A Survey of Attack and Defense Techniques for Reputation Systems

KEVIN HOFFMAN, DAVID ZAGE, and CRISTINA NITA-ROTARU

Purdue University

Reputation systems provide mechanisms to produce a metric encapsulating reputation for a given domain for each identity within the system. These systems seek to generate an accurate assessment in the face of various factors including but not limited to unprecedented community size and potentially adversarial environments.

We focus on attacks and defense mechanisms in reputation systems. We present an analysis framework that allows for the general decomposition of existing reputation systems. We classify attacks against reputation systems by identifying which system components and design choices are the targets of attacks. We survey defense mechanisms employed by existing reputation systems. Finally, we analyze several landmark systems in the peer-to-peer domain, characterizing their individual strengths and weaknesses. Our work contributes to understanding (1) which design components of reputation systems are most vulnerable, (2) what are the most appropriate defense mechanisms and (3) how these defense mechanisms can be integrated into existing or future reputation systems to make them resilient to attacks.

Categories and Subject Descriptors: C.0 [**General**]: Systems Specification Methodology; C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g. firewalls)*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms: Design, Reliability, Security, Theory

Additional Key Words and Phrases: Reputation, trust, incentives, peer-to-peer, attacks, collusion, attack mitigation, defense techniques

ACM Reference Format:

Hoffman, K., Zage, D., and Nita-Rotaru, C. 2009. A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. 42, 1, Article 1 (December 2009), 31 pages.
DOI = 10.1145/1592451.1592452, http://doi.acm.org/10.1145/1592451.1592452

1. INTRODUCTION

The rapid growth of communication networks such as the Internet and wireless mesh networks has spurred the development of numerous collaborative applications. Reputation and trust play a pivotal role in such applications by enabling multiple parties to establish relationships that achieve mutual benefit. In general, *reputation* is the

This work is supported by National Science Foundation CyberTrust Award No. 0430271. The views expressed in this research are not endorsed by the National Science Foundation.

Author's address: Department of Computer Science, 305 N. University St., West Lafayette, IN 47907-2107; email: {kjhoffman,zage,cnitarot}@purdue.edu.

DOI 10.1145/1592451.1592452 http://doi.acm.org/10.1145/1592451.1592452

opinion of the public toward a person, a group of people, an organization, or a resource. In the context of collaborative applications such as peer-to-peer systems, reputation represents the opinions nodes in the system have about their peers and peer-provided resources. Reputation allows parties to build *trust*, or the degree to which one party has confidence in another within the context of a given purpose or decision. By harnessing the community knowledge in the form of feedback, reputation-based trust systems help participants decide whom to trust, encourage trustworthy behavior, and deter dishonest participation by providing a means through which reputation and ultimately trust can be quantified and disseminated [Resnick et al. 2000]. Without such mechanisms, opportunism can erode the foundations of these collaborative applications and lead to peer mistrust and eventual system failure [Akerlof 1970].

A rich variety of environments and applications has motivated research in reputation systems. Within the context of peer-to-peer eCommerce interactions such as eBay, Amazon, uBid, and Yahoo, recent research has shown that reputation systems facilitate fraud avoidance and better buyer satisfaction [Houser and Wooders 2006; Resnick et al. 2006; Xiong and Liu 2002, 2003; Lin et al. 2005]. Not only do reputation systems help protect the buyer, but they have also been shown to reduce transaction-specific risks and therefore generate price premiums for reputable sellers [Ba and Pavlou 2002]. More recently, reputation systems have been proposed as a means to filter out inauthentic content (pollution) for file-sharing applications [Walsh and Sirer 2006], a method for selecting usable network resources [Aringhieri et al. 2006], a means to identify high-quality contributions to Wikipedia [Adler and de Alfaro 2007], and a way to punish [Adar and Huberman 2000] or prevent [Ham and Agha 2005; Piatek et al. 2007] free-riders in content dissemination networks.

The success of a reputation system is measured by how accurately the calculated reputations predict the quality of future interactions. This is difficult to achieve in an environment where any party can attempt to exploit the system to its own benefit. Some attacks have a narrow focus and only affect the reputation of the misbehaving identity or a few selected targets. Other attacks have a much broader influence, affecting large percentages of the identities within the system. Centralized or implicitly trusted elements of the reputation system are more prone to attack due to their identifiability and key role in the functioning of the system. The impact of attacks against reputation systems reaches beyond just the manipulation of virtual numbers, but turns into dollars fraudulently lost and ruined business reputations [Khopkar et al. 2005].

This article is the first survey focusing on the characterization of reputation systems and threats facing them from a computer science perspective. Our work contributes to understanding which reputation system design components are vulnerable, what are the most appropriate defense mechanisms, and how these defense mechanisms can be integrated into existing or future reputation systems to make them resilient to attacks. Specifically:

(1) We propose an analytical framework by which reputation systems can be decomposed, analyzed, and compared using a common set of metrics. This framework facilitates insights into the strengths and weaknesses of different systems and comparisons within a unified framework.

(2) We classify attacks against reputation systems, analyzing what system components are exploited by each attack category. We elucidate the relevance of these attacks by providing specific examples based on real systems.

(3) We characterize existing defense mechanisms for reputation systems, discussing their applicability to different system components and their effectiveness at mitigating the identified attacks.

(4) We analyze each system based on our analytical framework, drawing new insights into reputation system design. We also discuss each system's strengths and weaknesses based on our attack classification and defense characterization.

Roadmap: The rest of the article is organized as follows. We discuss related work in Section 2 and characterize the fundamental dimensions of reputation systems in Section 3. We describe attacks against reputation systems components and defense strategies in Sections 4 and 5, respectively. We analyze several well-known reputation systems in Section 6. Finally, we present concluding remarks in Section 7.

