The Bitcoin Network as Platform for Trans-Organizational Attribute Authentication

Technical Report · April 2015

CITATIONS
0

READS
497

2 authors:

Jason Paul Cruz
Osaka University
29 PUBLICATIONS **238** CITATIONS
SEE PROFILE

Yuichi Kaji
University Hospital Medical Information Network
163 PUBLICATIONS **3,740** CITATIONS
SEE PROFILE

The Bitcoin Network as Platform for Trans-Organizational Attribute Authentication

JASON PAUL CRUZ[1,a)] YUICHI KAJI[1,b)]

Abstract: The role-based access control (RBAC) is a natural and versatile model of the access control principle. In the real world, an organization commonly provides a service to a user who owns a certain role that was issued by a different organization. However, such a trans-organizational RBAC is not common in a computer network because it is difficult to establish both the security that prohibits malicious impersonation of roles and the flexibility that allows small organizations/individual users to fully control their own roles. This study proposes a system that makes use of Bitcoin technology to realize a trans-organizational RBAC mechanism. Bitcoin, the first decentralized digital currency, is a payment network that has become a platform for innovative ideas. Bitcoin's technology, including its protocol, cryptography, and open-source nature, has built a good reputation and has been applied in other applications, such as trusted timestamping. The proposed system uses Bitcoin technology as a versatile infrastructure to represent the trust and endorsement relationship that are essential in RBAC and to realize a challenge-response authentication protocol that verifies a user's ownership of roles.

Keywords: role-based access control, trans-organizational role, information security, Bitcoin, trusted-timestamping

1. Introduction

1.1 Roles and Role-Based Access Control

Roles and titles are often used to distinguish the eligibility of people to access certain services. Such mechanism is modeled as the role-based access control (RBAC) [1] framework, which describes the access control relation among users and services. In RBAC, users are associated with roles, and roles are associated with services. This framework is compatible with the access control requirements of real-world organizations and is employed in the computer systems of many organizations. However, it must be noted that roles of users are often used in a trans-organizational manner. For example, students can purchase computer software at an academic-discounted price. In this example, the "student" role that is issued by an organization (school) is used by another organization (computer shop) to determine if a guest is eligible to receive a certain service (discounted price). This kind of trans-organizational use of roles is common in face-to-face communication, but it is not obvious in computer networks. Even if one has a certain role (student role) that is issued by an organization (school), he/she has no systematic way of convincing a third-party organization (computer shop) that he/she really has that role.

For realizing a trans-organizational RBAC in a computer network, we need a mechanism that prevents malicious users from disguising their roles. This requirement is naturally accomplished in real-world services with the use of physical certificates, such as passports and ID-cards, which are difficult to forge or alter, but the problem is not obvious in a computer system. Digital certificates [2] can be utilized as an analogue of physical certificates, but the use of digital certificates is not favorable because it requires considerable and continuous elaborations to maintain secure public-key infrastructures. Another less sophisticated approach to the security problem is to let a service-providing organization (computer shop) inquire a role-issuing organization (school) about the user-role assignment. This approach works fine in some cases [3], but a focal point of this approach is the necessity for the agreed beneficial relationship among organizations. It is often difficult for a new organization to join existing partnership, which severely restrict the trans-organizational utilization of roles.

1.2 Bitcoin

Bitcoin is a decentralized global currency cryptosystem that has increased in value and popularity since its inception [4], [5]. Bitcoin aims to enable complete digital money that is secure and decentralized; it is based upon a peer-to-peer network powered by its users, and no central authority is assumed. To achieve this, transactions are publicly announced and the participants agree on a single history of these transactions, which are grouped into blocks, given timestamps, and then published. The hash of each timestamp includes the previous timestamp to form a chain, making accepted blocks difficult to alter. Based on this security, Bitcoin features many favorable properties, including easy mobile payments, reliability, high availability, fast international payments, zero or low fees, protected identity, and privacy [6].

