A Computational Dynamic Trust Model for User Authorization

Yuhui Zhong, Bharat Bhargava, Yi Lu, and Pelin Angin

Abstract—Development of authorization mechanisms for secure information access by a large community of users in an open environment is an important problem in the ever-growing Internet world. In this paper we propose a computational dynamic trust model for user authorization, rooted in findings from social science. Unlike most existing computational trust models, this model distinguishes trusting belief in integrity from that in competence in different contexts and accounts for subjectivity in the evaluation of a particular trustee by different trusters. Simulation studies were conducted to compare the performance of the proposed integrity belief model with other trust models from the literature for different user behavior patterns. Experiments show that the proposed model achieves higher performance than other models especially in predicting the behavior of unstable users.

Index Terms—Authorization, human factors, security, trust

1 Introduction

The everyday increasing wealth of information available online has made secure information access mechanisms an indispensable part of information systems today. The mainstream research efforts for user authorization mechanisms in environments where a potential user's permission set is not predefined, mostly focus on role-based access control (RBAC), which divides the authorization process into the role-permission and user-role assignment. RBAC in modern systems uses digital identity as evidence about a user to grant access to resources the user is entitled to. However, holding evidence does not necessarily certify a user's good behavior. For example, when a credit card company is deciding whether to issue a credit card to an individual, it does not only require evidence such as social security number and home address, but also checks the credit score, representing the belief about the applicant, formed based on previous behavior. Such belief, which we call dynamic trusting belief, can be used to measure the possibility that a user will not conduct harmful actions.

In this work, we propose a computational dynamic trust model for user authorization. Mechanisms for building trusting belief using the first-hand (direct experience) as well as second-hand information (recommendation and reputation) are integrated into the model. The contributions of the model to computational trust literature are:

- The model is rooted in findings from social science, i.e., it provides automated trust management that mimics trusting behaviors in the society, bringing trust computation for the digital world closer to the evaluation of trust in the real world.
- Unlike other trust models in the literature, the proposed model accounts for different types of trust. Specifically, it distinguishes trusting belief in integrity from that in competence.
- The model takes into account the subjectivity of trust ratings by different entities, and introduces a mechanism to eliminate the impact of subjectivity in reputation aggregation.

Empirical evaluation supports that the distinction between competence and integrity trust is necessary in decision-making [15]. In many circumstances, these attributes are not equally important. Distinguishing between integrity and competence allows the model to make more informed and fine-grained authorization decisions in different contexts. Some real-world examples are as follows:

1) On an online auction site, the competence trust of a seller can be determined by how quickly the seller ships an item, packaging/item quality etc., each being a different competence type. The integrity trust can be determined by whether he/she sells buyers' information to other parties without buyer consent. In the case of an urgent purchase, a seller with low integrity trust can be authorized if he/she has high competence trust.
2) For an online travel agency site, competence consists of elements such as finding the best car deals, the best hotel deals, the best flight deals etc., whereas integrity trust is based on factors like whether the site puts fraudulent charges on the customers' accounts. In a context where better deals are valued higher than the potential fraud risks, an agency with lower integrity trust could be preferred due to higher competence.
3) For a web service, the competence trust can include factors such as response time, quality of results etc., whereas integrity trust can depend on whether the service outsources requests to untrusted parties.

- *Y. Zhong and Y. Lu are with Microsoft Corporation, Seattle, WA. E-mail: yuhuiz@hotmail.com, yilu.cn@gmail.com.*
- *B. Bhargava and P. Angin are with the Department of Computer Science, Purdue University, West Lafayette, IN 47907. E-mail: {bb, pangin}@cs.purdue.edu.*

Manuscript received 6 June 2013; revised 8 Feb. 2014; accepted 18 Feb. 2014. Date of publication 27 Feb. 2014; date of current version 16 Jan. 2015.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2014.2309126

While government agencies would usually prefer high integrity in web services, high-competence services with low integrity could be authorized for real-time missions.

Experimental evaluation of the proposed integrity belief model in a simulated environment of entities with different behavior patterns suggests that the model is able to provide better estimations of integrity trust behavior than other major trust computation models, especially in the case of trustees with changing behavior.

2 Related Work

2.1 McKnight's Trust Model

The social trust model, which guides the design of the computational model in this paper, was proposed by McKnight and Chervany [16] after surveying more than 60 papers across a wide range of disciplines. It has been validated via empirical study [15]. This model defines five conceptual trust types: trusting behavior, trusting intention, trusting belief, institution-based trust, and disposition to trust. *Trusting behavior* is an action that increases a truster's risk or makes the truster vulnerable to the trustee.

Trusting intention indicates that a truster is willing to engage in trusting behaviors with the trustee. A trusting intention implies a trust decision and leads to a trusting behavior. Two subtypes of trusting intention are:

1) Willingness to depend: the volitional preparedness to make oneself vulnerable to the trustee.
2) Subjective probability of depending: the likelihood that a truster will depend on a trustee.

Trusting belief is a truster's subjective belief in the fact that a trustee has attributes beneficial to the truster. The following are the four attributes used most often:

1) Competence: a trustee has the ability or expertise to perform certain tasks.
2) Benevolence: a trustee cares about a truster's interests.
3) Integrity: a trustee is honest and keeps commitments.
4) Predictability: a trustee's actions are sufficiently consistent.

Institution-based trust is the belief that proper structural conditions are in place to enhance the probability of achieving a successful outcome. Two subtypes of institution-based trust are:

1) Structural assurance: the belief that structures deployed promote positive outcomes. Structures include guarantees, regulations, promises etc.
2) Situational normality: the belief that the properly ordered environments facilitate success outcomes.

Disposition to trust characterizes a truster's general propensity to depend on others across a broad spectrum of situations. Two subtypes of disposition to trust are:

1) Faith in human: The assumptions about a general trustee's integrity, competence, and benevolence.
2) Trusting stance: A truster's strategy to depend on trustees despite his trusting belief about them.

Trust intention and trusting belief are situation and trustee specific. Institution-based trust is situation specific. Disposition to trust is independent of situation and trustee. Trusting belief positively relates to trusting intention, which in turn results in the trusting behavior. Institution-based trust positively affects trusting belief and trusting intention. Structural assurance is more related to trusting intention while situational normality affects both. Disposition to trust positively influences institution-based trust, trusting belief and trusting intention. Faith in humanity impacts trusting belief. Trusting stance influences trusting intention.

2.2 Computational Trust Models

The problem of establishing and maintaining dynamic trust has attracted many research efforts. One of the first attempts trying to formalize trust in computer science was made by Marsh [13]. The model introduced the concepts widely used by other researchers such as context and situational trust.

Many existing reputation models and security mechanisms rely on a social network structure [1]. Pujol et al. propose an approach to extract reputation from the social network topology that encodes reputation information [19]. Walter et al. [22] propose a dynamic trust model for social networks, based on the concept of feedback centrality. The model, which enables computing trust between two disconnected nodes in the network through their neighbor nodes, is suitable for application to recommender systems. Lang [9] proposes a trust model for access control in P2P networks, based on the assumption of transitivity of trust in social networks, where a simple mathematical model based on fuzzy set membership is used to calculate the trustworthiness of each node in a trust graph symbolizing interactions between network nodes. Similarly, Long and Joshi [11] propose a Bayesian reputation calculation model for nodes in a P2P network, based on the history of interactions between nodes. Wang and Wang [23] propose a simple trust model for P2P networks, which combines the local trust from a node's experience with the recommendation of other nodes to calculate global trust. The model does not take the time of feedback into consideration, which causes the model to fail in the case of nodes with changing behavior. Reliance on a social network structure limits wide applicability of the mentioned approaches, especially for user authorization.

FCTrust [8] uses transaction density and similarity to calculate a measure of credibility of each recommender in a P2P network. Its main disadvantages are that it has to retrieve all transactions within a certain time period to calculate trust, which imposes a big performance penalty, and that it does not distinguish between recent and old transactions. SFTrust [25] is a double trust metric model for unstructured P2P networks, separating service trust from feedback trust. Its use of a static weight for combining local and recommendation trust fails to capture node specific behavior.

Das and Islam [3] propose a dynamic trust computation model for secure communication in multi-agent systems, integrating parameters like feedback credibility, agent similarity, and direct/indirect trust/recent/historical trust into trust computation. Matt et al. [14] introduce a method for modeling the trust of a given agent in a multi-agent system by combining statistical information regarding the past behavior of the agent with the agent's expected future behavior.

A distributed personalized reputation management approach for e-commerce is proposed by Yu and Singh [24]. The authors adopt ideas from Dempster-Shafer theory of evidence to represent and evaluate reputation. If two principals "a" and "b" have direct interactions, b evaluates a's reputation based on the ratings of these interactions. Otherwise, b queries a TrustNet for other principals' local beliefs about a. The reputation of "a" is computed based on the gathered local beliefs using Dempster-Shafer theory.

Sabater and Sierra propose a reputation model called the *Regret* system [20] for gregarious societies. The authors assume that a principal owns a set of sociograms describing the social relations in the environment along individual, social and ontological dimensions. The performance highly depends on the underlying sociograms, although how to build sociograms is not discussed.

The above mentioned trust computation approaches do not consider "context" as a factor affecting the value of trust, which prevents an accurate representation for real life situations. Skopik et al. [21] propose a dynamic trust model for complex service-oriented architectures based on fuzzy logic. Zhu et al. [26] introduce a dynamic role based access control model for grid computing. The model determines authorization for a specific user based on its role, task and the context, where the authorization decision is updated dynamically by a monitoring module keeping track of user attributes, service attributes and the environment. Fan et al. [5] propose a similar trust model for grid computing, which focuses on the dynamic change of roles of services. Liu and Liu [10] propose a Bayesian trust evaluation model for dynamic authorization in a federation environment, where the only context information is the domain from which authorization is requested. Ma and He [12] propose a genetic algorithm for evaluating trust in distributed applications. Nagarajan and Varadharajan [18] propose a security model for trusted platform based services based on evaluation of past evidence with an exponential time decay function. The model evaluates trust separately for each property of each component of a platform, similar to the consideration of competence trust in our proposed model. Although these approaches integrate context into trust computation, their application is limited to specific domains different from the one considered in our work.

3 Overview of the Trust Model

The trust model we propose in this paper distinguishes integrity trust from competence trust. Competence trust is the trusting belief in a trustee's ability or expertise to perform certain tasks in a specific situation. Integrity trust is the belief that a trustee is honest and acts in favor of the truster. Integrity and benevolence in social trust models are combined together. Predictability is attached to a competence or integrity belief as a secondary measure.

The elements of the model environment, as seen in Fig. 1, include two main types of actors, namely *trusters* and *trustees*, a database of trust information, and different contexts, which depend on the concerns of a truster and the competence of a trustee. For the online auction site example in Section 1, let us assume that buyer B needs to decide whether to authorize seller S to charge his credit card for an item I (authorize access to his credit card/contact information). The elements of the model in this case are:

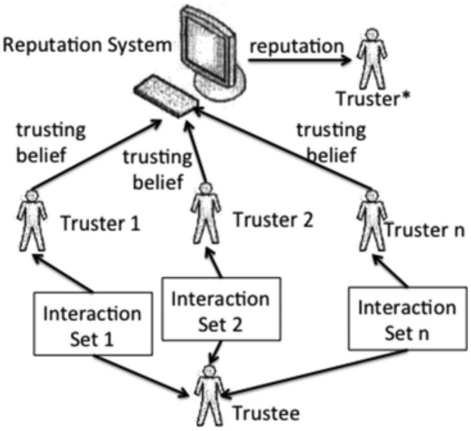


Fig. 1. Model elements.

