

# On the Evidence based TTP Cluster Election in the P2P Network<sup>1</sup>

Xu Yang, T. C. Lam and Jyh-Charn Liu<sup>2</sup>  
{xuyang,brianlam}@tamu.edu, liu@cs.tamu.edu  
Department of Computer Science  
Texas A&M University  
College Station, TX 77843-3112

**Technical Report 2003-8-1**

**August 4, 2003**

## Abstract

In this paper we proposed an e-coin based role election scheme for peer-to-peer (P2P) systems. A node in a P2P group can be elected as an application server, or a trusted third party (TTP) server, but not both. The attributes of the election process and the outcomes are mapped to the Ferguson's e-coin scheme so that undeniable evidences can be generated even when the P2P nodes remain anonymous. Nodes can nominate and cast their votes without revealing their identities, but violation of voting rules will subject them to disclosure of such misconducts, and their true identities, which were to be tied into the evidences in encrypted forms. Our scheme also supports the merging of multiple groups, to ensure that when a node is involved in multiple groups, it cannot compromise the mutual exclusion requirement of the role choices in the newly merged group.

*Index terms – peer-to-peer, trusted third party, mutual exclusion, nomination and election, e-coin, grouping.*

---

<sup>1</sup> This work is supported in part by an NSF ITR grant, EIA-0081761, and by a TAMU-CONACYT grant

<sup>2</sup> Correspondence author

## 1. INTRODUCTION

The peer-to-peer (P2P) networking model is ideal for applications in which the computing nodes do not belong to the same administration entity group together to provide computing services. P2P collaboration, although highly desirable for the mission-based partnership, is subject to cheating (impersonated identity, repudiation of activities, *etc.*) It is useful that the computing nodes can form (P2P) groups to either provide services, or serve as the *trusted third party* (TTP) of the service providers, so that group activities can be mediated by the TTP nodes to enforce certain service attributes, *e.g.*, fair exchange [1], transaction non-repudiation [2].

In the traditional networking design, the TTP is often considered a special class of computing nodes that run on some “well known” servers, *e.g.*, the certificate authority, domain controllers, *etc.* These statically defined TTP nodes are inflexible for P2P networks, and they represent performance and reliability bottlenecks. In this paper, we propose a role election scheme for the P2P peers to elect group members to serve as application servers (AS) or TTP servers (TS), but not both, for each *transaction group*. The election process is cast into undeniable, cryptographic evidence based on which the P2P peers cannot cheat on their choices.

A transaction group is a collection of TTP servers and application servers that work together to deliver a certain type of services, or “transactions.” In the election process, the peer nodes in the P2P network may remain anonymous to each other so that the true identity of the involved parties may remain sealed unless certain administrative policies are violated. For instance, an anonymous node may choose to act as a TTP node which listens to a multicast channel between some application servers. When a dispute needs to be resolved, the TTP node can disclose its role and present its opinions based on the information that it receives from the multicast channel.

Application anonymity in a P2P network could be achieved by using pseudonyms [3]-[8]. On the other hand, a node could abuse its anonymity by using multiple pseudonyms which make it impossible to determine whether or not the node is abusing its privileges by using different (pseudonyms) identities. The anonymous role election process (and its resulting transaction group) must be publicly verifiable, so that the authenticity of the node identity, the integrity of the election decisions, and the role choices can be verified based on the undeniable *election evidence*, without any centralized overseeing. Based on such evidence, a *dual-role commitment* where a transaction server pretends to be a TTP, so that it could fool others and act as its own TTP, can be detected and identified. The group should

be able to detect the offender without violating the privacy of other honest participants.

In this paper, we propose an (undeniable) evidence based TTP cluster election system in a P2P network using Ferguson’s single-term off-line e-coin [9]. The baseline e-coin scheme must be properly modified to meet several requirements. One concern is the unlinkable property, *i.e.*, the impossibility to recognize if two pieces of election evidence (spent coins) are created from the same source node, must be modified so that a node cannot use multiple voting tokens to mimic the distinct entities in the same transaction group, keeping the dual-role commitment and the over-voting undetected. Our first design is to incorporate the *conditional evidence linkability* into Ferguson’s coin based evidence so that one can recognize if two pieces of election evidence (from distinct voting tokens) are generated by the same node in the same transaction group. The nodes in different transaction groups cannot be recognized without the credential authority.

The second design is an *e-coin binding* technique so that the TS-AS *role choice* can be cryptographically tied with the election evidence. With the above constructs, a node cannot repudiate its role choice once it is selected and committed.

As a result, voting policy violations such as dual-role commitment and over-voting can be detected through public verifications. Votes cast by different nodes for the same anonymous node can be recognized and aggregated to form the *voting-sum*. Such violation detection and voting-sum computations from the election evidence are essential references for the final transaction group membership decision.

The TTP election and the political election share some common needs, but they also have very different security requirements. The e-coin crypto system is also employed in the political election [10] based on its *one-time anonymous proxy signature* property [11]. A voter casts a vote for the candidate by giving a proxy signature (spending) using a *voting token* (coin) authorized by the *credential authority* (the bank) without revealing its identity. The resulting proxy signature is an undeniable evidence of the voting action from the signing node. Based on the e-coin double-spending detection mechanism, the voter can be detected and identified after the fact if it used a single voting token to vote more than once, or  $N$  times (the total coin value) if a divisible coin [12] was used. Unauthorized multiple votes exceeding a certain voting limit is called *over-voting*.

## 2. SYSTEM MODEL

The evidence based TTP cluster election in a P2P network is a specific instance for the broader issue of *role exclusive group elections*. Our scheme can easily be

generalized to support elections of multiple excluded roles. In our system, each peer node can be either a TS or AS in a transaction group, but not both. Each node nominates and casts votes, which are represented by e-coin based election evidences, for other nodes to join the transaction group. The election evidences record the encrypted identities of the participating nodes, their committed roles, and the number of votes cast for them, *etc.* These records are the references directing the final transaction group membership decision.

Our main focus is the generation of an undeniable evidence system for the group formation. The integrity and the authenticity of the evidence can be verified without the intervention of the credential authority, *i.e.*, the bank in the conventional e-cash literature, while possible offences can be addressed and systematically examined after the fact. The application specific membership decision criteria and the actual interactions within the transaction after the group formation are not our concerns.

The system architecture of our proposed scheme is depicted in Figure 1, which also depicts an example on how  $U_1$ , being a TS, initiates the process of nominating  $U_2$ ,  $U_3$  and  $U_4$  to join the transaction group (and to serve as an AS or TS.) Details of how to manage the evidence system will be discussed shortly, after we give more explanation to the problem in hand.

