Future Generation Computer Systems 101 (2019) 420–433

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Privacy aware decentralized access control system

Sehrish Shafeeq *, Masoom Alam, Abid Khan

Cybersec Lab Dep of Computer Science, COMSATS University, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 29 December 2018
Received in revised form 26 May 2019
Accepted 19 June 2019
Available online 26 June 2019

Keywords:
Blockchain
ABAC
Access control
Tangle
Internet of things
IOTA

ABSTRACT

IoT security and privacy have proven to be a significant challenge. The traditional access control protocols are not suitable for IoT mainly due to a massive scale, ubiquitous connectivity and distributed nature. Blockchain based access control approaches provide decentralized security but they involve scalability problem, high transaction fees, a significant delay, and computational overhead that is not acceptable for resource-constrained IoT devices. Moreover, data published on the blockchain are public which is not ideal for many scenarios. In this paper, we proposed a new decentralized access control system based on the Tangle which empowers the users to dictate the access to their resource. In our proposed decentralized access control model the policies and access rights are published on the Tangle which guarantees distributed auditability and prevents the user from fraudulently denying the granted access rights. The main contribution of the paper is to provide privacy of the policy by leveraging Masked Authenticated Messaging (MAM) data communication protocol. The proposed work is validated by implementation and is tested with AVISPA tool which confirms security in the presence of the intruder.

1. Introduction

Nowadays, there has been a tremendous surge of interest in the Internet of Things (IoT). Today our lives are surrounded by billions of devices that make IoT a reality and this giant network of devices is expected to spread its wings in coming years [1–3]. The rapid increase in the number of IoT devices resulted in the exponential increase in the volume of data. To deal with this explosive growth in data especially mobile data, femtocells are deployed to boost the capacity of the network [4]. Advancement in the 5G network is converging with a variety of applications. For example, with the improvement in multimedia data delivery, Unmanned Aerial Vehicles (UAVs) are expected to be involved in various Industrial Internet of Things (IIoT) applications [5]. Intelligent Transportation Systems (ITSs) are also being developed to facilitate emergency situations in a smart city such as severe collisions, earthquakes, etc. [6].

However, the exponential increase in the rate of IoT devices deployment may incur crucial security issues such as access to confidential data, unauthorized alteration or denial of service [7]. Hence the IoT security and privacy issues are pivotal. The security and privacy are key obstacles to the widespread adoption of the IoT paradigm, in order to overcome these issues several studies have been conducted to offer efficient solutions against security and privacy threats. Accordingly, authors in [8] proposed a secure mutual authentication approach for Wireless Sensor Networks (WSNs) that mitigates several security threats such as replay, user masquerading and privileged insider attack. Authors in [9] have proposed an agile framework in WSNs that resolves various confidentiality and privacy challenges while collecting data by leveraging elliptic curve cryptography. Authors in [10] discussed the user motivation behind becoming part of the IoT network and the correlation between the user surrounding context and the application usage. Authors in [11], discussed network security vulnerabilities, attacks, threats and risks in firewall, switches, and routers. The paper also provides a network security model to mitigate the internal and external attack and threats in the IoT era. To overcome these issues, Access Control Systems (ACS) are defined, which aim to prevent unauthorized access and it has been considered as a vital research area in the IoT [12–14].

Access control schemes deployed in the IoT are *Role Based Access Control (RBAC)* [15], *Organization Based Access Control (OrBAC)* [16], *Trust-Based Access Control* [17], *Capability Based Access Control (CapBAC)* [18,19] and *Attribute Based Access Control (ABAC)* [20,21]. In the RBAC schemes, the access is restricted based on the roles (e.g. administrator, guest etc.) of the subjects (e.g. entities that access the resource) within an organization [22,23]. The OrBAC defines an access control policy independently of the implementation with roles, activities, and views. The subjects are abstracted into roles, activity is a set of actions and a view is a set of objects to be protected. The Trust Based Access Control grants privileges depending on the trust attribute which is an owner's assessment of results for the requester and trust condition that must be satisfied by the requester trust attribute. The

* Corresponding author.
E-mail address: sehrixhshafeeq@gmail.com (S. Shafeeq).

https://doi.org/10.1016/j.future.2019.06.025
0167-739X/

CapBAC scheme grants access to subjects based on the concept capability which is an unforgeable token of authority (e.g. key). The ABAC scheme grants access rights based on the policies which combine subject's attributes, object attributes and environment attributes to define a set of rules. Our proposed decentralized access control system leverages ABAC on the grounds that it is a flexible approach that can implement access control policies limited only by a number of the available attributes and the computational language; this makes ABAC ideal for distributed or volatile environments [20].

It is notable that in centralized access control system access rights are usually granted by a centralized entity which results in a single point of failure [24]. To address this issue, the decentralized access control system is required where access rights are granted by the requested IoT object rather than a centralized entity. In recent years, researchers leveraged blockchain technology to achieve distributed and trustworthy access control in the IoT. A blockchain is a revolutionary technology behind popular cryptocurrencies like Bitcoin [25], Ethereum [26]. A blockchain is decentralized, immutable, trustless, public data structure. At first, blockchain was mainly developed to provide secure cryptocurrency without a central entity but nowadays researchers found its capabilities far beyond cryptocurrency. The blockchain vastly improved existing technologies and enables new applications never formerly practical to be deployed [27].

Each block of the blockchain stores the transactions. Each block has a limited size and frequency, such as in the Bitcoin permissible size of the block is 1 MB and the average block creation time is 10 minutes. Bitcoin blockchain maximum throughput[1] is 3.3–7 transactions per second [28]. Thus, it can store a limited number of transactions and constrain the network throughput. As the Bitcoin network becomes more popular; miners prefer high fee transactions. Thus, the users have to pay high transaction fees to put the transaction in the block. This is not acceptable for the IoT in which devices communicate with each other frequently. The demand for micro-payments would increase with the growth of the IoT and paying high transaction fees for low-value transaction is not feasible.

1.1. Motivation

Many decentralized blockchain based access control models have been proposed in the literature that empower the users to control and own their data. However, applying blockchain to the IoT is complicated because of several key challenges including scalability, high resource requirements, high transaction fees, and transaction delays. Moreover, the data stored on the blockchain are public, thus offers no privacy which is sometimes crucial.

While the blockchain based access control schemes achieve data confidentiality, privacy preservation for the user remains an unsolved issue. Consider a scenario, in the decentralized blockchain based access control schemes, Bob stores electronic medical record access control policy in the blockchain via a transaction. Although an attacker cannot read the electronic medical record, it can read the access control policy stored on the blockchain. If the access control policy states that the doctor expert in the hepatitis disease has access, an attacker can infer that Bob may carry hepatitis without having access to Bob electronic medical record. This means an attacker can deduce information about Bob's disease by analyzing his access control policy, even without having access to the electronic medical record. Thus, the privacy of not only data but also of policy is crucial [29].

To address these issues, a new decentralized access control system is required which offers privacy and scalability. Thankfully, there is a new decentralized and tamper-proof distributed ledger, known as the Tangle [30] that is used in the IOTA [31]. The Tangle is specifically designed for the resource-constrained IoT devices. The Tangle does not involve blocks and miners; therefore, no transaction fees are involved.

1.2. Contribution

This paper proposes an access control framework based on the Tangle technology, which is used to transfer access rights among users. The resource owner publishes access control policies on the Tangle in the encrypted format and only the user having secret decryption key can read the policies. Thus, protects the privacy of the access control policy that is essential in the areas where privacy is mandatory by law, not only on the data but also who can access it is highly sensitive.

The main contributions of the paper are summarized as follows:

- According to the best of our knowledge, this is the first decentralized access control framework that guarantees the privacy of the policy.
- The owner of the resource can easily track whom access has been delegated and can revoke access at any time.
- The user does not have to pay high transaction fees for the transfer of access rights or policy publication.
- Our decentralized access control scheme is highly scalable because of underlying the Tangle technology.
- Our approach is specifically designed for resource constrained IoT devices.

1.3. Organization

The rest of the paper is organized as follows. Section 2 describes a survey of related work on the subject at hand. Section 3 explains the basic concepts necessary to understand the proposed model. In Section 4, we introduce the decentralized access control framework. Section 5 describes a case study for the proposed work. Section 6 provides formal security and privacy proof. Section 7 explains the conclusion and our future work.

2. Related work

A numerous blockchain based decentralized access control frameworks have been proposed for adoption since 2016, as depicted in Table 1. An effective literature review is an essential feature to uncover areas where research is required. Proposed decentralized access control frameworks are summarized as follows:

Ouaddah et al. [32] for the first time introduced the *FairAccess*: an access control framework that leverages the consistency of the blockchain. FairAccess enables the *Resource Owner (RO)* to control the data. Their framework utilizes a scripting language for evaluation of the policy to make a decision. FairAccess framework all data, including policies are stored in the blockchain, this data were publicly visible which sabotages the privacy.

Ouaddah et al. [33] proposed FairAccess proof of concept by the implementation. In their typical use case scenario, the resource is a smart security camera attached with Raspberry PI. The user requests using Bitcoin core wallet. The communication between requester device (laptop) and smart security camera is through Bitcoin transactions. The requester and the RO have to pay high transaction fees to the miner. The transaction fees are sometimes higher than the actual worth of the resource accessed which is not ideal in case of resource-constrained IoT devices.

Maesa et al. [34] made use of the blockchain technology to publish access control policies. Their scheme used *OP_RETURN*

[1] The rate at which blockchain can confirm transactions.

script opcode and MultiSignature transaction of Bitcoin network to store policies. Since the policies are stored in the transactions which increases the size of the transactions. The RO has to pay a high transaction fee which depends on the size of the transaction. The RO pays for policy creation, update and revoke while the requester pays for the transfer of access rights in the form of a transaction fee.

