Proceedings of the 2006 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY, 21-23 June 2006

Allowing Finer Control Over Privacy Using Trust as a Benchmark

Sudip Chakraborty Indrajit Ray

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.**

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 some confidence that her privacy will be protected at least to the extent she feels comfortable with.

There have been some research that address trust management in P2P systems. Most of this works are based on reputation-based trust. In [2] 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 [3] 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 [4] 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 [5] is a unified approach where the trust metric is based on peer's feedback, number of transactions that a peer has, credebility of feedback source, transaction context, and community context. However the scheme was not intended for privacy preservation of peers.

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 other discloses her personally identifying information to various degrees. The user should have a comfort level with this disclosure. Thus, before commencing on an activity, the user will try determine to what degree she trusts or distrusts the other entity to protect her privacy. Unlike binary trust models, trust in this

S. Chakraborty and I. Ray: Colorado State University, Fort Collins, CO. Email: {sudip, indrajit}@cs.colostate.edu

ISBN 1-4244-0130-5/06/ $20.00 ©2006 IEEE

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 peers – less reliable and/or p-Trusted neighbors within the network. The model restricts feedbacks from a group of p-Trusted neighbors and friends only. We propose a mutual-consent based two-level random filtering process of peers which is based on p-Trust thresholds to choose p-Trusted neighbors and friends. 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. However, malicious peers can affect the p-Trust decision by providing misleading information. This is reduced by the two-level random filtering process. Since p-Trust threshold vary from peer to peer this makes it difficult for the malicious peer to change its behavior in such a way so that it gets a 'neighbor invitation' or a 'friendship invitation' from the target peer.

The rest of the paper is organized as follows. In section II we introduce the p-Trust model. We describe the different parameters that are used to compute p-Trust values in the context of user privacy. Section III presents our approach to controlling personal privacy using the p-Trust model. 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. Finally section V concludes the paper.

II. The Privacy Trust (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 [6], [7], [8], [9], [10]. 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. To measure this degree we propose to use a trust-based model called p-Trust. p-Trust is trust as applied to user privacy. We begin by defining p-Trust and p-Distrust.

Definition 2: p-Trust is defined to be the belief in the competence of an entity to act dependably and securely in the context of user privacy.

Definition 3: p-Distrust is defined to be the belief in the incompetence of an entity to act dependably and securely in the context of user privacy.

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. As we elaborate on the model this will become more clear.

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. 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 $(A \xrightarrow{c} B) = [{}_AI^c_B,{}_A P^c_B,{}_A REP^c_B,{}_R REC^c_B]$, where ${}_AI^c_B$ represents the magnitude of A's interaction about B in context c, ${}_AP^c_B$ represents B's properties relevant to c as evaluated by A, ${}_AREP^c_B$ represents B's reputation in c and ${}_RREC^c_B$ represents the cumulative effect of all B's recommendations to A from different peers.

To compute a p-Trust relationship we assume that each of these four factors 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*. To indicate a lack of value due to insufficient information for any component we use the special symbol $\perp$.

A. The interactions *component*

We model *interactions* in terms of the number of events encountered by a peer A regarding a trustee peer B in the context c within a specified period of time $[t_0, t_n]$.

Definition 4: The *interactions* of a truster about a trustee is defined as the measure of the cumulative effect of a number of events that were encountered by the truster with respect to the trustee in a particular context and over a specified period of time.

An event can be p-Trust-positive, p-Trust-negative or, p-Trust-neutral depending on whether it contributes toward a p-Trust-positive interaction, a p-Trust-negative interaction or, a p-Trust-neutral interaction. We believe, events far back in time does not count as strongly as very recent events for computing p-Trust values. Hence we introduce the concept of *interaction policy* which specifies a length of time interval subdivided into non-overlapping intervals.

Definition 5: An *interaction policy* specifies a totally ordered set of non-overlapping time intervals together with a set of non-negative weights corresponding to each element in the set of time intervals.

Recent intervals in the interaction policy are given more weight than those far back. The whole time period $[t_0, t_n]$ is divided in such intervals and the truster A keeps a log of events occurring in these intervals. If e^i_k denote the k^{th} event in the i^{th} interval, then $v^i_k = +1$, if $e^i_k \in \mathcal{P}$, $v^i_k = -1$ if $e^i_k \in \mathcal{Q}$ or $v^i_k = 0$, if $e^i_k \in \mathcal{N}$, where, $\mathcal{P}$ = set of all p-Trust-positive events, $\mathcal{Q}$ = set of all p-Trust-negative events and $\mathcal{N}$ = set of all p-Trust-neutral events.