The credential authority is responsible for the initial authentication of the peer nodes and the distribution of the *voting tokens* before they are authorized to participate in any election. Then, the nodes can use these voting tokens to generate the election evidence during the election process without the intervention of the credential authority. The credential authority is also responsible for the anonymity revocation from the collected election evidence if necessary.

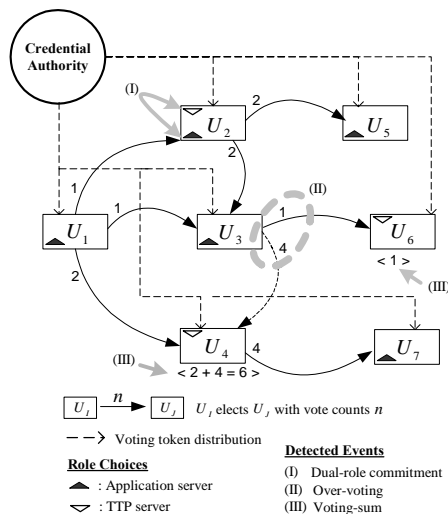


Figure 1. System architecture.

Throughout this paper, we assume that a voting token is associated with a special binary tree signed by the credential authority for the multiple-vote purpose.

Election evidences generated from the interactions between voting members form an *evidence chain*. Each evidence piece on the evidence chain carries the previous election records processed by the traversed nodes. To participate voting, each node makes nomination of the candidate and casts a certain number of votes for the nominee within its granted rights (by the CA.)

To initiate an election, a node uses its voting token to generate an evidence piece with its encrypted identity, its role choice (TS/AS) and an assigned value that stands for its *vote count*, *etc.*, to nominate and cast its votes for another node to join the transaction group. At this stage, this evidence piece is the only component of the evidence chain. The nominated node can further nominate and vote for other nodes by passing its evidence chain along with its newly generated evidence piece. A new evidence chain is formed by the appending of the new evidence piece to the received evidence chain(s).

When a node  $U_x$  uses a single voting token to generate evidence pieces for authorized multiple votes to  $U_y$  and to  $U_z$ , the traversing path of the evidence chain is split into two. The two new evidence chains in  $U_y$  and  $U_z$  hold the same past election records except their corresponding last pieces, which record the current votes to  $U_y$  and  $U_z$  respectively. If  $U_y$  and  $U_z$  cast votes for  $U_x$ , then  $U_x$  can merge the two evidence chains, so that the election records inside the two chains can be transferred in a single chain in the subsequent nominations and votes. Under normal circumstances, the election evidence does not compromise the anonymity of participating nodes. For better privacy, a node should have the freedom in the participating in two transaction groups, but in the mean time one must prevent a node from abusing its privacy privilege in the election process.

Dual-role commitment and the over-voting are the two types of misconduct that are of great concern. An example of the voting violation is depicted in Figure 2, A dual-role commitment occurs if a single node ( $U_3$ ) serves as both the TS ( $e_7$ ) and the AS ( $e_3$ ) in the same transaction group. An over-voting occurs if a single node ( $U_5$ ) casts its votes for multiple nodes ( $e_5$  and  $e_8$ ) of the same transaction group, but the total vote counts exceed its granted voting quota. Both violations, if undetected, impair the fairness of the election.

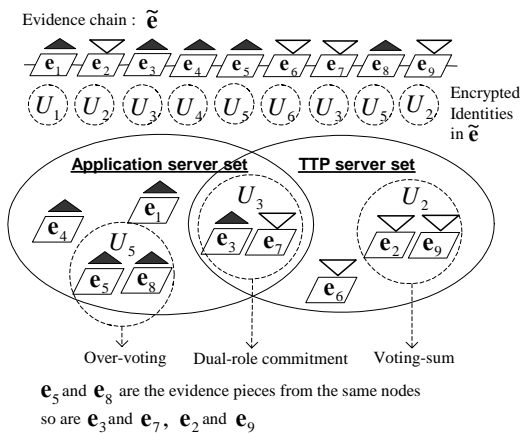


Figure 2. Identifying the same node from distinct evidence pieces in a transaction group.

One of our key contributions is to restrict the unconditional anonymous privilege, so that a single node cannot forge two distinct entities to offend the rules without being detected. This ability to recognize the same node in the same transaction group also helps us to aggregate the votes (voting-sum) cast for the same entity ( $U_2$ ) from different sources of the anonymous evidence pieces.

In summary, the construct of our election evidence achieves the following goals simultaneously:

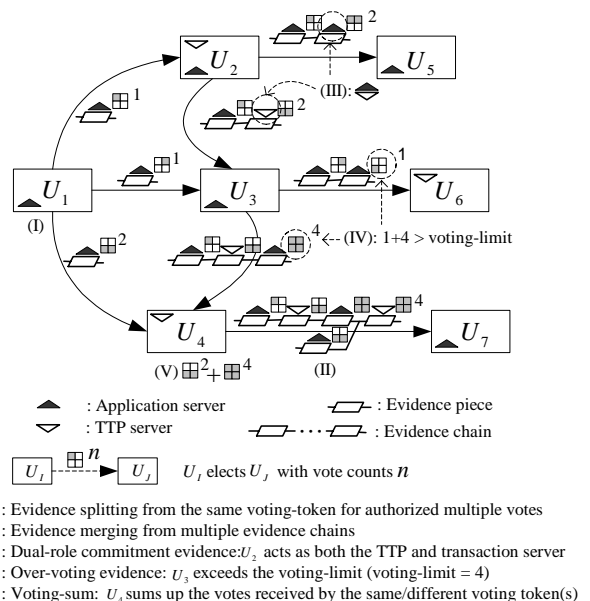
- Non-repudiation: a node cannot deny the election decisions that it made once the decisions are committed into the election evidence.
- Revocable anonymity: identities of participating nodes are concealed in the election evidence. Yet, the encrypted identity of the culprit can be revoked in violation of the election rules.
- Authenticity: a (anonymous) node that generates election evidence can be publicly verified.
- Integrity: the election records inside the election evidence can be verified to be intact even that they are passed along multiple anonymous nodes.
- Off-line (distributed) computation: the election (and its evidence generation) between the peer nodes is decentralized, without the credential authority's participation.
- Conditional evidence linkability: evidence pieces of the same transaction group can be recognized to be generated by the same node, while those of different transaction groups cannot be recognized without the credential authority.

