

# p-Trust: A New Model of Trust to Allow Finer Control Over Privacy in Peer-to-Peer Framework

Sudip Chakraborty, Indrajit Ray  
Computer Science Department, Colorado State University  
Fort Collins, CO 80523, USA  
Email: {sudip, indrajit}@cs.colostate.edu

**Abstract**— Every time a user conducts an electronic transaction over the Internet a wealth of personal information is revealed, either voluntarily or involuntarily. This causes serious breach of privacy for the user, in particular, if the personally identifying information is misused by the other users present in the network. Ideally, therefore, the user would like to have a considerable degree of control over what personal information to reveal and to whom. Researchers have proposed models to allow a user to determine what personal information to reveal while doing a transaction over the Internet. However, these models do not help the user in determining who to trust, how much to trust and why to trust them with the personal information. The models fail to address loss of privacy through the misuse of information. In this paper we propose a privacy enhancing trust model to measure the degree of confidence that a user can have in the context of preservation of her privacy during a transaction. The model considers several factor while computing trust which include a user's own experience and knowledge about the target user and feedback obtained from groups of peer users called 'trusted neighbors' and 'friends'. The proposed scheme provides a flexible and powerful approach for the secure handling of private data and offers a user considerable control over how she wishes to disseminate her personal data.

**Index Terms** – Security, trust, privacy, peer-to-peer network.

## I. INTRODUCTION

Researchers are getting increasingly concerned about protecting the user's privacy in an electronic world. Unfortunately, most of us would find it difficult to provide a concrete definition of privacy with enough information to be able to apply it to our real lives. As individuals, each of us have unique needs and views of what constitute personal and private data [1]. The task is considerably more difficult when we have to define what privacy means to us as we use the Internet. This is because the average Internet user has very little idea as to what the information profile they present on the Internet and how easily that information can be observed and captured [1]. A peer-to-peer (P2P) network is a portion of the general Internet and hence the above problem is also relevant to a P2P setup.

Almost all current privacy preserving technologies are based on the notion of trust. Before a user chooses to disclose personal information, these technologies require the user to establish a trust relationship with the recipient of the user's information. Almost always the process is based on exchange and evaluation of digital credentials. Privacy researchers have consequently proposed formalisms for defining credentials, languages for encoding policies into certificates, techniques

for selective disclosure of credentials and frameworks for trust negotiations. Since disclosure of credentials itself can lead to privacy violations, researchers have also looked into the problem how sensitive credentials can be protected during trust negotiations. However, a major problem with all these works is that the underlying trust model is always assumed to be a binary model. While from a theoretical point of view such a binary trust model is adequate, from a practical standpoint a binary model of trust prevents one from making reasoned decisions in the face of incomplete, insufficient or inconclusive information. In this work, we propose a framework by which a user can have confidence that her privacy will be protected to the extent she feels comfortable with.

We propose a trust-based model called "p-Trust" for making privacy related decisions by a peer in a framework similar to P2P system. The idea is that each interaction that a user performs with others is bound to disclose her personally identifying information to some degree. If the user feels comfortable with this level of disclosure the user's privacy is protected, otherwise not. Thus, before commencing on an activity, the user tries to determine to what degree she trusts or distrusts the other entity to protect her privacy. The p-Trust model is used to measure this trust degree. Unlike binary trust models, trust in this new model has different degrees and is computed based on several factors. The model takes into consideration the interaction history between the two entities, the properties or attributes that the trustee possesses, and the feedback about the trustee from her peers. The model restricts feedback from a group of trusted neighbors and friends only. We propose a mutual-consent based two-level random filtering process of peers to choose p-Trusted neighbors and friends. This process is based on p-Trust thresholds. The truster sets a p-Trust threshold and sends a 'neighbor invitation' message to those whose p-Trust level is above the threshold. A pre-determined number of peers are chosen to be in the list. A similar procedure is adopted for choosing friends from p-Trusted neighbors. The difference is that the threshold is more strict and only friends share the data needed to compute the p-Trust about a given peer. Malicious peers can affect the p-Trust decision by providing misleading information. The two-level random filtering process helps to mitigate this to a considerable extent. The fact that p-Trust threshold varies from peer to peer, makes it difficult for malicious peers to change their behavior so as to get a 'neighbor invitation' or

a ‘friendship invitation’ from the target peers.

The rest of the paper is organized as follows. In section II we introduce the p-Trust model. We start with defining what we mean by p-Trust followed by formally defining p-Trust context in section II-A. We describe the different parameters that are used to compute p-Trust values in the context of user privacy in section II-B. Normalization of a p-Trust relationship, value of a p-Trust relationship, and effect of time on a p-Trust relationship have been discussed in sections II-C, II-D and II-E respectively. We define a comparison operator on p-Trust relationships in section II-F. Section III presents our approach to controlling personal privacy using the p-Trust model. In this section we define concepts of ‘trusted neighbors’ and ‘friends’ together with an algorithm to create the ‘trusted neighbor’ set. We show how our p-Trust model can reduce the impact of malicious peers in section III-B followed by discussing how the parameters are computed, in section III-C. Section IV provides an architecture of the p-Trust management system that is needed at each peer to successfully evaluate and manage p-Trust relationships. We present related works in section V. Finally section VI concludes the paper with some discussion on future work.

## II. THE P-TRUST MODEL

Privacy has been defined in many ways often differing from each other quite radically. Each of these definitions is either based on some static categorization of data or deals with privacy from a single viewpoint of a specific type of user within a system [2], [3], [4], [5], [6]. For this work we use the following definition of user privacy

*Definition 1:* User privacy is an interest that the user has in maintaining her personal information including data and knowledge about herself and her actions and activities on the Internet, securely in her control without that control being compromised by other individuals and entities.

Defining user privacy as an interest enables us to use the notion of degree of privacy. We do not specify how a user can measure this degree of privacy. It can be measured in terms of qualitative metrics like ‘high’, ‘medium’, or ‘low’, or quantitative values like ‘real number between [0, 1]’. However, we do not assume that this measure is uniform across all users. In other words, if two users indicate having ‘medium’ (or 0.6) privacy level we assume that they express the same level of interest in protecting privacy. This view of privacy as an interest allows us to use the concept of “trusting others to maintain privacy to this level”. We propose the p-Trust model to measure this trust value. p-Trust is trust as applied to user privacy. We specify p-Trust in the form of a trust relationship between two entities – the truster – an entity that trusts the target entity – and the trustee – the target entity that is trusted. We begin by defining p-Trust and p-Distrust.

*Definition 2:* p-Trust is defined to be a measure of the truster’s belief in the competence of the trustee to act dependably and securely in maintaining the truster’s privacy to the level the former wants to have.

*Definition 3:* p-Distrust is defined to be a measure of the truster’s belief in the incompetence of the trustee to act

dependably and securely in maintaining the truster’s privacy to the level the former wants to have.

Although we define p-Trust and p-Distrust separately in our model, we allow the possibility of a neutral position where there is neither p-Trust nor p-Distrust.

We believe that there is a possibility of ambivalence in making p-Trust decisions. Hence we choose to define these three different regions for trust. Thus, p-Distrust is not just the negative of p-Trust. We, therefore, see p-Trust (and p-Distrust) as how far the truster can extend her confidence on the trustee in the positive sense (and in the negative sense) from the neutral level. As we elaborate on the model this will become more clear.

In our model p-Trust is always related to a privacy relevant context  $c$ . The simple p-Trust relationship between a user  $A$  and an entity  $B$ ,  $(A \xrightarrow{c} B)$ , is a four element vector. The components are *interactions*, *properties*, *reputation* and *recommendation*. It is represented by  $(A \xrightarrow{c} B) = [{}_A I_B^c, {}_A P_B^c, {}_A RE P_B^c, {}_R RE C_B^c]$ , where  ${}_A I_B^c$  represents the magnitude of  $A$ ’s interaction about  $B$  in context  $c$ ,  ${}_A P_B^c$  represents  $B$ ’s properties relevant to  $c$  as evaluated by  $A$ ,  ${}_A RE P_B^c$  represents  $B$ ’s reputation in  $c$  and  ${}_R RE C_B^c$  represents the cumulative effect of all  $B$ ’s recommendations to  $A$  from other entities.

We discuss our model in a peer-to-peer framework. A peer interacting with another peer evaluates the latter’s p-Trust in the context of keeping the former’s privacy. Peers, depending on some p-Trust condition, provide and share information to help each other to evaluate the above four components. Thus the information required to evaluate a peer’s p-Trust is distributed over the network and the evaluation is collaborative.

### A. p-Trust Context

As mentioned above, a p-Trust relationship between peers  $A$  and  $B$  is never absolute. For example, a peer (a truster) can trust the other peer (trustee) to protect her private information during a communication exchange. However, that does not necessarily mean that  $A$  also trusts  $B$  to store the private information in a proper manner. Similarly, if we want to compare two p-Trust values, we cannot compare two arbitrary p-Trust values. Instead, we need to compare the values for p-Trust with similar scopes. This leads us to associate a notion of *context* with a p-Trust relationship. For this purpose we adapt from similar notions supported in P3P [7].

