A Novel Method to Maintain Privacy in Mobile Agent Applications

Kun Peng, Ed Dawson, Juanma Gonzalez Nieto, Eiji Okamoto, and Javier López

Information Security Institute,
Queensland University of Technology
{k.peng, juanma, e.dawson}@qut.edu.au
http://www.isrc.qut.edu.au/people/pengk

Abstract. Two methods to implement privacy in network communication, anonymity and DCSC (data confidentiality and secure computation) are analysed and compared in regard to privacy in mobile agent applications. It is illustrated that privacy through DCSC is more suitable in mobile agent applications. To support this conclusion, privacy is concretely implemented in a bidding mobile agent scheme in this paper. Success of this example demonstrates that privacy can be practically achieved in mobile agent applications through DCSC without compromising the advantage of mobile agent.

Keywords: Mobile agent, privacy, DCSC, secure computation.

1 Introduction

Mobile agents [9, 8, 19, 20] are autonomous software entities that relay code, data and state through multiple nodes. Usually, an originator generates the mobile agent and sends it out to collect data, which is then used by the originator for a special purpose. The advantage of mobile agent is that it is a real-time service, so can visit dynamically chosen nodes to collect data instantly. For example, with the help of a bidding mobile agent, a buyer (seller) can instantly get the bids from a dynamic set of bidders. Then he can immediately choose one bid as the winning bid. Compared to the traditional e-auction schemes [12, 15, 17], a bidding-mobile-agent-based auction is more instant, flexible and convenient.

Usually, compared to traditional network applications like traditional e-auction and e-voting [14, 2, 10, 11], a mobile agent application has the following properties.

- Dynamic: the nodes in the communication network are usually temporally connected terminals fitted with a relay function.
- Instant: network service must be available instantly without preparation or delay.
- Flexible: various nodes and communication patterns may be involved.

Y.G. Desmedt et al. (Eds.): CANS 2005, LNCS 3810, pp. 247–260, 2005.

With these properties, mobile agent has its advantage in circumstances where dynamic and instant network services are needed. Without these properties, mobile agent has no advantage over the traditional network applications.

As the nodes usually may want to conceal their personal privacy in mobile agent applications, in certain cases no node may permit his identity to be linked to his data. More precisely, a node's privacy is the unlinkability between his identity and his data. A definition of privacy in a mobile agent application is as follows.

Definition 1. *A mobile agent application is private if no node's data can be linked to its identity.*

For example, a bidding mobile agent application is private if except for the winner no bidder can be linked to its bid. The only known private mobile agent schemes are [19, 20], two bidding agents. In [19, 20], privacy is implemented through anonymity of the nodes, a method which is inefficient and inconsistent with the properties and advantages of mobile agent application. So designing practical privacy mechanism in mobile agent application is a challenging task. The design must take into account the important fact that as a real-time network application mobile agent has its advantages, which should not be sacrificed in the implementation of privacy.

In this paper, a new privacy mechanism is proposed in mobile agent scheme. The new mechanism, called DCSC, employs data confidentiality and secure computation to achieve privacy in network communication. Basing privacy on data confidentiality and secure computation is not a new idea. For example, it is widely applied to traditional network applications like electronic auction [12, 15] and e-voting [10, 11]. Although this privacy mechanism has not been applied to mobile agent schemes, it has some advantages in regard to mobile agent over the privacy mechanism based on anonymity. The DCSC privacy mechanism is more efficient and does not conflict with the advantages of mobile agent applications. So it is more suitable to mobile agent than the privacy mechanism based on anonymity. DCSC is applied to a new bidding mobile agent scheme with the same circumstance as [19, 20]. The new bidding mobile agent scheme illustrates that privacy can be practically achieved in mobile agent applications without compromising its advantages.

The remainder of this paper is organised as follows. In Section 2, privacy in network communication is analysed and two privacy mechanisms are compared. It is shown that DCSC privacy has its advantages in some applications. In Section 3, it is illustrated that DCSC privacy is more suitable for privacy in mobile agent and often the only feasible solution for private mobile agent application. In Section 4, secure computation techniques are introduced to support DCSC. Especially, an efficient secure computation technique to be used later in the paper, ciphertext comparison, is recalled. In Section 5, a concrete application of DCSC privacy in mobile agent, private bidding mobile agent, is designed on the base of ciphertext comparison. In Section 7, the paper is concluded.

2 Privacy in Network Communication

A communication network is composed of a few nodes and used to transmit messages through the nodes. There are many security requirements on network communication. This paper focuses on one of them, privacy, a property widely desired in network applications.

Definition 2. *Network communication is private if no node in the network can be linked to his data transmitted in the network.*

This unlinkability in network communication is frequently required. For example, on-line buyers using e-cash [5], on-line bidders in e-auction [12, 15, 17] and on-line voters in e-voting [10, 11] do not want to be linked to the items they buy, their bids and their votes respectively.

