Secure e-Transactions Using Mobile Agents with Agent Broker

Song Han[1], Elizabeth Chang[1] Member IEEE, Tharam Dillon[2], Fellow IEEE

1: School of Information Systems, Curtin University of Technology, Australia.
2: Faculty of Information Technology, University of Technology, Sydney, Australia

Abstract— This paper presents an e-transactions protocol using mobile agents. However, when mobile agents travel to a number of servers for searching optimal purchase for the underlying customer, the mobile codes should be protected. We integrate a secure signature algorithm with the e-transaction algorithm to maintain the security. In addition, an agent broker is involved in the algorithm, that will help to reduce the communications among the mobile agents, the customer, and the servers. We have presented security and privacy analysis for the proposed protocol.

Keywords—Mobile Agent, Security, Short Signature, Privacy, e-Transaction.

I. Introduction

There are increasing number of applications that seek to use mobile agents in e-commerce and virtual communities. Security and privacy are major issues for such environments. Various solutions have been proposed for this issue, for example, encryption techniques, digital signature techniques (including general signature scheme, blind signature scheme, undeniable signature scheme, group signature scheme, and other cryptographic techniques [2], as well as steganography techniques.

Mobile agents are autonomous software entities [1] that can migrate autonomously from one networked computer to another. Therefore, mobile agents can help to fulfill e-transactions initiated by a client in electronic commerce. However, the mobile agent could encounter a hostile environment. For example, a server may compromise the mobile agent and try to obtain private information from the client. A solution to tackle this issue has been proposed. The existing solution is implemented using RSA signatures that result in long signatures and heavy workloads for the mobile agent. Mobile agents will migrate from the client to a server and from one server to another in order to accomplish the client's transaction plan. Therefore, it will be interesting to re-approach this issue. In this paper, we will utilize the short signatures to construct the mobile agents. That will increase the efficiency and reduce the mobile communication workload for the mobile agents.

Another issue is the mobile agent will make electronic transactions interaction with more than one server (electronic shop) in order to find an optimal purchase for its owner (a client or a customer). In this situation, the mobile agent should need to come back to its owner (the customer), since the owner will check whether the purchase is 'the best one' or 'not good'. However, that will increase the communication workload for the mobile agents, as well as for the customer. Therefore, it will be much more interesting if we can provide an agent broker, who will help the customer to make decision during the course of the whole purchase plan, since the agent broker will compare among the different electronic transactions made by the mobile agent and then select the best one.

In addition, it is known that the development of the electronic commerce is influencing and also has already influenced the financial areas in our society. The central point is that the electronic commerce is highly linked to the privacy [8] of the underlying participants in the electronic commerce, for example, the bank account, the credit card, the personal address, etc. Therefore, the privacy is highly related to the financial issues in the electronic commerce. Hence, it is necessary to provide the privacy protection mechanism for the customers in the electronic transactions in our new paper. We utilize some public key techniques to realize this point.

In brief, the characteristics of our new scheme for secure electronic transactions with mobile agents include: We construct the mobile agents using short signatures; we provide an agent broker for the underlying customer and the mobile agents; we provide privacy protection mechanism for the underlying customer and the servers (electronic shops).

The new design in our paper will provide an optimal choice for the current electronic commerce with mobile agents.

The organization of the rest of this paper is as follows. In section 2, we first provide the model of electronic transactions using mobile agents with an agent broker. A new transaction protocol using mobile agents with an agent broker is proposed in section 3. The analysis and proofs are provided, mainly including security analysis and privacy analysis in section 4. The performance analysis is discussed in section 5. In section 6, we present the conclusions for the paper.

II. Model of e-Transactions using MAs with AB

In this section, we will propose a new model for electronic transactions (e-transactions) protocol using mobile agents with an agent broker. This model outlines an overview of the procedures of the e-transactions protocol. In this model, there

0-7803-8971-9/05/$20.00 ©2005 IEEE

are at least four participants involving in the whole process of the purchase initiated by a customer.

Model (Model of e-Transactions Using MAs with AB)

There are at least four participants involving in the model. The participants are: a customer C (which plays the role of the identifier of the customer), a number of servers (i.e. electronic shops) $S_1, S_2, \ldots, S_n$ (which play the roles of all the servers, respectively), an agent broker AB (which plays the role of the identifier of the agent broker) and a number of mobile agents $MA_1, MA_2, \ldots, MA_n$ (which play the roles of these mobile agents, respectively) . Besides these participants, there are six procedures for the proposed model. These procedures deliver the specifications for the electronic transactions protocol using mobile agents (MAs) with an agent broker (AB). The details of this model are as follows:

(1) Setup Algorithm: This procedure is a probabilistic polynomial time algorithm. It generates public key and private key for the customer,. and also some public parameters for the corresponding servers. In this algorithm, the customer will construct her purchase requirements $\mathrm{Re}\, q_C$ according to her purchase plan.