- *Trusters* are the buyers registered to the auction site.
- *Trustees* are the sellers registered to the auction site.
- The *context* states how important for B the shipping, packaging and item quality competences of S for item I are. It also states how important for B the integrity of S is for this transaction.
- B can gather trust information about S from a database maintained by the site or a trusted third party. This information includes the ratings that S received from buyers (including B's previous ratings, if any) for competence in shipping, packaging and quality of I as well as S's integrity. It also includes the ratings of buyers (including B) for sellers other than S in different contexts and ratings of S for different items. Trust evaluation is recorded in the database when a buyer rates a transaction with a seller on the site.

3.1 Context and Trusting Belief

Context. Trust is environment-specific [13]. Both trusters' concern and trustees' behavior vary from one situation to another. These situations are called contexts. A truster can specify the minimum trusting belief needed for a specific context. Direct experience information is maintained for each individual context to hasten belief updating.

In this model, a truster has one integrity trust per trustee in all contexts. If a trustee disappoints a truster, the misbehavior lowers the truster's integrity belief in him. For integrity trust, contexts do not need to be distinguished. Competence trust is context-dependent. The fact that Bob is an excellent professor does not support to trust him as a chief. A representation is devised to identify the competence type and level needed in a context. Two functions that relate contexts are defined.

Let $\mathrm{S_c}$ denote the universe consisting of all types of competences of interest $\{\mathrm{c_1, c_2, \ldots, c_n}\}$, where each $\mathrm{c_i}$ is a different competence type. For example, $\mathrm{S_c} = \{$cooking, teaching, writing, ...$\}$. Let $\mathrm{M_c}$ denote the measurement of

Context ::= *<contextId, cType, cLevel>*
Trusting belief ::= *<truster, trustee, Type>*
Type ::= *IntegrityTrust* | *CompetenceTrust*
IntegrityTrust ::= *<value, predictability>*
CompetenceTrust ::= *<contextId, value, predictability, iNumber>*
value ::= $R^{[0,1]}$
predictability ::= $R^{[0,1]}$
iNumber ::= N^{+}

Fig. 2. Representation of context and trusting belief.

a competence type c. For example, $\mathrm{M_{cooking}}$ = {very bad, bad, ok, good, excellent}. A partial order $<: \mathrm{M} \times \mathrm{M} \rightarrow$ {true, false} and a function *dis*: $\mathrm{M} \times \mathrm{M} \rightarrow [0, 1]$ are defined on $\mathrm{M_c}$. For two elements $\mathrm{m_i}$ and $\mathrm{m_j}$ in $\mathrm{M_c}, \mathrm{m_j}$ is the higher competence level if $\mathrm{m_i} < \mathrm{m_j}$. The *dis* function measures the numerical distance between two elements (its two arguments), outputting a value between 0 and 1. $\mathrm{m_k}$ is closer to $\mathrm{m_i}$ than $\mathrm{m_j}$ is, if $\mathrm{dis(m_i, m_j) > dis(m_i, m_k)}$.

As shown in Fig. 2, a context representation is composed of *contextId*, *cType*, *cLevel*. contextId is a unique identifier assigned to each context. cType and cLevel are the competence type and level required respectively. cType assumes a value c from S_c. The domain of cLevel is M_c. A context needs one type of competence.

Two functions, *hContxt*: *contextId* × *contextId* → {*true*, *false*} and *simLCTX*: *contextId* × *contextId* × $R^{[0,1]}$ → {*true*, *false*}, are defined as follows. Here, $\mathrm{R}^{[0,1]}$ denotes a real number ranging from 0 and 1. *ct* and cl are abbreviations of cType and cLevel.

$$hContxt(c_1, c_2) = \begin{cases} true, & c_1.ct = c_2.ct \ and \ c_1.cl < c_2.cl, \\ false, & otherwise, \end{cases} \tag{1}$$

$$simLCTX(c_1, c_2, \delta) = \begin{cases} true, & c_1.ct = c_2.ct \ and \ dis(c_1.cl, c_2.cl) < \delta, \\ false, & otherwise. \end{cases} \tag{2}$$

If $hContxt(c_1, c_2)$ is true, c_2 requires the same type of competence with higher level as c_1 does. SimLCTX specifies whether the levels required for two contexts with the same type are sufficiently close to each other.

Trusting belief. Beliefs in two attributes, competence and integrity, are separated. Context identifier is included for competence belief. Values of both beliefs are real numbers ranging from 0 to 1. The higher the value, the more a truster believes in a trustee. *Predictability* is a positive real number. It characterizes the goodness of belief formed. The smaller the predictability or uncertainty, the more confident a truster is about the associated belief value. Both the variability of a trustee's behaviors and lack of observations negatively impact the goodness of belief formed. *iNumber* in competence belief records the number of observations accumulated. Trusting beliefs can be classified into initial and continuous trust. Initial trust is the belief established before a truster t_1 interacts with a trustee u_1. Continuous trust is the belief after t_1 has had appropriate direct experience with u_1.

TABLE 1
Trust Model Notation

$TC^{v}_{t_1 \to u_1}(c)$, $TC^{p}_{t_1 \to u_1}(c)$:	t_1's initial or continuous trusting belief in $\mathrm{u_1}$'s competence in context c.
$DTC^{v}_{t_1 \to u_1}(c)$, $DTC^{p}_{t_1 \to u_1}(c)$:	t_1's competence belief about $\mathrm{u_1}$ in c based on direct experience (called direct competence trust).
$RC^{v}_{u_1}(c)$, $RC^{p}_{u_1}(c)$:	$\mathrm{u_1}$'s competence reputation in context c.
$TI^{v}_{t_1 \to u_1}$, $TI^{p}_{t_1 \to u_1}$:	$\mathrm{t_1}$'s initial or continuous trusting belief in $\mathrm{u_1}$'s integrity.
$DTI^{v}_{t_1 \to u_1}$, $DTI^{p}_{t_1 \to u_1}$:	t_1's integrity belief about $\mathrm{u_1}$ based on direct experience (direct integrity trust).
$RI^{v}_{u_1}$, $RI^{p}_{u_1}$:	u_1's integrity reputation.

3.2 Operations Defined on Trust Model

This section presents the operations defined on the trust model. The notations in Table 1 are used for presentation. The notation with superscript v is the value of a belief. The one with superscript p is the associated predictability.

Direct trust for competence denoted by $DTC^{v}_{t_1 \to u_1}(c)$ is null, if t_1 has not interacted with u_1 in context c. Direct trust for integrity denoted by $DTI^{v}_{t_1 \to u_1}(c)$ is null if $\mathrm{t_1}$ had no direct experience with $\mathrm{u_1}$ before. Otherwise, it is a real number in the range of [0, 1]. Competence reputation denoted by $RC^{v}_{u_1}(c)$ is null, if no truster knows about $\mathrm{u_1}$ in context c. Integrity reputation denoted by $RI^{v}_{u_1}$ is null, if no trusters interacted with $\mathrm{u_1}$ before. Otherwise, they are real numbers in the range of [0,1]. Reputation is an aggregation of trust beliefs from different trusters. Details of competence and integrity reputation are presented below.

The trust model defines four types of operations:

[Operation 1.] Form direct trusting belief ($DTC^{v}_{t_1 \to u_1}(c)$, $DTC^{p}_{t_1 \to u_1}(c)$) or ($DTI^{v}_{t_1 \to u_1}, DTI^{p}_{t_1 \to u_1}$). A method that computes a direct trusting belief is called a *belief formation* method. This work assumes that after each interaction, a truster rates the competence and/or integrity of her counterpart (i.e., the trustee). Ratings are assigned by comparing a truster's expectation defined in a contract with a trustee's performance [6]. Competence and integrity of the trustee are assumed to be determinant factors of the corresponding ratings. A belief formation method takes a competence or integrity rating sequence assigned to a trustee as the input. It outputs the corresponding direct trusting belief about the trustee.

[Operation 2.] Aggregate trusting beliefs about u_1 from multiple trusters to his reputation ($RC^{v}_{u_1}(c), RC^{p}_{u_1}(c)$) or ($RI^{v}_{u_1}, RI^{p}_{u_1}$). A method that evaluates reputation is called a *reputation aggregation* method. A reputation aggregation method takes trusting beliefs from different trusters as input.

[Operation 3.] Build and test initial trust. This operation computes ($TC^{v}_{t_1 \to u_1}(c), TC^{p}_{t_1 \to u_1}(c)$) or ($TI^{v}_{t_1 \to u_1}, TI^{p}_{t_1 \to u_1}$). The results are compared with two constants δ_c and δ_p respectively. This is a more general form of Operation 1, where no direct interaction between $\mathrm{u_1}$ and $\mathrm{t_1}$ is required in context c, and is used for the initialization of the competence or integrity belief value for the pair $(\mathrm{t_1, u_1})$ in this context.

[Operation 4.] Update and test continuous trust. This operation computes ($TC^{v}_{t_1 \to u_1}(c), TC^{p}_{t_1 \to u_1}(c)$) or ($TI^{v}_{t_1 \to u_1}, TI^{p}_{t_1 \to u_1}$).

TABLE 2
Test a Competence Trusting Belief

	$TC^{v}_{t_1\rightarrow u_1}(c)\ \geq \delta_c$	$TC^{v}_{t_1\rightarrow u_1}(c)\ < \delta_c$
$TC^{p}_{t_1\rightarrow u_1}(c)\ \leq \delta_p$	*true*	*false*
$TC^{p}_{t_1\rightarrow u_1}(c)\ > \delta_p$	*uncertain*	*false*

The results are compared with two constants δ_c and δ_p respectively. This is the same as Operation 3 except that it is used for updating beliefs, not initializing.

Belief formation and reputation aggregation are atomic operations. They are presented in the next two sections. The last two operations are needed for user authorization. They output a test result according to Table 2 or Table 3.

The methods that can be used to build a trusting belief are summarized below, followed by a discussion of using these methods to establish initial or continuous trust.

