ISR Institute for Software Research
University of California, Irvine

TREF: A Threat-centric Comparison Framework for Decentralized Reputation Models

Girish Suryanarayana
University of California, Irvine
sgirish@ics.uci.edu

Richard N. Taylor
University of California, Irvine
taylor@uci.edu

January 2006

ISR Technical Report # UCI-ISR-06-2

Institute for Software Research
ICS2 110
University of California, Irvine
Irvine, CA 92697-3455
www.isr.uci.edu

http://www.isr.uci.edu/tech-reports.html

TREF: A Threat-Centric Comparison Framework for Decentralized Reputation Models

Girish Suryanarayana, Richard N. Taylor
Institute for Software Research
University of California, Irvine
{sgirish,taylor}@ics.uci.edu

ISR Technical Report # UCI-ISR-06-2

January 2006

Abstract: In a decentralized system, entities, also known as peers, directly interact with each other and make local autonomous decisions towards their individual goals. In an open decentralized system, there is no single centralized authority that can regulate the entry of peers in the system. As a result, the system may contain malicious peers that try to disrupt the system and carry out attacks on other peers. In the absence of a centralized authority that can help guard against such attacks, each peer must incorporate suitable measures to protect itself from such attacks. Trust management mechanisms serve to provide effective countermeasures against the attacks perpetrated by malicious peers. Reputation-based trust models allow peers to determine the trustworthiness of other peers in the system based on their perceived reputations. While a number of decentralized reputation-based trust models exist in the research literature, little effort has been directed towards their systematic evaluation and comparison. In this paper, we present TREF, a threat-centric framework for evaluating and comparing different reputation-based trust models as an initial step towards addressing this need. We also discuss how we validated the TREF framework in the context of four reputation-based trust models. Our evaluation reveals several key benefits of using the TREF framework.

TREF: A Threat-centric Comparison Framework for Decentralized Reputation Models

Girish Suryanarayana, Richard N. Taylor

Institute for Software Research
University of California, Irvine
{sgirish,taylor}@ics.uci.edu

ISR Technical Report # UCI-ISR-06-2

January 2006

Abstract

In a decentralized system, entities, also known as peers, directly interact with each other and make local autonomous decisions towards their individual goals. In an open decentralized system, there is no single centralized authority that can regulate the entry of peers in the system. As a result, the system may contain malicious peers that try to disrupt the system and carry out attacks on other peers. In the absence of a centralized authority that can help guard against such attacks, each peer must incorporate suitable measures to protect itself from such attacks. Trust management mechanisms serve to provide effective countermeasures against the attacks perpetrated by malicious peers. Reputation-based trust models allow peers to determine the trustworthiness of other peers in the system based on their perceived reputations. While a number of decentralized reputation-based trust models exist in the research literature, little effort has been directed towards their systematic evaluation and comparison. In this paper, we present TREF, a threat-centric framework for evaluating and comparing different reputation-based trust models as an initial step towards addressing this need. We also discuss how we validated the TREF framework in the context of four reputation-based trust models. Our evaluation reveals several key benefits of using the TREF framework.

1. Introduction

Consider a gnutella-based[28] open decentralized file-sharing application. In such an application, a peer directly queries its neighboring peers for a file. These neighboring peers in turn forward the query to their neighbors. Peers with matching files respond to the query. The original peer selects a peer from the list of all responding peers and directly downloads the file from that peer. In an open file-sharing application, peers can enter and leave the system at any time. Thus malicious peers can be present in the system who disguise viruses, trojans and fake files as reliable files. These files pose a significant risk to the downloading peer as well as to all those peers who may unknowingly receive the files from that peer. Using a centralized authority that can maintain information about all peers in the system, regulate the entry of peers in the system, and coordinate file-sharing between peers helps considerably alleviate the threat due to malicious peers. However, in the absence of such a centralized authority, each peer in the decentralized system must adopt suitable measures to safeguard itself.

The threat posed by malicious peers is not limited only to decentralized file-sharing applications. Other decentralized applications such as decentralized auctioning and emergency response are equally susceptible to these kinds of threats. Trust management has been found to serve as a potential countermeasure for addressing such threats. Trust relationships between peers helps a peer determine the extent of trustworthiness of other peers. This trust information can be used to make well-informed decisions about interaction with a peer.

Trust management has therefore received a lot of attention from researchers. Reputation-based trust management systems rely on the past behavior of peers in order to draw conclusions about their trustworthiness. Several reputation-based trust models exist in the research literature[44]. Different models are geared towards different objectives and different applications; however current literature is lacking an evaluation framework that can help contrast the capabilities of different models as well as guide the selection of a suitable model for a given setting.

This paper describes our efforts towards addressing this need and presents a threat-centric evaluation framework, called TREF (Threat-centric REputation Framework), for

decentralized reputation-based trust models. Trust models are primarily geared at safeguarding the system from the threats and attacks of malicious peer. These threats and attacks, thus, naturally provide an excellent starting ground to compare and evaluate these models. We, therefore, use them as the basis for the TREF framework. We have validated four sample reputation model evaluations based on the TREF framework by comparing them against actual results observed when those model implementations are subjected to various threat scenarios. Our experiments have revealed the feasibility and soundness of the TREF framework in evaluating and comparing decentralized reputation-based trust models. We believe that the threat-centric approach used in the TREF framework can be used as the basis for designing secure and capable reputation models in the future. We also believe that this threat-centric approach can serve as a fertile ground for future in depth comparisons of reputation models.

The rest of the paper is organized as follows. Section 2 provides background on trust and reputation and includes relevant related work. Section 3 presents our definition of a trust model and discusses the functional elements of a trust model. Section 4 classifies protective mechanisms employed by reputation models while section 5 introduces the critical threats of decentralization that form the basis of our threat-centric framework. Section 6 discusses the design of our threat-centric framework and section 7 describes our efforts towards validating the our framework. The paper ends with a discussion in section 8.

2. Background

In this section, we define as well as introduce the basic concepts of trust and reputation management. We then present related work in trust and reputation management.

2.1. Trust

The concept of trust is an integral part of man's social existence. Interactions in society are influenced by the perceived trust worthiness of others. Trust thus plays a significant role in our day-to-day life. Naturally, researchers from several disciplines including sociology, history, economics, computer science, and philosophy have investigated the issue of trust [37]. Given the fact that trust is a multi-disciplinary concept, there exist in the research literature several definitions of trust and discussions about the factors that determine trust. We limit our discussion here to some well-known definitions with an aim to provide a sufficient background for the purpose of this paper.

One of the most well-known definitions of trust is the one coined by Deutsch [12] which states that:

(a) an individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial or to an event perceived to be harmful; (b) he perceives that the occurrence of these events is contingent on the behavior of another person; and (c) he perceives the strength of a harmful event to be greater than the strength of a beneficial event. If he chooses to take an ambiguous path with such properties, he makes a trusting choice; else he makes a distrustful choice.

An interesting fact about the above definition pointed out by Marsh [37] is that trust subjective and dependent on the views of the individual. Deutsch further refined his definition of trust as *confidence that an individual will find what is desired from another, rather than what is feared* [13]. This definition is also echoed by the Webster dictionary which defines trust as *a confident dependence on the character, ability, strength, or truth of someone or something*.

Another popular definition of trust adopted by computer scientists is the one coined by Diego Gambetta [18]. He defined trust as *a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action*. Gambetta introduced the concept of using values for trust and also defended the existence of competition among cooperating agents. A recent definition of trust has been put forth by Grandison and Sloman [19] who define trust as *the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context*.

There are several interesting aspects of trust. Trust is conditionally transitive. This means that if A trusts B and B trusts C, A trusts C only if certain conditions are met. Trust can be multi-dimensional and depends upon the particular context. For example, A may trust B completely in the context of cars but may not trust B in the context of planes. Trust can also be expressed in terms of a set of continuous, or binary, or discrete values.