1.3 Bitcoin as an Infrastructure

This study aims to develop a practical system that uses Bitcoin technology to realize the trans-organizational utilization of roles. We investigate a realization of a user-role assignment that is secure (users cannot disguise roles), user-oriented (users can disclose their roles to any organization), and open (anyone can verify if a user has a certain role that is managed and issued by another organization). The key idea is to record the relation between users and roles as a transaction in the Bitcoin network. The service providing organization only needs to confirm if an unknown guest is really the user or not, which can be accomplished by a challenge-response protocol. Bitcoin's protocol and cryptography make the proposed system secure, flexible, and furthermore, allow flexible role management operations such as the endorsement and hierarchical roles.

1 Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0101, Japan
a) jpmcruz@ymail.com
b) kaji@is.naist.jp

2. Models for the Role-Based Access Control

Among many issues of the RBAC framework, this study mainly focuses on the realization of the user-role assignment in a trans-organizational scenario. Other issues of RBAC are also important, but they are excluded from our discussion. To clarify the scope of this study in the entire framework of RBAC, an abstract model of RBAC and its extension are discussed first.

In the simplest model of the RBAC [1], the access structure is defined by three sets and two relations; the set U of *users*, the set R of *roles*, the set S of *services*, a *user-role assignment* $\mathrm{UA} \subset U \times R$, and a *role-service assignment* $\mathrm{SA} \subset R \times S$. A user u is eligible to access a service s if and only if there is a role r such that $(u, r) \in \mathrm{UA}$ and $(r, s) \in \mathrm{SA}$. In real-world services, roles can be used in a trans-organizational manner. A role that was issued by a *role-providing entity* can be referred by a foreign *service-providing entity* to determine if a service should be given to an unknown guest. An interesting point here is that the role-providing entity is not always concerned about the service-providing entities. Indeed the foreign service-providing entity is not always allowed access to the user-role assignment, and thus, the service-providing entity needs to devise an alternative means to confirm if an unknown guest has a certain role. To deal with this kind of framework, we consider extending the basic model of RBAC by introducing a set of organizations.

The *trans-organizational RBAC* is defined similarly to the usual RBAC, but a set O of *organizations* is defined in addition to the sets of users, roles, and services. The set R of roles is partitioned into several subsets, with each subset of R associated with an element in O, that is, $R = R_{o_1} \cup \cdots \cup R_{o_n}$, where $o_1, \ldots, o_n \in O$ and $R_{o_i} \cap R_{o_j} = \emptyset$ if $i \neq j$. To make the relation among roles and organizations explicit, a role r in R_{o_i} is written as $o_i.r$. Similarly, the user-role assignment UA is partitioned into disjoint subsets; $\mathrm{UA} = \mathrm{UA}_{o_1} \cup \cdots \cup \mathrm{UA}_{o_n}$ where $\mathrm{UA}_{o_i} \subset U \times R_{o_i}$. We are intending that $o_i.r \in R_{o_i}$ means that the role $o_i.r$ is managed by the organization o_i and the assignment of users to $o_i.r$ is controlled by that organization o_i. In the trans-organizational RBAC, a user u demands a service s by asserting his/her role $o_i.r \in R_{o_i}$ that has been provided by a role-providing organization o_i. A service-providing organization provides service s to the user u if and only if $(u, o_i.r) \in \mathrm{UA}_{o_i}$ and $(o_i.r, s) \in \mathrm{SA}$. Note that the test of $(o_i.r, s) \in \mathrm{SA}$ is easy for the service-providing organization because the assignment SA is defined by the organization itself. On the other hand, the test of $(u, o_i.r) \in \mathrm{UA}_{o_i}$, which is sometimes called an *authentication*, is not obvious because the role-providing organization o_i may not disclose UA_{o_i} to the public.

The trans-organizational RBAC will be realistic only if the authentication of roles $(u, o_i.r) \in \mathrm{UA}_{o_i}$ is accomplished by a practical means. Physical certificates, such as passports and ID-cards, have been used widely for many years, but these certificates cannot be easily imported to the digitalized world over a computer network. Digital certificates have been studied for the replacement of physical certificates [2], but they are not accepted widely because of the cost issues for acquiring these certificates, keeping related keys secure, and maintaining a public-key infrastructure (PKI) [7], [8]. A less sophisticated but simpler approach is to arrange a mutual agreement between role-providing organizations and service-providing organizations. However, such a framework will be semi-closed, and possible among organizations that share identical benefits. The authors have studied another approach for realizing secure authentication of roles by utilizing a special cryptography known as hierarchical ID-based encryption [9]. The scheme in [9] offers some advantages over other existing approaches, but it still has some problem in managing cryptographic keys. Consequently, a scheme for secure and practical role authentication in the trans-organization scenario has not been established.