Outchakoucht et al. [35] improved the security of the FairAccess framework with machine learning algorithms [36]. Smart contracts are used to evaluate access control policies. The smart contract trains itself and tries to acquire the optimal knowledge to make an accurate decision. The limitations of the paper are lack of privacy and blockchain validation time. The model also needs a concrete case study.

Zhang et al. [37] framework include multiple access control contracts for access control. One of the contracts is Judge Contract (JC) that implements a misbehavior judging method and returns penalty. The paper provides a case study using Ethereum based smart contract with one laptop and two Raspberry Pi. The IoT devices do not run Ethereum wallet whereas IoT gateways serve as agents for the IoT devices to manage access control. The gateways store the account of IoT devices and can sign transactions on behalf of IoT device which is a serious problem for auditing. Moreover, the compromise of gateways results in security and privacy risk.

Dukkipati et al. [38] divided the policies into two types general policies and special policies. The general policies are stored in the public database whereas special policies are stored in the private blockchain. The model stores the URL link of the policies in the blockchain. The smart contract evaluates the policy and results in the acknowledgment. They validate the model by smart traffic signal and vehicle use case scenario. The paper has a drawback related to the privacy of general policies. Maesa et al. [39] proposed an access control system based on Ethereum blockchain technology. The main disadvantages of the approach are performance and high-cost smart contract function calls.

To sum up, a new access control framework is required that overcomes the limitations of the traditional access control frameworks. It has to achieve the following goals: Firstly, it provides privacy-aware access control in the context of policies. Guarantee of privacy of the policy is essential in areas where privacy is required by legislation (e.g. bank, healthcare) where not only a resource but also who can access it is very sensitive information. Secondly, in the previous blockchain based access control schemes resource owner and the requester have to pay high transaction fees to the miner which is unfeasible in case of IoT where communication between devices occurs frequently. In our proposed scheme resource owner, and the requester do not have to pay transaction fees. Thirdly, in the previous blockchain based frameworks on average one transaction confirmation takes 10 minutes, but ground truth is that only 63.2% of the time a subject can expect first confirmation within 10 minutes. Thus, a scalable access control framework is required for IoT devices.

3. Background

This section introduces the Tangle and MAM to lay the groundwork for the proposed access control framework.

3.1. Attribute-based access control

Traditionally, access control has been based on the identity of a user requesting to perform an operation on an object. Practitioners have noted that associating capabilities directly to users or groups to control access is often bulky to manage. Another option is to grant access based on the attributes of the user, object and environment conditions that can be globally accepted and more appropriate to the policies. This approach is often known as Attribute Based Access Control (ABAC).

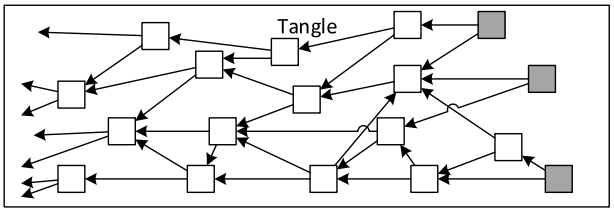

Fig. 1. The Tangle.

ABAC is logical access control that restricts access to the object (resource) by evaluating rules against the attributes of the subject (requester), the object (e.g. a file), actions (e.g. read) and environment relevant to a request. The access rules can be defined without specifying the relationship between each subject and object. For example, subject attributes are assigned upon employment such as Alice is a *doctor* in the *cardiology department*. An object attributes are assigned upon creation such as a file with the *medical history* of a *heart patient*. The owner of an object defines access control rules using the attributes of the subjects and the objects to regulate a set of permissible capabilities, for example, all *doctors* in the *cardiology department* can view a *medical record* of a *heart patient*.

ABAC provides a more dynamic access control management capability as access decisions can be changed between requests by modifying attribute values without demanding changes to the subject/object relationships defining the underlying policy. Furthermore, an object owner can define an access control policy for an infinite number of subjects without prior knowledge of the specific subject. For example, as the new doctor joins the hospital and necessary attributes are assigned to him/her to access the required object — no modification to existing values or object attributes are required. As a result of the versatility, ABAC has engaged interest across the industry and is the swiftly expanding access control model today [40].

3.2. The tangle

IOTA [31] is a lightweight quantum resistant cryptocurrency, specially engineered for the IoT. The principal objective behind the IOTA is to serve the machine economy by enabling feeless Machine-to-Machine (M2M) transactions. The revolutionary technology behind IOTA is the Tangle, an open source distributed ledger. The Tangle uses Directed Acyclic Graph (DAG) [30] instead of the blockchain as depicted in Fig. 1. A DAG is a collection of nodes which are connected to each other by edges. Each transaction is represented by the node and an edge represents the validated transaction.

The Tangle overcomes the scalability limitation of the blockchain [25]. The Tangle can become indefinitely scalable with zero cost because each new transaction issuer approves two previous transactions which are represented by edges. Thus, each transaction issuer itself attaches a transaction to the Tangle and broadcasts it to the network which eliminates the need of miners. It is important to note that the computational capability required to attach the transaction to the Tangle is minimal (e.g. hashcash [41]), in this way transaction can be attached by any resource-constrained IoT device.

Microtransactions are not feasible over the Bitcoin, Ethereum network because of high transaction fees. On the contrary, the Tangle empowers microtransactions. Due to the fact that miners

Table 1
A comparative study of existing techniques.

Technique	Access control technique	Smart contract	Ledger	Privacy	Transaction fee	Transaction confirmation delay	Implementation
Ouaddah et al. [32]	–	Yes	Bitcoin blockchain	No	High	10 min	No
Ouaddah et al. [33]	Permission based access control	No	Bitcoin blockchain	No	High	10 min	Yes
Maesa et al. [34]	ABAC	No	Bitcoin blockchain	No	High	10 min	No
Outchakoucht et al. [35]	Organization based access control	Yes	–	No	–	–	No
Zhang et al. [37]	–	Yes	Ethereum blockchain	No	High	10–20 s	Yes
Dukkipati et al. [38]	ABAC	Yes	–	Partial	–	Yes	Yes
Maesa et al. [39]	ABAC	Yes	Ethereum blockchain	No	Yes	10–20 s	Yes
Our approach	ABAC	No	IOTA Tangle	Yes	Zero	Quasi-infinitesimal. More nodes: faster network.	Yes

are abolished, it results in a zero transaction fee. This makes the possibility of microtransactions even more realism.

IOTA is resistant against quantum computer attacks in contrast to the Bitcoin and Ethereum. With the advancement of technology, the quantum computing era seems to be near when systems that are thought to be secure, may in fact compromised by quantum computers The reason behind IOTA quantum resistance is that IOTA uses Winternitz One Time Signature (WOTS) [42] scheme, which is quantum resistant. The use of one-time signature scheme, however, opens IOTA to some other security problems such as address reuse after snapshot compromises private key of the user [43].

In our proposed approach, the Tangle is used to publish policies and the rights exchange. The Tangle serves as a distributed and immutable repository for access control policies. Using Tangle distributed ledger, our system provides an immutable log of important security events such as access rights grant, revocation and delegation events. The Tangle ensures non-repudiation and integrity of the information.

3.3. The MAM channel

The Masked Authenticated Messaging (MAM) is the IOTA library that allows users to transmit encrypted data to the Tangle. To publish a message, a user needs to do a small amount of proof of work to attach it to the Tangle.

MAM uses a Merkle tree based signature scheme [44] to sign the cipher digest of an encrypted message. The Merkle root is used to compose *channel id* to which an encrypted message is published. To make it easy for the subscribers to follow the message stream each published message contains the next Merkle root because a single tree can be used to sign a limited number of messages. To achieve privacy of the published message, each message is encrypted using a one-time pad that comprises a Merkle root. When a subscriber retrieves a message from the Tangle, she first authenticates the message by verifying the signature. If the signature verification succeeds, then the subscriber unmasks the message using the symmetric key *Ek*. A unique feature of MAM is forward secrecy. If a user gains access to a channel at time t, she cannot read messages before her point of entry with the information contained in current and future messages.

MAM can be used in multiple modes (i.e. public, private, or restricted) to control data visibility. In *public mode*, the Merkle root is the address of a transaction Eq. (1). Thus anyone who finds the transaction can decrypt the message since the address is the Merkle root. In *private mode*, the hash of the Merkle root is used as the address of a transaction Eq. (2). As a consequence, only users who have the Merkle root can the decrypt message. If an adversary stumbles upon the transaction on the Tangle, it cannot extract encrypted data since the address is the hash of the Merkle root. In this case, only those users who are given the Merkle root can locate the message on the Tangle and decrypt its contents. In *restricted mode*, the hash of the Merkle root and an *authorization key* is used as the address of a transaction Eq. (3). Only users who know both the Merkle root and the authorization key can produce the address and locate the message on the Tangle. The message is decrypted using the authorization key. The authorization key enables the publisher to restrict access to future messages. This is possible because a change of the authorization key changes the address of future messages. This makes it very difficult for the next transaction in the stream to be located on the Tangle without knowing the authorization key.

In public mode

$$channelid = Merkleroot \tag{1}$$

In private mode

$$channelid = Hash(Merkleroot) \tag{2}$$

In restricted mode

$$channelid = Hash(Merkleroot + authorizationkey) \tag{3}$$

In each mode, channel id is the address of the MAM transaction on the Tangle.

4. Proposed access control framework

This section explains the proposed access control framework. Section 4.1 gives a brief overview of our proposed novel approach. In Section 4.2 and Section 4.3, we describe the architecture of the proposed access control scheme.

To help the reader, we have provided acronyms used in our proposed framework in Table 2.

Table 2
Main acronym and terms used in this paper.

