On Trust for Ubiquitous Computing

Narendar Shankar[1]
William A. Arbaugh
{*narendar,waa@cs.umd.edu*}
Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract

As we move into a world of ubiquitous and pervasive computing, there is an increased interaction between people and smart devices, which have computing power. In such a world, computing power is moving from big desktops to very small and miniature devices and there is a seamless integration of computing power and day-to-day life. For such a world of computing, we believe that there is a need for a continuum of trust, which models the real world, as closely as possible. In other words, we need to capture the real world model of trust where entities (people and devices) trust each other to varying degrees and extents.

Moreover, we believe that in the world of pervasive computing, an entity's physical context (which could be the location of the entity, or even a property like time) is an important factor in modeling trust (because of ad-hoc interactions). In other words, we need a unified model of trust relationship between entities, which captures both the needs of the traditional world of computing (where the continuum of trust is based on identity) and of the world of ubiquitous and pervasive computing (where the continuum of trust is based on identity, physical context or a combination of both). In this paper, we present a novel attribute vector calculus based approach for modeling the continuum of trust.

1. Introduction

Computing devices are becoming ubiquitous in our daily lives. The rapid decrease in the size and cost coupled with an increase in capability has permitted a rapid proliferation of small and very capable devices into our daily lives. As these devices become better connected, we finally have the basic *building blocks of smart environments* available [1,2]. In such a world of ubiquitous and pervasive computing, *trust modeling and management* has once again surfaced as one of the major problems to solve.

Until now, trust management has been studied from the perspective of establishing security policies and security credentials and in determining whether credentials match policies. The core philosophy of trust management remains the same, but for the world of pervasive computing, we perceive the need for newer models of trust. Specifically, we believe that there is a need for a *continuum of trust*, which models the real world, as closely as possible and is built upon a formal

[1] This work has been sponsored by a Critical Infrastructure Protection Grant from the National Institute of Standards and by Fujitsu Labs America ®.

basis. In other words, we must be able to formally represent trust in a realistic fashion, where entities (people and devices) trust each other to varying *degrees and extents*.

An interesting aspect of establishing trust relationships in the real world is that many of the real world scenarios attempt to establish trust, with minimal prior trust relationships, i.e. many real world scenarios attempt to establish *ad-hoc* trust relationships. We believe that in the world of pervasive computing, an entity's *physical context* [3,4] (which could be the location of the entity, or even a property like time) could be used to enable such ad-hoc trust relationships. Our goals in defining a new model of trust are first to provide a model capable of *also* modeling scenarios where identity may not be available and the context of the scenario is more appropriately used in establishing trust relationships, and second to design a model that provides a formal basis for making trust decisions.

Another important consideration for such a model is compatibility with the needs of the traditional world of computing (based on identity). That is- we need a model of trust between entities, which captures both the needs of the traditional world of computing (based on identity) and of the world of ubiquitous and pervasive computing (based on identity, physical context or a combination of both). In essence, we need a ***unified model*** for representing trust relationships between entities. Moreover, this unified model should be capable of representing the continuum of trust relationships in both worlds. In this paper, we present a novel *attribute vector calculus* based approach for modeling the continuum of trust.

2. Problem Definition and Analysis

Before we analyze the problem formally, let us look at what trust means and what identity-based trust and context based-trust mean.

2.1 On Trust and the Continuum of Trust Relationships

Trust modeling involves expressing trust relationship between entities. Trust requires some explanation. One of the definitions of trust *is qualified reliance on received information.* Another definition of trust is *`Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action'*.

Trust relationships are usually based on *identity*. Examples of such systems include most traditional authentication systems, where trust is established using shared secrets, public/private methods and certificates. Even in identity-based trust, there are many categories of trust like weak trust, strong trust and so on. In fact, there is a whole set of discrete trust relationships in identity-based trust relationships. This set of discrete trust relationships can be expressed formally as a *continuum of trust relationships*, where each relationship has a different degree of trust involved in it.