2. RELATED WORK

Previous research in the area has presented an overview of the design issues of reputation systems in peer-to-peer networks [Marti and Garcia-Molina 2006], surveyed the broader issue of trust management [Suryanarayana and Taylor 2004], provided an overview of deployed reputation systems [Jøsang et al. 2007], and surveyed a subset of attacks on and theoretical defense techniques for reputation systems [Friedman et al. 2007].

A method to categorize peer-to-peer reputation systems was presented in Marti and Garcia-Molina [2006]. Their work serves as an introduction to reputation systems and design issues relating to their use in peer-to-peer applications. Unlike their work, we decompose systems along three dimensions, to provide further insight into how implementation issues affect the effectiveness of the system. Additionally, we contribute a classification of attack strategies and survey of known defense techniques and their strengths and weaknesses.

Suryanarayana and Taylor [2004] surveyed trust management in the context of peer-to-peer applications. The scope of their survey was broader and included trust management systems not based on reputation. The authors analyzed eight reputation-based trust management systems with respect to five types of threats and eleven different characteristics. In contrast, in this survey we focus solely on reputation systems, allowing us to define an analysis framework and attack classification specific to reputation systems, and to more comprehensively survey the reputation system literature.

Jøsang et al. [2007] focused on surveying the calculation mechanisms and gave greater emphasis to discussing deployed systems rather than directly surveying the research literature. Their survey was presented in the context of a broader discussion of the meaning of trust and reputation. Our work presents a broader analysis framework for reputation systems and also focuses more on the analysis of attacks against reputation systems.

Friedman et al. [2007] provided an overview of reputation systems for environments with strategic users and provide theoretical solutions to three specific threats: whitewashing, fake feedback, and dishonest feedback. In our work, we examine a broader range of attacks, analyze how each of these attacks affect different design choices in the reputation systems, and review a myriad of possible solutions for each attack, including solutions which are feasible in distributed environments.

A large corpus of work on reputation systems exists in the management, economic, and behavioral literature. Our work does not cover all these areas, but focuses on the technical elements of peer-to-peer systems and resilience to network-centric attack rather than the social and economic elements. Dellarocas [2003] considered the role of reputation systems, their relation to more traditional methods of reputation assessment, their social and economic impact, and how they can be understood in the context of game theory and economics. The work gave insights into why reputation systems do

or do not work from a human perspective and presents how insights from management science, sociology, psychology, economics, and game theory must be considered beyond computer science when designing new reputation systems. Our work is complementary: whereas Dellarocas [2005] provided insight into the broader factors affecting the operational environments of reputation systems more from a management perspective, we consider the perspective of a reputation system builder and therein provide insight into the composition of the reputation system itself and also a characterization of corresponding threats and defense mechanisms.

3. ANALYSIS FRAMEWORK

Due to their common purpose, reputation systems naturally share similar structural patterns. Understanding these similarities and developing an analysis framework serves a twofold purpose. First, it provides greater insight into prior research, facilitating common ground comparison between different systems. Second, it provides insights into the fundamental strengths and weaknesses of certain design choices, contributing to the future design of attack-resilient reputation systems.

We identify the following three dimensions as being fundamental to any reputation system:

—*Formulation*. The ideal mathematical underpinnings of the reputation metric and the sources of input to that formulation. For example, a system may accept positive and negative feedback information, weighted as $+1$ and -1, and define an identity's reputation to be the summation of all of its corresponding feedback.

—*Calculation*. The algorithm to calculate the mathematical formulation for a given set of constraints (physical distribution of participants, type of communication substrate, etc.). For example, the algorithm to calculate the formulation could specify that a random set of peers is queried and the feedback received for each identity tallied.

—*Dissemination*. The mechanism that allows system participants to obtain the reputation metrics resultant from the calculation. Such a mechanism may involve storing the values and disseminating them to the participants. For example, a system might choose to use a distributed hash table to store the calculated reputation values and a gossip protocol to distribute new information.

Figure 1 presents the general structure of a reputation system, including the location of each of the fundamental dimensions. Figure 2 further demonstrates how each dimension of the reputation system can be comprised of different components, determining the unique properties of each system. The overarching goal of a reputation system is to produce a metric encapsulating reputation for a given domain for each identity within the system. Each system receives input from various types of sources. Based on this input, a system produces a reputation metric through the use of a calculation algorithm. Once calculated, reputation metric values are then disseminated throughout the system in advance or on demand as the metric values are requested. Finally, higher-level systems or users can then utilize these reputation metric values in their decision making processes for penalties or rewards in order to achieve the goals of the user application.

3.1. Formulation

Formulation of a reputation system is the abstract mathematical specification of how the available information should be transformed into a usable metric. This specification may be made through an explicit equation, or implicitly through describing an

