Secure Electronic Commerce with Mobile Agents

Song Han[1], Elizabeth Chang[1], Tharam Dillon[2]

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

Abstract—Online transactions using mobile agents need secure protocols to help the mobile agents to accomplish the transactions initiated by a client in an electronic commerce. However, the mobile agent could encounter hostile environment. For example, a server may compromise the mobile agent and try to obtain private information of the client. A solution to tackle this issue has been proposed. However, 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 other servers in order to accomplish the client's transaction plan. Therefore, it will be interesting to re-tackle this issue. We present a new scheme for secure transactions using mobile agents in potentially hostile environments. This transaction scheme is implemented by using a new undetachable signature scheme. The new undetachable signature protocol utilizes short signatures, which is desirable for low-bandwidth and efficient mobile communications.

Keywords—Mobile Agent, Information Security, Short Signatures, Privacy, e-Transaction, Virtual Community.

.

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, etc. [9, 10]), and other cryptographic techniques [10], as well as steganography techniques.

Mobile agents are autonomous software entities that can autonomously migrate from one networked computer to another while executing. It can execute across networks in behavior of users. Mobile agents can be useful for many applications, especially those in Electronic Commerce [1]. Despite its many practical benefits, mobile agent technology results in significant new security threats from both malicious agents and hosts.

.

Malicious hosts may cheat the mobile agents migrating to them and therefore interfere with the successful execution of the mobile agents. Therefore, it is interesting how to protect a mobile agent which is in transit or is executing on a remote site. In this paper, we provide an efficient tackle.

In a virtual community, delegation of signing rights is an important issue, since security and privacy are concerned. Consider such an scenario: There is an International Logistics Pty. Ltd. AuHouse, whose President is scheduled to sign a big contract with an Automobile Company in Europe on Feb 28. However, because of certain emergence case, the President has to take part in a meeting held in the General Building of AuHouse in Australia at the same day. This meeting will influence the future of the Auhouse. On the other hand, that contract in Europe is also very important to the AuHouse. For this case, how can the President sign the contract if he could not go to Europe? Undetachable signature protocol will help the President to solve this issue, since the undetachable signature protocol can provide the delegation of signing power whilst preserving the privacy of the President.

Therefore, two issues need to be tackled: The first is how to delegate the signing power? How to secure the private information of the customer? The second is to design short signatures for the mobile agents, which will enhance the capability of the mobile agents for communications in the e-transactions. In addition, it is still interesting whether the e-transactions protocol could preserve the privacy not only for the customer but also for the server. In this paper, we address these issues.

The organization of the rest of this paper is as follows. In section 2, we first provide the definition of undetachable signatures. In section 3, a new transaction protocol with mobile agents is proposed. In section 4, the analysis and proofs are provided, mainly including construction analysis, security analysis, as well as privacy analysis – a very important property for a practical virtual community. The performance analysis and the conclusions appear in section 5 and section 6, respectively.

II. MODEL OF UNDETACHABLE SIGNATURES

In this section, we will provide the definition of undetachable signatures . This is the first definition for undetachable signatures to the best of our knowledge.

An undetachable signature scheme consists of four algorithms, namely Setup, Key, Sign and Verify.

Setup is a probabilistic polynomial time algorithm which takes as input a security parameter *k* and outputs a family of system parameters.

Key is a probabilistic polynomial time algorithm which is executed by a trusted centre and the signers. The input contains system parameters, as well as random parameters which are chosen by the trusted centre and the signers. The output includes a public key $pk \in \underline{K}$ and a corresponding secret key *sk*.

Sign is a probabilistic polynomial time algorithm, which takes as input a secret key *sk* and a message $m \in \underline{M}$ and outputs a signature $Sig_{sk} \in \underline{S}$. In general, there are many valid signatures for any pair $(m, pk) \in \underline{M} \times \underline{K}$.

Verify is a deterministic polynomial time algorithm. The input includes a message and its alleyed signature $Sig_{sk} \in \underline{S}$, as well as system parameters. The output is "Accept" or "Otherwise".