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 neutral for the time interval. If n_j is the number of events that occurred in the j^{th} time interval, then $IN_j = \perp$, if there is no event in $[t_{j-1}, t_j]$, and $IN_j = \sum_{k=1}^{n_j} v^j_k$, otherwise. If w_i is a non-negative weight assigned to i^{th} interval, the *interactions* of A with regards to B in context c is given by

$$_A\mathrm{I}^c_B = \frac{\sum_{i=1}^{n} w_i IN_i}{\sum_{i=1}^{n} n_i} \tag{1}$$

B. *The* properties *component*

Definition 6: 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 must assign a value between -1 and +1 for each attribute of the trustee. How the values are assigned, depends on the scheme and 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

$$_AP^c_B = \frac{\sum_{i=1}^{k} pv_i}{k} \tag{2}$$

where $pv_i \in [-1, 1]$, $\forall i = 1, 2, \ldots, k$. pv_i is the value assigned to i^{th} attribute of B and is determined by the underlying property evaluation policy of the truster.

C. *The* reputation *component*

Definition 7: 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

$$_AREP^c_B = \frac{r_p - r_n}{r_p + r_n} \tag{3}$$

where r_p is number of p-Trust-positive reputations and r_n is number of p-Trust-negative reputations about B.

D. *The* recommendation *component*

Definition 8: 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. Truster A uses the "level of p-Trust" 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, $\mathbf{v}(A \xrightarrow{c} j)$ = 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

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

E. *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. So 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:

$$\begin{aligned}(A \xrightarrow{c} B)^N &= \mathbf{W} \odot (A \xrightarrow{c} B) \\ &= [W_I,\ W_P,\ W_{REP},\ W_{REC}] \odot [{}_AI^c_B, \\ &\quad {}_AP^c_B,\ {}_AREP^c_B,\ {}_RREP^c_B] \\ &= [W_I \cdot {}_AI^c_B,\ W_P \cdot {}_AP^c_B,\ W_{REP} \cdot_A REP^c_B, \\ &\quad W_{REC} \cdot {}_RREC^c_B] \\ &= [{}_A\hat{I}^c_B,\ {}_A\hat{P}^c_B,\ {}_AR\hat{E}P^c_B\ {}_RR\hat{E}C^c_B]\end{aligned}$$

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

F. 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 9: The *value* of a normalized trust relationship $(A \xrightarrow{c} B)^N = [{}_A\hat{I}^c_B, {}_A\hat{P}^c_B, {}_AR\hat{E}P^c_B, {}_RR\hat{E}C^c_B]$ is a number in the range $[-1,1] \cup \{\perp\}$ and is defined as

$$\mathbf{v}(A \xrightarrow{c} B)^N = {}_A\hat{I}^c_B + {}_A\hat{P}^c_B + {}_AR\hat{E}P^c_B + {}_RR\hat{E}C^c_B \qquad (5)$$

The value of 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*.

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 user can have a reasonable control over her privacy while interacting with a server. Note that the server can be a peer user. 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 other's 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 him to engender that trust and complete the negotiation.

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 that 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 p-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 while computing the reputation and recommendation, we propose the concept of 'p-Trusted neighbors' and 'friends'. The 'p-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 'p-Trusted neighbor' but j denies to be so, then the relationship breaks and neither i nor j can consider each other as neighbors. The same is true for friends.

A. 'p-Trusted neighbors' and 'friends'

Let there be m peers in the network. To choose the p-Trusted neighbor set, a peer i sets up a neighbor_p-Trust threshold τ_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 asking to be her neighbor. If i gets back n acceptance messages from each of these peers, then these n peers are considered as "p-Trusted neighbors" of i. The decision is taken by the other peer on the basis of the p-Trust she has 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 of peers. Therefore 'p-Trusted neighbors' can be defined as

Definition 10: The p-Trusted neighbors of a peer i is the set NBR_i^t (t for 'p-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, we can write $NBR_i^t = \{j \mid \mathbf{v}(i \xrightarrow{c} j)^N \geq \tau_i^{nbr} \ \wedge \ \mathbf{v}(j \xrightarrow{c} i)^N \geq \tau_j^{nbr}\}$ The condition 'j accepts i's neighbor invitation is formally represented with the expression $\mathbf{v}(j \xrightarrow{c} i)^N \geq \tau_j^n br$.