There are two methods to implement privacy in a communication network: anonymity and DCSC (data confidentiality and secure computation). Anonymity of a node requires that the identity of the node or its other identification information like IP address or geographic location is concealed. Anonymity ensures that no node is identified, not to mention to be linked to any data. Under DCSC, all the data are always confidential (encrypted) even when being processed such that no identification can be linked to any data in plaintext.

The idea of the anonymity mechanism is simple: if a party is anonymous, his behaviour cannot be linked to his identity. To implement anonymity of a party, a pseudonym for him and untraceability of his data are usually necessary. The party can use the pseudonym to label his data such that his identity does not appear in the network communication. The data in the network communication must be untraceable such that any data cannot be linked to its owner by tracing it back to its origin (e.g. address of its owner). Another role of the pseudonym is that recoverable pseudonym can be designed such that anonymity can be revoked by recovering the corresponding identity from a pseudonym. The only known practical method to implement untraceability is mix network [1, 7]. A mix network is an additional communication network interleaving with the existing communication network, whose role is to relay and shuffle the data transmitted between any two nodes in the existing communication such that data transmission in the existing communication network becomes untraceable. Although the idea of the anonymity mechanism is simple and direct, it has the following drawbacks.

- Anonymity is difficult to achieve in special applications with certain communication patterns. For example, implementation of privacy is difficult between neighbouring nodes when relay communication pattern is employed. Mobile agent is such an example. When a mobile agent visits a node, the node excutes the agent to determine the next node to relay the agent to. So each node definitely knows the identity of the next node.
- Pseudonym is usually implemented through special signature schemes like blind signature, group signature or ring signature. Especially, when authentication is required, complex and inefficient group signature [4] or ring signature must be employed. Compared to normal signature schemes, these

signature schemes require costly set-up, complex maintenance, inefficient generation and verification and intensive network communication.

– Mix network needs additional network communication interleaving with the existing network communication, which may affect or even conflict with the existing network communication. For example, when the existing network communication is temporal, instant and dynamic, it is inconsistent with mix network, which is not always temporally or instantly available and requires setting up beforehand and verification afterwards. Moreover, mix network is inefficient (especially when its correctness is required to be publicly verifiable) and needs intensive network communication.

DCSC is composed of two key cryptographic techniques: data confidentiality through encryption and secure computation of the encrypted data without revealing them. Under DCSC, data in the communication network are encrypted (with a semantically secure encryption algorithm[1]) and never decrypted. After the encrypted data is transmitted and collected and the network communication finishes, the encrypted data may be processed and used for a certain purpose. When the data is processed, a secure computation technique [18, 15, 16] is employed to compute a required function of the data without decrypting them. Although all the encrypted data is traceable and labelled with its owner's identity, they are kept confidential for ever. So no party can be linked to any known data (in plaintext). Note that the secure computation takes place after the network communication finishes and out of the communication network, so is independent of the network.

Data confidentiality can be easily and efficiently implemented as any semantically secure encryption algorithm can be employed. Complexity and cost of secure computation depends on which function of the data is computed. Usually, a general secure computation solution to compute any function is less efficient, while secure computation solution to certain functions are more efficient. With the progress in secure computation techniques, more and more functions can be efficiently computed with encrypted inputs. Another advantage of DCSC is that data confidentiality is achieved. As in some applications, it is desired to conceal the statistic information of the data, data confidentiality is needed even if anonymity has been implemented to prevent the link between the data and their owners.

Both privacy mechanisms are widely employed in cryptographic applications. Anonymity-based privacy is more popular in e-cash [5], while DCSC privacy is employed in most private e-auction schemes [12, 15] (the only known anonymity-based private e-auction is [17].). In e-voting, both anonymity-based privacy [14, 2] and DCSC privacy [10, 11] are common. When choosing which privacy mechanism to use, the following factors should be considered.

[1] An encryption algorithm is semantically-secure if given a ciphertext c and two messages m_1 and m_2, such that $c = E(m_i)$ where $i = 1$ or 2, there is no polynomial algorithm to find out i.

- Semantically secure encryption is much simpler and more efficient than group signature or ring signature, which require costly operation both before and during the network communication.
- Mix network is inefficient and needs an additional interleaving network service, which may affect or even compromise the existing network.
- Secure computation is independent of the communication network, so brings no side effect to communication.
- Appropriate and efficient secure computation technique is necessary for success of DCSC.

So, if secure computation of the function of the data is relatively efficient, or the pseudonym technique or mix network is inconsistent with the existing communication, or data confidentiality is desired, DCSC instead of the anonymity-based privacy mechanism should be applied.

3 Privacy in Mobile Agent Applications

Privacy is important in mobile agent like in other network applications. As stated before, the advantage of mobile agent over traditional network applications is that it is a temporal, dynamic, instant and flexible real-time service. Unlike traditional network services, a mobile agent application does not involve preparation or setting-up work, long-lasting network connection or communication delay. A mobile agent can instantly travel through temporal network connection and implement a certain application without any interference or delay. Without this advantage, mobile agent is useless. For example, if real-time service is not required, traditional e-auction and e-voting scheme are more mature, stable and reliable than bidding agent and voting agent.