In the real world, we can find many examples of trust establishment between entities, which go beyond identity and use *physical context*. Some examples will make this clear

Example 1: A customer goes to Starbucks Coffee and uses their wireless network, which is provided free of cost. Starbucks would want to ensure that the customer is in their physical

premises because that means that they will also buy coffee (which means money as far as Starbucks is concerned). In other words, Starbucks is more concerned about the location of the user rather than his identity. Such a scheme is called *location authentication*, which does not need any prior shared secrets or public/private key agreement. Here location acts as the contextual parameter.

Example 2: In the future, personal devices will be able to talk to each other and form *Personal area networks (PANs)*, or *Body Networks*. Trust relationships will be of the form, where each device in the body has to establish trust with other devices based on physical context, where the physical context here is that all the devices have to be in the same body. Classical trust relationship mechanisms based on identity can also be used here but will probably be too heavy weight for such devices, especially if the devices communicate with each other for small synchronization events, where the only concern for each device is that the recipient of the synchronization event is a device, which is in the same physical body.

Again, for context-based trust mechanisms, a trust model must be capable of capturing the entire continuum of context-based trust relationships.

Example 3: For a simple location authentication system granting access to some information, an entity in location A (which is a room which needs a key) can be trusted more than an entity in location B (which is an open lobby), though both of them are in locations. Extrapolating on similar lines, one can envision a continuum of context-based trust relationships.

Context-based trust mechanisms *enable* establishment of *ad-hoc* trust relationships because a context is an inherent property of the environment, in which the entity is rather than an inherent property of the entity itself (like identity).

2.2 Our Trust Model

We would like to express the continuum of trust for both identity-based trust and context-based trust in a formal manner and more importantly in a *unified model*. In other words, the model must be capable of capturing both the identity-based trust relationships and context-based trust relationships in a single model. We have chosen an *attribute vector model* for modeling trust relationships between entities. The reason for doing so is that the attribute vector model captures both the identity-based and context based trust relationships in a simple and expressive manner.

Let us assume that a group of **m** entities wants to establish a trust chain/web with/without any prior established trust.

- **The Attribute vector**

  Let $S_1, S_2 \ldots S_m$ *denote the m entities.* An entity S_i has the attribute vector $A(S_i)$. $A(S_i)$ is a vector of k individual attributes. It can be formally defined as

  - $A(S_i) = < A_{i1}, A_{i2} \ldots A_{ik} >$

  $< A_{i1}, A_{i2} \ldots A_{ik} >$ is a vector of attributes for the entity S_i. This vector of attributes is used to model both the traditional world of computing, where trust relationships are based on identity and also the new world of ubiquitous computing, where trust relationships are based on

context. For the former, this vector of attributes represents ***credentials of the entity*** S_i and for the latter it represents the ***context of the entity*** S_i. The word *context* represents physical world information like location, time etc.

- **The continuum of trust and a decentralized model**

  The trust relationships between entities can be expressed as follows

  - $D(S_i, S_j) = f(A(S_j))$

  The above states that the degree of trust (also called *trust value*) upon an entity S_j is a function of the attributes of the entity. The function uses numeric, alphanumeric and semantic orderings of various attributes to compute the trust value. We do not describe the details of the function for the sake of brevity. The interesting things to note here are the following

  1. S_i *and* S_j trust each other to varying degrees and this degree of trust varies for each pair {i,j}. This result is a continuum of trust relationships.
  2. The trust model is totally *decentralized* where each entity is trusted in a possibly totally manner by all the other entities.
  3. S_i *and* S_j *may or may not have any prior trust relationship.*

  Once the trust value for an entity has been calculated, a trust decision can be made if the trust value is beyond a certain threshold (*t*)

  - $T(S_i, S_j) \Leftrightarrow D(S_i, S_j) > t$

  The above states that there is a trust relationship between entities S_i *and* S_j *iff* the degree of trust upon S_j is greater than a threshold *t*.

