CAT: A Context-Aware Trust Model for Open and Dynamic Systems

Mohammad Gias Uddin
Dept. of Electrical and Computer Engineering
Queen's University, Kingston
Ontario, Canada K7L 3N6
gias@cs.queensu.ca

Mohammad Zulkernine
School of Computing
Queen's University, Kingston
Ontario, Canada K7L 3N6
mzulker@cs.queensu.ca

Sheikh Iqbal Ahamed
Dept. of Math, Statistics and Computer Science
Marquette University
Wisconsin, USA
iq@mscs.mu.edu

ABSTRACT

The requirements for spontaneous interactions in open and dynamic systems create security issues and necessitate the incorporation of trust management into each software entity to make decisions. Trust encompasses various quality attributes (*e.g.*, security, competence, honesty) and helps in making appropriate decisions. In this paper, we present CAT, an interaction-based Context-Aware Trust model for open and dynamic systems by considering services as contexts. We identify a number of trust properties including context and risk awareness and address those in the proposed model. A context-similarity parameter is proposed to make decisions in similar situations. A time-based ageing parameter is introduced to change trust values over time without any further interaction. We present direct and indirect recommendations and apply path-based ageing on indirect recommendations. A mechanism to calculate the accuracy of recommendations is described. This accuracy is used to differentiate between reliable and unreliable recommendations in the total trust calculation.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*

Keywords

Trust, Recommendation, Interaction.

1. INTRODUCTION

Mutual collaboration is necessary in open and dynamic systems, where entities rely on each other for achieving goals, utilizing resources, and performing tasks. Openness of such systems is necessary as entities need to interact with each other. Dynamism is inherent as entities can join and leave the system at any time. Entities in such a decentralized architecture need to successfully decide which entity they should interact with and which they should not. A promising and widely used way to deal with this problem is trust management [10]. Trust covers a number of quality attributes (*e.g.*, security, competence, honesty) and helps in dynamic decision making. *"Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action"* [25]. The trusting agent is called the trustor entity, and the trusted agent is called the trustee entity. A 'context' is a situation, which influences in the building of a trust relationship between the trustor and the trustee. For example, a service provider can provide services to its users. The users build trust relationships with the service provider based on the provided services. In this case, the services can be considered as contexts [18]. A user is an entity, as it is free to do whatever it wishes, however, it has to operate within a software system. The system itself is an entity, as it provides specific services to its users. Compared to the traditional *'hard security'* mechanisms (*e.g.*, access control policies, intrusion detection), trust is considered as *'soft security'* [19]. Due to the unavoidable uncertainty of open and dynamic systems, it may not be always effective to use *'hard security'* to protect software entities from malicious and unwanted incidents. For example, a malicious client may try to illegally access others' resources. Such security breach can be detected using intrusion detection and can be avoided using access control mechanism. However, the misbehaving entity is usually not punished in either cases, as it still has access to the server. It is obvious that the client is not trusted and should be considered carefully before granting server resources anymore. Evidence-based trust calculation [10] allows the analysis of interactions to enable better reasoning about future interactions. By doing so, a trust model does not provide security in the traditional sense, rather its purpose is to mitigate harm from future such incidents [21].

The type of protection we are concerned cannot be addressed using trust models based on cryptography and secret key sharing techniques [22]. Moreover, a central trust server [9] is not feasible as the compromise of this server puts all the entities of the system in grave danger. Therefore, we need a decentralized trust model incorporated to each interacting entity. By using this model, entities can avoid the risks of relying on a central authority and gain the flexibility of making autonomic decisions. Several trust models have been proposed for this purpose including FIRE [1], Beta Reputation System [23], B-Trust [24], REGRET [10], TRAVOS [12], and [2–7,11,13–20]. However, these models suffer from one or more of the following shortcomings. First, they are not risk-aware and do not judge interactions from risk perspective. Second, they do not address the dynamic aspect of trust that trust decreases over time without any further interaction. Third, they do not propose any mech-

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

anism to detect unreliable recommendations. Fourth, they treat all recommendations equally. However, recommendations from closer and known entities should be given more preference. Fifth, most of them consider general trust and overlook the multi-faceted nature of trust. For example, trust on an entity for context c_i may be different from trust for another context c_j.

In this paper, we propose an interaction-based trust model called CAT (Context-Aware Trust) for open and dynamic systems by addressing the above mentioned shortcomings. We identify some well known interaction-based trust properties for open and dynamic systems and incorporate those properties into CAT. We propose mechanisms to calculate trust from interactions based on rules. The rules are risk-aware and the calculation is context-aware. A context-similarity parameter is introduced, which provides similarity between contexts. The parameter is applicable, when the direct interaction value for a context is not available. To support the dynamism of trust, we introduce a time-based ageing parameter. This parameter reduces trust value over time, when no further interaction takes place between entities. We propose direct recommendations for known (neighbors) entities, indirect recommendations for unknown entities, and apply path-based ageing on indirect recommendations. A mechanism to calculate the accuracy of recommendations is described. This accuracy is used to differentiate between reliable and unreliable recommendations in the total trust calculation.