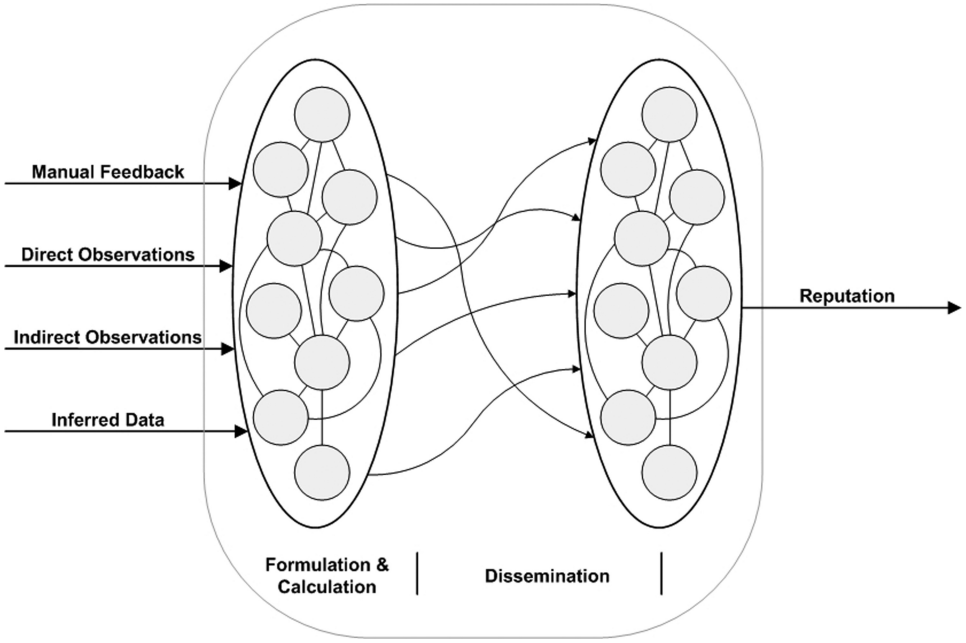

Fig. 1. Depiction of how a reputation system operates. The large ovals represent the reputation system itself, normally consisting of many different computers acting as a distributed system.

algorithm that will result in the correct values. The formulation determines the theoretical properties of the system and thus the upper bound on its resilience to attacks. As a result, the formulation is a critical component since any weakness in the design of the formulation allows malicious manipulation of the metric values. We identify and discuss three important components of the formulation: the source of the information, the type of information, and the reputation metric.

3.1.1. Source of Information. A core component of the formulation of reputation is the source of the raw information used as inputs to the algorithm. The source can either be a manual or an automatic source.

Manual sources are obtained from human feedback, usually in the form of user ratings of other identities based on the results of a single transaction such as the feedback within eBay [Houser and Wooders 2006], a specific time period [Singh and Liu 2003], or arbitrary feedback [Zhou and Hwang 2007]. Since these sources are naturally qualitative, the formulation component must specify some method of converting the qualitative metric into a quantitative one. For example, a user may feel satisfied with the ultimate outcome of a transaction, but be dissatisfied with the timeliness of it. The formulation specifies how the user can convert this qualitative information into quantitative information, such as by giving them the choice of giving a negative, neutral, or positive rating [Houser and Wooders 2006]. Other proposals include the use of Bayesian procedures [Aringhieri et al. 2006; Buchegger and Le Boudec 2004] or fuzzy decision logic [Song et al. 2005] to transform the user feedback into ratio-scaled variables. The formulation may also allow the user to tag the quantitative metric with qualitative information under the intention of aggregating this information with the reputation metric for later human consumption [Houser and Wooders 2006].

Automatic sources are obtained automatically either via direct or indirect observation. Direct, automatic sources of information result from data directly observed by

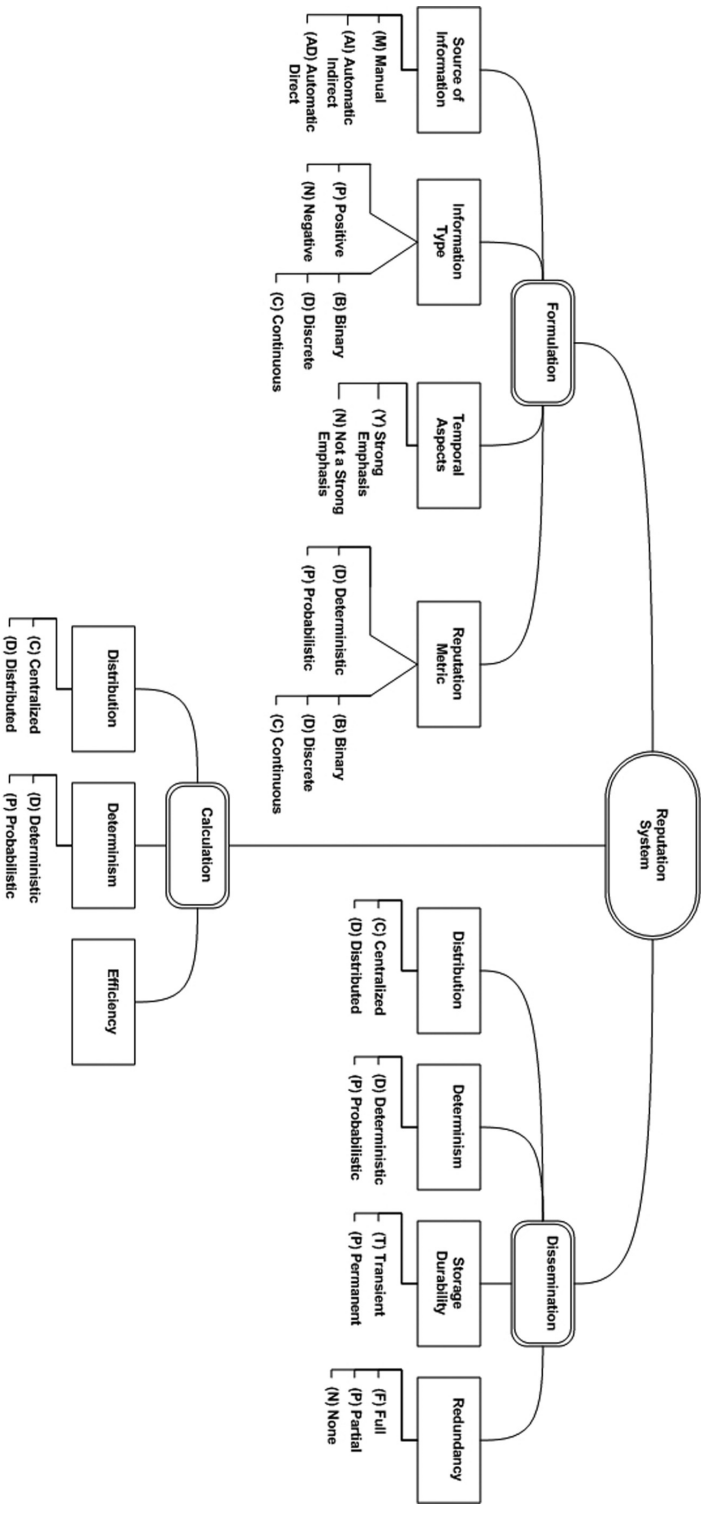

Fig. 2. Visualization of analysis framework for reputation systems.

