Building Flexible Trust Models for E-Commerce Applications

A. R. Patnaik, A. Srivastava, N. Goel, A. R. Nayak

***Abstract*—A flexible trust model based on human interactions is presented. It involves dynamic evolution of trust between entities (computers, mobile agents) which is independent of the usage of certificates from trusted third parties. Our scheme encourages one-to-one (direct) interactions. A trusted network evolves in a natural way starting from a base-level trust. This trust model is flexible in the sense that each host can define its own trust levels. We believe that our trust model offers a number of advantages over the existing, mostly statically defined trust models.**

I. Introduction

Trust has been identified as an important component in a security infrastructure for communication between two computers or two mobile agents [1,2,3]. These computers can be a part of a network of, either connected or wireless, computers. Trust between the mobile agent and the mobile agent host is required to provide security and protection to mobile agents. It is defined as the expectation of behaviour among parties [4], and also as a subjective probability that is non-reflexive, changing and context-driven [5]. Trust cannot be Boolean and here should be a concept from human experience where trust is earned. In short, trust is used as an additional parameter to enhance security for mobile computing [6, 7]. It is proposed to build trust infrastructures for large ad hoc wireless networks [8]. In this case, one uses both risk and trust to decide whether an interaction can occur. The above trust model has been expanded to define trust within an interval of 0 and 1 as a measure of uncertainty [9].

Trust is a fundamental quality in human relationships in a society. It plays a key role in defining friendship, love, families and organisations. It starts with a certain value between two entities. Independent of how the initial trust comes about, it can grow or reduce depending on mutual interactions and experience. The level of access one entity gives to the other critically depends on the level of trust one has. Moreover, trust is not necessarily reciprocated. The trust management defines a coherent framework and policies in terms of a parameter called trust. There are several trust management systems in the literature [8, 9, 10, 11, 12].

In general, trust management systems delegate permissions by using certificates from trusted third parties. In this paper, a new dynamic parameter is introduced in the authentication process between the agent and the host, called the level of trust. The level of trust an agent has with the host determines the privileges that are granted to that agent. A network of friendly computers can be established after some period of interaction between the computers.

2 BACKGROUND

Some of the parameters that can be used to define trust in mobile agents are that the host is expected to provide correct information with integrity, it should operate correctly, the transactions between the host and the agent must be kept secret, the protocols must be followed correctly and the information is not misused by the agent or the host.

The host H_C defines the privileges and the services it will provide to the visiting agent AG_X depending on the authentication and trust policy processes as shown in Figure 1. A trust model is a tool that helps to visualize and understand the degree of confidence that is granted to individuals, host networks, mobile agents, etc. The more completely the trust model is defined, the greater the awareness of the risks of the threats and vulnerabilities to the system [13].

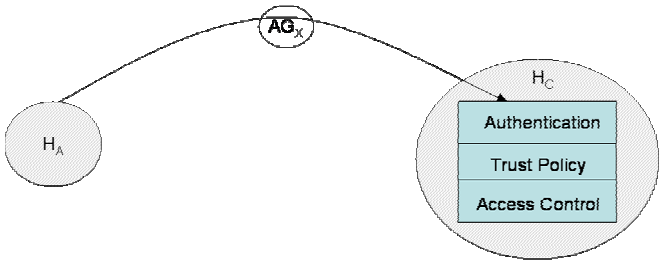

Figure 1 Application of Trust Model

A.R. Patnaik, A. Srivastava and N.Goel are with Tecsis Corporation, 200-210 Colonnade Road, Ottawa, Ontario K2E 7L5 (emails for A.R. Patnaik alok@tecsis.ca , for A.Srivastava alka@tecsis.ca and for N.Goel is nbansal@tecsis.ca)

A. R. Nayak is with School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Canada K1N 6N5 (email: anayak@site.uottawa.ca)

A human trust management model and framework that facilitates the construction of trust aware mobile systems and applications, hTrust, has been proposed [12]. A security model for Aglets based on a set of principles with distinct responsibilities and defined security policies give access to local resources [14]. A distributed access control architecture for mobile agents based on access control lists is proposed and prevents malicious code activity [15]. A model discussed in [16] focuses on the relationship between trust expression, security requirements of the application and the appropriate security mechanisms that are used.