The interactions between nodes to create evidence chains for the scenario depicted in Figure 1 are given in Figure 3. In this example, we assume that the voting limit for each node is four. To initiate the election,  $U_1$ ,

as a TS, nominated  $U_2$ ,  $U_3$  and  $U_4$  to join the transaction group.  $U_1$  used a single voting token to generate three evidence pieces, with one vote for each of  $U_2$  and  $U_3$  as AS and two votes for  $U_4$  as a TS. Three evidence chains, composed of one evidence piece each, were passed to  $U_2$ ,  $U_3$  and  $U_4$ . The diverging of evidence chains is regarded as evidence splitting, denoted by (I) in the figure. With the received evidence chains,  $U_2$ ,  $U_3$  and  $U_4$  could further nominate and vote other nodes to join the transaction group.

A dual-role commitment is illustrated by the following example. When  $U_2$  cast the votes for  $U_3$  and  $U_5$ , it committed different roles to each of their newly generated evidence pieces. The new evidence pieces were appended to the old evidence chain (from  $U_2$ ) to form two new evidence chains in  $U_3$  and  $U_5$  respectively. The dual-role commitment of  $U_2$  could be detected after the vote has been cast by examining the evidence chains containing these two evidence pieces, denoted by (III).

$U_3$  had two evidence chains, one from  $U_1$  and one from  $U_2$ . It cast one vote for  $U_6$  by using the evidence chain from  $U_1$ , and four votes for  $U_4$  by using the evidence chain from  $U_2$ . As a result, its total vote count was five, which exceeded the voting limit. The over-voting violation of  $U_3$  could be detected from the evidence chains containing the corresponding evidence pieces, denoted by (IV) in Figure 3.



- (I) : Evidence splitting from the same voting-token for authorized multiple votes
- (II) : Evidence merging from multiple evidence chains
- (III) : Dual-role commitment evidence:  $U_2$  acts as both the TTP and transaction server
- (IV) : Over-voting evidence:  $U_3$  exceeds the voting-limit (voting-limit = 4)
- (V) : Voting-sum:  $U_4$  sums up the votes received by the same/different voting token(s)

Figure 3. Scenario of the election process.

### 3. CONDITIONAL EVIDENCE LINKABILITY

At  $U_4$ , votes were received from multiple sources, with two votes from  $U_1$  and four votes from  $U_3$ . The resulting voting-sum for  $U_4$  was therefore six, denoted by (V). When  $U_4$  nominated and voted  $U_7$  to join the transaction group, a new evidence piece was formed. This evidence piece was appended to the two evidence chains in  $U_4$  (from  $U_1$  and  $U_3$ ) so that the two chains were merged into a single one at  $U_7$ .

Our election system generates the conditionally linkable election evidence from the voting token. The constructs of the voting token and the election piece are derived from Ferguson's single-term off-line coin [9]. The cryptographic relationship between the evidence pieces in an evidence chain is derived from the transferable coin [13][14]. An e-coin crypto system can be used for the election because of its *one-time anonymous proxy* signature properties [11]. A node can cast its votes by giving the proxy signature (spending) using the voting token (coin) authorized by the credential authority (bank) without revealing its identity. The resulting proxy signature is an undeniable election evidence for the voting action of the signing node. The node that cast votes more than once using a single voting token can be detected and identified after the fact by the double-spending detection mechanism of e-coin.

Figure 4 depicts the election evidence constructs from  $U_Y$  to  $U_Z$  based on the Ferguson's coin algorithm. Voting tokens are distributed from the credential authority to the nodes before the nodes are authorized to participate any election. The election evidence is constructed through a three-way interaction between  $U_Y$  and  $U_Z$ .

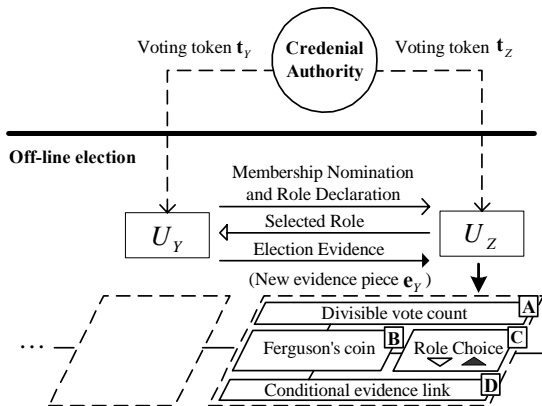


Figure 4. The three-way interaction in the election evidence construction process.

First,  $U_Y$  declares its role and sends  $U_Z$  the evidence chain as a nomination to join the transaction group.  $U_Z$  verifies that if the declared role of  $U_Y$  is consistent with

the one recorded in the received evidence chain.  $U_Z$  accepts the nomination by acknowledging  $U_Y$  with the role it would like to play in the transaction group. Lastly,  $U_Y$  sends  $U_Z$  the evidence piece, which records the transaction group id, the encrypted identity of  $U_Y$ , the vote counts it cast, their role choices, *etc.* A new evidence chain is formed in  $U_Z$  by the appending of this evidence piece to the evidence chain received from  $U_Y$ .

Ferguson's coin provides the basic framework for several privacy and security features of our evidence based election system. Security attributes, such as non-repudiation, revocable anonymity, authenticity, integrity and off-line computation, can be achieved by mapping Ferguson's solutions to our election evidence. Instances of dual-role commitment and over-voting can easily be detected by the use of a single token, for the same coin header implies the same source of the election evidence. Nevertheless, the direct mapping of Ferguson's solutions does not completely meet our system requirements.

First, Ferguson's coin is unlinkable, which implies that the evidence pieces generated by the distinct voting tokens cannot be recognized from the same source. As a result, the dual-role commitment and the over-voting cannot be detected if a node uses this trick to mimic two distinct entities. Second, Ferguson's coin is basically an authenticated random string carrying the face-value. System information, such as the role choice, is not included. Third, Ferguson's coin can be spent once only. It does not support multiple authorized votes using a single voting token.

To overcome these problems, we design several components, *i.e.*, the *(divisible) vote count*, the *role choice* and the *conditional evidence link*, on top of Ferguson's coin to create the evidence piece from the voting token, see Figure 4. The constructs of the several system primitives, the voting token, the evidence piece and the evidence chain, are based on the following assumptions on the public parameters. We assume that computations are done in modulo  $n$  in all discussions.

The RSA public key  $v$  and the public modulus  $n$  of the credential authority are known to all, where  $v$  is assumed to be a large prime. The private key  $1/v$  of the credential authority is kept secret in the credential authority. The parameter  $p$  is a large prime where  $p-1$  is a multiple of  $n$ .  $g_1, g_2, g_3$  are publicly known elements of large orders in the multiplicative group of  $Z_n^*$ .  $h_b, h_c$  are publicly known elements of order  $n$  in the multiplicative group  $Z_p^*$ .  $hash(\cdot)$  and  $H(\cdot)$  are suitable one-way hash functions mapping from an arbitrary domain to the multiplicative group  $Z_v^*$ .