(2) Key Algorithm: This procedure is a deterministic polynomial time algorithm. In this algorithm, the customer and these servers will choose a suitable public key encryption algorithm $E_{pub \otimes prv}$. By a self-certificate technique [2], the customer and the servers choose their own private keys and public keys, respectively. If the bid of one of the servers $S_1, S_2, \ldots, S_n$ is decided as an optimal one, the underlying server will communicate with the customer through the public key encryption algorithm (if needed).

(3) Preparing Mobile Agents: This procedure is a polynomial time algorithm. It involves the interactions between the customer and its mobile agents. The customer will construct some mobile codes for each mobile agent $MA_j (1 \le j \le n)$. These mobile codes include: C, $\mathrm{Re}\, q_C$, and a pair of undetachable signature functions. The undetachable signature function pair are used to generate the bids on the purchase requirement. Therefore, these mobile agents will travel with the mobile codes to these servers.

(4) Bids Generation: This procedure is a deterministic polynomial time algorithm. In this procedure, the servers $S_1, S_2, \ldots, S_n$ will generate the corresponding bid according to the purchase requirement, respectively. And each server equips the underlying mobile agent with their own bid.

(5) Agent Broker Making Decisions: This is a probabilistic polynomial time algorithm. Each mobile agent will send the new version of mobile codes to the agent broker. The agent broker will record and compare these mobile codes (de facto the bids), and provide the customer with the purchase recommendations $R_1 R_2, \ldots, R_n$ with an optimal e-transaction (i.e. the optimal bid).

(6) Accepting Transactions: This procedure is a deterministic polynomial time algorithm. The customer first checks whether the time-stamp is still valid. If it is valid, the customer will verify the signature on the bid. If it is legal, and also the bid is an optimal one, the customer will accept this bid. In the end, the customer will arrange to transfer some money into the bank account of the corresponding server.

III. New Protocol for Secure e-Transactions using Mobile Agents with Agent Broker

In this section, a new protocol for secure e-transactions is proposed. This protocol is implemented using a new undetachable signature scheme. This new undetachable scheme belongs to the domain of short signatures [6,7]. Short signatures have the characteristics of shorter bit-length of signatures, fast signature generation, as well as fast signature verification [4]. These characteristics are imperative for mobile agents, which take part in the secure transactions between a customer and any server.

Previous constructions of udetachable signatures essentially utilize two methods: One method is based on birational functions as introduced by Sharmir. This kind of construction has been proven to be not secure [4], since it is vulnerable against the attacks proposed by Coppersmith et al [5]. The other method is based on RSA signatures. It is known that the signature length will be at least 1024bit in order to maintain the security of the RSA cryptosystem included. That will increase the workload of the mobile agents involved. Therefore, it is still an open problem to construct an optimized undetachable signature scheme for mobile agents. In the following, we will present a new construction for secure transactions with mobile agents. This construction is based on elliptic curve cryptography (ECC) [2,10]. Generally speaking, signatures based on ECC by themselves do not mean they are short signatures. However, the proposed signatures in our paper are short signatures. The details are as follows:

A. Setup Algorithm

The details of the notations used in this paper are as follows:

1. G_1 and G_2 are two (multiplicative) cyclic groups of prime order p;
2. g_1 is a generator of G_1 and g_2 is a generator of G_2;
3. ψ is an isomorphism from G_2 to G_1, with $\psi(g_2) = g_1$; and

4. e is a bilinear map $e: G_1 \times G_2 \to G_T$.

For simplicity one can set G1 = G2. However, as in [2], we allow for the more general case where $G_1 \neq G_2$ so that we can take advantage of certain families of elliptic curves to obtain short signatures. Specifically, elements of G1 have a short representation whereas elements of G_2 may not. The proofs of security require an efficiently computable isomorphism $\psi: G_2 \to G_1$.

When $G_1 = G_2$ and $g1 = g2$ one could take ψ to be the identity map. On elliptic curves we can use the trace map as ψ.