an identity, such as the success or failure of an interaction, the direct observations of cheating, or the measurement of resource utilization by neighbors in a peer-to-peer network. Information that is obtained second-hand or is inferred from first-hand information is classified as an indirect, automatic source. Indirect, automatic input sources are relevant in many modern reputation systems which are developed to have a notion of the transitivity of trust. Nodes share information in order to combat the sparsity of first-hand information [Xiong et al. 2005] or to further refine the reputation metric [Marti and Garcia-Molina 2004; Xiong and Liu 2003]. Also, indirect sources of information are important in systems such as SuperTrust [Dimitriou et al. 2007], in which the outputs of one tier are used as inputs to the next higher tier.

3.1.2. Information Type. Another component of the formulation is whether the source information includes positive (trust building) events, negative (trust diminishing) events, or both. This design choice fundamentally influences the applications for which a given reputation system is most effective as well as determining the classes of attacks that are relevant to the system. For example, a system that only considers positive events will be immune to attacks where malicious identities try to falsely degrade others' reputations [Marti and Garcia-Molina 2004]. While it may seem beneficial to only consider one type of information in order to limit the possible attacks on the system, this also limits the flexibility of the system as well as the ability for honest peers to counteract the attacks that are still relevant [Guha et al. 2004]. Continuing the example above, honest participants would be unable to give negative feedback regarding those identities that are falsely promoting themselves.

3.1.3. Reputation Metric. The most important component of the formulation dimension is the mathematical or algorithmic representation of the reputation. The reputation metric can be classified as either binary, discrete, or continuous. A binary representation of trust converts the qualitative notion of reputable versus nonreputable into a numerical format and is utilized by systems like the ones proposed by Xiong and Liu [2003] and Guha et al. [2004]. Some systems, such as Scrivener [Nandi et al. 2005] and XRep [Damiani et al. 2002], utilize discrete metrics which have various predefined levels of reputability and allow for a more flexible application of the reputation information since different actions can correspond to different levels of reputability. Finally, a metric can be represented as a continuous variable, such as is done in many of the newer systems [Aringhieri et al. 2006; Walsh and Sirer 2006; Li and Wu 2007]. Continuous variables are often the easiest representation to compute since most formulations result in real number results.

Certain systems, such as PeerTrust [Xiong and Liu 2003] and EigenTrust [Kamvar et al. 2003], will choose to convert a continuous metric into a binary metric via heuristics or statistical measures, since it is often easier for users to base their decisions on a metric with predefined intervals. This is especially true if the continuous metric is not a linear representation of reputation [Kamvar et al. 2003].

Many systems consider the change of reputation over time, trying to balance the tradeoff between resiliency to attacks versus the acceptance of new or previously misbehaving identities. While it may seem that greater resiliency is more desirable than the easy acceptance of new identities, this may hamper overall user satisfaction and system utility as well as render systems deployed in less stable environments from functioning effectively [Morselli et al. 2004]. For example, if one input to the formulation is whether a peer is forwarding data correctly, even honest peers may be seen as having a low reputation and be denied service due to transient network conditions [Ham and Agha 2005].

Along with the domain and granularity of the reputation metric, such metrics can also be classified as symmetric or asymmetric [Cheng and Friedman 2005; Altman and Tennenholtz 2005a, 2006; Friedman et al. 2007]. If a reputation system utilizes a symmetric reputation metric, such as the one used in the PageRank [Cheng and Friedman 2006] algorithm, there is one, global, reputation value for each node in the system. In the case of systems using an asymmetric reputation metric, each node has an individual view of the overall reputation in the network. Systems such as EigenTrust [Kamvar et al. 2003] and PowerTrust [Zhou and Hwang 2007] utilize asymmetric reputation metrics in their formulation. Not only is the distinction between symmetric or asymmetric interesting for classifying systems, but, as we discuss in Section 5.1, it also plays an important part in reasoning about the attack resistance of reputation systems.

3.2. Calculation

As depicted in Figure 1, the calculation dimension is the concrete part of the reputation system that receives input information and produces the reputation metric values. While the formulation is an idealized method for determining a reputation value, the calculation dimension characterizes how the formulation is implemented within the constraints of a particular reputation system. This dimension strives to be accurate to the reputation metric formulation while remaining practical to implement and resilient to malicious attack. We identify two components relevant to the reputation calculation: the calculation structure (centralized or distributed) and calculation approach (deterministic or probabilistic).

Note. At first glance, it may seem that the calculation dimension is a direct result of the formulation. However, the physical constraints of the system may make the mapping between the formulation and calculation dimensions nontrivial. For example, the EigenTrust algorithm [Kamvar et al. 2003] can be represented as the centralized computation of the left eigenvector of a matrix of trust values, but to calculate this in a scalable fashion, the formulation had to be decomposed into an efficient distributed algorithm. Another factor causing this mapping to be nontrivial is the need to be resilient to malicious manipulation of values during the actual calculation. Even assuming that all source information is completely accurate, a malicious participant can try to manipulate the values during the calculation stage. If the system does not account for this possibility, reputation values may be manipulated without detection.

