A Model of a Secure Intelligent Trade Agent

Sthaphon Uraisin and **Tang Van To**
(**jack@panthai.com**) (**tvto@s-t.au.ac.th**)

Faculty of Science and Technology
Assumption University of Thailand

Abstract

In this paper, we propose a model of a secure intelligent trade agent. In building this agent, we adapt our agent to use with a secure and efficient electronic cash (executable digital-cash, x-cash) [1] to enable agents to carry funds and make payments on-site without running the risk of "pick-pocketing". It is a means of binding an offer to accompanying goods or payment, enabling the processes of searching and paying to be unified. The result is a mechanism by which electronic trades can occur in a highly distributed setting with strong security guarantees. This model provides an interface by which users obtain the certificates from the bank and authorizing them to make payments. The user can login to the agent repository (AR) for issuing offer programs w*. The program* w *along with the certificate constitutes a piece of x-cash. An agent together with the executable digital cash (x-cash) which represents a buyer will be sent across the Internet via "asymmetric proxy re-encryption protocol [2,9], executing methods* (w) *on a number of merchant's server (on remote server systems). It takes as input some item, and outputs the amount which the buyer is willing to pay for that item. If the seller wishes to sell an item (Q), he/she can take the x-cash and the item Q to the buyer's bank. By running the program* (w) *on Q, The buyer's bank can determine how much to pay the seller. The buyer's bank may then hold the item for buyer or otherwise arrange to send it to him. The trade is thus completed in a secure fashion without any direct contacts between seller and buyer. Intelligent trade agents must be able to buy goods and pay for them without any human intervention. In this paper, we introduce a model of agent based system, describe some variants and sketch proofs of its security properties*

Keywords*: Electronic commerce, intelligent agent, x-cash, quorum control and asymmetric proxy re-encryption*

1. Introduction

In this paper, a model of secure intelligent trade agent is developed, a way to conduct efficient electronic commerce [10] in an environment of mutual mistrust among the system's users. The promise of high bandwidth at very low cost has conjured visions of an information highway that turns into the World's largest shopping mall.

Billions of electronic transactions will be generated on a daily basis. Features such as efficiency, flexibility, velocity, security should be related to all the operations performed within distributed system. In this context, mobile agent technology is considered an enhancement of distributed systems communities as it provides powerful and efficient mechanisms to develop applications for distributed and heterogeneous systems. To deal with the increasing amount of data/resource available nowadays, and the large number of tasks which have to be performed to manipulate the data / resources, mobile agent technology offers the possibility of executing these tasks in an automated way, with minimal human intervention.[see7 for an overview] This allows the user to concentrate attention on other activities. The solution proposed by mobile agent technology for the distribution problem includes basic functions such as creation, migration, execution of the mobile agents, as well as specialized functions that involve agent security and management. It is possible in such a setting to let prospective trading partners seek each other out and then initiate peer-to-peer transactions. This, however, increases the risk of communications bottlenecks, as communicating with the originator of an offer may require costly traversals of a network. In addition, if the issuer of an offer receives many bids but has limited computational power, this means of commerce could overtax his or her resources. In order to obtain a realistic and efficient solution, we must consider alternative methods of establishing first contact between traders via "asymmetric proxy re-encryption (APR) which we believe that asymmetric proxy encryption allows more efficient communication to a large number of recipients that are physically clustered around the proxy, and develop methods to perform a transaction without peer-to-peer contact when a desirable match is found. To do this, we consider the mobile agent paradigm. Current suggestions for payment schemes are not well adapted to use with mobile agents: if an agent carries digital cash, for instance, it is vulnerable to "pick-pocketing"[10].

On the other hand, not allowing agents to carry funds to perform commerce requires a reduction to the peer-to-peer setting with its attendant bottlenecks. Our aim is to avoid these two types of problems, and to supply an efficient and practical payment scheme which is based on mobile agents. To this end we present a model of a secure intelligent trade agent. It is a means of binding an offer to the accompanying goods or payment, enabling the processes of searching and payment to be unified. When a party receives an x-cash offer, he or she can verify that it is bona fide and can initiate a trade immediately, without contacting the originator directly.

