Blockchain Based Delegatable Access Control Scheme for a Collaborative E-health Environment

Harsha S. Gardiyawasam Pussewalage and Vladimir A. Oleshchuk
Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Email: {harsha.sandaruwan; vladimir.oleshchuk}@uia.no

***Abstract*—Modern electronic healthcare (e-health) settings constitute collaborative environments requiring sophisticated fine-grained access control mechanisms to cater their access demands. Access delegatability is quite crucial to realize fine-grained, flexible access control schemes compatible with such environments. In this paper, we addressed this issue through proposing an attribute based access control scheme integrated with controlled access delegation capabilities suitable for a multi-domain e-health environment. We have utilized the blockchain technology to manage attribute assignments, delegations as well as revocations. The scheme enables delegations in a controlled manner without jeopardizing the security of the system. The control is achieved via granting each delegating user the capability of controlling the subsequent delegations made by the delegatee as well as limiting the length of a chain of delegations. Furthermore, it is equipped with a superior attribute revocation mechanism and induces substantially lower key management overhead to the end-users in comparison to the existing access control schemes with delegatability.**

I. Introduction

The recent technological advancements paved the way towards making significant strides in electronic healthcare (e-health) providing a means for efficient sharing of health resources and making it a part of day to day life of general public. E-health has made it possible to store the health information of patients in a digitized format, widely known as electronic health records (EHRs) [1]. Such records provide the platform for efficient sharing of patient health information among different healthcare settings and different care deliverers enabling better service for the consumers. However, given that every EHR accumulates a significant amount of private information, it is evident that any sort of illegitimate disclosure may potentially affect the privacy of the patient [2]. Many such security and privacy breaches have been identified in real life e-health systems as documented in [3]–[5].

User's ability of delegating access is a vital factor in achieving timely and flexible sharing of EHRs of patients [6], especially considering the collaborations of different entities which may be involved in the treatment process of a patient. For instance, consider the following scenario. Suppose, Alice is a patient of hospital A and her EHR stored in hospital A is associated with an attribute access structure $\mathcal{T}$ = (*Physician* $\wedge$ *Hospital_A*) giving permission to physicians in hospital A to access Alice's EHR. After a recent treatment session, Dr. Bob who is a physician at hospital A wants to refer the recent findings to Dr. Charlie who is working as a physician at hospital B. With the presence of delegation capability, Dr. Bob is able to temporarily delegate the attribute *Hospital_A* to Dr. Charlie which allows him to access the EHR of Alice. If, Dr. Charlie wants to obtain further expertise from Dr. John who is working as a pathologist at hospital B, with delegatability, it is possible to re-delegate the attributes *Physician*, *Hospital_A* to Dr. John allowing him to access Alice's EHR via satisfying the associated $\mathcal{T}$. Incorporating such delegating capabilities induces the following challenges which need to be taken into consideration. Suppose, Dr. Bob possesses the attribute set ω_{Bob} = {*Director*, *Hospital_A*, *Physician*}. During delegation, Dr. Bob may not want to delegate all his attributes to Dr. Charlie given that delegating the attribute *Hospital_A* is sufficient to satisfy $\mathcal{T}$. Furthermore, Dr. Bob may or may not be interested in allowing Dr. Charlie to re-delegate the attributes. We call the delegation capability adhering to the aforementioned challenges as controlled access delegation [6].

A variety of access control mechanisms such as access control lists (ACL), role based access control (RBAC) and attribute based access control (ABAC) has been widely utilized for ensuring secure and restricted access for computer resources, especially in multi-user data sharing settings [7]–[9]. Among them, ABAC is more conducive for establishing fine-grained access in collaborative e-health systems since ABAC inherently supports inter-domain data access which is not the case with RBAC [10]. In ABAC, access decisions are made based on the attributes of the subjects, attributes of the objects and environmental attributes instead of organizational roles (which is the case with RBAC), allowing both registered users in the home domain care deliverer and users from the foreign domains to claim access to the shared resources stored in the home domain care deliverer. However, ABAC is not yet formally standardized while a general guideline for ABAC implementations is published by the National Institute of Standards and Technology (NIST) recently [11]. In this paper, we have addressed the above mentioned challenge via proposing a delegatable ABAC scheme using blockchain technologies which not only suitable for collaborative systems but also ensure the pseudo-anonymity of users.

The organization of the rest of the paper is as follows. Sec. II summarizes the most prominent research efforts related to access control schemes integrated with delegatability. We introduce the case description, security model and the associated security requirements in Sec. III while the Sec. IV presents the preliminaries corresponding to the proposed scheme. Sec. V is dedicated to describe the proposed access control scheme in-depth. Sec. VI presents the security analysis of the proposed scheme whereas in Sec. VII, we compare the proposed scheme with the existing delegatable access control schemes. Finally, Sec. VIII concludes the paper.

II. Related Work

Recently, attribute based encryption (ABE) schemes have been considered as a means of realizing fine-grained access control while the data being in an encrypted format. An ABE scheme embeds (attribute based) access policies into an encryption algorithm, in such a way that any user who possesses a set of secret keys relevant for a set of attributes that satisfies the associated access policy can decrypt the associated ciphertext [12], [13]. Given that decryption is completely handled at the user end, users will be able to stay anonymous while accessing the data. There exists two solutions [14], [15], in which delegation capabilities are integrated to ABE models to achieve flexible access control. Despite the fact that the scheme in [14] capable of allowing a user to delegate access to another user (via delegating attributes), there is no mechanism to control the subsequent access delegations. The scheme in [15] provides a solution to the aforementioned issue by using a trusted third party (TTP). However, given that the TTP has a complete view of who delegates which attribute to whom might affect the privacy of users. In addition, both schemes in [14], [15] do not possess a mechanism to control the maximum number of delegations permitted for a given attribute. Hence, the existing ABE based solutions lack the control over access delegation.