3.2.1. Calculation Structure. The reputation system can be structured to calculate the reputation metric via a centralized authority or across multiple distributed participants. A centralized authority often leads to a simple solution with less potential for manipulation by malicious outsiders. Many eCommerce businesses such as eBay have successfully deployed centralized reputation systems which allow for the long-term storage and internal auditing of all reputation data [Houser and Wooders 2006]. However, a centralized approach relies on the assumption that the system participants completely trust the centralized authority which in turn must be correct and always available. If the centralized authority is not carefully designed, it can become a single point of failure for the entire system [Lee et al. 2003]. Centralized systems are especially susceptible to attacks on availability, which are discussed further in Section 4.2.5. In addition, such an approach suffers from the lack of scalability, especially if the formulation is complex or the information is obtained from a wide range of possibly high-latency sources.

In the open environment of most modern peer-to-peer applications, peers do not have a centralized authority or repository for maintaining or distributing reputation.

Instead, most reputation systems calculate global reputation in a fully distributed manner [Kamvar et al. 2003; Buchegger and Le Boudec 2004; Walsh and Sirer 2006; Li and Wu 2007]. Although these distributed calculations are inherently more complex, they scale well [Feldman et al. 2004], avoid single points of failure in the system [Lee et al. 2003], and balance load across multiple nodes [Marti and Garcia-Molina 2004]. Such designs must ensure that participants converge upon a usable solution as well as prevent malicious manipulation from degrading the performance of the entire system. The complexity of data and entity authentication in systems lacking a centralized authority and the reliance on multiple system participants provide opportunities for attackers to subvert the reputation calculation.

3.2.2. Calculation Approach. Reputation systems implement calculation by using either deterministic or probabilistic approaches. The output of a deterministic calculation can be determined solely from knowledge of the input, with very precise meaning often attached to this output. Deterministic calculations for global reputation values are often only practical for centralized calculations, unless the scope of the formulation is narrow, identities only incorporate feedback for a small subset of peers, or the total size of the system is small. Additionally, a deterministic calculation can be used in systems where each individual node calculates its own view of other nodes' reputation values and there is not a single global reputation for each node [Marti and Garcia-Molina 2004].

Probabilistic approaches, often known as *randomized algorithms*, were proposed to address some of the limitations posed by deterministic calculations. These algorithms rely on sources of randomness during the calculation process, causing their output to be predictable only within certain error bounds.

It is interesting to note that even when formulations are deterministic and thus would seem to imply a deterministic calculation, the actual calculation may have to be implemented probabilistically. For example, the EigenTrust [Kamvar et al. 2003] formulation represents reputation as the eigenvalues of a matrix (which is a deterministic formulation), but the distributed calculation is probabilistic in order for the algorithm to scale. Robust randomized formulations rely on statistical mechanisms, such as Markov models [Kamvar et al. 2003] and Bayesian models [Buchegger and Le Boudec 2004], which attach error bounds to and give meaning to the randomized algorithm.

3.3. Dissemination

Once reputation has been calculated, it needs to be readily accessible to interested parties while remaining resilient to alteration. Calculated values must be efficiently disseminated to other recipients or made available upon request. These responsibilities of a reputation system fall within the dissemination dimension. Although calculation and dissemination are often intertwined in the implementation, it is useful to separate them for analysis purposes. We discuss the following four aspects of the dissemination dimension: the dissemination structure, dissemination approach, storage strategies, and dissemination redundancy.

3.3.1. Dissemination Structure. Centralized dissemination mechanisms involve a central authority storing and disseminating calculated values. The central authority may actually be implemented via clusters of computers, but this remains classified as centralized since the entire structure is controlled exclusively by one entity. For example, in order for eBay to scale, it must be implemented by high-availability clusters. However, logically eBay utilizes a centralized authority to disseminate the calculated reputation

information. In a centralized dissemination, the central authority has greater power to protect the integrity of the process, but then also it becomes a single point of weakness—if the central authority is fully or partially compromised due to external attackers, insiders, or intentional misconduct, the damage to the reputation system is much higher than if the process was distributed.

In a distributed dissemination, each participant is responsible for some portion of the calculated reputation values. The distribution of responsibility may be symmetrical (e.g., distributed hash tables (DHTs) [Ratnasamy et al. 2000; Stoica et al. 2001; Zhao et al. 2001; Rowstron and Druschel 2001]) or asymmetrical (e.g., power-law peer-to-peer networks [Dimitriou et al. 2007; Matei et al. 2002]). Distributed dissemination is inherently more vulnerable to manipulation, and often employs data redundancy, cryptographic mechanisms, or other measures to preserve metric integrity. Distributed mechanisms are also more difficult to implement and test properly, and may thus be more vulnerable to exploitation [Dahan and Sato 2007].

3.3.2. Dissemination Approach. The communication pattern of dissemination mechanisms can be characterized as either deterministic or probabilistic. Examples of deterministic communication mechanisms include distribution hierarchies such as those in SuperTrust [Dimitriou et al. 2007] and DHTs such as those employed by EigenTrust [Kamvar et al. 2003], PowerTrust [Zhou and Hwang 2007], and PeerTrust [Xiong and Liu 2003]. Probabilistic communication techniques include epidemic-based dissemination techniques such as probabilistic broadcast [Eugster et al. 2001; Walsh and Sirer 2006] and flooding [Marti and Garcia-Molina 2004; Lee et al. 2003].

