Evaluation of the Iterative Multiplication Strategy for Trust Propagation in Pervasive Environments

Omar Hasan
INSA Lyon, France
omar.hasan@insa-lyon.fr

Lionel Brunie
INSA Lyon, France
lionel.brunie@insa-lyon.fr

Jean-Marc Pierson
IRIT, France
pierson@irit.fr

ABSTRACT

There are a number of models for access control in pervasive environments that are based on trust propagation. Iterative multiplication of the trust values on a path from a source entity to a target entity is one of the common strategies for trust propagation. In this paper, we evaluate the effectiveness of iterative multiplication for trust propagation. The data set used for this evaluation is the real web of trust of Advogato.org that comprises of over $11,000$ vertices (users) and over $50,000$ directed weighted edges (trust relationships between users). We find that a significantly strong positive linear correlation exists between trust values based on direct experience and the corresponding propagated trust values derived through the iterative multiplication approach. This finding provides empirical support for the access control models for pervasive environments that employ the iterative multiplication strategy for trust propagation.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*

General Terms

Human Factors, Security, Algorithms, Experimentation

Keywords

Trust, Trust Propagation, Access Control, Pervasive

1. INTRODUCTION

Pervasive or ubiquitous environments are characterized by seamless access to resources by users at foreign sites as well as at their home site. Access control for such an environment is inherently challenging since sites may have no knowledge of remote users who visit them. Traditional access control models based on roles (as in RBAC [2]) or identities (as in IBAC [4]) are often not suitable for such environments since roles or identities that exist at one site may not exist at another.

A number of works, which include [1, 3, 5, 6, 9, 10], have introduced solutions based on trust to access control in pervasive environments. The proposal is to consider the trustworthiness level of users as the criterion for granting access to resources. Ubiquitous access control is possible with trustworthiness since it is a universally recognized notion.

A site has direct knowledge of the trustworthiness of its own users. Whereas, the trustworthiness of an unknown user can be determined through trust propagation. Trust propagation is a technique that enables the foreign site to acquire trust in the unknown user through a path of trust recommendations that link the site to the user. For example, a site X may acquire trust in an unknown user u, if u's home site Y which is trusted by X, makes a recommendation to X about u.

In several models of access control for pervasive environments (including Hasan et al. [3] and Saadi et al. [9]), propagated trust is computed by iteratively multiplying the trust values on the path from a source entity to the target entity. In this paper we first summarize our access control model [3]. Then as the main contribution of this paper, we perform an experiment to determine the effectiveness of iterative multiplication for trust propagation.

The data set used for the experiment is the real web of trust of Advogato.org that comprises of over $11,000$ vertices (users) and over $50,000$ directed weighted edges (trust relationships between users). Results show that a significantly strong positive linear correlation exists between trust values established from direct experience and propagated trust values derived through the iterative multiplication approach. To the best of our knowledge, this is the first work to provide evidence of this correlation based on a real and large web of trust.

The rest of the paper is organized as follows: The next section outlines the problem setting. In section 3, we reproduce our access control model for pervasive environments. section 4 comprises of the experiment, results, and analysis. We present concluding remarks in section 5.

2. PROBLEM SETTING

The environment comprises of n sites given as the set $S = \{s_1, s_2, \ldots, s_n\}$. A site is defined as a geographically bounded collection of resources with an autonomous administration and access control policy. Some examples of sites include university campuses, corporate offices, airports, etc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICPS'09, July 13–17, 2009, London, United Kingdom.
Copyright 2009 ACM 978-1-60558-644-1/09/07 ...$10.00.

Each site has a number of member users associated with it. The set of users associated with a site x is given as $U_x = \{u_{x,1}, u_{x,2}, \ldots, u_{x,|U_x|}\}$, where $|U_x|$ is the number of users. For simplicity we assume that the set of users of any two sites x and y are disjoint, that is $U_x \cap U_y = \phi$.

Each site also has a number of resources under its ownership. The set of resources of site x is given as $R_x = \{r_{x,1}, r_{x,2}, \ldots, r_{x,|R_x|}\}$, where $|R_x|$ is the number of resources. A user may request access to a resource at his home site or he may roam in the environment and request access to the resources of foreign sites. Each site has an access control policy that determines if a user is qualified to access a resource that he has requested.