In contrast to the schemes proposed in [14], [15], the construction proposed in [6] is based on a multi-authority architecture and therefore it is more realistic and conducive for collaborative environments. Furthermore, it supports multi-level access delegation with each delegator can permit or deny further delegations. In addition, the scheme can also specify the number of maximum delegations permitted for a particular attribute through the attribute authority (AA) which administrates the attribute. This would also induce more control over the delegation process since users will not be able to extend the length of a delegation chain alarmingly. Given that an attribute is assigned to a user in the form of an attribute token signed by the secret key associated with the attribute, revoking that user from the attribute requires issuing newly signed tokens to all the other users who ascertained the same attribute. This makes the process of revoking a user from an attribute before the expiration of the associated attribute token rather expensive.

In this paper, we have addressed this issue through proposing a delegatable ABAC scheme conducive to a multi-domain e-health environment using blockchain technology. Blockchain [16], [17] is a data structure represented with a series of data blocks such that each data block embeds a hash pointer to the previous block. The head of the block is simply a hash pointer to the most recent data block which is stored securely. This has been the foundation for recording cryptocurrency transactions [18], [19] and we have adopted this to record attribute assignments, delegations as well as revocations.

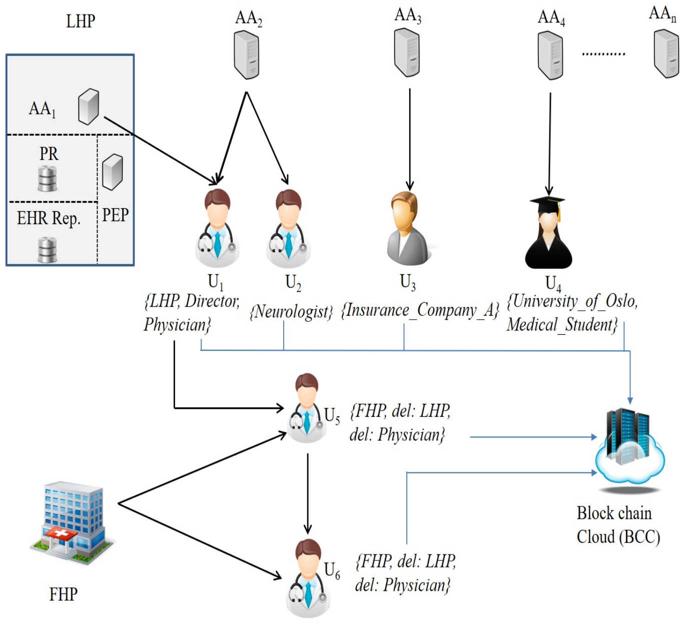


Fig. 1. System model.

III. Case description, security model and security requirements

In this section, we introduce the health information sharing scenario for which the ABAC scheme is proposed. We also present the security model as well as the security requirements which must be maintained in order to realize a secure and flexible access control scheme.

A. Case description

We consider a local healthcare provider (LHP) which is responsible for providing healthcare services for people who reside in a specific geographical area. Every patient who is registered in the LHP has an associated EHR, stored locally under the control of the LHP. We require EHRs of patients to be shared among a set of users with varying levels according to the consent of the patient. Users include both healthcare professionals associated with the LHP, users from other domains such as insurance companies, medical students as well as healthcare professionals attached to foreign healthcare providers (FHP). Fig. 1 represents the system model and its main components are defined below.

- *Local Healthcare Provider (LHP):* LHP provides healthcare services to its registered patients residing in a certain geographical area. It is composed of an EHR repository, policy repository (PR), and a policy enforcement point (PEP). PR is used to store the access policies relevant for each EHR stored in the EHR repository while the PEP operates as the decision point to make the decisions on access requests. We provide more information on access policies in Sec. IV.B.

- *Foreign Healthcare Providers (FHPs):* We denote FHPs as healthcare providers who are responsible for providing services in other geographical domains.
- *Attribute Authorities (AAs):* We call attribute issuing authorities as AAs. Each AA is responsible for managing a set of attributes and issuing relevant attributes upon validating attribute requirements. Further, we assume that all AAs are trusted. In addition, LHP also acts as an AA to issue attributes specific for the LHP.
- *Users:* Users are the subjects who are eligible to access stored EHRs. Subjects include entities who are affiliated to the LHP, members of other organizations such as insurance companies as well as entities affiliated to FHPs.
- *Delegator & Delegatee:* If a particular user is assigning an attribute (which he owns) to another user, the user who is assigning the attribute we call as the delegator. Furthermore, we call the user who is receiving the attribute as the delegatee.
- *Blockchain Cloud (BCC):* BCC maintains a set of blockchains which provide confirmations of attribute assignments, delegations as well as revocation of users from specific attributes. We have represented BCC as a single server for ease of illustration in this paper, but it can simply be extended to a set of servers to provide better availability. This would require an algorithm to achieve the consensus of stored blockchains in individual servers and we consider this outside the scope of this paper.

