The Bitcoin Network as Platform for Trans-Organizational Attribute Authentication

Jason Paul Cruz and Yuichi Kaji
Nara Institute of Science and Technology, Graduate School of Information Science
Nara, Japan
E-mail: jpmcruz@ymail.com, kaji@is.naist.jp

***Abstract*—The role-based access control (RBAC) is a natural and versatile model of the access control principle. In the real world, it is common that an organization 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.

I. INTRODUCTION

A. 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/companies. However, it must be noted that RBAC is a versatile framework, and roles are often used in a trans-organizational manner. For example, students are often allowed to 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.

To realize a trans-organizational RBAC mechanism in a computer network, a mechanism that prevents malicious users from disguising their roles is necessary. This requirement is naturally accomplished in real-world services with the use of physical certificates, such as passports and ID-cards, which are expected to be difficult to forge or alter. This problem, however, 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 (the computer shop in the above example) 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. Consequently, it is difficult for a new organization to join the partnership, severely restricting the trans-organizational utilization of roles.

B. Bitcoin

Bitcoin is a decentralized global currency cryptosystem that has increased in value and popularity since its inception by Satoshi Nakamoto in 2008 [4]. As of March 2015, Bitcoin has a market capitalization of approximately 4 billion USD, market price per Bitcoin (BTC) of approximately 280 USD, and on average, 100,000 transactions daily [5]. The main purpose of Bitcoin is to enable a payment system and complete digital money that is secure and decentralized; that is, it is a peer-to-peer network powered by its users and with no central authority. To achieve this, transactions are publicly announced and the participants agree on a single history of these transactions. The transactions 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, full control of one's own money, high availability, fast international payments, zero or low fees, protected identity, and privacy [6].

 ISBN: 978-1-61208-407-7

C. Bitcoin as an Infrastructure

The current 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 ideas are to define correspondence between the roles issued by organizations and the users and to employ a challenge-response authentication protocol that will be used for verifying if a user really has an asserted role.

Bitcoin's protocol and cryptography make the proposed system suitable for the trans-organizational utilization and authentication of roles, and furthermore, allow flexible role management operations, such as the endorsement and management of roles, with relatively small realization cost.

The rest of this paper is organized as follows. Section II introduces RBAC and the different models associated with it. Section III discusses the Bitcoin protocol to show the security it provides. Section IV presents the structure and procedures of the proposed framework. Section V presents the role management features of the system. Section VI provides the conclusion and future work.

II. Models for the Role-Based Access Control

Among the many technical 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 may be related to this study, but they are excluded from the scope of our discussion. To clarify the position of our 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. This suggests that the service-providing entity is not always allowed access to the user-role assignment, and thus, it needs to devise an alternative means to confirm if an unknown guest has a certain role or not. 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. Furthermore, 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 $\mathrm{UA}_{o_i} \subset U \times R_{o_i}$. Obviously, $o_i.r \in R_{o_i}$ 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 entity (organization) o_i. The service-providing organization provides the service to 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 as obvious as the test of $(o_i.r, s) \in \mathrm{SA}$ because the assignment UA_{o_i} is defined by a foreign service-providing organization.

The trans-organizational RBAC will be realistic only if the authentication of roles $(u, o_i.r) \in \mathrm{UA}_{o_i}$ is accomplished. Physical certificates, such as passports and ID-cards, have been widely used 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 widely accepted because of the cost issues for acquiring these certificates, keeping related keys secure, and maintaining a public-key infrastructure (PKI) [7][8] that should be always available. 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 only include 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 necessitates several users with the same role to have an identical secret key, which is not favorable from a security viewpoint. Other comparable systems include decentralized multi-authority systems for attribute-based encryption (MA-ABE) and attribute-based signatures (MA-ABS) [10][11]. The MA-ABE in [10] is decentralized but requires a trusted setup of common reference parameters. The MA-ABS in [11] is also decentralized and does not necessitate a trusted setup, but it requires the setting of a public parameter for a prime order bilinear group and hash functions. Even though these schemes are decentralized, all users and organizations must agree on the parameters first. Confusion and implementation problems may arise if several communities use different parameters, and therefore, such systems must be initiated by somebody who has strong leadership, grand design, and sufficient financial power for implementation. Consequently, a scheme for a role authentication mechanism that is secure, practical, and easy to set up has yet to be established.