Organization of paper: We start in section 2 by reviewing related work. Section 3 gives the definitions and notation used in the paper, describes our trust model, and formalizes the goals we are seeking to achieve. Section 4 describes how we achieve these goals using a secure intelligent trade agent with x-cash and presents our basic scheme for asymmetric proxy encryption. The conclusions and further work are dealt with in section 5.

2. Review

Public and Secret Information: Let p, q be primes such that $p=2q+1$, and g be a generator of G_p. The proxy servers share a secret key x_1 using a (k, n) threshold scheme (see [4,5]); their corresponding public key is $y_1 = g^{x1} \bmod p$. (Onwards, we assume all arithmetic to be modulo p where applicable, unless otherwise stated.) Likewise, the

recipient has a secret key x_2 with a corresponding public key $y_2=g^{x2}$

ElGamal: Our protocol uses ElGamal encryption [6]: To encrypt a value m using the public key y, a value $g \hat{I}_u Z_q$ is picked uniformly at random, and the pair $(a,b) = (my^g , g^g)$ calculated. Thus, (a, b) is the encryption of m. In order to decrypt this and obtain m, $m = a/ b^x$ is calculated.

Executable Digital Cash (x-cash)[1]**:** the x-cash coin Ω is an expression of an offer ω (as a program or a text description or in any other form) along with all accompanying signatures, certificates, programs, and instructions. X-cash coin Ω containing $[\sigma SK_A (w, P), C]$. When a party receives an x-cash offer, he or she can verify that it is bona fide and can initiate a trade immediately, without contacting the originator directly. X-cash therefore is used to enable the agent to carry funds and make payments on-site without running the risk of "pick-pocketing".

Asymmetric Proxy Re-encryption (APR)[2,9]**:** APR is a translation between ciphertexts from one encryption key to another encryption key. It can be used to forward-encrypted messages without having to expose the cleartexts to the participants involved, a primitive with many potentials commercial uses.

An entity with public key y_1 assigns a proxy, agrees with the proxy on rules for re-encryption, and distributes shares for his secret key x_1 to the servers of the proxy. Later, the proxy receives a transcript E_1, which is an ElGamal encryption of a message m using public key y_1. The proxy produces and outputs a transcript E_2, which is an ElGamal encryption of the same message m, but using a given public y_2, which is chosen according to the rules set by the entity associated with y_1.

3. Definitions, Model, and Goals

3.1 Definitions

Intelligent Trade Agent (ITA) is a mobile agent with the ability to visit many different sites on the network, analyze the data at each site and make decisions as to whether goods should be traded at the specific site. An ITA is autonomous: it can decide where to travel on a network and what goods or services to buy without human intervention. An Intelligent trade agent should be able to pay for goods or services bought. We show how an ITA can use the x-cash to provide payment to merchants.

Informally, an offer is a proposal to trade some collection of goods, moneys, or services for another collection of goods, moneys, or services according to a set of well defined terms. An offer may involve either buying or selling. A bid is a response to an offer. We may describe these ideas more formally in terms of an offer function, defined as a function $w: S \rightarrow T$. Here $S = \{0,1\}^*$ is the space of possible bids and $T = \{0,1\}^* \cup f$ is the space of possible goods, moneys, or services proposed in response to these bids. The symbol f indicates a null response, the bid is deemed unacceptable. We shall use w interchangeably to indicate an offer function and the code implementing an offer function. We denote by $w(Q)$ the output of w on a bid Q. S may be defined to include parameters like the current time and a list of all bids made in response to an offer. We define an x-cash coin W to be an expression of an offer w (as a program or a text description or in any other form) along with all accompanying signatures, certificates, programs, and instructions. A negotiable certificate is an authorization, issued by a financial or other institution, for a trader to make offers using some quantity of assets held by the institution. Let $(SK_{A,,}$

PK_A) denote a secret/public signature key pair held by a trader Alice and let (SK_F, PK_F) denote a secret/public signature key pair held by Alice's financial institution. A negotiable certificate C assumes the form $\sigma\ SK_F\ (PK_A)$, where $\sigma\ SK_F$ denotes a signature using the secret key SK_F. The units of value of the certificate may either be left implicit, or may be specified in an extra field. If Alice wishes to sign over a quantity m of assets to Bob, she creates the signature $\sigma\ SK_A\ (Bob, m)$ and gives it to Bob along with the negotiable certificate C to be redeemed by her financial institution. Thus a negotiable certificate may be loosely regarded as a license to write checks up to a certain amount.

