NIST IR 8403

Blockchain for Access Control Systems

Vincent C. Hu

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8403

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST IR 8403

Blockchain for Access Control Systems

Vincent C. Hu
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8403

May 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report 8403
28 pages (May 2022)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8403

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Submit comments on this publication to: ir8403-comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

The rapid development and wide application of distributed network systems have made network security – especially access control and data privacy – ever more important. Blockchain technology offers features such as decentralization, high confidence, and tamper-resistance, which are advantages to solving auditability, resource consumption, scalability, central authority, and trust issues – all of which are challenges for network access control by traditional mechanisms. This document presents general information for blockchain access control systems from the views of blockchain system properties, components, functions, and supports for access control policy models. Considerations for implementing blockchain access control systems are also included.

Keywords

access control; blockchain; authorization; ABAC; policy.

Acknowledgments

The author, Vincent C. Hu of the National Institute of Standards and Technology (NIST), wishes to thank Dylan Yaga, David Ferraiolo, Isabel Van Wyk, Jim Foti (NIST), and Antonios Gouglidis (Lancaster University, UK) who reviewed drafts of this document. The author also gratefully acknowledges and appreciates the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

Audience

This document assumes that readers are access control system experts who also have basic network and blockchain expertise. Because of the constantly changing nature of the information technology (IT) industry, readers are strongly encouraged to take advantage of other resources (including those referred in this document) for more current and detailed information.

Patent Disclosure Notice

NOTICE: The Information Technology Laboratory (ITL) has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

Following the ITL call for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, notice of one or more such claims has been received.

By publication, no position is taken by ITL with respect to the validity or scope of any patent claim or of any rights in connection therewith. The known patent holder(s) has (have), however, provided to NIST a letter of assurance stating either (1) a general disclaimer to the effect that it does (they do) not hold and does (do) not currently intend holding any essential patent claim(s), or (2) that it (they) will negotiate royalty-free or royalty-bearing licenses with other parties on a demonstrably nondiscriminatory basis with reasonable terms and conditions.

Details may be obtained from ir8403-comments@nist.gov

No representation is made or implied that this is the only license that may be required to avoid patent infringement in the use of this publication.

Executive Summary

Access control is concerned with determining the allowed activities of legitimate users and mediating every attempt by a user to access a resource in the system. The objectives of an access control system are often described in terms of protecting system resources against inappropriate or undesired user access. From a business perspective, this objective could just as well be described in terms of the optimal sharing of information. As current information systems and network architectures evolve to be more lightweight, pervasive, and interactive – such as the cloud and Internet of Things (IoT) – there is need for an access control mechanism to support the requirements of decentralization, scalability, and trust for accessing objects, all of which are challenging for traditional mechanisms.

Blockchains are tamper-evident and tamper-resistant cryptographically linked blocks of data (which create digital ledgers) implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company, or government). It uses replicated, shared, and synchronized digital blocks between the users of a private or public distributed computer network located in different sites or organizations. Blockchain can be utilized for access control systems as a trustable alternative for a single entity/organization or a member of a large-scale system to enforce access control policies. The robust, distributed nature of blockchain technology can overcome the limitations of traditional access control systems in a decentralized and efficient way. It is supported by the following infrastructural properties that are not included in traditional access control mechanisms unless specifically implemented:

- Tamper-evident and tamper-resistant design prevents the alteration of **access control data** (i.e., attributes, policy rules, environment conditions, and access requests) and **access control logs** (i.e., request permissions and previous access control data) and reduces the probability of frauds.
- The control of authorization processing is decentralized, and the storage of access control data/logs has no single point of failure, thus providing more system tolerance and availability.
- The traceability of blocks allows access control data/logs and system states to be seen and tracked.
- The execution of arbitrary programs in smart contracts allows for controls on distributed access control data and authorization processes.
- Consensus mechanisms and protocols jointly regulate the participating access control entities/organizations in determining policy rules through blocks or smart contracts.

Table of Contents

List of Figures

List of Tables

1 Introduction

Access control (AC) is concerned with determining the allowed activities of legitimate users and mediating every attempt by a user to access a resource in the system. The objectives of an AC system are often described in terms of protecting system resources against inappropriate or undesired user access. From a business perspective, this objective could just as well be described in terms of the optimal sharing of information [IR7316]. As current information systems and network architectures evolve to be more lightweight, pervasive, and interactive – such as the cloud and Internet of Things (IoT) – there is need for an AC mechanism to support the requirements of decentralization, scalability, and trust for accessing objects, all of which are challenging for traditional mechanisms.

Blockchains are tamper-evident and tamper-resistant cryptographically linked blocks of data (which create digital ledgers) implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company, or government). It uses replicated, shared, and synchronized digital blocks between the users of a private or public distributed computer network located in different sites or organizations. A block links to previous blocks by containing a cryptographic hash summary of the previous block's contents, thus making the blockchain tamper-resistant and tamper-evident (because to change a block, one must then change all subsequent blocks that follow it). A linked list of blocks (i.e., a blockchain) typically has no central control authority and utilizes a decentralized consensus mechanism for reliable data transactions. A smart contract is a transaction protocol that executes the terms of a contract (such as payment term, lien, confidentiality, and even enforcement) on a blockchain via code that is deployed to and executed by blockchain nodes. The main purpose of smart contracts is to satisfy common contractual conditions, as well as minimize exceptions (both malicious and accidental) and the need for trusted intermediaries [IR8202]. Every blockchain node that executes the smart contract should arrive at the same result given the same input.

Blockchain can be utilized for AC systems as a trustable alternative for a single entity/organization or a member of a large-scale system to enforce AC policies. The robust, distributed nature of blockchain technology can overcome the limitations of traditional AC systems in a decentralized and efficient way. It is supported by the following infrastructural properties that are not included in traditional AC mechanisms unless specifically implemented:

- Tamper-evident and tamper-resistant design prevents the alteration **AC data** (i.e., attributes, policy rules, environment conditions, and access requests) and **AC logs** (i.e., request permissions and previous AC data) and reduces the probability of frauds.
- The control of authorization processing is decentralized, and the storage of AC data/logs has no single point of failure, thus providing more system tolerance and availability.
- The traceability of blocks allows AC data/logs and system states to be seen and tracked.
- The execution of arbitrary programs in smart contracts allows for controls on distributed AC data and authorization processes.
- Consensus mechanisms and protocols jointly regulate the participating AC entities/organizations in determining policy rules through blocks or smart contracts.

Blockchain properties improve the security, flexibility, scalability, integrity, and confidentiality of AC data/logs and processes (compared to traditional AC systems) by allowing organizations to verify and audit AC data transactions and processes to track the states of their AC systems hosted on distributed sites [SP162].