3. The Bitcoin Protocol

Bitcoin is a collection of cryptographic protocols for secure online transactions between users [10], [11]. Users own digital wallets that handle the creation and storage of private keys and corresponding public *Bitcoin addresses*. A user can send a certain amount of Bitcoins (BTC) to another user by creating a *transaction* with the sender's Bitcoin address/es as input/s and the receiver's Bitcoin address/es as output/s. Transactions are validated by miners and recorded in a global public ledger that is called the *blockchain*. The validation of transactions requires some amount of computation, and the miner who succeeds in validating these transactions is rewarded with Bitcoins and the transaction fees. Validated transactions cannot be altered unless an attacker has computation power that overwhelms the total computation powers that are possessed by all other miners.

3.1 Bitcoin Addresses

A Bitcoin address is 160-bit hash of a public key of an Elliptic Curve Digital Signature Algorithm (ECDSA). The public key undergoes several cryptographic processes (SHA-256, RIPEMD-160 and Base58 Encoding) to be converted into a valid Bitcoin address. A new ECDSA keypair is generated for each Bitcoin address. A user can create any number of Bitcoin addresses easily and for free, and thus, users usually use a digital wallet to store multiple keypairs. The users should backup and secure the private keys (or the wallet data file) because these private keys are needed to use the Bitcoins that are stored in the corresponding Bitcoin addresses. A Bitcoin address is in the form of random numbers and letters, e.g., 19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6. There are 2^{160} possible Bitcoin addresses that can be created. A Bitcoin address is considered to be "unique" as it is extremely unlikely for two users to independently generate the same Bitcoin address.

3.2 Transactions, Blocks, and the Blockchain

A transaction is a digitally signed data that is broadcasted to the Bitcoin network and then included in a block in the blockchain. A transaction contains a transaction ID (used to identify the transaction), the list of input addresses (addresses from which Bitcoins are transferred) and the list of output addresses (which contains the receiving addresses and the amounts of BTC being transferred). A sender of the transaction

has to prove that he/she has control of the addresses in the list of input addresses by signing them with the corresponding private keys. Think of a Bitcoin address as a transparent vault where anyone can check the amount of Bitcoins inside, but only the one who has the private key can spend the coins inside it.

3.3 Blocks and the blockchain

Transactions are grouped together in blocks and then recorded in the public ledger of the Bitcoin network. A block contains a record of transactions that have not been recorded in any previous block. Blocks are connected and linked together to form a blockchain, where a new block is added to the block that came before it. Every block contains the hash of the previous block, creating a chain that connects the first block (genesis block) to the current block. A block contains, among other things, a hash of the previous block, a hash of the merkle root of valid transactions to be included in this block, and a nonce (a unique solution to a difficult mathematical puzzle), as shown in Fig. 1. The entire blockchain and every transaction included in the blocks can be viewed online using a blockchain browser.

3.4 Mining and Proof-of-Work

Blocks are added to the blockchain through the process called *mining*. This process uses a proof-of-work system wherein miners all around the world use special software to solve mathematical problems. The mathematical problem is inherently difficult to solve and requires a "brute force" solution; that is, miners scan and test for a nonce that gives a correct solution to the mathematical problem. During mining, the mining hardware of a miner (CPUs, GPUs, FPGAs, and ASICS) runs a cryptographic hash function composed of two rounds of SHA256 on the block header. The mining software increments the nonce as the random element in the block header until a valid hash is found. To control the difficulty of mining, a parameter called a difficulty target is agreed upon by miners. The difficulty target can be regarded as a threshold that is used in such a way that a miner is required to find a nonce that makes the hash of the block smaller than the difficulty target (equivalently, the hash values should start with a certain number of zeroes). To compensate for increasing hardware speeds, the difficulty target is adjusted every 2016 blocks so that it takes on average 10 minutes to find a valid nonce. The difficulty target is expressed as the difficulty on creating the current block compared to generating the first block and is determined as follows:

$$\text{difficulty} = \frac{\text{difficulty\_1\_target}}{\text{current\_target}}$$

When a miner finds the correct nonce value that creates a hash value less than the target, it forwards the block to the rest of the nodes in the network. After validating the solution for the block, miners move on to solving for the next block.