In the known private mobile agent applications [19, 20], privacy is implemented through anonymity of the nodes. However, in the privacy implementation in [19, 20] only pseudonym is covered while untraceability, a more essential primitive, is not mentioned. So these privacy implementations are incomplete and unreliable. Careful study shows that implementing privacy in mobile agent application through anonymity is unsuitable and in most cases infeasible. Besides the efficiency concern caused by group (ring) signature and mix network, the following drawbacks demonstrate that anonymity-based privacy is inconsistent with mobile agent.

- Group signature and ring signature require every participant to register at a certain time before the network communication starts, which is contradictory to the requirement of instant service in mobile agent applications.
- An additional mix network is involved in the communication. If the mix network is not ready between any two neighbouring nodes, communication fails. Note that a mix network is not often available locally at any temporal time and dynamic location (in many cases, it is impossible to set up a mix network instantly at a certain given location.). On the other hand, mobile agent employs the relay communication pattern and requires instantly

available local relay communication service. Even if a local mix network is instantly available, the relay communication pattern still reveals information about address or location between neighbouring nodes. Moreover, shuffling in a mix network is essentially a batch operation instead of a instant service. So the dynamic and instant behaviour of the mobile agent application must depend on mix network, a service not dynamically or instantly available. This is a serious inconsistency.
- Generation of group signature and ring signature is less efficient than normal encryption or signature generation. Additionally, group signature and ring signature produce longer messages. So the nodes with limited computation capability and wireless communication with limited bandwidth cannot afford this additional computation and communication.

On the other hand, DCSC is suitable for privacy in mobile agent. Data confidentiality is efficient to implement using encryption. Data confidentiality does not increase communication burden. Although secure computation brings additional computation and communication, it does not delay the communication. With progress of secure computation technology [16], efficient secure computation is possible in many mobile agent applications. The most important fact is that DCSC does not compromise the advantages of mobile agent. As a result, DCSC is a better solution to privacy than anonymity in mobile agent applications.

4 Secure Computation

Secure computation techniques are essential for DCSC privacy, so are introduced in this section. Secure computation [18, 15, 16, 13] is also called multiparty computation or secure evaluation in some literature. Its role is to compute a function with encrypted inputs. The private key is shared by multiple parties such that no input can be decrypted under a threshold trust assumption. The function is evaluated by the multiple parties such that the function result is obtained while no input is revealed. It is demonstrated in [13] that any Boolean function with a circuit of linear size can be efficienctly evaluated without revealing the inputs, while a function with a k bit output can be deduced to k Boolean functions. Current secure computation techniques [15, 13] can provide solution to a wide range of functions. So DCSC can be generally implemented in mobile agent schemes in many application. Function-oriented special evaluation techniques can be employed to improve efficiency. For example, addition through secure computation is very efficient with the help of an additive homomorphic encryption algorithm[2]. In another example, e-auction, both general secure computation techniques [15, 13] and more efficient specially-purposed secure computation techniques [12] have been proposed to process the encrypted bids without decrypting them in recent private auction schemes.

[2] An encryption with encryption function $E()$ is additive homomorphic if $E(m_1)E(m_2) = E(m_1 + m_2)$.

The millionaire problem is the most intensively studied function in secure computation. When Yao [18] first proposed secure computation, he studied the millionaire problem as an example. In the millionaire problem, two ciphertexts are compared without being decrypted to determine which encrypts a larger message. So solution to the millionaire problem is called ciphertext comparison. The millionaire problem is important as many complex computations can be deduced to it. In this paper, ciphertext comparison is employed to achieve privacy in a bidding mobile agent scheme. In the recent years, progress has been made in finding an efficient solution to the millionaire problem. The most efficient verifiable ciphertext comparison technique so far is proposed in [16], which is efficient enough for practical applications.

In [16], two L-bit messages m_1 and m_2 are bitwise encrypted and then compared as follows.

1. The two messages m_1 and m_2 are represented bit by bit as $(m_{1,1}, m_{1,2}, \ldots, m_{1,L})$ and $(m_{2,1}, m_{2,2}, \ldots, m_{2,L})$.
2. The two messages are bitwise encrypted $c_1 = (c_{1,1}, c_{1,2}, \ldots, c_{1,L}) = (E(m_{1,1}), E(m_{1,2}), \ldots, E(m_{1,L}))$ and $c_2 = (c_{2,1}, c_{2,2}, \ldots, c_{2,L}) = (E(m_{2,1}), E(m_{2,2}), \ldots, E(m_{2,L}))$ where $E()$ is a additive homomorphic encryption algorithm. The private key is shared by multiple participants such that any decryption is possible only when the number of cooperating participants is over a threshold.
3. c_1 and c_2 are sent to the participants, who are required to test whether