However, it may not always be possible to find n peers who satisfy both conditions (p-Trust value with at least τ_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 τ_i^{nbr} or n or both and run the algorithm again to choose the peers. It is preferable to reset n rather than setting τ_i^{nbr}. This is because τ_i^{nbr} gives the 'confidence' level that i should have on her neighbors. It should not be lowered just because enough peers do not 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 to choose neighbors. When i receives a similar neighbor invitation from j, i can accept or reject it based on the condition $\mathbf{v}(i \xrightarrow{c} j)^N \ \geq \ \tau_i^{nbr}$ or $\mathbf{v}(i \xrightarrow{c} j)^N < \tau_i^{nbr}$.

Algorithm 1 Formation of NBR_i^t for a peer i

```
Require: Set of peers in the network $\neq \emptyset$
  Let $|set\ of\ peers| = m$
  Set $\tau_i^{nbr}$ and $n$ $(0 < n \leq m)$
  $NBR_i^t = \{\}$
  for $k = 1$ to $m$ do
    if $\mathbf{v}(i \xrightarrow{c} k)^N < \tau_i^{nbr}$ then
      $k = k + 1$
    else
      Send 'neighbor invitation' to $k^{th}$ neighbor
      if Receives an acceptance notification then
        $NBR_i^t = NBR_i^t \cup \{k^{th}$ neighbor$\}$
        $k = k + 1$
      else
        $k = k + 1$
      end if
    end if
  end for
  if $|NBR_i^t| = n$ then
    EXIT
  else
    if $|NBR_i^t| > n$ then
      Select $n$ members randomly from $NBR_i^t$ and discard others
    else
      if $|NBR_i^t| \neq 0 \ \wedge \ |NBR_i^t| < n$ then
        Case 1: Return $NBR_i^t$ and EXIT
        Case 2: Set $n = n'(< n)$ and repeat the algorithm
      else
        if $|NBR_i^t| = 0 \ \wedge \ n \neq 0$ then
          Set $n = n'(< n)$ and repeat the algorithm
        else
          Set $\tau_i^{nbr} = \tau_i^{nbr'}$ where $\tau_i^{nbr'} < \tau_i^{nbr}$
          Repeat the algorithm
        end if
      end if
    end if
  end if
```

We posit that all 'p-Trusted neighbors' may not be 'friends' of peer i. The 'friends' are those p-Trusted neighbors who are more 'close' to i, i.e. i has greater confidence and importance on their feedbacks. A friend k, unlike a non-friend p-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 p-Trusted neighbors. After choosing n p-Trusted neighbors, the client i sets a friend_p-Trust threshold τ_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 11: Friends of a peer i is the set FR_i of all peers j who satisfy the following conditions: (i) j is a 'p-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, $FR_i = \{j \in NBR_i^t \mid \mathbf{v}(i \xrightarrow{c} j)^N \geq \tau_i^{fr} \ \wedge \ \mathbf{v}(j \xrightarrow{c} i)^N \geq \tau_j^{fr}\}$

The algorithm for forming the friends set is similar to the algorithm 1. Figure 1 shows the p-Trusted neighbors and friends of peers i and j where i acts as the client and j acts as the server.

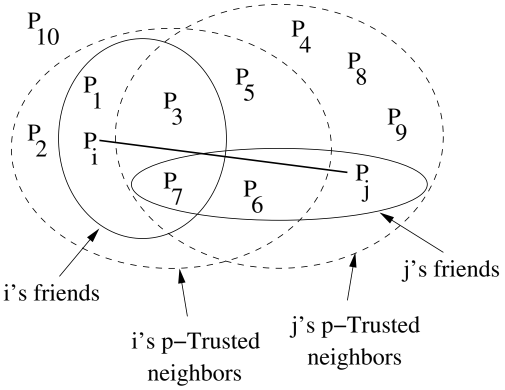

Fig. 1. Trusted neighbors and friends of peers i and j

It is clear from the above discussion that $\tau^{fr} > \tau^{nbr}$. Note, we could have chosen the 'p-Trusted neighbors' as the 'friends'. However, we choose to use another filter and create a higher tier of p-Trusted peers to diminish the effect of malicious peers. If there is a group of malicious peers in the population of m peers, there is a chance of some of them being included in the population of p-Trusted neighbors though a random selection has been used. The second filter, i.e. the second random selection of f peers further diminishes the chance of a malicious peer in the 'vicinity' of i. However, this type of two level random selection procedure does not completely remove the effect of malicious peers. There will be still some chance of a malicious peer being include in the 'friends' set. The chance will depend on the parameters $n, \tau^{nbr}, f, \tau^{fr}$ and the distribution of malicious peers in the network.