3.2.1 Methods to Build a Trusting Belief

Seven methods that can be used to build competence trust:

[M1.] Form trusting belief based on direct experience in a specific context.

$$Precondition\text{: } DTC^{v}_{t_1\rightarrow u_1}(c) \neq null$$

$$TC^{v}_{t_1\rightarrow u_1}(c) := DTC^{v}_{t_1\rightarrow u_1}(c), \tag{3a}$$

$$TC^{p}_{t_1\rightarrow u_1}(c) := DTC^{p}_{t_1\rightarrow u_1}(c). \tag{3b}$$

[M2.] Consider direct trust about u_1 in contexts that require a higher competence level than c. Use the maximum value and minimum predictability.

$$Precondition\text{: } \exists c' \left(hContxt\,(c, c') \,\&\, DTC^{v}_{t_1\rightarrow u_1}(c') \neq null \right)$$

$$\begin{aligned} TC^{v}_{t_1\rightarrow u_1}(c) : = \max(DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right)| \\ hContxt\,\left(c, c'_i\right) \,\&\, DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right) \neq null, \end{aligned} \tag{4a}$$

$$\begin{aligned} TC^{p}_{t_1\rightarrow u_1}(c) : = \min(DTC^{p}_{t_1\rightarrow u_1}\left(c'_i\right)| \\ hContxt\,\left(c, c'_i\right) \,\&\, DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right) = TC^{v}_{t_1\rightarrow u_1}(c). \end{aligned} \tag{4b}$$

[M3.] Consider direct trust about u_1 in contexts that require a lower competence level than c. Use the minimum value and maximum predictability.

$$Precondition\text{: } \exists c' \left(simLTCX(c, c', \delta) \,\&\, DTC^{v}_{t_1\rightarrow u_1}(c') \neq null \right)$$

$$\begin{aligned} TC^{v}_{t_1\rightarrow u_1}(c) : = \min(DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right)| \\ simLCTX\left(c, c'_i, \delta\right) \,\&\, DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right) \neq null, \end{aligned} \tag{5a}$$

$$\begin{aligned} TC^{p}_{t_1\rightarrow u_1}(c) : = \max(DTC^{p}_{t_1\rightarrow u_1}\left(c'_i\right)| \\ simLCTX\left(c, c'_i, \delta\right) \& DTC^{v}_{t_1\rightarrow u_1}\left(c'_i\right) = TC^{v}_{t_1\rightarrow u_1}(c). \end{aligned} \tag{5b}$$

TABLE 3
Test Integrity Trusting Belief

	$TI^{v}_{t_1\rightarrow u_1} \geq \delta_c$	$TI^{v}_{t_1\rightarrow u_1} < \delta_c$
$TI^{p}_{t_1\rightarrow u_1} \leq \delta_p$	*true*	*false*
$TI^{p}_{t_1\rightarrow u_1} > \delta_p$	*uncertain*	*false*

[M4.] Request u_1's competence reputation in context c.

$$Precondition\text{: } \exists t' \left(DTC^{v}_{t_1\rightarrow u_1}(c) \neq null \right)$$

$$TC^{v}_{t_1\rightarrow u_1}(c) \text{: } = RC^{v}_{u_1}(c), \tag{6a}$$

$$TC^{p}_{t_1\rightarrow u_1}(c) \text{: } = RC^{p}_{u_1}(c), \tag{6b}$$

[M5.] Use the most common belief value about trustees that t_1 encountered in c. Suppose the belief values are in the range of (*a*,*b*). Partition (*a*, *b*) into k (e.g., 10) intervals. Let (*a*′, *b*′) be the interval containing most values. If there are multiple such intervals (called multi-modal situation), use the *uncertainty handling* policies in t_1's profiles to choose one. *mode* $(DTC^{v}_{t_1\rightarrow u}(c))$ is defined as $\frac{a'+b'}{2}$. *mode* $(DTC^{p}_{t_1\rightarrow u}(c))$ is computed in the same way.

$$Precondition\text{: } \exists u' \left(DTC^{v}_{t_1\rightarrow u'}(c) \neq null \right)$$

$$TC^{v}_{t_1\rightarrow u_1}(c) \text{: } = \left\{ mode\left(DTC^{v}_{t_1\rightarrow u}(c)\right) | DTC^{v}_{t_1\rightarrow u}(c) \neq null \right\}, \tag{7a}$$

$$TC^{p}_{t_1\rightarrow u_1}(c) \text{: } = \left\{ mode\left(DTC^{p}_{t_1\rightarrow u}(c)\right) | DTC^{p}_{t_1\rightarrow u}(c) \neq null \right\}. \tag{7b}$$

[M6.] Use the most common belief about all trustees encountered by all trusters in c. This method is considered only if (1) both c and u_1 are new to t_1; and (2) no truster knows u_1. *mode* $(DTC^{v}_{t\rightarrow u}(c))$ and *mode* $(DTC^{p}_{t\rightarrow u}(c))$ are computed in the same way as *mode* $(DTC^{v}_{t_1\rightarrow u}(c))$.

$$Precondition\text{: } \exists u', t' \left(DTC^{v}_{t'\rightarrow u'}(c) \neq null \right)$$

$$TC^{v}_{t_1\rightarrow u_1}(c) \text{: } = \left\{ mode\left(DTC^{v}_{t\rightarrow u}(c)\right) | DTC^{v}_{t\rightarrow u}(c) \neq null \right\}, \tag{8a}$$

$$TC^{p}_{t_1\rightarrow u_1}(c) \text{: } = \left\{ mode\left(DTC^{p}_{t\rightarrow u}(c)\right) | DTC^{p}_{t\rightarrow u}(c) \neq null \right\}. \tag{8b}$$

[M7.] Use priori competence trusting belief specified in t_1's local or global profile (defined in Section 3.3). The priori belief in the local profile overrides the global one.

Four methods that can be used to build integrity belief:

[M8.] Form trusting belief based on direct experience if there are previous interactions.

$$Precondition\text{: } DTI^{v}_{t_1\rightarrow u_1} \neq null$$

$$TI^{v}_{t_1\rightarrow u_1} \text{: } = DTI^{v}_{t_1\rightarrow u_1}, \tag{9a}$$

$$TI^{p}_{t_1\rightarrow u_1} \text{: } = DTI^{p}_{t_1\rightarrow u_1}. \tag{9b}$$

TABLE 4
Candidate Method Set to Build Initial Competence Trust

	c is new	c is known
u_1 is new	$\{M4\} \succ \{M6, M7\}$	$\{M4\} \succ \{M5, M7\}$
u_1 is known	$\{M2, M3, M4\} \succ \{M7\}$	$\{M2, M3, M4\} \succ \{M5, M7\}$

[M9.] Request u_1's integrity reputation.

$$Precondition: \exists t' \left(DTI^{v}_{t' \to u_1} \neq null \right)$$

$$TI^{v}_{t_1 \to u_1} := RI^{v}_{u_1}, \tag{10a}$$

$$TI^{p}_{t_1 \to u_1} := RI^{p}_{u_1}. \tag{10b}$$

[M10.] Use the most common beliefs about trustees that t_1 encountered. *mode* $(DTI^{v}_{t_1 \to u})$ and *mode* $(DTI^{p}_{t_1 \to u})$ are computed in the same way as *mode* $(DTC^{v}_{t_1 \to u}(c))$. This method is always applicable except for the first trustee encountered by t_1.

$$Precondition: \exists u' \left(DTI^{v}_{t_1 \to u'} \neq null \right)$$

$$TI^{v}_{t_1 \to u_1} := \left\{ mode\left(DTI^{v}_{t_1 \to u} \right) | DTI^{v}_{t_1 \to u} \neq null \right\}, \tag{11a}$$

$$TI^{p}_{t_1 \to u_1} := \left\{ mode\left(DTI^{p}_{t_1 \to u} \right) | DTI^{p}_{t_1 \to u} \neq null \right\}. \tag{11b}$$

[M11.] Use priori integrity trusting belief specified in t_1's global profile.

3.2.2 Building and Testing Trusting Beliefs

Different methods are used under various situations for building and testing trusting beliefs. A *candidate method set* includes the methods considered in a specific situation. A method is *applicable* only if: (1) it is in the current candidate method set, and (2) its precondition holds.

Building and testing initial competence trust. There are four scenarios when t_1 is about to establish initial trust about u_1 in c: (1) both c and u_1 are new; (2) c is known but u_1 is new; (3) c is new but u_1 is known; (4) both c and u_1 are known. A context c is known if the truster has experience with some trustee in c. A trustee u_1 is known if she interacted with t_1 before. The candidate method set for all scenarios and the order of their priorities are summarized in Table 4. $\succ$ is a partial order defined on the method priority set. The relationship between two methods enclosed in one "{}" is undefined by the model itself. This is an ambiguous priority set. $\succ$ is extended to a total order according to t_1's *method preference policies*.

The algorithm to build and test an initial competence trusting belief is shown in Fig. 3. The algorithm initializes *unusedMS* using the appropriate candidate method set. It chooses the applicable method M with highest priority in *unusedMS*. The input threshold parameters δ_c and δ_p are compared with the trusting belief generated by M. If "true'" or "false" is obtained, this result is output. Otherwise M is removed, trusting belief is saved and the process is repeated with the next M. In the case that the algorithm outputs no result after all methods are considered, one trusting belief is chosen (i.e., r is chosen among all results) based on imprecision handling policies. The value of the belief is compared with δ_c.

```
Input: $t_1$, $u_1$, c, $\delta_c$, $\delta_p$
Output : true/false
unusedMS := candidate method set defined in Table 4
i := 1
while unusedMS $\neq \emptyset$ {
  M := the applicable method with highest priority
  result[i] := compute($TC^{v}_{t1 \to u1}(c)$, $TC^{p}_{t1 \to u1}(c)$) using M
  testResult := compare result[i] with $\delta_c$, $\delta_p$ based on Table 2
  if (testResult = uncertain){
    i := i + 1;
    delete M from unusedMS
  }
  else{
    return testResult
  }
}
Choose r from {results[i]$\cup$0} based on imprecision handling policy
return (r.value > $\delta_c$)
```

Fig. 3. Algorithm to build/test initial competence trusting belief.

Building initial integrity trust. Truster t_1 uses her priori integrity trusting belief for the first trustee she encountered. If u_1 is not the first trustee, the candidate method set and the order of their priorities are: $\{M9\} \succ \{M10, M11\}$. The algorithm to build and test initial integrity trusting belief is similar to that in Fig. 3.

Building continuous competence trust. The candidate method set and the order of their priorities are: $\{M1\} \succ \{M2, M4\}$. The algorithm to build and test continuous competence trust is similar to that in Fig. 3.

Building continuous integrity trust. The candidate method set and the order of their priorities are: $\{M8\} \succ \{M9, M10\}$. The algorithm to build and test continuous integrity trust is similar to that in Fig. 3.

3.3 Global and Local Profiles

Each truster t_1 has one global profile. The profile contains: (1) t_1's priori integrity and competence trusting belief; (2) method preference policies; (3) imprecision handling policies; (4) uncertainty handling policies; (5) parameters needed by trust-building methods. t_1 can have one local profile for each context. Local profiles have a similar structure as global profiles. The content in a local profile overrides that in the global one. Fig. 4 shows the definition of global and local profiles.