$$\begin{gathered}(D(c_{1,1}) = 1 \wedge D(c_{2,1}) = 0) \ \vee \\ (D(c_{1,1}) = D(c_{2,1}) \wedge D(c_{1,2}) = 1 \wedge D(c_{2,2}) = 0) \ \vee \ \ldots \ \vee \\ (D(c_{1,1}) = D(c_{2,1}) \wedge D(c_{1,2}) = D(c_{2,2}) \wedge \ldots \wedge D(c_{1,L-1}) = D(c_{2,L-1}) \\ \wedge D(c_{1,L}) = 1 \wedge D(c_{2,L}) = 0)\end{gathered} \tag{1}$$

   without decrypting any bit encryption. $m_1 > m_2$ if and only if logic test (1) returns TRUE.
4. The participants exploits homomorphism of the encryption algorithm and use two cryptographic primitives, batch verification and zero test, to test

$$\begin{gathered}D(c_{1,1}/(E(1)c_{2,1})) = 0 \ \vee \ D(c_{1,1}/c_{2,1})^{t_1}(c_{1,2}/(E(1)c_{2,2}))^{t_2} = 0 \ \vee \\ \ldots \ \vee \ D(\textstyle\prod_{i=1}^{L-1}(c_{1,i}/c_{2,i})^{t_i})(c_{1,L}/(E(1)c_{2,L}))^{t_L}\ ) = 0\end{gathered} \tag{2}$$

   where $t_1, t_2, \ldots, t_L$ are randomly chosen. Logic test (2) is equivalent to logic test (1). If and only if logic test (2) returns TRUE, the participants declare $m_1 > m_2$. Details of batch verification and zero test are described in [16].

It is proved in [16] that the ciphertext comparison technique is correct and sound: $m_1 > m_2$ if and only if the zero test in (2) returns *true*. It is also illustrated in [16] that the ciphertext comparison technique is private: if the colluding participants are not over the sharing threshold of the private key, no information about m_1 or m_2 is revealed except which one is larger. In this paper, the ciphertext comparison technique above is denoted as $CC(c_1, c_2)$.

5 Implementation of DCSC Privacy in Bidding Mobile Agent Scheme

In this section, DCSC privacy mechanism is implemented in a typical mobile agent application: bidding mobile agent. A bidding agent is generated and sent out by an originator to sell or buy an item. It migrates to multiple bidding nodes to collect their price quotes, and is free to choose its next move dynamically based on the data it acquired from its journey. The agent finally returns to the originator with the bids of all the bidders. Then the originator chooses a winning offer (bid). As the bidding nodes usually want to conceal their personal privacy, no node permits his identity to be linked to his bid. So privacy is necessary in the application of bidding mobile agent. The existing bidding mobile agent schemes with privacy are [19] and [20]. As stated before, these two schemes employ anonymity mechanism to implement privacy, which is incomplete, unreliable and inefficient.

A new bidding mobile agent scheme is designed, which employs DCSC to implement privacy. In the new scheme, there are an originator, some potential bidders and a third party. It is assumed that originator and the third party do not collude. The originator sends out a mobile agent to visit the nodes to collect bids. The mobile agent finally returns to the originator with encrypted bids from all the bidders. The function for the originator to compute is to find out the highest or lowest bid from all the encrypted bids without decrypting them. Namely, he has to find out the ciphertexts encrypting the largest or smallest message from multiple ciphertexts by executing secure computation with the third party. His task is similar to the auctioneer's task in e-auction schemes [15, 12]. However, the secure computation techniques in these traditional auction schemes cannot be employed in mobile agent schemes. The general secure computation techniques in the existing auction schemes [15] are too inefficient. The specially-purposed

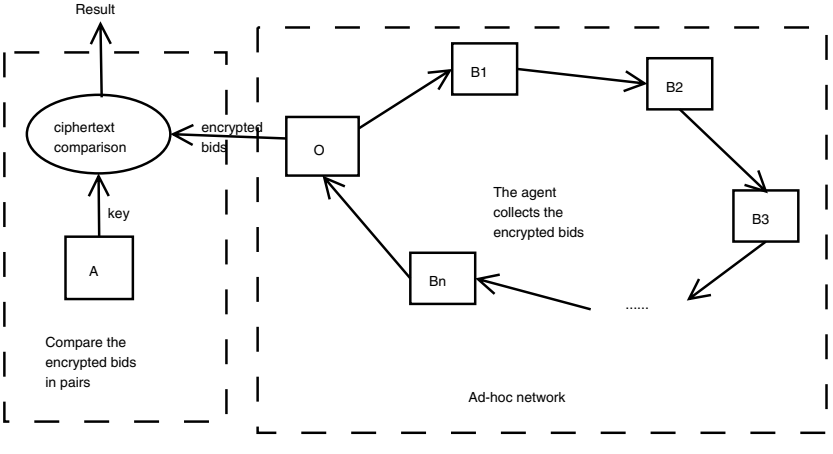

Fig. 1. DCSC private bidding mobile agent

secure computation techniques in the existing auction schemes [12] require each bidder to make a choice for every biddable price in his bid. This bid format causes high computational cost for bid encryption and heavy burden for communication. Fortunately, the function for the originator to compute in a bidding mobile agent application can be implemented through repeated ciphertext comparisons. If the encrypted bids are compared pair by pair, the highest or lowest bid can be found. The ciphertext-comparison-based secure computation is implemented between the originator and the third party (e.g. a hardware like smart-card), who do not collude with each other. More precisely, the third party shares the private key with the originator and cooperates with the originator to perform the ciphertext comparison on the encrypted bids pair by pair. As the ciphertext comparison technique in [16] is publicly verifiable, the third party does not need to be trusted in regard to correctness of computation. Nothing is revealed to the originator except the comparison result as the ciphertext comparison technique in [16] is private if the third party does not collude with the originator.