The rest of the paper is organized as follows. Section 2 presents CAT by first defining the trust properties and then providing the detailed calculation schemes. Section 3 compares the model with the state of the art of trust models. Finally, Section 4 concludes the work and provides a number of future research directions.

2. CAT: THE TRUST MODEL

CAT is based on interactions between entities. An interaction can have a number of trust rules. A trust rule tr encapsulates a particular condition that is necessary to determine the outcome O of an interaction I. Based on the trust rules, interactions are analyzed at run-time. Direct trust is derived from the outcomes of the interactions, which is updated after each interaction. Recommendations are collected and compared against the outcome of the corresponding interactions. This comparison follows an accuracy mechanism to differentiate between reliable and unreliable recommendations.

Before providing the detailed calculations used in CAT in Section 2.2, we elaborate the properties of CAT in Section 2.1.

2.1 Properties of CAT

CAT follows some well known interaction-based trust properties specified in [2, 3, 8, 10, 13, 14, 18]. The properties are as follows.

P1. Context-awareness (CA). Trust is context or service dependent. Trust of entity $E1$ on entity $E2$ for context c_i at time t does not imply the same trust for another context c_j.

$$T(E1, E2, c_i, t) \nRightarrow T(E1, E2, c_j, t),\ Where\ i \neq j.$$

P2. Rule-oriented (RO). Trust is condition or rule-dependent. The total outcome O of an interaction I is a function of all the outcomes as measured by the corresponding trust rules $(tr_i, tr_2, \ldots, tr_n)$.

$$O(I) = f(tr_1, tr_2, \ldots, tr_i, \ldots, tr_n).$$

P3. Risk-awareness (RA). For each interaction, the trust rules capture possible risks associated with it. The higher the risk of an outcome, the higher is the trust level assigned to the interaction.

P4. Recommendation-based (RB). An entity may need to take recommendations from other entities to (i) know about an unknown trustee, (ii) confirm belief on the trustee, and (iii) make decisions based on not only self observation but also from others' observations. The total trust of $E1$ on $E2$ is a function of direct trust $(T_D(E1, E2, c_i, t))$ and recommendation trust $(R(E1, E2, c_i, t))$.

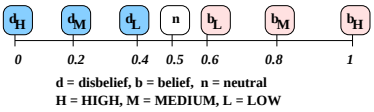

Figure 1: Trust values used in CAT

$$\exists_{c_i}(T(E1, E2, c_i, t) = f(T_D(E1, E2, c_i, t), R(E1, E2, c_i, t))).$$

P5. Recommendation-filtration (RF). All the recommendations may not be reliable. Therefore, CAT analyzes the recommendations after each interaction. CAT discards recommendations, which did not have the acceptable level of accuracy in the past.

P6. Semi-transitive (ST). If $E1$ trusts $E2$ and $E2$ trusts $E3$, it cannot be concluded that $E1$ will trust $E3$. However, if $E2$ recommends $E3$, then there may be a partial transitive trust relationship between $E1$ and $E3$.

P7. Non-symmetric (NS). For any context c_i, trust of $E1$ on $E2$ at time t does not imply the same trust of $E2$ on $E1$ for c_i.

$$\forall_{c_i}(T(E1, E2, c_i, t) \nRightarrow T(E2, E1, c_i, t)).$$

P8. Dynamic (D). The trust value of $E1$ on $E2$ changes over time due to new interaction and new recommendation.

P9. Time-based ageing (TA). Without any further interaction, trust value of $E1$ on $E2$ decreases over time.

$$\forall_{c_i}(T(E1, E2, c_i, t) > T(E1, E2, c_i, t + \triangle t)).$$

P10. Path-based ageing (PA). Recommendations from closer and known entities are given more importance than the recommendations from the entities which are unknown or farther away.

