. The subject (*Lock*) now requests to access the snapshot captured by the object (*Camera*). The request is

received by the PEP of the smart home zone from where it is forwarded to the PDPO. The PDPO collects the attributes and trust levels from the PIP and requests the PIP to check if any policy regarding the access of the object (*Camera*) by the subject (*Lock*) exists. As the policy exists, the PDP SC has triggered and implements the policy and accepts this request. If the policy does not exist, a policy generation request is sent to the PEO. The PEP enforces the policy, but generates an encrypted channel to facilitate secure communication between subject and object. The PIP records this transaction as it will be used later on for determining device trust level and identifying behaviour anomalies. The request and the decision taken on that request are both stored in the distributed ledger as transactions, creating an immutable history of all device activities on the IoT network.

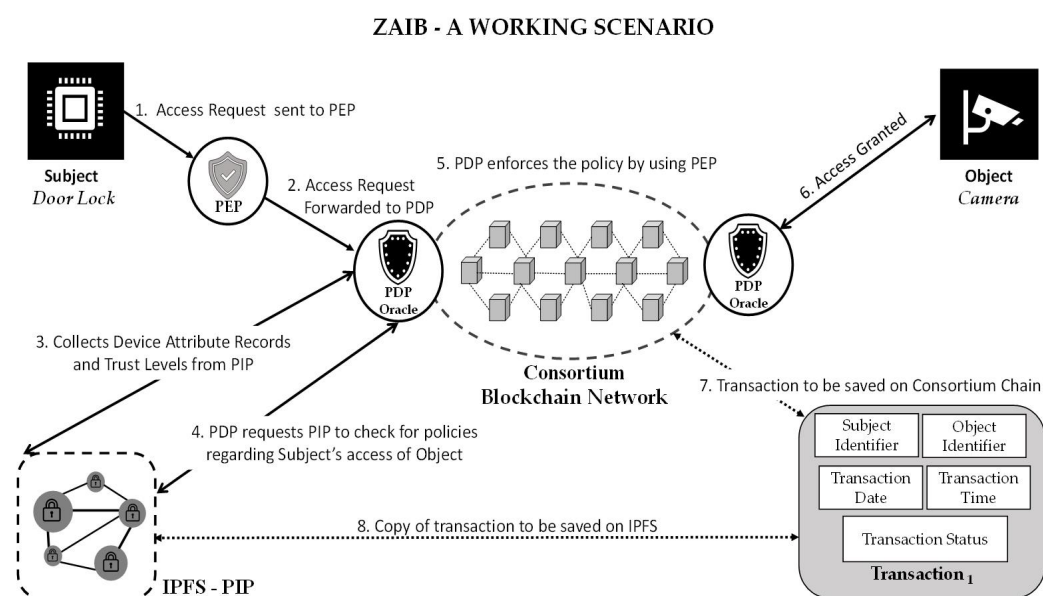


Figure 11. A working scenario of ZAIB implementation.

8. Evaluation

The distributed and vastly scattered nature of IoT devices, actuators, and sensors work flawlessly devoid of any human interventions. The standard centralized access control mechanisms cannot secure extremely distributed massive IOT infrastructures, such as smart cities where administration, traffic, business, hospitals, and citizens, i.e., all stakeholders, need everything to be connected to the Internet [50]. Millions of IoT devices and sensors will come together to form smart networks for smart transmission grids, smart security, smart healthcare, smart roads, and smart cars. A vulnerability in security will not only end in financial losses but also risks the lives of thousands of citizens [19]. The proposed framework provides fail-safe security as it not only ensures the authentication and authorization of users and devices but also maintains data privacy and confidentiality. This section discusses how these security requirements have been met by ZAIB. The analysis of all proposed sample ABAC policies is shown in Table 3.