Sometimes, more than one miner may find a valid solution at almost the same time, consequently forking the blockchain. This inconsistency is resolved when the solution for the next block is found and one of the branches becomes longer. In the Bitcoin protocol, miners always work on the longest chain , and thus, in case of a fork, the shorter chain is orphaned.

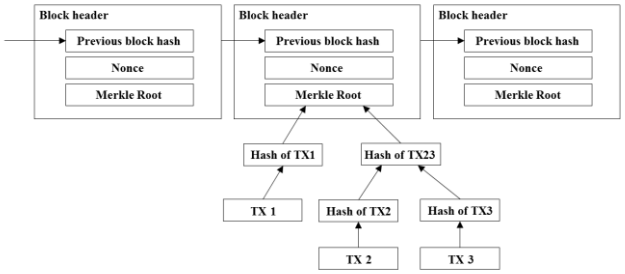

Fig. 1 A simplified representation of the Bitcoin blockchain

The miner who solves the block is awarded with newly "minted" Bitcoins (currently at 25 BTC) and transaction fees of the transactions included in the solved block. This process of mining guides the issuance of Bitcoins and incentivizes miners to keep mining and approving transactions.

The security of Bitcoin relies on this proof-of-work system, which means that a block cannot be modified without redoing the work spent on it, including the work spent on blocks chained after it. Given this design, as long as majority of the overall CPU power participating in the Bitcoin network is controlled by honest miners, an attacker will be outpaced by the honest miners, making it almost impossible to modify a published block.

3.5 Attacks on the Bitcoin Network

Some strategies, both theoretically and in practice, have been devised to attack the security of the Bitcoin protocol. These attacks have been designed for dishonest or rogue miners, i.e., those who do not follow the Bitcoin protocol, to get rewards higher than their contribution to the network. These strategies include the pool hopping attack [12], the mining cartel attack [13], selfish mining [14], block withholding attack [15], and hardware attacks. These attacks are designed to infiltrate factors outside the blockchain, targeting the client side and stealing Bitcoins from them. These attacks aim to steal Bitcoins and/or gain higher rewards and not to modify transactions or the blockchain. Therefore, Bitcoin's security remains intact. For our purposes, the transactions between organizations and users will remain secure because they are recorded in the blockchain.

4. Proposed Scheme

4.1 Overview

The proposed system is a non-conventional authentication mechanism that is suitable for the trans-organizational utilization of roles. The idea is to provide an irrefutable proof of the role of a user issued by an organization by verifying the connection of the user to the organization through the Bitcoin blockchain. Consider for example that A-university would like to manage a "student" role for its students. First, it would perform a Bitcoin transaction using its own public Bitcoin address/es as input/s and

the corresponding students' public Bitcoin address/es as output/s. Upon request for a service from an unknown user who asserts that he/she possesses the student role of A-university, a service-providing organization, for example a restaurant, will verify the Bitcoin transaction containing the Bitcoin addresses of A-university and the student, which connects the student role managed by A-university to the output address in the transaction. After establishing the connection, the restaurant can verify (through a challenge-response protocol) if the unknown user has access to the output address in the transaction, which finally connects the student role from A-university to the unknown user.

Note that the restaurant does not have to know anything about the role beforehand, and does not have to make any contract or inquiry to A-university that has assigned the role to the student because the details needed by the restaurant are published publicly and/or possessed by the user. In the proposed system, there is no essential difference between users and role-issuing organizations because they both can be the sender and receiver in the Bitcoin transactions (but for simplicity, the role-issuing organizations will be differentiated from the users).

4.2 Procedures

Fig. 2 shows the overall structure of the proposed system. In this model, we assume that the role-issuing organizations are Bitcoin users while the users and service-providing organizations may or may not be Bitcoin users. Bitcoin user means that the entity owns a Bitcoin wallet and performs Bitcoin transactions.