The new private bidding mobile agent scheme is described in Figure 1 where A is the third party, O is the originator and B_i is the i^{th} bidder. To suit the ciphertext comparison technique in [16], the bids are bitwise encrypted. The symbols to be used in this section are as follows.

- p and q are large primes such that $p = 2q + 1$.
- G is the cyclic subgroup in of Z_p with order q and g is a generator of G.
- L is the bit length of a bid.

The new bidding mobile agent scheme is as follows.

1. Publishing public key
   An additive homomorphic encryption scheme is chosen. The third party chooses private key x_1 from Z_q and publishes his public key $y_1 = g^{x_1}$. The originator chooses his private key x_2 from Z_q and publishes his public key $y_2 = g^{x_2}$.
2. Starting a mobile agent
   The originator generates a mobile agent, which will visit the potential bidders and collect their bids. The public key for data encryption, $y_1 y_2$, is published in the agent, while the corresponding private key is shared by the originator and the third party.
3. Visiting the potential bidders
   When the mobile agent arrives at a potential bidder, the bidder encrypts, signs and submits his bid to the agent. The i^{th} bidder B_i chooses a bid b_i with bitwise representation $(b_{i,1}, b_{i,2}, \ldots, b_{i,L})$. He then encrypts his bid into $c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,L})$ where $c_{i,k} = (a_{i,k},\ b_{i,k}) = (g^{r_{i,k}},\ g^{b_{i,k}}(y_1 y_2)^{r_{i,k}})$ for $k = 1, 2, \ldots, L$. Note this encryption is a variant of ElGamal encryption. Its difference from a standard ElGamal encryption is:
   - the public key is $y_1 y_2$, so the corresponding private key is $x_1 + x_2$, which is shared by the third party and the originator;
   - decryption of ciphertext (a, b) is $\log_g(b/a^{x_1+x_2})$

This modified ElGamal encryption is bitwise and additively homomorphic, so consistent with the ciphertext comparison technique in [16], which will be employed later to compare the encrypted bids. Although decryption of this modified ElGamal encryption algorithm requires computation of discrete logarithm, the computation of discrete logarithm is easy as the message is a bit.

4. Determining winning bid and winner
When the mobile agent returns to the originator, it brings n encrypted bids $c_1, c_2, \ldots, c_n$. The originator compares them in pairs to find the winning bid using the ciphertext comparison technique described in Section 4. For example, after $n-1$ comparisons, the highest or lowest bid can be found. Note that all the bids are additively homomorphically encrypted bit by bit, so is consistent with the ciphertext comparison technique. Comparison of two encrypted bids c_i and c_j is as follows.

   (a) The originator randomly chooses $R_{i,k}$ and $R_{j,k}$ for $k = 1, 2, \ldots, L$ from Z_q. He then calculates $c'_i = (c'_{i,1}, c'_{i,2}, \ldots, c'_{i,L})$ and $c'_j = (c'_{j,1}, c'_{j,2}, \ldots, c'_{j,L})$ where

$$c'_{i,k} = (a'_{i,k},\ b'_{i,k}) = (g^{R_{i,k}} a_{\pi(i),k},\ (y_1 y_2)^{R_{i,k}} b_{\pi(i),k})$$
$$c'_{j,k} = (a'_{j,k},\ b'_{j,k}) = (g^{R_{j,k}} a_{\pi(j),k},\ (y_1 y_2)^{R_{j,k}} b_{\pi(j),k})$$

   and $\pi()$ is a permutation of $\{i, j\}$. Finally, he sends c'_i and c'_j to the third party. Namely the originator re-encrypts and shuffles c_i and c_j and sends them to the third party. The oroginator demonstrates that $D(c'_i)$ and $D(c'_j)$ is a permutation of $D(c_i)$ and $D(c_j)$ without revealing the permutation by proving

