ISSN : 0974 - 7435 Volume 8 Issue 4

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 8(4), 2013 [459-463]

# Study of the trust store module

**Jiang He**
**Department of Mathematics, Henan Institute of Science and Technology, Xinxiang, 453003, (R. CHINA)**

## ABSTRACT

In this paper, the presence of the TS enabled a user to manage the behavior histories of its associates. The TS (Trust Store) also provided a source for the credential exchange process called introduction. While the TS was closely linked to the formulation of reputations and associations, the results of TS's work had to be compared against the system's risk assessment before reputations could yield a trust decision.


## KEYWORDS

Trust store;
Trust profiles;
Trust tendencies.

## INTRODUCTION

Trust, and more importantly decisions on trustworthiness, is omnipresent in life[1]. Luhmann's sociological approach[2] considered trust as "a means for reducing the complexity in society." This complexity was created as individuals interacted using their own perceptions, motivations, and goals. Solomon and Flores[3] contended that "trust forms the foundation, or the dynamic precondition, for any free enterprise society." They pointed out that what constituted freedom was the right to make promises and, more importantly, the responsibility for fulfilling them. Trust, therefore, was the basic underpinning of a cooperative environment. Trust was not an inherited trait but was learned as a member of the environment interacted with others. Another applicable definition of trust was provided by Gambetta[4]: "…trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action."

## TRUST PROFILES

A trust profile provided an easy way to quantify a user's initial trust expectations. Trust thresholds, which represented the amount risk a user was prepared to accept when forming or maintaining associations, were one example of information that was contained in the trust profile (Prietula 2000). Constraints on the risk assessment and reputation-scaling algorithms were also included in the profile.

Users established how trusting they could afford to be based on their individual goals and objectives. They selected the trust profile that most closely aligned with these individual needs, establishing their initial state before entering a collaborative environment. In a general sense, users were grouped into four trust profiles after Prietula's work[5]. The general tendencies are illustrated in Figure 1 and explained below.

Altruistic users did not interpret the behavior of their peers. Instead, they performed services for the network

FULL PAPER

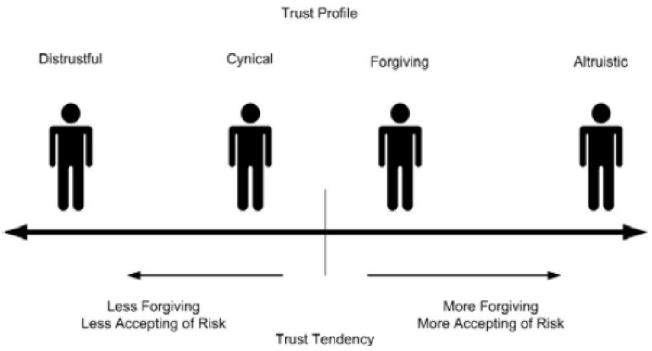


**Figure 1 : the Relationship of Trust Profiles to Trust Tendencies**

for their own benefit. Altruistic users trusted any peer and did not maintain reputation indices. Because of this constraint, peers could not receive referrals from Altruistic users, as they had no evidence to share.

Forgiving users exhibited responses to peer behavior by adjusting the reputation of their peers in a positive or negative manner based on direct experiences or the observations of other peers. Forgiving users aggregated peer reputations in such a way as to allow a misbehaving peer to redeem or rehabilitate its reputation given a sustained period of positive or desirable behavior. Forgiving users established trust thresholds and evaluated peer reputations against these thresholds to determine a peer's trustworthiness.

Cynical users were similar to Forgiving users in that they exhibited responses to peer behavior and maintained reputations of the peers who they interacted with. Importantly, however, Cynical users did not allow trust rehabilitation. Once a Cynical user determined that a misbehaving peer was untrustworthy, he made no attempt to collect further behavior information or readjust the peer's reputation.

Distrusting users did not trust any other user. These users typically relied on MAC or RBAC to form associations rather than trust. Because of this requirement for pre-configuration and run-time verification, a distrusting user would be unable to operate in an ad-hoc environment, except perhaps through contact with altruistic users.