B. Security model

We assume that the LHP is honest while BCC is semi-trusted meaning that it will follow the protocols specified accordingly while being curious on the information being stored. It is also assumed that the users may be curious on the stored data and potentially interested in extracting more information than what they are allowed through the access privileges. For instance, a pharmacist would be interested in accessing patient prescriptions and learn prescription patterns of different doctors which could be useful for marketing purposes and boosting profits. To do so, users may use forged attributes to gain access to data which cannot be recovered individually.

C. Security requirements

The main security requirements to be satisfied in the proposed EHR sharing scheme are summarized as follows.

- *Patient privacy:* It is important to prevent any sort of unauthorized disclosure of patient's health information. Therefore, any user who does not possess enough attributes to satisfy the access requirements must be prevented from granting access to stored EHR data of patients.
- *User privacy:* The linkability of a patient EHR to the accessing user's identity could impact the privacy of the EHR accessing user when such information is being exposed to interested parties. Thus, users must be allowed to access EHR data in an anonymous manner.
- *Resistance to attribute collusion:* Users should not be able to gain access to any stored EHR by colluding attributes with other users.
- *Controlled access delegation:* The issuer of an attribute must have the control of delegation right meaning that further delegations by the user who received the attribute are only feasible with the consent of the current issuer as well as the consent of the AA which administrates the considered attribute.
- *Attribute revocation:* The process of revoking an issued attribute from a user before its expiry is termed as attribute revocation. The revocation process should be decentralized, meaning that both AAs and users should be able to carry out revocation when the occasion demands.

IV. Preliminaries

This section is dedicated to providing the necessary background details associated with the proposed scheme which enables us to present the proposed scheme in-depth in the following section.

A. Types of attribute related transactions

We define three types of attribute based transactions; assignment transactions, delegation transactions and revocation transactions as explained below.

1) Assignment transactions: If an attribute is assigned to a user by an AA, we consider such an attribute as an assigned attribute. The information that enters the blockchain representing the aforementioned attribute assignment is referred to as an assignment transaction. For instance, in Fig. 1, the attribute *Physician* of the user U_1 is an assigned attribute, since it is assigned to U_1 by AA_2.

2) Delegation transactions: We consider an attribute as a delegated attribute if the attribute is assigned to a particular user by another user who owns the attribute. If the current delegator obtained the attribute directly from an AA, we denote the current delegation as the first level delegation. If the delegator has obtained the considered attribute from a first level delegation, then the delegator's following delegations of the same attribute (if permitted) are denoted as second level delegations. For instance, with reference to Fig. 1 the delegation of attributes $\{LHP, Physician\}$ from U_1 to U_5 is a first level delegation while the delegation of same attributes from U_5 to U_6 is a second level delegation. The representations of these delegations in the blockchain are referred to as delegation transactions.

3) Revocation transactions: The information which represents a revocation of an attribute from a given user in the blockchain is defined as a revocation transaction. A revocation transaction can be initiated by both AAs and users to revoke attributes of the users who acquired attributes from them.

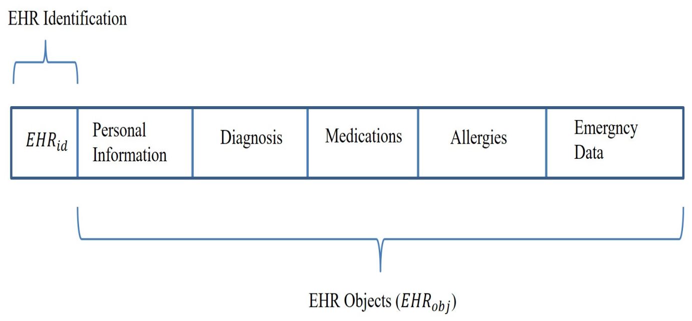


Fig. 2. Structure of an EHR

B. *Structure of EHRs & EHR access policies*

In this subsection, we describe the structure of a patient's EHR and EHR access policies which specify the access requirement for accessing EHRs of patients.

1) Structure of a patient's EHR: We assume that each registered patient at the LHP has an EHR having the unique identification EHR_{id}. The content of an EHR can be categorized as personal information, diagnosis, medications, allergies, emergency data, etc. and we define them as EHR objects (EHR_{obj}). Hence, a patient's EHR can have many different EHR objects as shown in Fig. 2.

2) EHR access policy: Each EHR_{obj} can have one or more access policies which specify the access requirement for the associated EHR_{obj} along with the permissions. Thus, an access policy $\mathcal{P}$ is in the form of,

$$\mathcal{P} : \mathcal{T} \mapsto A,$$

where $\mathcal{T}$ is a Boolean statement with logical conjunctions ($\wedge$) and logical disjunctions ($\vee$) combining required subject attributes and "$\mapsto$" has the meaning of "permitted to". A represents the permissions (read, write) that a user will obtain, after satisfying the access requirement defined by $\mathcal{T}$. For example, consider the following two access policies associated with the $EHR_{obj} = O$, corresponding to the EHR, $EHR_{id} = I$.

$$\mathcal{P}_{I,O} : (Physician \wedge LHP) \mapsto read\ \&\ write \qquad (1)$$

$$\mathcal{P}_{I,O} : (Nurse \wedge LHP) \mapsto read \qquad (2)$$

The statement (1) stipulates that a physician who is employed in LHP is authorized to carry out read and write operations on the EHR_{obj}, O whereas the statement (2) states that a nurse who works in LHP can only read the information.