3.3.3. Storage Durability. Transient storage is defined to be any nondurable, random access memory, whereas permanent storage is any storage in which data is preserved even during periods without power. Depending on the volatility of the system components and the computational complexity of the calculation, it may be beneficial to store calculated reputation values in permanent storage for retrieval later. Systems such as PowerTrust [Zhou and Hwang 2007] and TrustMe [Singh and Liu 2003] include long-term temporal information in their formulations and require permanent storage in order to be resilient to failures. PowerTrust relies on the ability to migrate data in a DHT to preserve historical data while TrustMe provides anonymous storage and migration protocols. On the other end of the spectrum, systems such as ARA [Ham and Agha 2005] calculate reputation values based on a small subset of recent transactions, in which case long-term global storage is unnecessary.

Whether or not a system uses permanent storage is more of an implementation issue than a core component of a reputation system. Permanent storage may be required by the calculation mechanisms to detect slow changes in behavior over long periods of time. Also, permanent storage must be guarded against malicious manipulation, physical data corruption, and data loss [Singh and Liu 2003].

3.3.4. Dissemination Redundancy. The degree of redundancy built into the dissemination mechanisms is a tradeoff between resiliency to manipulation and storage/communication efficiency. Redundancy can be employed in many of the components of the dissemination dimension, such as having redundant messaging in communications patterns or duplicate backups of stored values. For example, the TrustMe [Singh and Liu 2003] system assumes a copy of the reputation values are stored in several places. Many of the the modern reputation systems use messaging protocols with redundant messages to help ensure message delivery and provide some resiliency to malicious nodes at the cost of increased communication overhead. Each node requests

and receives multiple copies of a reputation value (from different storage locations), which increases the link stress of the network when viewed in comparison to traditional unicast messaging from a single storage location. Less efficient methods of implementing redundancy (e.g., complete duplication of data) are often favored over more theoretically desirable methods, such as Reed-Solomon codes [Reed and Solomon 1960], as these methods are often easier to implement. Finally, systems differ in how they resolve redundancy to produce a final reputation metric. Possibilities include but are not limited to leaving the redundancy unresolved and presenting the user with the raw information, majority voting [Kamvar et al. 2003], and using weighted averages [Li and Wu 2007].

4. ATTACKS ON REPUTATION SYSTEMS

In this section, we discuss attacks against reputation systems. We first state our assumptions about attackers and then discuss five separate classes of attack scenarios. We highlight both the attacks mechanisms and the system components that are exploited during each of the attacks.

4.1. Attacker Model

Several characteristics determine the capability of the attacker. They include the location of the attacker in relation to the system (insider vs. outsider), if the attacker acts alone or as part of a coalition of attackers, and whether the attacker is active or passive.

The open nature of reputation systems and their accompanying peer-to-peer applications lead us to assume all attackers are insiders. Insiders are those entities who have legitimate access to the system and can participate according to the system specifications (i.e., authenticated entities within the system), while an outsider is any unauthorized or illegitimate entity in the system who may or may not be identifiable. While reputation systems often employ some form of authentication to prevent unauthorized access, an attacker can obtain multiple identities, also known as the *Sybil attack* [Douceur 2002]. In addition, since reputation systems push trust to the fringes of the Internet where end nodes are more likely to be compromised [Survey 2005], they are more vulnerable to insider attacks.

We assume that attackers are motivated either by selfish or malicious intent. Selfish (or rational) attackers manipulate reputation values for their own benefit, while malicious attackers attempt to degrade the reputations of others or impact the availability of the reputation system itself.

In general, attackers can either work alone or in coalitions. Although both scenarios are possible and relevant with respect to reputation systems, we are primarily concerned with attacks caused by coalitions of possibly coordinated attackers. Coordinated attacks are more difficult to detect and defend against because attackers can exhibit multi-faceted behavior that allows them to partially hide within their malicious coalition. We note that, while coordinated attacks caused by multiple Sybil identities are a danger to any distributed system, even without such attacks, reputations systems are vulnerable to malicious collectives of nodes.

We consider all attacks to be active since any form of attack on the reputation system requires interaction with the system, such as injecting false information, modifying entrusted information, refusing to forward information, deviating from the algorithmic processes, or actively attempting to subvert the availability of the system.

4.2. Attack Classification

We classify attacks against reputation systems based on the goals of the reputation systems targeted by attacks. The goal of a reputation system is to ensure that the reputation metrics correctly reflect the actions taken by participants in the system and cannot be maliciously manipulated. This is not achieved if participants can falsely improve their own reputation or degrade the reputations of others. As a result of the attacks, misbehaving participants can obtain unwarranted service or honest participants can be prevented from obtaining service. Besides targeting the accuracy of the reputation system, malicious participants can target the availability of the system itself.

We identify several classes of attacks:

—*Self-promoting*. Attackers manipulate their own reputation by falsely increasing it.

—*Whitewashing*. Attackers escape the consequence of abusing the system by using some system vulnerability to repair their reputation. Once they restore their reputation, the attackers can continue the malicious behavior.

—*Slandering*. Attackers manipulate the reputation of other nodes by reporting false data to lower the reputation of the victim nodes.

—*Orchestrated*. Attackers orchestrate their efforts and employ several of the above strategies.

—*Denial of service*. Attackers cause denial of service by preventing the calculation and dissemination of reputation values.

Below, we discuss in detail the attack mechanisms, identifying the reputation system components that are exploited during the attack.

4.2.1. Self-Promoting. In self-promoting attacks, attackers seek to falsely augment their own reputation. Such attacks are only possible in systems that consider positive feedback in the formulation. Fundamentally, this is an attack against the formulation, but attackers may also exploit weaknesses in the calculation or dissemination dimensions to falsely increase reputation metric values.