This document presents analyses of blockchain AC systems from the perspectives of properties, components, architectures, and model supports, as well as discussions on considerations for implementation. It should not be deemed comprehensive due to the diverse applications of business and mission requirements. Before selecting and deploying a blockchain AC product or technology, the host organization should augment considerations in this document with testing and independent product reviews.

This document is organized as follows:

- Section 1 is the Introduction.
- Section 2 describes blockchain system components and their advantages over traditional AC systems.
- Section 3 illustrates the architecture of basic AC functions for blockchain systems.
- Section 4 demonstrates blockchain AC system supports for AC policy models.
- Section 5 discusses considerations for the implementation of blockchain AC systems.
- Section 6 is the Conclusion.

2 Blockchain System Components and Advantages for Access Control Systems

Blockchain systems provide an alternative (or complimentary) system for reliability, security, accountability, and scalability for AC systems. Blockchain characteristics – such as transparency, distributed computing/storage, and a tamper-evident/tamper-resistant design – help to prevent AC data from being accessed or modified by malicious users. Access logs are also recorded in blocks that allow for the detection of malicious activities. Blockchain system components and their advantages for AC systems are:

- A **node** is an individual computer system within a blockchain network. It can act as an AC system's entity or organization and is called an **AC node** within the AC network. AC nodes including lightweight nodes (i.e., a node that does not store or maintain a copy of the blockchain), full nodes (i.e., a node that stores the entire blockchain and ensures that transactions are valid), and publishing nodes (i.e., a full node that also publishes new blocks). Lightweight nodes must pass their transactions to full nodes. Depending on the design of the AC system, AC nodes can act as host servers for AC data (e.g., subject/object attributes, environment conditions, and policy rules) or as administrators for AC policy management and enforcement.

- A **block** contains trustable and tamper-resistant AC data as well as a history of access logs without third parties or centralized management. Distributed blocks solve the single point of failure problem and provide information for distributed architectures, which often involve a much larger set of AC entities or organization. Distributed ownership of blocks is necessary because of possible trust, security, and reliability concerns that are associated with the centralized management of AC enforcement or AC data ownership [IR8202].

- **Full blockchain nodes** are not only a repository of AC data and logs of blocks but can also store objects. Even though blockchain contents are tamper-evident and tamper-resistant, [Kuhn] proposed a data structure with similar features to a blockchain – the data block matrix data structure – that allows for the deletion of arbitrary records and preserves hash-based integrity assurance that other blocks are unchanged. Such a feature may be incorporated into AC systems that require integrity and privacy protection such that organizations or users are able to delete all information related to a particular access request.

- A **consensus mechanism** ensures that only valid transactions are recorded on the blockchain. Different kinds of consensus mechanisms can be used for AC systems, including proof of work (PoW), proof of stake (PoS), and single committee-based [LQLL]. For mandatory AC (MAC) policies, the integrity and consistency of AC administrations are maintained by consensus mechanisms configured for permissioned blockchains. Consensus mechanisms configured for permissionless blockchains are crucial for discretional AC (DAC) policies due to the dynamic management requirement for scalability and decentralization of the system.

- A **smart contract** is an event-driven computer program distributed to and executed by AC nodes to facilitate and enforce **AC processes** (i.e., authorization processes and AC data transitions) between them without going through a trusted third party. A smart contract can perform calculations, store data in storage spaces, expose environment conditions to reflect the current system state via callable functions, and – if appropriate – automatically send data or function calls to other smart contracts [IR8202]. Adding a smart contract to a block means executing code and updating the **AC state** (i.e., previous access permissions, environment conditions, and system status) accordingly [DMMR]. The smart contract code is also tamper-evident and tamper-resistant. It is copied to each AC node to reduce human error and avoid disputation, thus providing a secure way to specify AC policies and transform the authorization process into a distributed execution [KLG]. Such a capability works especially well for a system that requires each distributed AC entity to perform local authorization so that the authorization chain can be verified in a decentralized manner.

Blockchain systems' decentralized storage of AC data and the delegation of authorization processes not only optimize the performance and cost of an AC system but also help avoid single points of failure and many-to-one traffic problems for highly-dynamic and scalable systems (e.g., cloud, grid, IoT). This is especially true for AC systems that enforce attribute-based policy models, such as Role-Based Access Control (RBAC) [FK] and Attribute-Based Access Control (ABAC) [SP162], where the AC data management and policy enforcement are traditionally administrated by a central server. Blockchain also allows for AC log information collection and replicates data among AC nodes in a transparent and trustworthy way with verifiable and secure records. The following blockchain capabilities are not generally supported by traditional, centrally controlled AC mechanisms:

- Removes control from a centralized system and provides flexibility in AC data management and AC processes, such as workflow control or localization control, and thus avoids possible leakages or faults of access privileges by excessive powers of centralized server [LOLL]

- Increases performance for managing a large number of subjects and objects, such as IoT AC systems, where each IoT device is an AC node of an AC entity or organization

- Allows for the enforcement of flexible, fine-grained, and responsive policy by transferring or propagating access privileges from one AC node to others through smart contract functions

- Supports communication between subjects, AC administrators, and protocols for the administration of heterogeneous AC policies and security analysis

- Avoids tampering and single points of failure (e.g., caused by network attacks like distributed denial-of-service (DDoS)) to increase integrity, availability, and traceability [GBHC] through recording, distributing, and storing AC data and log information in the blockchain. However, as all subjects can see all entries in the blockchain, privacy can be an issue for this capability.

- Dispenses heavy and complex authorization or management tasks between AC nodes to enhance performance and scalability, as well as decrease the cost and responsibility of administration traditionally assigned to central or third-party services

3 Access Control Functions of Blockchain AC Systems

This section examines an integration of access control functional components and a blockchain framework in support of ABAC. Extensible Access Control Markup Language (XACML) [XACML] and Next Generation Access Control (NGAC) [INCITS] are two ABAC standards that could serve as a basis for this discussion. See [SP162] for a detailed comparison. Both standards encompass four layers of functional decomposition: enforcement, decision, administration, and access control data. Unfortunately, XACML and NGAC achieve this decomposition by involving components with often similar names but apply different access control data types, provide different interfaces, and result in different functional outcomes. To avoid confusion, the remainder of this section applies XACML's reference architecture as an example ABAC integration use case.

The Organization for the Advancement of Structured Information Standards (OASIS) standard XACML proposes basic processing entities for AC systems. Each entity handles a different stage of processing a user's access request, as shown in Figure 1. These functional components may be physically and logically separated and distributed rather than centralized, such as several functional "points" that are the service node for the retrieval and management of the policy, along with some logical components for handling the context or workflow of AC data retrieval and assessment.