3. Translation of the model into real world examples

3.1 The Unified Model and the Real World

Our goal was to build a ***unified*** trust modeling system, which could capture the traditional trust requirements of a cyber world, which are based on identity and also capture the requirements of a new world of ubiquitous computing, which will be based on a mixture of identity and context.

As shown in figure 1, the whole world is treated as one big distributed system as far as we are concerned. This world consists of a mixture of entities, which interact purely in cyber space (entities in the world of traditional internet/network based computing) and of entities, which interact directly, which is a flavor of the world of ubiquitous computing. An example of such an interaction would be devices in the body talking to each other over short-range channels.

As shown in the figure, we abstract the interaction between entities in a novel manner using a simple attribute vector. Whether the entities interact in cyberspace or directly, the attribute vector

is a common characteristic. As shown in figure 2, some entities have overlapping attributes and some entities have distinct attributes.

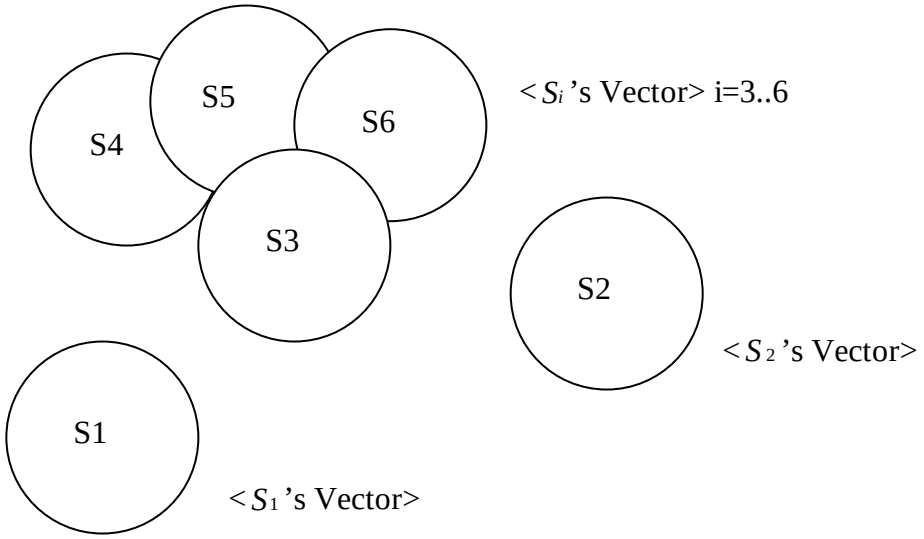

Figure 1. Attribute Vector representation for the World

Modeling trust is now a simple issue. Whether entities interact in the real world or in cyber space, the model described previously can be used to form trust relationships. This is because of the attribute vector abstraction. For cyberspace, the attribute vector is treated as a vector of *credentials* and for the ubiquitous world, the attribute vector is treated like a *context tuple*, representing the real world contextual parameters like location. We could also have a *combination* of credentials and contextual tuples in an attribute vector. We present two examples, one in the traditional world of cyberspace and one in the ubiquitous computing to illustrate the unified nature of the model.

3.1.1 Traditional authentication models

Any authentication procedure tries to bind a principal and its identity. Let us take the example of an authentication system like Kerberos [5]. Here two entities (call them A and B) want to mutually authenticate themselves. They do so by means of a trusted third party. In such a process, there are quite a few trust relationships involved. Modeling these relationships using our model is simple. Entities A and B have some credentials (like some shared secrets), which can be represented by some attribute vector. The trusted third party has a set of credentials, which can also be represented by another attribute vector. The trusted third party has a higher trust value than either A or B because the attribute vector of the trusted third party (its credentials, which happens to be keys possessed by the trusted third party) is of a higher order than A's or B's. If there are multiple trusted third parties, then each one has a particular trust value, which depends on the trusted third party's credentials (which here might be the keys, which the trusted third party has). In such a, manner there is a simple mapping between the credentials and the trust values, which in turn determines the level or degree of trust.