A voting token is a 4-tuple  $\mathbf{t} = (\mathbf{w}, U, k, \mathbf{Q})$ , where the warrant  $\mathbf{w}$  is an integral triple  $(a, b, c)$ , the node identity  $U$  and the secret number  $k$  are integers, the secret signature  $\mathbf{Q}$  is an integral triple  $(S, T, P)$ .

The warrant is a system attribute signed by the credential authority that can be publicly verified for its authenticity. In the 4-tuple voting token, only the warrant is sent in plain-text for the election evidence generation.

A voting token is legitimate if it satisfies  $S^v = (C^U B)$ ,  $T^v = (C^k A)$ ,  $P^v = (CA)$ , where  $A = ag_1^{\text{hash}(a)}$ ,  $B = bg_2^{\text{hash}(b)}$ ,  $C = cg_3^{\text{hash}(c)}$ . We denote  $A, B, C$  as derived from  $a, b, c$  in the same way as above in the subsequent discussion. We assume that the identity  $U$  of each node and the warrant  $\mathbf{w}$  of each voting token are unique in the system.

In fact, a voting token is essentially a Ferguson's coin before payment, with the additional signature  $P$  in the similar structure of the signature  $T$  from the original scheme. Different from [9], the secret  $k$  is not randomly generated but fixed and specific to  $U$ . Another difference is that the warrant of the voting token (and hence the evidence piece generated) is traceable by the credential authority by using the RSA signature, instead of the blind signature [19], in the voting token distribution (coin withdrawal). The enhanced traceability allows the credential authority to revoke the anonymity by mapping the unique warrant  $\mathbf{w}$  to the identity  $U$ . This enhances accountability and facilitates system re-configuration through the on-line operations.

An evidence piece is a 4-tuple  $\mathbf{e} = (\mathbf{w}, \boldsymbol{\pi}, \boldsymbol{\psi}, \Omega)$ , where  $\mathbf{w}$  is the warrant,  $\boldsymbol{\pi}$  is the conditional evidence link consisting of a 4-tuple  $(x, \mathbf{r}, \hat{x}, \hat{\mathbf{r}})$ ,  $\boldsymbol{\psi}$  is the vote count consisting of a 4-tuple  $(\sigma, K^{(0)}, K^{(j)}, \mathbf{H}^{(j)})$ , and  $\Omega$  is an integer that represents the TS/AS role choice.

In  $\boldsymbol{\pi}$ , the challenge  $x$  and the group id  $\hat{x}$  are integers, while the responses  $\mathbf{r} = (r, R)$  and  $\hat{\mathbf{r}} = (\hat{r}, \hat{R})$  are integral pairs.

In  $\boldsymbol{\psi}$ , the tree signature  $\sigma$ , the tree root  $K^{(0)}$  and the tree node  $K^{(j)}$  are integers, while  $\mathbf{H}^{(j)}$  is an integer vector of length  $(\ell - 1)$ , where  $j$  is an  $\ell$ -bit binary string starting with "0". As we will show in the next section, the vote count is determined by the partial knowledge on a binary tree signed by the credential authority, and so the components in  $\boldsymbol{\psi}$  are entitled as above.

An evidence piece  $\mathbf{e}_Y$  co-generated from  $U_Y$  to  $U_Z$  (with roles  $\Omega_Y$  and  $\Omega_Z$ ) in the transaction group  $\hat{x}$  is legitimate if it satisfies  $R_Y^v = C_Y^{\hat{r}_Y} B_Y^{x_Y} (K_Y^{(j)})^{\text{hash}(\Omega_Y, \Omega_Z)}$ ,  $\hat{R}_Y^v = C_Y^{\hat{r}_Y} B_Y^{\hat{x}} A_Y$ ,  $\sigma_Y^v = \text{hash}(A_Y, B_Y, C_Y, K_Y^{(0)})$  and a series of steps to reconstruct  $K^{(0)}$  from  $(K^{(j)}, \mathbf{H}^{(j)})$

detailed in the next section. The details of the parameters in an evidence piece and their relationship are depicted in Figure 5.

An evidence piece is essentially a Ferguson's coin after payment, with the following derivations. The Ferguson's coin  $(\mathbf{w}, x, \mathbf{r})$  provides the basic construct to encrypt the identity  $U$  from the voting token to the evidence piece, which accounts for the anonymity and the authenticity properties of the election evidence.

On top of it, we cryptographically tie the role choices with the coin by an e-coin binding technique [20], where the hash value of the role choices is embedded into the response parameter of the coin, to make the committed role choice undeniable after binding. Based on the divisible coin technique [12], we add the vote count component to Ferguson's coin, so that a node can cast multiple votes using a single voting token.

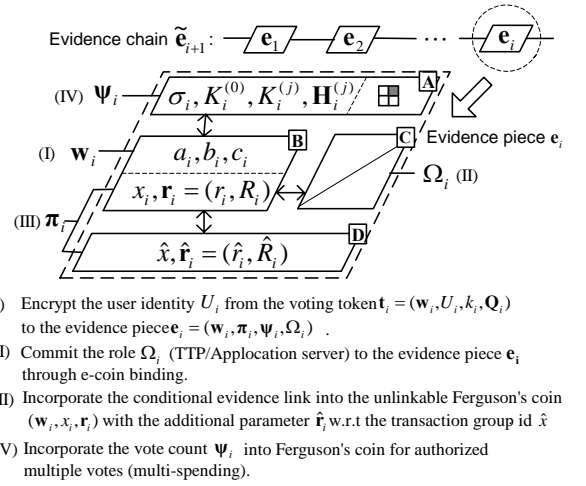


Figure 5. Details of the election evidence construct.

We incorporate the conditional evidence link to the unlinkable Ferguson's coin by adding an extra challenge-response pair  $(\hat{x}, \hat{\mathbf{r}})$  in the similar construct  $(x, \mathbf{r})$  of the original Ferguson's scheme, except that the challenge is not randomly generated, but set to be the transaction group id.