4.2.1 Pre-requisites

An organization (o_1) generates n Bitcoin addresses, where n is the number of roles that o_1 wants to manage. The creation of these Bitcoin addresses (and the corresponding private keys) can be accomplished using several options, including Bitcoin wallets and online/offline Bitcoin address generators. After generating the n keypairs, o_1 keeps the private keys secret and secure, and publishes the list of pairs of Bitcoin addresses and corresponding roles using chosen media (e.g., Website, database, etc.) to make it available to the public. We write $o_1.\mathrm{BPK}_i, o_1.\mathrm{BA}_i$, and $o_1{:}r_i$ for the private key, Bitcoin address, and the role that is associated with the address $o_1.\mathrm{BA}_i$, respectively, where $1 \le i \le n$.

The publication of these Bitcoin addresses will serve as proof that o_1 owns and manages the addresses (it should be noted that o_1 will not gain any benefit from publishing Bitcoin addresses that it does not own, and thus, will only publish addresses it owns).

Similarly, a user (u) generates a pair of a private key u.BPK and a Bitcoin address u.BA. Alternatively, o_1 can generate the (u.BPK, u.BA) keypair and send it to u through a secure communication channel. Note however, that it is recommended by the Bitcoin community that only the one who created the keypair should be in possession of the keypair because the private key is used for accessing the Bitcoins stored in the corresponding address.

4.2.2 Creating the role-issuer and user connection

The organization o_1 creates a simple Bitcoin transaction using $o_1.\mathrm{BA}_n$ as input address and u.BA as output address. In this transaction, o_1 sends an arbitrary amount of Bitcoins to u; currently the minimum amount that can be used for a transaction to be considered valid is 0.00010001 BTC = 0.04 USD (1 satoshi = 0.00000001 BTC plus 0.0001 BTC required transaction fee). Optionally, o_1 can include a higher transaction fee or miners' fee if it wants its transaction to be prioritized in the current round of solving for the block (but for our purposes, if the time of confirmation is not vital, the minimum transaction fee is sufficient). After confirming the details of the transaction, o_1 sends the transaction to the Bitcoin network awaiting for confirmations from miners that it is permanently included in a block in the blockchain. Once included in the blockchain, certain details will be publicly available, including $o_1.\mathrm{BA}$, u.BA, amount of BTC transferred, transaction ID, block number, received time, and the time it was included in the block.

4.2.3 Verifying a user-role assignment

Assume that user u visits a service-providing organization o_2 and asserts that he/she has the role of $o_1{:}r_n$ that was issued by o_1. The organization o_2 inquires u for the Bitcoin address, say u.BA, that was granted the asserted role of $o_1{:}r_n$ from o_1. Then o_2 will (i) determine the Bitcoin address $o_1.\mathrm{BA}_n$ that is associated with the role $o_1{:}r_n$, (ii) confirm the existence of the transaction from $o_1.\mathrm{BA}_n$ to u.BA, and (iii) verify if u is the genuine owner of u.BA. The Bitcoin address $o_1.\mathrm{BA}_n$ can be found through the medium where o_1 published the Bitcoin addresses it owns. The confirmation of the transaction can be done by using a blockchain browser or a similar program. Steps (i) and (ii) assure o_2 that the role $o_1{:}r_n$ and other related information associated with $o_1.\mathrm{BA}_n$ are assigned by o_1 to the owner of u.BA. The ownership of u.BA is verified by a challenge-response protocol where ECDSA keys that are associated with the Bitcoin address u.BA are utilized.

4.2.4 Challenge-Response Protocol

The organization o_2 chooses an arbitrary data m and requests u to sign it, together with u.BA, using the private key u.BPK. The signature is defined by $S = Sign(u.\mathrm{BPK}; u.\mathrm{BA}, m)$, and thus a correct S will only be created if u has u.BPK. User u then sends the signature back to o_2 and o_2 will verify using the function $Verify(u.\mathrm{BA}, m, S)$. Remark that o_2 can confirm if u has access to the role $o_1{:}r_n$ without querying o_1, and that u has little chance to disguise his/her role.