Let G_1 and G_2 be two groups as above, with an additional group G_T such that

$$|G_1| = |G_2| = |G_T|.$$

A bilinear map is a map $e: G_1 \times G_2 \to G_T$ with the following properties:

1. Bilinear: for all $u \in G_1$, $v \in G_2$ and a, $b \in Z$, $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g_1, g_2) = 1$.

We say that (G_1, G_2) are bilinear groups if there exists a group G_T, an isomorphism

$$\psi: G_2 \to G_1,$$

and a bilinear map

$e: G_1 \times G_2 \to G_T$ as above,

and e, ψ, and the group action in G_1, G_2, and G_T can be computed efficiently.

Each customer selects two generators $g_1 \in G_1$, $g_2 \in G_2$, and e(. , .) as above. He will choose $x \in Z_p^*$ and computes $v = g_2^x \in G_2$. H_1 and H_2 are two secure cryptographic hash functions, such as SHA-1 [2]. That is:

(1) Customer selects $g_1 \in G_1$, $g_2 \in G_2$ two generators.

(2) Customer Selects bilinear mapping $e(\cdot, \cdot)$ as above.

(3) Customer randomly selects $x \in Z_p^*$ and computes $v = g_2^x \in G_2$.

(4) Customer selects two securely cryptographic hash functions H_1 and H_2:

Therefore, the private key of the customer is x; the public key is g_1, g_2, $e(\cdot, \cdot)$, v, H_1, and H_2.

Since we are constructing a transactions protocol, we should specify some corresponding information about the customer and the server. For example, who is the buyer? And who is the bidder (de facto seller). That is, what is the corresponding information of the customer and the servers. Here, the servers represent the host computers (electronic shops) the mobile agents will visit in the transactions. Therefore, we let C be an identifier for the customer, and $S_1, S_2, \ldots, S_n$ be the corresponding identifier of these servers, respectively.

We denote the constraints of the customer by $\mathrm{Re}\, q_C$. This item is defined as follows:

$\mathrm{Re}\, q_C$ defines the requirements of the customer for a specific purchase. It includes: (1) the description of a desired product; (2) an expiration date and time stamp; (3) the maximum price that is acceptable to the customer; (4) a deadline for the delivery of the product.

In addition, the agent broker holds the public parameters: g_1, g_2, $e(\cdot, \cdot)$, v, u, and H_2, where $u = g_1^x \in G_1$. The agent broker will utilize these public parameters to help the customer make decisions and choose an optimal e-transactions for the purchase requirement.

B. *Key Algorithm*

The *Key algorithm* is a probabilistic polynomial time algorithm, which is executed by the customer and all the servers, as well as the agent broker.

(1) A practical public key encryption algorithm $E_{pub \otimes prv}$, will be used by the customer and the servers, as well as the agent broker, respectively. Here, pub and prv are the public key and the private key respectively. They may coexist or only one of them exists in the public key algorithm, since it is decided according to different encryption algorithm.

(2) The customer gets a pair of public key pub_C and private key prv_C. Both of them are self-certified by the customer.

(3) Each server $S_j (1 \le j \le n)$ gets a pair of public key pub_{S_j} and private key prv_{S_j}. Both of them are self-certified by these servers, respectively.

(4) The agent broker gets a pair of public key pub_A and private key prv_A. Both of them are self-certified by these servers, respectively.

All these public keys and private keys will be involved when the customer initiates the e-Transaction with all the servers $S_1, S_2, \ldots, S_n$, as well as the agent broker. The public key encryption algorithm can maintain the private communications between the customer and the servers, as well as the agent broker.

C. *Preparing the Agents*

The customer equips the Mobile Agents $MA_1, MA_2, \ldots, MA_n$ with executable codes. The executable codes are in fact an undetachable signature function pair:

$$f(\ ) = (\ ) - a \pmod p$$

and

$$f_{signed}(\ ) = b \times g^{H_2((\ ) - a)}$$

where $a = H_1(C, \mathrm{Re}\, q_C)$ is bounded by p; $b = g_1^{\frac{a}{x}} \in G_1$, where the exponentiation is computed modular p. This b is in fact a variant version of the short signature in the following:

$$a = H_1(C, \mathrm{Re}\, q_C) (\mathrm{mod}\ p)$$

$$b = g_1^{\frac{a}{x}} \in G_1$$

We look on C as a message, $\mathrm{Re}\, q_C$ as a random element. Then, the above a and b could be treated as the signature

$$\sigma = h(m, r)^{\frac{1}{x}}$$

on the message m; where $h(m, r) = g_1^a$. This signature scheme's security is based on an assumption of q-SDH [3].