The goal is to make the access control process for a user as ubiquitous at a foreign site as it is at his home site.

3. THE ACCESS CONTROL MODEL

In this section we summarize our model [3] for access control in pervasive environments that uses the iterative multiplication strategy for trust propagation. This model provides the context and the framework for the experiment in the next section.

3.1 General Framework

We define a set, $V = \bigcup_{x \in S} U_x \cup S$. The set V contains all the users and all the sites in the environment.

We define a binary relation, $T = \{(x, y) : x \in S \wedge y \in V\}$. The relation T represents the *trusts* relation between a site and another site or a user. We will use the notation $x \ \ T \ \ y$, $x \ \ trusts \ \ y$, and (x, y) interchangeably.

A Web of Trust is defined as a weighted directed graph, $G = (V, T)$. The sites and their users form the vertices of the graph. The trust relations between the members of set V given as ordered pairs in the set T form the edges of the graph. An edge that is incident from x and incident to y, implies (x, y) or $x \ \ trusts \ \ y$.

A weight is associated with every edge (x, y) in the graph, which represents the amount of trust that entity x holds for entity y. The weight associated with an edge (x, y) is given as the function $t(x, y)$. $t : T \rightarrow X$. The set X is defined as $X = [0, 1]$.

The range of $t(x, y)$ is real numbers bounded by 0 and 1. 0 implies "minimum trust" and 1 implies "maximum trust". Real numbers between 0 and 1 give us infinite resolution for expressing trust.

(x, y) exists for all x, y where $x \in S$ and $y \in U_x$. This implies that a site has direct trust relationships with all of its users.

A path $p = \langle x_1, x_2, \ldots, x_m, u \rangle$ from a site x_1 to a user u is said to exist if $x_1, x_2, \ldots, x_m \in S$ and $u \in U_{x_m}$ and $(x_1, x_2), (x_2, x_3), \ldots, (x_{m-1}, x_m), (x_m, u) \in T$.

3.1.1 Trust Recommendation and Propagation

If $(x_1, x_2), (x_2, x_3), \ldots, (x_{m-1}, x_m), (x_m, u) \in T$, then $t(x_2, x_3)$, $t(x_3, x_4)$, $\ldots$, $t(x_{m-1}, x_m)$, $t(x_m, u)$ may be considered as recommendations to x_1 from x_2, x_3, $\ldots$, x_{m-1}, x_m respectively. Taking into consideration this "chain of trust", x_1 may choose to establish (x_1, u) and $t(x_1, u)$. We say that the trust of x_m in u is propagated to x_1.

To facilitate the discussion we establish the following terminology:

Source site – the site from which the path originates; the site that may establish trust in a previously unknown user based on a recommendation

Recommender site – a site that recommends a site or one of its users to the source site

Target user – the user at whom the path terminates; the user whom the source site may choose to trust

In the preceding case, x_1 is the source site, $x_2, x_3, \ldots, x_m$ the recommender sites, and u the target user.

3.1.2 Access Control

With each resource $r_{x,j}$, the site x defines a threshold value which is given as the function $h(r_{x,j})$. $h : R_x \rightarrow X$. The access control policy of a site lists all its resources and associated thresholds.

Access is granted to a user u that requests a resource r at a site x if $t(x, u) \geq h(r)$. In other words, access to a resource is granted if the site has equal or greater trust in the requesting user than the threshold for that resource.

It is important to note that the user u may or may not be a member of site x. If u is a member of site x then the site has direct knowledge of the user's trustworthiness. In case u is not a member then access may still be granted if $t(x, u)$ can be established through trust propagation and $t(x, u)$ passes the trustworthiness threshold.

What makes the model ubiquitous is that a site does not need to have pre-defined access rights for a certain user to be able to grant them access to resources. The site can establish trust in a previously unknown user through trust propagation and it can grant them access based on that acquired trust. From the user's point of view access to resources at foreign sites is as seamless as at their home site.