$$\begin{aligned}
&(\log_g a'_{i,1}/a_{i,1} = \log_{y_1 y_2} b'_{i,1}/b_{i,1} \ \wedge\ \log_g a'_{i,2}/a_{i,2} = \log_{y_1 y_2} b'_{i,2}/b_{i,2} \ \wedge \ldots \\
&\wedge\ \log_g a'_{i,L}/a_{i,L} = \log_{y_1 y_2} b'_{i,L}/b_{i,L} \ \wedge\ \log_g a'_{j,1}/a_{j,1} = \log_{y_1 y_2} b'_{j,1}/b_{j,1} \ \wedge \\
&\log_g a'_{j,2}/a_{j,2} = \log_{y_1 y_2} b'_{j,2}/b_{j,2} \ \wedge\ \ldots \wedge\ \log_g a'_{j,L}/a_{j,L} = \log_{y_1 y_2} b'_{j,L}/b_{j,L}) \\
&\vee\ (\log_g a'_{i,1}/a_{j,1} = \log_{y_1 y_2} b'_{i,1}/b_{j,1} \ \wedge\ \log_g a'_{i,2}/a_{j,2} = \log_{y_1 y_2} b'_{i,2}/b_{j,2} \ \wedge \ldots \\
&\wedge\ \log_g a'_{i,L}/a_{j,L} = \log_{y_1 y_2} b'_{i,L}/b_{j,L} \ \wedge\ \log_g a'_{j,1}/a_{i,1} = \log_{y_1 y_2} b'_{j,1}/b_{i,1} \ \wedge \\
&\log_g a'_{j,2}/a_{i,2} = \log_{y_1 y_2} b'_{j,2}/b_{i,2} \ \wedge\ \ldots \wedge\ \log_g a'_{j,L}/a_{i,L} = \log_{y_1 y_2} b'_{j,L}/b_{i,L})
\end{aligned}$$

   This proof can be simplified using batch verification technique [3] into proof of

$$\begin{aligned}
&\log_g(\textstyle\prod_{k=1}^{L}(a'_{i,k}/a_{i,k})^{t_k} \prod_{k=1}^{L}(a'_{j,k}/a_{j,k})^{t'_k}) = \\
&\log_{y_1 y_2}(\textstyle\prod_{k=1}^{L}(b'_{i,1}/b_{i,1})^{t_k} \prod_{k=1}^{L}(b'_{j,1}/b_{j,1})^{t'_k}) \qquad (3) \\
\vee\ &\log_g(\textstyle\prod_{k=1}^{L}(a'_{i,k}/a_{j,k})^{t_k} \prod_{k=1}^{L}(a'_{j,k}/a_{i,k})^{t'_k}) = \\
&\log_{y_1 y_2}(\textstyle\prod_{k=1}^{L}(b'_{i,1}/b_{j,1})^{t_k} \prod_{k=1}^{L}(b'_{j,1}/b_{i,1})^{t'_k})
\end{aligned}$$

   where t_k and t'_k are short integers randomly chosen by the originator. Proof (3) can be implemented using ZK proof of equality of logarithm [5]

and ZK proof of partial knowledge [6]. The proof can be publicly verified by anyone.

(b) The third party re-encrypts and shuffles c'_i and c'_j. He randomly chooses $S_{i,k}$ and $S_{j,k}$ for $k = 1, 2, \ldots, L$ from Z_q. He then calculates $c''_i = (c''_{i,1}, c''_{i,2}, \ldots, c''_{i,L})$ and $c''_j = (c''_{j,1}, c''_{j,2}, \ldots, c''_{j,L})$ where

$$c''_{i,k} = (a''_{i,k},\ b''_{i,k}) = (g^{S_{i,k}} a'_{\pi'(i),k},\ (y_1 y_2)^{S_{i,k}} b'_{\pi'(i),k})$$
$$c''_{j,k} = (a''_{j,k},\ b''_{j,k}) = (g^{S_{j,k}} a'_{\pi'(j),k},\ (y_1 y_2)^{S_{j,k}} b'_{\pi'(j),k})$$

and $\pi'()$ is a permutation of $\{i, j\}$. The third party demonstrates that $D(c''_i)$ and $D(c''_j)$ is a permutation of $D(c'_i)$ and $D(c'_j)$ without revealing the permutation by proving

$$\begin{aligned}
&(\log_g a''_{i,1}/a'_{i,1} = \log_{y_1 y_2} b''_{i,1}/b'_{i,1} \ \wedge\ \log_g a''_{i,2}/a'_{i,2} = \log_{y_1 y_2} b''_{i,2}/b'_{i,2} \ \wedge \ldots \\
&\wedge\ \log_g a''_{i,L}/a'_{i,L} = \log_{y_1 y_2} b''_{i,L}/b'_{i,L} \ \wedge\ \log_g a''_{j,1}/a'_{j,1} = \log_{y_1 y_2} b''_{j,1}/b'_{j,1} \ \wedge \\
&\log_g a''_{j,2}/a'_{j,2} = \log_{y_1 y_2} b''_{j,2}/b'_{j,2} \ \wedge\ \ldots \wedge\ \log_g a''_{j,L}/a'_{j,L} = \log_{y_1 y_2} b''_{j,L}/b'_{j,L}) \\
&\vee\ (\log_g a''_{i,1}/a'_{j,1} = \log_{y_1 y_2} b''_{i,1}/b'_{j,1} \ \wedge\ \log_g a''_{i,2}/a'_{j,2} = \log_{y_1 y_2} b''_{i,2}/b'_{j,2} \ \wedge \ldots \\
&\wedge\ \log_g a''_{i,L}/a'_{j,L} = \log_{y_1 y_2} b''_{i,L}/b'_{j,L} \ \wedge\ \log_g a''_{j,1}/a'_{i,1} = \log_{y_1 y_2} b''_{j,1}/b'_{i,1} \ \wedge \\
&\log_g a''_{j,2}/a'_{i,2} = \log_{y_1 y_2} b''_{j,2}/b'_{i,2} \ \wedge\ \ldots \wedge\ \log_g a''_{j,L}/a'_{i,L} = \log_{y_1 y_2} b''_{j,L}/b'_{i,L})
\end{aligned}$$