*Definition 4:* The *atomic purpose* of a p-Trust relationship  $(A \xrightarrow{c} B)_t$  is one of (1) completion and support of activity for which information is provided, (2) research and development, (3) pseudonymous analysis, (4) individual analysis, (5) pseudonymous decision, (6) individual decision, and (7) historical preservation.

Note that the set of atomic purposes is really a subset of the concept of purposes from P3P. We use only that subset that is relevant for electronic transactions. The truster may also trust the trustee for some combination of these atomic purposes.

*Definition 5:* The *purpose* of a p-Trust relationship is defined as follows.

- 1) An atomic purpose is a purpose of a p-Trust relationship.

- 2) The negation of an atomic purpose, denoted by “not” atomic purpose, is a purpose.
- 3) Two purposes connected by the operator “and” form a purpose.
- 4) Two purposes connected by the operator “or” form a purpose.
- 5) Nothing else is a purpose.

Further, in our model of p-Trust we are interested in five categories of privacy. These are (1) *Physical privacy* – relating to individual’s physical attributes, (2) *Behavioral privacy* – relating to all aspects of an individual’s behavior. This include issues such as social habits, political view or activities, religious practices, buying habits, sexual preferences, food habits, movement patterns etc., (3) *Possession privacy* – concerning material possessions that are legally owned or represented by an individual, (4) *Communication privacy* – related to an individual’s ability to communicate with another entity (through any medium) without the monitoring of these communications by other entities, and (5) *Data privacy* – concerning information related to an individual. The information should not be automatically available to other entities without the consent of the individual. If it is possessed by another entity, the individual should have a desired level of control over the data and its use.

Combining the concepts of p-Trust purpose and p-Trust category, we define *p-Trust context* as the interrelated conditions in which p-Trust exists or occurs. For example, a peer *A* may be interested in evaluating a p-Trust relationship with another peer *B*. Peer *B* requires *A*’s private data for pseudonymous analysis and would also store the data for a period of time. The peer *A* thus wants to determine the degree of p-Trust in the peer *B* in the context “protect her data privacy in pseudonymous analysis and historical preservation”. Context is formalized as follows.

**Definition 6:** Let  $\mathcal{S}$  denote the set of p-Trust purposes and  $\mathcal{A}$ , the set of p-Trust categories identified above. Then the context,  $c(T)$ , of a p-Trust relationship  $T$  is defined as follows:

- 1) A tuple of the form  $\langle a_i, s_i \rangle$  is a context where  $s_i \in \mathcal{S}$  and  $a_i \in \mathcal{A}$ .
- 2) Two contexts connected by the operator “and” form a context.
- 3) Two contexts connected by the operator “or” is a context.
- 4) Nothing else is a context.

**Definition 7:** The context function  $c(T)$  of a p-Trust relationship  $T$  is a function that takes the p-Trust relationship as the input and returns the context of that p-Trust relationship.

### B. p-Trust Components

To compute a p-Trust relationship we assume that each of the aforementioned four factors – *interactions, properties, reputation, recommendation* is expressed in terms of a numeric value in the range  $[-1, 1] \cup \{\perp\}$ . A negative value for the component is used to indicate the *p-Trust-negative* type for the component, whereas a positive value for the component is used to indicate the *p-Trust-positive* type of the component. A 0 (zero) value for the component indicates *p-Trust-neutral*. A p-Trust-positive component increases trust

degree whereas a p-Trust-negative component diminishes trust degree. A p-Trust-neutral interaction contributes neither way. To indicate a lack of value due to insufficient information for any component we use the special symbol  $\perp$ . We define the following properties of  $\perp$ . If  $\mathbb{R}$  is the set of real numbers, then (i)  $a * \perp = \perp * a = \perp, \forall a \in \mathbb{R}$  (ii)  $a + \perp = \perp + a = a, \forall a \in \mathbb{R}$  (iii)  $\perp + \perp = \perp$  and  $\perp * \perp = \perp$ .

### Interactions:

The *interactions* component captures the behavioral history of a trustee peer with the truster peer in some context  $c$ . We model *interactions* in terms of the number of events encountered by a peer regarding a trustee peer in the context  $c$  within a specified period of time. It is the measure of the cumulative effect of a number of events (p-Trust-positive, p-Trust-negative, p-Trust-neutral) that were encountered by the truster  $A$  with respect to the trustee  $B$  in a particular context  $c$  and over a specified period of time  $[t_0, t_n]$ . Before formally defining *interactions*, we discuss the following concepts:

Let  $\mathbb{N}$  denote the set of natural numbers. The set of time instances  $\{t_0, t_1, \dots, t_n\}$  is a totally ordered set, ordered by the temporal relation  $\prec$ , called the *precedes-in-time* relation, as follows:  $\forall i, j \in \mathbb{N}, t_i \prec t_j \Leftrightarrow i < j$ . We use the symbol  $t_i \preceq t_j$  to signify either  $t_i \prec t_j$  or  $t_i = t_j$ . Let  $e_k$  denote the  $k^{th}$  event. Events happen at time instances. We define the concept *event-occurrence-time* as follows:

**Definition 8:** The *event-occurrence-time*,  $ET$ , is a function that takes an event  $e_k$  as input and returns the time instance,  $t_i$  at which the event occurred. Formally,  $ET : e_k \rightarrow t_i$ .

We divide the time period  $[t_0, t_n]$  into a set  $\mathcal{T}$  of  $n$  intervals,  $[t_0, t_1], [t_1, t_2], \dots, [t_{n-1}, t_n]$  such that for any interval  $[t_i, t_j], t_i \prec t_j$ . A particular interval,  $[t_{k-1}, t_k]$ , is referred to as the  $k^{th}$  interval. We extend the  $\prec$  relation on  $\mathcal{T}$  and the time intervals are also totally ordered by the  $\prec$  relation as follows:  $\forall i, j, k, l \in \mathbb{N}, [t_i, t_j] \prec [t_k, t_l] \Leftrightarrow t_j \prec t_k$ . The intervals are non-overlapping except at the boundary points, that is  $\forall i, j, k, l \in \mathbb{N}, [t_i, t_j] \cap [t_k, t_l] = \emptyset, t_j = t_k$  for consecutive intervals. Lastly, for two consecutive intervals  $[t_i, t_j]$  and  $[t_j, t_k]$  if  $ET(e_k) = t_j$  then we assume  $e_j \in [t_i, t_j]$ .

Let  $\mathcal{P}$  denote the set of all p-Trust-positive events,  $\mathcal{Q}$  denote the set of all p-Trust-negative events, and  $\mathcal{N}$  denotes all p-Trust-neutral events (that is  $\mathcal{E} = \mathcal{P} \cup \mathcal{Q} \cup \mathcal{N}$ , where  $\mathcal{E}$  is the set of all events). We assume that within a given interval all p-Trust-positive events contribute equally to the formation of a p-Trust value and all p-Trust-negative events also do the same. The p-Trust-neutral events contribute nothing. We assign equal numeric weights to all events, p-Trust-positive or p-Trust-negative, within the same given interval. Let  $v_k^i$  be the weight of the  $k^{th}$  event in the  $i^{th}$  interval. We assign a weight of +1 if an event is in the set  $\mathcal{P}$ , -1 if the event is in the set  $\mathcal{Q}$ , and 0 if the event is in  $\mathcal{N}$ . Formally, if  $e_k^i$  denote the  $k^{th}$  event in the  $i^{th}$  interval, then

$$v_k^i = \begin{cases} +1, & \text{if } e_k^i \in \mathcal{P} \\ -1, & \text{if } e_k^i \in \mathcal{Q} \\ 0, & \text{if } e_k^i \in \mathcal{N} \end{cases}$$

**Definition 9:** The *incidents*  $IN_j$ , corresponding to the  $j^{th}$  time interval is the sum of the values of all the events,

p-Trust-positive, p-Trust-negative, or p-Trust-neutral for the time interval. If no event happened in  $j^{th}$  time interval, then  $IN_j = \perp$ . If  $n_j$  is the number of events that occurred in the  $j^{th}$  time interval, then

$$IN_j = \begin{cases} \perp, & \text{if } \nexists e \in \mathcal{E} \text{ such that } ET(e) \in [t_{j-1}, t_j] \\ \sum_{k=1}^{n_j} v_k^j, & \text{otherwise} \end{cases}$$

Events far back in time does not count as strongly as very recent events for computing p-Trust values. We give more weight to events in recent time intervals than those in distant intervals. To accommodate this in our model, we assign a *non-negative* weight  $w_i$  to the  $i^{th}$  interval such that  $w_i > w_j$  whenever  $j < i$ ,  $i, j \in \mathbb{N}$ . We then define *Interactions* as follows:

**Definition 10:** The *Interactions* of an entity  $A$  about another entity  $B$  for a particular context  $c$ , is the accumulation of all p-Trust-positive, p-Trust-negative, and p-Trust-neutral events that  $A$  has with regards to  $B$  over a given period of time  $[t_0, t_n]$ , scaled to be in the range  $[-1, 1] \cup \{\perp\}$ .

