An Anonymous Electronic Commerce Scheme with an Off-Line Authority and Untrusted Agents

Josep Domingo-Ferrer and Jordi Herrera-Joancomartí

Dept. of Computer Science, Universitat Rovira i Virgili
Autovia de Salou s/n, E-43006 Tarragona, Catalonia
e-mail {jdomingo,jherrera}@etse.urv.es

Abstract

In the last years, the exponential growth of computer networks has created an incredibly large offer of products and services in the net. Such a huge amount of information makes it impossible for a single person to analyze all existing offers of a product on the net and decide which of them fits better her requirements. This problem is solved with the intelligent trade agents (ITA), which are programs that have the ability to roam a network, collect business-related data and use them to make decisions to buy goods on their owners' behalf. Known ITA systems do not provide anonymity in transactions, require an on-line trusted third party and implicitly assume that the user trusts the ITA. We present a new scheme for an intelligent untrusted trade agent system allowing anonymous electronic transactions with an off-line trusted third party.

Keywords: Intelligent agents, Electronic commerce, Security and electronic payment, Untraceability, Anonymity.

1 Introduction

In the last years, the exponential growth of computer networks has given birth to a new way of doing business, namely electronic commerce. This fact has created an incredibly large offer of products and services for the users of computer networks. Such a huge amount of information makes it impossible for a single person to analyze all the offers of a product existing on the net and decide which of them fits better her requirements. This human limitation in searching the net can be solved with the use of an *agent*, *i.e.* a program that roams the net looking for the items that best satisfy the user requirements. This feature of agents raises important security issues which are regarded from different points of view in the literature. Since the agent is a program that roams the network, it clearly has to be executed at different servers which are not necessarily trusted by the agent (or consequently by the user of the agent). Different solutions have been proposed to try to protect the agent from possible attacks from the servers ([9, 12, 1]). Another security problem is how to protect the servers from malicious agents. Related work can be found based on different approaches like type safe languages ([3, 8]) and proof-carrying code ([11]) among others. Although the usefulness of agents to buy or sell products on their owners' behalf is often pointed out, there exist few examples in the literature explicitly showing how this can be accomplished. In [10], the concept of *intelligent trade agent* (ITA) is defined as an agent that collects business-related data from the net and uses them to make decisions to buy goods on its owner's behalf. A secure intelligent trade agent is presented in the same paper that requires an on-line authorization server during the transaction; the transaction itself is made using the traditional credit card number, which does not preserve buyer anonymity. Furthermore, like in the rest of the literature, the security analysis in that paper focuses on the relationship between agent and server, leaving as obvious that the agent is trusted by the user (this is evident from the fact that the agent is given the credit card number of the user and authorization for using it). Such a confidence assumption cannot be accepted in environments where the user does not have a complete control over the agent. We propose a new scheme for electronic commerce with secure intelligent *untrusted* agents that does not need the on-line autorization center and preserves the anonymity of the buyer. Section 2 sketches the general scheme of the secure intelligent trade agent proposed in [10]. In Section 3, our alternative proposal is described in detail. Section 4 contains the conclusion.

2 A recent intelligent trade agent scheme

In the secure ITA scheme proposed in [10], the main security problem that the authors want to solve is the location of agents. From a security point of view, allowing the agent to roam the network and reside in each visited server is unsafe. The reason is that the