8.1. Device Authentication

ZAIB ensures device and user authentication by using blockchain wallets. Public key cryptography certifies the device credentials, whereas the device security check is conducted periodically to have the latest update on the device status. After making sure that only certified devices join the network, device behaviour history is also secured which helps identify any anomalies in device behaviour and immediately decreases its trust level while initiating a security check to verify if the device was tampered with in any way. To prevent a malicious device from misbehaving and infecting other devices, it is quarantined immediately and is only allowed to rejoin the network and communicate with other devices

when they have been restarted and have obtained security clearance. Therefore, ZAIB offers protection against attacks such as impersonation or masquerade attacks.

Table 3. Analysis of all proposed sample ABAC policies.

Policy	Authentication	Authorization	Confidentiality	Privacy
Policy 1	✓	✓	✓	✓
Policy 2	✓	✓	✓	✓
Policy 3	✓	✓	✓	✓
Policy 4	✓	✓	X	X
Policy 5	X	✓	✓	X
Policy 6	✓	✓	X	X
Policy 7	✓	✓	✓	✓
Policy 8	X	X	✓	✓

8.2. Authorization

ZAIB provides a dynamic access authorization by implementing attribute-based access control in a massive IoT infrastructure. Monitoring dynamic parameters like the latest trust and risk levels of IoT devices along with static device attributes such as the priority, category, and device network for every respective request ensures that access will not be granted to any device with behaviour anomalies. Hence, the proposed scheme provides comprehensive dynamic access control compared to DAC, MAC, RBAC, or even static ABAC.

8.3. Confidentiality

ZAIB provides the protection of system resources against unauthorized access. The degree of authorization required to access devices and data in ZAIB is set intelligently by using public-key authorization with smart wallets and continuous behaviour analysis of all IoT devices to ensure security against identity theft or masquerade attacks to maintain the confidentiality of classified information.

8.4. Privacy

Privacy means having full control or decision-making authority on how the user's data can be collected and used. ZAIB ensures that no unauthorized person or device can gain access to devices owned by an individual or data generated by them, hence ensuring the basic right to privacy for every individual.

To evaluate the effectiveness of the suggested policies, let us assume a scenario like a smart home. With all the home appliances and security equipment attached to the Internet and being monitored by central controlling devices, the proposed policies will make sure that no unauthorized perpetrator gets access to your devices. In case an attacker gets access to your home network and tries to plant a new IoT device of their own, policies 1 and 2 limit their communication requests and also limit their access to only a certain zone, hence reducing the attacker's access radius to a minimum area. Policies 3 and 4 stop the subject from communicating with any object device on the home network by applying the priority, age, trust level, and device type filters, allowing communication if the above-mentioned attributes are matched. Here, priority, age, and trust level are all dynamic attributes that change with time and hence provide improved security as devices once deemed eligible for communication will be denied the same privilege if any behaviour anomalies are detected and their trust levels are reduced. This dynamic behaviour analysis ensures security even if any device on the home network is compromised by an intruder. Policies 5 and 6 dictate the behaviour of monitoring devices and keep check on their commanding areas by limiting

them to particular zones. Hence, if a malicious device is registered in the kitchen, it can not access security devices or devices in other zones such as the bedrooms or lounges. Policies 7 and 8 ensure that only a monitoring device can connect to multiple devices simultaneously and broadcasting any message on the network is prohibited, hence the chances of an intruder infecting all devices on networks are very slim and almost down to zero. Therefore, the discussion suggests that the proposed system provides complete security, authentication, data privacy, and confidentiality to all users utilizing IoT devices registered on the network.

8.5. Computational and Storage Tradeoff

We have identified a trade-off between computational and spatial complexity for the realistic implementation of this system. Keeping all device attribute data on-chain will result in processing all access requests more quickly, but the larger ledger size will raise computation costs. However, if an off-chain data repository is maintained, the smaller ledger size will result in greater costs and delays while reducing computing complexity.