3.2 Trust Model

Let us now present the trust model in which we seek to conduct trades. We then give a formal statement for the goals, regarding both security and flexibility, which we are trying to achieve in this model.

Network: Alice activates the intelligent trade agent (ITA) that will dispatch the ITA in an open network. We assume the following about this network.

1. An adversary may inject ITA of his/her own construction into the network (such as *ITA¢* purporting to come from Alice).
2. The x-cash coin W may be freely read and executed by any party.
3. All parties have unimpeded access to financial institutions.

Parties: We assume the following about the parties in our model.

1. Financial institutions may be trusted to act on behalf of their patrons, but not necessarily of other parties.
2. Financial institutions trust one another.
3. Parties other than financial institutions, the agent repository (AR) and the proxy are not necessarily trustworthy.

Computational assumptions: We make the following computational assumptions.

1. All parties have conventional limited computational resources. (polynomial in an appropriate security parameter)
2. A digital signature scheme is employed in which it is not feasible to falsify the signatures.

3.3 Goals of this paper:

Our goal is to achieve electronic commerce with the following properties within the trust model described above:

1. Entitlement authentication, Any party considering an offer w issued by Alice must be able to determine from the x-cash coin W whether Alice has been issued the goods, services, or moneys being offered. This should be achievable off-line. Note that this property is different from authentication in the usual sense in that Alice's identity is not of concern (and may not even be known). Note also that entitlement authentication is a guarantee that Alice has been issued, but not necessarily that she currently possesses the funds or rights in question: these funds or rights may already have been spent.
2. Fairness [12,13] No one should be able to engage in any exchange not defined by w. Moreover, Alice should be able to specify (in her x-cash coin) how many such exchanges she wishes to engage in.
3. Perfect matchmaking. Any party that receives the x-cash coin W should be able to engage in a fair exchange with Alice. No information beyond publicly available information and that provided by W required.
4. Integrity. Any party must be able to verify that the x-cash coin W has not been tampered with.

5. Efficiency. The x-cash coin W should be compact, and offers and bids should be capable of being processed efficiently.

6. Fast and direct contact between traders: The asymmetric proxy re-encryption is efficient; the AR would only need to send one encryption, along with an authenticated list of recipients. Therefore this multi-cast mechanism would decrease the time overhead of business transactions, because an ITA could visit only the prospected sites which are in the authenticated list. The user need not have to wait for the results because an agent would visit thousands or more of trading servers.

7. Secure intelligent trade agents: The Intelligent Trade Agents (ITAs) are securely dispatched across hundreds of servers all over the world with the ability to buy or sell commodities when they think it is appropriate without running the risk of "pick pocketing".

4. Solution

In this section, we provide details of the protocols used to achieve the goals described above. Before presenting these protocols formally, let us take a brief look at the intuition behind them. Recall that before making an offer, Alice obtains a negotiable certificate C granting her rights to the funds or rights she wishes to offer, and enabling her to transfer those rights to another party. The key idea behind x-cash is the following. Alice constructs her x-cash coin W in such a way that the transfer of rights using C is conditional on having a suitable bid R as input to a piece of code w. In other words, instead of signing over funds or rights to an individual, Alice signs them over based on a piece of code w which evaluates the worth of a bid R. To make a bid, Bob creates a suitable, signed representation R of his bid, and submits it to Alice's financial institution along with W. This financial institution verifies that Alice's negotiable certificate still retains sufficient value for the transaction with Bob, and contacts Bob's financial institution to ensure that Bob too has sufficient funds available. The two financial institutions then process the exchange. The formal details of the protocol are given below. Note that for simplicity of notation, we assume that all signatures have full message recovery.

Initiation of trade

1. Alice has a negotiable certificate C from her financial institution F_A, attributing to her rights to all goods or moneys in T, the range of the offer function w to be used in her x-cash. This certificate is issued against public key PK_A for which Alice holds the corresponding private key SK_A.