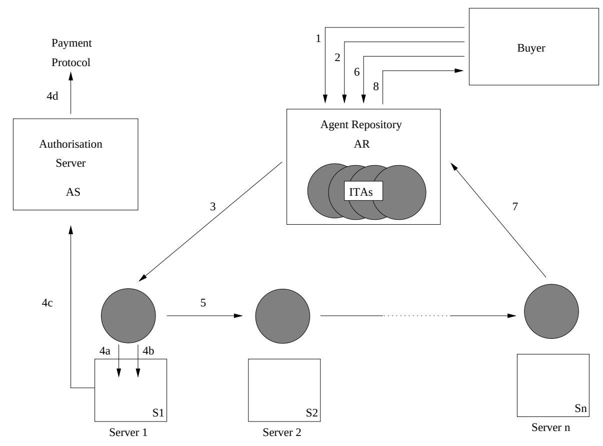

Figure 1: The Van der Merwe-Von Solms ITA scheme

agent carries the credit card number of the buyer, which can be stolen if the agent is attacked while residing in the server. Another place where the agent could reside is the buyer's computer but again this is not a good solution since this computer might not be logged-in while the agent travels the network. So the solution that the authors give is the creation of an Agent Repository (AR) where all agents can reside. Using a distributed-object technology, they propose a scheme where the ITA physically remains in the protective boundaries of the AR but logically it still roams from one server to the next. Figure 1 shows the architecture and the operation of the mechanism. The operation of the scheme depicted in Figure 1 can be summarized as follows:

Protocol 1 (Van der Merwe-Von Solms)

1. *The buyer identifies herself to the AR in order to instruct the agent.*
2. *The buyer gives the instructions to the agent regarding the goods she is interested in.*
3. *The ITA roams to the first server (S1).*
4. *Once the ITA finds an item that fits the requirements specified in Step 2, it starts the transaction via the Authorization Server (AS), which is typically a financial institution such as a bank.*
   (a) *The agent sends to the server the credit card number of the buyer encrypted with the public key of the AS, so only the AS can read this information.*
   (b) *The agent sends to the server a description of the items and prices agreed upon with the server. This information is signed with the private key of the agent and encrypted with the public key of the AS, so the information can only be read by the AS and can only be written by the agent.*
   (c) *The server writes a message with the same information agreed upon with the agent. He signs this information with his private key and encrypts the result with the public key of the AS. Then he forwards to the AS the two messages received from the agent together with his own message.*
   (d) *The AS decrypts and authenticates all messages. He checks that both the descriptions of the goods and the prices are the same and in that case, he executes the payment protocol for transferring the money to the seller's account.*
5. *The agent roams to the next server.*
6. *The buyer tells the agent to come back to the AR. This can be done automatically after some conditions given by the buyer in Step 2 are met.*
7. *The agent returns to the AR.*
8. *The buyer checks what the agent has bought.*

3 Our intelligent and untrusted trade agent scheme

In the scheme of Section 2, there are some weak points that can be improved:

Trust on the agent The user has to trust the agent since the latter is given the user's credit card number.

On-line AS For each transaction, the AS has to be on-line to authorize the transaction and this could be a problem in terms of availability, computing time and bandwidth of the connections to the AS.

Lack of anonymity From his knowledge of the buyer's credit card number and the definition of the objects bought, the AS can link a buyer's identity with a specific purchase, so anonymity is not preserved.

As will be shown later on, the problems above can be solved by combining a secure contract signing protocol (allowing a contract to be signed without the physical presence of the parties) with an electronic payment system allowing off-line anonymous payments. Secure contract signing is dealt with in Subsection 3.1 and anonymous off-line payment is dealt with in Subsection 3.2. Subsection 3.3 describes the architecture of the proposed system.

3.1 Secure contract signing

In any commercial transaction where buyer and seller can see each other, the exchange of goods is not a problem: the actions of paying and receiving the goods bought (from the buyer's viewpoint) or getting paid and delivering the goods sold (from the seller's viewpoint) can be done almost simultaneously. In electronic commerce, a security problem arises because there is no physical coincidence during the transaction between the server and the agent. None of the parties wants to give the first step since they do not trust each other. Basically, there are two ways of handling that problem:

- The first one, proposed in [10], relies on a trusted third-party (the authorization server), which is trusted by both agent and server and arbitrates the whole transaction.
- The second one, proposed in this paper, tries to avoid the need for an authorization server. Instead of an on-line AS, we use an off-line bank. The idea is to use a secure contract signing protocol based on the exchange of secrets. Secure contract signing minimizes the risk incurred by the party who does the first step.