An example of the initialization settings for each trust profile is given in TABLE 1. These values were determined through analysis and testing to offer gradient levels of protection when the user joined the network. These values, like those published by other researchers[6], provided a starting point or basic level of assurance. The TMS, as a self-protecting system, would begin adjusting these levels to current network or environment conditions immediately. A constraint listed as "None" indicates that the risk or reputation values are allowed to rise and fall based on periodically updated information. The Cynical user's "Rising Only" constraint reflects the profile's inclination to raise its thresholds without ever relaxing them, regardless of the current information.

**TABLE 1 : Example Initialization Values Based on Trust Profiles**

| Trust Profile | Trust Threshold | Distrust Threshold | Risk Constraint | Reputation Constraint |
|---|---|---|---|---|
| Altruistic | -0.99 | -1 | None | None |
| Forgiving | 0.6 | 0.3 | None | None |
| Cynical | 0.7 | 0.4 | Rising Only | Rising Only |
| Distrusting | NA | NA | None | None |

This research focused on what was considered the worst-case user – the Forgiving trust profile. In this profile, both reputation and risk were allowed to increase and decrease based on current information. This feature made the user accept risk and maintain the trustworthiness state of their peers. The mechanisms and effectiveness of the TMS in performing these tasks are described in the following sections.

## THE TRUST STORE MODULE

The TS was the TMS's memory. It held the behavior history of associates, providing a virtual space for maintaining and storing the behavior history of nodes that the user had interacted with. The TS was also responsible for the very important recognition process, implemented as a memory management algorithm.

The remainder of this section discusses the goals and objectives of the TS. The format of the Atomic Behavior Record (ABR) is presented, followed by an explanation of the memory management algorithm, the recognition process and, finally, the introduction procedure.

Where Interpersonal trust was dependent upon peer behavior trends and System trust was determined through an evaluation of system behavior tendencies, Situational trust was independent of the behavior of other users altogether. This type of trust used the trust store,

representing the user's memory of previous peers and situations, to determine what action it would take. A situational trust decision was predicated on remembering a previous decision that had yielded a positive outcome, regardless of the behavior of peers that may or may not have been involved.

The amount of memory had a direct impact on which associates could be remembered and how much information could be maintained on each associate. In theory, a system preferred to keep everything possible. In reality, however, there were two practical limitations on the amount of memory that was possible or practicable to devote to maintaining trust information.

The first limitation was that a user's node has a finite memory. Any memory consumed by the trust system was unavailable to other processes required by the node. The TMS required storage space to hold trust information. It also consumed communications capability by sending and receiving trust information. Since the trust system operated in the background, it was desirable for this system to occupy as little space as possible in the node's memory.

The second limitation was that trust information had a finite time duration during which the reports and observations were applicable. Users were expected to change their behavior, so keeping older information yielded diminishing returns. Because the system could not assume a shared time source, each user had to have a mechanism to eliminate old information. For the behavior grades, this mechanism was designed as part of the 3Win algorithm. For the identities and behavior information for past and present associates, the Trust Store managed a queue of records, called ABRs.

The TS created ABRs for associates as a result of the introduction process. Once the ABR was established, the TS accepted reports and observations as inputs, finding the appropriate ABR and storing the information. The TS provided the ABR to the RSM when requested, applying the memory management algorithm as it restored the ABR to the store.

The ABR was used as the trust credential within the network. Users carried their own ABRs, as well as those of their associates, as a means of eliminating the problem of credential discovery in distributed networks[7].

The TS contained the identities and the Reputation Indexing Windows (RIWs) for each of a node's TPs. The RIWs consist of collections of FIs that were organized by subject node and context. The TS also carried some *recognition evidence* for past TPs. The TMS needed to remember good and bad experiences from the past so that it would not waste time or place the user in danger while getting re-acquainted with former (possibly malicious) peers. Given this requirement, a node kept the identities and RIWs concerning former peers; those that have either moved out of range while on good terms or whose association was dissolved as the result of a complaint.

The permanent section of the ABR established the link to the user's identity certificate, shown in Figure 2. The identity section was derived from the Distinguished Name contained in X.509 v3-type certificates, issued by the KMS. Although it was expected that a user entered and left ad-hoc networks based on their own goals and objectives, their identity would not change. Because of the permanence of the user identity, the TMS could track a user's behavior over a period of time. Recording and storing a user's behavior allowed other users to analyze past performance as a means to predict future behavior[8].