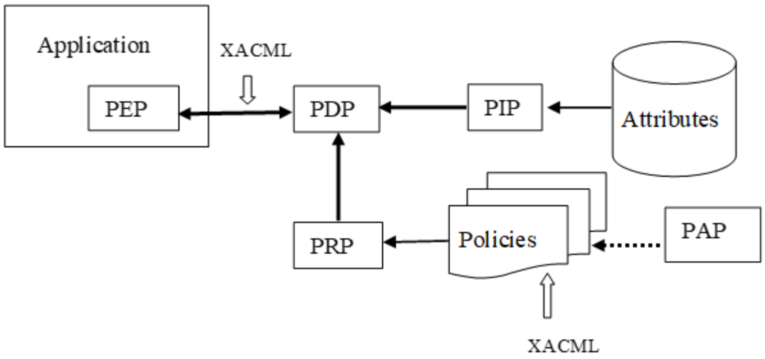

Figure 1 — XACML Architecture

In a blockchain AC system, these function points can be performed by an individual or combination of blockchain system components. The following describes the five basic XACML function points and their implementations by blockchain AC components.

1. Policy Administration Point (PAP): Provides a user interface for creating, testing, and debugging policies, as well as storing these policies in the appropriate repository. PAP can be created and maintained by AC nodes or smart contracts that are coded to access AC policies, depending on where the source of the AC data is maintained.

2. Policy Information Point (PIP): Serves as the source of subject/object attributes or environment condition data required for policy evaluation to provide the information needed by the PDP to make the authorization decisions. PIP can be performed in AC nodes,

coded in smart contracts that are coded to access AC data, or hosted in an off-chain processor, depending on where the source of the AC data is maintained.

3. Policy Retrieval Point (PRP): Where the policies are stored and fetched by the PDP. As PIP depends on where the source of AC data is maintained, the PRP can be implemented in AC nodes, coded in smart contracts, or hosted in an off-chain system, depending on where the source of the policy rules is maintained [IR7874].

4. Policy Decision Point (PDP): Computes access decisions by evaluating the applicable policies based on information provided by PIP and PRP. One of the main functions of the PDP is to mediate or deconflict policy rules. PDP can be coded in a smart contract, coded into distributed executions, or performed by AC nodes.

5. Policy Enforcement Point (PEP): Makes decision requests and enforces authorization decisions made by the PDP. PEP can be performed by AC nodes that contain objects, by smart contracts that are coded to access objects, or by an off-chain processor.

The basic AC function points can be processed through the uploading and updating of AC data to execute AC processes, smart contract functions, or even off-chain processes, depending on security/performance requirements and the resource availability of the AC system. Figures 2, 3, 4, and 5 illustrate examples of different assignments of function points in blockchain AC systems. Each function point in a picture is labeled (in blue) alongside the performing blockchain component.

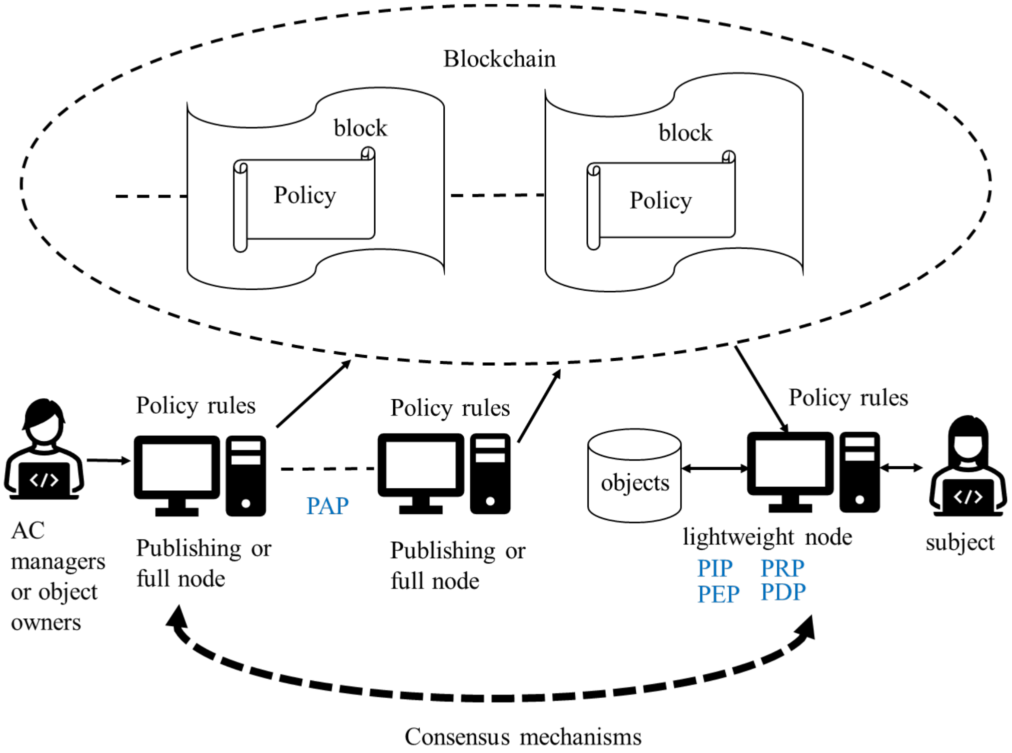

Figure 2 – Example 1 of access control function points implemented in a blockchain system