We next recall a secret exchange protocol described in [7] which can be used as a primitive for secure contract signing. Assume that two parties A and B each have $2n$ secret m-bit numbers which they wish to exchange: $\{a_i, 1 \leq i \leq 2n\}$ for A and $\{b_i, 1 \leq i \leq 2n\}$ for B.

Protocol 2 (Secret exchange protocol)

1. *A splits her $2n$ secret numbers in n pairs, for instance (a_{2j-1}, a_{2j}) for $j = 1, \cdots, n$. Then, she sends to B one element of each pair using a 1-2 oblivious transfer (*e.g.* the provably secure oblivious transfer [2] can be used), which means that B receives either a_{2j-1} or a_{2j}, for $j = 1, \cdots, n$, but A does not know which elements B received (each element of a pair has 50% probability of being transferred).*
2. *Simultaneously with Step 1, B does exactly the same with his $2n$ numbers: he splits them in pairs and sends one element of each pair to A using a 1-2 oblivious transfer.*
3. *A and B send to each other the first bit of all their numbers a_i and b_i for $i = 1, \cdots, 2n$, then the second bit, and so on. If A wants to cheat B, she only has a probability $1/2^n$ of success because B has already received n out of the $2n$ secret numbers at Step 1 and A does not know which ones. By symmetry, the same applies if B wants to cheat A.*

A drawback of protocol [7] is that, if B quits the protocol after A has sent the k-th bit of her secret numbers then B has a 2 to 1 advantage, since B has 2^{m-k} choices to obtain one pair while A has $2^{m-(k-1)}$ (twice as much). This problem was solved by Tedrick in [13] with a modified protocol that significantly minimizes the disadvantage of the party that starts the protocol.

A secure contract signing protocol [7] can be designed by taking a secret exchange protocol such as Protocol 2 as a building block:

Protocol 3 (Secure contract signing)

1. *The agent A randomly generates $2n$ DES-like keys K_i^a for $i = 1, \cdots, 2n$, and n pairs of messages (L_j^a, R_j^a) for $j = 1, \cdots, n$. Then she encrypts each message with a diferent key: $P_j^a = E_{K_j^a}(R_j^a)$ for $j = 1, \cdots, n$ and $Q_j^a = E_{K_{n+j}^a}(L_j^a)$ for $j = 1, \cdots, n$.*
2. *The server S does the same (key and message generation), so that he obtains $P_j^s = E_{K_j^s}(R_j^s)$ for $j = 1, \cdots, n$ and $Q_j^s = E_{K_{n+j}^s}(L_j^s)$ for $j = 1, \cdots, n$.*
3. *Either of both parties (or both) creates a contract containing the conditions of the transaction and a clause whereby the contract will be considered as signed if and only if A is able to decrypt P_j^s and Q_j^s for some $1 \leq j \leq n$ and S is able to decrypt P_j^a and Q_j^a for some $1 \leq j \leq n$.*
4. *A and S exchange the $2n$ secret keys using Protocol 2 (or any secret exchange protocol).*

3.2 Off-line anonymous electronic payment

In 1982, Chaum [5] presented the concept of *blind* signature to preserve buyer anonymity in electronic payments by making payments untraceable. Anonymous transactions were further developed in [6]. The idea is that the buyer "mints" her own notes and presents them to the bank, in order for them to be certified and given value. The important thing is that, upon certifying a note, the bank cannot see the note number; in this way, when the note is later spent, the note number cannot be used to trace the identity of the buyer who presented the note for certification to the bank. Chaum's blind signature protocol [6] is next recalled:

Protocol 4 (Blind signature)

1. *The buyer randomly chooses half the digits of a note number n and repeats these digits to form the note number n (this repeated halves property will be subsequently used to tell valid note numbers from junk). Then, the buyer picks a random integer r that will hide the note number to the bank's eyes. The buyer computes $x = nr^e$ where e is the bank's public key that certifies a certain amount of money, say one euro.*

2. *The bank withdraws one euro from the buyer's account and uses its private key d for certifying a one euro note, that is*

$$y = x^d = ((nr^e)^d) = n^d r^{ed} = n^d r$$

Then the bank sends to the buyer the value y.

