

## Regular Paper

# The Bitcoin Network as Platform for Trans-organizational Attribute Authentication

JASON PAUL CRUZ<sup>1,a)</sup> YUICHI KAJI<sup>1,b)</sup>

Received: February 2, 2015, Revised: March 19, 2015,  
Accepted: May 9, 2015

**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 and 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 and 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 a person 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.

### 1.2 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 block includes the hash of the previous block to

<sup>1</sup> Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan  
<sup>a)</sup> jpmcruz@ymail.com  
<sup>b)</sup> kaji@is.naist.jp

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].

### 1.3 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 2 introduces RBAC and the different models associated with it. Section 3 discusses the Bitcoin protocol to show the security it provides. Section 4 presents the structure and procedures of the proposed framework. Section 5 presents the role management features of the system. Section 6 provides the conclusion and future work.

## 2. 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  $UA \subset U \times R$ , and a role-service assignment  $SA \subset R \times S$ . A user  $u$  is eligible to access a service  $s$  if and only if there is a role  $r$  such that  $(u, r) \in UA$  and  $(r, s) \in SA$ . 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,  $R = R_{o_1} \cup \dots \cup R_{o_n}$ , where  $o_1, \dots, o_n \in O$  and  $R_{o_i} \cap R_{o_j} = \phi$  if  $i \neq j$ . To make the relation among roles and organizations explicit, a role  $r$  in  $R_{o_1}$  is written as  $o_1.r$ . Similarly, the user-role assignment  $UA$  is partitioned into disjoint subsets;  $UA = UA_{o_1} \cup \dots \cup UA_{o_n}$ , where  $UA_{o_i} \subset U \times R_{o_i}$ . Obviously,  $o_1.r \in R_{o_1}$  means that the role  $o_1.r$  is managed by the organization  $o_1$  and the assignment of users to  $o_1.r$  is controlled by that organization  $o_1$ . In the trans-organizational RBAC, a user  $u$  demands a service  $s$  by asserting his/her role  $o_1.r \in R_{o_1}$  that has been provided by a role-providing entity (organization)  $o_1$ . The service-providing organization provides the service to the user if and only if  $(u, o_1.r) \in UA_{o_1}$  and  $(o_1.r, s) \in SA$ . Note that the test of  $(o_1.r, s) \in SA$  is easy for the service-providing organization because the assignment  $SA$  is defined by the organization itself. On the other hand, the test of  $(u, o_1.r) \in UA_{o_1}$ , which is sometimes called an *authentication*, is not as obvious as the test of  $(o_1.r, s) \in SA$  because the assignment  $UA_{o_1}$  is defined by a foreign service-providing organization.

The trans-organizational RBAC will be realistic only if the authentication of roles  $(u, o_1.r) \in UA_{o_1}$  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 Ref. [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 Ref. [10] is decentralized but requires a trusted setup of common reference parameters. The MA-ABS in Ref. [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.

### 3. 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 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 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.

#### 3.1 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 public keys and the corresponding private keys (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 a user 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., 19zBWfkNidLdTTweZe37XRj2aFoYmHEX6, 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.

#### 3.2 Transactions and Transaction Fees

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 correspond-

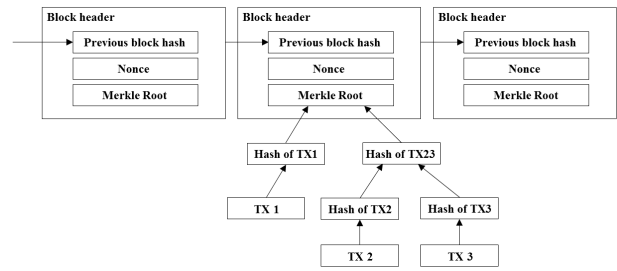


Fig. 1 A simplified representation of the Bitcoin blockchain.

ing 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.

Many transactions are typically processed without any transaction fee included if certain conditions are met [14]. However, a transaction fee of 0.0001 BTC (approximately 0.03 USD) per 1,000 bytes is required for transactions with outputs of less than 0.01 BTC, which are also known as “dust” transactions.

#### 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, 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 Fig. 1. The entire blockchain and every transaction included in the blocks can be downloaded or viewed online using a blockchain browser. An interesting feature in the Bitcoin protocol is that all transactions included in a block, independent from the amount of BTC they contain, have the same level of security, i.e., a transaction that contains 0.0001 BTC is as secure as a transaction that contains 100 BTC.

#### 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 hashing function composed of two rounds of SHA-256 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. 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], [15]. When a miner finds the correct nonce value that creates a hash value less than the target, he/she 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 (i.e., 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 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 computation 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 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 [16], the mining cartel attack [17], selfish mining [18], block withholding attack [19], 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.