3.2 Trust Propagation

3.2.1 Trust Propagation Function

We define a function *ptrust* (abbreviation of "propagated trust") that given a path $\langle x_1, x_2, \ldots, x_m, u \rangle$, suggests a weight for the edge (x_1, u). The value suggested by the function is an estimate of the amount of trust in u that may propagate to x_1.

$$\begin{aligned} t(x_1, u) &= ptrust(\langle x_1, x_2, \ldots, x_m, u \rangle) \\ &= t(x_1, x_2) \times t(x_2, x_3) \\ &\qquad \times \ldots \times t(x_{m-1}, x_m) \times t(x_m, u) \\ &= \prod_{i=1}^{m-1} t(x_i, x_{i+1}) \times t(x_m, u) \end{aligned} \tag{1}$$

3.2.2 Reasoning for Using Multiplication

The suggested propagated trust value is the product of all the trust values on the path. We implement the function as such for its simplicity and intuitiveness. We consider a few examples to illustrate our point.

Let's assume that all the trust values on the path are 1. The trust value suggested by the function in this case would be 1, which reflects the fact that absolute trust exists throughout the chain.

As another case let's consider that any one or more of the trust values on a path are 0. That is, one of the sites has no trust in the entity that it has a trust relationship with.

The trust value suggested by the function would be 0. Thus the fact that one of the sites does not trust an entity on the path is appropriately reflected in the suggested value.

Let's now consider a path of length 3 with each of the trust values as 0.9. The suggested trust value would be $0.9 \times 0.9 \times 0.9 = 0.73$. Although each of the sites has a high trust of 0.9 in the recommended site or user, the suggested trust value is a lower 0.73. This value is reflective of the degree of separation between the source site and the target user. Intuitively, trust attenuates as the degree of separation between the source site and the target user grows.

As the final example we consider the path $\langle x_1, x_2, x_3, u\rangle$ with $t(x_1, x_2) = 0.1$, $t(x_2, x_3) = 0.8$, and $t(x_3, u) = 0.9$. The suggested trust value would be $0.1 \times 0.8 \times 0.9 = 0.07$. Although x_2 and x_3 have very high trust in x_3 and u respectively, since x_1 has low trust in x_2, the propagated trust value remains low.

4. EXPERIMENT

The objective of this experiment is to determine whether it is prudent to establish trust in an unknown entity based on trust propagation. More precisely, whether a strong positive correlation exists between direct trust and propagated trust. Direct trust is the amount of trust that a source agent establishes in a target agent based on direct experience.

4.1 Correlation

Correlation is a coefficient that measures the strength and the direction of the linear relationship between two variables.

The correlation coefficient lies on the interval $[-1, 1]$. Values near 1 and -1 indicate a strong linear relationship between the two variables. Values close to 0 indicate a weak relationship. A positive value implies that the relationship is proportional, that is, increase in the value of one variable is likely to result in the increase of the value of the other variable. A negative value implies an inverse relationship.

The correlation r between two variables x and y is given as follows:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right) \qquad (2)$$

where, $\bar{x}$ and s_x are the *mean* and the *standard deviation* of variable x, and n is the size of the bivariate data.

Correlation computed with this specific method is also known as the Pearson Product-Moment Correlation.

4.2 Experiment Design

The weight of an edge from a source vertex to a target vertex represents the direct trust that the source vertex holds for the latter. If an alternate path exists from that source vertex to the target vertex, we can compute the propagated trust between the two vertices from that path. Based on these observations we design the experiment as follows:

We consider every edge in a given web of trust. An exception is those edges that have the same source and target vertex. A web of trust such as Advogato may have such edges but we leave them out as they do not conform with our model of a web of trust.

The direct trust of an edge's source vertex in its target vertex is the weight of that edge. Having noted the direct trust from the source vertex to the target vertex, we consider the scenario that direct trust between the two vertices does not exist. We remove the direct edge and using Dijkstra's algorithm find an alternate path from the source to the target. If an alternate path exists, we obtain the propagated trust using the *ptrust* function. Now we know the direct trust of the source vertex in the target vertex as well as the propagated trust. After obtaining all such pairs of direct trust and propagated trust, we calculate the correlation between the two variables. It is important to note that the values of direct trust and propagated trust are obtained independently of each other in this experiment.