2.2. Reputation

Related to trust is the concept of reputation. Abdul-Rahman [2] defines reputation as *an expectation about an individual's behavior based on information about or observations of its past behavior*. Kinateder and Rothermel treat reputation as a global aspect and consider trust to be local and subjective. They define reputation of an entity *as the average trust of all other entities towards the entity* [30].

An individual's reputation can be used to determine the extent to which he can be trusted. An individual who is more reputed is generally considered to be more trust worthy. Reputation can be generally determined in three ways, either by relying on one's personal and direct experiences,

or relying on the experiences of other people, or a combination of both.

2.3. Related Work in Trust Management

Existing decentralized trust models can be classified into two main categories: credential and policy-based, and reputation-based. This categorization is based upon the approach adopted to establish and evaluate trust relationships between peers.

In credential and policy-based trust management systems such as in [5, 26, 34, 50, 52], peers use credential verification to establish a trust relationship with other peers. The primary goal of such systems is to enable access control. Therefore their concept of trust management is limited to verifying credentials and restricting access to resources according to application-defined policies [19]. A resource-owner provides a requesting peer access to a restricted resource only if it can verify the credentials of the requesting peer either directly or through a web of trust [29]. This is useful by itself only for those applications that assume implicit trust in the resource owner. Since these policy-based access control trust mechanisms do not incorporate the need of the requesting peer to establish trust in the resource-owner, they by themselves do not provide a complete generic trust management solution for all decentralized applications.

Reputation-based trust management systems on the other hand provide a mechanism by which a peer requesting a resource may evaluate its trust in the reliability of the resource and the peer providing the resource. Examples of such systems include SPORAS and HISTOS [53], XREP[10], NICE[33], DCRC/CORC [20], Beta [22], EigenTrust [27], etc. Peers in such systems establish trust relationships with other peers and assign trust values to these relationships [54]. Trust value assigned to a trust relationship is a function of the combination of the peer's global reputation and the evaluating peer's perception of that peer.

Some reputation-based systems, in addition, utilize social relationships between peers when computing trust and reputation values. These systems analyze a social network which represents the relationships existing within a community and form conclusions about peers' reputations based on different aspects of the social network. Examples of such trust management systems include REGRET[40] that identifies groups using the social network, and NodeRanking[39] that identifies experts using the social network.

3. Trust Model

A trust model essentially helps model the trust relationships between peers in the system. Different types of trust and reputation models exist in the literature. These models have been developed towards different objectives and targeted at specific applications. A trust model means different things to different people. For some, it may mean just a trust algorithm and a way of combining different trust information to compute a single trust value [22, 27], while for others, a trust model may also encompass a trust-specific protocol to gather trust information[1, 10]. Yet others may want a trust model to also specify how and where trust data is stored.

3.1. Definition

In spite of having a number of trust and reputation models, the research community has devoted little effort towards presenting a unifying definition of a trust model and identifying the essential elements that constitute a trust model. To address this shortcoming, we present the following definition of a trust model. *A trust model describes what trust information is used to establish trust relationships, how that trust information is obtained, how that trust information is combined to determine trustworthiness, and how that trust information is modified in response to personal and reported experiences.*

3.2. Protective Mechanisms

Different trust models encompass different mechanisms to protect the peer in the face of attacks executed by malicious peers. These mechanisms can be classified as either *preventive*, *detective*, or *reactive*. Since we use these mechanisms to compare the abilities of different reputation-based trust models in the face of threats, it is important to understand the nature of these mechanisms.

Preventive mechanisms are those that provide the first level of security against any attacks. The goal of these mechanisms is to discourage any attack at the outset by preventing vulnerabilities. The best preventive mechanism in an open decentralized system would be for a peer to refrain from interacting with any other peer. However, this would hinder peers from interacting and forming relationships with each other, which is the single-most important objective of participation in such an application. Due to this essential trade-off between security measures and ability to interact, employed preventive measures may not be sufficient by themselves. Additionally, in spite of employing preventive mechanisms, newer forms of attacks are continuously invented. Thus, *detective* mechanisms are needed so that the system can, upon being attacked, be able to detect any attacks at the earliest.

Some threats are easy to detect while others require a careful analysis of data over a period of time to conclude that an attack is being carried out. This period of time may vary depending on various factors. Two principal factors are the constancy of the malicious behavior, and the extent to which the situation and data reported by other peers is

analyzed and understood. For example, it will take longer to detect a peer that indulges in fraudulent behavior once in ten interactions than a peer that exhibits such behavior constantly. Further, a peer that evaluates others based on their behavior only in the last few hours is more likely to overlook inconstant malicious activity. Thus these factors need to be understood and addressed in order to achieve a high level of effectiveness in detection mechanisms.

But detection mechanisms by themselves will not help achieve anything. It is imperative that, upon detection, the system react in such a manner that the effects of the threat are immediately nullified. Or in the worst case if the system cannot counter the threat, it should react in such a way that it can better protect itself against such threats in the future. Such mechanisms that define the reaction to threats and attacks are termed *reactive* mechanisms.

Reactive mechanisms define how the reputation of peers in the system gets affected as a result of the attack. It also includes a feedback mechanism that dictates how the information about the threat and the malicious peer is propagated to the rest of the system. This feedback mechanism can be of two types - *proactive* and *passive*. *Proactive* mechanisms include actions pursued by a peer towards actively informing the rest of the system about malicious peers. An example of such a mechanism is where a peer pushes trust information digests and warnings to other peers in the system. *Passive* mechanisms, on the other hand, are those that passively disseminate information about malicious peers. An example of a passive mechanism is where a peer may respond with relevant trust information only in response to specific queries.

To better understand these three mechanisms, consider the example of a burglar trying to rob a house. A preventive mechanism to discourage a burglar is to lock the house securely. Anticipating that a skillful burglar will still be able to gain entry into the house, security cameras can be installed to detect presence of burglars in the house. Once a camera detects the presence of a burglar, alarms can be triggered and proper authorities can be informed immediately as proactive responses. These protective mechanisms in the face of a burglary are illustrated in Figure 1.

4. Decentralization Threats

The decentralized nature of a system exposes peers in a system to threats that arise due to the actions of malicious peers. We classify these threats into two categories: direct and indirect. Direct threats are those that result from actual attacks on peers. Indirect threats refer to conditions/situations that exacerbate the risk due to direct threats. Below we describe critical direct and indirect threats that arise due to the nature of decentralization.

It should be noted that the attacks discussed in this section are not necessarily unique only to decentralized systems. These threats may be present in a centralized system as well; however, the absence of a centralized authority make it harder to safeguard against these attacks in a decentralized system. Hence these attacks warrant a detailed study in a decentralized scenario.

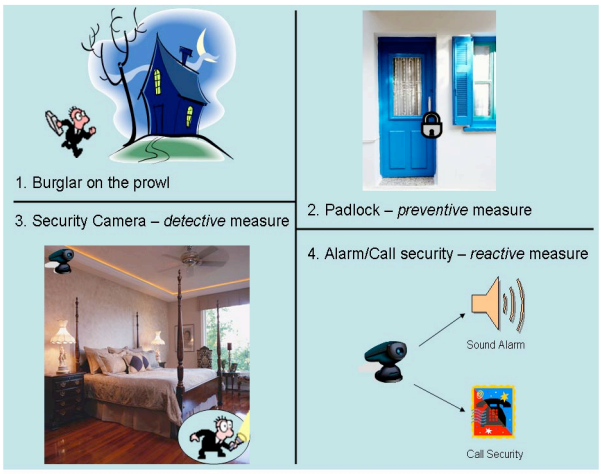

Figure 1. Protective measures against burglary

4.1. Direct Threats

Direct threats are those that are caused due to the attacks perpetrated by malicious peers in the system. These attacks are typically directed against two assets of a decentralized application - *data* and *service*. Data refers to both application-specific and trust-specific information that is exchanged among peers, and service refers to both application-specific and trust-specific utility of specific value that is being offered by a peer.

These attacks that are directed against either data or service can be classified into two categories - *data-targeted* and *peer-targeted*. *Data-targeted* attacks are those that directly target the service or data itself in order to exploit or misguide other peers. For example, such attacks may focus on obstructing availability or access of correct information or focus on propagating wrong information. *Peer-targeted* attacks, on the other hand, are those that target a peer offering a service or data in order to either take advantage of its reputation in the community or to lower its reputation by offering spurious or malicious information.