As aforementioned, method preference policies, defined as *PreferencePolicy*, are to extend the partial order $\succ$ to a total order. Therefore, no two methods have the same priority. *iCompetence* and *cCompetence* are used when building initial and continuous competence trust respectively. *iCompetence* consists of four parts corresponding to the four scenarios to build initial competence trust. *iIntegrity* and *cIntegrity* are for establishing integrity trusting belief. Relationships are separately defined on each ambiguous priority set. For

GlobalProfile ::=<truster, Priori, PolicySet, MethodPar>

Local Profile ::=<truster, contextId, Priori, PolicySet, MethodPar>

Priori ::=<IntegrityPriori, CompetencePriori >

IntegrityPriori ::=<value, predictability >

CompetencePriori ::=<value, predictability >

value ::= $R^{[0,1]}$

predictability ::= $R^{[0,1]}$

PolicySet ::=<PreferencePolicy, ImprecisionPolicy, UncertainPolicy >

PreferencePolicy ::=<iCompetence, iIntegrity, cCompetence, cIntegrity >

iCompetence ::=$<< mId^2 >, < mId^2 >, << mId^3 >, < mId^2 >>, << mId^3 >, < mId^2 >>>$

iIntegrity ::=$<mId^2>$

cCompetence ::=$<mId^2>$

cIntegrity ::=$<mId^2>$

mId ::= string

ImprecisionPolicy ::="false" | "priority" | MinPredictability | tValue

MinPredictability ::= "priority" | tValue

tValue ::= "min" | "max" | "median"

UncertainPolicy ::= tValue

MethodPar ::=$< mId, parList^+ >^+$

Fig. 4. Global and local profile definitions.

example, the fourth scenario in building initial competence trust has two ambiguous priority sets {$M2$, $M3$, $M4$} and {*M5, M7*}. Hence, the third part of *iCompetence* is $<<mId^3>$, $<mId^2>>$. Here, mId is the identifier of a method. $< mId^n >$ is the abbreviation of a string $< mId, \dots, mId >$ with n mIds. A method whose mId is in the ith place has the ith highest priority in that set.

Imprecision handling policies are used to choose a belief value when the tests on trusting beliefs generated by all applicable methods return "uncertain". There are three types of policies: If the "false" policy is specified, use "0" as the belief value. This implies that a test failed. If the "priority" policy is adopted, the belief generated by the method with the highest priority is chosen. If "MinPredictability" policies are used, the belief with the lowest predictability is selected. If multiple beliefs have the lowest predictability, they are distinguished by *tValue*, which is a constant specifying whether to choose the minimum, maximum or median belief value. The value of this constant is also set by the imprecision handling policy.

Uncertainty handling policies are used by three belief building methods, $M5$, $M6$, and $M10$, in the multimodal situation. Minimum, maximum or median values can be used based on the policy. They correspond to pessimistic, optimistic, and realistic attitudes as argued by Marsh [13].

MethodPar provides the values to the parameters a method needs. Currently, only the third method needs δ that specifies the proximity threshold between two contexts.

4 Belief Information and Reputation Aggregation Methods

4.1 Competence Belief

Belief about a trustee's competence is context specific. A trustee's competence changes relatively slowly with time. Therefore, competence ratings assigned to her are viewed as samples drawn from a distribution with a steady mean and variance. Competence belief formation is formulated as a parameter estimation problem. Statistic methods are applied on the rating sequence to estimate the steady mean and variance, which are used as the belief value about the trustee's competence and the associated predictability.

Let R denote the competence rating set $R = \{r_1, \dots, r_n\}$ where $r_i \in [0,1].r_1, r_2, \dots, r_n$ are independently drawn from an underlying distribution. The mean and variance of the distribution need to be inferred based on the ratings. Like any statistical inference problem, the inference contains two parts: (1) estimated value (2) a measure of its goodness. Usually, a distribution from a restricted family is chosen to approximate the underlying distribution. We use the Normal distribution $N(\mu, \sigma^2)$, where μ corresponds to the mean and σ^2 corresponds to the variance. μ estimates the trusting belief about the trustee's competence denoted as b_c . σ^2 characterizes the variability of the underlying distribution and is positively correlated with predictability. The goodness of estimation is measured via a confidence interval. 90 percent confidence intervals for μ and σ^2 are constructed. The length of the confidence interval of μ is used as the associated predictability p_c. This method assumes that the observations are drawn from a normal distribution. This assumption may not hold and the result may be misleading. Goodness-of-fit tests can check whether the assumption is valid or not. In summary, the approach is: (1) estimate μ and σ^2; (2) measure the goodness of the inferences; (3) test the normality of the distribution (4) Let $b_c = \mu$ and $p_c =$ length of the confidence interval if the inference is good enough and the data set approximately follows a normal distribution.

k-Statistic defined in (12) is used in computation. Let $\hat{\mu}$ and $\hat{\sigma}$ denote the estimation values of μ and σ. The unbiased estimators of μ and σ^2 are k_1 and k_2.

$$\begin{aligned} S_m &= \sum_{i=1}^{n} (r_i)^m, \\ k_1 &= S_1/n, \\ k_2 &= \left(nS_2 - S_1^2\right)/(n(n-1)). \end{aligned} \tag{12}$$

For μ, if the number of ratings n is greater than 45, the length of the confidence interval can be approximated by (13a). Here, 1.645 is the z value such that 5 percent of the whole area lies to its right in a standard normal distribution. In this equation, σ is replaced by $\hat{\sigma}$. This substitution is applicable only when the size of the rating set is large. In the case there are few ratings, i.e., $\mathrm{n} < 45$, (13a) is not suitable. Fortunately, the underlying population distribution is normal based on the assumption. t-distribution (i.e., student distribution) [17] is used to construct the confidence interval. Equation (13b) computes the length of the confidence interval in this case. In this equation, $\mathrm{t}_{0.05}$(n-1) denotes the critical value of t distribution with (n-1) degrees of freedom such that 5 percent of the area lies to its right:

$$interval\ length\ for\ \mu : = \begin{cases} 2 * 1.645 * \sqrt{\frac{k_2}{n}} & n \geq 45 \qquad (13a), \\ 2 * t_{0.05}(n-1) * \sqrt{\frac{k_2}{n}} & n < 45 \qquad (13b). \end{cases}$$

$\mathrm{n} = 45$ is chosen as the dividing line between large and small rating sets due to two reasons: (1) The critical value of t is always larger than the corresponding z value. t distribution approaches the normal distribution as n increases. (2) The critical value of $\mathrm{t}_{0.05} = 1.6794$ is quite close to $\mathrm{z}_{0.05}$, (i.e., 1.645) when the degree of freedom is 44.

The length of the 90 percent confidence interval for σ^2 is shown in (14a). Here, $\chi^2_{0.05}(n-1)$ is the critical value of x^2 distribution with (n–1) degrees of freedom such that 5 percent of the area lies to the right. $\chi^2_{0.95}(n-1)$ is defined similarly. Unlike z values, χ^2 is asymmetric and $\chi^2_{0.05}(n-1) \neq \chi^2_{0.95}(n-1)$. Equation (14a) is a straightforward application of Fisher's result (i.e., $\chi^2_\alpha(n) \approx \frac{1}{2}(z_\alpha + (\sqrt{2n-1})^2)$) when n is large) for (14b):

$$\begin{aligned} &\mathit{interval\,length\,for}\,\sigma^2 \\ &= \begin{cases} \frac{2(n-1)k_2}{(1.645+\sqrt{2n-1})^2} + \frac{2(n-1)k_2}{(-1.645+\sqrt{2n-1})^2} & n \geq 45 \qquad \text{(14a)}, \\ \frac{(n-1)k_2}{\chi^2_{0.05}(n-1)} - \frac{(n-1)k_2}{\chi^2_{0.95}(n-1)} & n < 45 \qquad \text{(14b)}. \end{cases} \end{aligned}$$

In this model, we assume the existence of a reputation server that acts properly on behalf of trusters. It is assumed that trusters are honest in providing information. The attacks discussed in [4] do not exist. Trusters are subjective and utilize different evaluation criteria. Reputation aggregation methods shall eliminate the effect of subjectivity and output a result close to the trusting belief the reputation requester would have obtained if she had directly interacted with the trustee.

Let t_* denote the truster who requests reputation information about trustee u. Let $t_1, t_2, \ldots, t_k$ denote the trusters who submit a direct competence trusting belief about u to the reputation server. Suppose u follows a distribution with mean μ_* and variance σ^2_* from the perspective of t_*'s. Please note μ_* and σ^2_* are true values, not estimated values from existing ratings. Let μ_i and σ^2_i denote the mean and variance of u from t_i's point of view. Because of the subjectivity, μ_* and σ^2_* are different from μ_i and σ^2_i even when u behaves consistently. Subjectivity between trusters is formulated in (15). Here, $\Delta\mu_i$ and c_i are constants. Equation (15a) is interpreted as "An excellent behavior for Alice is just good for Bob". Equation (15b) can be explained by "the rating range of Bob is greater than that of Alice".

$$\mu_i = \mu_* + \Delta\mu_i, \tag{15a}$$

$$\sigma^2_i = c_i\sigma^2_*. \tag{15b}$$

To eliminate the subjectivity of a truster from the perspective of a requestor is to calibrate such deviations. μ_* and σ^2_* are estimated based on estimated μ_i and σ^2_i, and interaction numbers submitted by k trusters. They are denoted as $\langle\hat{\mu}_1, \hat{\sigma}^2_1, n_1\rangle, \langle\hat{\mu}_2, \hat{\sigma}^2_2, n_2\rangle, \ldots, \langle\hat{\mu}_k, \hat{\sigma}^2_k, n_k\rangle$. From Section 4.1, we know $\hat{\mu}_i$ can be viewed as the value of a random variable with mean $\mu_* + \Delta\mu_i$and variance $\frac{c_i\sigma^2_*}{n_i}$. $\hat{\sigma}^2_i$ is the value of a random variable whose mean is $c_i\sigma^2_*$.

Equation (16) defines an estimator for μ_*. Two estimators for σ^2 are given in (17a) and (17b).

$$\mathit{estimator\ for}\ \mu_* = \frac{\sum_{i=1}^k \hat{\mu}_i}{k} - \frac{\sum_{i=1}^k \Delta\mu_i}{k}, \tag{16}$$

$$\mathit{estimator\ for}\,\sigma^2_* = \frac{\sum_{i=1}^k \frac{(n_i-1)\hat{\sigma}^2_i}{c_i}}{\sum_{i=1}^k (n_i-1)}, \tag{17a}$$

$$\mathit{estimator\ for}\,\sigma^2_* = \sqrt[k]{\prod_{i=1}^k \frac{\hat{\sigma}^2_i}{c_i}}. \tag{17b}$$

Estimation bound. Let X_i denote the random variable for $\hat{\mu}_i$. Let $Y_i = X_i - \Delta\mu_i.Y_1, Y_2, \ldots, Y_k$ are independent. Let M_k denote $(\sum_{i=1}^k Y_i)/k$. Equation (18) gives the mean and variance of Y_i:

$$E[Y_i] = \mu_*, D(Y_i) = c_i\sigma^2_*/n_i. \tag{18}$$

Let $r_{\max}$ denote $max\Big(\sqrt{c_1/n_1}, \sqrt{c_2/n_2}, \ldots, \sqrt{c_k/n_k}\Big)$. According to Liapunov's central limit theorem, when k is large, we have the following result:

$$P\left\{M_k - \frac{r_{max} * 1.645 * \sigma_*}{\sqrt{k}} < \mu_* < M_k + \frac{r_{max} * 1.645 * \sigma_*}{\sqrt{k}}\right\} \geq 0.9. \tag{19}$$

A threshold δ_1 on c_i/n_i is set with (19). i's trusting belief is taken into consideration only if $c_i/n_i \leq \delta_1$. An interval enclose $\mu*$ with at least 90 percent confidence coefficient can be constructed from (19). The intervals corresponding to different conditions are provided in (20a) and (20b):

$$P\left\{M_k - \frac{1.645\sigma_*}{\sqrt{k}} < \mu_* < M_k + \frac{1.645\sigma_*}{\sqrt{k}}\right\} \geq 0.9\ \mathit{if}\ c_i \leq n_i, \tag{20a}$$

$$P\left\{M_k - \frac{3.29\sigma_*}{\sqrt{k}} < \mu_* < M_k + \frac{3.29\sigma_*}{\sqrt{k}}\right\} \geq 0.9\ \mathit{if}\ c_i \leq 4n_i. \tag{20b}$$