V. Proposed Delegatable ABAC Scheme

In this section, the functionality of the access control scheme is described, by dividing it into four phases as system initialization, attribute distribution with insertions to the blockchain, revoking attributes from users and authorization decision making.

A. *System initialization*

Suppose there exists N AA's each managing a disjoint set of attributes and each AA has a public key infrastructure (PKI) certificate and an associated RSA public and private key pair. If the k^{th} AA is denoted with AA_k, its PKI certificate is denoted with PKI_k while (PK_k, SK_k) are used to denote the respective public and private keys. In addition, we assume that the BCC's PKI certificate is denoted with PKI_{BC} and the associated RSA key pair is denoted with (PK_{BC}, SK_{BC}). Furthermore, BCC maintains N blockchains such that the blockchain BC_k includes information about attribute related transactions associated with the attributes of AA_k.

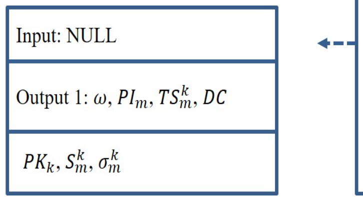
Suppose, the m^{th} user is denoted with U_m and to initialize, U_m first generates a RSA key pair (PK'_m, SK'_m) and computes his pseudo-identity PI_m such that $PI_m = SHA512(PK'_m)$. Note that a user has the capability to generate a new pseudo-identity at any time by simply generating a new RSA key pair.

B. *Attribute distribution*

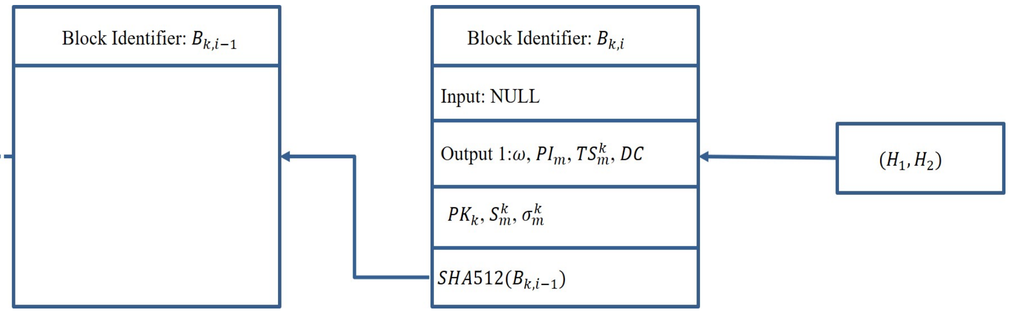
We subdivide the discussion on attribute distribution into two categories as process of assigning attributes and delegating attributes.

1) Assigning attributes: Let us assume that the user U_m wants to acquire the attribute ω from AA_k and the associated process is described below.

- U_m first mutually authenticates with AA_k using their respective RSA keys.
- Then, U_m requests for the attribute ω from AA_k, by providing evidence for the fact that he is eligible to ascertain the requested attribute.
- Given that AA_k is satisfied with the eligibility of U_m to acquire the requested attribute, AA_k constructs the assignment transaction block as shown in Fig. 3(a). The assignment block has an input which is set to NULL to represent that this transaction is an attribute assignment. In the output, AA_k specifies the assigning attribute ω, pseudo-identity of receiver (PI_m), expiration timestamp (TS^k_m) (defines the period of validity) and a delegation count index, DC where $DC \in \mathbb{Z}^{\geq 0}$. If $DC = 0$, U_m is not allowed to delegate the attribute ω and if $DC = k, k \in \mathbb{Z}^+$, the length of the delegation chain (maximum number of re-delegations permitted starting with this attribute assignment) is constrained to k. Note that, if U_m has requested r attributes from AA_k, AA_k will include a separate output vector for each assigning attribute as described above. Hence, there will be r outputs in the assignment transaction block. In addition, a random seed S^k_m is also generated and inserted to the block along with the issuer's public key PK_k.
- Then, AA_k generates the SHA512 hash of block contents $M^k_m = SHA512(Input||Outputs||PK_k||S^k_m)$ and signs M^k_m with the secret key SK_k to generate the transaction signature, $\sigma^k_m = [M^k_m]_{SK_k}$. Thereafter, AA_k includes the generated block signature in the transaction block.
- To transfer the aforementioned assignment block to the blockchain BC_k, AA_k mutually authenticates with BCC using the associated RSA keys and forwards the signed transaction block.