Term	Meaning
RO	Resource Owner
SR	Service Requester
ABAC	Attribute Based Access Control
DAG	Directed Acyclic Graph
MAM	Masked Authenticated Messaging
WOTS	Winternitz One Time Signature Scheme
WSN	Wireless Sensor Network
PDP	Policy Decision Point
P-MAM	Policy MAM channel
PEP	Policy Enforcement Point
PIP	Policy Information Point
AVISPA	Automated Validation of Internet Security Protocols

4.1. Architecture overview

Our proposed decentralized access control scheme empowers the RO to define and manage access control over her resources. The RO defines the security policies and the level of the authorization granularity of the resources. To grant access rights, the RO sends the authorization token to the requester. The RO grants access token only if the requester meets the conditions defined in the access control policy. Thus, a requester can access the resource if she has an authorization token.

The access rights are defined by ABAC which is a fine-grained and scalable access control model. Service Requester attributes are issued, stored and managed under her home domain. There is no user attributes provided to the RO, which protects the privacy of the user. All the SR attributes are verified at PDP end only and the PDP does not have the SR authentication credentials. The action attributes i.e. “read" and/or “write" is managed by RO or any other authority assigned by the RO. Similarly, the RO is responsible for the management of resource attributes and environment attributes i.e. the current time and location etc. Policies are a set of rules that need to be satisfied to get access to the protected resource. The owner of the object has the right to create a policy that describes who can perform an action on the object, what operations can be executed on the object and under what context subject can perform an action. If the subject satisfies conditions, then RO grants access otherwise access is denied.

The proposed framework makes effective use of the Tangle to store the policies. To ensure the privacy of the policies stored on the Tangle, we take advantage of the restricted MAM channel. Thus, no one except the entity having a secret decryption key can read the policies. The policies are stored in the encrypted format on the MAM channel. However, by sharing a secret decryption key the RO allows the *Policy Decision Point (PDP)* to read policies to make a decision.

Our access control model work as follows. The subject who wants to perform an action on the protected resource sends a request to the RO. The RO forwards the access request to the PDP. The PDP redirects the *Service Requester (SR)* to the authentication service of the SR domain for authentication. The SR is redirected back to the PDP along with her attributes. These attributes are then verified and evaluated by the PDP against the access control policies defined by the RO. The PDP makes a grant access decision based on the policy defined by the RO. The PDP then redirects the SR back to the RO along with her domain information, identity and authorization result, which grants or denies access based on the result. If the PDP permits, then the RO formulates the *GrantAccess* transaction. The requester formulates a request to the resource using *GetAccess* transaction. The PEP attaches GetAccess transaction to the Tangle. The network validates the transaction only if the access token is transferred to the requester in the previous GrantAccess transaction.

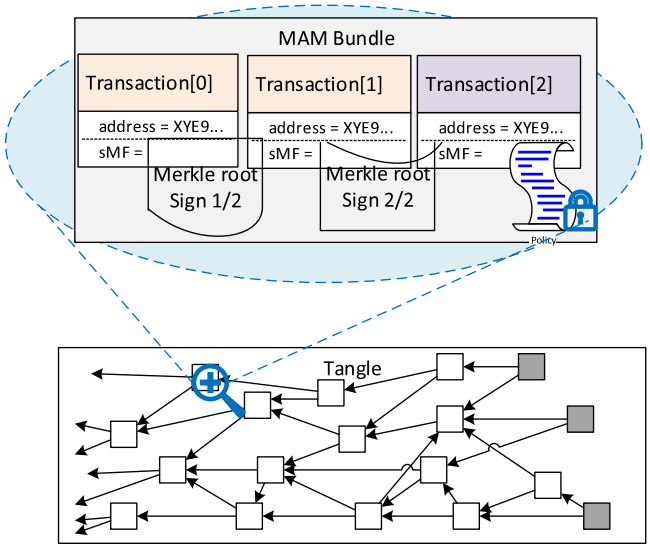

Fig. 2. Detailed view of MAM transaction.

Wireless communications and Wireless Medical Sensor Networks (WMSNs) Wireless Medical Sensor Networks (WMSNs) are under rapid research, given their wide range of applications i.e. smart grid, industry, and health applications [45]. The data collected by WMSNs networks is of paramount importance in medical diagnosis; thus only authorized users can access these data [46]. Traditional, security measures are not suitable for WSN in terms of processing overhead and energy consumption. Moreover, many state of the art security systems are centralized; therefore they are not well suited for WSNs due to a single point of failure [47], the difficulty of scale and many to one nature of traffic. Consequently, blockchain based decentralized access control framework has great potential in addressing the above-mentioned issues in an efficient manner.

Our approach guarantees the control of the RO with two facts: authorization decision made by the PDP is based on access control policies of the RO and each authorization is based on the authenticated information of the SR domain.

There are several advantages of the proposed framework in the cross-domain scenario. Firstly, the SR domain has knowledge of a subject accessing a service of another domain which provides a means of auditing without a glitch. Secondly, the SR has no authentication credentials on any domain, except her home domain which enables privacy. Thirdly, the RO can use pseudonyms of permissions and resources without exposing the real internal information to the PDP when publishing policies. Fourthly, the SR home domain can also enforce further security policies such as cross-domain access to a particular resource is only allowed to certain subjects. Therefore, the home domain may refuse to authenticate a subject after evaluating its own cross-domain policies.

Threat modeling. The main objective of our framework is to enable the RO to dictate access to her resource and prevent unauthorized access. We also focus on the privacy of the policies. Therefore, any attempt to get policies or unauthorized access to the resource is a potential threat.

We assume that PDP is an honest participant in the protocol, which makes decision-based on the RO policies. An honest PDP makes the authorization decision based on the pre-defined policies and attributes of the SR. Otherwise, there is a risk that a malicious PDP could provide false or invalid authorization decisions, leading to policy violation [48].

4.2. Policies management

4.2.1. Publishing policy

The user is allowed to access the resource only if she is allowed by the policies of the RO. Policies are a set of rules defined to get access to the resource and they are written in XACML language.

The RO creates a new *Policy MAM channel (P-MAM)* for broadcasting policies. Once the channel is created, the RO publishes a new policy by creating a MAM transaction. A zero-value transaction is created, and the encrypted policy is stored in the *SignatureMessageFragment* field. The RO formulates *PublishPolicy* transaction and propagates it through the network using the gossip protocol. The P-MAM channel messages can be of any size, however, smaller size messages yield higher potential for data integrity. If the policy is too large to be included in a single transaction, the policy issuer can create a MAM transaction that references the previous policy transaction. The advantageous feature of MAM is that transactions of the same channel are linked with each other. When a user decrypts a transaction, he/she gets the message (policy) and Merkle root for the next message stream.

The MAM bundle contains 3 transactions for storing signature and masked policy for security level 2 as depicted in Fig. 2. Each P-MAM policy transaction has mainly two parts signature section and masked policy section. The address of the transaction is the channel Id and it is shared with the subscriber node. The subscriber node uses this address to listen to the message stream published on the Tangle. Since it is a zero value transaction, the transaction value field contains 0. The RO attaches digital signatures to authenticate the P-MAM transaction contents and her identity.

The privacy of the policy is crucial. For example, it is not desirable for the patient to reveal that his medical data is accessible to an AIDS specialist or an oncologist or psychiatrist; in an enterprise, one may want to hide which files are accessible by the auditor. Therefore, a strong security compulsion is required when the RO publishes policy on the Tangle. To attain privacy of the policy, our scheme publishes policy in the Tangle using MAM restricted mode. In the restricted mode, the policy is encrypted using symmetric authorization key. The authorization key is defined by the RO (publisher). Only the subject having secret authorization key can decrypt messages. The RO can change the authorization key at any time to revoke future access of message stream. In this scenario, a subscriber cannot decrypt the message if she does not have a new authorization key.

The access control policies are categorized as *general policies* and *specific policies*. The general policy allows basic actions on the resource e.g. read. It is defined by RO once the resource is assigned to her. The requester requests the RO specific rights to be granted if she needs to perform actions other than granted in the general policy. The specific policy is defined based on the request of delegate. The general policies are stored in the Tangle without any request from the users. Thus, using varied policies saves time or memory and the number of transactions is lessened. Consider the following scenarios. (1) Suppose the patient allows the healthcare provider (personal or privileged) to read her personal care data. Now if the personal health care provider needs to perform write action on the patient data, he/she needs to request the owner to define a specific policy that meets his/her needs. (2) The government general policy allows anyone to read the status of the traffic signal. Now consider, if the vehicle i.e. ambulance needs to switch the signal from red to green, it needs a specific policy. So only the owner can grant this permission because access is sensitive, and it might affect traffic flow.

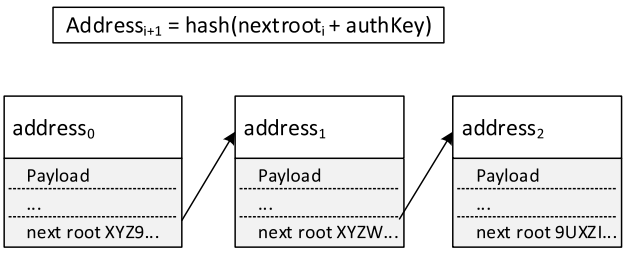

Fig. 3. Updating policy.

4.2.2. Policy update

As described earlier, to make it easy for the subscriber to follow policy update, each policy transaction on P-MAM channel contains the Merkle root of the next policy transaction. The policy issuer can update a policy any time. This simply involves publishing a new transaction on the P-MAM channel.

The owner of the P-MAM channel can publish policy update transaction because she is the only one who has the private key corresponding to the next channel id. Policy transactions broadcasted on the P-MAM channel are attached in sequential order as illustrated in Fig. 3.

The former policy conditions cannot be violated by latter policy transactions. This means resulting overall policy can be more restrictive than the original policy. This is correct since it is expected to restrict the access rights not to expand them.