A Dynamic Distributed Trust Model (DDTM) proposes a flexible access control solution for a distributed environment [17]. The core of the DDTM is the recommendation based trust model organized as a Trust Delegation Tree (TDT) and the authorization delegation realized by delegation certificate chains. Trust values for Direct Trust, Indirect Trust and Trust Authorization level are derived.

3 NEW TRUST MODEL

We assume that communication between hosts is by the use of public and private keys for authentication. The trust model is an additional parameter which determines the level of access given by the host. Each host defines its own levels of access depending on the use of its resources. A secure host may allow very limited access while a host of an e-business may allow greater access to its resources.

Trust is usually built up over time, starting from a pre-defined level between two entities that communicate. The fact that the trust level can go up or down leads us to propose a model that is independent of certification, and is based mostly on mutual interaction.

We write host A as H_A and the trust between hosts A and B as T_{AB}.

Assume that H_A communicates with H_B, H_C and H_D. The trust level T, (defined later this section) H_A has with H_B is T_{AB}, with H_C is T_{AC}, with H_D is T_{AD} and so on. Note that the trust level need not be symmetric, i.e. T_{AB} may not be same as T_{BA}. Also, if H_A trusts H_B at T_{AB}, and H_B trusts H_C at T_{BC}, then H_A would trust H_C at (T_{AB} x T_{BC}), whereas if H_A communicates with H_C directly, the trust level is T_{AC} (which may not be same as T_{AB} x T_{BC}). The interactions via neighbours reduce the trust level, and therefore are not desirable. Hence this scheme encourages direct interactions between hosts.

The trust value has the range between 0 and 1 (0 means no trust and 1 means full trust). If H_A trusts H_B at 0.9 level, and H_B trusts H_C at 0.9 level, then H_A trusts H_C at 0.81 level.
This value is used as the initial level of trust between H_A and H_C when the first contact is made. However, if H_A directly interacted with H_C then the start trust level is 0.81 and this can go up or down depending on the results of interaction between them. If this chain of indirect interaction were to increase, then resulting trust level will go down, as it happens in case of human interactions. Assigning a probability-like trust parameter can help building a network of trusted systems.

An example is presented in Figure 2.

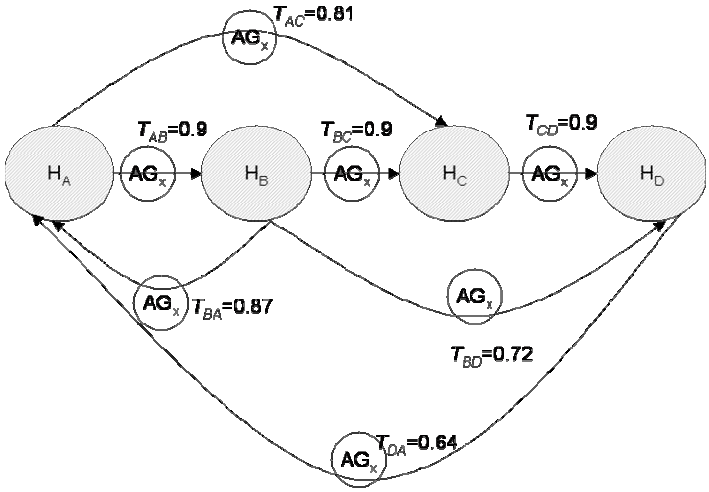

Figure 2 Example of assigning trust levels between hosts

We define the trust metric as:

$$T_{xy} = T_0 + \frac{\sum_{i=1}^{n} w_i E_i}{\sum_{i=1}^{n} w_i} + f(t)$$

where T_0 is the base level trust or the default value of the trust between the computers X and Y. The trust parameter takes values between 0 and 1. E_i are the events or the tasks or services that the visiting computer or agent carries out at the host, w_i are the weights assigned to the events or tasks. The events E_i are recorded in a table kept by the computer H_B. The weights w_i for the event E_i are defined by H_A and each host defines its own weights depending on its needs. The weighted events are summed over n events. Since the weights can be negative, the trust parameter can go down as well.

The last term $f(t)$ is included to reflect any time-dependent activity (or inactivity) to suggest gain or loss of reliability. If the host is not accessible due to network being down or the system itself being down, then the trust level is reduced. We do not, at this stage, associate any trust parameter to the path that the agent may take to reach its destination. The trust level can

go down temporarily if the route to a host is down, or the host itself is down. If this state continues then the reliability of the host goes down. This should be reflected in the trust level. Simulations are needed to generate various scenarios so that *f(t)* can be better defined.