 ISBN: 978-1-61208-407-7

III. THE BITCOIN PROTOCOL

Bitcoin is a collection of cryptographic protocols that allow secure online transactions between users [12][13]. These users own digital wallets that handle the creation and storage of private keys and the corresponding public *Bitcoin addresses*. A user can send a certain value or amount of 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 then recorded in a global public ledger called the *blockchain*. The validation of transactions requires some amount of computation, and the miner who succeeds in validating these transactions is rewarded or compensated with Bitcoins and the transactions fees for his/her efforts. Validated transactions cannot be altered unless an attacker has computation power that overwhelms the total computation powers possessed by all other miners.

A. Bitcoin Addresses

A Bitcoin address is 160-bit hash of the 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. The private key is generated first, and then the public key is derived from the private key so that each private key corresponds to only one 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 the private and public keypairs and the corresponding Bitcoin addresses (some wallet services create new Bitcoin addresses deterministically, i.e., using a seed). The users should backup and secure the private keys (or the wallet data file) because these private keys are needed to access the Bitcoins stored in the corresponding Bitcoin addresses. Each Bitcoin address contains a built-in checksum, and thus mistyped addresses can be detected. However, Bitcoins sent to a well-formed mistyped address with no owner are lost forever. Similarly, Bitcoins stored in Bitcoin addresses are lost if the owner loses control of the wallet file or the corresponding private keys.

A Bitcoin address is in the form of random numbers and letters, e.g., 19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6, and 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.

B. 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 the following information: transaction ID (used to identify the transaction in the blockchain); list of input addresses (addresses from which Bitcoins are transferred and should be outputs of previous transactions); 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 (input addresses) with the corresponding private keys. Think of a Bitcoin address as a transparent vault where anyone can see or know the amount of Bitcoins inside, but only the one who has the key (private key) can spend or transfer the Bitcoins inside that vault.

C. Blocks and the blockchain

Transactions are grouped together in blocks and then recorded in the global 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, and thus, 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 Figure 1. The entire blockchain and every transaction included in the blocks can be downloaded or viewed online using a blockchain browser.

D. 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 hashing 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 lower 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.

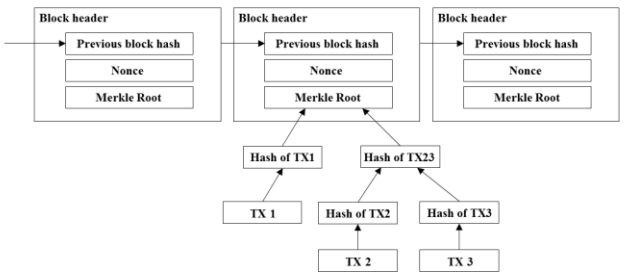

Figure 1. A simplified representation of the Bitcoin blockchain.

 ISBN: 978-1-61208-407-7

The difficulty target is expressed as the difficulty on creating the current block compared to generating the first block and is determined as follows:

$$\text{difficulty} = \frac{\text{difficulty\_1\_target}}{\text{current\_target}} \quad (1)$$

As of writing, the difficulty is 46,684,376,316.86 (the probability of each hash to be a valid solution is less than 10^{-20}) [5][14]. 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 approximately the same time, consequently forking the blockchain and dividing the nodes to work on different versions of 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 (the chain with the most work put into it), and thus, in case of a fork, the shorter chain is orphaned.

The miner who solves the block is awarded with newly "minted" Bitcoins (currently at 25 BTC) and the 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 inherently 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.