4.2.3. Policy revocation

The Tangle is an immutable distributed ledger. This means once the policy is added to the ledger, it cannot be revoked. However, if some contention occurs between involved parties, then the RO can revoke their future access control policy. To do so, the RO changes the authorization key of the P-MAM, that prevents the subscribers from reading future policies as depicted in Fig. 4.

Consider an example, the patient wants to publish his/her access control policy for the healthcare provider HP_a and health care provider HP_b. The RO, general access control policy is the same for both HP_a and HP_b. When a patient wants to publish specific access policies for HP_a but not for HP_b. The patient forks the channel by using a new authorization key. To allow HP_a to read specific P-MAM stream, the new authorization key needs to be shared with the HP_a.

4.3. Rights exchange

4.3.1. Grant access

The requester needs an access token to access the resource. To get a token, the requester sends a request to the RO specifying the resource, action he/she needs to perform and the address to which he/she wants to receive a token. Suppose, the RO has already defined her access control policy and is updated m times. The policy is stored in the P-MAM channel. Then, the *Resource Owner-Policy Enforcement Point (RO-PEP)* sends the access request to the PDP as illustrated in Fig. 5.

Given a request PDP can access the Tangle and can decrypt P-MAM channel message stream using the secret authorization key. The PDP also needs information about subject attributes to make a decision. To do so, the PDP requests attributes of the subject from the *Policy Information Point (PIP)*. The PDP has been assigned multiple PIP, the PDP sends the request of subject attributes to the domain to which a subject belongs to. The PIP gets attributes of the subject and forwards them to the PDP.

The PDP then evaluates the policies given the subject attributes. The PDP then sends the grant or the deny decision to the

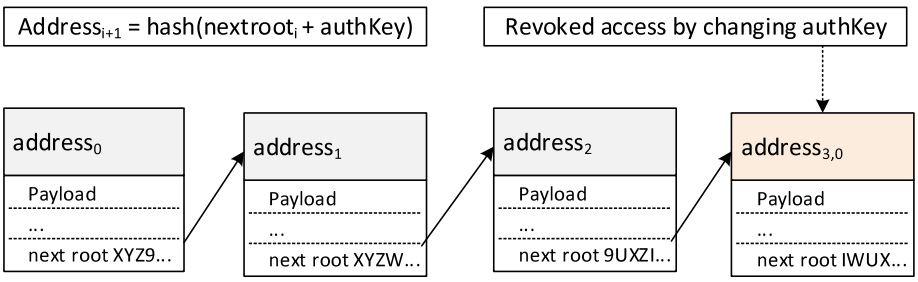

Fig. 4. Policy revocation by changing authorization key.

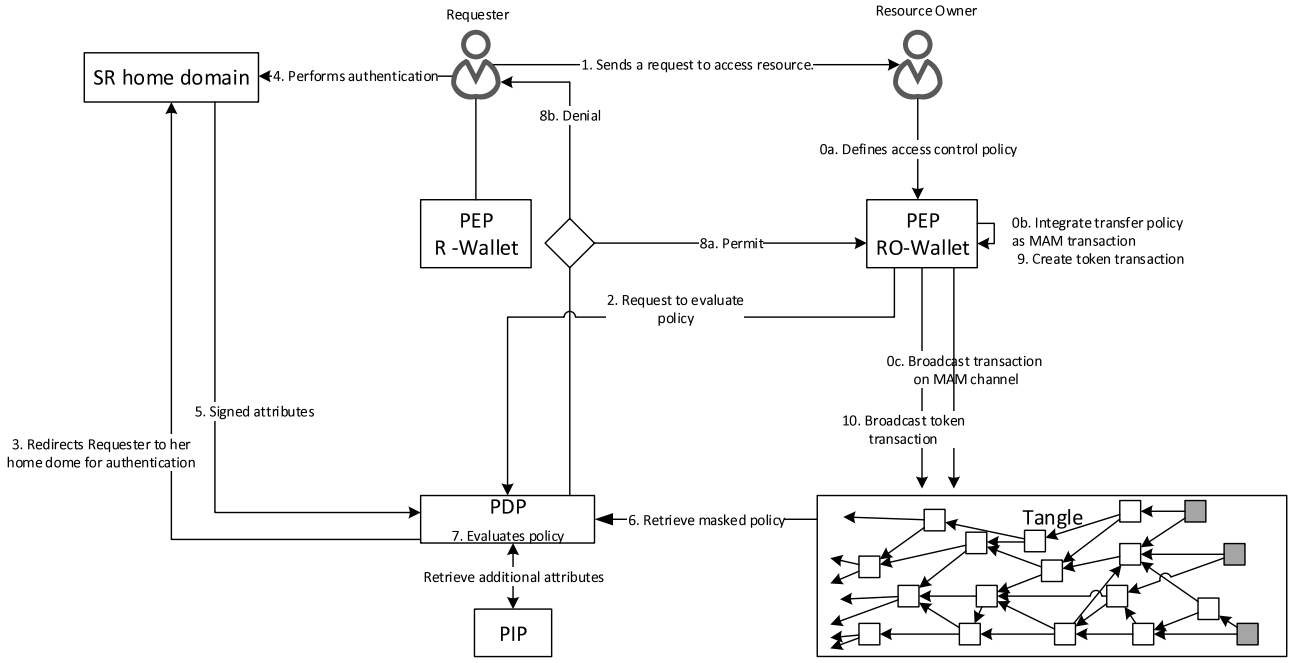

Fig. 5. GrantAccess transaction process.

RO-PEP. Upon receiving the authorization result, the RO formulates the token transaction; however, the RO wants to oversight over the token for subsequent use. In other words, RO does not want the requester to use or delegate the access token without her permission. To do so, RO sends the access token to the requester using a *multisignature* transaction. Thus, the requester cannot redeem or delegate access token in the future without the digital signature of the RO.

4.3.2. Get access

At this stage, the requester redeems the access token and access a service hosted by the RO. As a matter of fact, GetAccess transaction input is an Unspent Transaction Output UTXO of the previous GrantAccess transaction. Since the RO sent the token to a multisignature address in the GrantAccess transaction. Thus, the token redeem transaction requires the signature of both requester and the RO. The requester can spend this token herself or she can delegate access token to someone else. It is important to note that the RO signatures the redeem token only if she agrees to the transaction as depicted in Fig. 6.

4.3.3. Delegate access

The requester let us say Bob can grant access to another requester let us say Charlie via the *DelegateAccess* transaction. This simply involves taking the unspent token and transfer it to the Charlie address.

However, Bob cannot delegate access to any other requester without the permission of the RO. The RO expresses her consent by the digital signature as part of the multisignature transaction. After the signature from Bob and the RO, PEP broadcasts DelegateAccess transaction. If the transaction is valid, the network validates it by either referencing it directly or indirectly.

4.3.4. Revoke access

The RO sent the access token to the requester using GrantAccess multisignature transaction. To keep greater control over the sent access token, the RO sends the 3 of 2 multisignature transaction.

The RO sends the GrantAccess multisignature transaction token to 2 addresses owned by RO and 1 address owned by the requester. The reason behind it is that RO can revoke access of requester at any time by simply creating a new transaction that transfers the access token to solely RO address. The RO can revoke access token from the requester without her consent as depicted in Fig. 7.

Giving greater authority to the RO by sending the token to 2 addresses owned by the RO is crucial. Consider a scenario, suppose a dispute occurs between RO and requester Eve, who has an access token. The RO wants to revoke access token of Eve. The RO cannot revoke access token if the RO has sent the token using a single or 2 of 2 signature transaction. Yet, Eve cannot spend or delegate the access token without the digital signature of the RO in case of 2 of 2 signature transaction. However, the RO cannot

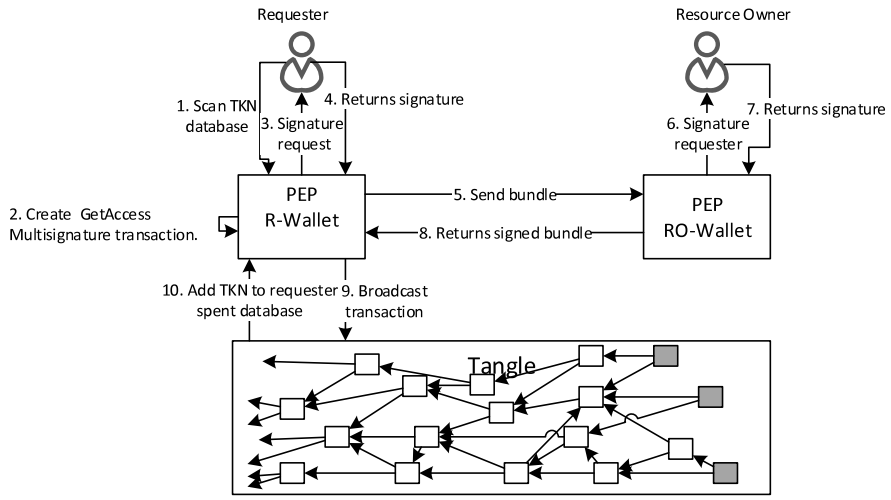

Fig. 6. GetAccess transaction process.

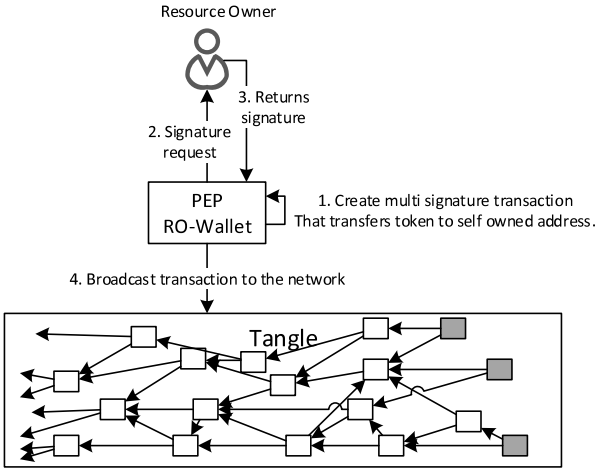

Fig. 7. RevokeAccess transaction process.