The Dijkstra's algorithm may return several alternate shortest paths. In this experiment, we always consider the first path that is returned by the algorithm. As future work, a variation on the experiment could be to identify and select the path that yields the optimal trust value. Such a path may or may not be the shortest one.

The experiment is algorithmically described in Figure 1. G is a web of trust. $dijkstra(x, y)$ is a function which returns a path from vertex x to vertex y, given as $p(x, y)$, using Dijkstra's shortest-path algorithm. $correlation(\overrightarrow{\mathcal{X}}, \overrightarrow{\mathcal{Y}})$ is a function which returns the correlation r between two variables represented by vectors $\overrightarrow{\mathcal{X}}$ and $\overrightarrow{\mathcal{Y}}$.

The experiment has been implemented using the Java Graph library (JGraphT).

```
EXPERIMENT-1(G)
 1  i ← 0
 2  for all edges in G, whose source vertex
                (given as a_s) and target vertex
                (given as a_t) are not the same
 3        do direct-trust ← t(a_s, a_t)
 4           remove the edge (a_s, a_t)
 5           p(a_s, a_t) ← dijkstra(a_s, a_t)
 6           if |p(a_s, a_t)| > 0
 7              then prop-trust ← ptrust(p(a_s, a_t))
 8                   add direct-trust to vector D
                           at index i
 9                   add prop-trust to vector P
                           at index i
10                   i ← i +1
11           restore the edge (a_s, a_t)
12  r ← correlation(D, P)
13  print r
```

Figure 1: Experiment Design.

4.3 Data Set

The data set that we use for our experiment is the real web of trust of Advogato.org [7, 8]. Advogato.org is a web-based community of open source software developers. A major focus of the site is a peer rating system. The members of the site rate each other in terms of their trustworthiness. The choice of trust values are *master*, *journeyer* and *apprentice*, with *master* being the highest level in that order. The result of these ratings among members is a rich web of trust, which comprises of 11,558 users and 51,119 trust ratings. The distribution of trust values in the Advogato web of trust is as follows: *master*: 17,478, *journeyer*: 22,894, and *apprentice*: 10,747.

The instance of the Advogato web of trust referenced in this paper was retrieved on November 19, 2007 by crawl-

ing the Advogato.org web site with a script that we wrote in Python. To conform the Advogato web of trust to our framework, we substitute its three trust values as follows: $master = 1.0$, $journeyer = 0.66$, and $apprentice = 0.33$.

The Advogato web of trust may be viewed as a directed weighted graph, with users as the vertices and trust ratings as the directed weighted edges of the graph. The number of vertices with no outgoing edges is $5,832$ and the number of vertices with no incoming edges is $5,548$.

4.4 Experiment Runs and Analysis

We run the experiment with the adapted Advogato web of trust as G. The number of instances when an alternate path was found between two vertices with a direct edge is $44,959$. The final value of i in the algorithm of the experiment gives this value. A histogram of the lengths of the alternate paths is given in figure 2. A scatterplot of the direct trust values and the corresponding propagated trust values is given in figure 3. The outcome of the experiment, the correlation between direct trust and propagated trust, is 0.61 (rounded down to two decimal places).

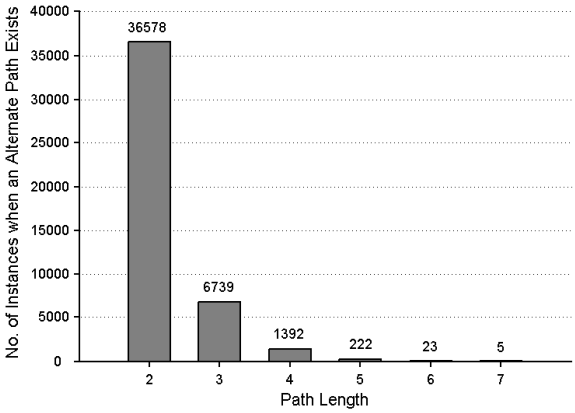

Figure 2: Histogram of Path Lengths.