There are also other attacks that target the ability of a peer to respond to important and legitimate queries by flooding it with either ill-formed or redundant queries. These are typical denial-of-service (DoS) attacks that can be perpetrated by malicious peers who may want to delay the propagation of certain critical information to other peers in the system. One way to prevent these attacks is to detect and isolate as early as possible all ill-formed or

already received messages so that they are not processed again by the peer.

Attacks in a decentralized system either These attacks may even aim to avail of privileges of other peers. This information may be either application-related or trust-related data.

4.1.1. Impersonation

Impersonation refers to the threat posed by a malicious peer that portrays itself as another peer. The goal behind this threat could be to either misuse the privileges made available to the impersonated peer by other peers, or malign the impersonated peer through fraudulent interactions with other peers. Pseudospoofing and man-in-the-middle attacks can also be categorized as impersonation threats. Pseudospoofing exploits the use of pseudonyms in a P2P system[10]. Malicious peers can create and control multiple false identities. Once an identity gains a bad reputation, it can be easily discarded and a new one can be adopted. Man-in-the-middle attacks are typical in P2P systems that rely upon application-level routing. In such cases, interacting peers need to rely upon intermediate peers to forward their queries or responses. This offers intermediate peers with malicious intentions the opportunity to tamper with the responses.

4.1.2. Fraudulent Actions

In a peer-to-peer application, peers interact with each other in a variety of ways such as exchanging information, transacting deals, etc. While interacting with other peers in the system, a fraudulent peer may not completely fulfill its part of the transaction, or it may promise availability of certain services that it does not really offer. A trust model should (a) help peers identify such fraudulent peers to prevent these attacks, and (b) post-interaction enable peers to inform others about these fraudulent peers.

4.1.3. Misrepresentation

A malicious peer may mis-represent the extent of trust it has in a victim peer and communicate these incorrect values to other peers. For example, a malicious peer could actually trust a victim peer but send out reports contrary to its knowledge. Depending upon the influence of the malicious peer, this may adversely affect the interaction of the victim peer with other peers in the system. Moreover, such a peer with malicious intentions could also mis-communicate the extent of trust another peer has in the victim peer. This problem is further compounded if the malicious peer acts as a forwarding relay between peers. Solutions to this problem include actively informing other peers about malicious peers and incorporating the opinions of multiple peers while making trust decisions in order to reduce the effect of mis-representation.

4.1.4. Collusion

Collusion refers to the threat posed when a group or groups of malicious peers actively try to subvert the system. Their actions may include spreading negative accounts of good peers and reporting greatly exaggerated positive accounts of other malicious peers in their clique[3, 11]. This leads to a situation where good peers are isolated and cannot decide whom to trust and may lead to a complete disruption of the system. Additionally, collusion includes the problem of shilling[10], where a malicious peer can create multiple false identities, each of which maps to a unique IP address, in order to augment misleading information.

Collusion can be addressed by encouraging good peers to actively (a) recognize groups of malicious peers and spread information about them, and (b) form robust groups themselves to counter the effects of collusion [33].

There are other attacks such as Denial of Service(DoS) attacks that are direct threats but these are not considered here since they are not directly relevant to trust management systems.

4.2. Indirect Threats

In addition to the direct threats introduced in the previous section, there are a number of scenarios that make these direct threats difficult to address. We term these scenarios as indirect threats and discuss the critical ones below.

4.2.1. Addition of Unknowns

When a new peer joins an existing system, it does not possess trust-based knowledge about other peers in the system which may hinder it from interacting with other peers. Similarly existing peers in the system may tend to isolate the new peer since they lack trust information about the new peer. A trust model, therefore, should have a low barrier of entry for new peers so that new peers can easily participate in the system. Yet, at the same time, the trust model should provide sufficient measures to protect the system if the new peer turns out to be malicious. Addition of Unknowns also encompasses the cold start problem which arises when the peer-to-peer system is first initialized and none of the peers have any trust information about any peer.

5. TREF Framework

In this section, we discuss for each of the threats the different types of preventive, detective and reactive mechanisms employed by existing reputation-based trust models in order to counter that threat.

5.1. Impersonation

Preventive - Impersonation can be discouraged through the use of unique digital identities. A digital identity may represent one or more physical peers and can even repre-

sent an organization. The use of digital identities protects against pseudospoofing. Thus, any relationships will exist between digital identities instead of physical peers. This allows for the possibility that a physical peer may have multiple digital identities each of which is considered a separate peer. Standard Public Key Infrastructure (PKI)[14] can be used for generating digital identities where each peer will be represented by a public-private key pair that uniquely identifies it.

Detective - Impersonation can be detected by verifying the authenticity of messages through the use of digital signatures and authentication. This has been adopted by several existing trust models [1, 10, 16, 23, 33]. Each peer first generates a pair of public-private keys. When an originating peer sends a message to another peer, it signs the message with its private key. The receiving peer uses the public key of the originating peer to verify the authenticity of the message. This helps detect unsigned and forged messages. While authentication cannot prevent other peers from reading the contents of the message, use of authentication can also help detect repudiation attacks and man-in-the middle attacks.

Reactive - Once impersonation is detected and the signature on a particular message is not found to be authentic, or if the message is unsigned, the message can be flagged with a warning or can be dropped depending upon the nature of the trust model[43].

5.2. Fraudulent Actions

Preventive - A typical mechanism for preventing fraudulent actions is to adopt the policy of interacting only with those peers who are considered trustworthy beyond a threshold level. This threshold level can be set in several ways. It may depend upon the nature of the application or the trust model and may be a pre-determined value. Or it may depend upon the context of the particular interaction where a peer may decide to trust or distrust based upon the degree of need for interaction and the risk posed by the interaction[24, 46]. Another preventive mechanism could be for peers to collectively enforce a trust policy that enforces a severe penalty for indulging in fraudulent actions. The punishment could range from reducing the fraudulent peer's trust to a system-wide isolation of the fraudulent peer. If these policies and mechanisms are advertised system-wide before-hand, it may have the effect of potentially discouraging peers from engaging in fraudulent actions.

It must be pointed out that these mechanisms would require peers to collectively agree on a policy and enforce it individually. However, in a decentralized system, peers are autonomous and make their own decisions. Collectively agreeing and adopting a policy is an inherently difficult thing to achieve in a real decentralized system. Even if peers promise to enforce a policy, in a decentralized system, peers cannot be really trusted to stick to the agreement, even if it may be for their benefit.

A reverse mechanism could also be used i.e. peers could be encouraged to behave in a reliable and trustworthy fashion by offering them suitable rewards proportional to their trustworthiness. These mechanisms are called incentive-based mechanisms and have been used in several P2P applications to counter the problem of free-riding[15, 21, 25, 31]. Schemes such as those in Credence[48] motivate peers to speak the truth consistently for their opinions to have an effect. A novel incentive compatible trading mechanism has been proposed that proves the most optimal solution for untrustworthy peers is to truthfully report their untrustworthiness before interacting with other peers[6].

Detective - A peer can detect a fraudulent peer by either looking up its own history for past interactions with the concerned peer or querying other peers in the system for trust information about the concerned peer or a combination of both[1, 40]. Thus it is important that peers maintain a detailed history of the behavior or the past reputation of other peers so that a well-informed decision about the trustworthiness of those peers can be made in the future.

Peers that are queried may be limited to those who are either completely trusted or trusted beyond a certain threshold. This threshold as described earlier may depend upon the application, the trust model, and the risk factor of the particular interaction[46]. The queries may also be broadcast to all peers in the system with a suitable hop count. Received trust information may be combined appropriately to include the trustworthiness of the responding peers in order to determine whether the concerned peer is likely fraudulent or not[2, 10]. Several reputation models employ some or the other form of conditional transitivity on trust relationships to form their own estimates of the reported trust[1]. Trust topologies to capture the diversity in trust relationships have also been proposed[23].