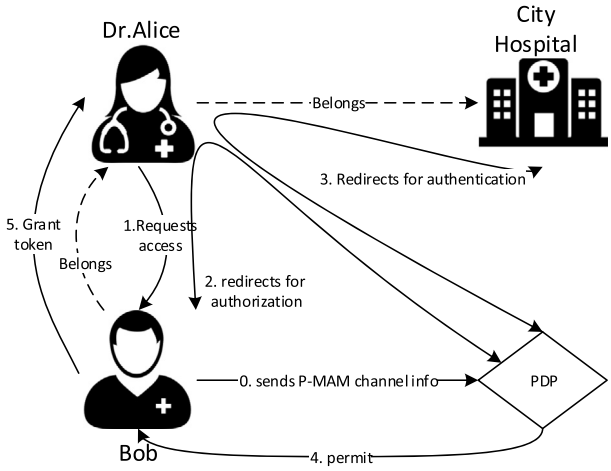

Fig. 8. Flow diagram of the case study.

revoke the access token from Eve without her digital signature which may deny. Thus, in case of token sent to 2 of 3 signature transaction, the RO can revoke Eve access by transferring token to solely herself address.

5. Implementation and evaluation

5.1. Case study

Dr. Alice is a heart specialist, and she belongs to 'City Hospital' as depicted in Fig. 8. The patient 'Bob' of Dr. Alice shows some heart attack symptoms. Dr. Alice needs to monitor the Bob heart rate. The patient data is highly sensitive and Bob does not want access to his Electronic Health Record (EHR) without his permission. Thus to restrict access, Bob deploys decentralized access control framework and makes access token mandatory to access his EHR.

Bob defines an access control policy as follows

A cardiologist can read patient data if they are patient's assigned cardiologist.

Bob publishes the policy to the Tangle using P-MAM protocol. When Dr. Alice requests to access Bob EHR. Bob PEP redirects Dr. Alice (e.g. a browser) to the PDP. After Dr. Alice is redirected to the PDP, a list of domains is presented. Dr. Alice selects her home domain (e.g. City Hospital) from the list and is redirected to the home domain interface for authentication. Dr. Alice authenticates herself using authentication credentials (e.g. user name and password) after successful authentication, Dr. Alice is redirected back to PDP along with her security attributes. The PDP then verifies and evaluates attributes against RO policies. The PDP then redirects the authorization result to Bob, who takes a decision based on the results.

One of the advantages of this scheme is that the privacy of SR is protected. The SR attributes cannot be accessed without her home domain authentication credentials. Thus, the PDP cannot access the SR attributes, as the PDP does have access to the SR home domain credentials. For authentication, the PDP redirects SR to her home domain. Upon successful authentication, the SR is redirected back to the PDP along with the authentication results such as attributes. The redirection is signed by the shared secret key between the SR's home domain and PDP such that PDP can verify the integrity and authenticity of the authentication results. When PDP receives authentication results, it evaluates the access request based on the SR attributes and RO's policies.

5.2. Experimental setup

The experiments were done on Windows 10 with Intel Core *i*5 and 4 Gb RAM. We used official IOTA Core javascript library *iota.lib.js* for the transfer of access rights. We used MAM javascript library *mam.lib.js* for the publication of the policy.

Algorithm 1 Publish Policy Algorithm Javascript

```
1: procedure PublishPolicy
2:     Input: policy
3:     Output:
4:     state ← Mam.init(iotaObject, seed, security)
5:     state ← Mam.changeMode(state, restricted, authorizationKey)
6:     message ← Mam.create(state, policy)
7:     Mam.attach(message.payload, message.address‘)
8: end procedure
```

Algorithm 2 Grant Access Algorithm

```
1: procedure GrantAccess
2:     Input: resource, action, sr, domain
3:     Output: token
4:     if SR belongs to domain 'X' then
5:         policy ← Mam.fetch(P-MAM channel)
6:         if (SR, action ∈ policy) then
7:             m ← (input(rs), output(sr_a, ro_a, Token))
8:             tx ← Sign_ro(m)
9:             Tangle.attach(tx)
10:        end if
11:    end if
12: end procedure
```

```
namespace com.axiomatics.hl7{
    import Attributes.*
    import com.axiomatics.hl7.action.*
    import com.axiomatics.hl7.object.*

    policyset global{
        apply firstApplicable
        medicalRecords
    }

    policyset medicalRecords{
        target clause resourceType == "medical record"
        apply firstApplicable
        cardiologistAccess
    }

    policyset cardiologistAccess{
        target clause Attributes.userRole == "cardiologist"
        apply firstApplicable

        policy primaryCardiologist{
            apply firstApplicable

            rule read{
                target clause action == "read"
                condition subjectId == assignedDoctorId
                permit
            }
```

Fig. 9. Policy written in ALFA.

We used the *Abbreviated Language For Authorization (ALFA)* plugin for *Eclipse*, which is a domain specific high-level description of XACML. We are using ALFA to implement *HL7* Healthcare policies.

5.3. Implementation

As the resource (EHR) is assigned to the RO (Bob). To restrict access to his medical record, Bob defines the access control policy, according to Algorithm 1. Bob wallet creates MAM channel $P-MAM$ for publishing policies. To do so, Bob generates *seed s*, a seed is a master secret key and chooses security level (e.g. 1, 2, or 3). From the seed, Bob can deterministically generate addresses and signatures. The only individual having seed *s* can deterministically generate addresses and signature transactions broadcasted on $P-MAM$ channel.

Bob wallet initializes state object and binds the *iota.lib.js* library (Line: 4). Then changes the channel mode from default “public” to “restricted” mode by defining authorization key (Line: 5). The algorithm then creates a message payload from a state object and returns updated payload and address (Line: 6). Bob wallet then attaches a payload to the Tangle asynchronously (Line: 7).

Grant access for the resource is shown in Algorithm 2. First, it checks whether Alice belongs to City Hospital domain (Line: 4). To do so, the PDP redirects Alice to the City Hospital domain for authentication. Upon successful authentication, Alice is redirected back to the PDP along with her attributes. The PDP then verifies the attributes if Dr. Alice belongs to the City Hospital. Next, the PDP extracts the policy defined by Bob from the Tangle (Line: 5). Then, the PDP evaluates Dr. Alice attributes against the policy defined by Bob (Line: 6). If Dr. Alice request meets the rules defined by Bob, Bob wallet formulates a multisignature transaction that sends a token to Alice (Line: 7). Bob wallet then signatures transaction (Line: 8) and broadcast it to the network (Line: 9).

5.4. Results

Based on the source code, the software and the hardware we carried out experiments to prove the feasibility of the proposed access control framework. The healthcare policy is written in the ALFA language as depicted in Fig. 9. Then we published generated XACML policy in the MAM channel restricted mode. The PDP fetches the policy from the Tangle and decrypts it using secret authorization key as shown in Fig. 10.

We characterize the performance of the current implementation of policy MAM channel as depicted in Fig. 11. We analyzed three major steps create, attach and fetch of the P-MAM. Each point represents the average of 50 trials for the payload create, attach and fetch process. It can be seen that the time it takes to attach policy payload to the Tangle is significantly greater than the time to create the payload and policy fetch . We also plotted corresponding 95% confidence interval, where it is appropriate.

The results in Fig. 12 show the mask and unmask the payload creation time. We tested the time to mask and unmask the payload for different policy sizes. Each point represents the average of 50 trials. The delay to mask the payload is greater than the delay to unmask the payload. It is also observed that the size of the policy does not significantly impact efficiency.

Figs. 13 and 14 show the time to attach and broadcast P-MAM, GrantAccess and GetAccess transactions to the Tangle. Fig. 13 shows the delay to send access token transaction. It also shows the time to redeem the token. The time varies depending on solving the computational puzzle of POW. We have conducted this experiment on 50 trials. According to the figure, the average time to attach and broadcast P-MAM transaction to the Tangle is 17 s. The time to attach P-MAM transaction to the Tangle depends on the policy size. In our framework, the average time GrantAccess transaction and GetAccess transaction take to attach to Tangle is 5 s and 5 s respectively. Ouaddah et al. [33] Maesa et al. [39] schemes GrantAccess and GetAccess transaction average confirmation time is 10 min and 12 s respectively, but this happens when miner prioritize transaction and include it in the block. It is worth noting that transaction fee and load on network affect the transaction time of Ouaddah et al. [33] and Maesa et al. [39] scheme. We have not compared P-MAM transaction confirmation time with [33] and [39] because these frameworks do no have masked policy feature. The performance results prove that our framework has better performance than Ouaddah et al. [33] and Maesa et al. [39] scheme with zero transaction fee.

```
[redacted]>node fetchAsync
<?xml version="1.0" encoding="UTF-8"?><!--This file was generated by the ALFA Plugin for Eclipse from Axiomatics AB (htt
p://www.axiomatics.com).--><!--Any modification to this file will be lost upon recompilation of the source ALFA file--><
xacml3:PolicySet PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable" PolicyS
etId="http://axiomatics.com/alfa/identifier/com.axiomatics.hl7.cardiologistAccess" Version="1.0" xmlns:xacml3="urn:oasis
:names:tc:xacml:3.0:core:schema:wd-17"><xacml3:Description/><xacml3:PolicySetDefaults><xacml3:XPathVersion>http://www.w3
.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion></xacml3:PolicySetDefaults><xacml3:Target><xacml3:AnyOf><xacml3:All
Of><xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal"><xacml3:AttributeValue DataType="http://ww
w.w3.org/2001/XMLSchema#string">cardiologist</xacml3:AttributeValue><xacml3:AttributeDesignator AttributeId="urn:oasis:n
ames:tc:xacml:1.0:subject:authentication-role" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" D
ataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/></xacml3:Match></xacml3:AllOf></xacml3:AnyOf></
xacml3:Target><xacml3:Policy PolicyId="http://axiomatics.com/alfa/identifier/com.axiomatics.hl7.cardiologistAccess.prima
ryCardiologist" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable" Version="1.0
"><xacml3:Description/><xacml3:PolicyDefaults><xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:
XPathVersion></xacml3:PolicyDefaults><xacml3:Target/><xacml3:Rule Effect="Permit" RuleId="com.axiomatics.hl7.cardiologis
tAccess.primaryCardiologist.read"><xacml3:Description/><xacml3:Target><xacml3:AnyOf><xacml3:AllOf><xacml3:Match MatchId=
"urn:oasis:names:tc:xacml:1.0:function:string-equal"><xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#s
tring">read</xacml3:AttributeValue><xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action" DataType="http://www.w3.org/2001/XMLSchema#string" Mus
tBePresent="false"/></xacml3:Match></xacml3:AllOf></xacml3:AnyOf></xacml3:Target><xacml3:Condition><xacml3:Apply Functio
nId="urn:oasis:names:tc:xacml:3.0:function:any-of-any"><xacml3:Function FunctionId="urn:oasis:names:tc:xacml:1.0:functio
n:string-equal"/><xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" Category="urn
:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePres
ent="false"/><xacml3:AttributeDesignator AttributeId="assignedDoctorId" Category="urn:oasis:names:tc:xacml:3.0:attribute
-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/></xacml3:Apply></xacml3:Co
ndition></xacml3:Rule></xacml3:Policy></xacml3:PolicySet>
...Policy Ends
Fetching Policy...
```