Let X_i denote the random variable for $\hat{\sigma}^2_i$. Let $Y_i = ((n_i-1)X_i)/c_i$. Here, n_i and c_i are constants. Let S_k denote$\sum_{i=1}^k Y_i / \sum_{i=1}^k (n_i - 1)$. We have

$$P\left\{\frac{S_k}{\left(\frac{1.645}{\sqrt{2\sum_{i=1}^k(n_i-1)}} + 1\right)^2} < \sigma^2_* < \frac{S_k}{\left(\frac{-1.645}{\sqrt{2\sum_{i=1}^k(n_i-1)}} + 1\right)^2}\right\} = 0.9. \tag{21}$$

Let $r_{\min}$ denote $1.645/\sqrt{2\min(n_1, n_2, \ldots, n_k)}$. We can get a simplified bound from (21):

$$P\left\{\frac{S_k}{\left(\frac{r_{min}}{\sqrt{k-1}} + 1\right)^2} < \sigma^2_* < \frac{S_k}{\left(1 - \frac{r_{min}}{\sqrt{k-1}}\right)^2}\right\} \geq 0.9. \tag{22}$$

If a truster submits her trusting belief only if interaction number is greater than a threshold δ_2, an interval enclose σ^2_*

with at least 90 percent confidence coefficient can be constructed from (22). Particularly, $\delta_2 = 2$ leads to the following bound:

$$P\left\{\frac{S_k}{\left(\frac{1}{\sqrt{k-1}}+1\right)^2} < \sigma_*^2 < \frac{S_k}{\left(1-\frac{1}{\sqrt{k-1}}\right)^2}\right\} \geq 0.9. \tag{23}$$

The aforementioned estimators for μ_* and σ^2 use $\Delta\mu_i$ and c_i that are unknown. Two methods to estimate them are discussed in the rest of this section.

4.2 Estimation of $\Delta\mu_{\mathrm{i}}$ and $\mathrm{c_i}$ Based on Previous Knowledge

Two trusters become acquainted if they share a set of commonly rated trustees. It is assumed that a truster uses the consistent rating criteria for all trustees. $\Delta\mu_i$ and c_i are estimated by comparing the trusting beliefs about trustees known by both t_* and t_i. $\Delta\mu_I$ and c_i are computed using (15a) and (15b). This approach is named as competence reputation evaluation based on knowledge (CRE-K). The prerequisite of CRE-K is that the reputation requester has a set of commonly rated trustees with each of the trusters who provide the trusting beliefs.

Suppose t_* is the truster who requests information. We want to evaluate $\Delta\mu_i$ for truster t_i. Let $\{u_1, u_2, \ldots, u_n\}$ be the trustees about whom both t_* and t_i submit trusting beliefs. $\{\mu_{t_*\to u_1}, \mu_{t_*\to u_2}, \ldots, \mu_{t_*\to u_n}\}$ and $\{\mu_{t_i\to u_1}, \mu_{t_i\to u_2}, \ldots, \mu_{t_i\to u_n}\}$ denote the competence trusting beliefs from t_* and t_i respectively. Plugging $\mu_{t_*\to u_j}$ and $\mu_{t_i\to u_j}$ into (15a) yields n equations:

$$\mu_{t_i\to u_1} = \mu_{t_*\to u_1} + \Delta\mu_i, \ldots, \mu_{t_i\to u_n} = \mu_{t_*\to u_n} + \Delta\mu_i. \tag{24}$$

$\Delta\mu_i$ is the only unknown in above equation array. We find the $\Delta\mu_i$ that minimizes the sum of square errors in (25):

$$\Delta\mu_i = \frac{\sum_{j=1}^{n}(\mu_{t_i\to u_j} - \mu_{t_*\to u_j})}{n}. \tag{25}$$

Similarly, we construct n equations where c_i is the only unknown and find the $\mathrm{c_i}$ minimizing the sum of square errors in (26):

$$c_i = \frac{\sum_{j=1}^{n}\sigma^2_{t_i\to u_j}\sigma^2_{t_*\to u_j}}{\sum_{j=1}^{n}\sigma^4_{t_*\to u_j}}. \tag{26}$$

If this method is used, we will use the first estimator for σ_*^2. Plugging the above results in (16) and (17a) yields:

$$estimator\ for\ \mu_* = \frac{\sum_{i=1}^{k}\hat{\mu}_i}{k} - \frac{\sum_{i=1}^{k}\frac{\sum_{j=1}^{n}(\mu_{t_i\to u_j}-\mu_{t_*\to u_j})}{n}}{k}, \tag{27a}$$

$$estimator\ for\ \sigma_*^2 = \frac{\sum_{i=1}^{k}\frac{(n_i-1)\hat{\sigma}_i^2}{\sum_{j=1}^{n}\sigma^2_{t_i\to u_j}\sigma^2_{t_*\to u_j}/\sum_{j=1}^{n}\sigma^4_{t_*\to u_j}}}{\sum_{i=1}^{k}(n_i-1)}. \tag{27b}$$

The method discussed requires truster t_* to have a lot of acquaintances in the truster set. If this requirement is not satisfied, we can enlarge t_*'s acquaintance set using the idea of delegation. t_* appoints some trusters t_d he knows as delegators and uses $\Delta\mu_{t_d\to t_i}$ and $c_{t_d\to t_i}$ as $\Delta\mu_{t_*\to t_i}$ and $c_{t_*\to t_i}$. A delegator of t_* shall satisfy two constraints: (1) $\Delta\mu_{t_*\to t_d} = 0$, and (2) $c_{t_*\to t_d} = 1$. The method of hypothesis testing is used to check whether t_* shall choose a truster t_i as a delegator.

TABLE 5
Hypothesis Test to Choose a Delegator

	Test statistic	Rejection condition
$k \geq 45$	$z = \frac{\overline{\mu_{diff}}}{s_{\mu_{diff}}/\sqrt{k}}$	$z > z_{0.05}\ or\ z < -z_{0.05}$
$k < 45$	$t = \frac{\overline{\mu_{diff}}}{s_{\mu_{diff}}/\sqrt{k}}$	$t > z_{0.05}(k-1)\ or$ $t < -z_{0.05}(k-1)$

First, we will test the hypothesis related to $\Delta\mu_{t_d\to t_i}$ based on the data set $\{\mu_{t_i\to u_1} - \mu_{t_*\to u_1}, \ldots, \mu_{t_i\to u_k} - \mu_{t_*\to u_k}\}$. Here, $\mathrm{u_1, u_2, \ldots, u_k}$ are the trustees evaluated by both t_* and t_i. The mean and variance of the data set are computed as follows:

$$\overline{\mu_{diff}} = \frac{\sum_{j=1}^{n}(\mu_{t_i\to u_j} - \mu_{t_*\to u_j})}{k}, \tag{28a}$$

$$s_{\mu_{diff}} = \frac{\sum_{j=1}^{k}\left((\mu_{t_i\to u_j} - \mu_{t_*\to u_j}) - \overline{\mu_{diff}}\right)^2}{k-1}. \tag{28b}$$

The following null and alternative hypothesis is used:

$$\begin{cases} H_0: & \Delta\mu_{t_*\to t_d} = 0, \\ H_1: Two-tailed\ test: & \Delta\mu_{t_*\to t_d} \neq 0. \end{cases}$$

The test statistic and rejection condition are summarized in Table 5. If t_i and t_* evaluate a lot of trustees together, we use z value. Otherwise, t value is used.

Second, we will test the hypothesis related to $c_{t_*\to t_d} = 1$ in the same way. The data set is $\{\frac{\sigma^2_{t_i\to u_1}}{\sigma^2_{t_*\to u_1}}, \ldots, \frac{\sigma^2_{t_i\to u_k}}{\sigma^2_{t_*\to u_k}}\}$. Those trusters who pass both tests are selected as delegators that are introduced to broaden the applicability of CRE-K.

4.3 Estimation Based on Priori Assumptions

The second method to estimate $\Delta\mu_i$ and c_i is based on priori assumptions about the distribution of trusters. This method uses the second estimator of σ_*^2, i.e., (17b). Instead of estimating each $\Delta\mu_i$ and c_i, this method estimates $E[(\sum_{i=1}^{k}\Delta\mu_i)/k]$ and $\sqrt[k]{\prod_{i=1}^{k}c_i}$ based on assumptions and uses them to substitute $(\sum_{i=1}^{k}\Delta\mu_i)/k$ and c_i in (16) and (17b). This approach is named as competence reputation evaluation based on assumption (CRE-A).

To estimate $E[(\sum_{i=1}^{k}\Delta\mu_i)/k]$, we assume:

1. $\Delta\mu_i$'s are independently drawn from the same distribution.
2. All states are equally preferable (i.e., the principle of insufficient reasoning).

The first assumption states that $\Delta\mu_1, \Delta\mu_2, \ldots, \Delta\mu_k$ are independent and identically distributed. According to the second assumption, given $\mu_* = \alpha, \Delta\mu_i$ has a uniform distribution in the range $(\alpha, 1-\alpha)$. Therefore, we get the conditional expectation in Equation (29).

$$E\left[\frac{\sum_{i=1}^{k}\Delta\mu_i}{k}\middle|\mu_* = \alpha\right] = 0.5 - \alpha \tag{29}$$

Based on the same assumption, μ_* has the same probability to assume any value in [0, 1]:

$$\begin{aligned} E\left[\frac{\sum_{i=1}^{k}\Delta\mu_i}{k}\right] &= \int_0^1 E\left[\frac{\sum_{i=1}^{k}\Delta\mu_i}{k}\middle|\mu_* = \alpha\right] d\alpha \\ &= \int_0^1 (0.5 - \partial) d\alpha = 0. \end{aligned} \tag{30}$$

To estimate $\sqrt[k]{\prod_{i=1}^{k} c_i}$, we assume:

1. c_i's are independently drawn from the same distribution.
2. It is equally likely that $\sigma_i^2 = \alpha\sigma_*^2$ and $\sigma_i^2 = \sigma_*^2/\alpha$.

Let $\mathrm{Y_k}= \ln(\sqrt[k]{\prod_{i=1}^{k} c_i}) = (\sum_{i=1}^{k}\ln(c_i))/k$. The second assumption states that the probability density function of ln c_i is symmetric and centered at 0, i.e., $E[ln(c_i)] = 0$. According to the strong law of large numbers theorem, we have $P(\lim_{k\to\infty} Y_k = 0) = 1$ i.e., $P(\lim_{k\to\infty}\sqrt[k]{\prod_{i=1}^{k} c_i} = 1) = 1$.

In summary, if this method is used, the estimators are:

$$estimator\ for\ \mu_* = \frac{\sum_{i=1}^{k}\hat{\mu}_i}{k}, \tag{31a}$$

$$estimator\ for\ \sigma_*^2 = \sqrt[k]{\prod_{i=1}^{k}\hat{\sigma}_i^2}. \tag{31b}$$

5 Integrity Belief

Integrity may change fast with time. Furthermore, it possesses a meaningful trend. Evaluation of integrity belief is based on two assumptions:

- We assume integrity of a trustee is consistent in all contexts.
- Integrity belief may vary largely with time. An example is a user behaving well until he reaches a high trust value and then starts committing fraud.