This proof can be simplified using batch verification technique [3] into proof of

$$\begin{aligned}
&\log_g(\textstyle\prod_{k=1}^{L}(a''_{i,k}/a'_{i,k})^{t_k} \prod_{k=1}^{L}(a''_{j,k}/a'_{j,k})^{t'_k}) = \\
&\quad \log_{y_1 y_2}(\textstyle\prod_{k=1}^{L}(b''_{i,1}/b'_{i,1})^{t_k} \prod_{k=1}^{L}(b''_{j,1}/b'_{j,1})^{t'_k}) \qquad (4) \\
\vee\ &\log_g(\textstyle\prod_{k=1}^{L}(a''_{i,k}/a'_{j,k})^{t_k} \prod_{k=1}^{L}(a''_{j,k}/a'_{i,k})^{t'_k}) = \\
&\quad \log_{y_1 y_2}(\textstyle\prod_{k=1}^{L}(b''_{i,1}/b'_{j,1})^{t_k} \prod_{k=1}^{L}(b''_{j,1}/b'_{i,1})^{t'_k})
\end{aligned}$$

where t_k and t'_k are short integers randomly chosen by the originator. Proof (4) can be implemented using ZK proof of equality of logarithm [5] and ZK proof of partial knowledge [6]. The proof can be publicly verified by anyone.

(c) The originator verifies the third party's proof, then performs $CC(c''_i, c''_j)$ with him.

The winning bid can be found by repeated comparisons of the encrypted bids in pair. For example, in a first bid auction, the ciphertext containing a larger bid in c''_i and c''_j is compared in the next comparison with a ciphertext which has not been compared. After $n-1$ such comparisons, the highest bid is found as the winning bid. After the winning bid is found, the originator and the third party cooperate to decrypt it. The winner can claim his winning by revealing his bid and encryption details. If no bidder claims to be the winner, the originator and the third party cooperate recover the shuffling of the winning bid to trace it back to its submitted format. As each submitted bid is signed by the bidder, the winner cannot deny he submitted the winning bid.

6 Analysis

The winning bid is determined through ciphertext comparison. Before each encrypted bid is compared, it is re-encrypted and shuffled by both the originator and the third party. As the re-encryption and shuffling have been publicly verified to be correct, no bid is tampered with before the comparison. As the ciphertext comparison technique in [16] is correct and sound, the comparison of the encrypted bids finds the winning bid.

As the private key is shared between the originator and the third party, no losing bid is decrypted if they do not collude. As the modified encryption algorithm in this paper is semantically secure, no information about the losing bids is revealed before they are compared if the originator and the third party do not collude. The ciphertext comparison technique in [16] is private, so no information about the bids is revealed in each comparison of ciphertext pair except which ciphertext in the pair contains a larger message. As each pair of bids are shuffled by the originator and the third party before they are compared, each comparison does not reveal which bid is larger although it can find the ciphertext containing the larger bid. So no information about the losing bids is revealed in the comparison if the originator and the third party do not collude. Therefore, the new mobile agent scheme achieves data confidentiality and privacy. Note that shuffling of each compared bids is very important for the sake of privacy. Wihtout the shuffling, ranking of all the bids is publicly known, which compromises privacy.

The new private bidding mobile agent is compared against the existing private bidding mobile agents [19, 20] in Table 1. Efficiency advantage of the new private

Table 1. Comparison

Schemes	Data confi--dentiality	Anonymity	Privacy	Advantegs of mobile agent	Implemen--tation
[19, 20]	No	Incomplete	Incomplete	Inconsistent	Mix network not implemented
New scheme	Yes	No	Complete	Consistent	Completely implemented

Table 2. Efficiency advantage

[20]			The new scheme		
computation		communication	computation		communication
recoverable anonymity and encryption	anonymous channel		encryption	ciphertext comparison	
$2n^2 + 4n$	not mentioned but inefficient	$n^2(n+1)$	$2nL$	$(13L+2)$ $(n-1)$	$n(n-1)L$

mobile agent is demonstrated in Table 2 where first bid auction is run. [19] is not included in Table 2 as [20] is an optimisation of [19]. In Table 2, full-length exponentiations are counted in terms of computation, while full-length integers are counted in terms of communication. In Table 2, n is the number of servers and L is the bit-length of the bids. Usually, L is a small integer, while n is much larger. Comparisons in the two tables show that the new private bidding mobile agent scheme is more efficient and provides better service than the existing private bidding mobile agent schemes.