Figure 6 depicts how to achieve the conditional evidence linkability by the above construct. Here, two evidence pieces generated by distinct voting tokens  $\mathbf{t}_Y$  and  $\mathbf{t}'_Y$  can be recognized as from the source node  $U_Y$  with the unique secret number  $k_Y$  in the same transactions group  $\hat{x}$  because they have the identical response  $\hat{r}_Y = \hat{r}'_Y = U_Y \hat{x} + k_Y$ . On the other hand, for different transaction groups  $\hat{x}$  and  $\hat{x}^*$ , we cannot tell whether two evidence pieces generated by  $\mathbf{t}_Z$  and  $\mathbf{t}'_Z$  are created from the same node  $U_Z$  with the secret  $k_Z$

because their responses  $\hat{r}_z = U_z \hat{x} + k_z$  and  $\hat{r}'_z = U_z \hat{x}^* + k_z$  are distinct and appear to be irrelevant.

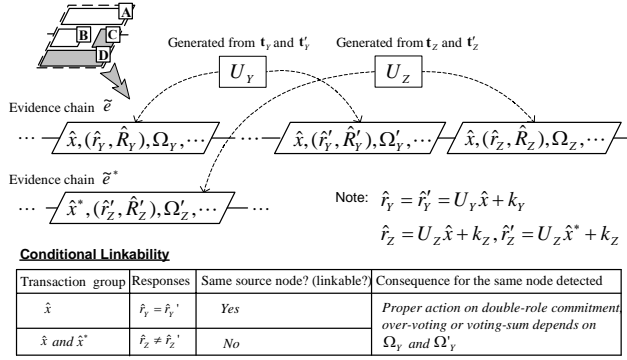


Figure 6. Conditional linkability of evidence pieces in the identical and the distinct transaction groups.

In an evidence chain, any evidence piece  $e$  originating from no other evidence piece is the *head* of the chain. The unique group id set to be  $\hat{x} = hash(\mathbf{w})$ , where  $\mathbf{w}$  comes from the head  $e$ . It ensures the election of this transaction group is initiated by a single node (with  $\mathbf{w}$ ) only.

The evidence piece  $e'$  connected to no other evidence piece is the *tail* of the chain. Each evidence chain contains a single tail only because of the unique assumption of  $\mathbf{w}'$  and the special connection construct  $x = hash(\mathbf{w}')$ . An evidence chain is legitimate if every evidence piece in it is legitimate and the chain head and tail satisfy the requirements described above.

Violations of the voting rules can be examined from the evidence pieces in an evidence chain. Violations recorded on the evidence pieces on the same evidence chain can be detected immediately when the evidence chain is received. Violations recorded on the evidence pieces on different evidence chains can be detected after the fact when the evidence chains are collected. Offenders of over-voting using the same voting token can be identified off-line with the double-spending detection mechanism of e-coin (detailed in the next section.) On the other hand, conditional evidence linkability does not lead to the identification of  $U$ . However, it is an off-line indicator for the on-line revocation of the concealed identity if necessary.

#### 4. EVIDENCE CHAIN GENERATION AND THE VOTE COUNT MANAGEMENT

In this section, we describe the detailed steps for the evidence chain generation. In particular, we give the details of how the vote count component is created and interpreted from a specially constructed binary tree [12]. The off-line culprit identification of over-voting using a single voting token will also be discussed.

The three-way interaction for evidence chain construction in Figure 4 is detailed in Figure 7. The voting tokens of  $U_Y$  and  $U_Z$  are denoted by  $\mathbf{t}_Y = (\mathbf{w}_Y, U_Y, k_Y, \mathbf{Q}_Y)$  and  $\mathbf{t}_Z = (\mathbf{w}_Z, U_Z, k_Z, \mathbf{Q}_Z)$  respectively. To nominate  $U_Z$  to join the transaction group  $\hat{x}$ ,  $U_Y$  sends to  $U_Z$  the evidence chain  $\tilde{e}_Y$ , the transaction group id  $\hat{x}$ , the warrant  $\mathbf{w}_Y$  and its declared role  $\Omega_Y$ , where  $\mathbf{w}_Y$  and  $\Omega_Y$  are used for creating  $\tilde{e}_Y$  if  $U_Y$  is nominated from other node(s). In response,  $U_Z$  verifies the legitimacy of  $\tilde{e}_Y$  (see the last section,) and that the tail of  $\tilde{e}_Y$  has the correct challenge  $x = hash(\mathbf{w}_Y)$ .  $U_Z$  accepts the nomination by sending  $U_Y$  its selected role  $\Omega_Z$  and the challenge  $x_Y \leftarrow hash(\mathbf{w}_Z)$ .

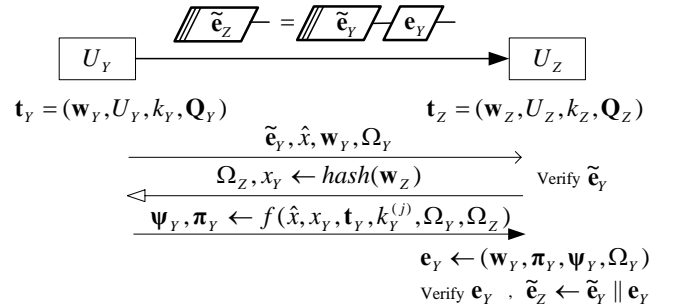


Figure 7. Detailed steps of evidence chain generation.

To cast votes for  $U_Z$ ,  $U_Y$  constructs the vote count component  $\psi_Y$  using the binary tree associated with  $\mathbf{t}_Y$ .  $U_Y$  also computes the conditional evidence link  $\pi_Y \leftarrow f(\hat{x}, x_Y, \mathbf{t}_Y, k_Y^{(j)}, \Omega_Y, \Omega_Z)$ , where  $k_Y^{(j)}$  is a parameter from the binary tree representing the vote count. The specification of the function  $f$  is to set  $\hat{r}_Y \leftarrow U_Y \hat{x} + k_Y$ ,  $r_Y \leftarrow U_Y x_Y + hash(\Omega_Y, \Omega_Z) k_Y^{(j)}$ ,  $R_Y \leftarrow S_Y^x P_Y^{hash(\Omega_Y, \Omega_Z)}$  and  $\hat{R}_Y \leftarrow S_Y^x T_Y$ .  $U_Y$  sends  $\psi_Y$  and  $\pi_Y$  to  $U_Z$  for constructing the election evidence.

$U_Z$  constructs a new evidence piece  $e_Y \leftarrow (\mathbf{w}_Y, \pi_Y, \psi_Y, \Omega_Y)$  from the received parameters and verifies its legitimacy. A new evidence chain  $\tilde{e}_Z$  is formed in  $U_Z$  for further nomination by the appending of  $e_Y$  to  $\tilde{e}_Y$ . In Figure 7, and in the rest of this paper, the symbol “||” stands for the concatenation of two entities.