5. Role Management

In the proposed framework, the relation between users and roles is represented by the users' possession of the private keys. This approach involves a possible security risk; the leakage and loss of keys.

5.1 Personalization of roles

If a user leaks his/her private key, then the people who happen to know the key can also prove ownership of the corresponding Bitcoin address (which in turn can be used to prove that a role associated with the address was assigned to them). Note that the intended user can still prove ownership of the Bitcoin address even after leakage, although such an inappropriate usage of keys can obstruct fair and reliable access control. To deter such irresponsible behavior of users, the proposed system offers three possible measures:

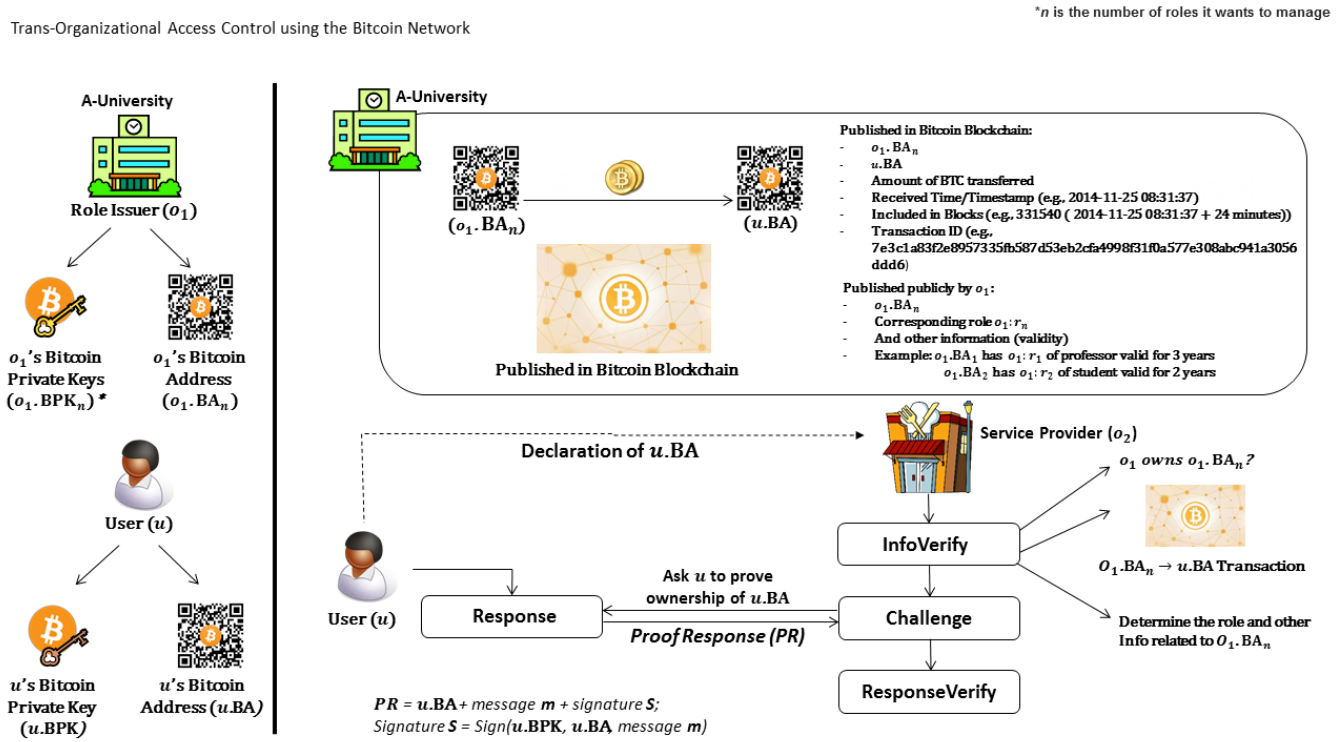

Fig. 2 Overview of the proposed structure

1. Given that the proposed system uses Bitcoin technology, it inherently has a "traitor tracing" capability because the Bitcoin addresses are unique (thus, they can be possibly mapped to users) and the transactions are published publicly. Thus, if a user receives a leaked key and maliciously uses the role associated to the corresponding Bitcoin address, a consequent investigation can possibly lead to the original user associated with the Bitcoin address.
2. Role-issuing organizations can "personalize" roles by including some unique identifiers (which can be encrypted as well) to the data it will publish publicly. For example, A-university can publish "o_1.BA$_1$ issues a 'student' role to student #123 with 6 months validity."
3. Role-issuing organizations can "personalize" roles by making use of a public note that is included in the Bitcoin transaction. This public note is a feature offered by an online wallet (blockchain.info) [16] and is not part of the Bitcoin protocol.