9. Conclusions

This paper addresses the security challenges for large IoT-based infrastructures such as smart cities and provides a zero-trust and ABAC-based dynamic solution for confronting these challenges. Issues such as user privacy, device authentication, and authorization have been resolved by modelling a framework that provides a completely secure device-to-device communication mechanism by not only considering the existing security levels of the network but also considering the behaviour anomalies of the devices to instantaneously detect any kind of intrusion or device misconduct by using device trust levels. The framework is modelled by dividing IoT networks into various zero-trust zones and an ABAC framework that specifies subject, object, and network attributes. Several policies were defined based on these attributes to enforce reliable and secure machine-to-machine communication which is being recorded by the immutable distributed ledger for advanced accountability. We have also discussed an attribute management framework that captures and calculates the important device attributes required for implementing the ABAC policies. In future, some still-relevant major problems in this field, such as computation and space overheads, need to be addressed to find the optimal equilibrium for the best system performance.

Author Contributions: All authors contributed equally to this article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors do not have financial or professional conflicting interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC	Attribute-Based Access Control
D2D	Device-to-Device
DAC	Discretionary Access Control
IoT	Internet of Things
IPFS	Interplanetary File System
MAC	Mandatory Access Control
MCAP	Microcore And Perimeter
PA	Policy Administrator
PAP	Policy Administration Point
PDP	Policy Decision Point
PDPO	Policy Decision Point Oracle

PE	Policy Engine
PEO	Policy Engine Oracle
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role-Based Access Control
SC	Smart Contract
TE	Trust Engine
TEO	Trust Engine Oracle
ZAIB	The name of the proposed architecture (ZTA and ABAC for IoT using Blockchain)
ZT	Zero Trust
ZTA	Zero-Trust Architecture

References

- Chen, B.; Qiao, S.; Zhao, J.; Liu, D.; Shi, X.; Lyu, M.; Chen, H.; Lu, H.; Zhai, Y. A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture. *IEEE Internet Things J.* **2021**, *8*, 10248–10263. [CrossRef] [PubMed]
- Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghaby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 24. [CrossRef]
- What Is Stuxnet? 1999. Available online: <https://www.trellix.com/en-us/security-awareness/ransomware/what-is-stuxnet.html> (accessed on 30 December 2022).
- U.S. Institute of Peace. Israeli Sabotage of Iran's Nuclear Program. 2021. Available online: <https://iranprimer.usip.org/blog/2021/apr/12/israeli-sabotage-iran%E2%80%99s-nuclear-program> (accessed on 12 April 2021).
- Zetter, K. Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid, Published in Wired. 2010. Available online: <https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/> (accessed on 30 October 2010).
- Razzaq, M.A.; Gill, S.H.; Qureshi, M.A.; Ullah, S. Security issues in the Internet of Things (IoT): A comprehensive study. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 383. [CrossRef]