E. Attacks on the Bitcoin Network

Some strategies, both theoretically and in practice, have been devised to attack the security of the Bitcoin protocol and possibly put it in danger. Some 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 [15], the mining cartel attack [16], selfish mining [17], block withholding attack [18], and hardware attacks. These attacks are designed to infiltrate factors outside the blockchain, targeting the client side and stealing Bitcoins from them (pool and wallet infiltration). These attacks mainly aim to steal Bitcoins and/or gain higher rewards and not to modify transactions or the blockchain. Therefore, Bitcoin's security remains intact and "backed by math". For our purposes, the transactions between organizations and users will remain secure and unmodifiable because they are recorded in the blockchain.

IV. Proposed Scheme

A. 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 (details below). 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 and explanatory purposes, the role-issuing organizations will be differentiated from the users).

B. Procedures

Figure 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.

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 $1 \leq i \leq n$.

 ISBN: 978-1-61208-407-7

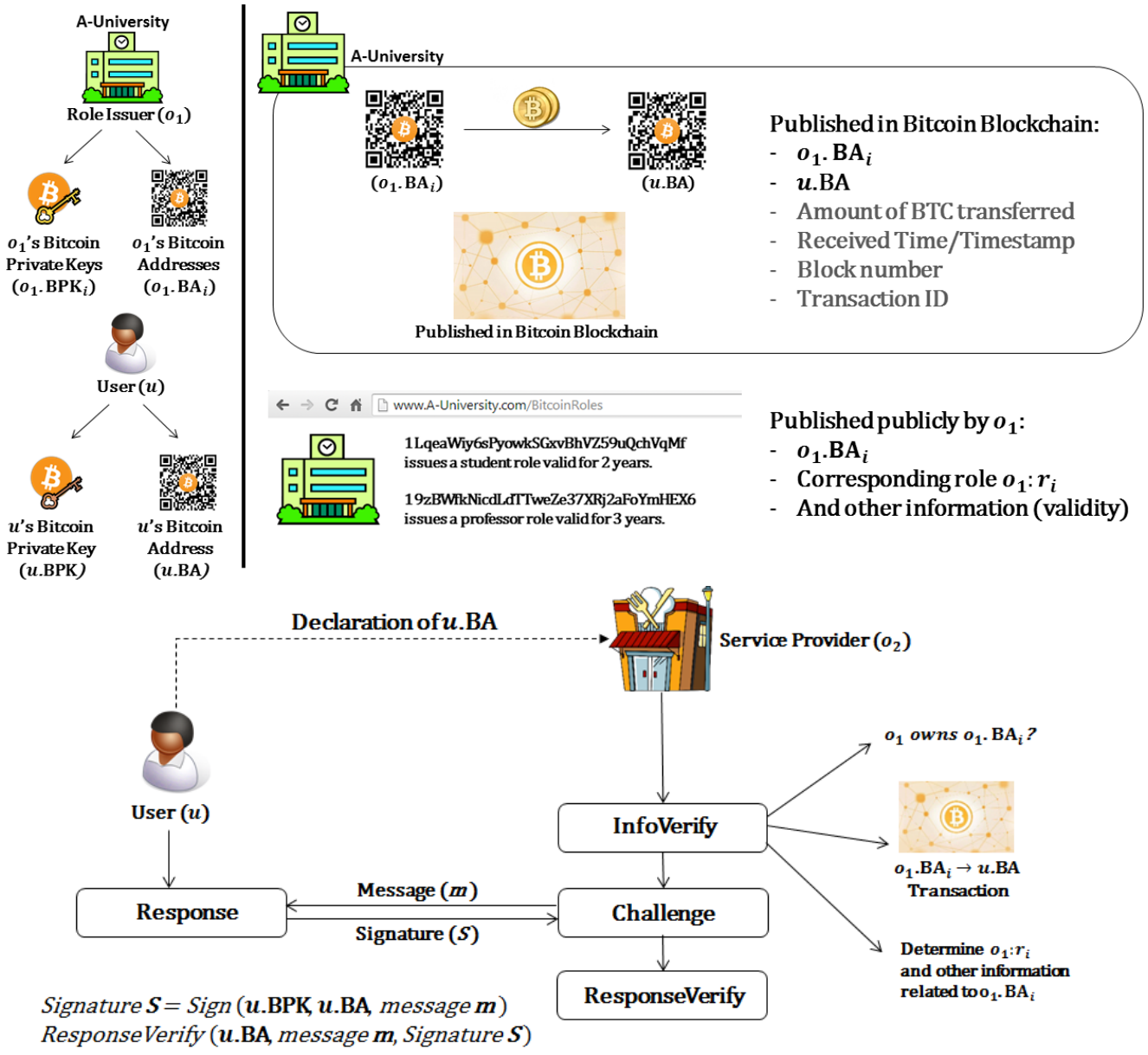

Figure 2. Overview of the proposed structure.

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 Bitcoin 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.

2) Creating the role-issuer and user connection

The organization o_1 creates a simple Bitcoin transaction using o_1.BA$_i$ 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 = 0.03 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 $o_1.\mathrm{BA}_i$, u.BA, the amount of BTC transferred, transaction ID, block number, received time, and the time it was included in the block.