2.2 Trust Calculation Using CAT

2.2.1 *Interaction Trust: Calculating Confidence (μ)*

'Confidence' is a quantified satisfaction of a trustor on the competence, security, and honesty of a trustee. The calculation of trust begins whenever an interaction takes place. The outcome of an interaction is used to calculate the confidence from the interaction. The calculation of confidence is thus the first step of the direct trust calculation (Section 2.2.2). Each interaction has certain trust rule(s) to examine the outcome of the interaction at run-time. Each of the trust rules of an interaction is assigned a particular trust value. For example, a trust rule can be assigned high (H), medium (M), or low (L) trust value. The satisfaction of each trust rule awards the trustee a certain level of belief (*i.e.*, b_H, b_M, b_L). However, the violation of a trust rule penalizes the trustee with the certain level of disbelief (*i.e.*, d_H, d_M, d_L). Fig. 1 provides the trust values used in the model. The highest trust value is 1 (denoted by high belief b_H), and the lowest trust value is 0 (denoted by high disbelief, d_H). The total belief (I_b) of $E1$ on $E2$ for context c_i at time t from interaction I is calculated using Eq. 1. Similarly, the total disbelief (I_d) is calculated using Eq. 2. $b_H(rn)$ denotes that the the trust rule indexed as rn has high belief outcome. In Eq. 3, n_{b_H} is the total number of trust rules with high belief outcome from I, and n_{d_L} is the total number of trust rules with low disbelief outcome. n_b is the total number of trust rules related to belief, and n_d is the total

number of the trust rules related to the disbelief in an interaction. The confidence (μ) of $E1$ on $E2$ from I about context c_i at time t is calculated using Eq. 4.

$$I_b(E1, E2, c_i, t) = \frac{\sum_{rn=0}^{n_{b_H}} b_H(rn) + \sum_{rn=0}^{n_{b_M}} b_M(rn) + \sum_{rn=0}^{n_{b_L}} b_L(rn)}{n_b}. \quad (1)$$

$$I_d(E1, E2, c_i, t) = \frac{\sum_{rn=0}^{n_{d_H}} d_H(rn) + \sum_{rn=0}^{n_{d_M}} d_M(rn) + \sum_{rn=0}^{n_{d_L}} d_L(rn)}{n_d}. \quad (2)$$

$$n_{b_H} + n_{b_M} + n_{b_L} = n_b, \quad n_{d_H} + n_{d_M} + n_{d_L} = n_d. \quad (3)$$

$$\mu(E1, E2, c_i, t) = w_b I_b(E1, E2, c_i, t) + w_d I_d(E1, E2, c_i, t), \quad Where,\ w_b + w_d = 1. \quad (4)$$

In Eq. 4, w_b and w_d (range $[0, 1]$) are weights assigned to I_b and I_d respectively. The trust rules having malicious outcomes are given higher priority (*i.e.*, H). Whenever such a malicious event occurs, the interaction is terminated. This run-time monitoring of interaction helps making decisions in two ways. First, the trustor becomes self-protected. Second, since the trustee is given the lowest trust value (*i.e.*, 0) for its misbehavior, the trustor may not interact with it in near future. The calculation of confidence thus satisfies context-awareness, rule-oriented, and risk-awareness.

2.2.2 Calculating Direct Trust (T_D)

The direct trust (T_D) of an entity ($E1$) on another entity ($E2$) is calculated by using the confidence gained from the interactions between them. The value of T_D changes after each interaction based on the outcome of the interaction. This satisfies the dynamism of trust. The calculation of direct trust is the most important, as it is based on self-observation and does not depend on recommendations. The direct trust of $E1$ on $E2$ for context c_i at time t is calculated using Eq. 5, where δ (range $[0, 1]$) is the weighting factor. When δ is given less weight (*e.g.*, for a fast changing environment), the latest confidence is preferred more than the previous confidences. The calculation of T_D has two options. When confidence value for a context c_i is found, T_D is calculated by taking the old direct trust value ($T_D(E1, E2, c_i, t)_o$) and the new confidence value. However, it is not always possible to get the direct trust value for a context c_i. Eq. 5 handles this problem by calculating direct trust using $\Re(c_i, c_j)$ (range $[0, 1]$). $\Re(c_i, c_j)$ is called as the context-similarity parameter. Eq. 6 calculates the generalized trust of $E1$ on $E2$, where $\rho(c_i)$ (range $[0, 1]$) is the priority of context c_i. This priority is necessary as it is highly unlikely that $E1$ will value each context equally in real world. Moreover, without any further interaction, direct trust decreases over time. Eq. 7 calculates recent direct trust $T_D(E1, E2, c_i, t)_r$ using the direct trust obtained at previous time t_o. The calculation is performed using a time-based ageing parameter $\gamma(t_o, t, c_i)$ (range $[0, 1]$).

$$T_D(E1, E2, c_i, t) = \begin{cases} \delta T_D(E1, E2, c_i, t)_o + (1 - \delta)\mu(E1, E2, c_i, t). \\ T_D(E1, E2, c_j, t)_o \Re(c_i, c_j), \\ if T_D(E1, E2, c_i, t)_o = \emptyset,\ \mu(E1, E2, c_i, t) = \emptyset. \end{cases} \quad (5)$$

$$T_D(E1, E2, t) = \frac{\sum_{i=0}^{n} \rho(c_i) T_D(E1, E2, c_i, t)}{\sum_{i=0}^{n} \rho(c_i)}. \quad (6)$$