Fig. 10. XACML policy fetch results.

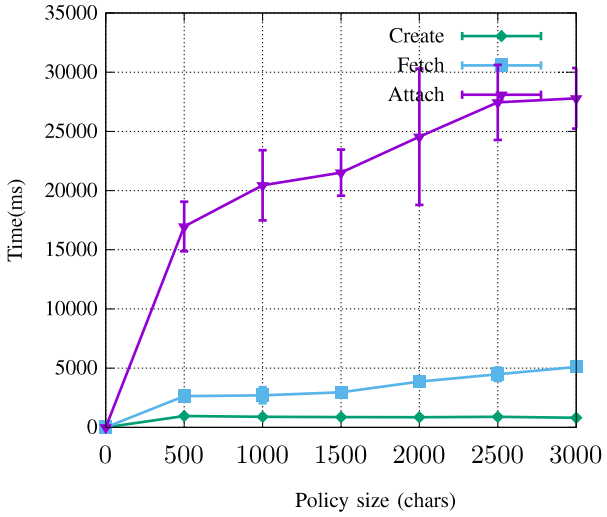

Fig. 11. Policy MAM transaction create, attach and fetch time.

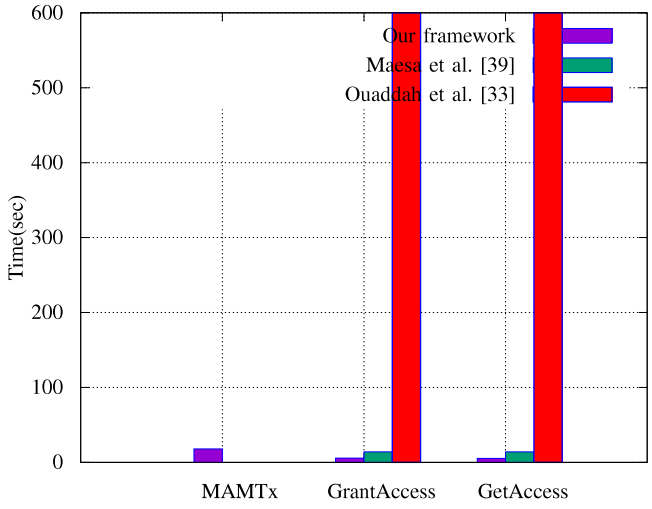

Fig. 13. P-MAM transaction, GrantAccess transaction and GetAccess transaction time.

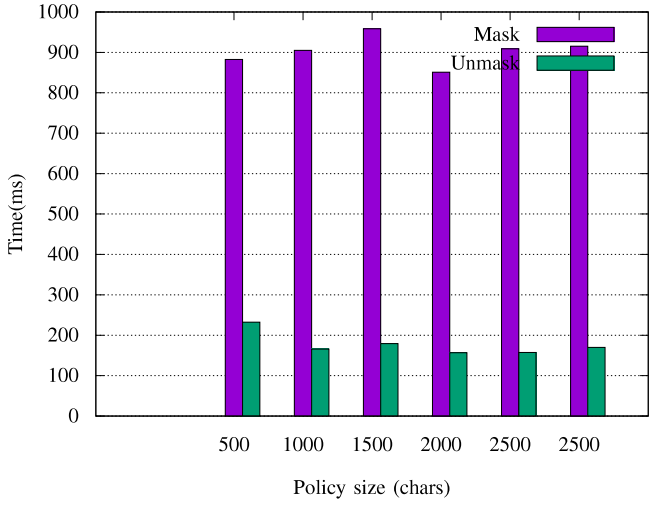

Fig. 12. Mask and unmask payload creation time.

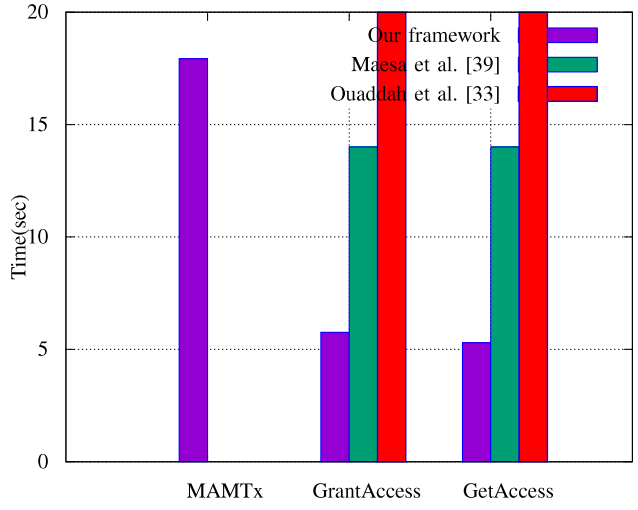

Fig. 14. Represents zoom on the results of the experiments illustrated in Fig. 13.

1) $RO \rightarrow PDP : \{Nb\}\_kbp$
2) $PDP \rightarrow RO : \{Np\}\_kbp$
3) $RO \rightarrow Tangle : \{Policy\}\_Ekbp$
4) $Tangle \rightarrow PDP : \{Policy\}\_Ekbp$
5) $SR \rightarrow RO : \{AccessRequest\}$
6) $RO \rightarrow PDP : \{AccessRequest\}\_kbp$
7) $PDP \rightarrow RO : \{Permit\}$
8) $RO \rightarrow Tangle : \{Token\}$
9) $Tangle \rightarrow SR : \{Token\}$

Fig. 15. Decentralized access control in Alice–Bob notation.

1) $RO \rightarrow PDP : \{Nb\}\_kbp$
2) $PDP \rightarrow RO : \{Np\}\_kbp$
3) $RO \rightarrow PDP : \{Policy\}\_Ekbp$
4) $SR \rightarrow RO : \{AccessRequest\}$
5) $RO \rightarrow PDP : \{AccessRequest\}\_kbp$
6) $PDP \rightarrow RO : \{Permit\}$
7) $RO \rightarrow SR : \{Token\}$

Fig. 16. Decentralized access control in Alice–Bob notation, short version.

6. Formal security and privacy analysis of the proposed scheme

This section discusses the security and privacy properties of the proposed decentralized access control system.

6.1. Formal security analysis using theorem-proving

In this section, we give an analysis of the claimed properties of the access control scheme. This section substantiates the security and privacy properties (i.e. privacy of the policy, integrity, accountability, and non-repudiation of the exchanged rights).

Theorem 1. *The proposed decentralized access control system protects the privacy of the policy.*

Proof. The Tangle MAM channel symmetric key encryption is secure if for every polynomial time algorithm $\mathcal{A}$ and security parameter κ,

$Pr[\mathcal{A}(E_k(m_0)) = 1] - Pr[\mathcal{A}(E_k(m_1)) = 1] \leq neg(\kappa)$

The RO publishes the policy on the Tangle using MAM in restricted mode. Thus, the policy published on the MAM restricted mode is semantically secure and only user having secret Merkle root and authorization key can decrypt MAM transaction message. □

Theorem 2. *The proposed decentralized access control system provides forward secrecy for policies published on the Tangle.*

Proof. To enable forward secrecy, the $P - MAM$ channel uses an encryption key Ek_i for a single message M_i which reduces the damage caused by the compromise of one encryption key Ek_i.