B. 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).

B.1 Computation of *properties*

To quantify the 'properties' component of the p-Trust relation, the client i first needs to gather certain information about the service provider 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 the client may look further for following information: (i) Version – What versions of SSL, if any, are supported by j, (ii) Encryption method – Which encryption method is being used in the communication. Under this category the client can have specific criteria for the following: (a) Encryption algorithm (like AES or DES or RSA), (b) Key type and size (symmetric key or asymmetric key; 56-bit or 128-bit or 512-bit), (iii) Message digest algorithm – What type of message digest algorithm is used (e.g. MD5 or SHA), (iv) Authentication – What authentication mode is used (e.g. authentication of both peers, or only j's authentication or, it is totally anonymous), (v) Key exchange – Which key exchange algorithm is used (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. For example 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 presence of following policies in the 'policy document' of j (i) *Data collection policy* – Explaining how j is going to receive and 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 j is going to disclose it? (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 the 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.

B.2 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 client'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 p-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. Otherwise the 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.

B.3 Computation of the *reputation* and *recommendation*

To compute these two components a peer needs information from other peers. We assume that i asks only the p-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 p-Trusted neighbors. Therefore the information collected from those peers are almost non-attributable to i though she knows about the source. Alternatively, information from p-Trusted neighbors carry more importance to i to make her p-Trust decision. Note, a recommendation from a p-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.

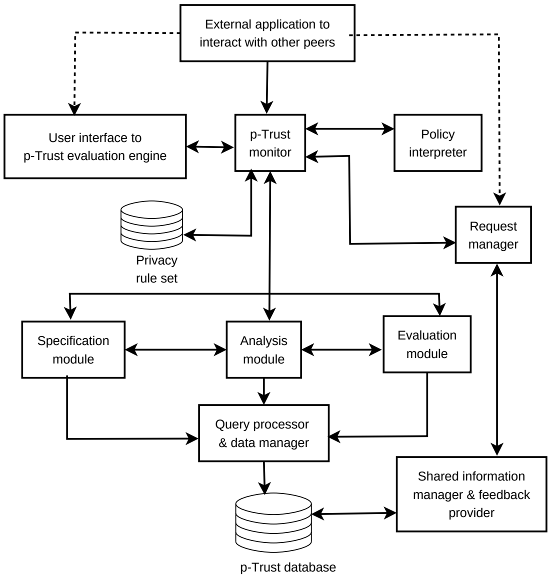
External application to interact with other peers
User interface to p-Trust evaluation engine
p-Trust monitor
Policy interpreter
Privacy rule set
Request manager
Specification module
Analysis module
Evaluation module
Query processor & data manager
Shared information manager & feedback provider
p-Trust database

Fig. 2. Components of p-Trust evaluation module of a peer

Recommendation is computed using information from p-Trusted neighbors and friends. Client i sends 'recommendation requests' to all her p-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. p-Trust Management System Architecture

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 the p-Trust data is stored across the network in a distributed manner where peers have partial or all information. Figure 2 gives a schematic diagram of a peer's p-Trust management module. This 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 that are needed to compute p-Trust value. This includes values of the parameters, event-logs, property information about specific p-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 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 peers to gather information relevant to evaluation of p-Trust. 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 This module processes p-Trust queries from either the p-Trust monitor or from the client peer.

Evaluation module This module retrieves information about the components from the database and also 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 client 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 client 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 client i may store some or all of these data for future use. When client 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 other peers.

V. Conclusions

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 communication 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 and less reliable sources, and recommendation feedback from more trustworthy peers. This 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 peer are not accepted. The framwork does not support sharing of data other than '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.

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 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] 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 (CIKM 2001)*, (Atlanta, Georgia), ACM, November 2001.

[3] F. Cornelli, E. Damiani, S. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," in *Proceedings of the 11th Internatioal Conference on World Wide Web*, 2002.

[4] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proceedings of the 12th Internatioal Conference on World Wide Web*, pp. 640–651, May 2003.

[5] L. Xiong and L. Liu, "PeerTrust: Supporting Reptation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 843–857, July 2004.

[6] 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 / Conference on Freedom and Privacy 2000 CFPI*, (Toronto, Canada), pp. 57–65, April 4-7 2000.

[7] F. Lategan and M. Oliver, "On Granting Limited Access to Private Information," *Communications of the ACM*, pp. 21–25, May 2001.

[8] A.Kobsa and J.Schreck, "Privacy Through Pseudonymity in User-Adaptive Systems," *ACM Transactions on Internet Technology*, vol. 3, pp. 149–183, May 2003.

[9] D. Kristol, "HTTP Cookies: Standards, Privacy, and Policies," *ACM Transactions on Internet Technology*, vol. 1, pp. 151–198, November 2001.

[10] S.Srinivasan, "On Piracy and Privacy," *IEEE Computer*, pp. 36–38, July 2003.