A host H_C, say, sends an agent for malicious activity to H_D. H_C may have defined H_D as a trusted host but H_D may note that H_C is not a trusted host if it can find out malicious activity by its agent. Eventually, after many interactions, H_C would be noted as an untrustworthy host by many hosts. This will ensure reduced interactions with H_C by other hosts. This way, one can develop a network of trusted as well as untrustworthy hosts in a natural way.

Our scheme does not require authentication by trusted third party in terms of certificates. The trust is based on one-to-one interaction and is developed over time. Each destination host should be associated with a measure of its reliability.

4 DETAILS OF THE TRUST MODEL

4.1 Trust Table

Each host must keep a record of the visiting hosts, agents, their activities and determines for itself the trust level of the visiting agent and determines the amount of access it can provide. Two tables are needed as each host, one for sending information and one for receiving information.

The table of the sender host H_S should have information about the number of visits, addresses of the hosts, actions executed, pre-defined trust level, current level of trust etc. The tables must be updated after every visit or communication between two entities. The trust level can be modified to reflect the current situation. These tables are available to be read by hosts with trust level higher than a certain value. If H_A want to communicate with H_E and does not have any information about it in its database, it can look up the tables of its trusted hosts to see if there is an entry for H_E. It can then determine the basic trust level H_E for. If there is no entry, then it assumes the trust level it has for a new host (default value set by T_0) and allows the appropriate access. The contents of the trust table reflect the interactions between the computers, i.e. the tasks that the agent executed at the visiting host.

One can define the operation of the agents as successful or malicious (attempt to access unauthorized information). The trust parameter can be modified accordingly.

A few basic examples are presented here.

Case 1

Agent AG_X is sent from H_A to H_B for the first time and H_A has no information about H_B and H_B has no information about H_A. The trust values would be computed as follows;
$T_{0AB} = T_{0A}$ (the predefined trust level that H_A assigned)
$T_{0BA} = T_{0B}$ (the predefined trust level that H_B assigned)
Host H_A will send Agent AG_X to H_B with trust level T_{0AB} and host H_B will allow agent AG_X to execute at trust level T_{0BA}.

Case 2

Agent AG_X is sent from H_A to H_B for the second time the trust values would be computed as follows;

$$T_{2AB} = T_{0A} + \frac{w_{1A} * E_{1A} + w_{2A} * E_{2A} + w_{3A} * E_{3A}}{(w_{1A} + w_{2A} + w_{3A})}$$

$$T_{2BA} = T_{0B} + \frac{w_{1B} * E_{1A} + w_{2B} * E_{2A} + w_{3B} * E_{3A}}{(w_{1B} + w_{2B} + w_{3B})}$$

Host H_A will send agent AG_X to H_B with trust level T_{2AB} and H_B will allow agent AG_X to execute at trust level T_{2BA}.

A function of time can be added to increase or decrease the trust level depending on the frequencies of visits in a given time.

Case 3

Agent AG_X is sent from H_A to H_B for the first time and H_A has no information about H_B and H_B has information about H_A from a trusted Host H_C.

$T_{0AB} = T_{0A}$ (the predefined trust level that H_A assigned)

$$T_{0BA} = \frac{(w_{1B} * E_{1C} + w_{2B} * E_{2C} + w_{3B} * E_{3C} + \ldots)}{(w_{1B} + w_{2B} + w_{3B} + \ldots)}$$

Host H_A will send agent AG_X to H_B with trust level T_{0AB} and H_B will allow agent AG_X to execute at trust level T_{0BA}.

4.2 Implementation Issues

Our algorithm relies on maintaining the trust table. As the number of users of the machine grows, the table grows linearly. The tables can be sorted in terms of trust value, and entries will be recorded for those sites with a trust value above a given minimum, which is decided by the host. For example, for a host more concerned with security may like to limit the table,

whereas a host in the e-commerce activity may like to keep a larger table.

The security conscious sites would perhaps limit the size of their trust table as they would have a threshold of trust higher than others. The size of the table, and hence the operations on the tables are rather small in such situations and hence the additional computations to search, compute trust values and update the entries are expected to be minimal.