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 $o_1{:}r_i$ 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 $o_1{:}r_i$ from o_1. Then o_2 will (i) determine the Bitcoin address $o_1.\mathrm{BA}_i$ that is associated with the role $o_1{:}r_i$, (ii) confirm the existence of the transaction from $o_1.\mathrm{BA}_i$ to u.BA, and (iii) verify if u is the genuine owner of u.BA. The Bitcoin address $o_1.\mathrm{BA}_i$ 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 program similar to that. Steps (i) and (ii) assure o_2 that the role $o_1{:}r_i$ and other related information associated with $o_1.\mathrm{BA}_i$ 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) 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.BPK, u.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 *ResponseVerify*(u.BA, m, S). Examples of signing/verifying a message to prove ownership of a Bitcoin address are shown in Figure 3.

Remark that o_2 can confirm if u has access to the role $o_1{:}r_i$ without querying o_1, and that u has little chance to disguise his/her role.

V. ROLE MANAGEMENT

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

A. 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:

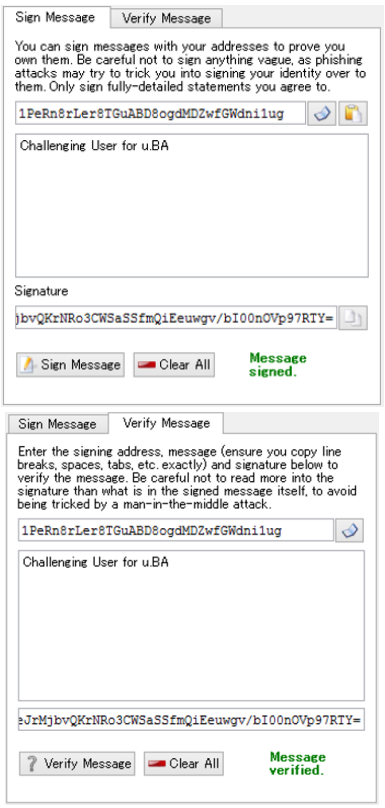

Figure 3. The signing (top) and verifying (bottom) of a message features of a Bitcoin wallet.

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 "19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6 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. It should be noted that this public note is a relatively new feature offered by an online wallet (blockchain.info) [19] and is not part of the Bitcoin protocol.

 ISBN: 978-1-61208-407-7

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. The theft/loss of keys remains a risk, but such risk also exists for ID-cards used in the real world.

B. 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, a role-issuing organization can create a revocation list containing these addresses in the media where it publishes the Bitcoin addresses it owns.