Self-promotion attacks can be performed by a lone identity or organized in groups of collaborating identities. One very basic form of the attack occurs when an attacker fabricates fake positive feedback about itself or modifies its own reputation during the dissemination. Systems lacking mechanisms to provide data authentication and integrity are vulnerable to such attacks as they are not able to discern between fabricated and legitimate feedbacks.

However, even if source data is authenticated using cryptographic mechanisms, self-promotion attacks are possible if disparate identities or a single physical identity acquiring multiple identities through a Sybil attack [Douceur 2002] collude to promote each other. Systems that do not require participants to provide proof of interactions which result in positive reputations are particularly vulnerable to this attack. To perform the attack, colluding identities mutually participate in events that generate real feedback, resulting in high volumes of positive feedback for the colluding participants. Because the colluders are synthesizing events that produce verifiable feedback at a collective rate faster than the average, they are able to improve their reputations faster than honest participants or counter the effects of possible negative feedback. Such patterns of attack have been observed in the Maze file sharing system [Lian et al. 2007]. Attackers that are also interacting with other identities in honest ways are known as *moles* [Feldman et al. 2004]. Colluding attackers can also contribute further to the self-promotion of each other by manipulating the computation dimension when aggregating reputation values.

Techniques to mitigate self-promoting attacks include requiring reputation systems to provide accountability, proof of successful transactions, and the ability to limit or prevent an attacker from obtaining multiple identities. The computation dimension should also include mechanisms to prevent colluding adversaries from subverting the computation and storage of the reputation values. Complementarily, the impact of the attacks can be decreased by detecting and reacting to groups of colluders that interact almost exclusively with each other. However, finding these colluders, which can be formulated as finding a clique of a certain size within a graph, is known to be NP-complete and only heuristic-based solutions have been proposed so far [Cormen et al. 2001].

4.2.2. Whitewashing. Whitewashing attacks occur when attackers abuse the system for short-term gains by letting their reputation degrade and then escape the consequences of abusing the system by using some system vulnerability to repair their reputation. Often attackers will attempt to reenter the system with a new identity and a fresh reputation [Lai et al. 2003]. The attack is facilitated by the availability of cheap pseudonyms and the fact that reciprocity is much harder to maintain with easily changed identifiers [Friedman and Resnick 2001].

This attack fundamentally targets the reputation system's formulation. Formulations that are based *exclusively on negative* feedback are especially vulnerable to this type of behavior since newcomers have equal reputation metric values to participants which have shown good long-term behavior. Separately, a reputation system using either type of feedback (positive and/or negative) is vulnerable if the formulation relies exclusively on long-term history without discriminating between old and recent actions. If an attacker is able to generate a beneficial reputation based solely on history, it can perform short duration malicious attacks with little risk of negative consequences as the previous history will heavily outweigh current actions. This can have a large impact on the system as the malicious node will continue to have a high reputation for a substantial period of time during which the system is slow to identify the malicious behavior and unable to sufficiently lower the reputation of the malicious node. In systems with formulations that include positive feedback, attackers may have to behave honestly for an initial period of time to build up a positive reputation before starting the self-serving attack. Attackers that follow this pattern are also known as *traitors* [Marti and Garcia-Molina 2006].

Whitewashing attacks may be combined with other types of attacks to make each attack more effective. For example, in systems with both positive and negative feedback, concurrently executing a self-promoting attack will lengthen the duration of effectiveness of a whitewashing attack. Likewise, whitewashing identities may slander those identities that give negative feedback about the attacker so that their negative feedback will appear less reputable since many systems weight the opinions of an identity by its current level of trustworthiness. In this case, slandering minimizes the amount of whitewashing an attacker must perform to maintain a good reputation.

Mitigating whitewashing attacks requires reputation systems to use a formulation that does not result in the same reputation for both newcomers and participants that have shown good behavior for a long time, that takes into account limited history, and that limits users from quickly switching identities or obtaining multiple identities.

4.2.3. Slandering. In *slandering attacks*, one or more identities falsely produce negative feedback about other identities. As with self-promoting attacks, systems that do not authenticate the origin of the feedback are extremely vulnerable to slander. In general, these attacks target the formulation dimension of a reputation system.

The attack can be conducted both by a single attacker and a coalition of attackers. As typically the effect of a single slandering node is small, especially if the system limits the rate at which valid negative feedback can be produced, slandering attacks primarily involve collusion between several identities. Depending on the application of the system, slandering attacks may be more or less severe than self-promotion attacks. For example, in high-value monetary systems, the presence of even small amounts of negative feedback may severely harm an identity's reputation and ability to conduct business [Ba and Pavlou 2002].

The lack of authentication and high sensitivity of the formulation to negative feedback are the main factors that facilitate slandering attacks. Reputation systems must consider the inherent tradeoffs in the sensitivity of the formulation to negative feedback. If the sensitivity is lower, then the formulation is robust against malicious collectives falsely slandering a single entity, but it allows entities to exhibit bad behavior for a longer time, for the same decrease in reputation. On the other hand, if sensitivity is higher, the bad behavior of a single identity can be punished quickly, but honest identities are more susceptible to attacks from malicious collectives. If malicious collectives are well behaved except to slander a single identity it may be difficult to distinguish that slander from the scenario where the single identity actually deserved the bad feedback that was received.

Defense techniques to prevent false feedback include employing stricter feedback authentication mechanisms, validating input to make sure that feedback is actually tied to some transaction, and incorporating methods to limit the number of identities malicious nodes can assume.

Systems may also limit the impact of slandering attacks by using formulations that compute reputations based exclusively on direct information. However, this is not possible in reputation systems with sparse interaction, where trust inference is needed [Xiong et al. 2005]. In such systems, the trust inference mechanisms must be robust to malicious attacks.