An e-commerce organisation that inherently has a lower threshold of trust for the mobile agents, maintains a larger trust table. However, since the allowed transactions with mobile agents are limited, minimal amount of information may be kept in these tables. Even though the number of entries is large, the volume of the table (in terms of storage size) is not particularly large.

This algorithm, therefore, offers simple computations based on the entries in the trust table. Host sites are free to distribute their weights (the importance of the entries in the trust table) according to their requirement.

5 DISCUSSION

We model our trust based on human experience of trust. The level of access given to the visiting computers/agents depends on the trust level the host has for the visitor (and sender computer). When we are introduced to a new individual, we accept that person with a trust value that we are comfortable with. We may use the trust value of the friend who introduced the new individual to us to determine the initial trust. After the first introduction, the trust between us and the new individual entirely depends on the interactions between the two of us. This trust value can go up or down depending on the nature of our interactions (called events). We may prioritise the events according to our set of rules. Another friend may determine a different trust level from the same set of actions as their requirements may be different (these variations are called the weights).

If we do not know someone, i.e. a complete stranger, then we may do one of the two things. We may give a default trust level to the stranger and start to build up trust from there. We could look up the trust table of our most trusted friends (say, top 10 sites) to see if there is an entry for this stranger in their database. If there is an entry, then we use the actions/events listed in that table and use our weighting system to determine the starting trust level. If no entry can be found in any of our trusted sites, then we use our default trust level (which can be set low for a stranger).

As it happens in a relationship, the trust value is a function of time and the actions of the mobile agent. By defining negative weights for malicious or unexpected activities, one can reduce the trust value. The host should define the magnitude of the negative weight.

Obviously, interaction that is more positive mean more trust. For example, host A gives importance to number visits by an agent. Then a malicious host B could send an agent very frequently to boost its trust level with host A. However, to prevent some system trying to boost the trust value artificially, one can define a functional form of the action. Instead of using the number of visits as a parameter in the trust formula, one could use, say, the logarithm of the number of visits, thereby slowing down the growth of the trust value. In general, the host has the flexibility to select the weights as well as the functional form of the actions to determine its trust level. Note that the weights can be negative as well.

We have introduced a term for inactivity over a period of time. If there is no interaction between two sites then one reduces the trust level. These sites essentially become strangers and the trust value should reflect the current state of their relationship.

A trust management model where it does not depend on a trusted third party for a certificate, rather favours an anarchic model where the agent is responsible for its own fate. This trust model simulates human approach to trust management [12, 17 and 18]. A distributed trust model based on recommendations, organized as a Trust Delegation Tree and the authorisation delegation by certificate chains has been proposed where a direct and an indirect trust parameter is computed [17]. The direct trust depends on the performance and loyalty. Indirect trust depends on the recommendation of the intermediaries, as determined by Dempster-Shafer theory. Past behaviour is used as a predictor of likely future behaviour. In this model, the trusted entity behaves like a stochastic process [18]. Most likely behaviour is determined from the distribution of the behaviour of the trusted entity.

Our trust model differs from all the above models in the following ways.

First, we require neither the certificates nor the recommendations from third parties. The trust in our model is a dynamic and flexible parameter. It is based on one-to-one interactions, i.e. from direct relationships. Trust parameter can go up or down depending on the activity of the trusted entity.

We define trust in most flexible way, giving the control to the host. The fact that we allow the trust parameter to change and since it depends on the past activities on an agent, a table must be maintained at each site to record the activities. It is possible that two tables, one for sending, another for receiving agents, may be maintained. The actions of the agents are recoded in a table (or a database). These actions may have different value or importance to each of the hosts, and therefore, each host defines its own weighting scheme depending on its needs or applications.

The trust table is accessible to other (trusted) hosts. We have proposed a format of the trust table which is open for discussion.

In the long term one can build up a network of trusted sites by using our trust model in a natural way. Our model should work in a network of, both connected and wireless, nodes. It is possible to develop a trusted P2P network out of a set of nodes. We are carrying out simulations to test this scenario.

6 CONCLUSION

Our proposed trust model has a variable trust level that is dynamic and can either increase or decrease over time. The advantages of the proposed model are:

- allows dynamic evolution of trust,
- it is flexible since each host defines its trust parameter,
- information is maintained in a table that is modified only by the host which is accessible to other hosts,
- natural way to build up a network of trusted hosts,
- it is independent of certification from third party.