C. 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 or incentives to steal the private key (even if an attacker can steal the private keys, the addresses will not contain large amounts of Bitcoins to steal).

D. 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. One possible example is the endorsement of another person. 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 or referred by a current member. This mechanism can be realized by extending the proposed system.

Based on the system shown in Figure 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.

E. Trusted Timestamping as Proof of Validity or Expiration Dates

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) [20].

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 (the time a transaction is included in the blockchain) [21][22]. 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 (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. This timestamping scheme is decentralized and is not controlled by a single authority; it can easily be accessed over the Internet and is always available; it is anonymous (one's identity, the data being given a timestamp, or even the fact that one wants something to be given a timestamp are kept secret); it is relatively low cost; the timestamp is accurate; and it is secure.

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.

 ISBN: 978-1-61208-407-7

VI. Conclusion and Future Work

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 or validities in the created roles. Future research will focus on the realization and quantitative analysis of a more comprehensive hierarchical, multi-authority, and attribute-based system, which can offer additional role management features, such as role transfer and access policies in terms of Boolean formulas, and on the development of a prototype for easier use and interoperability.

References

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

[2] S. Farrell and R. Housley, "An internet attribute certificate profile or authorization," RFC 3281, 2002.

[3] Internet2, "The Shibboleth System," accessible online: http://shibboleth.internet2.edu/ [accessed: 2015-03-05].

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, URL: https://bitcoin.org/bitcoin.pdf [accessed: 2015-03-05].

[5] Blockchain Info, "Currency Stats," URL: https://blockchain.info/stats [accessed: 2015-03-05].

[6] Bitcoin.org, "Bitcon for Individuals," URL: https://bitcoin.org/en/bitcoin-for-individuals [accessed: 2015-03-05].

[7] C. M. Ellison et al., "SPKI certificate theory," RFC 2693, 1999.

[8] P. Gutmann, "Simplifying public key management," IEEE Computer, vol. 37, number 2, 2004, pp. 101–103.

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

[10] A. B. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," EUROCRYPT, 2011, pp. 568–588.

[11] T. Okamoto and K. Takashima, "Decentralized Attribute-Based Signatures," Public-Key Cryptography-PKC, 2013, pp. 125–142.

[12] Bitcoin Wiki, "Protocol Specification," URL: https://en.bitcoin.it/wiki/Protocol_specification [accessed: 2015-03-05].

[13] Bitcoin.org, "Bitcon Developer Guide," URL: https://bitcoin.org/en/developer-guide#block-chain [accessed: 2015-03-05].

[14] Blockexplorer, URL: http://blockexplorer.com/q/probability [accessed: 2015-03-05].

[15] M. Rosenfeld, "Analysis of Bitcoin Pooled Mining Reward Systems," 2011, URL: https://bitcoil.co.il/pool_analysis.pdf [accessed: 2015-03-05].

[16] RHorning, "Mining cartel attack," Bitcoin Forum, 2010, URL: https://bitcointalk.org/index.php?topic=2227.0 [accessed: 2015-03-05].

[17] I. Eyal and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," Financial Cryptography and Data Security, 2014.

[18] N. T. Courtois and L. Bahack, "On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency," CoRR, 2014, URL: http://arxiv.org/pdf/1402.1718v5.pdf [accessed: 2015-03-05].

[19] Blockchain Info, "Blockchain Website FAQ," URL: https://blockchain.info/wallet/website-faq [accessed: 2015-03-05] .

[20] SSL4NET, "Trusted Timestamping," URL: http://www.ssl4net.com/technology/trusted-timestamping [accessed: 2015-03-05].

[21] J. Klusácek, "Trusted Timestamp in Cryptocurrency Block Chain," Student EEICT, 2014.

[22] BTProof, "Trusted timestamping on the Bitcoin blockchain," URL: https://www.btproof.com/ [accessed: 2015-03-05].

 ISBN: 978-1-61208-407-7