2. Before Alice at client computer Cl_1 can issue any instructions to the ITA, the Agent Repository must authenticate her. In section 4.3 we describe the security issues in the process in more details.

3. Alice decides what offer she wishes to make, and constructs an offer function $w : S \rightarrow T$. Again, $S = \{0,1\}^*$ is the space of possible bids and $T = \{0,1\}^* \cup f$ is the space of possible responses to these bids. Alice creates a piece of executable code for her offer function w.

4. Alice decides what policy she wishes to use in accepting bids. For the sake of simplicity, we might allow three possible policies: (1) She accepts all bids until all rights attributed by C are exhausted; (2) She accepts the first j valid bids; or (3) She accepts the best bid received before date d. Alice encodes her policy choice in a field P.

5. Alice constructs the x-cash coin Ω containing $[\sigma_{SK_A}(w, P), C]$ and sets up the ITA by providing it with a list of trade rules (what to buy and sell and under what conditions). All merchants advertise their existence and the products they trade in, at some directory service (yellow pages). When

the ITA is instructed to trade, it consults this directory service for a list of merchant distributed object addresses to visit and trade with.

6. Alice instructs the ITA to start trading. At this point, Alice could disconnect from the client computer and leave the agent to trade on its own.

7. The Agent Repository has encrypted the ITA with its public key (E_1) before it is forwarded to the proxy who works as the post office distributes these ITA(s) to the merchants according to the contact list.

8. Later, the proxy receives a transcript E_1, which is an ElGamal encryption of the ITA using public key of AR. The proxy produces and outputs a transcript E_2, which is an ElGamal encryption of the same ITA, but using a given receiver's public key, which is chosen according to the rules set by the AR. In section [4.1] we describe how to implement asymmetric proxy re-encryption in more detail.

Initiation of bid

1. An ITA implemented as a distributed object executing methods on a number of merchant distributed objects (on remote server systems) to obtain a catalogue of goods/services sold by these merchants. The ITA analyzes all product data and decided whether to buy or sell goods. The ITA then trades with certain merchants.

In this example: ITA trades with Bob by giving x-cash to Bob.

2. On receiving Alice's offer, Bob verifies the correctness of $\sigma\ SK_A\ (w, P)$.

3. Bob evaluates Alice's offer w. (This may involve reading or automatically processing an attached prose description of the offer and /or executing w on possible bids.)

4. Bob executes w on input Q, which is his matching bid. He verifies that the output indicates acceptance of the bid, i.e., that $w(Q) \neq f$ and that the corresponding offer is as desired.

5. Bob obtains from the financial institution F_B a certificate C' bound to a public key PK_B for which Bob holds the corresponding secret key SK_B. (Note that Bob may have to perform this step earlier if ω checks certificate.)

6. Bob create a bid capsule $R = [\sigma\ SK_B\ (W, Q, w(Q)), C¢\ ]$

7. Bob sends R to financial institution F_A

Clearing Process

1. On receiving the first bid capsule with the x-cash coin W, the financial institution F_A reads the policy P in W, verifies that W is correctly formed (that all signatures and certificates are valid), and then stores W.

2. In accordance with the policy P in W, the financial institution F_A collects all valid bid capsules $R_1, R_2,\ldots, R_{m.}$ (containing bids $Q_1, Q_2,..,Q_m$).

3. For each R_i in $\{R_1, R_2,\ldots, R_{m.}\}$, the financial institution F_A does the following:

(a) F_A checks that R_i is correctly formed.

(b) F_A then runs w on the bid Q_i contained in capsule R_i

(c) If $w(Q_i) \neq f$, then F_A checks that Alice has funds worth at least $w(Q_i)$ remaining against the negotiable certificate C. If not, F_A does not process R_i

(d) F_A checks with the appropriate financial institution F_B that there are funds to back the bid $Q_{i.}$ If not, then F_A does not process R_i

4. If Alice has sufficient funds, and there are sufficient funds remaining to support the bid $Q_{i.}$,then F_A and F_B perform the exchange specified by offer and bid.