*Interactions* has a value in the range  $[-1, 1] \cup \{\perp\}$ . To ensure that the value is within this range we restrict the weight  $w_i$  for the  $i^{th}$  interval as  $w_i = \frac{i}{S}$ ,  $\forall i = 1, 2, \dots, n$ , where  $S = \frac{n(n+1)}{2}$ . Then the ‘interactions’ of  $A$  with regards to  $B$  for a particular context  $c$  is given by

$$AI_B^c = \frac{\sum_{i=1}^n w_i IN_i}{\sum_{i=1}^n n_i} \quad (1)$$

If there is a situation where nothing happened between two peers  $A$  and  $B$  over the entire time period  $[t_0, t_n]$ , then  $IN_i = \perp$ ,  $\forall i = 1, 2, \dots, n$ . As a result, we have  $w_i IN_i = \perp$ ,  $\forall i = 1, 2, \dots, n$  which implies  $AI_B^c = \perp$ . The above is different from the situation when  $AI_B^c = 0$ . Since, if the number of p-Trust-positive events is equal to number of p-Trust-negative events in each interval, then  $IN_i = 0$ ,  $\forall i = 1, 2, \dots, n$ . As a result we get  $AI_B^c = 0$ . But  $AI_B^c = \perp$  occurs only when there is no interaction between the truster and the trustee over the entire time period.

### Properties:

**Definition 11:** The *properties* of a trustee for a particular context is defined as a measure of the characteristic attributes or information of the trustee for which the truster can have some assertion to be truly related to the trustee.

The parameter “properties” is more difficult to compute and is, to some extent, subjective. To begin with, each truster must define its own criteria for gradation of properties regarding a particular entity. To assign a value to the *properties* component, the truster assigns a value between -1 and +1 for each attribute of the trustee depending on the policy (called, *property evaluation policy*) of the truster. Also the truster is solely responsible for assigning the relative weights to different attributes or information. Average of these values gives the value for the component *properties*.

It is possible that the truster has insufficient information to assign a value to properties. For these cases, we assign  $\perp$  to the component. If the truster is aware of  $k$  attributes of the trustee, then properties of trustee  $B$  according to truster  $A$  in

context  $c$  is evaluated as

$$AP_B^c = \frac{\sum_{i=1}^k pv_i}{k} \quad (2)$$

where  $pv_i \in [-1, 1]$ ,  $\forall i = 1, 2, \dots, k$ .  $pv_i$  is the value assigned to  $i^{th}$  attribute of  $B$  and is determined by the underlying property evaluation policy of the truster. Note,  $AP_B^c = \perp$  is different from  $AP_B^c = 0$ . Value 0 implies that after evaluating the information according to p-Trust policy, the truster’s decision is neutral. This happens because there were some positive and some negative attributes such that the influence of the two balances each other. The value ‘ $\perp$ ’ implies “lack of information”, that is, there is not enough data to determine ‘properties’ of the trustee.

### Reputation:

**Definition 12:** A *reputation* of a trustee is defined as a measure of the non-attributable information (in terms of feedback or properties) about the trustee to the truster in a particular context.

A trustee’s reputation is non-attributable to any specific source. Thus the truster does not have any guarantee for it to be useful. However, with this reputation, the truster can build an opinion about the trustee in the context. The component *reputation* is difficult to compute objectively. It is more subjective in nature and completely depends on the truster’s discretion. We evaluate ‘reputation’ *REP* about the trustee  $B$  in context  $c$  as

$$AREP_B^c = \frac{r_p - r_n}{r_p + r_n} \quad (3)$$

where  $r_p$  is number of p-Trust-positive reputations and  $r_n$  is number of p-Trust-negative reputations about  $B$ . Like *properties*, the truster is responsible to classify a reputation information as p-Trust-positive or p-Trust-negative. Note,  $AREP_B^c = 0$  implies that the truster has equal number of positive and negative reputation information whereas the  $AREP_B^c = \perp$  implies “lack of reputation information”, that is, there is not enough data to determine ‘reputation’ of the trustee.

### Recommendation:

**Definition 13:** A *recommendation* about a trustee is defined as a measure of the subjective or objective judgment of a recommender about the trustee to the truster.

*Recommendation* is evaluated on the basis of a *recommendation value* returned by a recommender to  $A$  about  $B$ . Unlike reputation, the recommendation is attributable to specific source. Truster  $A$  uses the p-Trust level she has on the recommender as a weight to the value returned. This weight multiplied by the former value gives the actual *recommendation score* for trustee  $B$  in context  $c$ .

If  $R$  is a group of  $n$  recommenders,  $\mathbf{v}(A \xrightarrow{c} j)$  is p-Trust-value of  $j^{th}$  recommender and  $V_j = j^{th}$  recommender’s recommendation value about the trustee  $B$ , then the *recommendation* of  $A$  with regards to  $B$  for a particular context  $c$  is given by

$$RREC_B^c = \frac{\sum_{j=1}^n \mathbf{v}(A \xrightarrow{c} j) \cdot V_j}{\sum_{j=1}^n \mathbf{v}(A \xrightarrow{c} j)} \quad (4)$$

C. Normalized p-Trust vector

During evaluation of a p-Trust value, a truster may assign different weights to the different factors that influence p-Trust. The weights will depend on the *p-Trust evaluation policy* of the truster. Thus if two different trusters assign two different sets of weights, then the resulting p-Trust value will be different. We capture this factor using the concept of a *normalization policy*. The normalization policy is a vector of same dimension as  $(A \xrightarrow{c} B)$ ; the elements are weights that are determined by the corresponding p-Trust evaluation policy of the truster and assigned to interactions, properties, reputation, and recommendation components of  $(A \xrightarrow{c} B)$ . We use the notation  $(A \xrightarrow{c} B)^N$ , called *normalized p-Trust relationship* to specify *A's normalized p-Trust on B* in a particular context *c*. This relationship is obtained from the simple p-Trust relationship –  $(A \xrightarrow{c} B)$  – after combining the former with the normalizing policy. The normalized p-Trust vector is given by

$$\begin{aligned} (A \xrightarrow{c} B)^N &= \mathbf{W} \odot (A \xrightarrow{c} B) \\ &= [W_I, W_P, W_{REP}, W_{REC}] \odot [A\hat{I}_B^c, \\ &\quad A\hat{P}_B^c, A\hat{R}EP_B^c, R\hat{R}EC_B^c] \\ &= [W_I \cdot A\hat{I}_B^c, W_P \cdot A\hat{P}_B^c, W_{REP} \cdot A\hat{R}EP_B^c, \\ &\quad W_{REC} \cdot R\hat{R}EC_B^c] \\ &= [A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}EP_B^c, R\hat{R}EC_B^c] \end{aligned}$$

where  $W_I, W_P, W_{REP}, W_{REC} \in [0, 1]$  and  $W_I + W_P + W_{REP} + W_{REC} = 1$ .

D. Value of the normalized p-Trust vector

We now introduce a concept called the *value* of a p-Trust relationship. This is denoted by the expression  $\mathbf{v}(A \xrightarrow{c} B)^N$  and is a number in  $[-1, 1] \cup \{\perp\}$  that is associated with the normalized p-Trust relationship. This value represents a p-Trust of certain degree.

*Definition 14:* The *value* of a normalized trust relationship  $(A \xrightarrow{c} B)^N = [A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}EP_B^c, R\hat{R}EC_B^c]$  is a number in the range  $[-1, 1] \cup \{\perp\}$  and is defined as  $\mathbf{v}(A \xrightarrow{c} B)^N = A\hat{I}_B^c + A\hat{P}_B^c + A\hat{R}EP_B^c + R\hat{R}EC_B^c$ .

The value for a p-Trust relationship allows us to revise the terms “p-Trust” and “p-Distrust” as follows: (i) If the value, *T*, of a normalized p-Trust relationship is such that  $0 < T \leq 1$  then it is p-Trust. (ii) If the value, *T*, of a normalized p-Trust relationship is such that  $-1 \leq T < 0$  then it is p-Distrust. (iii) If the value, *T*, is 0 then it is neither p-Trust nor p-Distrust. (iv) If the value, *T*, is  $\perp$  then it is *undefined*.

E. Effect of Time on p-Trust Relationships

p-Trust (and p-Distrust) changes over time. Let us assume that we have initially computed a p-Trust relationship  $\vec{T}_{t_i}$  at time  $t_i$ , based on the values of the underlying parameters at that time. Suppose now that we try to recompute the p-Trust relationship  $\vec{T}_{t_n}$  at time  $t_n$ . We claim that even if the underlying parameters do not change between times  $t_i$  and  $t_n$ , the p-Trust relationship will change. To model *p-Trust dynamics* (the change of p-Trust over time) we borrow

from observations in the social sciences that indicate that human abilities and skills respond positively to practice, in a learning-by-doing manner, and negatively to non-practice [8]. We observe that the general tendency is to forget about past happenings. This leads us to argue that p-Trust (and p-Distrust) tends towards neutrality as time increases. Initially, the value does not change much; after a certain period the change is more rapid; finally the change becomes more stable as the value approaches the neutral (value = 0) level. We assert that  $\lim_{t \rightarrow \infty} \mathbf{v}(\vec{T}_t) = 0$ . This p-Trust dynamics can be represented by the graph shown in figure 1. How fast p-Trust

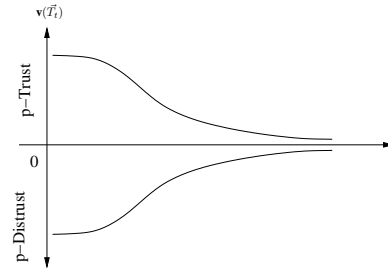


Figure 1. Graph showing the nature of p-Trust dynamics

(or p-Distrust) will decay over time, is, we propose, dependent on the truster’s policy. The truster may choose to forget about p-Trust relationships which are 3 years old or 5 years old. The model cannot dictate this. Our goal is to provide a basis by which the truster can at least estimate, based on the her individual perception about this, the p-Trust at time  $t_n$ . We further believe that p-Trust relationship at present time is not only dependent on the values of the underlying parameters, but also on the “decayed” value of the previous p-Trust value.