Equipped with the executable codes, the mobile agents will migrate from the customer to the servers. These mobile agents will carry C and $\mathrm{Re}\, q_C$ as part of their data, and $f(\ )$ *and* $f_{signed}(\ )$ as mobile codes, respectively.

C. *Bids Generation*

Without loss of generality, we may assume that the *j*-th mobile agent $MA_j (1 \le j \le n)$ will migrate to the *j*-th server $S_j (1 \le j \le n)$, respectively. After each mobile agent arrives at each corresponding server, the underlying mobile agent $MA_j (1 \le j \le n)$ will give all its data and the executable code to the underlying server $S_j (1 \le j \le n)$. The server will execute the *executable code* provided by the underlying mobile agent, i.e. $f(\ )$ and $f_{signed}(\ )$. The details are as follows:

(1) Each server $S_j (1 \le j \le n)$ will construct the bid of the $Bid_{S_j} (1 \le j \le n)$. $Bid_{S_j} (1 \le j \le n)$ is defined as follows:

$Bid_{S_j} (1 \le j \le n)$ defines the bid of the j-*th* *server* $S_j (1 \le j \le n)$ for a selling activity. It includes: *1.* the description of the server's product; *2.* the minimum price that will be acceptable to the server; *3.* a deadline for the delivery of the product; *4.* a deadline for paying money into the bank account of the server; *5.* an expiration date and time stamp.

(2) The j-*th* *server* $S_j (1 \le j \le n)$ computes $\alpha_j = H_1(C, S_j, Bid_{S_j})$ with a bid $Bid_{S_j} (1 \le j \le n)$.

(3) The j-*th* *server* $S_j (1 \le j \le n)$ computes

$$m_i = f(x) = \alpha_i - a \pmod p$$

If $m_j \equiv 0 \bmod p$, he will stop, since that is a meaningless transaction for the j-*th* *server* $S_j (1 \le j \le n)$. Otherwise, he will go on the transaction.

(4) The j-*th* *server* $S_j (1 \le j \le n)$ computes:

$$\begin{aligned} \beta_j &= f_{signed}(\alpha_j) \\ &= b \times g^{H_2(\alpha_j - a)} \\ &= g_1^{\frac{a}{x}} \times (g_1^x)^{H_2(\alpha_j - a)} \\ &= g_1^{\left(\frac{a}{x} + xH_2(\alpha_j - a)\right) (\mathrm{mod}\ p)} \in G_1 \end{aligned}$$

Where $g = g_1^x \in G_1$.

(5) The j-*th* *server* $S_j (1 \le j \le n)$ outputs the x-coordinate γ_j of β_j, where γ_j is an element in Z_p.

(6) The j-*th* *server* $S_j (1 \le j \le n)$ hands the mobile broker a tuple

$$C, S_j, Bid_{S_j}, \alpha_j, m_j, \gamma_j;$$

This tuple will represent part of the transaction.

D. *Agent Broker Making Decisions*

In this procedure, the agent broker will help the customer compare the transactions and make decision on them , and then propose a recommendation to the customer. The details are the followings:

(1) When the agent broker receives all the transactions tuples:

$$C, S_1, Bid_{S_1}, \alpha_1, m_1, \gamma_1;$$

$$C, S_2, Bid_{S_2}, \alpha_2, m_2, \gamma_2;$$

$$\vdots$$

$$C, S_j, Bid_{S_j}, \alpha_j, m_j, \gamma_j;$$

$$\vdots$$

$$C, S_n, Bid_{S_n}, \alpha_n, m_n, \gamma_n.$$

from the mobile agents $MA_1, MA_2, \ldots, MA_n$, she will record all the tuples.

(2) The agent broker will check each time-stamp $t_j (1 \le j \le n)$ on the *j-th* bid $Bid_{S_j} (1 \le j \le n)$. If it is legal, she will verify the signature (m_j, γ_j) on it. The verification process is as follows: The agent broker will search a point in G_1: $g_j = (\gamma_j^*, \omega)$ (where ω is an element in Z_p) such that

$$e(g_3, v^{H_2(m)}) = e(g_1, g_2)^{aH_2(m)} e(u, v)^{H_2(m)^2}.$$

If such a point $g_j = (\gamma_j^*, \omega)$ exists, the signature on the bid is valid. Otherwise, the agent broker will put a satisfactory weight *0* on the transaction.