Performing the exchange

When the two financial institutions, F_A

and F_B have agreed on an exchange as specified by W and some bid capsule R_i, the ownership rights need to be exchanged

Correspondingly, this can be done in a variety of ways, out of which we suggest two:

1. If the same public key is to be used for the newly acquired merchandise, the financial institutions simply re-issue certificates on the public keys corresponding to the new owners of the merchandise. These certificates can then be forwarded by either financial institution to the acquirers, or "picked up" by the same.

2. If a new public key is to be employed, the financial institutions may enter the old public keys of the parties acquiring the merchandise that they certify in a database, and the new owners have to supply a new public key to be certified, and prove knowledge of the secret key corresponding to the old public key in order for the exchange to occur.

4.1 On Quorum Controlled Asymmetric Proxy Re-encryption

An entity with public key y_1 assigns a proxy, agrees with the proxy on rules for re-encryption, and distributes shares of his secret key x_1 to the servers of the proxy. Later, the proxy receives a transcript *E1*, which is an ElGamal encryption of a message *m* using public key y_1. The proxy produces and outputs a transcript *E2*, which is an ElGamal encryption of the same message *m*, but using a given public key *y2*, which is chosen according to the rules set by the entity associated with y_1. The transformation is controlled by the use of quorum actions. Informally, the requirements on our scheme are [2]:

1. **Correctness:** Any quorum *Qr.* of proxy servers, sharing a secret key x_1 will be able to perform the above re-encryption.

2. **Robustness:** If any participant in the transformation protocol would output incorrect transcripts, then this will be detected by all honest participants. The protocol will allow the honest participants to determine what participants cheated, and to substitute these.

3. **Public Verifiability:** Any body must be able to verify that the correct transformation was performed, without having or receiving knowledge of any secret information.

4. **Asymmetry:** The proxy servers must need no information about the secret key *x2* corresponding to the receiver's public key y_2 in order to perform the computation, and the receiver will need no information about x_1 or y_1 in order to decrypt *E2*.

5. **Privacy:** The proxy re-encryption does not leak any information about IAX to any set of proxy servers smaller than a quorum.

4.2 Gradual and Simultaneous Proxy Re-Encryption

The concept of our solution is to use gradual and simultaneous translation of transcripts. The translation is called gradual, since it is performed by quorum action, and each server's contribution to the computation is only a partial translation. We call it simultaneous since each server performs one partial decryption and one partial encryption, outputting such gradual re-encryptions without the cleartext ever being exposed. This approach makes all the partial translations simultaneous in the sense that no result is obtained until all the portions are accounted for.

Let (a_1, b_1) be an ElGamal encryption of a message m w.r.t a public key y_1, and let x_1 be the corresponding secret key, which is shared by the proxy servers using a threshold scheme. The proxy servers wish to compute the ElGamal encryption (a_2, b_2) of *m* w.r.t the public key y_2. They do not wish to

expose m to any set of dishonest proxy servers (or an other set of servers); according to our assumptions, they do not know the secret key x_2 of y_2. For simplicity of denotation, we assume that x_{1j} is the Lagrange-weighted secret key (using the methods in [4] of proxy server j w.r.t a given active quorum *Qr.*; $y_{1j} = g^{x1j}$ is the corresponding public key share. The servers in the quorum perform the following computation:

1. Server j selects a random value d_j uniformly at random from Z_q and computes $(c_j,d_j) = (b_1^{-x1j} y2^{d_j}, g^{d_j})$. This pair is sent to the other proxy servers.

2. The servers (or alternatively, a non-trusted gateway) compute the pair $(a_2, b_2) = (a_1 \tilde{O}_{j \hat{I} Q} cj, \tilde{O}_{j \hat{I} Q} dj)$. The pair (a_2, b_2) is output.

4.3 User authentication

To send instruction to an ITA, Alice must first be authenticated. This is done in the following way; Alice sends the AR her password, client computer Cl_1's network address and the current time. This message (M_1) is first encrypted with Alice's secret key (private key) and then encrypted again with AR's public key. The AR decrypts the message from Alice with its secret key (private key) (this ensures that only the AR can read the message), and then decrypts the result again with Alice's public key (this proves that Alice sent the message). Alice's password is kept inside the AR. This means that only the AR can check if the password is legitimate. If the password and time stamp is valid, then Alice is authenticated and the ITA will accept instructions (via the AR) from Alice at client computer Cl_1 (and only from Cl_1 for this session). A time stamp is valid if it has not been used by user Alice; this mean that the message M_1 can only be used once to authenticate Alice and thereafter it is useless.