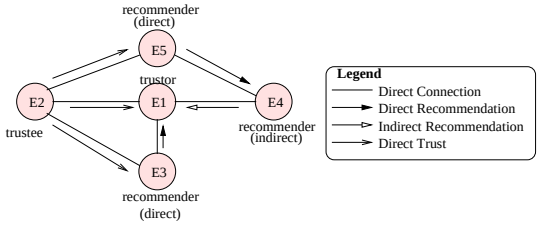

Figure 2: Direct and indirect recommendations in CAT

$$T_D(E1, E2, c_i, t)_r = T_D(E1, E2, c_i, t_o)\gamma(t_o, t, c_i). \quad (7)$$

Guidelines to Use Context-Similarity Parameter ($\Re(c_i, c_j)$). It provides similarity measurement between context c_i and context c_j. The calculation is performed using Eq. 8. The parameter is necessary, when no previous interaction value is available in the record database of a trustor ($E1$) for a trustee ($E2$) regarding context c_i. We assume that every context has some keywords to describe it. Suppose a file-server application has three types of services (*i.e.*, contexts): `uploadPDFFile` with keywords {`write, pdf, file`}, `uploadDOCFile` with keywords {`write, doc, file`}, `login` with keywords {`LoginInfo, userName, passWD`}. In Eq. 8, $\kappa(c_i)$ denotes the total number of keywords describing context c_i. The numerator takes total number of similar keywords in two contexts c_i and c_j. The denominator takes total number of distinct keywords between c_i and c_j. Therefore, for `login` and `uploadPDFFile` the value of $\Re(c_i, c_j)$ is 0 (*i.e.*, the contexts are not similar at all). However, for `uploadPDFFile` and `uploadDOCFile` the value is 0.75 (*i.e.*, if $E1$ trusts $E2$ for uploading `doc` files, it can almost trust $E2$ for uploading `PDF` files also). This parameter helps in making decisions in similar situations. The idea is to decrease the reliance on recommendations and to rely on self-observation more.

$$\Re(c_i, c_j) = \frac{\kappa(c_i) \cap \kappa(c_j)}{\kappa(c_i) \cup \kappa(c_j)}. \quad (8)$$

Guidelines to Use Time-Based Ageing Parameter ($\gamma(t_o, t, c_i)$). It is used to reduce a trust value over time, when no further interaction takes place. It is calculated using Eq. 9, where Υ (range $[0, 1]$) is the time-based ageing factor. t_o is the last time of interaction between entity $E1$ and $E2$ for c_i, and t is the time of the trust calculation to make a decision on $E2$ for c_i. t_u is the time unit considered for a particular context. Some example values of t_u are 1 for seconds, 60 for minutes, and 3600 for hours. The value of t_u may change depending on the context. For an almost static environment, where very less number of interactions take place, this value can be chosen as a month or even a year. The more Υ close to 1, the less will be the value of γ. The value of direct trust decreases with the decrease in γ. For considerably safe environments, where entities do not change their behavior rapidly, the value of Υ should be chosen close to 0. However, for very uncertain and insecure environments, this value should be chosen close to 1.

$$\gamma(t_o, t, c_i) = 1 - [\frac{(t - t_o)\Upsilon}{t}] * (1/t_u). \quad (9)$$

2.2.3 Calculating Recommendation Trust

We consider two types of recommendations, direct and indirect.

Recommendations obtained from neighbors are denoted as direct. The recommendations obtained from someone other than the neighbors are denoted as indirect. We consider the entities with which the trustor has direct connections as neighbors [18]. Fig. 2 describes the two types of recommendations. The recommendation from $E5$ on $E2$ to $E1$ is indirect as it comes through $E4$ (*i.e.*, path-length = 3). The recommendation from $E3$ on $E2$ to $E1$ is direct (*i.e.*, path-length = 2). A recommendation path is indirect, if it has path-length more than 2. An entity $E3$ provides the direct recommendation (R_D) to $E1$ for $E2$ about context c_i using Eq. 10. η (range $[1, 0]$) is the weighting factor an entity imposes on its direct trust for the purpose of recommendation. A recommendation value is at most equal to the direct trust value in CAT.

$$R_D(E3, E1, E2, c_i, t) = \eta T_D(E3, E2, c_i, t). \tag{10}$$

The indirect recommendation (R_I) from entity $E5$ to $E1$ on $E2$ about c_i at time t is calculated using Eq. 11, where $\vartheta(M, \Lambda, c_i)$ is the path-based ageing parameter. $\vartheta(M, \Lambda, c_i)$ is adapted from [18], but it is calculated in a different manner using Eq. 12. Λ is the maximum allowed path-length for this recommendation path, and Ψ is the distance-based ageing factor. Using this parameter, indirect recommendations are given less weight than direct recommendations. The original parameter proposed by [18] is $1 - \frac{(M-1)\Psi}{10}$. The problem with the original parameter is that it has 10 as denominator, and therefore, at some point it may become negative. For example, for M = 16, $\Psi = 0.9$, the value of the parameter is -0.35. We address this problem by using maximum allowed path length Λ instead of 10. The value of Λ can be changed based on preferences.