The ABR's temporary section contained the evidence of the user's past behavior. Every time a user participated in a transaction (e.g., buying, selling, ex-

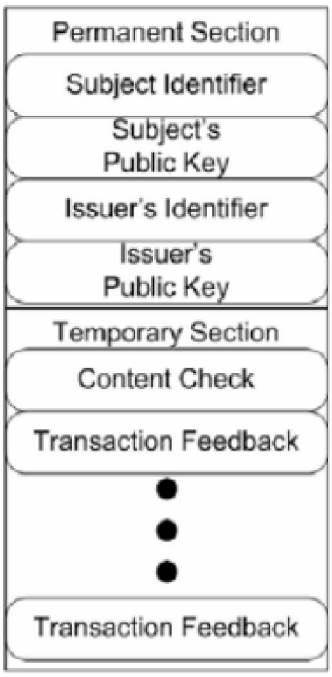


**Figure 2 : Format of an Atomic Behavior Record**

FULL PAPER

changing files), they received feedback on their performance. Feedback was a common feature of ecommerce sites and provided observations on a user's behavior that were used by peers to judge trustworthiness before engaging in a transaction. The number of transaction records stored within the temporary section varied with the user's activity within the network, as the network's reputation calculating mechanism read these records and determined the user's trustworthiness. Because the reputation calculating mechanism only read a certain number of transactions, there was no utility in keeping transactions above and beyond the required amount but there was no system-wide policy against doing so. Nodes discarded the unnecessary records, preferring to economize memory consumption through properly formatted ABRs.

The TS was more than a storage location. Information stored in the trust store was used in three procedures: recognition, introduction, and context tuning. Certificates in the trust store facilitated *recognition* by remembering former peers and reactivating their reputations based on verification of their identities. When a node came into contact with another node, it checked its TS to see if it had ever had an association with the new node. Because the node maintained a memory, it made a decision on whether or not to pursue an association with the new node based on information in the TS[9]. This action was a form of passive recognition; passive because another process triggered the recognition procedure. Nodes also had the ability to "self-trigger" the recognition process, as in the SECURE project. In other words, a node might actively search for nodes that it *recognized* by monitoring communications and comparing the identities of communicators against its TS.

The TS was also crucial in the *introduction* procedure, a critical element in the integration of new nodes into the network and is shown in Figure 3. This introduction included a mechanism for the two nodes to share observed behavior history in such a way that they could derive the reputation of their prospective partner by having the proof to substantiate the given value. As a result, the TMS kept a certain number of its behavior observations so that it could provide non-reputable, references to other nodes when requested. A node confirmed the new peer's identity by verifying its certificate. It then filled its reputation windows with the FI received from its TPs and calculated the new associate's reputation. Finally, the node decided to establish an association with the new peer by comparing the new peer's RI against its trust thresholds.

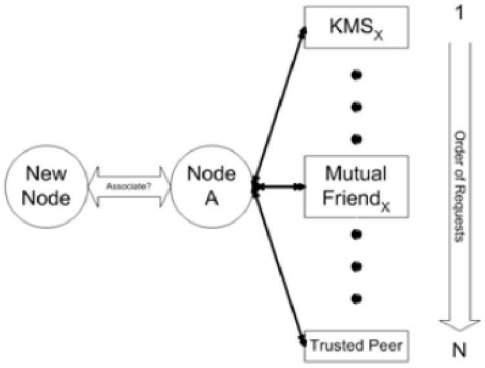


**Figure 3 : Introduction Process**

A user took advantage of the contents of the ABR in the introduction process by soliciting a prospective associate for their ABR before initiating the transaction. The introduction process was iterative and halted whenever the user was satisfied of the peer's identity and had accumulated enough FI to fill their RIW. A user (e.g., Alice) began the introduction process, illustrated in Figure 3, by attempting to contact the peer's (e.g., Bob's) home domain to verify his identity. If Bob's home domain was not available, usually for reasons of network partitioning, Alice examined his ABR and attempted to find mutual TPs. These TPs were users that both Alice and Bob had chosen to trust in the past. If there are no mutual TPs or none could be reached, Alice had to make a decision based on the evidence within Bob's ABR on whether or not to trust him. It should be noted that relying on Bob to vouch for himself was the least desirable option and Alice would involve the risk assessment and situational trust elements of the trust management system before taking this decision.