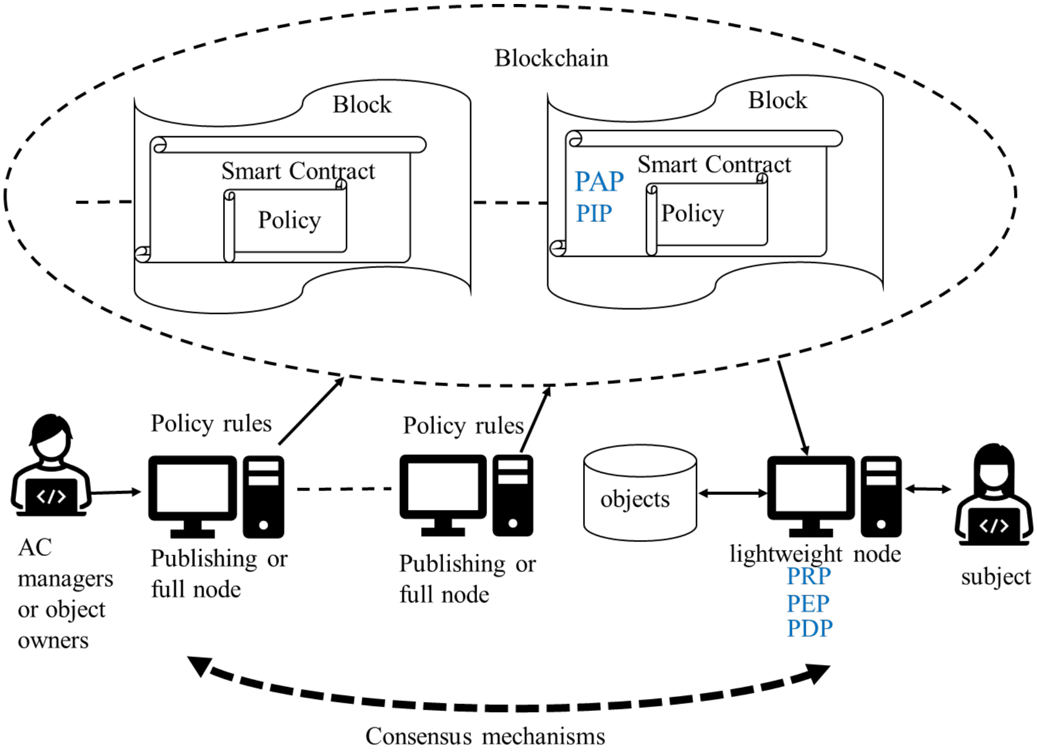

Figure 3 – Example 2 of access control function points implemented in a blockchain system

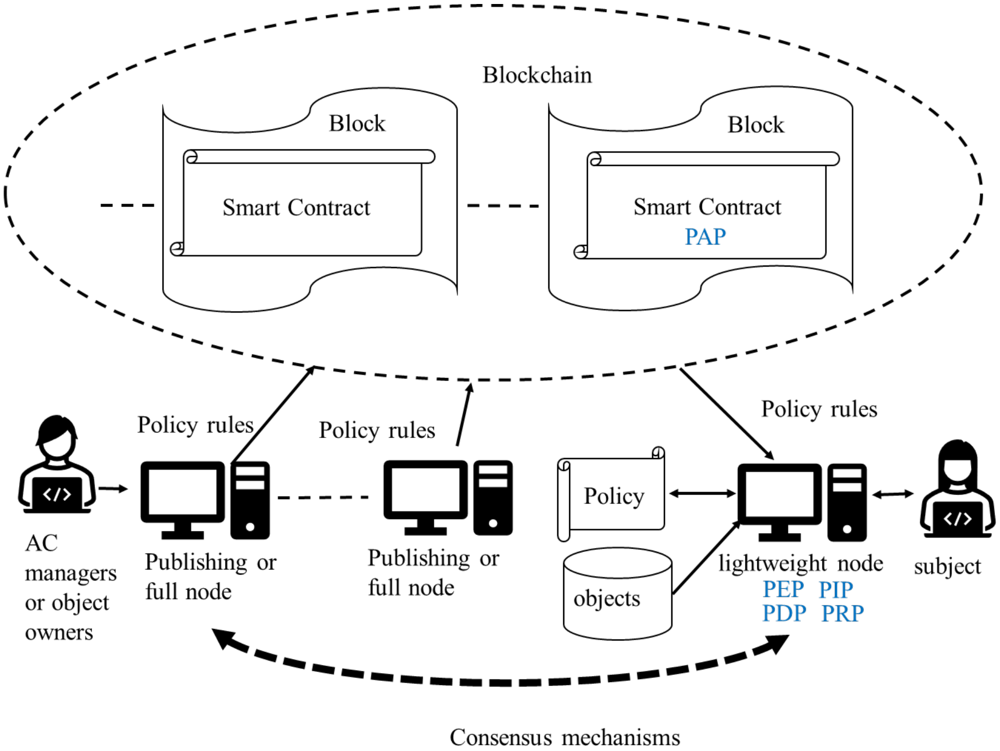

Figure 4 – Example 3 of access control function points implemented in a blockchain system

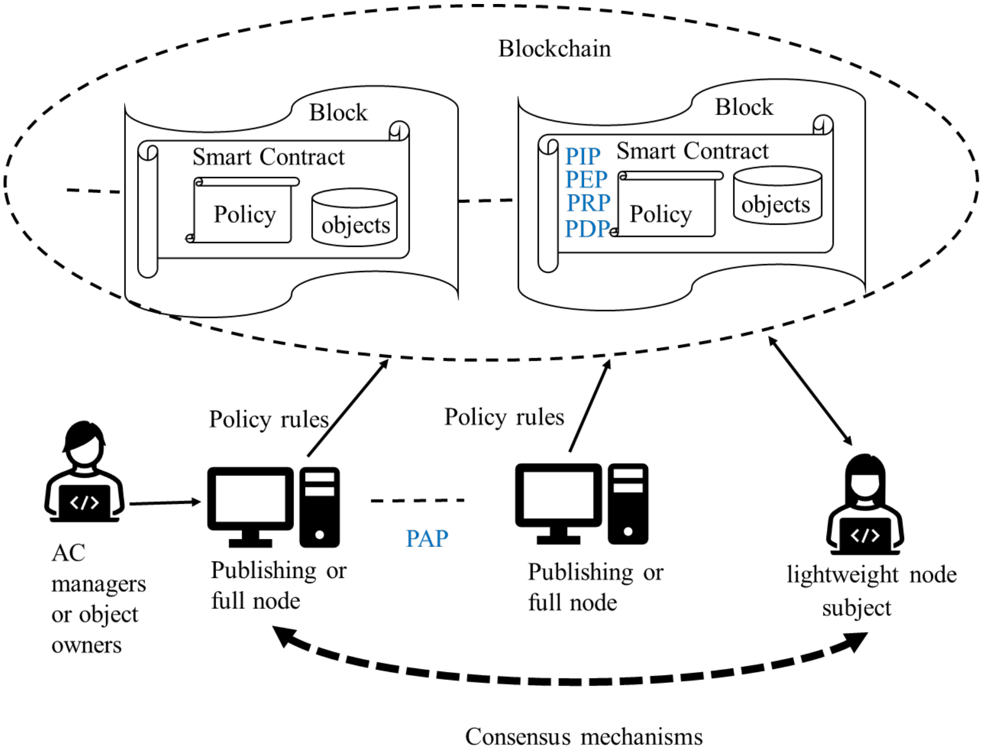

Figure 5 ─ **Example 4 of access control function points implemented in blockchain systems**

A centralized management AC system, as shown in Figure 2, requires the blockchain to play the role of a trusted storage of AC data such that most function points are hosted in a lightweight node connected to the blockchain to obtain the AC data and current system states. However, it still inherits the shortcomings of centralization, such as the problem of a single point of failure. In contrast, Figure 5 shows how the AC system requires decentralized management and adopts blockchain as a trusted platform to maximize system availability and minimize the possibility of AC data forgery and tampering. All function points, except for PAP, are implemented in the blockchain (with smart contracts) that also stores AC data. For this implementation, AC policy administrators use a publishing node for policy management, and subjects use lightweight nodes for access requests that will be processed by smart contracts, thereby ensuring that it can be processed promptly. For conciseness, the examples in Figures 2 through 5 address both the subject and object attributes associated with the policies (i.e., stored and managed by the same authority). Otherwise, they can be separately administrated by PIP and PRP either in or out of the chain hosts, as shown in Figure 6 – an example of options for a federated AC system.