3. *The buyer computes $z = y/r = n^d$ to obtain the certified note.*

In Chaum's anonymous payment system, when the buyer wants to spend her one euro note, she sends z to the shop and the shop uses the bank's one euro public key e to authenticate z by computing $z^e = n$ and checking that n is a valid note number. Later on, the shop sends z to the bank and the bank does the same authentication check and deposits one euro in the shop's account provided that the note number n has never been used before. The bank records n to guard against double spending.

With the above approach, buyer anonymity is guaranteed but the bank still has to be on-line to prevent double-spending of notes. In [4], an untraceable off-line cash system was presented, also based on blind signatures, which becomes traceable after double-spending. Since double-spenders will be identified later on, the bank does not need to be on-line to check the note numbers of each payment transaction. The payment model of [4] is based on the representation problem in groups of prime order. The basic idea is that during the withdrawal protocol, the buyer embeds her identity into the coin in such a way that nobody can obtain it, provided that computing discrete logarithms is hard. Then, during the payment protocol, the buyer has to answer a time-related challenge from the server in such a way that the answer contains some information about her identity. This information must be such that one instance of it does not reveal anything about the buyer's identity, whereas knowledge of the answers to two different challenges enables the bank to obtain the buyer's identity. In this way, double-spending is deterred because it implies loss of anonymity for the double-spender.

3.3 Architecture of the new scheme

We are now ready to define a new scheme for an intelligent untrusted trade agent. We use the solution presented in [10] for locating the agents, and thus the ITA remains physically in the AR but logically roams the network, thanks to the distributed-object technology. In this scheme, the user does not trust the bank nor the server nor the agent, so we have to introduce an electronic device such a smart card which the user can completely trust and (physically) control. The user uses this card to make all the operations such as withdrawing money from her account or signing a contract with the agent in the way described in Subsection 3.1. In our scheme the whole transaction involves three main steps:

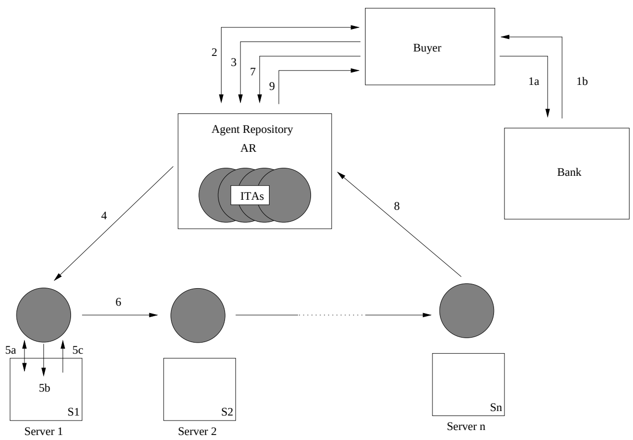

Figure 2: The new anonymous ITA scheme

1. The user withdraws money from her bank account using her smart card to execute a protocol such as the one developed in [4].

2. Once the user and the agent have signed a contract, the user sends to the agent the amount of money specified in that contract.

3. The agent pays the articles bought using the money received from the user.

The architecture and the operation of the new scheme are shown in Figure 2. The operation of the scheme of Figure 2 can be summarized as follows:

Protocol 5 (Anonymous ITA)

1. *The buyer withdraws money from her account using the withdrawal protocol of an off-line untraceable payment system such as the one proposed in [4]:*

   (a) *The buyer's smart card presents to the bank for certification the notes it has minted.*

(b) *The bank withdraws the amount specified by the notes from the buyer's account, certifies the notes and sends them back to the user's smart card.*

2. *The ITA and the buyer sign a contract using a secure contract signing protocol such as Protocol 3; in that protocol, the buyer's role is delegated to her smart card. In the contract, the conditions of their agreement (attributes of the goods to buy, amount of money transferred to the agent, etc.) are specified.*