III. New Protocol for Secure Transactions with Mobile Agents Using Short Signatures

A new undetachable signature scheme will be proposed for the protocol of secure transactions. This new undetachable scheme belongs to the domain of short signatures [2-6, 11, 12]. As described in the previous section, short signatures have the characteristics of shorter bit-length of signatures, fast signature generation, as well as fast signature verification [8]. 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 [8]. This kind of construction has been proven to be not secure [7], 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 1024 or much greater 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) [10]. Generally speaking, signatures based on ECC by themselves do not mean they are short signatures, for example [14]. However, the proposed signatures in our paper are short signatures. The details are as follows:

A. *Setup Algorithm*

We follow the notations in [2]:

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.

Joux and Nguyen [15] showed that an efficiently computable bilinear map *e* provides an algorithm for solving the Decision Diffie-Hellman problem (DDH).

Therefore, we use a setting of bilinear mapping groups in reference [2]. 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 [10]. 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, $e(\cdot,\cdot)$, 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 server. Here, the server represents the host computer the mobile agents will visit in the transactions. Therefore, we let C be an identifier for the customer, and S be an identifier of the server.

In addition, we denote the constraints of the customer by $\mathrm{Re}\,q_C$, and the bid of the server by Bid_S. The two items are 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.

Bid_S defines the bid of the server 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.

B. *Key Algorithm*

The *Key algorithm* is a probabilistic polynomial time algorithm, which is executed by the customer and the server; if possible, there exists a Trusted Third Party which is as a justice .

(1) The customer and the server will agree on a practical public key encryption algorithm $E_{pub\otimes prv}$, which will be used by the customer and the server 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 may be authenticated by the the Trusted Third Party, if needed.

(3) The server gets a pair of public key pub_S and private key prv_S. Both of them may be authenticated by the the Trusted Third Party, if needed.

All these public keys and private keys will be involved when the customer initiates the e-Transaction with the server. The public key encryption algorithm can maintain the private communications between the customer and the server.

C. *Preparing the Agents*

The customer equips the Mobile Agent with executable codes. The executable codes are in fact an undetachable signature function pair:

$$f(\ ) = (\ ) - a \ (\mathrm{mod}\, p)$$

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 agent will migrate from the customer to the server. This agent will carry C and $\mathrm{Re}\,q_C$ as part of its data.

D. *Mobile Agent Execution*

After the mobile agent arrives at the server, the agent will give all its data and the executable code to the server. The server will execute the *executable code* provided by mobile agent, i.e. $f(\ )$ and $f_{signed}(\ )$. The details are as follows:

(1) The server computes alpha= H1(C, S, bid_S) $\alpha = H_1(C, S, Bid_S)$ with a bid.

(2) The server computes

$$\begin{aligned} m_0 &= f(x) \\ &= \alpha - a \ (\mathrm{mod}\ p) \end{aligned}.$$

If $m_0 \equiv 0 \bmod p$, he will stop, since that is a meaningless transaction for the server.

(3) The server computes:

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

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

(4) The server outputs the x-coordinate γ of β, where γ is an element in Z_p.

(5) The server hands the mobile agent a tuple

$$C, S, Bid_S, \alpha, m, \gamma;$$

This tuple will represent part of the transaction.

(6) The mobile agent with the tuple migrates to its owner, i.e. the customer.

E. Checking the Transaction

When the mobile agent returns from the server, the customer will check the returned data provided by the mobile agent. The customer will need to follow these steps:

(1) The customer will check the undetachable signature (m, γ) for this transaction by utilizing the following formaula.

(2) The customer will find whether there is a point in G_1: $g_3 = (\gamma, t)$ (where t is an element in Z_p)

Such that the following equation holds in G_1:

$$e\left(g_3, v^{H_2(m)}\right) = e\left(g_1, g_2\right)^{(a + x^2 H_2(m))H_2(m)}$$

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. 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 first provide the construction analysis. That is, how the protocol works? What's the principal of the protocol? How to allow the customer to obtain the optimal purchase? How the mobile agent help Transactions? In the second subsection, 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.

A. Construction Analysis

We will deploy the proposed transactions protocol from the construction point of view. This will help us to further understand the transaction protocol.