Let  $\mathbf{v}(\vec{T}_{t_i})$ , be the value of a p-Trust relationship,  $\vec{T}_{t_i}$ , at time  $t_i$  and  $\mathbf{v}(\vec{T}_{t_n})$  be the decayed value of the same at time  $t_n$ . Then the *time-dependent value* of  $\vec{T}_{t_i}$  is defined as follows:

*Definition 15:* The *time-dependent value* of a p-Trust relationship  $\vec{T}_{t_i}$  from time  $t_i$ , computed at present time  $t_n$ , is given by  $\mathbf{v}(\vec{T}_{t_n}) = \mathbf{v}(\vec{T}_{t_i})e^{-\mathbf{v}(\vec{T}_{t_i})\Delta t}^{2k}$  where  $\Delta t = t_n - t_i$  and  $k$  is any small integer  $\geq 1$ .

The effect of time is captured by the parameter  $k$  which is determined by the truster *A's dynamic policy* regarding the trustee *B* in context *c*. The current normalized vector together with this time-affected vector are combined according to their relative importance. Relative importance is determined by truster’s *history\_weight policy* which specifies two values  $\alpha$  and  $\beta$  in  $[0, 1]$  as weights to current vector and the vector obtained from previous p-Trust value. The new vector thus obtained gives the actual normalized p-Trust vector at time  $t$  for the p-Trust relationship between truster *A* and trustee *B* in context *c*. This is represented by the following equation.

$$(A \xrightarrow{c} B)_{t_n}^N = \begin{cases} [A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}EP_B^c, R\hat{R}EC_B^c], & \text{if } t_n = 0 \\ [\frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}], & \text{if } t_n \neq 0 \text{ and} \\ & A\hat{I}_B^c = A\hat{P}_B^c = A\hat{R}EP_B^c = R\hat{R}EC_B^c = \perp \\ \alpha \cdot [A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}EP_B^c, R\hat{R}EC_B^c] + \\ \beta \cdot [\frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}, \frac{\mathbf{v}(\vec{T})}{4}], & \text{if } t_n \neq 0 \\ & \text{and at least one of } A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}EP_B^c, \\ & R\hat{R}EC_B^c \neq \perp \end{cases}$$

where  $\alpha, \beta \in [0, 1]$  and  $\alpha + \beta = 1$ . Also  $[\frac{\mathbf{v}(\hat{T})}{4}, \frac{\mathbf{v}(\hat{T})}{4}, \frac{\mathbf{v}(\hat{T})}{4}, \frac{\mathbf{v}(\hat{T})}{4}]$  is the time-effected vector and  $\mathbf{v}(\hat{T}) = \mathbf{v}(T_{t_n})$ .

#### F. Comparing Two p-Trust Relationships

We are now in a position to determine the relative trustworthiness of two peers in the context of truster peer's privacy. By 'trustworthiness' we mean how worthy a trustee is to act accordingly in the truster's opinion. The need for such comparison occurs in many real life scenarios. Consider the following example. Suppose peer  $A$  behaving as a client has a choice of two other peers  $B$  and  $C$  working as service providers for the same service. In this case  $A$  will probably want to compare its p-Trust relationships with entities  $B$  and  $C$  and decide to go with the peer that  $A$  trusts more to protect her private information. This motivates us to define a comparison operator on p-Trust relationships. However, not all p-Trust relationships can be compared. Since p-Trust depends on contexts, the two p-Trust relationships that are being compared should have the same context.

Therefore, let  $T$  and  $T'$  be two normalized p-Trust relationships at time  $t$ . We introduce the following notion of compatibility between two p-Trust relationships  $T$  and  $T'$  as

**Definition 16:** Two p-Trust relationships,  $T$  and  $T'$  are said to be *compatible* if the p-Trust relationships have been defined under the same p-Trust evaluation policy, the p-Trust relationships are at the same time instances, and the context  $c(T)$  for the p-Trust relationship  $T$  is the same as the context  $c(T')$  for  $T'$ , that is  $c(T) = c(T')$ . Otherwise the two p-Trust relationships are said to be *incompatible*.

The most intuitive way to compare two p-Trust relationships  $T$  and  $T'$  is to compare the values of the p-Trust relationships in a numerical manner. Thus for  $A$  to determine the relative levels of p-Trustworthiness of  $B$  and  $C$ ,  $A$  evaluates  $\mathbf{v}(A \xrightarrow{c} B)_t^N$  and  $\mathbf{v}(A \xrightarrow{c} C)_t^N$ . If  $\mathbf{v}(A \xrightarrow{c} B)_t^N > \mathbf{v}(A \xrightarrow{c} C)_t^N$ , then  $A$  trusts  $B$  more than  $C$  in the context  $c$ . We say that  $T$  *dominates*  $T'$ , given by  $T \succ T'$ . However, if  $\mathbf{v}(A \xrightarrow{c} B)_t^N = \mathbf{v}(A \xrightarrow{c} C)_t^N$ ,  $A$  cannot judge the relative p-Trustworthiness of  $B$  and  $C$ . This is because there can be two vectors whose individual component values are different but their scalar values are the same. For such cases we need to compare the individual elements of the two p-Trust relationships to determine the relative degree of p-Trustworthiness. In addition, for the same reasons, it is better to determine relative p-Trustworthiness of  $B$  and  $C$  on the basis of component values rather than breaking the tie arbitrarily.

Let  $(A \xrightarrow{c} B)_t^N = [A\hat{I}_B^c, A\hat{P}_B^c, A\hat{R}E\hat{P}_B^c, R\hat{R}E\hat{C}_B^c]$  and  $(A \xrightarrow{c} C)_t^N = [A\hat{I}_C^c, A\hat{P}_C^c, A\hat{R}E\hat{P}_C^c, R\hat{R}E\hat{C}_C^c]$  such that  $\mathbf{v}(A \xrightarrow{c} B)_t^N = \mathbf{v}(A \xrightarrow{c} C)_t^N$ . Let also the underlying trust evaluation policy vector be given by  ${}_A W = (w_1, w_2, w_3, w_4)$  where  $w_1 + w_2 + w_3 + w_4 = 1$  and  $w_i \in [0, 1], \forall i = 1, \dots, 4$ . To determine the dominance relation between  $T$  and  $T'$  we first determine the *ordered* trust relationships  $\bar{T}$  and  $\bar{T}'$  corresponding to  $T$  and  $T'$ .

**Definition 17:** The *ordered* trust relationship  $\bar{T}$  is generated from a normalized trust relationship  $T$  as follows:

- 1) Order the  $w_i$ 's in the trust evaluation policy vector corresponding to  $T$  in descending order of magnitude.
- 2) Sort the components of the trust vector  $T$  according to the corresponding weight components.

We compare the two ordered trust relationships  $\bar{T}$  and  $\bar{T}'$ , corresponding to  $T$  and  $T'$ , component-wise to determine the dominance relation between the two. Note that we assume that the same underlying trust evaluation policy vector has been used to determine the trust relationships. If the first component of  $\bar{T}$  is numerically greater than the first component of  $\bar{T}'$  then  $T \succ T'$ . Else if the first components are equal then compare the second components. If the second component of  $\bar{T}$  is greater than the second component of  $\bar{T}'$  then  $T \succ T'$ , and so on. If weights are equal for first three (or, all four) components in the ordered trust relationships, then  $T \succ T'$  only when the three components (or, all four components) of  $T$  are numerically greater than those of  $T'$ . In the comparison process we assume that the value  $\perp$  is dominated by all real numbers. If we cannot conclude a dominance relation between two p-Trust relationships, then we say that the two p-Trust relationships are *incomparable*. This is formalized by the following definition.