We used mean as an estimator for competence belief as it is relatively steady with time. For integrity belief, this assumption is excluded. When behavior patterns are present, the mean is no more a good estimator. The similarity between a rating sequence and a time series inspires us to adopt the method of double exponential smoothing [7] to predict the next rating based on a previous rating sequence.

Let r_i denote the ith rating and f_{i+1} denote the forecast value of r_{i+1} after observing the rating sequence $r_1, \ldots, r_i$. Equation (32) shows how to compute f_{i+1}:

$$S_i = \alpha(r_{i-2} + r_{i-1} + r_i)/3 + (1-\alpha)S_{i-1} + b_{i-1}, \tag{32a}$$

$$b_i = \beta(S_i - S_{i-1}) + (1-\beta)b_{i-1}, \tag{32b}$$

$$f_{i+1} = S_i + b_i. \tag{32c}$$

In the above equation array, (32a) computes the overall exponential smoothing, (32b) is the trend smoothing, and (32c) is the prediction after observing $r_1, \ldots, r_i$. The reason that we use the average of r_{i-2}, r_{i-1}, r_i is to make the model tolerant to random noise. We use the initial condition defined in (33):

$$S_1 = r_1, b_1 = r_2 - r_1. \tag{33}$$

Equations in (32) may generate results out of the range (0, 1). In this case, we resort to a single exponential smoothing (SES) to bound double exponential smoothing. We call it BDES-S.

There are two parameters in (32), α and β. Both values are in the range (0, 1). Their initial values are set by a truster. α and β are updated after the ratings cumulate. They are determined such that the mean of the squared error (MSE) between the rating sequence and predictions are minimized. Given a rating sequence $\{r_1, r_2, \ldots, r_n\}$ we can determine α and β that minimize the MSE between two sequences using non-linear optimization algorithms such as the Marquardt algorithm [2]. In order to reduce computational complexity, we approximate α and β using a simplified procedure:

- The parameters are updated every time a new rating is added;
- Only the latest sequence with length 4 is used to update the parameters;
- We find the best parameters with a range between 0.1 and 0.9 precise in 1 decimal place.

We find α_i and β_i by minimizing by solving the optimization problem in (34):

$$\begin{aligned} MSE_i = \min\left((S_{i-1} - r_i)^2 + (S_{i-2} - r_{i-1})^2 + (S_{i-3} - r_{i-2})^2\right), \\ \alpha_i, \beta_i\ \varepsilon\ \{0.1, 0.2, \ldots, 0.9\}. \end{aligned} \tag{34}$$

Let t_i denote the trusting belief in integrity after observing i ratings. Let p_i denote the predictability associated with t_i. When $i \geq 2, t_i$ and p_i are defined in (35a)-(35b):

$$t_i = f_{i+1}, \tag{35a}$$

$$p_i = \frac{\sum_{j=1}^{i}\frac{|f_j - r_j|}{r_j}}{i} = \frac{(i-1)p_{i-1} + \frac{|f_j - r_j|}{r_j}}{i}. \tag{35b}$$

The $(i+1)$th prediction computed using (32c) is used as the ith integrity trusting belief. Predictability characterizes the confidence in the belief (i.e., prediction). Mean of squared errors between predictions and ratings are used as p_i. The lower the value of p_i, the higher the confidence is. Like t_i, p_i is computed using a recursive formula.

To evaluate t_i and p_i, the latest four ratings instead of the whole rating sequence are needed. In addition, we have to store two S values (i.e., S_i and S_{i-1}) and two b values (i.e., b_i and b_{i-1}). A time threshold is set by a truster. Any S, b and ratings before that time are discarded.

TABLE 6
Algorithms to Build Integrity Belief

	Equation	Initial condition	Boundary
Average	$t_i = \frac{\sum_{k=1}^{i} r_k}{i}$	t_1=c	
SES	$t_i = \alpha r_i + (1-\alpha)t_{i-1}$ $\alpha \epsilon (0,1)$	t_1=c	
Regret	$t_i = \frac{\sum_{k=1}^{i} w(k,i) r_k}{\sum_{k=1}^{i} w(k,i)}$	t_1=c	
BDES	Equation 32	t_1=c S_1=c b_1=r_2-r_1	$t_i = \alpha \frac{r_{i-2}+r_{i-1}+r_i}{3}$ $+(1-\alpha)t_{i-1}$ $if\ t_i \geq 1\ or\ t_i \leq 0$

Each time a new rating is added, the trusting belief and predictability are reevaluated. The procedure to evaluate t_i and p_i when $\{r_{i-3}, r_{i-2}, r_{i-1}, r_i\}$ are available is outlined as follows:

1. Compute α_i and β_i using (34).
2. Compute S_i, b_i, f_{i+1} using (32).
3. Compute t_i and $p_i$using (35).

If $i < 4$, we ignore the step of updating α_i and β_i and use the initial parameters instead.

The aforementioned approach is extended to evaluate reputation. Let L_t denote the length of a time interval. Integrity trusting beliefs from different trusters are sorted in an ascending order based on time-stamp. The direct evaluation algorithm is applied on this sequence. The prediction generated is the evaluated integrity reputation.

6 Experimental Study of Trust Model

Experimental studies were conducted to evaluate the integrity belief model proposed in Section 5. The objective is to identify the suitable approaches for various scenarios (different types of trustees) and obtain guidelines to determine the appropriate values of parameters for the algorithms. Sections 6.1, 6.2 and 6.3 evaluate the approaches to build integrity belief based on direct experience.

We also conducted experiments to evaluate the competence belief model introduced in Section 4. The CRE-A and CRE-K methods were evaluated under different scenarios, with trustee behavior generated using a normal distribution. Experiments were conducted to compare the true mean and variance with the estimated mean and variance of competence reputation for different number of trusters. The relative error (RE) of CRE-A was found to be around 5 percent, and that of CRE-K was less than 3.5 percent, which are promising results. We omit detailed experiment results due to space constraints.

6.1 Study on Integrity Belief Building Methods

In this section, the BDES algorithm is compared with three other algorithms for five trustee behavior patterns.

Algorithms compared. The algorithms compared are BDES, simple average, single exponential smoothing, and the time-weighted average, called REGRET, proposed in [20]. Let t_i denote the trusting belief after observing rating sequence $r_1, r_2, \ldots, r_i$. Table 6 summarizes how t_i is evaluated under the four algorithms. $w(k, i)$ in REGRET is a time dependent function giving higher values to ratings temporally close to r_i. Table 7 shows the initial values of the parameters of BDES and SES. A function linearly decreasing with $(i - k)$ is used as $w(k, i)$ in REGRET.

TABLE 7
Parameters Used in Experiments

SES	Regret	BDES
$\alpha = 0.3$ (initial value)	w(k,i) is a function linearly decreasing with (i-k)	$\alpha = 0.3$ (initial value) $\beta = 0.7$ (initial value)

Experiment setup. For the experiments discussed in Sections 6.2 and 6.3 below, trustee behavior was simulated using the five different integrity rating generation functions detailed below. A rating for trustee u generated by a behavior pattern function at time i is considered to be the true integrity rating submitted for u by a truster t at time point i. For each behavior pattern experiment, a sequence of 100 ratings for each trustee were generated using the pattern function and the performances of the four integrity belief building methods listed above were evaluated by measuring the difference between the true rating and the rating output by the integrity belief method at each point in the sequence. Note that the identity of the trusters is not relevant in this case: The 100 ratings for a trustee could be submitted by a single truster or by 100 different trusters.

Generate ratings based on trustee behavior patterns. The true values of integrity trusting belief about a trustee can be viewed as the range of a time dependent function $f(i)$. A pattern is a family of $f(i)$s with the same form. It is impossible and unnecessary to enumerate all possible forms of $f(i)$s. We are interested in meaningful patterns revealing the trend and intention of a trustee's behavior. Five types of patterns, random trustee, stable trustee, trend trustee, jumping trustee and two-phase trustee, are identified and used in the experiments. The random pattern shows that the trustee's behavior is variable. Prediction based on previous knowledge may not lead to good results. On the other hand, we can expect to precisely predict the next performance of a trustee with a stable pattern. The trend pattern captures the improving or deteriorating behavior pattern. The jumping and two-phase patterns indicate a sudden shapely change. Usually, they imply misbehaving of trust builders.

TABLE 8
Trustee Behavior Patterns

	Form of f(i)	Figure
Random	$f(i)=U(0,1)$ for $\forall$i	Fig. 5
Stable	$f(i)=c_1$ for $\forall$i	Fig. 6
Trend	$f(i)=c_1+ic_2$ for $\forall$i	Fig. 7
Jumping	$f(i)=c_1$ for $i \leq n_0$ $f(i)=c_2$ otherwise	Fig. 8
Two-phase	$f(i)=c_1$ if $i \leq n_0$ $f(i) = c_1 - \frac{(c_1-c_2)(i-n_0)}{n_1-n_0}$ if $n_0 < i \leq n_1$ $f(i)= c_2$ if $n_1 < i$	Fig. 9

TABLE 9
Instances of Trustee Behavior Patterns

	Definition of f(i)	Figure
Random	$f(i)=U(0,1)$ for $i\epsilon[1,100]$	Fig. 5
Stable	$f(i)=0.6$ for $i\epsilon[1,100]$	Fig. 6
Trend	$f(i)=0.3+0.005i$ for $i\epsilon[1,100]$	Fig. 7
Jumping	$f(i)=0.8$ if $i \leq 50$ $f(i)=0.3$ if $50 < i \leq 100$	Fig. 8
Two-phase	$f(i)=0.8$ if $i \leq 40$ $f(i) = 0.8 - 0.025(i-40)$ if $40 < i \leq 60$ $f(i)= 0.3$ if $60 < i \leq 100$	Fig. 9

Table 8 shows the form of $f(i)$ for each pattern. In this work, the independent variable of $f(i)$ is the number of interactions. c_is and n_is are constants. Based on the behavior patterns, we can systematically evaluate a belief formation algorithm. The effectiveness of an algorithm in an environment is determined by (1) how the algorithm performs for each type of trustee, and (2) what is the distribution of trustees belonging to each?

In this section, we study the first issue. Algorithms are evaluated against the interaction sequences representing different trustee behaviors. Each interaction sequence is generated to reflect certain trustee behavior patterns. A trustee's behavior is determined by her trustworthiness and is influenced by some unpredictable factors. Therefore, the ith rating is generated using (36). The ith rating falls into $[f(i)-0.1, f(i)+0.1]$ with probability 90 percent. The interval is interpreted as the region where relative error is smaller than 10 percent:

$$N\big(f(i), (0.1/1.645)^2\big). \tag{36}$$

6.2 Distribution of Errors

The first set of experiments compares absolute error (AE) and relative error, as defined in (37a) and (37b) respectively, of the four algorithms. We choose this measurement because the purpose of evaluating t_i is to forecast r_{i+1}, i.e., a good trust building algorithm shall output good predictions. Absolute and relative errors characterize how close one prediction is to the true value:

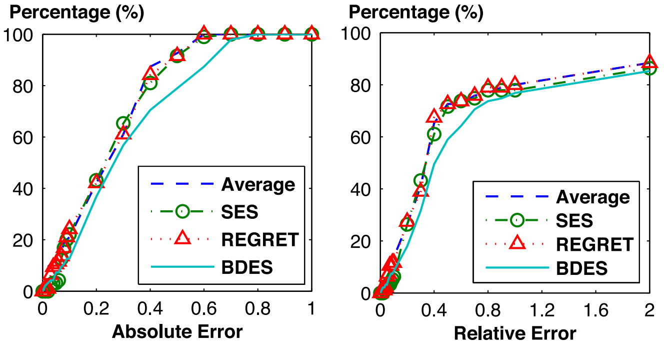
$$AE = |t_i - r_{i+1}|, \tag{37a}$$

$$RE = |t_i - r_{i+1}|/|r_{i+1}|. \tag{37b}$$

We generate 100 ratings for each type of behavior pattern. The parameters are summarized in Table 9. Four algorithms are applied on each trustee. The absolute and relative error for each predication is computed. The distribution of errors generated by each algorithm is plotted using *cumulative frequency figures*.