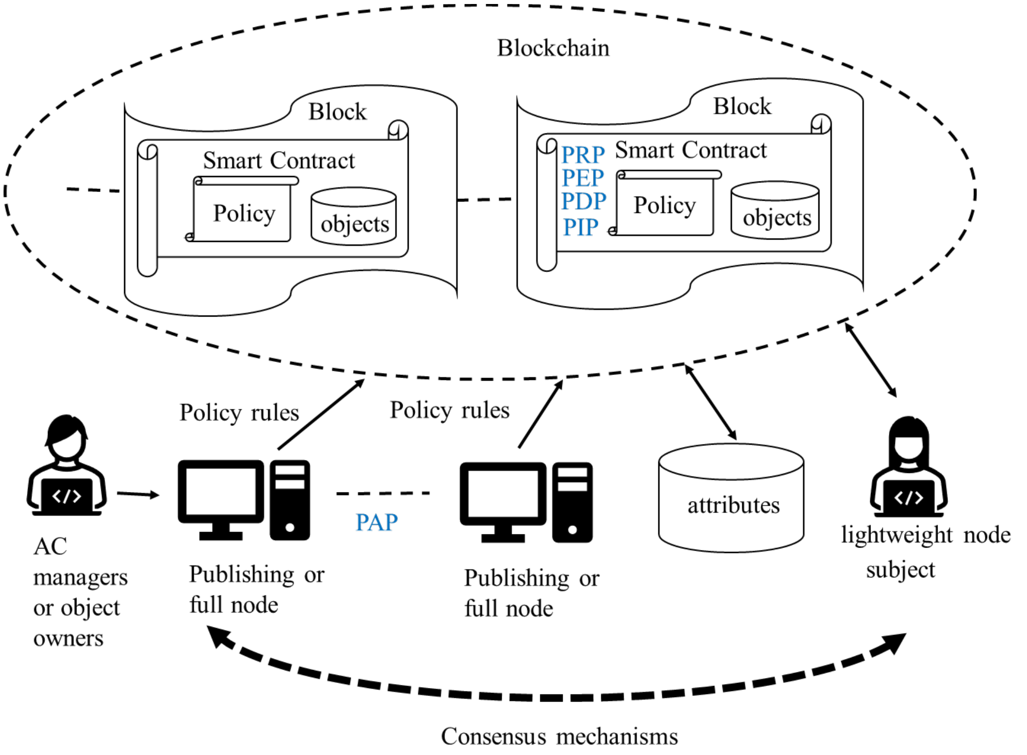

Figure 6 – Attribute source is out of the chain

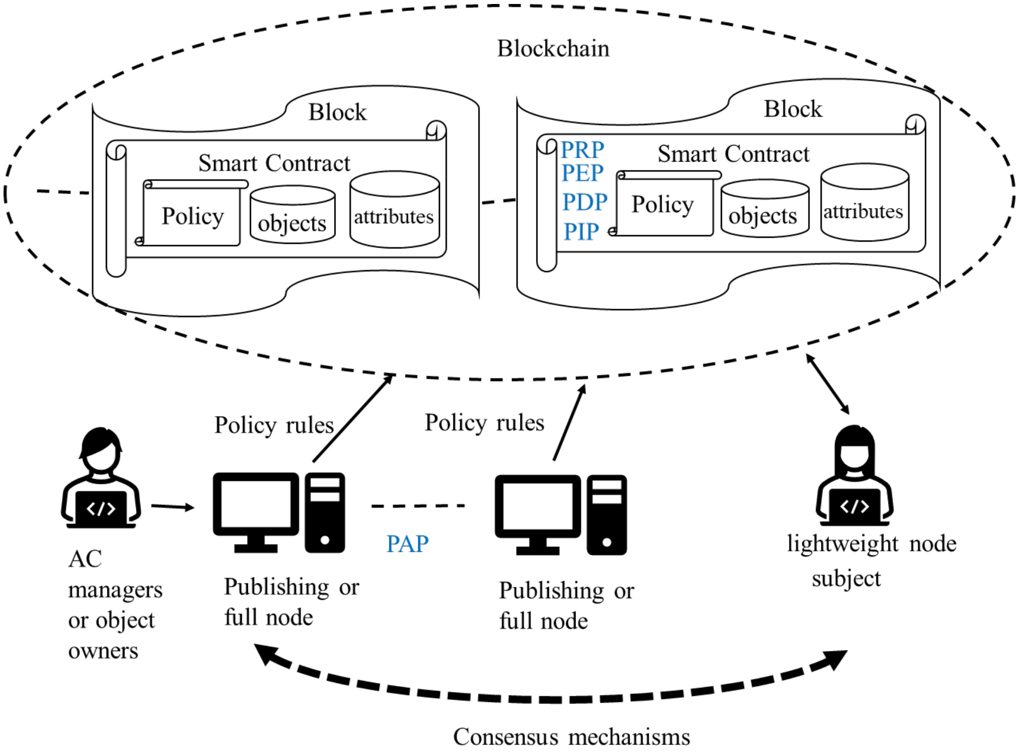

Figure 7 — Examples of Figure 2d with attribute source options

Architectures for blockchain AC systems offer flexibility based on the AC policy models enforced, such as separate blockchain networks for the separation of duty (SoD) policy model or external expansion of the AC system, which connects off-chain oracles for accessing AC data provided by third parties. Note that the architecture of a blockchain AC system is independent from the AC policy models (e.g., ABAC, RBAC, Capability Bases AC (CBAC) [GPR]) that the AC system intends to apply. For example, if a CBAC model is applied, then the policy rules in Figure 2 should be replaced by access tokens.