**Definition 18:** Let  $T$  and  $T'$  be two p-Trust relationships and  $\bar{T}$  and  $\bar{T}'$  be the corresponding ordered p-Trust relationships. Let also  $\bar{T}_i$  and  $\bar{T}'_i$  represent the  $i^{th}$  component of each ordered p-Trust relationships and  $w_i$  represent the  $i^{th}$  weight component in the corresponding p-Trust evaluation policy vector.  $T$  is said to *dominate*  $T'$  if any one of the following holds:

- 1)  $\mathbf{v}(T) > \mathbf{v}(T')$ ; or
- 2) if  $\forall i, j, i \neq j, (w_i = w_j)$  then  $\forall i, \bar{T}_i > \bar{T}'_i$ ; or
- 3) if  $\exists i, \bar{T}_i > \bar{T}'_i$  and for  $k = 0 \dots (i-1), \bar{T}_{i-k} \not\prec \bar{T}'_{i-k}$

Otherwise  $T$  is said to be *incomparable* with  $T'$ .

### III. PRESERVING PRIVACY USING THE P-TRUST MODEL

We look into the privacy preservation scheme from a client's perspective. That is, we investigate how a peer, acting as a client, can have a reasonable control over her privacy while interacting with another peer, acting as a server. We first identify following activities that a peer can perform as a client: (a) *Downloading* – The client downloads some resources from the server. This requires the client to specify (in active or passive manner) the download destination. (b) *Purchasing* – The client acquires some product, service, or access to a resource via a purchase. This requires the client to exchange funds and reveal a destination for whatever she is purchasing. In the case of acquiring access to some resource, that 'destination' is an identity to which that access is related. (c) *Sending/Receiving email* – The client exchanges electronic messages with other individuals to pass along digital information. (d) *Negotiating* – A series of proposal-response messages are passed between the client and the server, until either both parties reach an agreement with each others proposals, or one or both parties terminate the activity without an agreement. A certain level of trust is typically assumed in negotiation, and the client may have to reveal various characteristics about her to engender that trust and complete the negotiation. (e) *Filling out web*

forms – A client fills out a form presented by a website to willingly share information.

During any of these activities there are many different ways that the peer’s (client) privacy can be violated. We categorize the violations as follows (i) *Confidentiality breach* – when private and personal information of the client is intercepted and collected by an entity to whom the client is not intended to disclose that piece of information. (ii) *Integrity breach* – when private and personal information of a client is modified without the knowledge or consent of that client. This can occur even if the modification is done by a legitimate receiver, but who is not authorized to do so. (iii) *Information exploitation* – when private and personal information about the client, collected with her consent, is misused or allowed to be exploited. This would include, personal data of the client is made available for sale, use of the data by the receiver for profiling when the client has not so consented, use of the data that was not agreed to by the client prior its collection, and allowing unauthorized access to the data by other entities. (iv) *Personal space violation* – when an entity other than the client places data of any kind on the computing system of that client without the knowledge or expressed consent of the client. (v) *Pretexting/Identity theft* – when private or personal data of the client is used by someone other than the client without her consent to do so to gain access to resources, products, or services intended for the client only. (vi) *Anonymity violation* – when the identity of the client is disclosed despite the client’s effort to remain anonymous. (vii) *Linkability* – when personal or private data about the client, collected under the condition of anonymity of that client, is maintained/used/distributed in such a manner as to link that data to the identity of that client, or contribute to the linking of the identity of that client to that data. Some of the above listed violations can lead to other violations. For example, a breach in confidentiality can lead to integrity violation, information exploitation, or identity theft.

Before each transaction, a user evaluates the trustworthiness of the server using the p-Trust model described in section II. To evaluate this p-Trust the client uses her personal interactions with the server, information about characteristics of the server and information that she gathers from her peers. Note, however, a group of malicious peers can send false good/bad reviews about the server to influence the p-Trust decision of the client. The server may or may not be a member of that malicious group. To diminish the effect of such collusion we propose the concept of ‘trusted neighbors’ and ‘friends’. The ‘trusted neighbors’ and ‘friends’ share p-Trust information among themselves. However, the ‘friends’ of a peer will have more influence on the p-Trust decision of the peer. Note, we do not use the term ‘neighbor’ to mean the physical distance (in terms of length or hop) of a peer from the client. We intend to measure how ‘close’ the peer is with the client in terms of p-Trust relationship. Note also, these two relationships exist with mutual consent of peers at both end. If a peer  $i$  considers a peer  $j$  to be her ‘trusted neighbor’ but  $j$  denies to be so, then the relationship breaks and neither  $i$  nor  $j$  can consider each other as neighbors. Similar is true for friends. We now discuss how a peer builds these sets.

#### A. ‘Trusted neighbors’ and ‘Friends’

Let there be  $m$  peers in the network. To choose the trusted neighbor set, a peer  $i$  sets up a neighbor\_p-Trust threshold  $\tau_i^{nbr}$  and a number  $n$  ( $0 < n < m$ ). From the population of  $m$  peers,  $i$  chooses at most  $n$  peers whose p-Trust value is  $\geq \tau_i^{nbr}$ . Then  $i$  sends a message (‘neighbor\_invitation’) to each of these  $n$  peers requesting to build a neighbor relationship. A peer who returns an acceptance message is considered a “trusted neighbor” of  $i$ . The decision is taken by the other peer on the basis of her p-Trust on  $i$  and her own neighbor\_p-Trust threshold. If there are more than  $n$  peers who satisfy both the conditions, then  $i$  chooses  $n$  peers at random from that set. Therefore ‘trusted neighbors’ is defined as

*Definition 19:* The trusted neighbors of a peer  $i$  is the set  $NBR_i^t$  ( $t$  for ‘trusted’) of all peers  $j$  who satisfy the following two conditions: (i) the p-Trust value of  $j$  as evaluated by  $i$  is greater than or equal to the neighbor\_p-Trust threshold set by  $i$  and (ii)  $j$  accepts  $i$ ’s neighbor invitation according to her own basis. Formally,  $NBR_i^t = \{j | \mathbf{v}(i \xrightarrow{c} j)_i^N \geq \tau_i^{nbr} \wedge \mathbf{v}(j \xrightarrow{c} i)_j^N \geq \tau_j^{nbr}\}$

The condition ‘ $j$  accepts  $i$ ’s neighbor invitation’ is formally represented with the expression  $\mathbf{v}(j \xrightarrow{c} i)_j^N \geq \tau_j^{nbr}$ .

However, it may not always be possible to find  $n$  peers who satisfy both conditions (p-Trust value with at least  $\tau_i^{nbr}$  and acceptance of neighbor invitation). In that case  $i$  has two choices: (a)  $i$  can accept all available peers, say  $n'$  ( $n' < n$ ) who meet the specified conditions, or (b)  $i$  can reset  $\tau_i^{nbr}$  or  $n$  or both and run the algorithm again to choose the peers. It is preferable to reset  $n$  rather than setting  $\tau_i^{nbr}$ . This is because  $\tau_i^{nbr}$  gives the ‘confidence’ level that  $i$  should have on her neighbors. It should not be lowered for not having enough peers to meet that level. If there is no peer in the population of  $m$  peers who satisfies the specified p-Trust level, then only  $i$  should lower the threshold. When  $i$  receives a similar neighbor invitation from  $j$ ,  $i$  can accept or reject it based on whether  $\mathbf{v}(i \xrightarrow{c} j)_i^N \geq \tau_i^{nbr}$  or  $\mathbf{v}(i \xrightarrow{c} j)_i^N < \tau_i^{nbr}$ .

*Algorithm 1:* Get the trusted neighbors of a peer  $i$

**Input:** (i)  $S$  – Set of peers in the network ( $|S| = m > 0$ )  
(ii)  $i, \tau_i^{nbr}$  and  $n$  ( $0 < n \leq m$ )

**Output:**  $NBR_i^t$  – set of trusted neighbors of  $i$

**Procedure** *FindTrustedNeighbors*( $S, i, \tau_i^{nbr}, n$ )

**begin**

$NBR_i^t = \{\}$ ;

**for** each  $j \in S$

**if**  $\mathbf{v}(i \xrightarrow{c} j)_i^N \geq \tau_i^{nbr}$

Send ‘neighbor invitation’ to  $j$ ;

**if** receives an acceptance notification

$NBR_i^t = NBR_i^t \cup \{j\}$ ;

**if**  $|NBR_i^t| = n$  **return**  $NBR_i^t$ ;

**else if**  $|NBR_i^t| > n$

Select  $n$  members randomly from  $NBR_i^t$ ;

**return**  $NBR_i^t$ ;

**else if**  $|NBR_i^t| \neq 0 \wedge |NBR_i^t| < n$

**Case 1:** **return**  $NBR_i^t$ ;

**Case 2:** Set  $n = n' (< n)$ ;

*FindTrustedNeighbors*( $S, i, \tau_i^{nbr}, n'$ )

**if**  $|NBR_i^t| = 0 \wedge n' = 0$

Set  $\tau_i^{nbr} = \tau_i^{nbr'}$  where  $\tau_i^{nbr'} < \tau_i^{nbr}$ ;

*FindTrustedNeighbors*( $S, i, \tau_i^{nbr'}, n$ )