6.2.1 Results and Observations

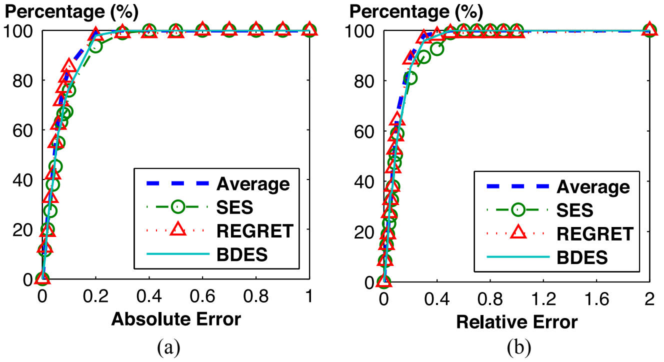
A trustee with random behavior pattern. For a trustee who has the random behavior pattern, the next behavior has no relation to the previous behaviors. The rating can increase or decrease sharply at any time. Because the behavior of the trustee is completely unpredictable, none of the evaluated algorithms is able to provide a good prediction of how the next behavior will be. The Average, SES, and REGRET algorithms have almost the same performance in terms of absolute error, as shown in Fig. 5a. The Average algorithm performs slightly better than the other two. About 88 percent of its results have an absolute error less than 0.4, while the percentages of the SES and REGRET algorithms are 85 and 81 percent respectively. Nearly all results of these three algorithms have an absolute error less than 0.6. The BDES algorithm fails to achieve low error rate in this experiment. Only 70 percent of its results have an absolute error less than 0.4. The upper bound of the error is 0.8 instead of 0.6. Fig. 5b shows that all algorithms generate large relative errors. For the Average, SES, REGRET, and BDES algorithms, the percentages of the results that have a relative error less than 100 percent are respectively, 80, 78, 80, and 77 percent. The percentages of the results that have a relative error greater than 200 percent are 12, 14, 12, and 15 percent respectively. The Average and REGRET algorithms perform the best.



(a) (b)

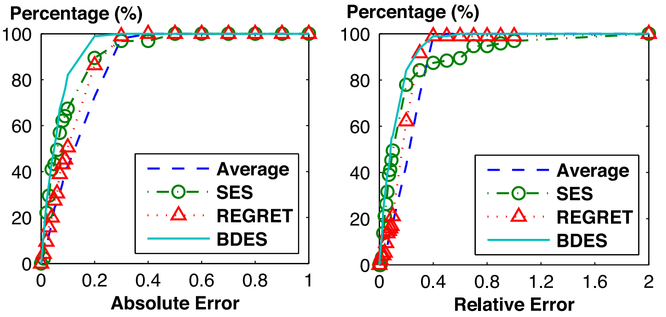
Fig. 5. (a) Absolute error, and (b) Relative error for a trustee with the random behavior pattern.

A trustee with stable behavior pattern. For a trustee who has the stable behavior pattern, the next behavior tends to fluctuate around the mean of the previous behaviors. The rating has a greater probability of being closer to the mean of the previous ratings. All algorithms are able to produce very good results in terms of absolute error and relative error as shown in Figs. 6a and 6b. The REGRET algorithm performs the best, slightly better than the BDES algorithm. For these two algorithms, around 98 percent of the results have an absolute error that is less than 0.2. The corresponding



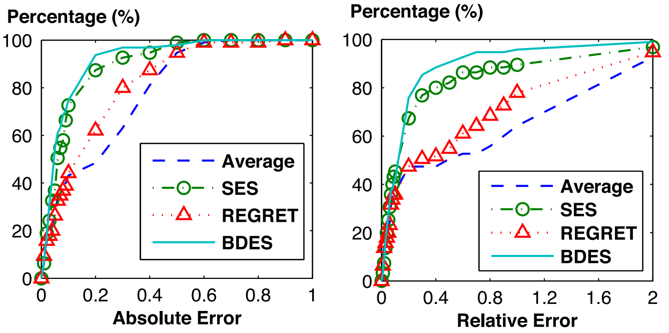
(a) (b)

Fig. 6. (a) Absolute error, and (b) Relative error for a trustee with the stable behavior pattern.



(a) (b)

Fig. 7. (a) Absolute error, and (b) Relative error for a trustee with the trend behavior pattern.



(a) (b)

Fig. 8. (a) Absolute error, and (b) Relative error for a trustee with the jumping behavior pattern.

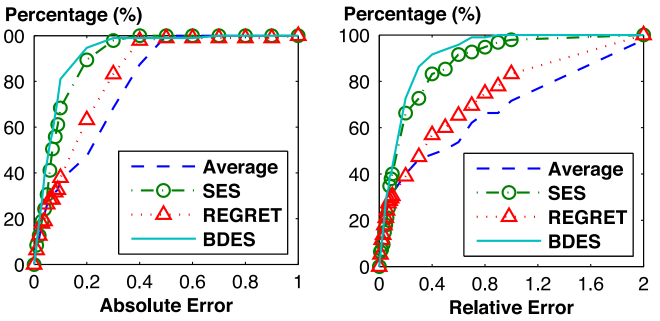
percentage for the SES algorithm is 94 percent. The percentages of the ratings that have less than 0.1 absolute error are 86 percent for the Average and REGRET algorithms, and 78 percent for the SES and BDES algorithms. As shown in Fig. 6b, for every algorithm, almost all results have a relative error less than 60 percent. Ninety percent of the results generated by the Average and REGRET algorithms have a relative error less than 20 percent. The corresponding percentages for the SES and BDES algorithms are 82 and 87 percent respectively.

A trustee with a trend behavior pattern. For a trustee who has the trend behavior pattern, the behavior becomes better and better (or worse and worse depending on the trend) as time passes, i.e., the number of interactions increases. The BDES algorithm outperforms the other algorithms in terms of absolute and relative error when the trustee has a trend behavior pattern.

As shown in Fig. 7a, 88 percent of its results have an absolute error less than 0.2 and 83 percent of its results have an absolute error less than 0.1. The corresponding percentages are 72 and 42 percent for the Average algorithm, 89 and 67 percent for the SES algorithm, and 87 and 50 percent for the REGRET algorithm. As shown in Fig. 7b, although the percentage of the results that have less than 40 percent relative error is around 98 percent for the BDES, Average and REGRET algorithms, only the BDES algorithm is able to make 85 percent of its results having a relative error less than 20 percent. The Average and REGRET algorithms can achieve 42 and 61 percent respectively. 87 percent of the results obtained using the SES algorithm have less than 40 percent relative error, 78 percent of the results have less than 20 percent relative error.

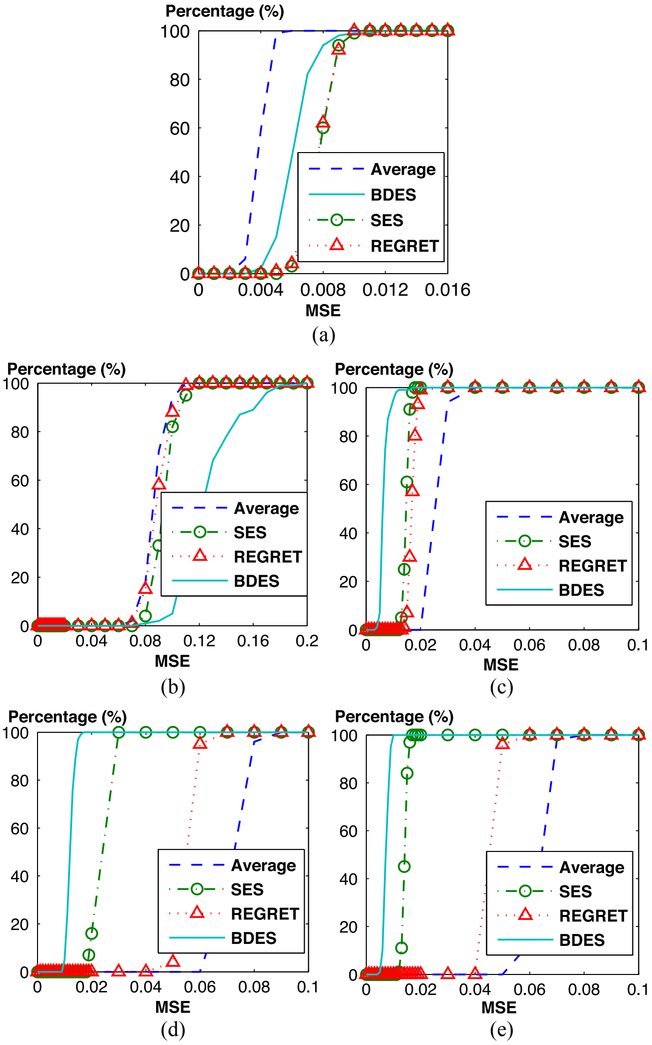
A trustee with jumping behavior pattern. A trustee with the jumping behavior pattern behaves as if he had the stable behavior pattern, and suddenly changes his behaviors. Comparing the results of this experiment with those of the previous two experiments, we can see that the performance downgrades for all, especially the Average and REGRET algorithms. As shown in Fig. 8a, the BDES and SES algorithms still make, respectively, 93 and 88 percent of the results have less than 0.2 absolute error. The corresponding percentage is 48 percent for the Average algorithm and 61 percent for the REGRET algorithm. The upper bound of the absolute error is 0.6 for the BDES and SES, and 0.7 and 0.9 for the Average and REGRET algorithms respectively. Fig. 8b shows that the BDES algorithm has the highest percentage of the results with less than 100 percent relative error, which is 96 percent. For the Average, REGRET, and SES algorithms, the percentages are, respectively, 63, 78 and 90 percent. From another perspective, 90 percent of the results obtained using the BDES algorithm have less than 47 percent relative error. The same percentage of results obtained using the Average, REGRET, and SES algorithms have a relative error less than 190, 170, and 100 percent respectively. The BDES algorithm has the best performance among the evaluated algorithms.