As described earlier, to make it easy for PDP_i to follow future policy stream, each P-MAM transaction contains the next Merkle root $Merkleroot_{i+1}$; there is no a way for the PDP_i to read MAM stream prior to the Merkle root $Merkleroot_i$ it has been assigned. To revoke future access from PDP_i, the RO can change the authorization key. □

Theorem 3. *The proposed decentralized access control system is secure and permits access only to the authorized users.*

$role\ ro(B, P, A : agent, Policy : text,$
$Kbp, EKbp : symmetric\_key, SND, RCV : channel(dy))$
$played\_by\ B$
$def =$
$local$
$State : nat,$
$ResourceRequest, Permit, Token, Nb, Np : text,$
$Address : public\_key, H : hash\_func$
$init$
$State := 0$
$transition$
$1.State = 0 \wedge RCV(start) = |>$
$State' := 1 \wedge Nb' := new()$
$\wedge SND(\{B.P.Nb'\}\_Kbp)$
$2.State = 1 \wedge RCV(\{P.B.Nb'.Np'\}\_Kbp) = |>$
$State' := 2 \wedge SND(H(B.P.Policy).\{B.P.Policy\}\_EKbp)$
$\wedge witness(B, P, bob\_pdp\_Np, Np') \wedge$
$secret(Policy, sec_1, \{B, P\})$
$3.State = 2 \wedge RCV(A.B.ResourceRequest) = |>$
$State' := 3 \wedge SND(\{A.B.P.ResourceRequest\}_K bp)$
$\wedge witness(B, P, bob_p dp_N p2, Np)$
$4.State = 3 \wedge RCV(\{P.B.Permit\}\_Kbp) = |>$
$State' := 4 \wedge request(B, P, pdp\_bob\_Nb, Nb)$
$\wedge SND(H(B.A.Token.Address).B.A.Token.Address)$
$\wedge witness(B, A, alice\_bob\_address, Address)$
$end\ role$

Fig. 17. Ro role in HLPSL. (It is important to note that the name of parameters have been changed conforming to the syntax of AVISPA tool).

Proof. A user can access data only if she has an access token, which is defined as follows

$\forall s \in S,\ \forall o \in O,\ \forall a \in A$
$GetAccess(s, o, a) \Leftarrow TKN$

where S is a set of subject attributes, O is a set of object attributes and A is a set of all actions that a subject can perform on an object. The access tokens are granted by the RO, if the PDP evaluates to true. The PDP makes a decision by loading XACML policy from the P-MAM channel. Next, the PDP decrypts the P-MAM channel transaction using secret decryption key Ek_i and evaluates the request against the P-MAM channel policy.

$policy \leftarrow D_{Ek_i}(E_{Ek_i}(policy))$

where Ek_i is a symmetric encryption key that consists of Merkle root and authorization key. If the subject does have the requested privileges then the PDP returns true.

$((s, o, a) \in policy) \Rightarrow (PERMIT \leftarrow GrantAccess(s, o, a))$

The RO then transfers access token to the service requester.

$(RO \rightsquigarrow SR : TKN) \Leftarrow PERMIT$

where $\rightsquigarrow$ represents a transfer of the token transaction. □

Theorem 4. *The proposed decentralized access control system ensures integrity, non-repudiation and the requester cannot fraudulently deny granted access rights.*

Proof. If the PDP returns a permit, the RO transfers access token to the requester. The access token is transferred via the IOTA transaction. IOTA broadcasts transaction through the network using gossip protocol. Once the IOTA network confirms the transaction. It becomes non-repudiation evidence that RO has permitted a given access.

The SR cannot forge the access token because the transaction is signed by the RO using a quantum secure Merkle signature scheme to preserve integrity and authenticity.

$\sigma \leftarrow Sign(Tx(RO \rightsquigarrow SR : TKN))$

```
role pdp(B, P, A : agent, Kbp, EKbp : symmetric_key,
SND, RCV : channel(dy))
played_by P
def =
local
State : nat, Policy, ResourceRequest,
Permit, Nb, Np : text, H : hash_func
init
State := 0
transition
1.State = 0 ∧ RCV({B.P.Nb'}_Kbp) = | >
State' := 1 ∧ Np' := new()
∧SND({P.B.Nb'.Np'}_Kbp)
2.State = 1 ∧ RCV(H(B.P.Policy').B.P.Policy'_EKbp) = | >
State' := 2 ∧ request(P, B, bob_pdp_Np, Np)
3.State = 2 ∧ RCV({A.B.P.ResourceRequest}_Kbp) = | >
State' := 3 ∧ request(P, B, bob_pdp_Np2, Np)
∧SND({P.B.Permit}_Kbp) ∧ witness(P, B, pdp_bob_Nb, Nb)
end role
```

Fig. 18. PDP role in HLPSL.

where *Sign* is Merkle signature algorithm that returns post quantum signature σ. □

Theorem 5. *By leveraging decentralized ledger our proposed framework provides an immutable log of meaningful events such as access rights grant, revocation, and transfer.*

Proof. The access rights are transferred by making use of the IOTA transactions. Once the transaction becomes part of the Tangle, it cannot be changed. An attacker needs 34% of the total hashing power to tamper the access logs. Due to immutable nature of the Tangle, the proposed decentralized access control system provides trustworthy logs to analyze who performed actions. □

6.2. Formal security analysis using AVISPA

The Automated Validation of Internet Security Protocols (AVISPA) [49] is a tool which verifies the security of the protocol against attacks. The AVISPA uses the High-Level Specification Language (HLPSL) for modeling security protocols.

In the proposed decentralized access control scheme, there are four basic roles RO, Tangle, SR and PDP as shown in Fig. 15. The Tangle is used to publish policies and access rights. For simplicity, we have considered three roles *RO*, *SR* and *PDP* in the AVISPA, as shown in Fig. 16. The protocol described in AVISPA sends directly to the corresponding entity (i.e. SR, PDP), as a substitute of broadcasting it to the network. It is worth noting that the simplified protocol described in AVISPA still captures the original protocol and does not affect the security.

As described earlier, there are three roles the RO, the PDP, and the SR, which represents agents A, P and B. We represent the HLPSL coding of the RO role in Fig. 17, SR in Fig. 19 and PDP in Fig. 18. After receiving the start signal, the RO_i changes its state to 1 from 0. A random number *Nb* is also generated by the RO_i. Next, the RO_i sends *SND*()_*SK* to the PDP by utilizing symmetric key SK and SND operation. The PDP_i changes its state to 1 from 0 and replies via a secure channel with *Np* stored in it. RO_i, then changes its state to 2 from 1. The policy publication phase is then stated by RO_i by sending restricted $P - MAM$ transaction via a secure channel to the PDP_i. The P-MAM transaction contains the encrypted policy. Upon receiving policy transaction, PDP_i authenticates it and changes state from 1 to 2.

The SR_i starts *GrantAccess* phase and sends a request to access the resource by means of an open channel. Upon receiving the request, the RO_i changes its states to 3 from 2 and redirects

```
role sr(B, P, A : agent, SND, RCV : channel(dy))
played_by A
def =
local
State : nat, Policy : text, ResourceRequest,
Token : text, Address : public_key, H : hash_func
init
State := 0
transition
1.State = 0 ∧ RCV(start) = | >
State' := 1 ∧ SND(A.B.ResourceRequest)
2.State = 1∧
RCV(H(B.A.Token.Address).B.A.Token.Address) = | >
State' := 2 ∧ request(A, B, alice_bob_address, Address)
end role
```

Fig. 19. SR role in HLPSL.

```
role session(B, P, A : agent, Policy : text,
Kbp, EKbp : symmetric_key)
def =
local
SND3, RCV3, SND2, RCV2, SND1, RCV1 : channel(dy)
composition
sr(B, P, A, SND3, RCV3)
∧ro(B, P, A, Policy, Kbp, EKbp, SND1, RCV1)
∧pdp(B, P, A, Kbp, EKbp, SND2, RCV2)
end role
role environment()
def =
const
kbp, ekbp, kbp2, ekbp2, kip, ekip : symmetric_key,
bob, pdp1, alice : agent,
policy1, policy2, policy3 : text,
sec_1, bob_pdp_Np, bob_pdp_Np2, pdp_bob_Nb,
alice_bob_address : protocol_id
intruder_knowledge = bob, pdp1, alice, kip, ekip
composition
session(bob, pdp1, alice, policy1, kbp, ekbp)
∧session(bob, pdp, i, policy2, kbp2, ekbp2)
∧session(i, pdp, alice, policy3, kip, ekip)
end role
goal
secrecy_of sec_1
authentication_on bob_pdp_Np
authentication_on pdp_bob_Nb
authentication_on bob_pdp_Np2
authentication_on alice_bob_address
end goal
environment()
```

Fig. 20. Roles for session, goal and environment in HLPSL.

```
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL

PROTOCOL
/home/span/span/testsuite/results/rosrpdp8.if

GOAL
As Specified

BACKEND
CL-AtSe

STATISTICS

Analysed : 19 states
Reachable : 7 states
Translation: 0.01 seconds
Computation: 0.00 seconds
```