Some models such as PeerTrust[49] also use transaction and community context to determine trustworthiness. PeerTrust also requires recommenders to report the total number of transactions they have had with the target peer since the ratio of unsuccessful to successful transactions better reflects the trustworthiness of the target peer. This requirement helps address the problem of skewed transaction distribution found in systems such as eBay[17].

Reactive - It should be noted that in a decentralized system, it is not always possible to conclude with certainty that a particular interaction is successful or unsuccessful or whether a perceived fraudulent action is the fault of the other peer. However, once a peer has determined that it has been the target of a fraudulent action, it can react in a number of ways. First, the victim peer must take suitable action to stop the attack. For example, if the attack is in the form

of a worm that was downloaded because it was disguised as a legitimate file, the victim peer must block the worm so that it does not flood the network. Or if a node on the network is using the victim node to launch attacks against other nodes, the victim node must block the attacker's traffic as soon as it detects the attack on itself[9].

Second, the victim peer can reduce its trust in the fraudulent peer that was responsible for the attack. Third, the victim node can reduce its trust in those who recommended that fraudulent peer. Existing trust models typically reflect both these reactions[1, 16, 40, 48]. However, these models employ different kinds of policies to determine the extent of these reductions in trust. These policies may also depend upon the nature of the application and the severity of the attacks. For example, some applications may not tolerate even a single fraudulent action even though the concerned peer may have been considered completely trustworthy in the past. In such a case, the trust model may employ a stringent trust reduction policy that will reduce trust in the peer to a level where it is considered untrustworthy.

Some reputation models that rely on negative reputation issue negative recommendations[9] or complaints[4] when victimized by a fraudulent action. Since complaints only contain information about the attack and the fraudulent peer, these models do not decrease the reputation of the fraudulent peer but rather generate complaint statements and distribute them in the network.

Third, the victim peer can choose to actively or passively disseminate information about the fraudulent peer to other peers in the system. An active dissemination mechanism sends explicit reputation revocation messages [1, 51] or to spread a warning throughout the system[9]. A passive dissemination mechanism, on the other hand, spreads the information through recommendations in response to relevant queries[10, 49]. Another passive mechanism is the one employed in the NICE[33] trust model where peers exchange trust data "digests" once in a while to keep themselves abreast of the reputation of other peers.

5.3. Misrepresentation

Preventive - Mechanisms for preventing misrepresentation closely resemble those for preventing fraudulent actions. Thus, peers could collectively enforce and advertise a system-wide policy of seeking trust information from only those peers who are considered completely trustworthy or trustworthy beyond a certain threshold determined by the application and the nature of the trust model. Similarly, peers could also enforce a severe penalty if any peer is found to be lying. However, these mechanisms would require peers to agree on a policy and enforce it individually which is inherently a difficult thing to achieve in a decentralized system where peers are autonomous and have different individual goals. Peers could also be encouraged to report reliable and trusted information by offering them suitable incentives in return.

An extremely simple mechanism to prevent misrepresentation is for a peer to rely only on its own past experience because this would eliminate the need for determining the trustworthiness of recommenders[35]. However, a peer may have little or no trust data on a peer it wants to interact with, and so may have to rely on others for trust information. Additionally, if the trust model uses the concept of trust context described earlier, a peer can choose to listen to peers who are reporting information in that context and considered trustworthy in reporting information in that context. As an example, it would be difficult to trust an expert in cars if information on planes was being reported. Using such a policy will help a peer to filter out recommendations that are not relevant and reduce the number of recommenders whose trustworthiness needs to be determined.

Detective - A peer can detect if a recommender is misrepresenting by comparing the trust information reported by the recommender against the peer's own perceptions and beliefs. Further, the peer can check its own past interactions to determine whether the recommender has a history of engaging in misrepresentation. Additionally, a peer could query other peers in the system for trust information about the recommender peer. Again, peers that are queried may be limited to those who are either completely trusted or trusted beyond a specific threshold, or these queries can be broadcast to all peers in the system with a suitable hop count. Peers that respond to these queries essentially serve as recommenders for a recommender.

Received trust information may be aggregated and appropriately combined to include the trustworthiness of the responding peers in order to determine the trustworthiness of the original recommender. A simplistic mechanism adopted is to average the opinions received from all the peers. Since only one peer is malicious, this mechanism would work as long as there is more than one truthful recommender. In the event, however, where there is just one truthful response along with a misrepresented response, a peer must evaluate the reputation of both the recommenders in order to determine who can be trusted more. For this, it can make use of its own personal past information, group relationships, and opinions provided by other trusted recommenders[40]. Another mechanism that has been adopted in some trust models is to include the trustworthiness of the responders as weights to compute a weighted average trust value for a recommender.

Interestingly, some trust models such as PET[35] adopt the stance that recommenders cannot be trusted completely or even to the extent of the peer's own experience. They, therefore, recommend averaging received recommendations and assigning the average a low weight to ensure that the personal experience has a bigger role in the determina-

tion of trust. Recent work by Marti and Garcia-Molina also shows that limited reputation sharing may in fact be a good thing because it can reduce the number of failed transactions by a factor of 20[38].

Another interesting model is Credence[48], a decentralized object reputation management system which employs a voter correlation scheme to help identify distinguish truthful peers from lying ones. The use of the correlation scheme help identify a consistent liar quickly. If an attacker, however, mixes up the lies and truths, the correlation will tend to become zero, in which case the attacker's opinion is not given much value. In such a scheme, a peer must consistently speak the truth for its opinion to have some value. A similar mechanism has been proposed by Buchegger and Boudec[7] that uses a Bayesian approach to exclude opinions that deviate substantially from personal opinion and the majority of recommender opinions.

Reactive - In spite of the preventive and detective measures, a peer may become the target of misrepresented information. It could be because the malicious peer's past history is unavailable or because this was the first attack by the malicious peer. The only way a peer can detect an attack in these cases is by examining for itself whether the information reported by a peer matches its own interaction experience in the future.

The reactive mechanisms to counter misrepresentation resemble those for countering fraudulent actions. Once a peer realizes that it has been a target of a misrepresentation attack, it can react by reducing its trust both in the misrepresenting peer as well as in those peers who recommended the misrepresenting peer. A trust model may employ different kinds of policies to determine the degree of trust reduction. Again these policies may also depend upon the nature of the application. Additionally, the victim peer can choose to use an active or passive dissemination mechanism to spread information about the misrepresenting recommender so as to warn other peers in the system.

5.4. Collusion

Preventive - The preventive mechanisms used to counter misrepresentation are also effective in preventing collusion attacks. Thus, querying only trusted recommenders, enforcing severe penalties for collusion and using incentive mechanisms serve as preventive mechanisms for collusion. As mentioned previously in the case of misrepresentation, it is also better for a peer to rely on its own past experience with respect to peers and recommenders than to believe others in the system[35]. This helps reduce the risk of misrepresentation and collusion threats. However, if a peer has limited or no previous interaction with a peer or a recommender, it has no other option but to make a decision based on the opinions of other peers.

Dellarocas[11] proposes a controlled anonymity scheme that conceals the identities of the peers. This makes it difficult for malicious peers to collude and attack a specific peer. However, in a decentralized system, there is no single centralized authority that can be trusted to control and conceal identities. Additionally, peer anonymity makes it harder to establish trust relationships[44]. Further, anonymity is not always possible for all decentralized applications.

Some additional mechanisms can also be employed to discourage collusion. For example, in the NICE trust model[33], each peer creates and maintains a list of "friend" recommenders that the peer absolutely trusts. While the peer may query other peers in the system for trust information, only the recommendations of the friend recommenders are completely trusted and used in the determination of trustworthiness.

Detective - When the opinion of recommenders differ, it can be difficult to determine who is lying. The querying peer cannot assume that the majority opinion amongst all received opinions is always true. It is possible that colluding recommenders may outnumber good recommenders and so the peer could possibly end up believing in the false information. It is therefore essential for the peer to evaluate the reputation of the recommenders in order to trust a particular opinion. There are several mechanisms that can help a peer determine the trustworthiness of recommenders. Some of these have already been discussed in the case of misrepresentation.