4.4 Agent Repository (AR)

In our model, it is also possible to set up the ITA in such a way that it automatically reports back to the AR if certain conditions are met, for example they have already finished trading with the merchants'servers or even in the case that the ITAs decide not to buy or sell goods with some servers. Alice may inspect the ITA to see what goods were bought or sold before contacting her financial institution for performing the exchange.

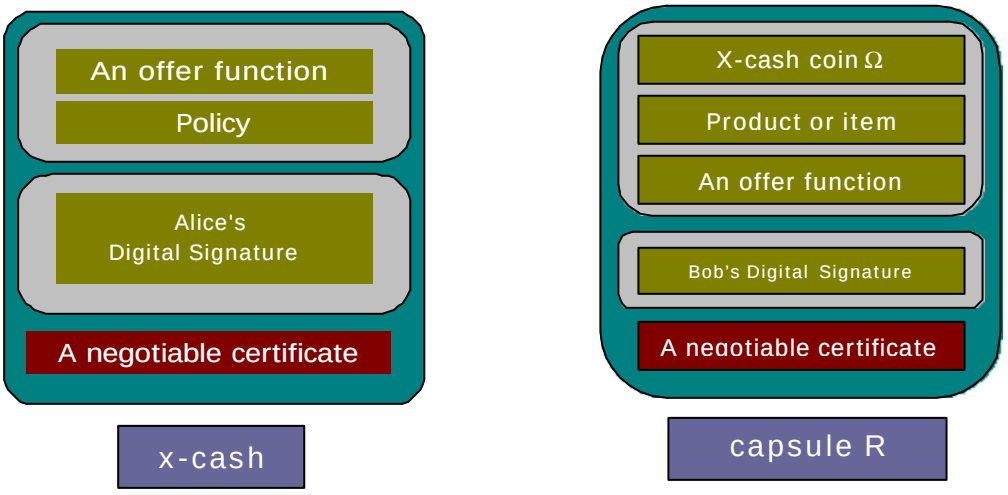

Figure 1: The structure of the x-cash coin W and the capsule R

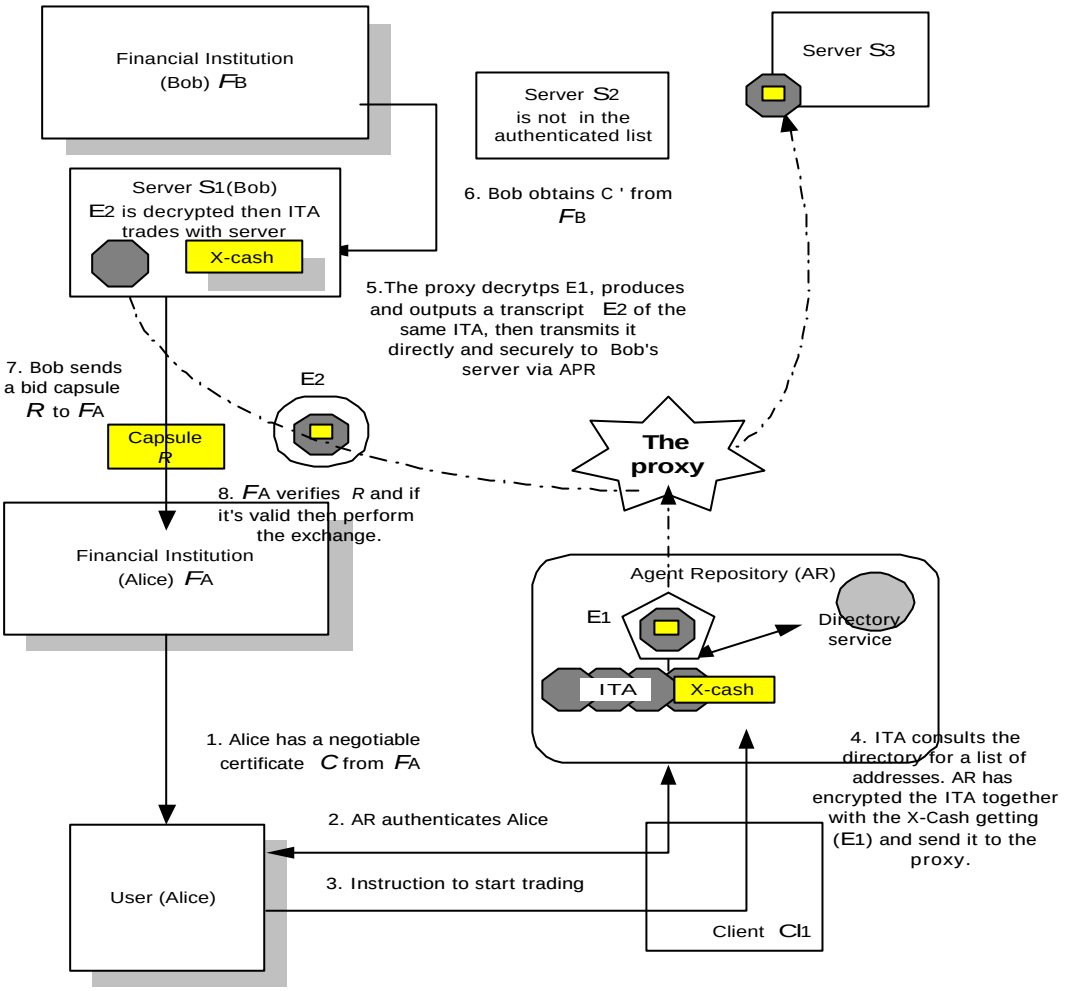

Figure 2: The logical steps in the process / the secure payment mechanism that is used.

The steps in Fig. 2 are as follows.

(1) Alice has a negotiable certificate C from her financial institution F_A, attributing to her rights to all goods or moneys in T, the range of the offer function w to be used in her x-cash. This certificate is issued against public key PK_A for which Alice holds the corresponding private key SK_A.

(2) Before Alice at client computer Cl_1 can issue any instructions to the ITA, it must be authenticated by the Agent Repository.

(3) Alice constructs the x-cash coin Ω containing [σ SK_A (w, P), C] and sets up the ITA by providing it with a list of trade rules (what to buy and sell and under what conditions). All merchants advertise their existence and the products they trade in, at some directory service. When the ITA is instructed to trade, it consults this directory service for a list of merchant distributed object addresses to visit and trade with.

(4) When the ITA is instructed to trade, it consults this directory service for a list of merchant distributed object addresses to visit and trade with. The Agent Repository has encrypted the ITA, which is an ElGamal encryption using its public key before it is forwarded to the proxy.

(5) The proxy produces and outputs a transcript E_2 of the same ITA, then transmits it directly and securely to Bob's server via APR , trading goods using the rules Alice provided. In this step, the ITA trades with server S_1 (Bob). This means that the ITA analyzes all product data and decides whether to buy or sell goods or services. The ITA then trades with certain merchants. The ITA gives the x-cash coin W to Bob. Note that server S_2 address is not in the authenticated list so the ITA has not been sent to. For the server S_3 address is in the list but the ITA doesn't trade with. This might be the merchant's condition and the rules Alice provided have some conflicts. We can say that this multi-cast mechanism would decrease the time overhead of business transactions, because an ITA could visit only the prospected Web sites, which are in the authenticated list. The user need not have to wait for the results because an agent would visit thousands or more of trading servers.

(6) Bob obtains from the financial institution F_B a certificate C' bound to a public key PK_B for which Bob holds the corresponding secret key SK_B.