## 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 (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).

### 4.2 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.

#### 4.2.1 Prerequisites

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 them available to the public. We write  $o_1.BPK_i$ ,  $o_1.BA_i$ ,  $o_1:r_i$  for the private key, Bitcoin address, and the role that is associated with the address  $o_1.BA_i$ , respectively, where  $1 \leq i \leq n$ . 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 Bitcoin addresses it owns).

Similarly, a user ( $u$ ) generates a pair of a private key  $u.BPK$



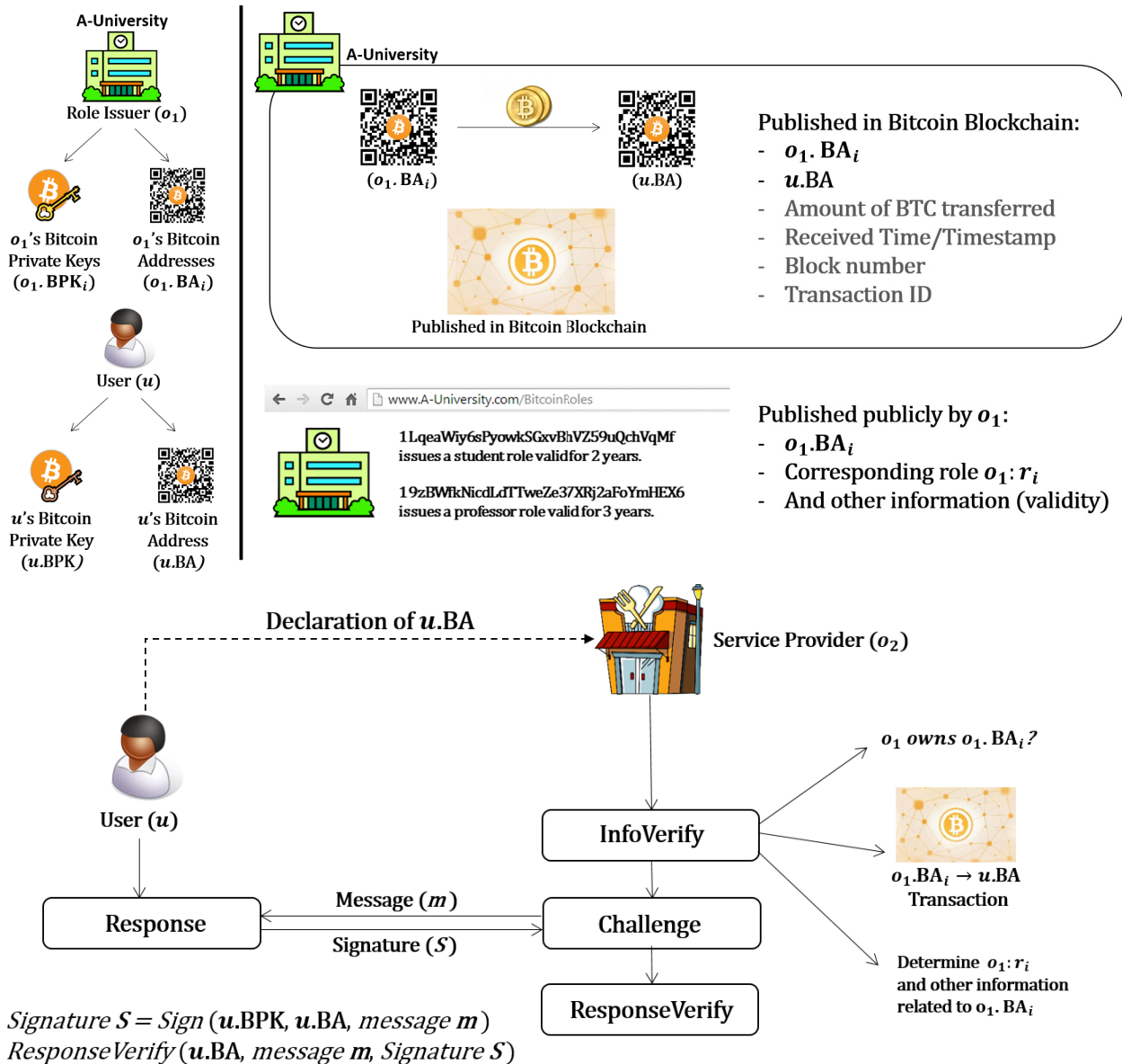


Fig. 2 Overview of the proposed structure.