One essential step to protect the integrity of an evidence chain is  $x_Y \leftarrow hash(\mathbf{w}_Z)$  in the second step of the three-way interaction, which is derived from the transferable coin technique [13][14]. The construct of the evidence chain ensures that, 1) each evidence piece is unforgeable, based on the unforgeable Ferguson’s coin, 2) the head of the chain cannot be replaced as only the

head with the correct warrant can derive the correct transaction group id, and 3) the subsequent evidence pieces in the chain cannot be re-ordered or truncated (if we do not consider evidence chain merging) based on the one-way and collision resistant relationship between the adjacent evidence pieces. The integrity of the evidence chain is crucial for the non-repudiation and the authenticity properties of our system.

The three-way interaction in Figure 7 only demonstrates how a single evidence chain is generated by another single evidence chain. Evidence chain merging is the operation to combine multiple evidence chains in the same transaction group into a single one as depicted in Figure 8(b). This can be done by using a single warrant to generate an identical challenge to receive multiple evidence chains. As a result, these evidence chains will be delivered in a bunch with a single tail in the subsequent nomination and voting processes.

We note that the merging of evidence chains does not compress its size, see Figure 8(a). It carries the same amount of election records as the evidence chains without merging. However, by consolidating different evidence chains into a single one does simplify the management of the evidence chains in the future voting, and hence lowers the computation overhead.

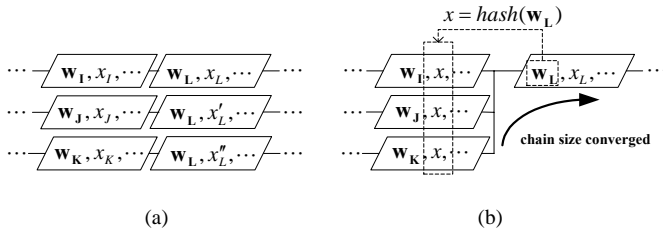


Figure 8. Evidence chain merging in the same group.

Note that the simplified construct in Figure 8(b) is vulnerable to a truncation attack [15], where multiple election records in the evidence chain (with merging) are possibly removed from the malicious traversed node without being detected. We can solve this problem by incorporating branch number in the e-coin binding process proposed in [20]. Details of this technique are not discussed in this paper due to space limit.

An evidence chain can be split, as the scenario depicted in Figure 9. An evidence chain is split when a node uses a single voting token to cast authorized votes for different nodes. In Figure 9, the evidence chain  $\tilde{e}_x$  from  $U_x$  is split into  $\tilde{e}_A$ ,  $\tilde{e}_B$  and  $\tilde{e}_C$ , where two votes are cast for  $U_A$  and one vote is cast for each of  $U_B$  and  $U_C$ . The vote count is determined by the partial knowledge revealed from a special binary tree (at the left of the figure) derived from a divisible coin [12]. Without

this divisible vote count construct, a nominee can further cast at most one vote only without revealing its identity.

Every voting token is associated with a specially constructed binary tree of  $t$  levels, together with its tree signature distributed from the credential authority. Each tree node is associated with a secret value, called the  $k$ -value.

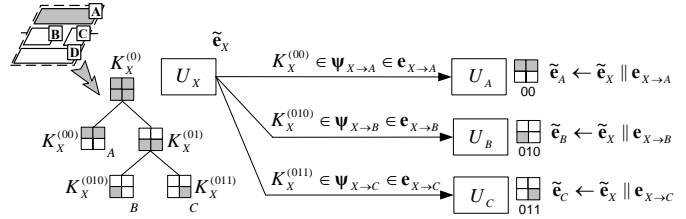


Figure 9. Evidence chain splitting for authorized multiple votes using a single voting token.

The  $k$ -value of the root is denoted by  $k^{(0)}$ . The left child and the right child of  $k^{(j)}$  are denoted by  $k^{(j0)}$  and  $k^{(j1)}$  respectively, where  $j$  is a binary string starting with "0". The  $k$ -values of the leaf nodes are assigned to be  $k^{(j)} = H(\varepsilon \| j)$ , while those of the non-leaf nodes to be  $k^{(j)} = H(H(K^{(j0)}) \| H(K^{(j1)}))$ , where  $K^{(j)} = A^{k^{(j)}}$  and  $\varepsilon$  is a random seed of this binary tree. Figure 10 demonstrates an example of the  $k$ -value assignments with the tree level  $t = 3$ .

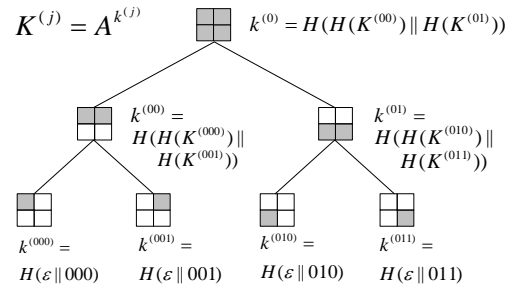


Figure 10. Tree node assignment for authorized multiple votes of a single voting token.

Suppose the voting limit of a single voting token is  $M$ . Then a tree node at the  $i$ -th level represents a vote count of  $M/2^{i-1}$ . The atomic unit of vote splitting is determined by the parameter  $t$ . To cast votes with the vote count represented by  $k^{(j)}$ ,  $\Psi = (\sigma, K^{(0)}, K^{(j)}, \mathbf{H}^{(j)})$  is constructed and revealed, where  $\mathbf{H}^{(j)}$  is composed of all  $H(A^{\bar{k}^{(j)}})$  if  $\bar{k}^{(j)}$  is a direct offspring of an ancestor of  $k^{(j)}$ , but not on the route from  $k^{(j)}$  to  $k^{(0)}$ .  $K^{(j)}$  and  $\mathbf{H}^{(j)}$  actually reveal the  $k$ -values of all ancestors of  $k^{(j)}$  and finally reach the value of  $K^{(0)}$ . Figure 11 illustrates an example on how to compute  $K^{(0)}$  from  $K^{(000)}$  and  $\mathbf{H}^{(000)}$ .

The culprit for the over-voting using multiple voting tokens needs to be identified by the credential authority. The binary tree construct provides an off-line identification mechanism for over-voting resulting from a single voting token.

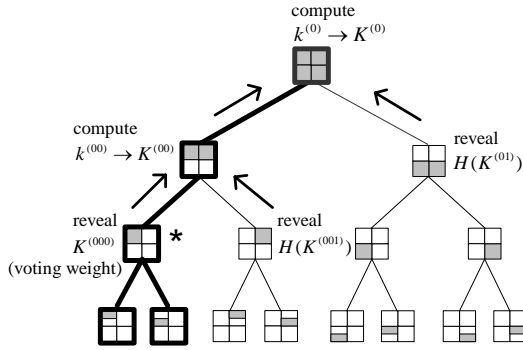


Figure 11. Computation of  $K^{(0)}$  from  $K^{(000)}$  and  $H^{(000)}$ .