The benefit of the introduction process was that Alice could be assured of Bob's identity and behavior through communication with trusted parties. In other words, she asked someone that she had already chosen to trust to recommend or refer Bob as a trustworthy user. Once someone verified that Bob was who he says he was, Alice used the transaction records from Bob's ABR to calculate his reputation. If his reputation was above her trust threshold, Alice trusted him; other-

wise she recorded the incident and moved on in search of a more trustworthy partner.

The costs of the introductory process were in the time it took Alice to attempt to contact reputable users to provide references for Bob. Once Bob had been vouched for, Alice evaluated the contents of his ABR and calculated his reputation. In all, this was a necessary process but was one that should be performed as few times as possible.

Operating the TS encompassed more than just managing memory space. Identities and RIWs needed to be stored and organized for efficient searching. Most importantly, the TS needed to be protected against the possibility of compromise or unauthorized access during an operation. Protection was built into the design of the certificates and feedback items themselves, as they were digitally signed. Unauthorized insertion or deletion of certificates and associated RIW would cause a node to have to "re-introduce" itself unnecessarily. This caused undue network traffic and forced a node to wait while the re-introduction took place.

As a node collected identities, reports, and behavior feedback items, these items were stored in ABRs and placed in the TS. Based upon the assumption that the amount of memory that a node could apportion to the trust store was bounded in some way, this store had to be managed efficiently[10]. When the bound was reached, the node selectively eliminated or "forgot" items to make space for new objects. Efficiency, in this case, was defined as limiting the number of "re-introductions" that a node required. Re-introductions were cases where a node forgot a peer that it once had security associations with and had to go through the entire introduction process, as if the two nodes had never dealt with each other.

The presence of the TS enabled a user to manage the behavior histories of its associates. The TS also provided a source for the credential exchange process called introduction. While the TS was closely linked to the formulation of reputations and associations, the results of TS's work had to be compared against the system's risk assessment before reputations could yield a trust decision.

## REFERENCES


**[1]** S.Marsh; Formalising Trust as a Computational Concept, Ph.D. Dissertation, Department of Mathematics and Computer Science, University of Stirling, **(1994)**

**[2]** N.Luhmann; Trust and Power.Wiley, **(1979)**.

**[3]** R.Solomon, F.Flores; Building Trust. New York, NY, Oxford University Press, **(2001)**.

**[4]** D.Gambetta; Can we trust? Trust, Making and breaking cooperative relations, electronic edition. D.Gambetta, Univ. of Oxford: Ch 13, 213-237 **(1988)**.

**[5]** M.Prietula, K.Carley; Boundedly rational and emotional agents-cooperation trust and rumor. Trust and Deception in Virtual Societies. C.Castelfranchi and Y.H.Tan, M.A.Norwood, Kluwer Academic Publisher, 169-193 **(2001)**.

**[6]** M.Michalakopoulos, M.Fasli; On Deciding to Trust. Proceedings of the Third International Conference on Trust Management (iTrust2005), Paris, FR, 23-26 May 2005, 61-76 **(2005)**.

**[7]** P.Dewan, P.Dasgupta; Securing Reputation Data in Peer-to-Peer Networks. Proceedings of the International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), Cambridge, MA, 1-10 **(2004)**.

**[8]** W.J.Adams, G.C.Hadjichristofi, et al.; Calculating a Node's Reputation in a Mobile Ad-Hoc Network. Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC 2005), Phoenix, AZ, 6-9 April, 303-307 **(2005)**.

**[9]** J.-M.Seigneur, S.Farrell, et al.; End-to-end Trust Starts with Recognition. Proceedings of the First International Conference on Security in Pervasive Computing, Dallas, TX, 23-26 March, 130-142 **(2003)**.

**[10]** E.Gray, J.-M.Seigneur, et al.; Trust propagation in small worlds. Proceedings of the First International Conference on Trust Management (iTrust2003), Heraklion, Crete, 239-254 **(2003)**.