One simple mechanism is for the peer to match reported opinions against the peer's own perceptions and beliefs. A peer can also look up its past interactions to determine whether any of the recommenders have previous engaged in any malicious activity. It can also evaluate the reputation of the recommenders by using existing group relationships[40].

A peer may decide to query other peers about the trustworthiness of the recommender. The reported information can be used suitably in determining trustworthiness of the recommenders. While models such as PET[35] advocate the reduced use of recommendations and increased use of personal experience in determining trust and promote, models such as NICE[33] rely more or exclusively upon the opinion of peers that a peer considers trustworthy.

However, it should be noted that if all recommenders have excellent reputations, it becomes difficult to decide whom to trust. In such a case, the peer can use suitable mechanisms such as risk analysis[24] to decide whether it should trust certain recommenders or proceed with certain interactions.

Sometimes, the same malicious peer may create multiple identities and operate from different machines in order to give the impression that multiple peers have the same opinion. One way to protect against such an attack when IP

addresses are known is to use IP clustering[10]. The assumption behind IP clustering is that such a malicious peer will need access to a bunch of machines that could very likely be in the same subnet. IP clustering helps aggregate the similar opinions of these peers with matching subnet addresses into a single opinion in order to reduce the impact of multiple malicious entities.

Using the example of the Byzantine General's problem, it has also been proven that in a system where unforged signed communication is used, a malicious group of peers can be overcome[32]. Thus, the use of unique digital identities and explicitly signed messages between peers can help detect a collusion attack.

Reactive - In spite of the preventive and detective measures, a peer may become the target of a collusion attack. This could be if information about the malicious peers is unavailable or if this was the first instance of such an attack. The only way a peer can truly recognize that an attack has taken place in these cases is by examining for itself whether the information reported by peers matches its own experience in the future.

Once a peer finds out that it was the target of a collusion attack, it can take several actions. It can reduce its trust in the attackers as well as those who recommended those attackers. A trust model may employ different kinds of policies to determine the degree of this trust reduction. As already mentioned previously, these policies may also depend upon the nature of the application. Additionally, the victim peer can choose to use an active or passive dissemination mechanism to spread information about the colluding peers so as to warn other peers in the system.

5.5. Addition of Unknowns

A decentralized peer primarily relies on its own and others' past experience in order to establish trust relationships. A peer's past behavior while insufficient to precisely guarantee its future actions, however serves as a reasonable basis for predicting its future actions. However, an inherent problem with open decentralized applications is the presence of unknown peers in the system. These peers may be either first-time entrants to the system or old participants about whom there is no longer any information available in the system. While the presence of such unknown peers in the system does not pose a direct threat to the system, interacting with these unknown peers in an open decentralized system may be fraught with danger. In the absence of any information about the unknown peer, peers in the system may hesitate to trust a new peer and vice-versa.

There are several mechanisms that can help in alleviating this problem. A peer may evaluate the cost of the outcome[8] to decide whether to proceed with interaction with the unknown peer. If the interaction is critical from the peer's perspective, the peer may decide to not risk interaction until reliable information about the peer is available in the future. On the other hand, if the interaction is of a low-risk nature, the peer may proceed with the interaction and help form the unknown peer's reputation. If the ensuing interaction is successful, the unknown peer's reputation will increase. A negative interaction will on the other hand alert other peers in the system about the unknown peer.

A new entrant peer to the system, prior to an interaction, can query other peers about the peer it wants to interact with. These opinions can be combined to evaluate the interacting peer's reputation. However, since the new peer does not know whom to trust, it could easily become the victim of a collusion attack. The new peer therefore must discover trusted peers by engaging in low-risk interaction with peers in the system. These trusted peers can in the future be then queried for recommendations. If the new peer does indeed become the target of any attack, it can react by reducing its trust in the attackers and informing other peers in the system about the attackers.

Different trust models provide different kinds of measures to address the problem of new or unknown peers in the system. For example, the Distributed Trust Model[1] provides a new peer with an initial list of trusted peers that the peer can interact with. In the XREP model[10], new peers can build up their reputation by providing well-known reliable resources. Some models, such as REGRET[40] and Community-based Reputation[51] are liberal and have a low barrier of entry for new peers. A peer's reputation in REGRET rapidly increases with every good interaction, so that in a short while the peer can build up a good reputation and can participate fully in the system. Similarly complaint-based models completely trust new peers until complaints about them are reported. However, some other models are more cautious and take into consideration factors such as the outcome's intrinsic cost[8] and cost of the transaction[24, 36] before trusting a new peer.

6. TREF Validation

We validated the TREF framework in the context of four candidate decentralized reputation models. We used TREF to perform a threat-based theoretical comparison of the models. Then each model was separately implemented in an emergency response system called CRASH. Threat scenarios corresponding to the five critical threats were designed and executed on the four CRASH prototypes and observed results were compared against expected results from the TREF analysis. This comparison allowed us to examine the validity of the theoretical results obtained from the TREF framework. Below, we first summarize the four reputation models that were used in our evaluation. Next, we provide a brief description of the CRASH system

and how we used it our validation.

6.1. Candidate Reputation-based Trust Models

The TREF framework is based on the threats of decentralization, and describes and compares the protective, detective and reactive mechanisms of different reputation-based trust models in the face of those threats. This section describes the various aspects of the TREF framework using four sample reputation-based trust models. These models are the Distributed Trust Model[1], NICE[33], REGRET[40], and a Complaint-based Model. Below, each of these models is first introduced followed by a discussion of the various mechanisms in the model that help counter the threats.

6.1.1. Distributed Trust Model

6.1.1.1. Description

In the Distributed Trust model proposed by Abdul-Rahman [1], a trust relationship is always between exactly two entities, is non-symmetrical, and is conditionally transitive. There are two distinct trust relationships. A direct trust relationship is when one peer trusts another. But if a peer trusts another peer to give recommendations about another peer's trustworthiness, then there is a recommender trust relationship between the two [2]. Trust relationships exist only within each peer's own database and hence there is no global centralized map of trust relationships. Corresponding to the two types of trust relationships, two types of data structures are maintained by each peer - one for direct trust experiences and another for recommender trust experiences. Recommender trust experiences are utilized for computing trust only when there are no direct trust experiences with a particular peer.

Trust categories are used by peers to classify trust towards other peers depending upon which aspect of that entity is under consideration. For example, a peer may trust another peer on a certain issue but may not trust it in another context. Similarly, since a peer may trust a certain peer more than other peers, comparable trust values are needed. A reputation is defined as a tuple consisting of a peer's name, the trust category and the specific trust value. A recommendation is defined as communicated trust information which contains reputation information.

6.1.1.2. Countering Threats

The Distributed Trust model uses key-based encryption of messages so that recommendation information is not easily obtained by malicious peers. The model uses discrete levels of reputation that limit the ability of a peer to express confidence in other peers. However, it provides the use of explicit reputation revocation to inform other peers about fraudulent peers. A peer can guard against misrepresentation and collusion by sending requests for recommendations only to trusted recommenders. However, once a peer is subjected to mis-representation and collusion attacks, its resistance is limited to decreasing the recommender trust of the malicious peers. Unless explicitly questioned by other peers in the system, these targeted peers do not inform others about the actions of misrepresenting and colluding peers. New peers that join the system are equipped with an initial list of trusted peers with whom they can interact and slowly build up their reputation through good interactions.

6.1.2. NICE

6.1.2.1. Description

NICE [33] is a platform for implementing distributed cooperative applications. Applications based on NICE barter local resources in exchange for access to remote resources. NICE provides three main services: resource advertisement and location, secure bartering and trading of resources, and distributed trust valuation. The trust evaluation is necessary since malicious peers may threaten the reliable functioning of the cooperative system. Consequently, the objective of the NICE trust inference model is to a) identify cooperative users so that they can form robust cooperative groups, and b) prevent malicious peers and clusters to critically affect the working of the cooperative groups.