Fig. 21. CL-AtSe result.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/rosrpdp8.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.03s
visitedNodes: 15 nodes
depth: 9 plies
```

Fig. 22. OFMC result.

access request to the PDP_i via a secure channel. The PDP_i alters its state from 2 to 3 after receiving authorization request. The PDP_i evaluates access request against the RO_i policies. The PDP_i then forwards a permit decision to the RO_i. Once the RO_i receives the message, the RO_i performs an authentication and sends the token transaction to the SR_i via the public channel. It is worth noting that policies and tokens are sent and received via the Tangle.

After mutual verification, a session is set up between roles. Then, the composed roles are defined in the session as depicted in Fig. 20. Next, the environment role is defined which consists of a composition of sessions. After execution of the protocol, using AVISPA, one secrecy and four authentication goals are validated. The *secrecy_of_sec*1 signifies that the policy is a secret between the RO and the PDP. It is important to notice that in the longer version the RO first sends policy transaction to the Tangle and then the Tangle forwards it to the PDP. However, it do not affect the policy secrecy because policy is encrypted using symmetric key that is kept private to RO and PDP.

The proposed decentralized access control scheme is experimented by using On-the-Fly Model-Checker (OFMC) and CL-based Attack Searcher (CL) backends and Figs. 21 and 22 shows results obtained after simulation. The intruder has full control over the network in a manner that it can intercept, modify and analyze messages if it knows the corresponding key. The intruder can impersonate as any agent and sends messages to any other agent. The simulation results confirm that the proposed decentralized access control scheme is secure. Thus, the proposed decentralized access control scheme can be deployed safely.

7. Conclusion and future work

This paper leverages the scalability offered by the Tangle to create, manage and enforce access control policies. The main advantage of this scheme is that policies are published in the encrypted format on the Tangle and only user having secret decryption key can read it. Thus protects the privacy of the access control policies which is a long awaited feature that was missing in the literature. The access rights are transferred publicly through transactions. Previously proposed decentralized access control schemes have high transaction fees and delays. The proposed approach is validated by an implementation based on the IOTA and the results show that our scheme has zero transaction fee, low resource requirements and low delays. The simulation of the protocol in the AVISPA proved its resistance against several attacks.

Following are the main future work research directions

- We plan to extend our work to study how to better embed our decentralized access control framework in the IoT. In particular, we are exploring the possibility of utilizing smart contracts to achieve automatic evaluation and enforcement of the policies. IOTA does not natively support smart contracts, but smart contracts would be provided as an additional layer by IOTA Foundation.
- Another interesting research direction is to leverage machine learning algorithms that are already existed, in order to provide a optimized, dynamic and self-adjusted security policy.
- Moreover, we are investigating the implications of using permissioned Tangle in our proposed access control framework.
- Finally, we want to deploy the proposed model in different IoT use case scenarios to achieve distributed and trustworthy access control.

Declaration of competing interest

The authors declared that they had no conflicts of interest with respect to their authorship or the publication of this article.

CRediT authorship contribution statement

Sehrish Shafeeq: Methodology, Formal analysis, Conceptualization, Data curation, Formal analysis, Investigation, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Masoom Alam:** Project administration, Supervision. **Abid Khan:** Supervision.

References

[1] J. Rivera, R. van der Meulen, Gartner says the internet of things installed base will grow to 26 billion units by 2020, in: Stamford, conn., December, vol 12, 2013.
[2] J. Rivera, R. van der Meulen, Gartner says the internet of things will transform the data center, in: Retrieved August, vol. 5, 2014, p. 2014.
[3] N. Lin, W. Shi, The research on internet of things application architecture based on web, in: Advanced Research and Technology in Industry Applications (WARTIA), 2014 IEEE Workshop on, IEEE, 2014, pp. 184–187.
[4] F. Al-Turjman, E. Ever, H. Zahmatkesh, Small cells in the forthcoming 5g/IoT: Traffic modeling and deployment overview, in: Smart Things and Femtocells, CRC Press, 2018, pp. 17–82.
[5] F. Al-Turjman, S. Alturjman, 5G/IoT-enabled UAVs for multimedia delivery in industry-oriented applications, Multimedia Tools Appl. (2018) 1–22.

[6] F. Al-Turjman, QoS—aware data delivery framework for safety-inspired multimedia in integrated vehicular-IoT, Comput. Commun. 121 (2018) 33–43.
[7] D.M. Mendez, I. Papapanagiotou, B. Yang, Internet of things: Survey on security and privacy, 2017, arXiv preprint arXiv:1707.01879.
[8] F. Al-Turjman, S. Alturjman, Context-sensitive access in industrial internet of things (IIoT) healthcare applications, IEEE Trans. Ind. Inf. 14 (6) (2018) 2736–2744.
[9] F. Al-Turjman, S. Alturjman, Confidential smart-sensing framework in the IoT era, J. Supercomput. 74 (10) (2018) 5187–5198.
[10] F. Al-Turjman, 5G-enabled devices and smart-spaces in social-IoT: an overview, Future Gener. Comput. Syst. 92 (2019) 732–744.
[11] S.A. Alabady, F. Al-Turjman, S. Din, A novel security model for cooperative virtual networks in the IoT era, Int. J. Parallel Program. (2018) 1–16.
[12] S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Comput. Netw. 76 (2015) 146–164.
[13] J. Singh, T. Pasquier, J. Bacon, H. Ko, D. Eyers, Twenty security considerations for cloud-supported internet of things, IEEE Internet Things J. 3 (3) (2016) 269–284.
[14] A. Ouaddah, H. Mousannif, A.A. Elkalam, A.A. Ouahman, Access control in the internet of things: Big challenges and new opportunities, Comput. Netw. 112 (2017) 237–262.
[15] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, Role-based access control models, Computer 29 (2) (1996) 38–47.
[16] N. Boustia, A. Mokhtari, Representation and reasoning on orbac: Description logic with defaults and exceptions approach, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 1008–1012.
[17] L. Wang, Y. Zhu, L. Jin, X. Luo, Trust mechanism in distributed access control model of P2P networks, in: Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on, IEEE, 2008, pp. 19–24.
[18] R.S. Sandhu, P. Samarati, Access control: principle and practice, IEEE Commun. Mag. 32 (9) (1994) 40–48.
[19] S. Gusmeroli, S. Piccione, D. Rotondi, Iot access control issues: a capability based approach, in: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, IEEE, 2012, pp. 787–792.
[20] V.C. Hu, D.R. Kuhn, D.F. Ferraiolo, Attribute-based access control, Computer 48 (2) (2015) 85–88.
[21] V.C. Hu, D. Ferraiolo, R. Kuhn, A.R. Friedman, A.J. Lang, M.M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., Guide to attribute based access control (ABAC) definition and considerations (draft), NIST Spec. Publ. 800 (162) (2013).
[22] A. Yavari, A.S. Panah, D. Georgakopoulos, P.P. Jayaraman, R. van Schyndel, Scalable role-based data disclosure control for the internet of things, in: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, IEEE, 2017, pp. 2226–2233.
[23] Q. Liu, H. Zhang, J. Wan, X. Chen, An access control model for resource sharing based on the role-based access control intended for multi-domain manufacturing internet of things, IEEE Access 5 (2017) 7001–7011.
[24] A. Sharma, D. Srinivasan, D.S. Kumar, A comparative analysis of centralized and decentralized multi-agent architecture for service restoration, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 311–318.
[25] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. (Accessed on 20 December 2018).
[26] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, in: Ethereum Project Yellow Paper, vol. 151, 2014.
[27] S. Underwood, Blockchain beyond bitcoin, Commun. ACM 59 (11) (2016) 15–17.
[28] K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E.G. Sirer, et al., On scaling decentralized blockchains, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 106–125.
[29] K. Seol, Y.-G. Kim, E. Lee, Y.-D. Seo, D.-K. Baik, Privacy-preserving attribute-based access control model for XML-based electronic health record system, IEEE Access 6 (2018) 9114–9128.
[30] S. Popov, O. Saa, P. Finardi, Equilibria in the tangle, 2017, arXiv preprint arXiv:1712.05385.
[31] S. Popov, The tangle, in: IOTA, 2016. [Online]. Available: https://iota.org/IOTAWhitepaper.pdf. (Accessed 20 December 2018).
[32] A. Ouaddah, A.A. Elkalam, A.A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in iot, in: Europe and MENA Cooperation Advances in Information and Communication Technologies, Springer, 2017, pp. 523–533.
[33] A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman, Fairaccess: a new blockchain-based access control framework for the internet of things, Secur. Commun. Netw. 9 (18) (2016) 5943–5964.
[34] D.D.F. Maesa, P. Mori, L. Ricci, Blockchain based access control, in: IFIP International Conference on Distributed Applications and Interoperable Systems, Springer, 2017, pp. 206–220.
[35] A. Outchakoucht, E.-S. Hamza, J.P. Leory, Dynamic access control policy based on blockchain and machine learning for the internet of things, Int. J. Adv. Comput. Sci. Appl. 8 (7) (2017) 417–424.
[36] L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: A survey, J. Artif. Intell. Res. 4 (1996) 237–285.
[37] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things, 2018, arXiv preprint arXiv:1802.04410.
[38] C. Dukkipati, Y. Zhang, L.C. Cheng, Decentralized, blockchain based access control framework for the heterogeneous internet of things, in: Proceedings of the Third ACM Workshop on Attribute-Based Access Control, ACM, 2018, pp. 61–69.
[39] D.D.F. Maesa, P. Mori, L. Ricci, A blockchain based approach for the definition of auditable access control systems, Comput. Secur. (2019).
[40] A. Corp, Leveraging today's megatrends to drive the future of identity management, in: Video Presentation, Gartner Identity and Access Management (IAM) Summit, 2012, [Online]. Available: www.avatier.com/products/identity-management/resources/gartner-iam-2020-predictions. (Accessed 20 December 2018).
[41] A. Back, et al., Hashcash-a denial of service counter-measure, 2002, [Online]. Available: ftp://sunsite.icm.edu.pl/site/replay.old/programs/hashcash/hashcash.pdf. (Accessed 20 December 2018).
[42] D.J. Bernstein, J. Buchmann, E. Dahmen, Post-Quantum Cryptography, Springer Science & Business Media, 2009.
[43] Very basic estimates on the risk of reusing the winternitz signatures, 2017, https://public.tangle.works/winternitz.pdf, (Accessed 20 December 2018).
[44] R.C. Merkle, A certified digital signature, in: Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 218–238.
[45] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Commun. Mag. 40 (8) (2002) 102–114.
[46] M. Li, W. Lou, K. Ren, Data security and privacy in wireless body area networks, IEEE Wirel. Commun. 17 (1) (2010) 51–58.
[47] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Comput. Netw. 57 (10) (2013) 2266–2279.
[48] O. Standard, Extensible access control markup language (xacml) version 2.0, 2005.
[49] The AVISPA project, automated validation of internet security protocols and applications, 2018, [Online]. Available: http://www.avispa-project.org/. (Accessed 20 December 2018).

Muhammad Masoom Alam received his Ph.D. in information security from University of Innsbruck Austria, in 2007. He currently leads cyber security lab which is involved in various industrial projects in blockchain, threat intelligence and container security. He is an Associate Professor in the Department of Computer Science, Comsats University Islamabad, Pakistan. His Ph.D. thesis has got the best thesis award at the Models 2006 conference. He has a total of 17 years experience in teaching, system administration and research and development. His research lab is hosting over 35 graduates and undergraduates.