Offenders can be identified from the election evidence if the evidence violates the *route node rule* or the *same node rule* [12][16]-[18]. The route node rule states that if a tree node is used for casting a vote, then all the ancestors and all the descendants of this tree node cannot be used for casting votes anymore. The same node rule states that no tree nodes can be used for casting vote more than once.

As shown in Figure 12(a), if two tree nodes on the same route are used, then one node should be an ancestor of another. Suppose  $U_Y$  uses  $k_{Y \rightarrow Z}^{(i)}$  to cast a vote for  $U_Z$ , and  $k_{Y \rightarrow Z'}^{(j)}$  for  $U_{Z'}$  where  $k_{Y \rightarrow Z'}^{(j)}$  is an ancestor of  $k_{Y \rightarrow Z}^{(i)}$ . When  $e_{Y \rightarrow Z}$  is examined,  $k_{Y \rightarrow Z}^{(i)}$  is exposed from  $H_{Y \rightarrow Z}^{(i)}$  and  $K_{Y \rightarrow Z}^{(i)}$ . When  $e_{Y \rightarrow Z'}$  is examined, we can compute the offender identity  $U_Y$  from  $r_{Y \rightarrow Z'} = U_Y x_{Y \rightarrow Z'} + \text{hash}(\Omega_Y, \Omega_{Z'}) k_{Y \rightarrow Z'}^{(j)}$ .

Figure 12(b) illustrates the case when a tree node is used for casting votes more than once. Since  $K_{Y \rightarrow Z}^{(i)} = K_{Y \rightarrow Z'}^{(j)}$  implies that  $k_{Y \rightarrow Z}^{(i)} = k_{Y \rightarrow Z'}^{(j)}$ , the offender  $U_Y$  can be identified by the polynomial interpolation of the responses  $r_{Y \rightarrow Z} = U_Y x_{Y \rightarrow Z} + \text{hash}(\Omega_Y, \Omega_Z) k_{Y \rightarrow Z}^{(i)}$  and  $r_{Y \rightarrow Z'} = U_Y x_{Y \rightarrow Z'} + \text{hash}(\Omega_Y, \Omega_{Z'}) k_{Y \rightarrow Z'}^{(j)}$ .

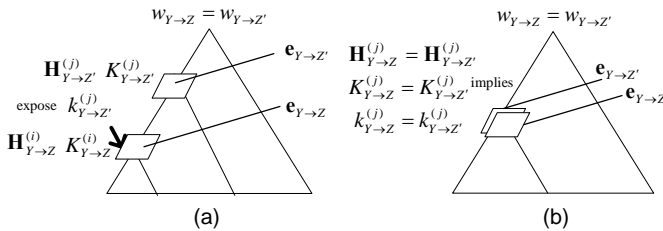


Figure 12. Off-line identification of culprit for over-voting (with identical voting token) derived from distinct evidence pieces.

## 5. TRANSACTION GROUP MERGING FROM DIFFERENT ELECTIONS

Election evidence is the reference for the transaction group formation. In a membership re-configuration, election evidence can also be used to detect the conflict of interests under the membership decision criteria. When a group splits into two, we assume that all the shared secrets are flushed and rebuilt for the two new groups. In the rest of this discussion, we are primarily interested in the merging of groups.

In the merged transaction group, we need to ensure that no node commits dual roles. Because of the conditional linkable feature of the election evidence, it is impossible to recognize the same node from different transaction groups without the credential authority. On-line operation is thus necessary for the transaction group merging operation.

The transaction group merging operation is illustrated in Figure 13. The credential authority takes the evidence chains from different transaction groups as inputs. It verifies that no violations are recorded in the chains and no conflicts exist between the two chains. Then it gives a signature on the new configuration as a substitute of the evidence chain for the merged transaction group. The new configuration includes the essential attributes such as the warrant  $w$  (for the traceability), the challenge-response  $(\hat{x}, \hat{r})$  (for the conditional evidence linkability) and the role choice  $\Omega$  (for the dual-role commitment detection) from each related evidence piece that contributes the merged transaction group. Additional attributes can be included in the new configuration, depending on the specific membership decision criteria.

In the example of Figure 13,  $U_H$  belonged to the transaction group  $\hat{x}$ ,  $U_I$  belonged to the transaction group  $\hat{x}'$ , while  $U_J$  belonged to both transaction groups  $\hat{x}$  and  $\hat{x}'$  before the group merging. The merged group reserved the group id  $\hat{x}$  so that attribute renewal was only needed for evidence pieces from transaction group  $\hat{x}'$ . For example,  $(w'_I, \hat{r}'_I, \Omega'_I)$  was refreshed to  $(w_I, \hat{r}_I, \Omega_I)$  with the merged group id  $\hat{x}$ . The new configuration is simplified by the removal of the redundant evidence pieces from the same node. For instance,  $(w'_I, \hat{r}'_I, \Omega'_I)$  was removed because the attribute  $(w_I, \hat{r}_I, \Omega_I)$  in another chain had provided the necessary information for the new configuration.

The credential authority should not leave unnecessary hints in the new configuration that could violate the conditional evidence linkability. We are not allowed to recognize the same node from two evidence chains by comparing them with the new configuration of the merged transaction group.

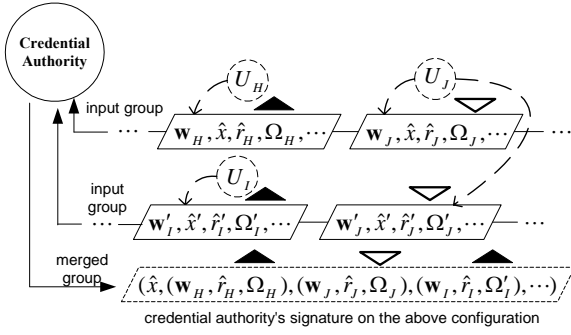
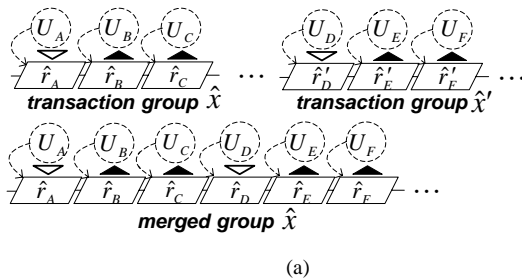


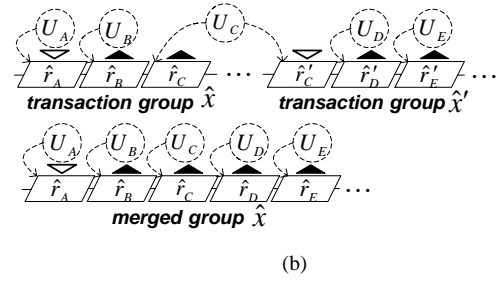
Figure 13. Transaction group merging.