$$R_I(E5, E1, E2, c_i, t) = R_D(E5, E1, E2, c_i, t)\vartheta(M, \Lambda, c_i). \tag{11}$$

$$\vartheta(M, \Lambda, c_i) = 1 - \frac{(M-2)\Psi}{\Lambda},\ Where\ M \geqslant 2. \tag{12}$$

The accuracy (A) of $E3$ to $E1$ in providing a direct recommendation for $E2$ about context c_i is calculated using Eq. 13. Each entity keeps an accuracy table (AT), where it updates the accuracy of every recommendation after corresponding interaction. The update in the AT is performed using Eq. 15. ζ (range $[1, 0]$) puts importance to previous accuracy and current accuracy of a recommender for context c_i. This is necessary, since an entity may be completely wrong at one point. However, this cannot be used to conclude on all of its recommendations. Using Eq. 13, 14, and 15, unreliable recommendations are detected. A recommender is considered most reliable, if it has accuracy 1, and most unreliable, if it has accuracy 0. This satisfies the recommendation filtration property.

$$A_{R_D}(E3, E1, E2, c_i, t) = 1 - \Delta R_D(E3, E1, E2, c_i, t). \tag{13}$$

$$\Delta R_D(E3, E1, E2, c_i, t) = |R_D(E3, E1, E2, c_i, t) - T_D(E1, E2, c_i, t)|. \tag{14}$$

$$AT_{R_D}(E3, E1, E2, c_i, t) = \zeta AT_{R_D}(E3, E1, E2, c_i, t)_o + (1 - \zeta)\Delta R_D(E3, E1, E2, c_i, t). \tag{15}$$

Eq. 13, 14, and 15 are also used for calculating accuracy of indirect recommendations by replacing R_D by R_I. The calculation of recommendation points out that an entity can be considered based on recommendations to a certain extent. However, the recommendation has to be specified by the recommender. This satisfies the semi-transitive and recommendation-based trust properties.

2.2.4 Calculating Total Recommendation Trust

The total direct/indirect recommendation trust (θ) is calculated by binding recommendations with the corresponding accuracy of the recommenders. The total direct recommendation trust as measured by $E1$ on $E2$ based on the recommendation provided by $E3$ for context c_i at time t is calculated by Eq. 16. Similarly, indirect recommendation trust is calculated using Eq. 17. The total recommendation trust of $E1$ on $E2$ about context c_i is calculated using Eq. 18, where N_D is the total number of direct recommenders and N_I is the total number of indirect recommenders. The generalized recommendation trust of $E1$ on $E2$ at time t is measured by Eq. 19, where $\rho(c_i)$ is the priority of c_i.

$$\theta_{R_D}(E3, E1, E2, c_i, t) = R_D(E3, E1, E2, c_i, t)AT_{R_D}(E3, E1, E2, c_i, t). \tag{16}$$

$$\theta_{R_I}(E5, E1, E2, c_i, t) = R_I(E5, E1, E2, c_i, t)AT_{R_I}(E5, E1, E2, c_i, t). \tag{17}$$

$$R(E1, E2, c_i, t) = \frac{\sum_{e=0}^{N_D} \theta_{R_D}(e, E1, E2, c_i, t) + \sum_{e=0}^{N_I} \theta_{R_I}(e, E1, E2, c_i, t)}{N_D + N_I}. \tag{18}$$

$$R(E1, E2, t) = \frac{\sum_{i=0}^{m} \rho(c_i)R(E1, E2, c_i, t)}{\sum_{i=0}^{m} \rho(c_i)}. \tag{19}$$

2.2.5 Calculating Total Trust

After obtaining the direct trust and the recommendation trust on $E2$, $E1$ calculates total trust (T) on $E2$ for context c_i using Eq. 20. α (range $[0, 1]$) is the self-trust level. An entity can provide more value to α to rely on self-observation more. An entity can discard all the recommendations by using $\alpha = 1$.

$$T(E1, E2, c_i, t) = \begin{cases} \alpha T_D(E1, E2, c_i, t) + (1 - \alpha)R(E1, E2, c_i, t). \\ \alpha\lambda(E1, E2, t) + (1 - \alpha)R(E1, E2, c_i, t), \\ \quad if\ T_D(E1, E2, c_i, t) = \emptyset. \\ \lambda(E1, E2, t), \\ \quad if\ T_D(E1, E2, c_i, t) = \emptyset, R(E1, E2, c_i, t) = \emptyset. \end{cases}$$

$$Where,\ \lambda(E1, E2, t) = \begin{cases} T(E1, E2, t). \\ 0.5 \end{cases}$$

$$And,\ T(E1, E2, t) = \begin{cases} \alpha T_D(E1, E2, t) + (1 - \alpha)R(E1, E2, t). \\ T_D(E1, E2, t),\ \ if\ R(E1, E2, t) = \emptyset. \end{cases} \tag{20}$$

3. RELATED WORK

Trust-based dynamic decision making has been addressed from two angles: centralized and decentralized. While centralized models focus on building a central trust and reputation server, decentralized models aim to make each entity trust-aware.