(3) For those transactions on which the checking and verifying are both successful, the agent broker will compare the context of each bid and then put different values of satisfactory weight on these transactions according to their *satisfactory degree* with the purchase requirement.

(4) The agent broker will choose the transaction whose satisfactory weight is the greatest one as an optimal transaction. Assume the optimal transaction is the i-*th* bid Bid_{S_i}. The agent broker will make a recommendation R_i.

(5) The agent broker will arrange the *the i-th mobile agent* MA_i with the code (i.e. recommendation tuple)

$$R_i, C, S_i, Bid_{S_i}, \alpha_i, m_i, \gamma_i.$$

to travel back to the customer. Here, (m_i, γ_i) is an undetachable signature on the transaction.

E. Accepting Transactions

After receiving the recommendation tuple, the customer will verify whether the signature on the recommendation tuple is valid, through searching a point G_1: $g_i = (\gamma_i, z_i)$ (where z_i is an element in Z_p) such that

$$e(g_i, v^{H_2(m)}) = e(g_1, g_2)^{(a+x^2 H_2(m))H_2(m)}.$$

If the signature is valid, the customer will accept the transaction as an optimal purchase.

In detail, if there is no such point, then the customer will not accept this transaction. Otherwise, she will accept this transaction.

That is to say, If the above equality holds, that certifies the transaction is valid. Also, since the recommendation R_i indicates this is an optimal transaction, and then the customer will accept the transaction. Otherwise, the customer will arrange the current mobile agent or another mobile agent to migrate to another server to seek a desirable bid and accomplish the transaction.

IV. Analysis of the Transactions Protocol

This section we will analyze the proposed protocol of transactions with mobile agents. We will provide the security analysis for the proposed protocol. That is, how to extract the signature scheme from transactions? Why it is secure against the server attack? At the same time, we will give a definition on what is server attack. In the second subsection, we will prove that the proposed protocol that answer the questions how the privacy is preserved for both the customer and the server.

A. Security Analysis

It is known that the mobile agents will be vulnerable even in a virtual community, where some servers may be hostile. Therefore, it is necessary for us to analyze the security of the proposed transaction protocol. In this paper, we give the security analysis based on the undetachable signature scheme, which has already been used in this transactions protocol. We first give a new definition, by which the server's attack is formalized; and then the security analysis will be processed with respect to this definition.

Definition A server is successful in attacking this transaction protocol, if by utilizing some valid earlier transactions, the server can forge a new signature $\{\theta, \rho\}$ for a new requirement $\mathrm{Re}\, q_C^*$ of the customer, where $\theta = H_1(C, \mathrm{Re}\, q_C^*) \pmod p$

and $\rho = g_1^{\frac{\theta}{x}}$ (in G_1) (where x is the private of key of the customer) such that:

$$e(f_{signed}(\alpha), v^{H_2(\alpha-\theta)}) = e(g_1, g_2)^{(\theta+x^2 H_2(\alpha-\theta))H_2(\alpha-\theta)}$$

and

$$f_{signed}(\alpha) = g_1^{xH_2(\alpha-\theta)}\rho .$$

In the following, we prove that the proposed transaction protocol is secure against a server's attack.

Theorem 1 The proposed transaction protocol is secure against the attacks made by a hostile server.

Proof By the definition above, the hostile server needs to produce a new valid signature *(a, b)* for a special transaction (α, m, γ), given a history of valid transactions. In fact, it is easy to produce a valid transaction (α, m, γ) for a given (a, b) by the procedures of Executing the Mobile Agent. However, It is hard to produce a new signature (a, b) of the customer such that a includes a new requirement $\mathrm{Re}\, q_C^*$, and also the transaction is accepted by the customer. However, the server will encounter the problem of solving q-SDH. And the q-SDH problem is difficult [2].

B. *Privacy Analysis*

In a virtual community, privacy is imperative with respect to every participant. In fact, it is known that privacy is paramount particularly in resptect to financial issues of the participants in the *electronics transactions* (known as *e-transaction* or *e-business*). Therefore, besides the security analysis, it is also necessary to analyze the privacy of the proposed protocol. We will analyze the privacy of the e-transactions protocol from the following four aspects:

1. Privacy of the signing key of the customer: This privacy is maintained by the mobile agent's executable code, i.e. the pair of functions $f(\,)$ *and* $f_{signed}(\,)$, since the signing key is implied and embedded in the content of $f_{signed}(\,)$.
2. Privacy of the identity of the customer: This privacy is maintained through the encrypted communication. In fact, when the customer sends the mobile agent to some servers to seek "optimal purchase", she will encrypt the whole or part of the tuple $(f(\,), f_{signed}(\,), C, \mathrm{Re}\, q_C)$ (if necessary for the whole content), by utilizing her private key prv_C of the underlying public key encryption.
3. Privacy of the context of the e-Transaction initiated between the customer and a server: This privacy is maintained through the mutual encrypted communications between the customer and the servers, as well as the agent broker, who will utilize the public key encryption algorithm established in the Setup algorithm of the e-Transaction protocol.
4. Privacy of the identity of the underlying servers: This privacy is maintained through the fact: when the agent broker hands the *recommendation tuple* $R_i, C, S_i, Bid_{S_i}, \alpha_i, m_i, \gamma_i$ to the mobile agent to migrate to the customer, the agent broker will encrypt the part of the tuple in which is related to its identity information., by utilizing her private key prv_A of the underlying public key encryption.

V. Performance Analysis

In one-time successful e-transaction initiated by the customer, there are two rounds of communications between the customer and the underlying server. The computation workload is decided by the pair of functions $f(\,)$ *and* $f_{signed}(\,)$. However, the function $f(\,)$ has only one modular minus calculation. The function $f_{signed}(\,)$ and the public key encryption algorithm (if needed) are two important factors, which will influence the performance of the e-transaction protocol. In fact, the function $f_{signed}(\,)$ implies two exponentiation modular computations, and one of them is modular inversion exponentiation computation. Fortunately, the latter can be precomputed by the customer. At the same time, the computation workload of the public key encryption algorithm is directly linked to what public key encryption algorithm will be utilized. In addition, there involved two Weil pairings computation in the procedure of the Accepting the Transaction in subsection 3.E as above.

VI. Conclusions

In this paper, we presented a new transaction protocol using mobile agents. This protocol could be looked on as an instant of models of a virtual community. In a virtual community environment, security and privacy are two important issues. Therefore, this paper provides two aspects of analysis, i.e. security and privacy. Apart from these, we have also provided the overview for the construction of the protocol. In addition, as an important associated product, a new undetachable signature scheme is implied in the proposed transaction protocol. This signature scheme is of short signatures, which are only about 128bits or 160 bits for a practical security level. That will be very efficient for the mobile agents, since they need low computational workloads.

Acknowledgement

The authors would like to give thanks to the anonymous reviewers. This work is supported by Research Fellowship and ARC Founding at CEEBI and School of Information Systems, Curtin University of Technology.

References

[1] Ciarán Bryce, Jan Vitek: The JavaSeal Mobile Agent Kernel. ASA/MA 1999: 103-117.
[2] W. Mao, "*Modern cryptography: theory and practice,*" Prentice-Hall, PTR, USA, ISBN 0-13-066943-1, 2004.
[3] Tomas Sander and Christian F. Tschudin: Protecting Mobile Agents Against Malicious Hosts. Mobile Agents and Security 1998, LNCS 1419, pp. 44-60.
[4] Adi Shamir, Efficient signature schemes based on birational permutations. Advances in Cryptology - CRYPTO '93, LNCS 773, pp. 1-12.
[5] Don Coppersmith, Jacques Stern, Serge Vaudenay, "Attacks on the Birational Permutation Signature Schemes". CRYPTO 1993, LNCS 773, pp. 435-443.
[6] Jacques Patarin, Nicolas Courtois, Louis Goubin: QUARTZ, 128-Bit Long Digital Signatures. CT-RSA 2001: 282-297.
[7] NNicolas Courtois, Short Signatures, Provable Security, Generic Attacks and Computational Security of Multivariate Polynomial Schemes such as HFE, Quartz and Sflash. Eprint 2004/143.
[8] Electronic Privacy Information Centre and Privacy International; Privacy and Human Rights 2003 – An International Survey of Privacy Laws and Developments. *http://www.privacyinternational.org.*
[9] Panayiotis Kotzanikolaou, Mike Burmester, Vassilios Chrissikopoulos: Secure Transactions with Mobile Agents in Hostile Environments. ACISP 2000, LNCS 1841, pp. 289-297
[10] S. S. M. Chow, L. C. K. Hui, S. M. Yiu, K. P. Chow, "*A secure modified id-based undeniable signature scheme based on Han et al.'s scheme against Zhang et al.'s attacks,*" Cryptology ePrint Archive, Report 2003/262.