- Arshad, J.; Azad, M.A.; Abdeltaif, M.M.; Salah, K. An intrusion detection framework for energy constrained IoT devices. *Mech. Syst. Signal Process.* **2020**, *136*, 106436. [CrossRef]
- Arshad, J.; Azad, M.A.; Mahmoud Abdellatif, M.; Ur Rehman, M.H.; Salah, K. COLIDE: A collaborative intrusion detection framework for Internet of Things. *IET Netw.* **2019**, *8*, 3–14.
- Trilles, S.; Calia, A.; Belmonte, Ó.; Torres-Sospedra, J.; Montoliu, R.; Huerta, J. Deployment of an open sensorized platform in a smart city context. *Future Gener. Comput. Syst.* **2017**, *76*, 221–233. [CrossRef]
- Pacheco, J.; Hariri, S. Anomaly behavior analysis for IoT sensors. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3188. [CrossRef]
- Samaniego, M.; Deters, R. Zero-trust hierarchical management in IoT. In Proceedings of the 2018 IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, USA, 2–7 July 2018; pp. 88–95. [CrossRef]
- Bruno, E.; Gallier, R.; Gabillon, A. Enforcing access controls in IoT networks. In *Proceedings of the International Conference on Future Data and Security Engineering*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 429–445. [CrossRef]
- Zimmer, B. LISA: A Practical Zero Trust Architecture. In *Proceedings of the Enigma 2018 (Enigma 2018)*; USENIX Association: Santa Clara, CA, USA, 2018.
- Alramadhan, M.; Sha, K. An overview of access control mechanisms for internet of things. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–6. [CrossRef]
- Kindervag, J.; *Build Security into Your Network's DNA: The Zero Trust Network Architecture*; Forrester Research Inc.: Cambridge, MA, USA, 2010; pp. 1–26.
- Muralidharan, S.; Ko, H. An InterPlanetary file system (IPFS) based IoT framework. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–2.
- Rose, S.W.; Borchert, O.; Mitchell, S.; Connelly, S. Zero Trust Architecture. 2020. Available online: <https://www.nist.gov/publications/zero-trust-architecture> (accessed on 1 February 2023). [CrossRef]
- Babiker Mohamed, M.; Matthew Alofe, O.; Ajmal Azad, M.; Singh Lallie, H.; Fatema, K.; Sharif, T. A comprehensive survey on secure software-defined network for the Internet of Things. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4391. [CrossRef]
- Dhar, S.; Bose, I. Securing IoT Devices Using Zero Trust and Blockchain. *J. Organ. Comput. Electron. Commer.* **2020**, 1–17. [CrossRef]
- Zhang, Y.; Li, B.; Liu, B.; Wu, J.; Wang, Y.; Yang, X. An attribute-based collaborative access control scheme using blockchain for IoT devices. *Electronics* **2020**, *9*, 285. [CrossRef]
- Liu, H.; Han, D.; Li, D. Fabric-IoT: A blockchain-based access control system in IoT. *IEEE Access* **2020**, *8*, 18207–18218. [CrossRef]
- Benet, J. Ipfs-content addressed, versioned, p2p file system. *arXiv* **2014**, arXiv:1407.3561.
- Naz, M.; Al-zahrani, F.A.; Khalid, R.; Javaid, N.; Qamar, A.M.; Afzal, M.K.; Shafiq, M. A secure data sharing platform using blockchain and interplanetary file system. *Sustainability* **2019**, *11*, 7054. [CrossRef]
- Assunção, P. A Zero Trust Approach to Network Security. In Proceedings of the Digital Privacy and Security Conference 2019, Miami, FL, USA, 15–17 May 2019.