7 Conclusion

Possible methods to implement privacy in network communication are analysed and compared. As a result, DCSC, a privacy mechanism never employed in mobile agent schemes before, is demonstrated to be the appropriate mechanism to implement privacy in mobile agent schemes. DCSC privacy in bidding mobile agent scheme is designed and analysed to demonstrate the advantages of DCSC privacy in mobile agent schemes.

References

1. Masayuki Abe and Fumitaka Hoshino. Remarks on mix-network based on permutation networks. In *Public Key Cryptography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 317–324, Berlin, 2001. Springer-Verlag.
2. Masayuki Abe and Hideki Imai. Flaws in some robust optimistic mix-nets. In *Advances in Cryptology—ACISP 03*, pages 39–50, 2003.
3. Riza Aditya, Kun Peng, Colin Boyd, and Ed Dawson. Batch verification for equality of discrete logarithms and threshold decryptions. In *Second conference of Applied Cryptography and Network Security, ACNS 04*, volume 3089 of *Lecture Notes in Computer Science*, pages 494–508, Berlin, 2004. Springer-Verlag.
4. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *ACISP 2003*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270, Berlin, 2000. Springer-Verlag.
5. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105, Berlin, 1992. Springer-Verlag.
6. R. Cramer, I. B. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Berlin, 1994. Springer-Verlag.
7. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. In *Public Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 145–160, Berlin, 2003. Springer-Verlag.
8. G. Karjoth. Secure mobile agent-based merchant brokering in distributed marketplaces. In D. Kotz and F. Mattern, editors, *Proceedings of the 2nd International Symposium on Agent Systems and Applications and 4th International Symposium on Mobile Agents*, volume 1882 of *Lecture Notes In Computer Science*, pages 44 – 56. Springer-Verlag, London, UK, 2000.

9. G. Karjoth, N. Asokan, and C. Gülcü. Protecting the computation results of free-roaming agents. In K. Rothermel and F. Hohl, editors, *Proceedings of the 2nd International Workshop on Mobile Agents (MA '98)*, volume 1477 of *Lecture Notes in Computer Science*, pages 195–207. Springer-Verlag, Berlin Heidelberg, 1998.
10. Jonathan Katz, Steven Myers, and Rafail Ostrovsky. Cryptographic counters and applications to electronic voting. In *Advances in Cryptology—EUROCRYPT 01*, volume 2045 of *Lecture Notes in Computer Science*, pages 78–92, Berlin, 2001. Springer-Verlag.
11. Aggelos Kiayias and Moti Yung. Self-tallying elections and perfect ballot secrecy. In *Public Key Cryptography, 5th International Workshop—PKC 02*, volume 2274 of *Lecture Notes in Computer Science*, pages 141–158, Berlin, 2002. Springer-Verlag.
12. Hiroaki Kikuchi. (m+1)st-price auction. In *The Fifth International Conference on Financial Cryptography 2001*, volume 2339 of *Lecture Notes in Computer Science*, pages 291–298, Berlin, 2001. Springer-Verlag.
13. Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosn. Characterizing linear size circuits in terms of pricacy. *Journal of Computer System Science 58(1)*, pages 129–136, 1999.
14. Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In *Information Security and Cryptology, ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 389–406, Berlin, 2002. Springer-Verlag.
15. Moni Naor, Benny Pinkas, and Reuben Sumner. Privacy perserving auctions and mechanism design. In *ACM Conference on Electronic Commerce 1999*, pages 129–139, 1999.
16. Kun Peng, Colin Boyd, Ed Dawson, and Byoungcheon Lee. An efficient and verifiable solution to the millionaire problem. In *Pre-Proceedings of ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 315–330, Berlin, 2004. Springer-Verlag.
17. Kun Peng, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Efficient implementation of relative bid privacy in sealed-bid auction. In *The 4th International Workshop on Information Security Applications, WISA2003*, volume 2908 of *Lecture Notes in Computer Science*, pages 244–256, Berlin, 2003. Springer-Verlag.
18. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *IEEE Symposium on Foundations of Computer Science 1982, FOCS 1982*, pages 160–164, 1992.
19. M. Yao, M. Henricksen, E. Foo, and E. P. Dawson. A mobile agent system providing offer privacy. In *proceedings of 9th Australasian Conference on Information Security and Privacy (ACISP 2004)*, pages 301–312, Berlin, 2004. Springer-Verlag. Lecture Notes in Computer Science 3108.
20. M. Yao, M. Henricksen, E. Foo, and E. P. Dawson. Offer privacy in mobile agents using conditionally anonymous digital signatures. In *proceedings of First International Conference on Trust and Privacy in Digital Business (Trustbus 2004)*, pages 132–141, Berlin, 2004. Springer-Verlag. Lecture Notes in Computer Science 3184.