Another simple example is the use of the model to map entities using the Extensible Authentication Protocol (EAP) [6], where the trust relationships between entities can be modeled

using the authentication parameters used (EAP is a mechanism to support multiple authentication methods). The authentication parameters form the attribute vector for each entity and each entity's trust relationship with other entities is based on the authentication parameters, which the entity possesses.

3.1.2 Authentication models in the ubiquitous world

Location authentication tries to bind a principal to a context (which is the location of the entity). A simple location authentication system uses parameters like physical location for making security decisions. In a location authentication system, the primary interest for an authenticator is the location of the supplicant (entity requesting to be authenticated). Location forms the basis for trust relationships in such an authentication model. Each entity at a different location has a different attribute vector, which is primarily composed of location attributes (like coordinates, for example). Based on the ordering of their attribute vectors (which happens to be contextual information like location), the entities have different trust values. Thus, each of the entities can be trusted to a different extent based on the attribute vector, which it has.

3.2 Expressing ad-hoc trust relationships

The above model can be used to express ad-hoc trust relationships. Let us assume that two entities A and B want to establish some ad-hoc trust relationship. Each of the entities A and B has its corresponding attribute vectors. The ad-hoc trust relationship can be established using some *common* attribute (or a set of common attributes), which is more of an inherent property of the environment, rather than an inherent property of the entity. One would expect such an attribute to be an implicit attribute or a context-based attribute (like location).

The simple idea behind establishing an ad-hoc trust relationship is the fact that two entities have to agree upon some *common ground* for trust, which can be expressed as an attribute or a vector of attributes.

4. Related Work

Trust management has been studied in great detail until now. PolicyMaker [7] serves as a toolkit, which applications can use to incorporate trust management. PolicyMaker makes its decisions based on the presented key and an application specific action. Similarly, KeyNote [8] serves as a toolkit for public key infrastructures. Trust management and access control has also been studied in depth [9,10]. In this paper, we *do not* deal with the issue of trust management but instead deal only with *trust modeling.*

There have been many models and definitions of trust, which have been proposed previously but we have not encountered a unified model, which captures the continuum of trust relationships (for both the virtual world of computing and for the real world of ubiquitous computing including ad-hoc trust relationship scenarios). The main reason for this is the fact that the unified model is a new requirement, which has been necessitated by the emergence of the world of ubiquitous computing.

The work by Yahalom et.al. [11] proposes a formal definition of trust based on seven trust classes. The work done in [12] builds on the work proposed by [11] and primarily discusses

recommendations for trust propagation, but neither [11] or [12] capture the continuum of trust in a unified manner.

The have also been models based on BAN logic [13] and on evaluation assurances [14]. All of [11], [12], [13] and [14] have their own advantages in that they can all represent a trust relationships to a reasonable extent. Such models have been sufficient for the world of virtual computing, but for the world of ubiquitous computing, we need a model, which represents a ***rich diversity*** of trust relationships (which we term as the continuum of trust and also includes ad-hoc trust relationships), which is so common in day-to-day life.

PGP's [15] trust model captures trust at certain levels, but does not capture the continuum of trust. Work in [16] and [17] tries to capture trust relationships at a social level, which also deal with real world models of trust, but their models are quite different from the unified model for the continuum of trust proposed in this paper. [16] talks about reputation based trust relationships. However neither [16] nor [17] capture in a unified manner the trust requirements of both identity and context based trust relationships (or ad-hoc trust relationships).

It must be however be noted that all of the above models serve their purpose in their own *domains*, which are probably sufficient for the current world of computing and it must again be stressed that the unified model is indeed a *new requirement*. As computing devices become more integrated into our daily life, we believe that the unified model presented in this paper is the necessary first step in capturing the trust relationships of both identity based and context based trust relationships in a single model.