**end**

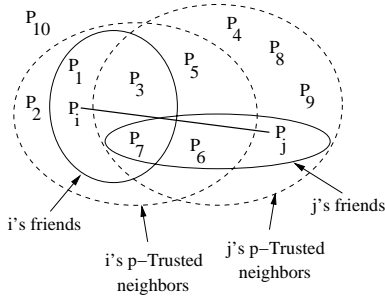


Figure 2. Trusted neighbors and friends of peers  $i$  and  $j$

We posit that all ‘trusted neighbors’ may not be ‘friends’ of peer  $i$ . The ‘friends’ are those trusted neighbors who are more ‘close’ to  $i$ , i.e.  $i$  has greater confidence and importance on their feedback. A friend  $k$ , unlike a non-friend trusted neighbor, can share her personal p-Trust data (data that she uses or has used to compute p-Trust of other peers) with  $i$ . However, we do not allow sharing of p-Trust evaluation policies even among friends to prevent possible manipulation in p-Trust value by a peer to become a ‘friend’. The ‘friends’ are chosen in the same manner from the set of trusted neighbors. After choosing  $n$  trusted neighbors, the client  $i$  sets a friend\_p-Trust threshold  $\tau_i^{fr}$  and a number  $f$  ( $f < n$ ). Peer  $i$  sends a ‘friendship invitation’ to each of those  $f$  peers and include them in her list after receiving acceptance notifications from the peers. The peer receiving the invitation accepts it only if  $i$ ’s p-Trust value with him is greater than equal to her friend\_p-Trust threshold. Therefore we can define ‘friends’ of a peer  $i$  as

**Definition 20:** Friends of a peer  $i$  is the set  $FR_i$  of all peers  $j$  who satisfy the following conditions: (i)  $j$  is a ‘trusted neighbor’ of  $i$ , (ii) the p-Trust value of  $j$  as evaluated by  $i$  is greater than or equal to the friend\_p-Trust threshold set by  $i$  and (iii)  $j$  accepts  $i$ ’s friendship invitation according to her own basis. Formally,  $FR_i = \{j \in NBR_i^t | \mathbf{v}(i \xrightarrow{c} j)_i^N \geq \tau_i^{fr} \wedge \mathbf{v}(j \xrightarrow{c} i)_j^N \geq \tau_j^{fr}\}$

The algorithm for forming the friends set is similar to the algorithm 1. Figure 2 shows the trusted neighbors and friends of peers  $i$  and  $j$  where  $i$  acts as the client and  $j$  acts as the server. Next we discuss how the ‘trusted neighbors’ and ‘friends’ can reduce the impact of malicious peers in the network.

### B. Reducing the Impact of Malicious Peers

As mentioned in section III-A, peers use information from trusted neighbors as ‘recommendation’ to evaluate other peers’ p-Trust level. The ‘friends’ share trust related information among each other to evaluate the p-Trust. Thus it is possible for a malicious peer to provide wrong information so as to influence a peer’s p-Trust evaluation. We intend to use the two tiers of trusted ‘neighbors’ and ‘friends’ to reduce the impact of such malicious peers in p-Trust evaluation.

By differentiating ‘neighbors’ from all other peers on the basis of their p-Trust values reduces the chance of a malicious peer being included in a peer’s neighbor list. This is due

to the fact that it is difficult for a malicious peer to behave improperly while keeping the p-Trust level sufficiently high. Since, in this case, to influence a peer’s p-Trust evaluation, a malicious peer needs to keep its p-Trust level to at least  $\min\{\tau_i^{nbr}\}$ , which is the minimum neighbor\_p-Trust threshold in the network. Since, every peer sets its own threshold, it is difficult for a particular peer to know this value as well as maintain it.

However, if there is a large group of malicious peers in the population of  $m$  peers, there is a chance of some of them being included in the population of trusted neighbors. This, despite a random selection being used. The second filter, i.e. the second random selection of  $f$  ‘friend’ peers, further diminishes the chance of a malicious peer in the ‘vicinity’ of  $i$  as  $\tau_i^{fr} > \tau_i^{nbr}$  for any peer  $i$ . Thus the effect of misleading information from malicious neighbors is restricted to affecting the recommendation component only. They are unable to influence the other peer’s p-Trust evaluation by sharing wrong information. Since, our framework does not allow sharing of p-Trust information among neighbors. Thus, to influence other peers’ p-Trust evaluation the malicious peer needs to achieve even a higher level of p-Trust.

Nonetheless, this type of two level random selection procedure does not completely remove the effect of malicious peers. There will still be some chance, though very little, of a malicious peer being included in the ‘friends’ set. The chance will depend on the parameters  $n, \tau_i^{nbr}, f, \tau_i^{fr}$  and the distribution of malicious peers in the network.

### C. Computation of the components

Now we discuss how a peer  $i$  (client) computes the p-Trust components to evaluate the p-Trust level of peer  $j$  (server).

**Computation of properties:** To quantify the ‘properties’ component of the p-Trust relation, the client  $i$  first needs to gather certain information about the peer  $j$  with respect to the following:

**Communication method** – Presence of a secure communication protocol like SSL can directly prevent *confidentiality breach, integrity breach, identity theft* and thereby can prevent other indirect violations of privacy. In communication method peer  $i$  may look further for following information: (a) Encryption method – which encryption method is being used in the communication. Under this category the client can have specific criteria for the following: (i) Encryption algorithm (e.g. AES or DES or RSA), (ii) Key type and size (symmetric key or asymmetric key; 56-bit or 128-bit or 512-bit), (b) Message digest algorithm (e.g. MD5 or SHA), (c) Authentication – which authentication mode is used (e.g. authentication of both peers, or only  $j$ ’s authentication or, it is totally anonymous), (d) Key exchange algorithm (e.g. RSA, Diffie-Hellman).

**Credential** – Presence of a certificate from a well-known certifying authority (e.g. Verisign) about policies, methods and tools applied and used by  $j$  in a particular transaction. The client  $i$  can have following sub-criterion: (i) Certifying authority – who the certifying authority is (i.e. how well-known the certifying authority is), (ii) Validation period –

how long the certificate is valid (e.g. if it is an old certificate and is still valid for sufficiently long, then that would create a positive impression about  $j$ ).

**Policy** – Presence of an explanation of policies adopted by  $j$  for a transaction. In particular,  $i$  looks for the following policies in the ‘policy document’ of  $j$  (i) *Data collection policy* – explaining how  $j$  is going to collect private and personal data from  $i$ , (ii) *Data storage policy* – explaining how  $j$  is going to store the private data of  $i$  so that it remains secure from the privacy violating threats, (iii) *Data handling policy* – explaining how  $j$  is going to use the data, (iv) *Data disclosure policy* – discussing whether  $j$  is going to disclose the data to third parties. If so, to whom it will be disclosed. (v) *Data retention policy* – explaining how long  $j$  is going to keep the private information of  $i$  in the storage, (vi) *Applicability & Validity* – applicability shows which entities are going to follow this policies (or, a part of the policies). Validity explains for how long  $j$  (or other entities) is going to stick to this policy. The lifetime of a policy tells the user how long she can rely on the claims made in the policy, or whether there is any exception in these policies, (vii) *Cookie policy* – a cookie policy must cover any data that is stored in that cookie or linked via that cookie. It must also reference all purposes associated with data stored in that cookie or enabled by that cookie. In addition, any data/purpose stored or linked via a cookie must also be put in the cookie policy. It must clearly specify the path of the cookie (this would give the idea about the parties that are going to get the data), (viii) *Dispute handling policy* – explaining how  $j$  is going to resolve dispute issues, or if  $i$  lodges a complain about her privacy being violated, what compensation  $j$  is offering.

Once some or all of these information are available,  $i$  assigns a value from  $[-1, 1]$  to each. Absence of information for any of the items is considered as  $\perp$ . For a category where  $i$  has options, she chooses a list of method with some pre-assigned value within  $[-1, 1]$ . This value is assigned according to  $i$ 's *property evaluation policy*. For example, for encryption method, let  $i$  assign a value 0.9 for 128-bit AES and 0.5 for 56-bit DES. If  $i$  finds that  $j$  uses 128-bit AES, then for that criterion,  $i$  has a value 0.9. The property component is then calculated using equation 2. Note, successful evaluation of properties depends on how well the peer  $i$  designs its property evaluation policy.

**Computation of the interactions:** Most of the information that goes toward forming the properties of the peer  $j$  in a particular privacy context by itself does not necessarily enhance/diminish the peer  $i$ 's p-Trust on  $j$ . This is because majority of the above criteria are examples of self-assertions. There is no guarantee that the peer  $j$  conforms to these self-assertions.  $j$ 's behavior as a peer (it includes behavior as a trusted neighbor or friend during a transaction where  $j$  is not the peer whose p-Trust is being determined) manifests in the form of *events*. If there are events that conforms to the properties that  $i$  has gathered then these events will be termed p-Trust-positive. If the events are contrary to the properties then they are p-Trust-negative. A false or misleading recommendation is also a p-Trust-negative event. All other events

are p-Trust-neutral.

Categorizing an event to positive or negative depends on the client  $i$ 's policy, specific activities and violations. Interactions is computed by counting how many times (i.e., in how many events)  $j$  has deviated from or conformed to self-assertions or provided wrong information. During a specific period of time, number of deviations from the stated self-assertions give number of p-Trust-negative events in that period. The events where  $j$  adhered to the self-assertions or provided correct feedback generate p-Trust-positive events.