To ensure AC data security, functions to satisfy the following three security requirements may also need to be included [SP205]:

1. The semantic and syntactic correctness (i.e., veracity of AC data) needs to be ensured or trusted. If such data are from out-of-the-chain sources, an authority for oracle needs to be applied to validate and oversee the correctness of the data. However, if it is provided by different AC nodes, then multiple authorities working in coordination can take part in validating different sources or functions embedded in the smart contracts that need to be developed for the tasks [GMS].

2. In addition to secure transmission and repositories of AC data in the blockchain, inherited hash cypher schemes may be required to avoid exposing vulnerabilities or other types of malicious actions performed by unauthorized entities in AC notes or smart contracts. Smart contracts may also be created to define the secure communications between AC notes for AC data owners, creators, or managers.

3. Cache synchronization and failover/backup capabilities for readiness of the AC system refers to the frequency of refresh for AC data change. A blockchain AC system needs to adequately perform AC data update and retrieval frequencies to ensure that a recent set of AC data in question is cached in the blockchain if the most current AC data from authoritative sources or repositories cannot be accessed during an emergency (e.g., low bandwidth, loss of service).

4 Access Control Model Support

AC systems are basically categorized as Discretionary AC (DAC), which leaves a certain amount of AC to the discretion of the object's owner or anyone else who is authorized to control the object's access. In general, all AC policies other than DAC are grouped under the category of non-discretionary AC (NDAC). As the name implies, policies in this category have rules that are not established at the discretion of the user. NDAC establishes controls that can only be changed through administrative action, not by subjects. For example, a capability list is a popular model of DAC, and Identity based AC (IBAC) [IR7316], RBAC, ABAC, and CBAC models are popular examples of NDAC. In general, permissionless blockchains are suitable for DAC implementations, and permissioned blockchains are preferred for NDAC implementations for their control mechanisms.

As part of the mandatory nature of NDAC, consensus mechanisms of permissioned blockchains are mostly required so that only permitted AC administrators or security officials are allowed to create and modify AC rules through the restricted publishing of AC nodes, such that the consensus mechanism is restricted to general subjects. Note that the coordination of the permitted AC nodes can be centrally managed by a designed AC node, an out-of-the-chain process, or smart contracts published by authorized administrators.

For DAC policy models, the consensus mechanism configured for permissionless blockchain needs to be available to all authenticated subjects who are usually also object owners and who can use publishing AC nodes for managing policy for the authorized objects. However, for a large number of subjects, the mechanism needs to consider performance and operation requirements. For example, in general service environments, the consensus mechanism needs to be fair for generating and updating AC data for all publishing or full AC nodes. The system also needs to ensure that AC nodes can only manage policy rules associated with the object owned by the subject.

An example policy model that supports NDAC for resource-constrained devices (e.g., size, battery energy, processing speed) on an IoT network is the CBAC, which is relatively lightweight because it uses a communicable and unforgeable token for access rights associated with devices. If the CBAC is implemented by a traditional AC mechanism, it is inefficient to satisfy AC data management and AC processes due to the scale and heterogeneity of IoT devices. The reason for this is that tokens can only be granted to one subject, which makes them difficult to specify centrally and in advance. Further, devices have to use the tools provided by a central AC server or a third party to manage their AC data, which may end up with a single point of failure and privilege leakages [BXANL]. These issues can be eliminated by the blockchain system where tokens for AC data management and AC processes are distributed to each IoT device hosted in an AC node.

[PDA] published a survey of blockchain AC systems compared to traditional AC mechanisms for the implementations of RBAC, ABAC, and CBAC policy models for the IoT AC system. As shown in Table 1, the survey presents capabilities to satisfy the listed general requirements of IoT networks, including scalability, ease of use, data trust, security, and cross-domain control, which are also applicable to other AC systems that enforce the policy models.

Table 1 – Comparison of IoT AC system capabilities for general access control requirements by blockchain and traditional mechanisms enforcing RBAC, ABAC, and CBAC policy models

AC Requirements	Capabilities of traditional AC mechanisms implementing AC policy models (in parenthesis)	Capabilities of Blockchain AC systems implementing any of the RBAC, ABAC, CBAC models
Scalability	Low (RBAC) , Medium (ABAC), High (CBAC)	High
Ease of use	Medium (RBAC), High (ABAC, CBAC)	High
Architecture	Centralized (RBAC, ABAC), Distributed (CBAC)	Distributed
Data Trust	Low (CBAC), Medium (ABAC), High (RBAC)	High
Continual Control	Medium (RBAC), High (ABAC, CABC)	High
Security	Low (CBAC), Medium (ABAC), High (RBAC)	High
Cross-domain AC	Yes (CBAC), No (RBAC, ABAC)	Yes

In addition to the static policy models listed in Table 1, dynamic policy models can also be supported through smart contracts. For example, historical policies regulate access permissions by historical access states or recorded and predefined series of events. The representative models for this type of AC policy are Chinese Wall and Workflow [IR7316], which can be best described by synchronous or direct specification and expressions of a finite state model. For instance, a synchronous algorithm specified a policy of Chinese Wall model where there are three conflict of interest groups – C_1, C_2, and C_3 – for the access of object groups O_1 and O_2. Instead, implemented in a traditional AC mechanism that relies on a central process to monitor each transition of the entire AC states, the blockchain AC system can specify and enforce the policy via smart contracts, which every AC node can execute to maintain the policy states. The following is an example algorithm for the smart contract code for the Chinese Wall policy model.

```
Contract Chinese_Wall {
    Public variables {
        next_state {1,2,3}:= 1;
        subject_attribute {$C_1$, $C_2$, $C_3$};
        object_attribute {$O_1$, $O_2$};
        permission {grant, deny};
        // a FSM of state, subject attribute, object attribute, and permission //
    }
    Function Public Access (state, subject attribute, object attribute) {
        IF next_state == 1;
```

```
                CASE {
                        subject _attribute == $C_1$ AND object_ attribute == $O_1$: next_state =2;
                        permission = grant;
                        subject _attribute == $C_2$ AND object_ attribute == $O_1$: next_state =2;
                        permission = grant;
                        subject _attribute == $C_3$ AND object_ attribute == $O_1$: next_state =2;
                        permission = grant;
                        subject _attribute == $C_1$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        subject _attribute == $C_2$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        subject _attribute == $C_3$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        OTHERWISE: permission = deny;
                }
                IF next_state == 2;
                CASE {
                        subject _attribute == $C_1$ AND object_ attribute == $O_1$: next_state =2;
                        permission = grant;
                        subject _attribute == $C_2$ AND object_ attribute == $O\ _1$: next_state =2;
                        permission = grant;
                        subject _attribute == $C_3$ AND object_ attribute == $O\ _1$: next_state =2;
                        permission = grant;
                        OTHERWISE: permission = deny;
                }
                IF next_states == 3;
                CASE {
                        subject _attribute == $C_1$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        subject _attribute == $C_2$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        subject _attribute == $C_3$ AND object_ attribute == $O_2$: next_state =3;
                        permission = grant;
                        OTHERWISE: permission = deny;
                }
                ELSE permission = deny;
                RETURN permission;
        }
}
```

5 Considerations

This section discusses considerations for the implementation of the blockchain AC system from the perspectives of the management, security, privacy, performance, and standardization of AC systems.