(7) On receiving Alice's offer, Bob verifies the correctness of σ SK_A (w, P). Bob evaluates Alice's offer w. (This may involve reading or automatically processing an attached prose description of the offer and /or executing w on possible bids.) Bob executes w on input Q, which is his matching bid. He verifies that the output indicates acceptance of the bid, i.e., that $w(Q) \neq f$ and that the corresponding offer is as desired. Bob obtains from the financial institution F_B a certificate C' bound to a public key PK_B for which Bob holds the corresponding secret key SK_B. (Note that Bob may have to perform this step earlier if ω checks certificate.)Bob create a bid capsule R = [σ SK_B (W, Q, $w(Q)$), $C¢$] Bob sends R to financial institution F_A

(8) On receiving the first bid capsule with the x-cash coin W, the financial institution F_A reads the policy P in W, verifies that W is correctly formed (that all signatures and certificates are valid), and then stores W. In accordance with the policy P in W, the financial institution F_A collects all valid bid capsules R_1, R_2,…, $R_{m.}$ (containing bids Q_1, Q_2,..,Q_m). For each R_i in {R_1, R_2,…, $R_{m.}$ }, the financial institution F_A does the following:

(a) F_A checks that R_i is correctly formed.

(b) F_A then runs w on the bid Q_i contained n capsule R_I

(c) If $w(Q_i) \neq f$, then F_A checks that Alice has funds worth at least $w(Q_i)$

remaining against the negotiable certificate C. If not, F_A does not process R_i

(d) F_A checks with the appropriate financial institution F_B that there are funds to back the bid Q_i. If not, then F_A does not process R

(e) If Alice has sufficient funds, and there are sufficient funds remaining to support the bid Q_i,then F_A and F_B perform the exchange specified by offer and bid.

5. Conclusion

This paper has presented secure intelligent trade agents that have been dispatched securely to a network via the asymmetric proxy re-encryption (APR) protocol, collect and analyze data from servers on the network and make decisions to buy and sell goods on behalf of a user. The combination of the x-cash and the APR scheme make these agents secure and efficient intelligent trade agents. The agents have been protected against potentially malicious hosts by public key encryption mechanisms. Furthermore, the agents will visit only the authenticated list of hosts approved by the AR. Last, we believe that our proposal may become well suited as a useful building block in the design of secure agent protocols.

Further work will be related to the security of the intelligent agents and of the hosts that receive them in public networks. The intelligent agents should be protected against potentially malicious hosts. The hosts should also be protected against malicious actions that may be performed by the mobile code they receive and execute.

Acknowledgements

The authors wish to express thanks to Markus Jakobsson and Tet Toe for their many helpful suggestions. We thank Jittin Taisup and an anonymous referee, whose remarks helped to us improve our presentation.

***

References

[1] Markus Jakobsson, Ari Juels “X-cash: Executable Digital Cash” *Financial cryptography’98*

[2] Markus Jakobssson “On Quorum Controlled Asymmetric Proxy Re-Encryption” *PKC’99*

[3] Jaco van der Merwe and S. H. von Solms “Electronic commerce with secure intelligent trade agents” *Computer & Security* Vol.17, No.5, pp.435-446, 1998.

[4] T. P. Pedersen. “A threshold crypto-system without a trusted party”. In D.W. Davies, editor, *Advances in Cryptology – EUROCRYPT’91*, volume 547 of Lecture Notes in Computer Science, pp.522-526. Springer-Verlag, 1991.

[5] A. Shamir, “How to Share a Secret,” *Communications of the ACM*, Vol.22,1979, pp. 612-613

[6] T. ElGamal “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *Crypto’84*, pp.10-18

[7] D. Rus, R. Gray and D.Kotz, "Transportable Information Agents", 1st *Intl. Conf. Autonomous Agents,* 1997.

[8] J. G. Lee, J. Y. Kang, E. S. Lee- "ICOMA: An Open Infrastructure for Agent-based Intelligent Electronic Commerce on the Internet", *Proceedings of the International Conference on Parallel an Distributed Systems*, 1997, p648-655.

[9] M. Blaze, G. Bleumer, M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography", *Eurocrypt'98,* pp.127-144.

[10] B. Venners, "Solve Real Problems with Aglets, a Type of Mobile Agent," *Java world*, May 1997.

[11] M. Jakobsson and M. Yung, "Revokable and Versatile Electronic Money," *3rd ACM Conference on Computer and Communications Security,* 1996, pp76-87.

[12] J. Camenisch, J-M. Piveteau and M. Stadler, "An Efficient Fair Payment System," *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, 1996, pp88-94.

[13] M. Jakobsson, "Ripping Coins for a Fair Exchange," *Advances in Cryptology Proceeding of Eurocrypt'95*, pp220-230.

***