In the transaction protocol, the mobile agent is awarded a pair of functions ($f(\ )$ *and* $f_{signed}(\ )$) and migrates with them to the server. This pair of functions maintains the un-leakage of the signing algorithm (actually the signing private key) of the customer. The input x of the server is linked to the server's bid. At the same time, the mobile agent is also given the certified requirements of the customer (a, b), satisfying $f(\ ) = (\ ) - a\ (\mathrm{mod}\ p)$, and $f_{signed}(\ ) = b \times g^{H_2((\ ) - a)}$ *in* G_1. The parameters of function $f(\ )$ are such that the output of this function includes the customer's constraints. The server modifies these by including the bid, Bid_S in the input α, in such a way as to satisfy:

- The message m links the constraints of the customer to the bid of the server.
- Get an undetachable signature (m, γ) for the transaction, where $m = (\alpha - a)\ (\mathrm{mod}\ p)$ and γ is the x-coordinate of the point *beta.* This serves as a certificate which is authenticated by the customer as follows

$$e\left(g_3, v^{H_2(m)}\right) = e\left(g_1, g_2\right)^{(a + x^2 H_2(m))H_2(m)}.$$

The certified constraints of the customer $\mathrm{Re}\,q_C$, and the bid of the server, Bid_S restrict the scope of *the context* of *the transaction*, i.e. the certificate (m, γ) to "optimal bid" transactions with the appropriate time-limits (or more generally, to whatever requirements the customer and the server stipulate).

Note that even if a server ignores the customer's constraints $\mathrm{Re}\,q_C$ and executes the mobile agent associated with the executable code ($f(\ )$ *and* $f_{signed}(\ )$) in order to produce an undetachable signature of the customer for a bogus bid., the signature will be invalid. If a server is not willing to bid for a purchase, then the mobile agent will travel to another server to obtain an optimal bid for the transaction..

B. 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 θ = $\theta = H_1(C, \mathrm{Re}\,q_C{}^*)(\mathrm{mod}\ p)$

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, 25].

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 Checking 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.

We will implement our transaction protocol and provide a test-bed for the virtual community. In the next stage of our work, we will implement our scheme in JAVA or C.

References

[1] J. Claessens, B. Preneel, J. Vandewalle: (How) can mobile agents do secure electronic transactions on untrusted hosts? ACM Trans. Internet Techn. 3(1): 28-48 (2003)

[2] D. Boneh, H. Shacham, and B. Lynn, "Short signatures from the Weil pairing". J. of Cryptology, Vol. 17, No. 4, pp. 297-319, 2004.

[3] D. Boneh and X. Boyen, "Short Signatures Without Random Oracles". In proceedings of Eurocrypt 2004, LNCS 3027, pp. 56-73, 2004.

[4] D. Boneh, X. Boyen, and H. Shacham, “Short Group Signatures”. In proceedings of Crypto '04, LNCS 3152, pp. 41-55, 2004.

[5] D. Boneh and H. Shacham, “ Group Signatures with Verifier-Local Revocation”. In proceedings of the 11'th ACM conference on Computer and Communications Security (CCS), pp. 168-177, 2004.

[6] Don Coppersmith, Jacques Stern, Serge Vaudenay, “Attacks on the Birational Permutation Signature Schemes”. CRYPTO 1993, LNCS 773, pp. 435-443.

[7] Nicolas Courtois, Matthieu Finiasz, Nicolas Sendrier: How to Achieve a McEliece-Based Digital Signature Scheme. ASIACRYPT 2001: 157-174.

[8] Tomas Sander and Christian F. Tschudin: Protecting Mobile Agents Against Malicious Hosts. Mobile Agents and Security 1998, LNCS 1419, pp. 44-60.

[9] Adi Shamir, Efficient signature schemes based on birational permutations. Advances in Cryptology - CRYPTO '93, LNCS 773, pp. 1-12.

[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.

[11] W. Mao, “*Modern cryptography: theory and practice,*” Prentice-Hall, PTR, USA, ISBN 0-13-066943-1, 2004.

[12] Jacques Patarin, Nicolas Courtois, Louis Goubin: QUARTZ, 128-Bit Long Digital Signatures. CT-RSA 2001: 282-297.

[13] NNicolas Courtois, Short Signatures, Provable Security, Generic Attacks and Computational Security of Multivariate Polynomial Schemes such as HFE, Quartz and Sflash. Eprint 2004/143.

[14] Digital Signature Algorithm (DSA), RSA (as specified in ANSI X9.31), and Elliptic Curve DSA (ECDSA; as specified in ANSI X9.62), FIPS 186-2.

[15] Antoine Joux, Kim Nguyen: Separating Decision Diffie-Hellman from Computational Diffie-Hellman in Cryptographic Groups. J. Cryptology 16(4): 239-247 (2003).