Centralized Trust Models. In eBay trust and reputation model [8], users can provide feedback ratings after each interaction. The model is not context-aware, and so, big and small transactions are all treated equally. SPORAS [9] proposes solution to this problem by calculating and assigning trust based on some conditions. However, it discourages newcomers by providing very low trust value. Moreover, the central trust server can be dangerous, if it is compromised by malicious entities. In BambooTrust [7] reputation system, users can query about another user. The query requests are handled in

parallel machines to provide fast reply. BambooTrust is applicable for global computing, and it has the same risk of being compromised. Beta model [23] uses statistical representation of trust and reputation.

Decentralized Trust Models. These models focus on solving two problems: determining the most trustworthy service provider (*i.e.*, a client assesses the trustworthiness of the service providers) and deciding whether a service requester is trustworthy (*i.e.*, a server assesses the trustworthiness of the clients).

FIRE [1] selects service providers based on previous interactions and recommendations from other entities. Although FIRE uses interactions to gather trust, it does not rely on rules specifically to monitor it. Moreover, FIRE is not risk-aware and does not provide any ageing parameter. Billhardt *et al.* [4] propose trust model for selecting a service provider in service-oriented computing. Similar to them, we propose a context-similarity parameter. However, our parameter uses word-based similarity calculation, while they use organization-level service hierarchy. Toivonen *et al.* [26] propose an ontology-based context similarity method. However, while they use network ontology to calculate the similarity between two nodes, we use keywords. The keywords are used to describe and distinguish the services (*i.e.*, contexts). Therefore, we do not assume any hierarchical ontology-based node structure compared to [26]. TRAVOS [12] is a trust and reputation mechanism based on direct interactions and recommendations. TRAVOS identifies unreliable recommendations. Certain trust [5] separates humans from software entities and provides mapping between them for autonomic decision making. Humans are capable of stating their certainty over a trust opinion. The corresponding software representation uses the certainty for decision making and recommendation purpose. Fuzzy notations are used by Sherchan *et al.* [6] to make trust reasoning while choosing a service provider. Similar to them, we use rule-based trust reasoning in CAT. Bayesian trust models are used by [13] and [24]. In REGRET [10], users can rate each other after each interaction. It proposes a time-based ageing parameter. However, the parameter takes only the difference between the two times of an interaction and does not provide preference to the new time. A broker based decentralized reputation system is proposed in [11], which penalizes entities for providing false recommendations. An automatic trust prediction model is used in [17] to select service providers, where trustworthiness is measured from the advertised and monitored entity attributes. Trust and recommendations are formally defined and analyzed in [2,3,19] by incorporating belief, disbelief and uncertainty to each interaction.

Trust-based decision making for a service request in the server side is mostly handled for providing access to system resources. TBAC [15] and TRBAC [16] propose access control policies to server resources based on trust reasoning. Both of these approaches are risk-aware. [14] integrate trust with risk management to grant authorization. This model considers the utility of an outcome to grant access. [18] propose a formal trust model for pervasive computing to handle service request in mutual collaboration. We adapt path-based ageing from [18] and propose a different calculation scheme for the path-based ageing parameter. However, they do not use time-based ageing, context-similarity, and recommendation filtration. [20] operate on the online rating mechanism for e-commerce transaction (*e.g.*, eBay).

Table 1 compares and contrasts the above discussed related work with respect to some interaction-based trust properties of Section 2.1.