5.1 Management Considerations

Blockchain AC system management needs to coordinate with the business and resource requirements of the system. For instance, a federated AC system may spread over multiple organizations for cooperation and communication between participating organizations. Hence, AC policies needs to be flexible and fine-grained. A blockchain AC system can transform the policy evaluation process to executable smart contracts so that each organization can control its own system while communicating with other federated organizations. Optionally, some federation schemes may use the blockchain as a database for storing only the policies but not use the blockchain for access enforcement, such that PDP and PEP functions are performed off-the-chain. However, the main problems of the traditional mechanism, like single point of failure, will be inherited [GBHC]. In addition, considerations should include governance frameworks for legal processes, applicable smart contracts, and the responsibilities of participating AC nodes even though the AC system is based on permissioned blockchains.

Another challenge of managing blockchain AC systems is to develop a trust management and evaluation framework for the decentralization of constrained resource systems, such as an IoT network, where each AC node embedded in a device has limited battery power, memory capacity, and processing speed, and it is often impossible to store extensive interaction history or employ heavy-weight security functions (e.g., microservice of mesh service for SecDevOps implementation).

General AC management requirements, such as allowing runtime policy rule changes and policy administration delegation, may further complicate the design of the blockchain AC system, especially the consensus mechanisms and smart contract functions [IR7874].

5.2 Security Considerations

Any vulnerability of a blockchain AC system on the level of the entire system or an underlying function of a smart contract can be hacked (e.g., reentrancy vulnerability) or misused. For instance, the publicly available smart contract's byte code might generate erroneous system state data that will be securely logged on the blockchain. The only way to fix errors is to delete, correct, and redeploy the entire smart contract. Thus, it is necessary for smart contracts to be correctly deployed (i.e., they work as intended by the developer and cannot be exploited by attackers).

Optimizing smart contract codes can effectively reduce potential vulnerabilities and ensure the efficient execution of contracts. For instance, running smart contracts in parallel can speed up contract execution but requires the consideration of how to execute contracts that depend on each other at the same time (especially for dynamic AC policy models). Further, smart contracts might require communicating with out-of-chain services, such as receiving AC data from a PIP host, and rely on the oracle of off-chain resources from trusted third parties to retrieve the data and then push them to the blockchain at predetermined times. Although existing oracles are well-tested, their use may introduce a potential point of failure (e.g., an oracle might be unable to push out or provide erroneous data) [KLG].

Due to the tamper-evident and tamper-resistant design of blockchain systems, the system performance evaluation should include extensive and possibly expensive reviews of the smart contract performed by experts before its deployment [SGLSFB, KLG]. If smart contracts are required to provide a way to report and correct any errors, then the system should allow actions to nullify and replace a smart contract.

Protocol of the consensus mechanism is another security concern of vulnerability. For example, PoS mechanisms are vulnerable to attacks such as nothing-at-stake, grinding, long-range, and stake bleeding attacks. PoW and PoS mechanisms may cause low throughput and long transaction confirmation delay, leading to weak consistency problems because an AC process cannot be finalized until its block reaches a certain depth in the blockchain. These might degrade the performance of AC process, so consistency – including common-prefix, chain growth, and chain quality properties of the consensus mechanism – needs to be considered [PDA].

Migrating from a legacy centralized system to a blockchain AC system needs to address the changes in security assumptions and the assurances of tools and communication protocols for the new decentralized architecture. For example, reimplementing a system using a smart contract language requires changes to secure programming practices, or changing a legacy protocol to Hypertext Transfer Protocol Secure (HTTPS) requires new communications acknowledgements.

5.3 Privacy Considerations

Storing AC data and logs on the blockchain raises questions of privacy as all subjects can see all entries, and auditable access history in the blockchain can violate user privacy. The exposed information may also provide information for attacking systems or users. If regulations require AC data owners who are accountable for all data privacy, then instead of storing private data on the blockchain, consider storing index numbers that are tied to private data in an off-the-chain system. Thus, subjects can own, secure, and even delete their privacy data. Otherwise, methods or tools facilitating cryptography need to be considered for privacy protection [GT].

5.4 Performance Considerations

The performance of a blockchain AC system should consider process throughput and confirmation delay. The former refers to the number of AC access requests/processes that the AC system can confirm per unit time (e.g., the Ethereum blockchain can verify 14 transactions per second, which is slow compared to Visa, which can handle up to 24,000 transactions per second), while the latter measures the time it takes for them to be finalized. A blockchain AC system may generate a large volume of access requests that need to be processed and handled or a large number of AC nodes that generate a large amount of data to form an oversized chain (e.g., IoT AC system) [ZZH]. In such cases, AC requests/processes may be constrained by the fact that blockchain data can only be added, not deleted. Due to the scalability limitation of the block memory size, a reduction in performance (bottleneck for the end users) is inevitable. The consequences will be increased synchronization time, increased commission fees (if required), and increased time to confirm an AC request/process [PDA, KLG].

Scalability is another performance concern, and one of the major affecting factors is the consensus mechanism's consistency and liveness. Consistency means that legitimate AC nodes have an identical view of the AC system state, and liveness means that a valid AC process is sure to be processed and written on the blockchain for a certain period along with the ability to respond to and recover from an attack. To address these issues, a consensus mechanism can be adjusted to decrease resource consumption, especially for resource-constrained AC systems such as IoT AC systems (it has some disadvantages on security regarding to immutability). For example, the AC system uses a permissionless type of blockchain. Although the PoW algorithm enables security in the blockchain, it wastes resources. Thus, consider switching from the PoW algorithm to others that can improve scalability as well as lower fees and energy costs (if required) for AC processes [GBHC, KLG], such as proof-of-activity (PoA) or delegated proof-of-stake (DPoS). The selection of consensus mechanisms also needs to comply with the AC policy models applied. When an AC node is under attack (e.g., DDoS), a blockchain AC system may not have sufficient performance for AC functions, such as the distribution of policy and capturing log files in real time. Alternative communication channels or appropriate safeguard mechanisms might be required to handle such situations.

As a result, a blockchain AC system must consider hardware and communication limitations in order to economically design an architecture for its memory that is lightweight with limited computing and communication power and storage capabilities, especially for systems with massive AC nodes. Considerations must also include a fast response consensus mechanism to comply with performance requirements.