Like other trust models, the NICE model utilizes the opinion of each transacting peer to rate the quality of the transaction. This opinion signed by a peer is called a cookie and is the measure of reputation in the NICE model. This cookie is stored on the other transacting peer which can use this cookie to prove its trustworthiness to other users. If, however, the opinion is negative, the peer storing it has no incentive to retain it, so in this case, the peer signing the opinion stores the cookies itself.

When a peer P wants to access Q's resources, it sends Q a set of cookies signed by Q. Upon receiving this, Q verifies that the cookies were indeed signed by it. Depending on the set of cookies, Q may also decide to search for further references for P. These references along with the cookies are then used to compute the extent of Q's trust in P. In case, P does not have cookies directly signed by Q, it can generate a trust graph originating from P to Q and can present this to Q who can then use the trust graph to infer the trustworthiness of P.

6.1.2.2. Countering Threats

In NICE, each cookie containing trust information about the requestor is signed by the owner and is verified by the owner upon receipt. Identifiers and public keys are used to verify the credentials of the requestor. Storing negative cookies and exchanging digests with other peers allows information about malicious peers to be dissipated to other peers in the system. This makes the system aware of fraudulent peers. When a peer initiates a search for negative cookies on a target peer, it only relies upon negative cookies received from trustworthy peers. Thus even if a

malicious peer were to misrepresent its trust in the target peer, combining the opinions of other peers will help counter the misrepresentation. One of the main contributions of the NICE approach is the ability of good peers to form groups themselves and isolate malicious peers. To form such groups efficiently, peers maintain a preference list of potentially trustworthy peers that is constructed based on previous interactions and observations. This helps address the threat of collusion, since good peers only ask other good peers for digests.

There are no cookies at system start and peers can build up reputation only with successful interactions with other peers. Thus, there is no well-defined solution to the problem of addition of unknowns.

6.1.3. REGRET

6.1.3.1. Description

REGRET [40] is similar in concept to models such as TrustNet [42] that include the social dimension of peers and their opinions in its reputation model [41]. REGRET adopts the stance that the overall reputation of a peer is an aggregation of different pieces of information. REGRET is based upon three dimensions of reputation - *individual*, *social*, and *ontological*. REGRET combines these three dimensions to yield a single value of reputation. When a member peer depends only on its direct interaction with other members in the society to evaluate reputation, the peer uses the *individual dimension*.

If the peer also uses information about another peer provided by other members of the society it uses the *social dimension*. The social dimension relies on group relations. In particular, since a peer inherits the reputation of the group it belongs to, the group and relational information can be used to attain an initial understanding about the behavior of the peer when direct information is unavailable. Thus, there are three sources of information that help peer A decide the reputation of a peer B - the individual dimension between A and B, the information that A's group has about B called the *Witness* reputation, and the information that A's group has about B's group called the *Neighborhood* reputation. Figure 2 illustrates these various reputation relationships.

REGRET believes reputation to be multi-faceted implying that a reputation in a specific context may summarize the reputations of other dependent factors. The different types of reputation and how they are combined to obtain new types of reputation is defined by the *ontological dimension*. Clearly, since reputation is subjective, each peer typically has a different ontological structure to combine reputations and has a different way to weigh the reputations when they are combined. REGRET is richer than other trust models since it considers both group reputation and the ontological dimension in the computation of reputation.

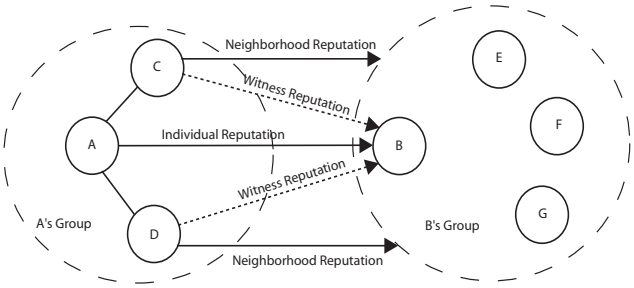

Figure 2. Individual and Social Reputation in REGRET

The use of social relationships in the REGRET model help peers to better defend against attacks. Upon detection of fraudulent actions, affected peers can modify not only the reputation value of the malicious peer and the group that it belongs to, but also that of the witnesses who recommended the fraudulent peer. These changed values will forewarn other peers in the future. In addition to using this technique, a peer can combine opinions of multiple witnesses to detect mis-representation. Collusion can be prevented by combining the various social reputation mechanisms provided the number of good peers is sufficiently greater than the number of malicious peers. New peers that join the system start with zero reputation but quickly build up their reputation through successful interactions. A significant disadvantage of the REGRET model is the lack of credential verification thus making it susceptible to impersonation attacks. Another shortcoming is that each peer assumes an implicit trust in other peers belonging to the same group, thus exposing itself to possible malicious activity within its own group.

6.1.4. Complaint-based Model

In a complaint-based model, negative reputation information is encapsulated and stored as a complaint. An instance of such a model is the P-Grid approach which focuses on an efficient data management technique to construct a scalable trust model for decentralized applications [4].

The complaint-based trust model is based on binary trust. Peers perform transactions and if a peer cheats in a transaction, it becomes untrustworthy from a global perspective. This information in the form of a complaint about dishonest behavior can be sent to other peers. Complaints are the only behavioral data used in this trust model. Reputation of a peer is based on the global knowledge on complaints.

Upon interaction, peers evaluate each other. A peer can if needed file a complaint about another peer and send it to other peers who maintain copies of the same complaint. When a peer wants to evaluate the trustworthiness of another peer, it searches for complaints about that peer. Upon receiving a query, peers that have the required complaints respond accordingly. Since these peers themselves

Table 1. TREF-based comparison of four reputation models

Threats		Distributed Trust Model	NICE	REGRET	Complaint-based
Impersonation	**Preventive**	Use digital identities	Use digital identities	None	None
	Detective	Use signature verification	Use signature verification	None	None
	Reactive	Ignore or flag message	Ignore or flag message	None	None
Fraudulent Actions	**Preventive**	None	None	None	None
	Detective	Use past interaction data; query other peers and use recommender trust	Use negative cookies; query other peers and build trust graphs to determine trust	Use individual and social reputation to determine trust	Search for complaints pre-interaction.
	Reactive	Decrease direct and recommender trust; explicit reputation revocation to inform other peers	Dissipate information about malicious peers through digests	Reduce individual as well as social (group) reputation	Actively file complaints post-interaction
Misrepresentation	**Preventive**	Use personal experience first	Depend primarily on self-signed cookies	Use context to filter out responses so that trustworthiness of fewer peers needs to be determined	Let peers know complaints will be filed if misrepresentation is detected
	Detective	Use past interaction data; query other peers and use recommender trust	Use negative cookies; query other peers and build trust graphs to determine trust. Trust "friends" more	Use "Witness Reputation" and combine multiple responses	Combine replicated trust data to detect; check trust of informants and combine their opinions
	Reactive	Decrease direct and recommender trust; explicit reputation revocation to inform other peers	Dissipate information about malicious peers through digests	Reduce individual as well as social (group) reputation	Actively file complaints post-interaction
Collusion	**Preventive**	Use personal experience first	Trust only "friend" peers	Use context to filter out responses so that trustworthiness of fewer peers needs to be determined	Let peers know complaints will be filed if misrepresentation is detected
	Detective	Use past interaction data; query other peers and use recommender trust	Form robust cooperative groups using a preference list of "friend" peers. Depend upon these groups for information	Use "Social Reputation". Works only if number of good peers > number of malicious peers	Combine replicated trust data to detect; check trust of informants and combine their opinions
	Reactive	Decrease direct and recommender trust; explicit reputation revocation to inform other peers	Sign negative cookies and dissipate data about colluding peers through digests	Reduce individual as well as social (group) reputation	Actively file complaints post-interaction
Addition of Unknowns		New peers have an initial list of trusted peers that they interact with	No cookies at start. Peers build up trust with successful transactions	Zero at start, but rises quickly with successful interactions	All new peers are trusted until complaints against them are found

can be malicious their trustworthiness needs to be determined. Consequently, queries for complaints about these peers are sent out by the original peer and so on. In order to prevent the entire network from being explored, which would become expensive in a large system, if similar data about a specific peer is received from a sufficient number of peers, no further checks are carried out.