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  $o_1.BA_i$  as input address and  $u.BA$  as output address. In this transaction,  $o_1$  sends an arbitrary amount of Bitcoins to  $u$ . In the proposed system, the role-issuing organizations are expected to transfer the minimum amount of BTC, which is equal to 1 satoshi or 0.00000001 BTC. This kind of transaction is considered a “dust” transaction and will incur a required transaction fee of 0.0001 BTC. Therefore, the current minimum amount of BTC that can be used for a transaction to be considered valid is 0.00010001 BTC = 0.03 USD or 3.36 YEN. 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, the time of confirmation is not vital and 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  $o_1.BA_i$ ,  $u.BA$ , 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  $o_1:r_i$  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  $o_1.BA_i$  that is associated with the role  $o_1:r_i$ , (ii) confirm the existence of the transaction from  $o_1.BA_i$  to  $u.BA$ , and (iii) verify if  $u$  is the genuine owner of  $u.BA$ . The Bitcoin address  $o_1.BA_i$  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  $o_1.BA_i$  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.

**4.2.4 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  $S = Sign(u.BPK, u.BA, m)$ , 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 Fig. 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.

**4.2.5 Comparison of Practicality and Costs**

The proposed system provides flexibility for parties who want to join or leave the system (in this subsection, parties refer to role-issuing organizations, service-providing organizations, and users, unless otherwise specified). New parties only need to follow the Bitcoin protocol to participate in the system, while role-issuing organizations who want to leave the system can simply retract or disable the media where they published the Bitcoin addresses they own. The MA-ABE [10] and MA-ABS [11] systems provide the same flexibility as the proposed scheme in terms of parties leaving or joining because they are decentralized, but PKI-based and server-based systems are not as flexible because the issuance and acquisition of certificates are not free of charge and because parties need permission from the central authority to participate.

In the proposed, MA-ABE, and MA-ABS systems, the responsibility of the authentication activities belongs to the parties, while in the PKI-based and server-based systems, the responsibility belongs to the central, single authority. Decentralized and centralized systems have different ways of handling their activities and cannot be compared directly, although it can be argued that decentralized systems are more flexible than centralized systems because centralized systems break down completely if the central authority or server becomes corrupted.

The proposed system is relatively easy to implement given that it adopts the Bitcoin network, which is completely online and is

run by a peer-to-peer network. Bitcoin has already established a currently secure, always available, and complete system that uses common parameters that have been set and agreed upon by the Bitcoin community based on significant amount of research. It is also noted that the setup cost of the proposed system is reduced to a minimum because the Bitcoin network has already been established and is currently functioning. The MA-ABE [10] and MA-ABS [11] are relatively new technologies, and they can possibly be widely accepted in the future. However, as of writing this manuscript, the practical contribution of these techniques to existing network systems and the cost issues of their deployment are yet to be discussed. PKI-based and server-based systems are well-recognized frameworks and have been utilized in many systems. However, it is often pointed out that, in these frameworks, the setup and service/labor costs become more expensive as the number of users increases. For example, a 5,000-user PKI infrastructure can cost approximately 200,000 USD annually, with the initial setup cost amounting to 10,000 USD [20], [21]. These costs can also apply for other systems that rely on a central authority, including Identity Providers (IdPs) which authenticate users on the Internet by using security tokens.

To avoid confusion, we remark that the computational cost and the participation to the peer-to-peer Bitcoin network are not essential for the parties participating in the proposed scheme. Heavy computations are only necessary during the process of mining Bitcoins, but the parties do not need to mine Bitcoins to participate in the system. Parties only need to generate Bitcoin addresses, create Bitcoin transactions for role issuance, and perform the challenge-response protocol, all of which have reasonably small computational cost and thus can easily be applied in today’s computer systems and devices.

**5. 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.

**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:

- (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

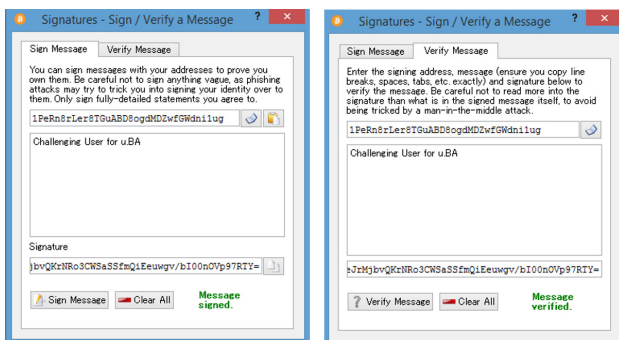


Fig. 3 The signing (left) and verifying (right) of a message features of a Bitcoin wallet.

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. It should be noted that this public note is a relatively new feature offered by an online wallet (blockchain.info) [22] and is not part of the Bitcoin protocol.

With these measures, a user 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.