The histogram shows that the edge count is at most 3 for over 96% of the instances when a path is found from the source vertex to the target vertex. As we discussed in section 3.2.2, fewer edges on the path between two entities leads to lower attenuation of trust propagated over that path. The observation thus implies that a high percentage of the propagated trust values have low attenuation.

The experiment provides evidence that a significantly strong positive linear correlation (0.61) exists between direct trust and propagated trust acquired through the iterative multiplication approach. We note again that the values of direct trust and propagated trust are obtained independently of each other in the experiment. This result is significant since the data set used is a real and large web of trust.

To the best of our knowledge, this is the first work to provide evidence of a strong positive linear correlation between direct trust and propagated trust (acquired through iterative multiplication) based on a real and large web of trust.

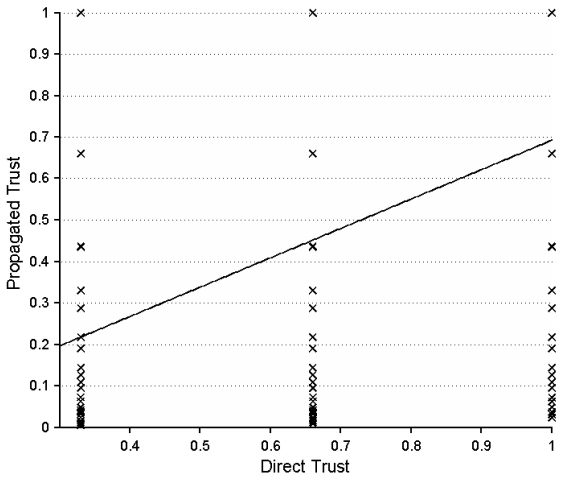

Figure 3: Correlation between Direct Trust and Propagated Trust.

5. CONCLUSION

In this paper we analyzed the iterative multiplication strategy for trust propagation employed by various access control models for pervasive environments. Through an experiment on the real and large web of trust of Advogato.org, we showed that a significantly strong positive linear correlation exists between direct trust and propagated trust acquired through the iterative multiplication approach. This result raises confidence in the notion of establishing trust in an unknown entity through the said trust propagation method.

6. REFERENCES

[1] S. I. Ahamed, M. M. Haque, and N. Talukder. Service sharing with trust in pervasive environment: Now it's time to break the jinx. In *Proc. of the 23rd Annual ACM Symposium on Applied Computing (SAC 2008)*, Fortaleza, Ceará, Brazil, March 2008.

[2] D. Ferraiolo and R. Kuhn. Role based access control. In *Proceedings of the 15th National Computer Security Conference*, pages 554 – 563, October 13 - 16 1992.

[3] O. Hasan, J.-M. Pierson, and L. Brunie. Access control for ubiquitous environments based on subjectivity eliminated trust propagation. In *Proc. of the 3rd Intl. Symposium on TRUST, in conjunction with the 2008 IEEE/IFIP Intl. Conf. on Embedded and Ubiquitous Computing (EUC 2008)*, Shanghai, China, December 2008.

[4] HP. Identity-based access control. Technical report, Hewlett-Packard, 2006. ProCurve Networking by HP.

[5] L. X. Hung, P. D. Giang, Y. Zhung, T. V. Phuong, S. Lee, and Y.-K. Lee. A trust-based security architecture for ubiquitous computing systems. In *Proc. of the IEEE Intl. Conf. on Intelligence and Security Informatics (ISI 2006)*, San Diego, CA, USA, May 2006.

[6] L. Kagal, T. Finin, and A. Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, December 2001.

[7] R. Levien. Attack resistant trust metrics. Manuscript,

University of California - Berkeley. www.levien.com/thesis/compact.pdf, 2002.

[8] R. Levien and A. Aiken. Attack-resistant trust metrics for public key certification. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, Texas, January 26-29 1998.

[9] R. Saadi, J.-M. Pierson, and L. Brunie. Authentication and access control using trust collaboration in pervasive grid environment. In *Proceedings of the International Conference on Grid and Pervasive Computing (GPC 2007)*, 2007.

[10] K. Shin and H. Yasuda. Practical anonymous access control protocols for ubiquitous computing. *Journal of Computers*, 1(8), December 2006.