Table 1: Property-based placing of different trust models

Model	RO	CA	RA	RB	TA	PA	C/D*	RF
FIRE [1]	Y	Y	Y	Y	N	N	D	N
FTMAS [2]	N	N	N	Y	N	N	D	N
TDAS [3]	N	Y	N	Y	N	N	D	Y
TSOC [4]	N	Y	N	Y	N	N	D	N
CertainTrust [5]	N	Y	Y	Y	N	N	D	N
Fuzzy [6]	N	Y	Y	Y	N	N	D	Y
BambooTrust [7]	N	Y	Y	N	N	N	C	N
SPORAS [9]	N	N	N	Y	N	N	C	N
REGRET [10]	N	Y	N	Y	N	N	D	N
ICRM [11]	N	N	N	Y	N	N	D	Y
TRAVOS [12]	N	N	N	Y	N	N	D	Y
TRP2P [13]	N	Y	N	Y	N	N	D	Y
TBRM [14]	N	Y	Y	Y	N	N	D	N
TRBAC [16]	Y	Y	Y	Y	N	N	D	N
TBAC [15]	Y	Y	Y	Y	N	N	D	Y
ATP [17]	N	N	N	N	N	N	D	N
OFTM [18]	N	Y	Y	Y	N	Y	D	N
EMDR [19]	N	N	N	Y	N	N	D	Y
PeerTrust [20]	N	Y	N	Y	N	N	D	Y
Beta [23]	N	Y	N	Y	N	N	C	N
B-trust [24]	N	Y	Y	Y	N	N	D	N
our work, CAT	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**D**	**Y**

*C = Centralized, D = Decentralized

4. CONCLUSIONS

In this work, we introduce an interaction-based trust model called CAT (Context-Aware Trust) for open and dynamic systems. We identify some prominent interaction-based trust properties and integrate those into CAT. CAT uses rule-based trust calculation, direct trust calculation, and direct and indirect recommendation calculation. We introduce a context-similarity parameter in direct trust calculation, which is necessary when direct interaction value for a particular context is not available. A time-based ageing parameter is introduced, which reduces trust over time to address dynamic nature of trust calculation. We propose direct and indirect recommendations and apply path-based ageing on indirect recommendations. We consider recommendations from known entities (neighbors) as direct. A mechanism to calculate the accuracy of recommendations is described. This accuracy is used to differentiate between reliable and unreliable recommendations in the total trust calculation. The incorporation of CAT to a system makes the system autonomic and trust-aware. CAT also makes the system adaptive, robust, responsive, and flexible to the changes in the environment.

In CAT, it is considered that the trust network is secure from malicious attacks, and therefore, the recommendation from an entity is considered solely from that entity. However, in real situations, this might not be the case. For example, CAT does not consider defense against the Sybil attacks [28]. In Sybil attack, a malicious entity creates a large number of pseudonymous entities for the purpose of providing false recommendations. The focus of the malicious entity is to influence the trust-based decision making. We consider that the problem of pseudonymity is handled by the authentication system. The authentication systems is responsible for providing unique identifications to the interacting entities. For example, in a resource sharing grid, the users belong to particular organizations [15]. The organizations subscribe to specific service providers for services. Therefore, the authentication of the users are performed basically by the organizations. We use CAT for evidence-based trust decision making, where the trust rules are used to monitor the interactions between the entities. Based on the monitoring, trustworthiness of the interacted entities are measured. However, CAT measures the accuracy of the recommendations after each interaction. Since the accuracy is performed by calculating the difference between direct trust and recommendations, the unreliable recommendations are detected and discarded accordingly. In this way, we can also partially address the colluding malicious node attack [27]. In the colluding malicious node attack, the ma-

licious nodes combine to provide false recommendations about an entity. To provide defense against the above mentioned attacks, CAT needs to be examined and extended properly. Another limitation of CAT is that the collection of recommendations may take considerable time for a long recommendation chain. We are currently developing a trust monitor framework using CAT by addressing the above issues.

Acknowledgement

This research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

5. REFERENCES

[1] Huynh, T., Jennings, N., Shadbolt, N. FIRE: An integrated trust and reputation model for open multi-agent systems, in *Proc of the 16th European Conference on Artificial Intelligence*, Valencia, Spain, 2004. 18–22.

[2] Wang, Y., Singh, M. Formal trust model for multiagent systems, in *Proc of the 20th Int Joint Conference on Artificial Intelligence*, India, 2007. 1551–1556.

[3] Wang, Y., Singh, M. Trust representation and aggregation in distributed agent systems, in *Proc of the 21st Int Conference on Artificial Intelligence*, Boston, 2006. 6pp.

[4] Billhardt, H., Hermoso, R., Ossowski, S., Centeno, R. Trust based service provider selection in open environments, in *Proc of the 22nd Annual ACM Symposium on Applied Computing*, Seoul, Korea, 2007. ACM Press: 1375–1380.

[5] Ries, S. Certain trust: a trust model for users and agents, in *Proc of the 22nd Annual ACM Symposium on Applied Computing*, Seoul, Korea, 2007. ACM Press: 1599–1604.

[6] Sherchan, W., Loke, S., Krishnaswamy, S. A fuzzy model for reasoning about reputation in web services, in *Proc of the 21st Annual ACM Symposium on Applied Computing*, Dijon, France, 2006. ACM Press: 1886–1892.

[7] Kotsovinos, E., Williams, A. BambooTrust: practical scalable trust management for global public computing, in *Proc of the 21st Annual ACM Symposium on Applied Computing*, Dijon, France, 2006. ACM Press: 1893–1897.