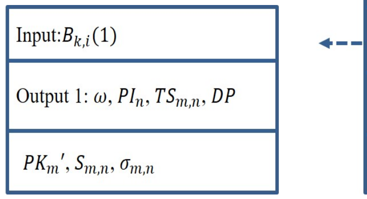

Input: NULL
Output 1: ω, PI_m, TS_m^k, DC
PK_k, S_m^k, σ_m^k

(a) Signed assignment block generated by AA_k

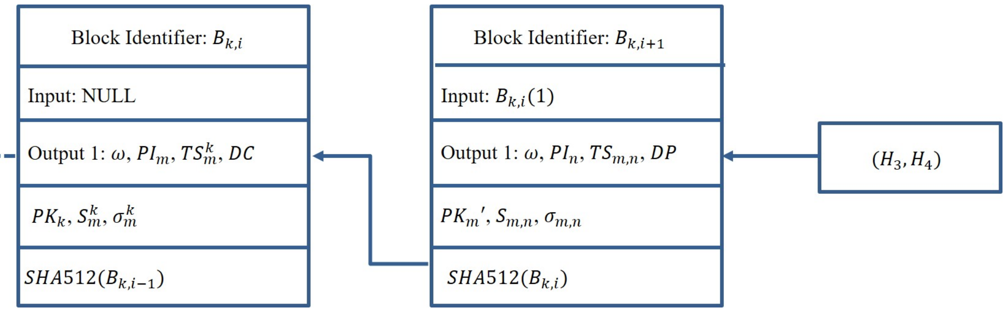

Block Identifier: $B_{k,i-1}$
Block Identifier: $B_{k,i}$
Input: NULL
Output 1:ω, PI_m, TS_m^k, DC
PK_k, S_m^k, σ_m^k
$SHA512(B_{k,i-1})$
(H_1, H_2)

(b) Blockchain after inserting the signed transaction block generated by AA_k

Fig. 3. Inclusion of an assignment transaction block to the blockchain


Input:$B_{k,i}(1)$
Output 1: ω, PI_n, $TS_{m,n}$, DP
$PK_m{}'$, $S_{m,n}$, $\sigma_{m,n}$

(a) Signed delegation block generated by U_m


Block Identifier: $B_{k,i}$
Input: NULL
Output 1: ω, PI_m, TS_m^k, DC
PK_k, S_m^k, σ_m^k
$SHA512(B_{k,i-1})$
Block Identifier: $B_{k,i+1}$
Input: $B_{k,i}(1)$
Output 1: ω, PI_n, $TS_{m,n}$, DP
$PK_m{}'$, $S_{m,n}$, $\sigma_{m,n}$
$SHA512(B_{k,i})$
(H_3, H_4)

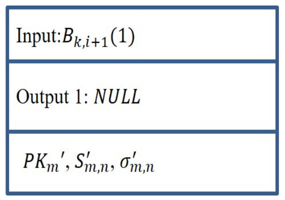
(b) Blockchain after inserting the delegation transaction block generated by U_m

Fig. 4. Inclusion of a delegation transaction block to the blockchain

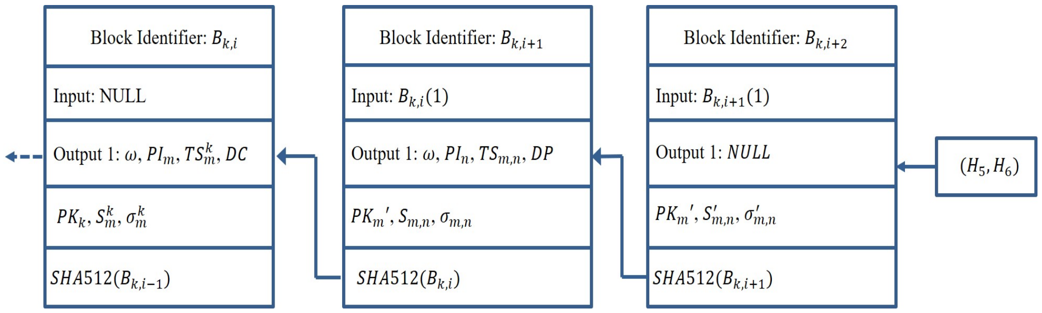
- Suppose, there exists $(i-1)$ number of blocks in the blockchain BC_k. First, BCC will validate the signature σ_m^k with AA_k's public key PK_k. If validated, BCC assigns a unique block identifier $B_{k,i}$ ($B_{k,i} = B_{k,i-1}+1$) and inserts it into the received assignment transaction block. Then, the validated transaction block is connected to the blockchain BC_k by inserting a hash pointer into the most recent block ($B_{k,i-1}$) as shown in Fig. 3(b).
- Then, BCC forwards the block $B_{k,i}$ to AA_k which allows AA_k to generate the header of the blockchain BC_k, (H_1, H_2) where $H_1 = SHA512(B_{k,i})$ and $H_2 = [H_1]_{SK_k}$. AA_k sends the chain header (H_1, H_2) to BCC. Finally, BCC validates the received block header and appends it to the blockchain BC_k as shown in Fig. 3(b). Given that each block in the blockchain includes the hash of the preceding block coupled with the fact that the header of the block includes the AA_k's signature of the preceding block's hash value, the blockchain will be tamper resistant.

2) Delegating attributes: In order to explain the procedure associated with delegating attributes, we will extend the previously discussed assigned attribute scenario to a first level delegation, in which U_m delegates the attribute ω obtained from AA_k to a fellow user U_n. The delegation process is as follows.