5.5 Standardization Considerations

A blockchain AC system may handle a variety of devices, infrastructures, and governments. For example, a system may have different types of AC data (e.g., types of subject or object attribute values) or proprietary protocols between AC nodes that make it difficult to communicate using a single blockchain platform. Thus, assurance and standardization of the guidelines allow for universal acceptance of the AC data and smart contracts for AC processing [RGD, PDA].

6 Conclusion

The rapid development and wide application of distributed network systems have made network security – especially access control and data privacy – ever more important. Blockchain technology offers features such as decentralization, high confidence, and tamper-resistance, which are advantages to solving auditability, resource consumption, scalability, central authority, and trust issues – all of which are challenges for network access control by traditional mechanisms.

Blockchain is particularly applicable to access control for network systems where authorization processes are based on subject and object attribute data because it improves security, flexibility and scalability for management, and the enforcement of access control data and processes. It also improves the capability of organizations to verify and audit access control processes with function calls to track the global access control system state. Blockchain system components can function as a resource repository or executable process, allowing it to be neutral for access control policy models. As blockchain access control systems address some challenges from traditional mechanisms, the management, security, privacy, performance, and standardization of the implementation need to be considered.

This document presents general information for blockchain access control systems from the views of blockchain system properties, components, functions, and supports for access control policy models. Considerations for implementing blockchain AC systems are also included.

References

[BXANL] Bouras MA, Xia B, Abuassba AO, Ning H, Lu Q (2021) IoT-CCAC: A Blockchain-Based Consortium Capability Access Control Approach for IoT. *PeerJ Computer Science* 7:e455. https://doi.org/10.7717/peerj-cs.455

[DMMR] Di Francesco Maesa D, Mori P, Ricci L (2019) A Blockchain Based Approach for the Definition of Auditable Access Control Systems. *Computers & Security* 84(July):93-119. https://doi.org/10.1016/j.cose.2019.03.016

[GBHC] Ghaffari F, Bertin F, Hatin J, Crespi N (2020) Authentication and Access Control Based on Distributed Ledger Technology: A survey. *2nd conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS 2020)* (IEEE, Paris), pp 79-86. https://doi.org/10.1109/BRAINS49436.2020.9223297

[GMS] Guo H, Meamari E, Shen CC (2019). Multi-Authority Attribute-Based Access Control with Smart Contract. *Proceedings of the 2019 International Conference on Blockchain Technology (ICBCT 2019)* (ACM, Honolulu, Hawai'i), pp 6–11. https://doi.org/10.1145/3320154.3320164

[GPR] Gusmeroli S, Piccione S, Rotondi D (2013) A Capability-Based Security Approach to Manage Access Control in the Internet of Things. *Mathematical and Computer Modelling* 58(5–6), pp 1189-1205. https://doi.org/10.1016/j.mcm.2013.02.006

[GT] Grant Thornton US (2020) Blockchain and Privacy: How Do You Protect Data That's Distributed?[video]. Available at https://www.youtube.com/watch?v=hcXz3EQoDF8

[INCITS] InterNational Committee for Information Technology Standards (2020) *INCITS 565-2020 – Information technology – Next Generation Access Control* (INCITS, Washington, DC). Available at https://standards.incits.org/apps/group_public/project/details.php?project_id=2328

[IR7316] Hu VC, Ferraiolo DF, Kuhn DR (2006) Assessment of Access Control Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 7316. https://doi.org/10.6028/NIST.IR.7316

[IR7874] Hu VC, Scarfone KA (2012) Guidelines for Access Control System Evaluation Metrics. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 7874. https://doi.org/10.6028/NIST.IR.7874

[IR8202] Yaga DJ, Mell PM, Roby N, Scarfone KA (2018) Blockchain Technology Overview. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8202. https://doi.org/10.6028/NIST.IR.8202

[KLG] Khan SN, Loukil F, Ghedira-Guegan C, Benkhelifa E, Bani-Hani A (2021) Blockchain Smart Contracts: Applications, Challenges, and Future Trends. *Peer-to-Peer Networking and Applications* 14:2901-2925. https://doi.org/10.1007/s12083-021-01127-0

[Kuhn] Kuhn DR, (2022) A Data Structure for Integrity Protection with Erasure Capability. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Cybersecurity White Paper (CSWP) 25. https://doi.org/10.6028/NIST.CSWP.25

[LQLL] Liu Y, Qiu M, Liu J, Liu M (2021) Blockchain Based Access Control Approaches. 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) (IEEE, Washington, DC), pp 127-132. https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00032

[PDA] Pal S, Dorri A, Jurdak R (2021) Blockchain for IoT Access Control: Recent Trends and Future Research Directions. *arXiv preprint*. Available at https://arxiv.org/abs/2106.04808

[RBAC] Ferraiolo DF, Kuhn DR (1992) Role-Based Access Controls. *Proceedings of the 15th National Computer Security Conference* (NIST, Baltimore, MD), pp 554-563. Available at https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1992/10/13/role-based-access-controls/documents/ferraiolo-kuhn-92.pdf

[RGD] Rani PL, Guru Gokul AR, Devi N (2021) Blockchain-Based Access Control System. *Transforming Cyber Security Solution Using Blockchain,* eds Agrawal R, Gupta N (Springer, Singapore), pp 91-114. https://doi.org/10.1007/978-981-33-6858-3_6

[SGLSFB] Schiffl J, Grundmann M, Leinweber M, Stengele O, Friebe S, Beckert B (2021) Towards Correct Smart Contracts: A Case Study on Formal Verification of Access Control. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies (SACMAT '21)* (ACM, [Virtual], Spain) pp 125–130. https://doi.org/10.1145/3450569.3463574

[SP162] Hu VC, Ferraiolo DF, Kuhn DR, Schnitzer A, Sandlin K, Miller R, Scarfone KA (2014) Guide to Attribute Based Access Control (ABAC) Definition and Considerations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-162, Includes updates as of August 02, 2019. https://doi.org/10.6028/NIST.SP.800-162

[SP205] Hu VC, Ferraiolo DF, Kuhn DR (2019) Attribute Considerations for Access Control Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-205. https://doi.org/10.6028/NIST.SP.800-205

[XACML] OASIS eXtensible Access Control Markup Language (XACML) TC (2020) *Organization for the Advancement of Structured Information Standards*. Available at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[ZZH] Zhai P, Zhang L, He J (2021) A Review of Blockchain-Based Access Control for the Industrial IoT. *CONVERTER* 2021(3):308-316. https://doi.org/10.17762/converter.62