3. *The buyer sends to the agent the money she has withdrawn at Step 1.*

4. *The ITA roams to the first server (S1).*

5. *Once the ITA finds an item that fits the requirements specified in the contract of Step 2, it starts the transaction:*

   (a) *The ITA and the server sign a contract using again a secure contract signing protocol such as Protocol 3. In that contract, they specify the conditions of the transaction (price, no. of items, warranty, commitment to payment, commitment to delivery of goods, etc.).*

   (b) *The ITA sends to the server certified notes received in Step 3 from the buyer to satisfy the amount specified in the contract.*

   (c) *The server sends to the ITA the goods specified in the contract.*

6. *The agent roams to the next server.*

7. *The user tells the agent to come back to the AR. This can be done automatically after some conditions given by the user in the contract signed in Step 2 are met.*

8. *The agent returns to the AR.*

9. *The user checks what the agent has bought.*

4 Conclusion

We have presented a new scheme for an untrusted and intelligent trade agent system. This system presents clear improvements with respect to previous proposals in the following points:

- The new scheme can deal with untrusted agents thanks to the inclusion of a contract signing protocol between the user and the agent that establishes the terms of the agreement.
- Buyer anonymity is preserved even if the bank and the server collude to obtain the user identity.
- The presented scheme does not need a third party to be on-line during the payment transactions between the agent and the server.

Acknowledgment

This work was partially supported by the Spanish CICYT under grant no. TEL98-0699-C02-02.

References

[1] S. Y. Bennet, "A sanctuary for mobile agents". *DARPA Workshop on Fundations for Secure Mobile Code Workshop*, Mar. 1997, Monterrey, California, U.S.A.

[2] R. Berger, R. Peralta and T. Tedrick, "A provably secure oblivious transfer protocol". *Advances in Cryptology-EUROCRYPT'84*, LNCS 209. Springer-Verlag, 1985, pp. 408-416.

[3] B. N. Bershad, S. Savage, P. Pardyak, E. G. Sirer, M. E. Fiuczynski, D. Becker, C. Chambers and S. Eggers, "Extensibility safety and performance in the spin operating system". In *Proceedings of the Fifteenth Symposium on Operating System Principles*, Dec. 1995.

[4] S. Brands, "Untraceable off-line cash in wallets with observers". *Advances in Cryptology-CRYPTO'93*, LNCS 773. Springer-Verlag, 1993, pp. 302-318.

[5] D. Chaum, "Blind signatures for untraceable payments". *Advances in Cryptology-CRYPTO'82*, Plenum Press, 1983, pp. 199-203.

[6] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete". *Communications of the ACM*, vol. 28 no. 5, Oct. 1985, pp. 1030-1044.

[7] S. Even, O. Golderich and A. Lempel, "A randomizing protocol for signing contracts". *Communications of the ACM*, vol. 28, no. 6, Jun. 1985, pp. 637-647.

[8] W. C. Hsieh, M. E. Fiuczynski, C. Garrett, S. Savage, D. Becker and B. N. Bershad, "Language support for extensible operating systems". In *Proceedings of the Workshop on Compiler Support for System Software*, Feb. 1996.

[9] F. Hohl, "An approach to solve the problem of malicious hosts". Technical Report Number TR-1997-03, Fakultät Informatik, Universität Stuttgart, Mar. 1997.

[10] J. Merwe and S.H. von Solms, "Electronic commerce with secure intelligent trade agents". *Information and Communications Security*, LNCS 1334. Springer-Verlag, 1997, pp. 452-462.

[11] G. Necula and P. Lee, "Safe kernel extensions without run-time checks". In *Proceedings of the Second Sumposium on Operating Systems Design and Implementation*, Oct. 1996.

[12] T. Sander and C.F. Tschudin, "Protecting mobile agent against malicious hosts". *Mobile Agents and Security*, LNCS 1419. Springer-Verlag, 1998, pp. 44-60.

[13] T. Tedrick, "Fair exchange of secrets". *Advances in Cryptology-CRYPTO'84*, LNCS 196. Springer-Verlag, 1985, pp. 434-438.