25. Lukaseder, T.; Halter, M.; Kargl, F. Context-based Access Control and Trust Scores in Zero Trust Campus Networks. In *Sicherheit 2020*; Gesellschaft für Informatik e.V.: Bonn, Germany, 2020. [CrossRef]
26. Picard, N.; Colin, J.N.; Zampunieris, D. Context-aware and attribute-based access control applying proactive computing to IoT system. In Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDs 2018). SCITEPRESS, Madeira, Portugal, 19–21 March 2018; pp. 333–339. [CrossRef]
27. Zhang, X.; Jiang, X. IoT architecture based on ABAC smart contract. In Proceedings of the 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China, 24–26 April 2020; pp. 122–128. [CrossRef]
28. Tomaz, A.E.B.; Do Nascimento, J.C.; Hafid, A.S.; De Souza, J.N. Preserving Privacy in Mobile Health Systems Using Non-Interactive Zero-Knowledge Proof and Blockchain. *IEEE Access* **2020**, *8*, 204441–204458. [CrossRef]
29. Ruan, P.; Anh Dinh, T.T.; Lin, Q.; Zhang, M.; Chen, G.; Chin Ooi, B. Revealing Every Story of Data in Blockchain Systems. *SIGMOD Rec.* **2020**, *49*, 70–77. [CrossRef]
30. Ruan, P.; Chen, G.; Dinh, T.T.A.; Lin, Q.; Ooi, B.C.; Zhang, M. Fine-Grained, Secure and Efficient Data Provenance on Blockchain Systems. *Proc. VLDB Endow.* **2019**, *12*, 975–988. [CrossRef]
31. Ferraiolo, D.; Chandramouli, R.; Hu, V.; Kuhn, R. A comparison of attribute based access control (ABAC) standards for data service applications. *NIST Spec. Publ.* **2016**, *800*, 178.
32. Arnold, R.; Longley, D. Zero-knowledge proofs do not solve the privacy-trust problem of attribute-based credentials: What if alice is evil? *IEEE Commun. Stand. Mag.* **2019**, *3*, 26–31. [CrossRef]
33. Arasteh, H.; Hosseinneshad, V.; Loia, V.; Tommasetti, A.; Troisi, O.; Shafie-khah, M.; Siano, P. Iot-based smart cities: A survey. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–6. [CrossRef]
34. Waheed, U.; Khan, M.S.A.; Awan, S.M.; Khan, M.A.; Mansoor, Y. Decentralized Approach to Secure IoT based Networks using Blockchain Technology. 3C Tecnología_Glosas de innovación aplicadas a la pyme (2019). Available online: <https://dialnet.unirioja.es/servlet/articulo?codigo=6933920> (accessed on 13 January 2023).
35. Ouaddah, A.; Abou Elkalam, A.; Ait Ouahman, A. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2016**, *9*, 5943–5964. [CrossRef]
36. Durga, R.; Poovammal, E.; Ramana, K.; Jhaveri, R.H.; Singh, S.; Yoon, B. CES Blocks—A Novel Chaotic Encryption Schemes-Based Blockchain System for an IoT Environment. *IEEE Access* **2022**, *10*, 11354–11371. [CrossRef]
37. Bezawada, B.; Haefner, K.; Ray, I. Securing home IoT environments with attribute-based access control. In Proceedings of the Third ACM Workshop on Attribute-Based Access Control, Tempe, AZ, USA, 21 March 2018; pp. 43–53.
38. Peng, Z.; Xu, J.; Hu, H.; Chen, L.; Kong, H. BlockShare: A Blockchain empowered system for privacy-preserving verifiable data sharing. *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.* **2022**, *1*, 14–24.
39. Alevizos, L.; Ta, V.T.; Eiza, M.H. Augmenting Zero Trust Architecture to Endpoints Using Blockchain: A Systematic Review. *arXiv* **2021**, arXiv:2104.00460.
40. Ding, S.; Cao, J.; Li, C.; Fan, K.; Li, H. A novel attribute-based access control scheme using blockchain for IoT. *IEEE Access* **2019**, *7*, 38431–38441. [CrossRef]
41. Yan, X.; Wang, H. Survey on Zero-Trust Network Security. In *Proceedings of the International Conference on Artificial Intelligence and Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 50–60. [CrossRef]
42. Weerapanpisit, P.; Trilles, S.; Huerta, J.; Painho, M. A Decentralized Location-Based Reputation Management System in the IoT Using Blockchain. *IEEE Internet Things J.* **2022**, *9*, 15100–15115. [CrossRef]
43. Bernabe, J.B.; Ramos, J.L.H.; Gomez, A.F.S. TACIoT: Multidimensional trust-aware access control system for the Internet of Things. *Soft Comput.* **2016**, *20*, 1763–1779. [CrossRef]
44. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot. *Computers* **2018**, *7*, 39. [CrossRef]
45. Cruz-Piris, L.; Rivera, D.; Marsa-Maestre, I.; De La Hoz, E.; Velasco, J.R. Access control mechanism for IoT environments based on modelling communication procedures as resources. *Sensors* **2018**, *18*, 917. [CrossRef]
46. Eidle, D.; Ni, S.Y.; DeCusatis, C.; Sager, A. Autonomic security for zero trust networks. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017; pp. 288–293. [CrossRef]
47. François, J.; Abdelnur, H.; Festor, O. Automated behavioral fingerprinting. In *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 182–201. [CrossRef]
48. Radhakrishnan, S.V.; Uluagac, A.S.; Beyah, R. GTID: A technique for physical device and device type fingerprinting. *IEEE Trans. Dependable Secur. Comput.* **2014**, *12*, 519–532. [CrossRef]

49. Sivanathan, A.; Gharakheili, H.H.; Sivaraman, V. Can we classify an iot device using tcp port scan? In Proceedings of the 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 21–22 December 2018; pp. 1–4. [[CrossRef](#)]
50. Gabillon, A.; Gallier, R.; Bruno, E. Access controls for IoT networks. *SN Comput. Sci.* **2020**, *1*, 1–13. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.