- Suppose, U_m has the pseudo-identity PI_n of the delegatee U_n. Then, U_m generates the delegation transaction block as illustrated in Fig. 4(a). Similar to an assignment block, delegation transaction block also includes an input, output, public key of the delegator (PK'_m), a random seed $(S_{m,n})$ and a delegation permission value, DP where $DP \in \{0,1\}$. If $DP = 0$, U_n is not allowed to re-delegate the attribute and otherwise U_n will have the permission to re-delegate further. Furthermore, it also includes the signature of the block generated with the help of U_m's secret key, SK'_m. Given that this delegation should reference the initial assignment transaction from AA_k to U_m, U_m sets the input of the delegation transaction to $B_{k,i}(1)$ (i.e. the first output of the block with the identity $B_{k,i}$ of the blockchain, BC_k). The output vector includes the delegating attribute ω, pseudo-identity of the delegatee (PI_n), the expiration timestamp $(TS_{m,n})$ and the delegation permission index DP.
- Thereafter, U_m generates the hash of block contents $M_{m,n} = SHA512(Input||Output||PK'_m||S_{m,n})$ and signs $M_{m,n}$ with the secret key SK'_m to generate the transaction signature, $\sigma_{m,n} = [M_{m,n}]_{SK'_m}$. Afterward, U_m includes the generated block signature in the delegation transaction block as shown in Fig. 4(a). The signed delegation transaction block is forwarded to the BCC.
- After receiving the signed block, BCC checks whether this first level delegation transaction is initiated by a user who has already ascertained the attribute ω from AA_k. To realize this, BCC examines the input vector



(a) Signed revocation block generated by U_m



Block Identifier: $B_{k,i}$
Input: NULL
Output 1: ω, PI_m, TS_m^k, DC
PK_k, S_m^k, σ_m^k
$SHA512(B_{k,i-1})$
Block Identifier: $B_{k,i+1}$
Input: $B_{k,i}(1)$
Output 1: ω, PI_n, $TS_{m,n}$, DP
$PK_m{}', S_{m,n}, \sigma_{m,n}$
$SHA512(B_{k,i})$
Block Identifier: $B_{k,i+2}$
Input: $B_{k,i+1}(1)$
Output 1: $NULL$
$PK_m{}', S'_{m,n}, \sigma'_{m,n}$
$SHA512(B_{k,i+1})$
(H_5, H_6)

(b) Blockchain after inserting the revocation transaction block generated by U_m

Fig. 5. Inclusion of a revocation transaction block to the blockchain

of the delegation transaction block (which points to the associated attribute assignment) and checks whether the receiver of the assigned attribute is U_m. If so, BCC uses the public key of U_m, PK'_m and validate the signature of the received delegation transaction block to ensure that it has been generated by U_m. If the signature is verified, coupled with the fact that both conditions $DC > 0$ and $TS_m^k \geq TS_{m,n}$ are satisfied, BCC deems the delegation as valid. Otherwise, the delegation transaction block is discarded.

- If the delegation block is valid, BCC inserts it into the blockchain after adding its block identifier, $(B_{k,i+1})$ as well as making a hash pointer to the previous block $B_{k,i}$ as shown in Fig. 4(b).
- To generate the new header of the blockchain, BCC mutually authenticates with AA_k using their respective RSA key pair and forwards the accepted delegation transaction block $B_{k,i+1}$ to AA_k. AA_k computes, (H_3, H_4) such that $H_3 = SHA512(B_{k,i+1})$ and $H_4 = [H_3]_{SK_k}$. AA_k sends the newly minted chain header (H_3, H_4) to BCC. Finally, BCC validates the received block header and appends it to the blockchain BC_k as shown in Fig. 4(b).

Although we described the construction of delegation transactions with respect to a first level delegation, it can simply be extended to higher level delegations. For instance let us consider a l^{th} level delegation of the attribute ω. This delegation transaction should have an input which points to the block that represents the associated $(l-1)^{st}$ level delegation transaction on the blockchain, an output which provides the information on the delegating attribute ω, pseudo-identity of the delegatee, expiration timestamp for this delegation and the DP exponent. Similar to the first level delegation transaction block, this block should also include the delegator's public key, a random seed and the block signature. When this signed block reaches the BCC, it accepts the block, if and only if the following conditions are satisfied.

- The signature of the new delegation transaction block should be validated with the public key of the user who is the delegatee in the associated $(l-1)^{st}$ level delegation transaction which is already on the blockchain.
- The output of the $(l-1)^{st}$ level delegation transaction should have a DP index of 1.
- l^{th} level delegation should expire on or before the expiry of the $(l-1)^{st}$ level delegation (i.e. $TS_{l-1} \geq TS_l$ where TS_{l-1}, TS_l correspond to the timestamps embedded in the respective delegation transactions).
- The associated attribute assignment block should have a DC exponent such that $DC \geq l$.

C. Revoking users from attributes

Both attribute assignments and delegations include expiration timestamp which determines the valid period of the considered attribute. However, if we depend upon this information completely, there is no way of revoking the attribute until it gets expired. We introduce revocation transactions to achieve this requirement. Let us assume that, U_m wants to revoke U_n from using the attribute ω. The associated process is described below.

- U_m first generates a revocation block as shown in Fig. 5(a). In this case, the input of the revocation block must reference the transaction that reflects the delegation of the attribute from U_m to U_n. Hence, the input is set to $B_{k,i+1}(1)$. To denote the revocation, the output of the newly minted revocation block is set to NULL. U_m also generates a new random seed $S'_{m,n}$ and embeds it to the block along with his public key PK'_m.
- Then, U_m generates the hash of block contents $M'_{m,n} = SHA512(Input||Output||PK'_m||S'_{m,n})$ and signs $M'_{m,n}$ with the secret key SK'_m to generate the transaction signature, $\sigma'_{m,n} = [M'_{m,n}]_{SK'_m}$ and forwards the signed revocation transaction block to the BCC.
- When BCC receives the signed revocation block, it verifies the signature and if validated, the block is appended to the blockchain with a hash pointer to the previous block (i.e. $B_{k,i+1}$) as shown in Fig. 5(b).
- As described previously in Sec. V.B, BCC fetches the updated chain header from AA_k via sending the newly accepted revocation block to AA_k. If the new block header is denoted by (H_5, H_6), then, $H_5 = SHA512(B_{k,i+2})$ and $H_6 = [H_5]_{SK_k}$.
- Finally, BCC validates the received block header and appends it to the blockchain BC_k.