[8] Grandison, T., Sloman, M. A survey of trust in internet application, in *IEEE Communications Surveys & Tutorials*, September 2000, **3**(4). IEEE Communications Society: 15pp.

[9] Zacharia, G., Maes, P. Trust management through reputation mechanisms, in *Applied Artificial Intelligence*, **14**(9), 2000. Taylor and Francis Ltd: 881-908.

[10] Sabater, J., Sierra,C. REGRET:A reputation model for gregarious societies, in *Proc of the 4th workshop on deception fraud and trust in agent societies*, Montreal, Canada, 2001. 61–70.

[11] Jurca, R., Faltings, B. Towards incentive-compatible reputation management, in *Proceedings of the AAMS Workshop on Trust, reputation and security: theories and practice (LNCS v2631)*, Bologna, Italy, 2002. Springer Berlin/ Heidelberg: 138–147.

[12] Teacy, W., Patel, J., Jennings, N., Luck, M. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model, in *Proc of 4th Int Joint Conference on Autonomous Agents and Multiagent Systems*, Bologna, Italy, 2002. 997–1004.

[13] Wang, Y., Vassileva, J. Trust and reputation model in peer-to-peer networks, in *Proc of the 3rd Int Conference on Peer-to-Peer Computing*, Sweden, 2003. IEEE CS Press: 150–157.

[14] Lin, C., Varadharajan, V. Trust based risk management for distributed system security - a new approach, in *Proc of the 1st Int Conference on Availability, Reliability and Security*, Vienna, Austria, 2006. IEEE CS Press: 6–13.

[15] Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., Ingram, D., Moody, K. Using trust and risk in role-based access control policies, in *Proc of the 9th ACM symposium on Access control models and technologies*, New York, USA, 2004. ACM Press: 156–162.

[16] Dimmock, N., Bacon, J., Ingram, D., Moody, K. Risk models for trust-based access control (TBAC), in *Proc of the 3rd Annual Conference on Trust Management (LNCS v3477)*, Rocquencourt, France, 2005. Springer Berlin/ Heidelberg: 364–371.

[17] Capra, L. Musolesi, M. Autonomic trust prediction for pervasive systems, in *Proc of the 20th Int Conference on Advanced Information Networking and Applications*, Vienna, Austria, 2006. IEEE CS Press: 481–488.

[18] Haque, M., Ahamed, S. An omnipresent formal trust model (FTM) for pervasive computing environment, in *Proc of the 31st Annual IEEE Int Computer Software and Applications Conference*, Beijing, China, 2007. IEEE CS Press: 49–56.

[19] Yu, B., Singh, M. An evidential model of distributed reputation mechanism, in *Proc of the 1st Int Joint Conference on Autonomous Agents and multiagent systems*, Bologna, Italy, 2002. ACM Press: 294–301.

[20] Xiong, L., Liu, L. Building trust in decentralized peer-to-peer electronic communities, in *Proc of the 5th Int Conference on Electronic Commerce Research*, Montreal, Canada, 2002. 15pp.

[21] English, C., Terzis, S., Nixon, P. Towards self-protecting ubiquitous systems: monitoring trust-based interactions, in *Personal and Ubiquitous Computing*, **10**(1), 2005. Springer-Verlag: UK, 50–54.

[22] Jøsang, A. The right type of trust for distributed systems, in *Proc of the 1996 Workshop on New Security Paradigms*, California, USA, 1996. ACM Press: 119–131.

[23] Jøsang, A, Ismail, R. The beta reputation system, in *Proc of the 15th Bled Conference on Electronic Commerce*, Slovenia, 2002. 14pp.

[24] Quercia, D., Hailes, S., Capra, L. B-trust: Bayesian trust framework for pervasive computing, in *Proc of the 4th Int Conference on Trust Management (LNCS v3986)*, Tuscany, Italy, 2006. Springer Berlin/ Heidelberg: 298–312.

[25] Gambetta, D. Can we trust trust?, in *Trust: Making and Breaking Cooperative Relations*, Gambetta, D. (ed.), Chapter 13, 1988. University of Oxford: 213–237.

[26] Toivonen, S., Lenzini, G., Uusitalo, I. Context-aware trust evaluation functions for dynamic reconfigurable systems, in *Proc of the Workshop on Models of Trust for the Web*, Edinburgh, Scotland, 2006. 11pp.

[27] Damon, M. Doug, S., Dirk, G. A mechanism for detecting and responding to misbehaving nodes in wireless networks, in *Proc of the 4th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, California, USA, 2007. IEEE CS Press: 678–684.

[28] Douceur, J. The sybil attack, in *Proc of the 1st Int Workshop on Peer-to-Peer Systems (LNCS v2429)*, Cambridge, MA, USA, 2002. Springer Berlin/ Heidelberg: 251–260.