With these measures, a student will be more conscious of leaking/losing his/her key to another person because he/she will have the risk of being identified and subsequently punished for irresponsible behavior.

5.2 Role Re-issuance

If the private keys are lost or forgotten, or if access to the digital wallet is lost or forgotten, then control over the corresponding Bitcoin addresses is also lost. The ownership of the Bitcoin addresses cannot be verified or proven without the corresponding private keys. In this case, the role-issuing organizations can easily re-issue the roles by creating another Bitcoin transaction to the new Bitcoin address of the compromised user. The overhead of role re-issuance is relatively low for both the role-issuing organizations and the user.

Moreover, to make sure that the compromised Bitcoin addresses will not be used maliciously, role-issuing organizations can create revocation lists containing these addresses in the media where they publish the Bitcoin addresses they own.

5.3 Additional Security Measures

Wallets are the most common target of attacks, but of course, security measures have been implemented and are recommended to minimize such cases. In the proposed system, the purpose of the Bitcoin transaction is to connect the user to an organization and to a role. If the user is a Bitcoin user, he/she is recommended to use other wallets or other addresses to store Bitcoins. Ultimately, the user only needs to store the private key safely, and even keep it offline. The challenge-response protocol can be performed offline. Moreover, an attacker with no prior knowledge of the proposed system and the role associated with the address will have no motives to steal private keys.

5.4 Endorsement

The Bitcoin network provides a natural connection between Bitcoin addresses published in the blockchain. This function can be used to realize some personal activities that are not considered in the conventional RBAC approach. For example, in the real world, an endorsement among individuals sometimes plays an important role. Semi-closed organizations, such as academic societies and golf clubs, have the tradition or policy that a newcomer must be endorsed by a current member. This mechanism can be realized by extending the proposed system.

Based on the system shown in Fig. 2, consider for example that Alice (u) is an authorized member of XYZ golf club (o_1). This relationship is realized by the Bitcoin transaction from o_1.BA to u.BA. If Alice would like to endorse Bob (u_2) to o_1, then she can similarly create a Bitcoin transaction from u.BA to u_2.BA, linking their addresses. Then, Bob can go to o_1 and declare u_2.BA. The club can look up the blockchain and check that u_2.BA was endorsed by u.BA, which was originally endorsed by the club, as represented by o_1.BA. Once the connection is established, o_1 can verify if u_2 is the owner of u_2.BA by using the challenge-response protocol. By querying the blockchain and through the challenge-response authentication, the club does not have to inquire Alice for the verification of the endorsement.

5.5 Trusted Timestamping as Proof of Validity

Trusted timestamping is the process of securely creating a proof, i.e., timestamp, of the creation or modification time of a document. It is used for proving that certain information or document existed at some point in time and has not been tampered or modified since. Traditional timestamping processes follow the RFC 3161 standard, wherein the timestamp is issued by a trusted third party acting as a Time Stamping Authority (TSA) [17]. The Bitcoin network features a timestamp server used in the blockchain to link the blocks together in a chronological manner. This timestamp server has been used, outside the main purpose of Bitcoin, as a trusted timestamping mechanism for digital documents given that it is secure (extremely difficult to attack and modify), robust (DoS resistant), and a trustworthy source of time (i.e., the time a transaction is included in the blockchain) [18], [19]. Put simply, a hash of the data that a user wants to timestamp is converted into a Bitcoin address. The timestamping service (or the user his/herself) then creates a Bitcoin transaction and makes a small payment to the converted Bitcoin address. This transaction is then stored in the public blockchain. Anyone who wants to verify the point in time a data (i.e., the hash of it) existed can be connected to the time the transaction that includes the corresponding converted Bitcoin address was included in the blockchain. This timestamping scheme is innovative and provides additional features as compared to the traditional trusted timestamps issued by TSAs, which are prone to data corruption and tampering.