D. Authorization decision making

In this subsection, we explain how access decisions are made at the LHP by verifying whether the access requester U_m, possesses a set of attributes that satisfy the governing access policy. First, U_m sends an EHR access request to LHP, indicating the EHR_{id}, EHR_{obj} and actions intended to be performed on the requested EHR_{obj} along with his pseudo-identity PI_m. Upon receiving the request, the PEP of the LHP fetches the associated Boolean statement $\mathcal{T}$ from PR and challenges U_m to mutually authenticate with LHP using the RSA keys associated with the pseudo-identity PI_m. If successfully authenticated, LHP examines the blockchains published by BCC to determine whether U_m has a set of attributes that satisfies $\mathcal{T}$.

First, we describe the process followed by LHP, to validate U_m's ownership of the attribute ω which is managed by AA_k. Given that it is managed by AA_k, LHP examines the blockchain BC_k (starting from the rightmost block) to identify a block with an output vector having ω, PI_m as the attribute and the receiver of the attribute respectively. If found, LHP checks the input of the block to determine whether it is an assignment transaction or a delegation. In the following, we describe the process associated with validation of an assigned attribute and a delegated attribute.

1) Validating an assigned attribute: LHP determines the validity of an assigned attribute, if and only if the following conditions are satisfied.

- The timestamp (embedded in the output of the block) is valid.
- The assignment is not been revoked with a revocation transaction. Suppose, the attribute assignment is represented in the first output of the block $B_{k,i}$. To determine the attribute assignment is revoked or not, LHP searches for a block (starting from the block $B_{k,i}$) which is having the input $B_{k,i}(1)$ and an output of NULL. If such block is found, the assignment transaction is deemed as revoked.

2) Validating a delegated attribute: If the block $B_{k,i}$ (which represents the delegation transaction) has an input vector pointing to an output of a different block, then it represents a delegation transaction. The associated procedure for validating a delegation transaction is as follows.

- First, the transaction in $B_{k,i}$ is validated using the two steps used for validating an assigned attribute. This will only provide evidence for the fact that the current delegation to U_m is valid and the user who delegated the attribute to U_m has not revoked it.
- However, still it is necessary to check whether the delegator still owns the attribute. LHP examines the input of the block $B_{k,i}$, to find out the block that embeds the transaction which shows the delegator's ownership of the attribute ω. Then, the second step associated with validating an assigned attribute is carried out to figure out whether the delegator is revoked from the attribute or not.
- The aforementioned process is continued until an assignment transaction is reached and the same mechanism is followed to check whether the AA who issued the attribute has not revoked the initial attribute assignment.
- If all the above conditions are satisfied, the delegated attribute is deemed to be valid.

According to the protocols mentioned above, LHP evaluates whether the user with pseudo-identity PI_m owns a valid set of attributes (either assigned or delegated) to satisfy the access requirement of the requested EHR_{obj} and thereby makes the decision on provisioning or denying access.

VI. Security Analysis

In this section, we intend to show that the proposed scheme is secure against attribute forgery, attribute collusion while providing user privacy through enforcing pseudo-anonymity guarantees to users.

Resistance against attribute forgery: Let us consider that an adversary $\mathcal{A}$ intends to forge the attribute ω with a first level delegation. Further note that $\mathcal{A}$ has the knowledge that the user with pseudo-identity PI_m has already ascertained it from the AA which manages ω. If $\mathcal{A}$ to succeed with forgery of the first level delegation, $\mathcal{A}$ should be able to generate a delegation transaction block and signs it with the secret key associated with the pseudo-identity PI_m. This requires a universal forgery (i.e. The ability of an adversary to sign a given message while having only access to the public parameters) of the RSA signature. Given that the RSA signature scheme is universally unforgeable under the RSA assumption, the attribute forgery will be a failure. Although, we considered a delegation transaction, every attribute related transaction (assignment, delegation and revocation) is secure against forgery under the RSA assumption, given that the generation of each and every transaction requires the RSA signature of the entity which generates the transaction.

Resistance against attribute collusion: The proposed scheme inherently supports resistance against the collusion of attributes. This is due to the fact that a user will only be able to prove the ownership of attributes which are associated with the pseudo-identity that he used to mutually authenticate with the LHP.

Pseudo-anonymity: In this scheme, users generate their own pseudo-identities and they do not require to register with any trusted party to initialize and obtain necessary keys. Although users' pseudo-identities are included in plain in the public blockchain, it is not feasible for a third-party to simply relate a given pseudo-identity to the real world identity of the user. Hence, the proposed scheme provisions pseudo-anonymous guarantees to users.

VII. Discussion

In this section, we compare the characteristics of the proposed scheme with the existing delegatable ABAC schemes.