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

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

### 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. 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 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 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) [23].

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) [24], [25]. 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 him/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. 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 expira-

tion 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 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] Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E.: Role-based access control models, *IEEE Computer*, Vol.29, No.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, available from (<http://shibboleth.internet2.edu>) (accessed 2015-01-20).

[4] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, available from (<https://bitcoin.org/bitcoin.pdf>) (accessed 2015-01-20).

[5] Blockchain Info: Currency Stats, available from (<https://bitcoin.org/bitcoin.pdf>) (accessed 2015-03-05).

[6] Bitcoin.org: Bitcon for Individuals, available from (<https://bitcoin.org/en/bitcoin-for-individuals>) (accessed 2015-01-20).

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

[8] Gutmann, P.: Simplifying public key management, *IEEE Computer*, Vol.37, No.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] Lewko, A.B. and Waters, B.: Decentralizing Attribute-Based Encryption, *EUROCRYPT 2011*, pp.568–588 (2011).

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

[12] Bitcoin Wiki: Protocol Specification, available from ([https://en.bitcoin.it/wiki/Protocol\\_specification](https://en.bitcoin.it/wiki/Protocol_specification)) (accessed 2015-01-20).

[13] Bitcoin.org: Bitcon Developer Guide, available from (<https://bitcoin.org/en/developer-guide#block-chain>) (accessed 2014-12-10).

[14] Bitcoin Wiki: Transaction fees, available from ([https://en.bitcoin.it/wiki/Transaction\\_fees](https://en.bitcoin.it/wiki/Transaction_fees)) (accessed 2015-03-05).

[15] Blockexplorer: available from (<http://blockexplorer.com/q/probability>) (accessed 2015-03-05).

[16] Rosenfeld, M.: Analysis of Bitcoin Pooled Mining Reward Systems (2011), available from ([https://bitcoil.co.il/pool\\_analysis.pdf](https://bitcoil.co.il/pool_analysis.pdf)) (accessed 2014-12-10).

[17] RHorning: Mining cartel attack, Bitcoin Forum (2010), available from (<https://bitcointalk.org/index.php?topic=2227.0>) (accessed 2014-12-10).

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

[19] Courtois, N.T. and Bahack, L.: On Subversive Miner Strategies and

Block Withholding Attack in Bitcoin Digital Currency, *CoRR* (2014), available from (<http://arxiv.org/pdf/1402.1718v5.pdf>) (accessed 2014-12-10).

[20] VeriSign: Total Cost of Ownership for Public Key Infrastructure, available from ([http://www.verisign.com/stellent/groups/public/documents/white\\_paper/005321.pdf](http://www.verisign.com/stellent/groups/public/documents/white_paper/005321.pdf)) (accessed 2015-03-05).

[21] TC TrustCenter: The Costs of Managed PKI, available from ([http://www.safebiopharma.org/VendorPartnerFiles/ChosenSecurity/The%20Costs%20of%20Managed%20PKI%20Whitepaper\\_Web%20and%20QuickStart\\_Feb08.pdf](http://www.safebiopharma.org/VendorPartnerFiles/ChosenSecurity/The%20Costs%20of%20Managed%20PKI%20Whitepaper_Web%20and%20QuickStart_Feb08.pdf)) (accessed 2015-03-05).

[22] Blockchain Info: Blockchain Website FAQ, available from (<https://blockchain.info/wallet/website-faq>) (accessed 2015-01-20).

[23] SSL4NET: Trusted Timestamping, available from (<http://www.ssl4net.com/technology/trusted-timestamping>) (accessed 2015-01-20).

[24] Klusáček, J.: Trusted Timestamp in Cryptocurrency Block Chain, *Student EEICT* (2014).

[25] BTProof: Trusted timestamping on the Bitcoin blockchain, available from (<https://www.btproof.com>) (accessed 2015-01-20).



**Jason Paul Cruz** was born in Manila, Philippines, on May 8, 1988. He received his B.S. degree in Electronics and Communications Engineering and M.S. degree in Electronics Engineering from the Ateneo de Manila University, Quezon City, Philippines, in 2009 and 2011, respectively. He is currently a Ph.D. candidate in the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His current research interests include role-based access control, blockchain technology (Bitcoin and Ethereum), hash functions and algorithms, and Android programming.



**Yuichi Kaji** was born in Osaka, Japan, on December 23, 1968. He received his B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1992 and 1994, respectively. In 1994, he joined Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. In 2003 and 2004, he visited the University of California Davis and the University of Hawaii at Manoa as a visiting researcher. His current research interests include the theory of error correcting codes, fundamental techniques for information security, and the theory of automata and rewriting systems. He is a member of IPSJ and IEEE.