**Computation of the reputation and recommendation:** To compute these two components a peer needs information from other peers. We assume that  $i$  requests only the trusted neighbors for recommendation or considers their feedback as recommendation. Information obtained from other peers are used to compute reputation. The reason is as follows:  $i$  will have a low p-Trust value for the peers other than trusted neighbors. Therefore the information collected from those peers are almost non-attributable to  $i$  though she knows about the source. Alternatively, information from trusted neighbors carry more importance to  $i$  to make her p-Trust decision. Note, a recommendation from a trusted source is more reliable than a reputation information (reputation is as we have defined).  $i$  can gather information about  $j$ 's *reputation* from the following: (i) general description of  $j$ 's activities and performance – this can be available from the other peers, (ii) report of other peers about  $j$  – this report can contain evaluation of  $j$  and comments by those peers. The report can have two categories: (a) general – general remark about  $j$  by the other peer, (b) specific – action specific remark about  $j$ . For example, how  $j$  has performed to handle private data, how it has collected and stored sensitive data etc. After collecting these data,  $i$  classifies each piece of information as positive or negative. The reputation ( ${}_iREP_j^c$ ) is calculated using the equation 3.

Recommendation is computed using information from trusted neighbors and friends. Client  $i$  sends ‘recommendation requests’ to all her trusted neighbors including friends. Trusted neighbors respond to the request with a recommendation value within  $[-1, 1]$ . The recommendation is then computed using the equation 4.

#### IV. ARCHITECTURE OF P-TRUST MANAGEMENT SYSTEM

From the above sections it is clear that there is no central authority to manage the p-Trust information required to compute a peer's p-Trust. Rather p-Trust data is stored across the network in a distributed manner where peers have partial or all information about other peers. Figure 3 gives a schematic diagram of a peer's p-Trust management module. The module evaluates a peer's p-Trust about another peer and co-operates with other peers. This module is also responsible to store and manage data that is shared with friends. It comprises the following components:

**p-Trust database:** The database stores all related information needed to compute p-Trust value. This includes values of the parameters, event-logs, property information about specific p-

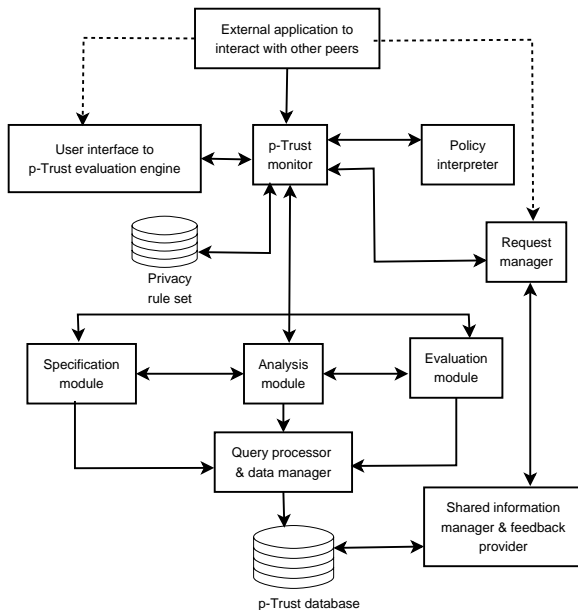


Figure 3. Components of p-Trust evaluation module of a peer

Trust relationships. It also stores information about different policies that the peer needs to evaluate p-Trust.

**Privacy rule set:** During system initiation the client peer has to specify her set of privacy rules. These rules define how the peer intends to evaluate specific privacy preserving steps.

**p-Trust monitor:** The p-Trust monitor gets relevant inputs from either the client peer or the external application.

**User interface to evaluation engine:** This module is responsible for interacting with the peer to gather information relevant to evaluate p-Trust in the current session. It also provides feedback to the peer in the form of computed p-Trust values. One important function of this module is to assist the peer in formulating/updating her privacy rule sets.

**Specification module:** This module is responsible for defining and managing p-Trust relationships. It creates database entries corresponding to specific peers when a new p-Trust relationship is established. It codifies general evaluation policies. The specification module conveys this information to the analysis module and the evaluation module as and when needed.

**Analysis module:** The analysis module processes p-Trust queries from either the p-Trust monitor or from the client.

**Evaluation module:** This module retrieves information about the components from the database and other pertinent information from the p-Trust monitor to compute p-Trust vector according to the theory specified in this paper. It also stores back resulting values in the database.

**Request manager:** The request manager receives requests from other peers and responds to those requests. It interacts with p-Trust monitor module to determine the p-Trust of the source peer (i.e., it checks whether the request came from a friend or a p-Trusted neighbor or any other peer). It also interacts with 'shared information manager and feedback provider' module.

**Shared information manager and feedback provider:** It

manages the portion of the p-Trust data that has been shared with some other peer(s). It receives feedback requests and instructions from 'request manager'. Depending on the instruction it fetches relevant data from the p-Trust database and pass it to request manager.

A peer  $i$  while computing p-Trust for another peer  $j$  may not have all the necessary information to compute the components. Sharing of data among 'friends' provides the required data that is not available directly from the local data manager of  $i$ . It also ensures that the peer  $i$  can have a reasonable confidence on the data to compute p-Trust as those are provided by a 'friend' which have relatively high p-Trust values. The peer  $i$  may store some or all of these data for future use. When  $i$  as a peer receives request from a peer  $k$  for p-Trust data about  $j$ ,  $i$  forwards these data to  $k$ . This allows peers to get current information to compute p-Trust of each other.

## V. RELATED WORK

A number of researchers have previously explored the idea of modeling privacy. The approaches that are closest to our approach are the ones by Goeck and Mynatt [9], Shand et al. [10] and Nguyen and Mynatt [11]. Goecks and Mynatt treat reputation and trust as separate independent entities and proposes an approach to combine trust networks with reputation to provide privacy [9]. Shand et al. [10] on the other hand relies on recommendation alone to direct the sharing of private information. Nguyen and Mynatt [11] address the problem of trust in pervasive computing environment. Their goal is to make the user more aware of privacy issues. The goal of enhancing consumer confidence in privacy practices of service providers has been explored by privacy seal programs such as TrustE (<http://www.truste.org>) that relies heavily on policy statements similar to P3P statement. The privacy standard P3P [7] provides a framework for service providers to express their privacy policies to the user with the goal that a user can form a reasoned opinion about the state of her privacy at the service provider. However, P3P does not provide mechanisms by which policies are enforced. Nor can policies be used to verify or prove that the services accurately reflect the original policies.

Assurance levels have often been used as a reflection for degree of trust. The idea of formally determining trust in computer systems was first developed within the defense community. The U.S. Department of Defense Trusted Computer System Evaluation Criteria [12] defines a grading of seven assurance levels for systems that convey a gradation of the degree of trust in systems. The assurance levels are discrete and have significance only within the particular context of the department of defense. The European Commission's Information Technology Security Evaluation Criteria [13] and, more recently, the International Standard Organization's Common Criteria [14] also use similar notion of assurance levels to evaluate competence of systems. These assurance levels allow one to evaluate a system in isolation but fail when a system's dependability is affected by that of another system. Our work on the other hand provides an algorithm for determining trust levels and has the ability to address this particular problem.

A number of logic-based formalisms of trust have been proposed by researchers [15], [16], [17], [18]. Forms of first order logic and modal logic or its modification have been variously used to model trust in these cases. Simple relational formulas of the form  $T_{a,b}$  (stating  $A$  trusts  $B$ ) are used to model trust between two entities. Abdul-Rahman and Hailes [15] propose a trust model based on “reputation” that allows artificial agents to reason about trustworthiness and allows real people to automate that process. Jones and Firozabadi [16] model trust as the issue of reliability of an agent’s transmission. They use a variant of modal logic to model various trust scenarios. Yahalom et al. [19] propose a formal model for expressing new trust relations in authentication protocols, together with an algorithm for deriving trust relations from recommendations from existing ones. The work does not define what is meant by trust. Beth et al. [20] extend the ideas presented by Yahalom et al. to include relative trust. This work proposes a method for extracting trust values based on experiences and recommendations and also a method for deriving new trust values from existing ones within a network of trust relationships. Jøsang [21] proposes a model for trust based on a general model for expressing relatively uncertain beliefs about the truth of statements. This model does not have a mechanism for monitoring trust relationships to re-evaluate their constraints. Marsh [22] introduces a computational model for trust in the distributed artificial intelligence community. In this model trust is represented as a real number in the range  $[-1,1]$ . The author, however, does not consider what factors determine the value of trust. Our p-Trust model, on the other hand, has similar features as these models and, in addition, does not suffer from the inability to re-evaluate trust relationships.

There have been some research that address trust management in P2P systems. Most of this works are based on reputation-based trust. In [23] the reputation is measured as the number of complaints a peer receives. This type of metric is vulnerable to wrong information put by the malicious peers and can be misleading to measure trust. The P2PRep [24] proposes a protocol where a peer chooses a reputable servant (peer) on the basis of the polling opinion of other peers. The scheme uses quality of service offer and the past experience to compute the reputation without formally defining the trust metric. EigenTrust [25] describes an algorithm to decrease the number of downloads of inauthentic files and assigns each peer a global trust value based on peer’s history of upload. This trust metric, we believe, is very restrictive and can not capture all relevant information that are needed to impose a trust. PeerTrust [26] is a unified approach where the trust metric is based on peer’s feedback, number of transactions that a peer has, credibility of feedback source, transaction context, and community context. Our scheme is close to this approach. However, the scheme was not intended for privacy preservation of peers.