5. Conclusions

Secure communications cannot occur between parties unless a *trust* relationship has been established. The vast majority of such relationships use identity as a basis for establishing trust. In the future, however, scenarios will exist where identity is either not available or insufficient for establishing a trust relationship. Examples of such scenarios occur in several ad-hoc networking situations where other attributes such as device context may be required. A flexible trust model, which is adaptable to the wide range of possible communication scenarios, and is built upon a formal basis is essential in providing secure communications in the future world of pervasive computing.

In this paper, we have proposed an attribute vector calculus and shown how it is capable of modeling a diverse set of real world situations from ad-hoc networking scenarios to the usual identity based scenario. In our model, the each participant determines the set of attributes (note that the participants need not use the same set of attributes) they utilize to establish trust along with an ordering relation for each attribute, which induces a trust value. The induced trust value serves as the "continuum of trust" upon which the participant can set any acceptable trust level. The beauty of this novel and simple approach is that it not only permits "shades of grey" within a trust model, but it also provides a formal basis upon which more complex models can be built and reasoned about. In addition, we believe that the model is easily implemented and used in the small and resource limited devices quickly becoming ubiquitous in our daily lives.

References

[1] M. Weiser. *Some computer science issues in ubiquitous computing.* Communications of the ACM, 36(7):75--85, July 1993.

[2] L. Kleinrock. *Nomadicity: Anytime, anywhere in a disconnected world*. Mobile Networks and Applications, 1997, pp. 351-357.

[3] D. Salber, A.K. Dey, G.D Abowd. *The Context Toolkit: Aiding the Development of Context-Enabled Applications*. In the Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99), May 15-20, 1999, pp. 434-441.

[4] A.K. Dey, D. Salber, G.D Abowd. *A Context-based Infrastructure for Smart Environments*. In the Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), December 13-14, 1999, pp. 114-128.

[5] S.P. Miller, C. Neuman, J.I. Schiller, and J.H. Saltzer, *Kerberos Authentication and Authorization System*, Project Athena Technical Plan, Section E.2.1, Massachusetts Institute of Technology, October, 1988

[6] PPP Extensible Authentication Protocol (EAP). RFC 2284.

[7] M. Blaze, J. Feigenbaum, and J. Lacy. *Decentralized trust management*. In Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1996. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press.

[8] M. Blaze, J. Feigenbaum, and A. D. Keromytis. *KeyNote: Trust management for public-key infrastructures*. Lecture Notes in Computer Science, 1550:59--63, 1999.

[9] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. *A calculus for access control in distributed systems*. TOPLAS, 15(4):706--734, Sept. 1993.

[10] A. Chander, D. Dean, J. Mitchell. *A State-transition Model of Trust Management and Access Control*. In Proceedings of the 14th Computer Security Foundations Workshop (CSFW), June 2001.

[11] R. Yahalom, B. Klein and T. Beth. *Trust relationships in secure systems-A distributed authentication perspective*. In Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy, pages 150--164, May 1993.

[12] T. Beth, M. Borcherding, and B. Klein. *Valuation of trust in open networks*. In ESORICS 94. Brighton, UK, November 1994

[13] M. Burrows, M. Abadi, and R.M. Needham. *A Logic of Authentication*, ACM Transactions on Computer Systems, Vol. 8, No. 1, Feb 1990, pp. 18-36.

[14] USDoD. *Trusted Computer Systems Evaluation criteria (TCSEC)*. U.S Department of Defense, 1985.

[15] P. Zimmermann. *PGP User's Guide*. MIT. October 1994.

[16] A. Abdul-Rahman, S. Hailes. *Supporting Trust in Virtual Communities.* In: Hawaii Int. Conference on System Sciences 33 , Maui, Hawaii, January 2000

[17] S. Marsh. *Formalising Trust as a Computational Concept*. Ph.D. Thesis, University of Stirling, 1994.