The timestamp server of Bitcoin provides a natural solution to the inclusion of expiration dates or validity of the roles in the proposed system. The role-issuing organization includes the expiration dates or validity of the roles it manages in the information it publishes publicly. In this way, the service-providing organization can verify the validity of a role simply by investigating the timestamp of the block where the transaction was included in the blockchain and comparing it with the details published by the role-issuing organization.

6. Conclusion

The Bitcoin protocol was utilized as an infrastructure to realize a trans-organizational RBAC and represent the trans-organizational usage of roles. The proposed system provides a secure mechanism for verifying the user-role assignments of organizations. Compared to other similar approaches, the proposed scheme provides more flexibility and autonomy while maintaining security. This mechanism allows the realization of many collaborative right managements that are common in physical communication but are difficult to implement over computer networks. Finally, the timestamping mechanism provided in the Bitcoin protocol provides a natural solution to the inclusion of expiration dates in the created roles. Future research will focus on the realization of a hierarchical system and on the development of a prototype for easier use and interoperability.

References

[1] Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E.: “Role-based access control models,” IEEE Computer, 29, 2, pp. 38–47 (1996).

[2] Farrell, S. and Housley, R.: “An internet attribute certificate profile or authorization,” RFC 3281 (2002).

[3] Internet2: “The Shibboleth System,” (retrieved on January 20, 2015). http://shibboleth.internet2.edu.

[4] Nakamoto, S.: “Bitcoin: A Peer-to-Peer Electronic Cash System” (2008) (retrieved on January 20, 2015). https://bitcoin.org/bitcoin.pdf.

[5] Blockchain Info: “Currency Stats,” (retrieved on December 10, 2014). https://blockchain.info/stats.

[6] Bitcoin.org: “Bitcon for Individuals,” (retrieved on January 20, 2015). https://bitcoin.org/en/bitcoin-for-individuals.

[7] Ellison, C.M. et al.: “SPKI certificate theory,” RFC 2693 (1999).

[8] Gutmann, P.: “Simplifying public key management,” IEEE Computer, 37, 2, pp. 101–103 (2004).

[9] Cruz, J.P. and Kaji, Y.:, “Trans-Organizational Role-Based Access Control in Android,” INFOCOMP 2014, pp. 114–119 (2014).

[10] Bitcoin Wiki: “Protocol Specification,” (retrieved on January 20, 2015). https://en.bitcoin.it/wiki/Protocol_specification.

[11] Bitcoin.org: “Bitcon Developer Guide,” (retrieved on December 10, 2014). https://bitcoin.org/en/developer-guide#block-chain.

[12] Rosenfeld, M.: “Analysis of Bitcoin Pooled Mining Reward Systems” (2011) (retrieved on December 10, 2014). https://bitcoil.co.il/pool_analysis.pdf.

[13] RHorning: “Mining cartel attack,” Bitcoin Forum (2010) (retrieved on December 10, 2014). https://bitcointalk.org/index.php?topic=2227.0.

[14] Eyal, I. and Sirer, E.G.: “Majority is not Enough: Bitcoin Mining is Vulnerable,” Financial Cryptography and Data Security (2014).

[15] Courtois, N.T. and Bahack, L.: “On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency,” CoRR (2014) (retrieved on December 10, 2014). http://arxiv.org/pdf/1402.1718v5.pdf.

[16] Blockchain Info: “Blockchain Website FAQ,” (retrieved on January 20, 2015). https://blockchain.info/wallet/website-faq.

[17] SSL4NET: “Trusted Timestamping,” (retrieved on January 20, 2015).http://www.ssl4net.com/technology/trusted-timestamping

[18] Klusácek, J.: “Trusted Timestamp in Cryptocurrency Block Chain,” Student EEICT (2014).

[19] BTProof: “Trusted timestamping on the Bitcoin blockchain,” (retrieved on January 20, 2015). https://www.btproof.com.