6.1.4.1. Countering Threats

Peers can protect themselves against fraudulent actions by accessing complaints filed against fraudulent peers. Using trust data replicated across peers protects against the possibility that a complaint is altered by a malicious peer. Mis-representation and collusion are further addressed by checking the trustworthiness of the peer that stores the complaint and the peer that reported the complaint, and combining opinions obtained from multiple trustworthy peers. The threat of addition of unknowns is addressed by trusting all new peers until complaints against them are reported. Table 1 presents a sample TREF framework with the above-described four decentralized reputation models.

6.2. CRASH System

In order to validate our evaluations of the four different decentralized reputation models, we chose to incorporate the models within a decentralized application and execute threat scenarios based on the threats identified in the TREF framework. The application domain we chose for this was decentralized crisis management. In a crisis situation, there are multiple independent parties that exchange information in order to make crucial decisions that can affect lives and property. A significant characteristic of the crisis domain is its dependence upon reliable and accurate information. However, participants in a crisis situation, such as criminals and disgruntled sections of the media, may have malicious intentions and may hamper communication or provide contradictory and incorrect information. Thus, it is very important for entities in a decentralized crisis response situation to employ safeguards against such malicious attacks.

The Crisis Response and Situation Handling system (CRASH) that we developed for our evaluation models a collection of governmental and non-governmental organizations that coordinate and make local autonomous decisions. Each CRASH entity consists of three main subsystems (see Figure 3): Display, Information Sources, and Command and Control (C&C). The Display sub-system facilitates the visualization of the information currently known to the organization. Information Source sub-systems report feedback and information to the entity's C&C sub-system. These sub-systems are connected to the entity's C&C through internal networks. Each entity's C&C sub-system is also connected to the C&C sub-systems of other entities through external networks. The C&C subsystem collects data reported by its information sources and other C&C subsystems. Using this data, the C&C system makes suitable decisions and conveys corresponding information and instructions to its affiliated resources.

Each CRASH entity was built in the PACE architectural style[43]. PACE guides the incorporation of trust management in decentralized applications. Using the guidelines specified by the PACE style, each of the four decentralized reputation models was incorporated within a separate CRASH prototype.

6.3. Executing Threat Scenarios

In order to evaluate the reputation models, each corresponding CRASH prototype was subjected to several demonstration scenarios. These scenarios were based on the threats of decentralization described earlier in Section 5. These scenarios are similar to the attack scenarios mentioned in [47] and were designed to facilitate an evaluation of these models in the context of decentralization threats.

We designed a total of 100 threat scenarios in the context of the CRASH application with an average of about 5

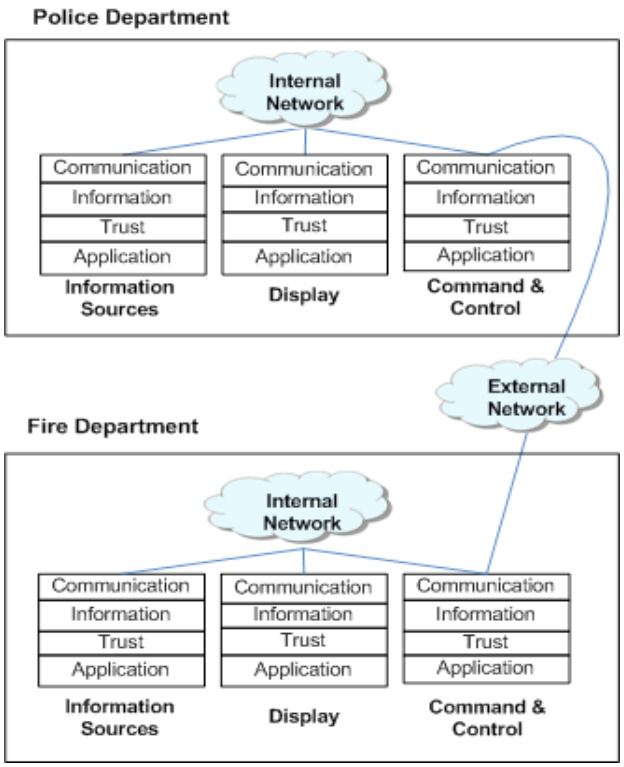

Figure 3. CRASH with Police and Fire Departments

scenarios per threat per model. Each of these scenarios was then executed on the corresponding CRASH prototype, and the effects observed and compared with expected results. Since each trust model has varying capabilities and behaves differently, threat scenarios for each model though similar in objective were not identical.

An in depth discussion of these threat scenarios and the various expected and observed results cannot be included here due to space constraints. Further description of these scenarios can be found in [45].

7. Discussion

Our development of the TREF framework and examining and comparing existing reputation-based trust models using the TREF framework has revealed several interesting insights. The first is that each candidate reputation model only addressed some of the threats to various degrees; none of them fully addressed all the threats. We believe further work is necessary in order to be able to fully address these threats of decentralization. This includes identifying preventive, detective and reactive mechanisms for each threat and leveraging these mechanisms into a more capable reputation model in the future. Adopting such a threat-centric approach for developing reputation-based trust models will result in the future in more secure and capable models that are better able to counter decentralization threats.

The second insight is that the threats of decentralization serve as an excellent means to compare the abilities of different reputation models. Since these models are employed to help peers better protect themselves, the reaction to various threats and attacks provides a more reliable estimate of what model should be adopted and under what conditions. Thus the TREF framework does not just provide a means to compare models, but also has the potential to point towards a suitable reputation model given application requirements. It should also be pointed out that the set of threats currently included in the TREF framework is not the complete set of threats. Rather, it has been so chosen to include critical threats, and can be extended in the future.

However, while we realize the potential of using threats to compare models, we believe a deeper investigation of these models is necessary. Therefore, in the future, we plan to develop a simulation-based platform that will enable us to simulate and compare the actual behavior of these models in the face of the threats of decentralization.

8. Acknowledgements

The authors thank Justin R. Erenkrantz and Mamadou Diallo for their comments and suggestions. This material is based upon work supported by the National Science Foundation under Grant No. 0524033.

9. References

[1] Abdul-Rahman, A. and Hailes, S. A Distributed Trust Model. In *Proceedings of the New Security Paradigms Workshop.* Langdale, Cumbria UK, 1997.

[2] Abdul-Rahman, A. and Hailes, S. Supporting trust in virtual communities. In *Proceedings of the Hawaii International Conference on System Sciences.* Maui, Hawaii, Jan 4-7, 2000.

[3] Abdul-Rahman, A. *A Framework for Decentralised Trust Reasoning.* Dissertation Thesis. Department of Computer Science, University College London, 2005.

[4] Aberer, K. and Despotovic, Z. Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the Conference on Information and Knowledge Management.* Atlanta, Georgia, November 5-10, 2001.

[5] Blaze, M., Feigenbaum, J., et al. Decentralized Trust Management. In *Proceedings of the IEEE Symposium on Security and Privacy.* p. 164-173, May, 1996.

[6] Brainov, S. Incentive Compatible Trading Mechanism for Trust Revelation. In *Proceedings of the IJCAI Workshop on Economic Agents, Models and Mechanisms.* p. 62-70, Seattle, WA, Aug. 6, 2001.

[7] Buchegger, S. and Boudec, J.-Y.L. Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks. In *Proceedings of the WiOpt 2003: Modeing Optimization in Mobile, Ad Hoc and Wireless Networks.* Sophia-Antipolis, France, March, 2003.

[8] Cahill, V., Gray, E., et al. Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing Mobile and Ubiquitous Computing.* 2(3), p. 52-61, August, 2003.

[9] Coull, S. and Szymanski, B. *A Reputation-based System for the Quarantine of Widespread Malicious Behavior.* Rensselaer Polytechnic Institute, Troy, NY, Technical Report Report 05-01, Oct. 31st, 2005.