There have only been few related works which have explored the issue of flexible access delegation in attribute based systems. As we have pointed out in Sec. II, the schemes proposed in [14], [15] lack the control over access delegation. In contrast, the proposed scheme achieves control over delegation via allowing a delegator to permit or deny further delegations by the delegatees as well as enforcing a limit on the maximum permissible length of a delegation chain similar to the scheme presented in [6]. Although, the scheme proposed in [6] provides flexible delegatable access to users, it imparts high computational overhead when revoking a specific attribute from a user, since this requires sending new attribute tokens to all users who share the revoking attribute except the user to be revoked. However, this can simply be achieved in the proposed scheme via generating a signed revocation transaction. When we consider the end-user key management overhead, the scheme presented in [6] requires a user to have a separate secret key for each of the attribute that the user is delegating to other users (delegating user acts as a virtual AA). In comparison, the proposed scheme has a very low end-user key management overhead due to the fact that each user only requires to store a single secret key and it is independent of whether the user is involved in delegation or not.

VIII. Conclusion

In this paper, we have proposed an attribute based access control scheme integrated with controlled access delegation capabilities. The access control scheme is presented with respect to a health information sharing scenario where flexible access should be granted for both registered users in the LHP as well as users from foreign domains. We have utilized blockchains to manage attribute assignments, delegations as well as revocations which makes the user authorization for accessing patient EHRs, a light-weight process. Our scheme achieves control over delegation by allowing the delegating user to specify whether or not the delegatee is allowed to further delegate (through the delegation transaction) as well as bounding the maximum number of permissible delegations for a given chain of delegations (through the assignment transaction). Furthermore, the proposed scheme is integrated with a user revocation mechanism which imparts a significantly lower computational overhead as well as end-user key management overhead compared to the existing ABAC schemes with delegatability. Given the fact that the blockchains provide a tamper resistant activity log of a particular user's attribute based transactions; we intend to use this feature to extend the proposed ABAC scheme with an appropriate trust model in our future work, which would further enhance the trustworthiness of access decision making as well as the security of the access control scheme.

References

[1] H. S. G. Pussewalage and V. A. Oleshchuk, "Privacy preserving mechanisms for enforcing security and privacy requirements in e-health solutions," *International Journal of Information Management*, vol. 36, no. 6, Part B, pp. 1161 – 1173, 2016.

[2] A. Ghazvini and Z. Shukur, "Security challenges and success factors of electronic healthcare system," *Procedia Technology*, vol. 11, pp. 212 – 219, 2013.

[3] "Privacy rights clearinghouse: Security breaches 2005 - present," Accessed: 16.10.2015. [Online]. Available: http://www.privacyrights.org/data-breach

[4] "Health care and cyber security increasing threats require increasing capabilities," Accessed: 30.05.2016. [Online]. Available: http://tinyurl.com/srvy2015

[5] "Perspectives on cybersecurity in healthcare," Accessed: 30.11.2016. [Online]. Available: http://www.wedi.org/docs/test/cyber-security-primer.pdf

[6] H. S. G. Pussewalage and V. A. Oleshchuk, "Attribute based access control scheme with controlled access delegation for collaborative e-health environments," *Journal of Information Security and Applications*, vol. 37, pp. 50 – 64, 2017.

[7] X. Yi, Y. Miao, E. Bertino, and J. Willemson, "Multiparty privacy protection for electronic health records," in *Proceedings of the IEEE Global Communications Conference*, Dec. 2013, pp. 2730–2735.

[8] H. S. G. Pussewalage and V. A. Oleshchuk, "An attribute based access control scheme for secure sharing of electronic health records," in *Proceedings of the 19th IEEE International Conference on E-health, Networking, Application and Services*, Sep. 2016, pp. 526–531.

[9] H. S. G. Pussewalage and V. A. Oleshchuk, "Privacy preserving mechanisms for enforcing security and privacy requirements in e-health solutions," *International Journal of Information Management*, vol. 36, no. 6, Part B, pp. 1161 – 1173, 2016.

[10] H. S. G. Pussewalage and V. A. Oleshchuk, "An efficient multi-show unlinkable attribute based credential scheme for a collaborative e-health environment," in *Proceedings of the 3rd IEEE Collaboration and Internet Computing*, Oct. 2017, pp. 421–428.

[11] V. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations," *National Institute of Standards and Technology Special Publication*, vol. 800-162, 2014, Available: http://dx.doi.org/10.6028/NIST.SP.800-162.

[12] H. S. G. Pussewalage and V. A. Oleshchuk, "A distributed multi-authority attribute based encryption scheme for secure sharing of personal health records," in *Proceedings of the 22nd ACM Symposium on Access Control Models and Technologies*, Jun. 2017, pp. 255 – 262.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2007, pp. 321–334.

[14] S. Ding, Y. Zhao, and H. Zhu, "Extending fuzzy identity-based encryption with delegating capabilities," in *Proceedings of the 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, Aug. 2011, pp. 19–23.

[15] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-policy attribute-based threshold decryption with flexible delegation and revocation of user attributes (extended version)," Apr. 2009.

[16] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Crytocurrency Technologies*. Princeton, NJ: Princeton University Press, 2016.

[17] A. A. Antonopoulos, *Mastering Bitcoin*. Sebastopol, CA: O'Reilly Media Inc., 2014.

[18] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Accessed: 15.01.2018. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[19] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 459–474.