REFERENCES

[1] M. Witkowski, A. Artikis and J. Pitt, "Experiments in Building Experimental Trust in a Society of Objective-trust Based Agents",*Lecture Notes in Computer Science*, Volume 2246, Jan 2001, pp.111

[2] Y. Wang and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks", p2p , *Third International Conference on Peer-to-Peer Computing(P2P'0*3), 2003, pp.150.

[3] H.K. Tan, and L. Moreau, "Trust Relationships in a Mobile Agent System", *Proc. Fifth IEEE International Conference on Mobile Agents*, 2001, pp. 15-30.

[4] M. Burmester and A. Yasinsac, "Trust Infrastructures for Wireless, Mobile Agents", *WSEAS Transactions on Telecommunications*, January, 2004 Vol 3(1), pp. 377-382.

[5] D. Gambetta, "Can we Trust Trust? Trust: Making and Breaking Cooperative Relations, Basil Blackwell", *Oxford*, 1990, pp. 213-238.

[6] T. Grandison and M. Sloman, "A Survey of Trust in Internet Applications", *IEEE Communications Surveys*, 4th Quarter, 2000, Vol 3, no. 4

[7] L. Kagal, T. Finn and A. Joshi, "Moving from security to distributed trust in ubiquitous computing environments", *IEEE Computer*, Vol 34 (12), 2001, pp. 154-157

[8] V. Cahill, E. Gray, J.-M. Seigneur, C.D. Jensen, Chen Yong, B. Shand, N. Dimmock, A. Twigg, J.Bacon,C. English, W. Wagealla, S. Terzis, P. Nixon, G. Di Marzo Serugendo, C. Bryce, M. Carbone, K. Krukow, M. Nielson, M., "Using trust for secure collaboration in uncertain environments", Pervasive Computing, IEEE, Vol 2(3), July-Sept. 2003, pp. 52 - 61

[9] M. Carbone, M. Nielsen, V. Sassone, "A formal Model for Trust in Dynamic Networks", *Proc of 1st International Conference on Software Engineering and Formal Methods (SEFM'03) Brisbane, Australia,* Sept. 2003, pp. 54-63

[10] R. Yahalom, B. Klein and T.Beth, "Trust relationships in secure systems- A distributed authentication perspective", *Proc. IEEE Symposium on Research in Security and Privacy*, 1993, pp. 150-164.

[11] M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized Trust Management", *Proc. Of IEEE Conference on Security and Privacy, Oakland, CA*, May 1996, pp. 164-173

[12] L.Capra, "Engineering Human Trust in Mobile System Collaborations", *Proc of the 12th International Symposium of the Foundations of Software Engineering*, November, 2004, pp. 107-116

[13] J.A. Stinson, S.V. Pellissier, and A.D. Andrews, "Defining and Applying Generic Trust Relationships in a Network Computing Environment", *ATI IPT Special Report*, May 2000, pp. 00-07.

[14] D. Karjoth, D.B. Lange and M. Oshima ,"A security model for Aglets", *IEEE Internet Computing* , July-August, 1997, pp. 68-77,.

[15] N. Antonopoulos and K. Koukoumpetsos, K. Ahmad , "A Distributed Access Control Architecture for Mobile Agents", *Proc. of the International Network Conference,* July 2000.

[16] J.T. McDonald and A. Yasinsac, "Trust in Mobile Agents", *Technical Report Florida State University, Department of Host Science*, 2004. http://www.cs.fsu.edu/research/reports/TR-050330.pdf

[17] L. Hui and G.C. Shoja , "A Distributed Trust Model for e-Commerce Applications", *Proc of IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)* **,** 2005, pp. 290-293.

[18] J. Shi, G.V. Bochman, and C. Adams, "A trust model with Statistical Foundation, Workshop on Formal Aspects in Security and Trust" *(FAST'04) Toulouse, France 18th IFIP World Computer Congress*, Aug 6-27, 2004, pp. 169-181.

[19] J.Ioannidis and A.D. Keromytis, "Distributed Trust", *Practical Handbook of Internet Computing*, 2004.

[20] A. Abdul-Rahman and S. Hailes, " Supporting Trust in Virtual Communities" *Proc. of the 33rd Hawaii International Conferences on System Sciences*, Volume 6, 2000.