A number of research projects have investigated the problem of trust negotiations for web-based applications [27], [28], [29]. PSPL [27] provides a language to express access control policies for services and release policies for client and server. However, PSPL, unlike our work, does not address the

problem of estimating the level of threat to individual privacy. TrustBuilder [28] defines a set of negotiation protocols that define the ordering of messages and the type of information the messages will contain. Bertino et al. proposes Trust- $\mathcal{X}$  [29] as a comprehensive framework for trust negotiations. They provide both a language for encoding policies and certificates, and a system architecture. Recently, in [30] they have extended the system by adding techniques for preserving privacy, such as the selective disclosure of credentials and the integration with the P3P platform.

## VI. CONCLUSION AND FUTURE WORK

We have presented a trust-based approach to allow personal control over privacy in a P2P framework. The p-Trust model allows a peer in the P2P network to measure the degree of confidence she can have on another peer to protect her privacy during a transaction with that peer. The model considers four factors while evaluating trustworthiness of peers. It takes into account the behavioral history of the target peer, the target peer’s attributes, reputation of the target peer in terms of feedback from non-attributable sources, and recommendation feedback from other trustworthy peers. This trust level which we call p-Trust, is evaluated in a distributed and dynamic manner. There is no central database to store the p-Trust data. Instead, peers contain partial or complete data that is needed to compute p-Trust of the target entity. This way of distributed and replicated storage provide greater availability. However, data from any arbitrary peer are not used for the computation. The framework supports sharing of data only between ‘friends’. Friends are chosen from all peers in a two-level p-Trust threshold based random filtering process. This way of choosing peers to share data minimizes the chance of getting affected by malicious peers.

We plan to extend this work in future. We are currently working on defining a composition operator on p-Trust relationships. This will help a peer to evaluate p-Trust of a group of peers acting as a single trustee. It will also include p-Trust evaluation of two peers collaboratively acting as one single truster to evaluate a third peer’s p-Trust. We also have plan to evaluate the model by implementing it and then analyzing its performance for privacy protection.

## ACKNOWLEDGMENT

This work was partially supported by the U.S. Air Force Research Laboratory (AFRL) and the Federal Aviation Administration (FAA) under contract F30602-03-1-0101, by the U.S. Air Force Office of Scientific Research under contract FA 9550-07-1-0042 and by the National Science Foundation (NSF) of the USA under grant IIS-0242258. Any opinions, findings, and conclusions or recommendations expressed in this publication are solely those of the authors and do not necessarily represent those of the AFRL, the FAA, or the NSF.

## REFERENCES

- [1] M. A. L. Cranor and J. Reagle, “Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences,” *Communications of the ACM*, 1999.

- [2] O. Berthold, H. Federrath, and M. Kohntopp, "Project Anonymity and Unobservability in the Internet," in *Proceedings of the Workshop on Freedom and Privacy by Design*, Toronto, Canada, April 2000, pp. 57–65.
- [3] F. Lategan and M. Oliver, "On Granting Limited Access to Private Information," *Communications of the ACM*, pp. 21–25, May 2001.
- [4] A. Kobsa and J. Schreck, "Privacy Through Pseudonymity in User-Adaptive Systems," *ACM Transactions on Internet Technology*, vol. 3, no. 2, pp. 149–183, May 2003.
- [5] D. Kristol, "HTTP Cookies: Standards, Privacy, and Policies," *ACM Transactions on Internet Technology*, vol. 1, no. 2, pp. 151–198, November 2001.
- [6] S. Srinivasan, "On Piracy and Privacy," *IEEE Computer*, pp. 36–38, July 2003.
- [7] L. Cranor *et al.*, "The Platform for Privacy Preferences 1.1 (P3P 1.1)," World Wide Web Consortium, Tech. Rep., February 2004.
- [8] A. Hirschman, "Three Ways of Complicating Some Categories of Economic Discourse," *American Economic Review*, vol. 74, no. 2, 1984.
- [9] J. Goecks and E. Mynatt, "Enabling Privacy Management in Ubiquitous Computing Environments Through Trust and Reputation," in *Proceedings of CSCW 2002 Workshop on Privacy in Digital Environments*, New Orleans, LA, November 2002.
- [10] B. Shand, N. Dimmock, and J. Bacon, "Trust for Ubiquitous, Transparent Collaboration," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications*, Ft. Worth, TX, March 2003.
- [11] D. Nguyen and E. Mynatt, "Privacy Mirrors: Understanding and Shaping Socio-technical Ubiquitous Computing Systems," Georgia Institute of Technology, Technical Report GIT-GVU-02-16, 2002.
- [12] *Trusted Computer System Evaluation Criteria*, U.S. Department of Defense, December 1985, Department of Defense Standard DOD 5200-28-STD.
- [13] *Information Technology Security Evaluation Criteria*, Office for Official Publications of the European Communities, June 1991.
- [14] *Common Criteria for Information Technology Security Evaluation*, International Standards Organization, August 1999, CCIMB-99-031, CCIMB-99-032, CCIMB-99-033.
- [15] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000.
- [16] A. Jones and B. Firozabadi, "On the Characterization of a Trusting Agent – Aspects of a Formal Approach," in *Trust and Deception in Virtual Societies*, C. Castelfranchi and Y. Tan, Eds. Kluwer Academic Publishers, 2000.
- [17] N. Li, J. Mitchell, and W. Winsborough, "Design of a Role-Based Trust-Management Framework," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2002, pp. 114–130.
- [18] P. Rangan, "An Axiomatic Basis of Trust in Distributed Systems," in *Proceedings of the 1988 IEEE Computer Society Symposium on Security and Privacy*, Oakland, CA, April 1988, pp. 204–211.
- [19] R. Yahalom, B. Klein, and T. Beth, "Trust Relationship in Secure Systems: A Distributed Authentication Perspective," in *Proceedings of the 1993 IEEE Computer Society Symposium on Security and Privacy*, Oakland, CA, May 1993, pp. 150–164.
- [20] T. Beth, M. Borchherding, and B. Klein, "Valuation of Trust in Open Networks," in *Proceedings of the 3rd European Symposium on Research in Computer Security*, Brighton, UK, November 1994, pp. 3–18.
- [21] A. Jøsang, "A Subjective Metric of Authentication," in *Proceedings of the 5th European Symposium on Research in Computer Security*, Louvain-la-Neuve, Belgium, September 1998.
- [22] S. Marsh, "Formalising Trust as a Computational Concept," Ph.D. dissertation, University of Stirling, 1994.
- [23] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," in *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, GA, November 2001.
- [24] F. Cornelli, E. Damiani, S. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," in *Proceedings of the 11th International Conference on World Wide Web*, Honolulu, Hawaii, May 2002.
- [25] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proceedings of the 12th International Conference on World Wide Web*, Budapest, Hungary, May 2003, pp. 640–651.
- [26] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, July 2004.
- [27] P. Bonatti and P. Samarati, "Regulating Access Services and Information Release on the Web," in *7th ACM Conference on Computer and Communications Security*, Athens, Greece, November 2000, pp. 134–143.
- [28] T. Yu, M. Winslett, and K. E. Seamons, "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation," *ACM Transactions on Information and System Security*, vol. 6, no. 1, pp. 1–42, February 2003.
- [29] E. Bertino, E. Ferrari, and A. Squicciarini, "Trust-X: A Peer to Peer Framework for Trust Establishment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 827–842, July 2004.
- [30] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Privacy Preserving Trust Negotiations," in *4th International Workshop on Privacy Enhancing Technologies*, Toronto, Canada, May 2004, pp. 283–301.

**Sudip Chakraborty** is currently a Ph.D candidate at Computer Science Department of Colorado State University, USA. He received his M.Tech in Computer Science from Indian Statistical Institute, India in 2001. He received his M.Sc and B.Sc in Mathematics from University of Calcutta, India, in 1999 and 1997, respectively. His main research interests include trust modeling, application of trust in security, specifically in access control, privacy, mobile networks, and in pervasive computing. He is a student member of ACM.

**Dr. Indrajit Ray** is an Assistant Professor in the Computer Science Department at Colorado State University. He joined the faculty of Computer Science in August 2001. He teaches courses in computer networks, database systems and computer security. His main research interests are in the areas of security models, database and network security and computer forensics. His research is supported by the U.S. National Science Foundation, the U.S. Federal Aviation Administration, U.S. Air Force Research Laboratory and the U.S. Air Force Office of Scientific Research. He has served on the program committees of a number of national and international conferences. He serves on the editorial board of Digital Investigations. He is also the current chair of the IFIP TC-11 Working Group 11.9 on Digital Forensics. He is a member of IEEE, IEEE CS, ACM, ACM SACMAT and IFIP WG 11.3 and 11.9.