[10] Damiani, E., di Vimercati, S.D.C., et al. A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security.* Washington DC, November, 2002.

[11] Dellarocas, C. Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior. In *Proceedings of the 2nd ACM Conference on Electronic Commerce.* Minneapolis, Minnesota, Oct. 17-20, 2000.

[12] Deutsch, M. Cooperation and Trust: Some Theoretical Notes. In *Nebraska Symposium on Motivation*, Jones, M.R. ed. Nebraska University Press, 1962.

[13] Deutsch, M. *The Resolution of Conflict: Constructive and Destructive Processes*. Yale University Press: New Haven, 1973.

[14] Diffie, W. and Hellman, M.E. New Directions In Cryptography. *IEEE Transactions on Information Theory.* 22(6), p. 644-654, November, 1976.

[15] Dingledine, R., Freedman, M., et al. Accountability. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies* 2001.

[16] Dragovic, B., Kotsovinos, E., et al. XenoTrust: Event-based distributed trust management. In *Proceedings of the Second International Workshop on Trust and Privacy in Digital Business.* Prague, Czech Republic, Sep, 2003.

[17] eBay. *www.ebay.com*. <www.ebay.com>.

[18] Gambetta, D. *Trust*. Gambetta, D. ed. Blackwell: Oxford, 1990.

[19] Grandison, T. and Sloman, M. A Survey Of Trust in Internet Applications. *IEEE Communications Surveys.* 3(4), December, 2000.

[20] Gupta, M., Judge, P., et al. A Reputation System for Peer-to-Peer Networks. In *Proceedings of the Thirteenth ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video.* Monterey, California, June 1-3, 2003.

[21] Horne, B., Pinkas, B., et al. Escrow Services and Incentives in Peer-to-Peer Networks. In *Proceedings of the 3rd ACM Conference on Electronic Commerce.* Tampa, FL, USA, Oct. 14-17, 2001.

[22] Josang, A. and Ismail, R. The Beta Reputation System. In *Proceedings of the 15th Bled Electronic Commerce Conference.* Bled, Slovenia, June 17-19, 2002.

[23] Josang, A., Gray, E., et al. Analysing Topologies of Transitive Trust. In *Proceedings of the 1st International Workshop on Formal Aspects in Security and Trust.* Pisa, Sep., 2003.

[24] Josang, A. and Presti, S. Analysing the Relationship Between Risk and Trust. In *Proceedings of the 2nd International Conference on Trust Management.* Oxford, UK, Mar. 29 - Apr. 01, 2004.

[25] Jurca, R. and Faltings, B. An Incentive Compatible Reputation Mechanism. In *Proceedings of the IEEE International Conference on E-Commerce.* Newport Beach, California, USA, June 24-27, 2003.

[26] Kagal, L., Cost, S., et al. A framework for distributed trust management. In *Proceedings of the Second Workshop on Norms and Institutions in MAS, Autonomous Agents.* May, 2001.

[27] Kamvar, S., Schlosser, M., et al. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the WWW.* Budapest, Hungary, May 20-24, 2003.

[28] Kan, G. Gnutella. In *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, Oram, A. ed. p. 94-122, O'Reilly, 2001.

[29] Khare, R. ed. *Weaving a Web of Trust*. World Wide Web Journal. 2(3), O'Reilly & Associates, 1997.

[30] Kinateder, M. and Rothermel, K. Architecture and Algorithms for a Distributed Reputation System. In *Proceedings of the 1st International Conference on Trust Management.* 2692, p. 1-16, Springer-Verlag. Crete, Greece, May 28-30, 2003.

[31] Lai, K., Feldman, M., et al. Incentives for Cooperation in Peer-to-Peer Networks. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems.* Berkeley, CA, Jun. 5-6, 2003.

[32] Lamport, L., Shostak, R., et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems.* 4(3), p. 382-401, July, 1982.

[33] Lee, S., Sherwood, R., et al. Cooperative peer groups in NICE. In *Proceedings of the IEEE Infocom.* San Francisco, USA, April 1-3, 2003.

[34] Li, N., Mitchell, J., et al. Design of a role-based trust management framework. In *Proceedings of the IEEE Symposium on Security and Privacy.* Oakland, California, May, 2002.

[35] Liang, Z. and Shi, W. PET: A PErsonalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing HICSS-38, January, 2005. In *Proceedings of the Hawaii International Conference On System Sciences.* Waikoloa Village, Hawaii, Jan 3-6, 2005.

[36] Manchala, D. E-Commerce Trust Metrics and Models. *IEEE Internet Computing.* 4(2), p. 36-44, Mar-Apr, 2000.

[37] Marsh, S. *Formalising Trust as a Computational Concept.* Thesis. Department of Mathematics and Computer Science, University of Stirling, 1994.

[38] Marti, S. and Garcia-Molina, H. Limited Reputation Sharing in P2P Systems. In *Proceedings of the ACM Conference on Electronic Commerce.* New York, USA, May 17-20, 2004.

[39] Pujol, J., Sanguesa, R., et al. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems.* Bologna, Italy, July 15-19, 2002.

[40] Sabater, J. and Sierra, C. REGRET: A Reputation Model for Gregarious Societies. In *Proceedings of the 4th Workshop on Deception, Fraud and Trust in Agent Societies.* Montreal, Canada, 2001.

[41] Sabater, J. and Sierra, C. Reputation and social network analysis in multi-agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems.* Bologna, Italy, July 15-19, 2002.

[42] Schillo, M., Funk, P., et al. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence Journal, Special Issue on Trust, Deception and Fraud in Agent Societies*. 2000.

[43] Suryanarayana, G., Erenkrantz, J.R., et al. PACE: An Architectural Style for Trust Management in Decentralized Applications. In *Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture.* p. 221-230, Oslo, Norway, June, 2004.

[44] Suryanarayana, G. and Taylor, R.N. *A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications.* UCI Institute for Software Research, Technical Report UCI-ISR-04-6, July, 2004.

[45] Suryanarayana, G., Diallo, M., et al. Architectural Support for Trust Models in Decentralized Applications. To appear in *Proceedings of the 28th International Conference on Software Engineering.* Shanghai, China, May 20-28, 2006.

[46] Tan, Y.-H. and Thoen, W. Toward a Generic Model of Trust for Electronic Commerce. In *Proceedings of the 2nd International Workshop on Deception, Fraud and Trust in Agent Societies.* Seattle, May 1, 1999.

[47] Vigna, G. and Kemmerer, R.A. NetSTAT: A Network-based Intrusion Detection Approach. In *Proceedings of the 14th Annual Computer Security Application Conference.* Scottsdale, AZ, Dec, 1998.

[48] Walsh, K. and Sirer, E.G. *Thwarting P2P Pollution Using Object Reputation.* Department of Computer Science, Cornell University, Report cul.cis/TR2005-1980, 2005.

[49] Xiong, L. and Liu, L. A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities. In *Proceedings of the Fourth ACM Conference on Electronic Commerce.* p. 228-229, San Diego, CA, USA, June 09-12, 2003.

[50] Yao, W. Fidelis: A Policy-Driven Trust Management Framework. In *Proceedings of the First International Conference on Trust Management.* Crete, Greece, May 28-30, 2003.

[51] Yu, B. and Singh, M.P. A social mechanism of reputation management in electronic communities. In *Proceedings of the Fourth International Workshop on Cooperative Information Agents.* p. 154-165, 2000.

[52] Yu, T., Winslett, M., et al. Interoperable strategies in automated trust negotiation. In *Proceedings of the 8th ACM Conference on Computer and Communications Security.* Philadelphia, USA, Nov 5-8, 2001.

[53] Zacharia, G. and Maes, P. Collaborative Reputation Mechanisms in Electronic Marketplaces. In *Proceedings of the 32nd Hawaii International Conference on System Sciences.* Hawaii, 1999.

[54] Zacharia, G. and Maes, P. Trust Management Through Reputation Mechanisms. *Applied Artificial Intelligence.* 14, p. 881-907, 2000.