Suppose  $U_H$  and  $U_J$  are the only members in group  $\hat{x}$ , while  $U_I$  and  $U_J$  the only members in group  $\hat{x}'$ . Only  $U_J$  serves as the TTP in both groups. After the groups are merged, only three entities remain in the new configuration. We can easily deduce that  $(w_J, \hat{r}_J, \Omega_J)$  and  $(w'_J, \hat{r}'_J, \Omega'_J)$  are generated from the same source node in different transaction groups. As a result,  $U_J$  can be identified from the challenge-response pairs and other necessary parameters inside the evidence pieces of the two evidence chains although  $U_J$  has not committed a crime. A new election is required if the group merging results in inevitable conflicts. Special cases similar to the one described above should be handled with great care by the credential authority.

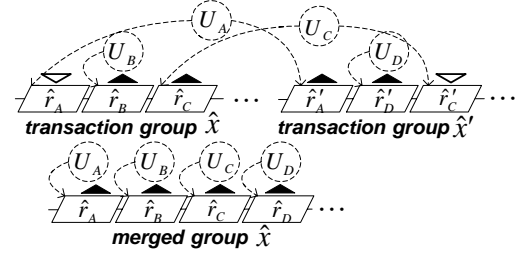
Another type of potential conflicts is caused by changing roles of the participating nodes when different transaction groups are merged, as depicted in Figure 14. Figure 14(a) demonstrates the simple case that no role is changed for every node before and after the group merging. Figure 14(b) shows that  $U_C$  (as a TS in  $\hat{x}'$ ) changed to an AS in the merged group because it acts as an AS in  $\hat{x}$ . This change ensures that  $U_C$  cannot act as a TS of itself in the merged group. Figure 14(c) illustrates a similar situation, except that both  $U_A$  and  $U_C$  are required to change from TS to AS in the merged group. Since there is no TTP in the merged group, a new TTP election is required.



(a)



(b)



(c)

Figure 14. Case study of transaction group merging.

## 6. CONCLUSION

Evidence based resource management is a very effective way to deal with the cheating issues in P2P networks. Only minimal involvement from the credential authority is necessary for nodes to interact with one another for the management of critical issues, such as the selection of the critical group role, TTP, among independent members. Further expansion of this approach to other resource management applications is being investigated.

## REFERENCES

- [1] H. Pagnia, H. Vogt and F. C. Gärtner. *Fair Exchange*. The Computer Journal 46(1), pp. 55-75, Oxford University Press, 2003
- [2] S. Kremer, O.Markowitch and J. Zhou, *An Intensive Survey of Non-repudiation protocols*. Computer Communications, 25(17), pp. 1606-1621, 2002.
- [3] J. Claessens, et al., "Revocable anonymous access to the Internet?" <http://www.esat.kuleuven.ac.be/~joclaess>
- [4] D. Chaum. *Security without identification: transaction systems to make big brother obsolete*. Communications of the ACM, 28(10), pp.1030-1044, 1985
- [5] B. Goddyn. *Defining Anonymity and its Dimensions in the Electronic World*. 2001 <http://users.skynet.be/bgoddyn/publpdf/defining.pdf>
- [6] S. Steys, et al.. *APES Anonymity and Privacy in Electronic Services*. 2001. [https://www.cosic.esat.kuleuven.ac.be/apes/docs/d2\\_final.pdf](https://www.cosic.esat.kuleuven.ac.be/apes/docs/d2_final.pdf)
- [7] A. Lysuanskaya, R. Rivest, A. Sahai and S. Wolf. *Pseudonym Systems*. Selected Areas in Cryptography, LNCS 1785, 1999.
- [8] G. Stubblebine, P. F. Syverson and D. M. Goldschlag, *Unlinkable serial transactions: protocols and applications* Stuart, ACM Transactions on Information and System Security (TISSEC), 2(4), pp. 354-389, 1999
- [9] N. Ferguson. *Single term off-line coins*. Technical Report CS-R9318, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993.
- [10] R. Y. Chan, J. C. Wong and A. C. Chan, "Anonymous E-voting Systems with Non-Transferable Voting Passes." *World Computer Congress, SEC2000*, 2000.
- [11] T. C. Lam and V. K. Wei, "Mobile Agent Clone Detection using General Transferable E-Cash." *International Conference on Information Security (InfoSecu'2002)*, 2002.

- [12] T. Eng and T. Okamoto, "Single-Term Divisible Electronic Coins." *In Advances in Cryptology – Proceedings of EUROCRYPT '94*, LNCS 950, pp. 306 – 319, Springer-Verlag, 1995.
- [13] D. Chaum and T. P. Pedersen, "Transferred Cash Grows in Size." *EUROCRYPT '92*, pp. 390 -407, 1992.
- [14] H. Y. Wong, *Issues in Electronic Payment Systems: A New Off-line Transferable E-coin Scheme and a New Off-line E-check Scheme*. Master's Thesis, Department of Information Engineering, The Chinese University of Hong Kong, 2001.
- [15] J. S. Cheng, and V. K. Wei, "Defenses against Truncation of Computation Results of Free-Roaming Agents.", *the Forth International Conference on Information and Communications Security, ICICS '02*, 2002
- [16] T. Okamoto and K. Ohta., "Universal Electronic Cash." *In Advances in Cryptology – CRYPTO'91*, pp. 725-729, 1991.
- [17] T. Okamoto, "An Efficient Divisible Electronic Cash Scheme." *In Advances in Cryptology – Proceedings of CRYPTO '95*, LNCS 950, pp. 438 – 451, Springer Verlag, 1995.
- [18] Y. Tsiounis, Y. Frankel, and A. Chan, "Easy come – easy go divisible cash (updated version, GTE Tech report)." *EUROCRYPT '98, LNCS*, pp. 561-575, 1998.
- [19] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, Fla, 1997.
- [20] T. C. Lam and J-C Liu. *On the Evidence Based Peer-to-peer Resource Management in Distributed Computing Systems.*, Technical Report 2003-7-2, Department of Computer Science, Texas A & M University.