A trustee with two-phase behavior pattern. A trustee who has the two-phase behavior pattern has similar behaviors as compared to the trustee with the jumping behavior pattern, except that he changes his behaviors gradually instead of suddenly. In terms of absolute errors, the BDES and SES algorithms perform a little better, while the Average and REGRET algorithms perform slightly worse as compared to the jumping behavior pattern. As shown in Fig. 9a, the percentages of the results with less than 0.2 absolute errors are 85, 90, 62 and 47 percent for the BDES, SES, REGRET, and Average algorithms, respectively. The percentages of the results with less than 0.1 absolute error are 82, 69, 37, and, 37 percent correspondingly. Fig. 9b shows that all algorithms have a better performance in terms of relative errors compared to what they achieve in the previous experiment. All the results obtained using the BDES algorithm have less than 100 percent relative error. For the SES, REGRET, and Average algorithms, 98, 83, and 71 percent of the results, respectively, have a relative error less than 100 percent. Ninety percent of the results obtained from the BDES, SES,



(a) (b)

Fig. 9. (a) Absolute error, and (b) Relative error for a trustee with the two-phase behavior pattern.



(a)

(b)

(c)

(d)

(e)

Fig. 10. Distribution of mean squared error for (a) Stable, (b) Random, (c) Trend, (d) Jumping, and (e) Two-phase behavior patterns.

REGRET, and Average algorithms have less than 38, 58, 140, and 170 percent relative errors respectively.

6.3 Distribution of Mean Squared Errors

Previous experiments studied the errors generated by a single user per type. The second set of experiments explores the distribution of mean squared errors, as defined in (38). Here, n is the number of predictions:

$$MSE = (t_i - r_i)^2/n. \tag{38}$$

We generate the 100 trustees per behavior pattern using the parameters in Table 9. Please note the trustees are not the same. Four algorithms are applied on each trustee. For each trustee, the MSE generated by each algorithm is computed. For each type of trustee, the distribution of MSE generated by each algorithm is plotted using cumulative frequency figures. Also, average MSE is computed.

6.3.1 Results and Observations

When the trustee has the stable behavior pattern, the Average algorithm outperforms the other algorithms in terms of MSE as shown in Fig. 10a. Its MSEs range from 0.002 to 0.01. Around 99 percent of them are less than 0.005. The REGRET and SES algorithms have almost the same performance, which is worse than that of the BDES algorithm. Ninety percent of the MSEs of the REGRET and SES algorithms are less than 0.009, while the same percentage of MSEs of the BDES algorithm are less than 0.0078. As shown in Fig. 10b the BDES algorithm introduces larger MSE than the other three algorithms when the trustee has the random behavior pattern. The MSEs range from 0.07 to 0.20. Ninety percent of them are less than 0.16. The MSEs of the other three algorithms are very close. All of them are in the range of 0.06 to 0.12. Fig. 10c shows that the BDES performs better than the other algorithms in terms of introducing less MSE when the trustee has the trend behavior pattern. Its smallest MSE is about 0.005. Ninety nine percent of its MSEs are less than 0.012, which is the smallest one among all the MSEs introduced by the other algorithms. The Average algorithm has the worst performance. Its MSEs are in the range of 0.02 to 0.04, 94 percent of them are less than 0.03. The SES algorithm performs slightly better than the REGRET algorithm. Its MSEs range from 0.012 to 0.018, while 99 percent of the MSEs of REGRET are in the range of 0.014 to 0.02.

As shown in Figs. 10d and 10e, when the trustee has the jumping or two-phase behavior pattern, the BDES algorithm has much better performance than the other algorithms. Even its largest MSE is smaller than the smallest one introduced by the other algorithms. For a trustee with the jumping behavior pattern, the ranges of the MSEs are 0.009 to 0.017 for the BDES algorithm, 0.018 to 0.03 for the SES algorithm, 0.04 to 0.07 for the REGRET algorithm, and 0.06 to 0.09 for the Average algorithm. For a trustee with the two-phase behavior pattern, the corresponding ranges are 0.004 to 0.001, 0.012 to 0.017, 0.04 to 0.06, and 0.05 to 0.08, respectively.

TABLE 10
Average MSE for Each Behavior Pattern

	Random	Stable	Trend	Jumping	Two-Phase
Average	0.086069	0.0037669	0.026811	0.072545	0.063554
SES	0.093301	0.007613	0.014744	0.021522	0.014272
REGRET	0.08932	0.0075901	0.016907	0.055423	0.046255
BDES	0.12558	0.0056795	0.0062433	0.012282	0.0074293

Table 10 shows the average MSE for each behavior pattern obtained in the experiment. For a completely unpredictable trustee, i.e., the one with random behavior, no algorithm is able to provide practically useful integrity trusting belief. For a stable trustee, all algorithms can provide satisfactory information, with the Average algorithm being the best. When a trustee has the trend to change his behavior, e.g., the trend, jumping, and two-phase behavior pattern, only the BDES algorithm is able to catch this trend. The accuracy of the integrity trusting belief computed using the BDES algorithm is not affected much by the change of behavior, as seen in the last row.

7 Conclusion

In this paper we presented a dynamic computational trust model for user authorization. This model is rooted in findings from social science, and is not limited to trusting belief as most computational methods are. We presented a representation of context and functions that relate different contexts, enabling building of trusting belief using cross-context information.

The proposed dynamic trust model enables automated trust management that mimics trusting behaviors in society, such as selecting a corporate partner, forming a coalition, or choosing negotiation protocols or strategies in e-commerce. The formalization of trust helps in designing algorithms to choose reliable resources in peer-to-peer systems, developing secure protocols for ad hoc networks and detecting deceptive agents in a virtual community. Experiments in a simulated trust environment show that the proposed integrity trust model performs better than other major trust models in predicting the behavior of users whose actions change based on certain patterns over time.

References

[1] G.R. Barnes and P.B. Cerrito, "A Mathematical Model for Interpersonal Relationships in Social Networks," *Social Networks*, vol. 20, no. 2, pp. 179-196, 1998.

[2] R. Brent, *Algorithms for Minimization without Derivatives.* Prentice-Hall, 1973.

[3] A. Das and M.M. Islam, "SecuredTrust: A Dynamic Trust Computation Model for Secured Communication in Multiagent Systems," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 2, pp. 261-274, Mar./Apr. 2012.

[4] C. Dellarocas, "Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior," *Proc. Second ACM Conf. Electronic Commerce*, pp. 150-157, 2000.

[5] L. Fan, "A Grid Authorization Mechanism with Dynamic Role Based on Trust Model," *J. Computational Information Systems*, vol. 8, no. 12, pp. 5077-5084, 2012.

[6] T. Grandison and M. Sloman, "A Survey of Trust in Internet Applications," *IEEE Comm. Surveys*, vol. 3, no. 4, pp. 2-16, Fourth Quarter 2000.

[7] J.D. Hamilton, *Time Series Analysis.* Princeton University Press, 1994.

[8] J. Hu, Q. Wu, and B. Zhou, "FCTrust: A Robust and Efficient Feedback Credibility-Based Distributed P2P Trust Model," *Proc. IEEE Ninth Int'l Conf. Young Computer Scientists (ICYCS '08)*, pp. 1963-1968, 2008.

[9] B. Lang, "A Computational Trust Model for Access Control in P2P," *Science China Information Sciences*, vol. 53, no. 5, pp. 896-910, May 2010.

[10] C. Liu and L. Liu, "A Trust Evaluation Model for Dynamic Authorization," *Proc. Int'l Conf. Computational Intelligence and Software Eng. (CiSE)*, pp. 1-4, 2010.

[11] X. Long and J. Joshi, "BaRMS: A Bayesian Reputation Management Approach for P2P Systems," *J. Information & Knowledge Management*, vol. 10, no. 3, pp. 341-349, 2011.

[12] S. Ma and J. He, "A Multi-Dimension Dynamic Trust Evaluation Model Based on GA," *Proc. Second Int'l Workshop Intelligent Systems and Applications*, pp. 1-4, 2010.

[13] S. Marsh, "Formalizing Trust as a Concept," PhD dissertation-Dept. of Computer Science and Math., Univ. of Stirling, 1994.

[14] P. Matt, M. Morge, and F. Toni, "Combining Statistics and Arguments to Compute Trust," *Proc. Ninth Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '10)*, pp. 209-216, 2010.

[15] D. McKnight, V. Choudhury, and C. Kacmar, "Developing and Validating Trust Measures for E-Commerce: An Integrative Topology," *Information Systems Research*, vol. 13, no. 3, pp. 334-359, Sept. 2002.

[16] D. McKnight and N.L. Chervany, "Conceptualizing Trust: A Typology and E-Commerce Customer Relationship Model," *Proc. 34th Ann. Hawaii Int'l Conf. System Sciences (HICSS '01)*, 2001.

[17] W. Mendenhall and R.J. Beaver, *Introduction to Probability and Statistics*. PWS-Kent Publishing Company, 1991.

[18] A. Nagarajan and V. Varadharajan, "Dynamic Trust Enhanced Security Model for Trusted Platform Based Services," *Future Generation Computer Systems*, vol. 27, pp. 564-573, 2011.

[19] J.M. Pujol, R. Sangesa, and J. Delgado, "Extracting Reputation in Multi Agent Systems by Means of Social Network Topology," *Proc. Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '02)*, pp. 467-474, 2002.

[20] J. Sabater and C. Sierra, "Social ReGreT, a Reputation Model Based on Social Relations," *ACM SIGecom Exchanges*, vol. 3, no. 1, pp. 44-56, 2002.

[21] F. Skopik, D. Schall, and S. Dustdar, "Modeling and Mining of Dynamic Trust in Complex Service-Oriented Systems," *Information Systems*, vol. 35, pp. 735-757, 2010.

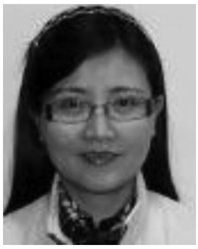
[22] F.E. Walter, S. Battiston, and F. Schweitzer, "Personalized and Dynamic Trust in Social Networks," *Proc. ACM Conf. Recommender Systems (RecSys '09)*, pp. 197-204, 2009.

[23] X. Wang and L. Wang, "P2P Recommendation Trust Model," *Proc. IEEE Eighth Int'l Conf. Intelligent Systems Design and Applications (ISDA '08)*, pp. 591-595, 2008.

[24] B. Yu and M.P. Singh, "An Evidential Model of Distributed Reputation Management," *Proc. Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '02)*, pp. 294-301, 2002.

[25] Y. Zhang, S. Chen, and G. Yang, "SFTrust: A Double Trust Metric Based Trust Model in Unstructured P2P Systems," *Proc. IEEE Int'l Symp. Parallel and Distributed Processing (ISPDP '09)*, pp. 1-7, 2009.

[26] X. Zhu, "Dynamic Authorization of Grid based on Trust Mechanism," *Proc. Int'l Symp. Intelligence Information Processing and Trusted Computing (IPTC)*, pp. 417-421, 2010.



Yuhui Zhong received the PhD degree in computer science from Purdue University in 2005. She is a senior engineer at Microsoft. Her research interests are in digital rights management and trusted computing.

Bharat Bhargava received the PhD degree in electrical engineering at Purdue University in 1974. He is a professor of computer science at Purdue University. His research focuses on distributed systems security/privacy.

Yi Lu received the PhD degree in computer science from Purdue University in 2004. He is a principal engineer at Microsoft. His research interests are in real-time media communication and wireless networking.

Pelin Angin received the PhD degree in computer science from Purdue University in 2013. She is a visiting researcher at Purdue University. Her current research focuses on secure mobile-cloud computing.