4.2.4. Orchestrated. Unlike the previously described attacks that employ primarily one strategy, in *orchestrated* attacks, colluders follow a multifaced, coordinated attack. These attacks utilize multiple strategies, where attackers employ different attack vectors, change their behavior over time, and divide up identities to target. While orchestrated attacks fundamentally target a system's formulation, these attacks also may target the calculation and dissemination dimensions. If the colluding attackers become a significant part of the calculation or dissemination of reputation within an area of the system, they can potentially alter reputation metric values to their benefit.

One example of an orchestrated attack, known as an *oscillation attack* [Srivatsa et al. 2005], is where colluders divide themselves into teams and each team plays a different role at different times. At one point in time, some teams will exhibit honest behavior while the other teams exhibit dishonest behavior. The honest teams serve to build their own reputations as well as decrease the speed of decline of the reputation of the dishonest teams. The dishonest teams attempt to gain the benefits of dishonest behavior for as long as possible, until their reputation is too low to obtain benefit from the system. At this point, the roles of the teams switch, so that the dishonest teams can rebuild their reputation and the previously honest teams can begin exhibiting dishonest behavior. Even more complex scenarios are possible where there are more than two roles. For example, one team of nodes may self-promote, another may slander benign nodes, and the final team may misbehave in the context of the peer-to-peer system, such as dropping maintenance messages used to maintain the system structure and connectivity.

Orchestrated attacks are most effective when there are several colluders for each role. Larger numbers allow each colluder to be linked less tightly to other colluders, which makes detection much more difficult. Colluders performing orchestrated attacks balance between maximizing selfish or malicious behavior and avoiding detection. Robust formulations increase the number of colluders that must participate in order to achieve the desired effect.

Identifying orchestrated attacks is difficult since instead of trying to identify cliques in a graph representing identities and their relationships, systems need to identify partially connected clusters where each colluder may not appear to be connected to every other colluder due to the differing behaviors of the different roles in the orchestrated strategy. Within a window of time, it is possible that two colluders have no direct interaction observable by the system and thus appear completely separated, while they are actually colluding indirectly. For example, the two colluders may produce negative feedback against identities that gave negative feedback against another, different colluder.

4.2.5. Denial of Service. Finally, attackers may seek to subvert the mechanisms underlying the reputation system itself, causing a denial of service. Such attacks are conducted by malicious nonrational attackers, making them difficult to defend against. Systems using centralized approaches and lacking any type of redundancy are typically vulnerable to denial of service attacks. Attackers can attempt to cause the central entity to become overloaded (e.g., by overloading its network or computational resources). These attacks target the calculation and dissemination dimensions of a system and are performed by groups of colluding attackers.

Preventing a reputation system from operating properly with a denial of service attack may be as attractive to attackers as corrupting the reputation values, especially if the application employing the reputation system is automated and needs to make decisions in a timely fashion. For example, consider a peer-to-peer data dissemination application where data is routed along the most trustworthy paths. If the reputation system is inoperable, the system relying on reputation may need to continue to route data even if reputation information is unavailable, allowing malicious identities to participate for periods of time without their negative reputations being known (or without being punished for their negative behavior).

Distributed calculation and dissemination algorithms are often less vulnerable to attacks if enough redundancy is employed such that misbehavior or loss of a few participants will not affect the operation of the system as a whole. However, some distributed storage components such as DHTs may have their own vulnerabilities [Dahan and Sato 2007] and they can be in turn exploited by an attacker to create denial of service against the reputation system.

5. DEFENSE STRATEGIES

In this section, we survey the defense mechanisms employed by existing reputation systems. Although none of the existing systems provide defenses against all the attacks presented in Section 4, many of them use techniques to address attacks conducted by selfish rational attackers and a limited number of attacks conducted by coalitions of malicious attackers. These techniques can be grouped around several major design characteristics that facilitate the attacks. We discuss mechanisms to defend against attackers acquiring multiple identities in Section 5.1. We discuss techniques to ensure that direct observations reflect reality in Section 5.2 and techniques to defend against generation and propagation of false rumors in Section 5.3. The

major reason behind whitewashing attacks is the fact that systems do not distinguish newcomers from participants that have demonstrated good behavior over time. We discuss techniques to address this issue in Section 5.4. Finally, we review techniques used by reputation systems to address more general denial of service attacks in Section 5.5.

5.1. Preventing Multiple Identities (Sybil Attacks)

The problem of obtaining multiple identities received significant attention in recent years as it impacts not only reputation systems but peer-to-peer systems in general. In online environments where new identities may be created with minimal cost, malicious entities may acquire multiple identities for the sole purpose of creating phantom feedback in the system [Friedman et al. 2007]. Proposed solutions to deal with Sybil attacks fall into centralized and decentralized approaches.

In a centralized approach, a central authority issues and verifies credentials unique to each entity. To increase the cost of obtaining multiple identities, the central authority may require monetary or computational payment for each identity. Although this may limit the number of identities an attacker can obtain, there are scenarios in which it may not be possible or practical to have a centralized authority. Additionally, the central authority represents a single point of failure for the system and can itself be subjected to attacks.

Decentralized approaches do not rely on a central entity to issue certificates for identities. Some solutions proposed include binding a "unique" identifier such as IP addresses to public keys [Douceur 2002] or using network coordinates to detect nodes with multiple identities [Bazzi and Konjevod 2005]. However, IP addresses can be spoofed and network coordinates can be manipulated by attackers [Zage and Nita-Rotaru 2007]. Other solutions, such as those proposed in BBK [Beth et al. 1994], PGP [Zimmermann 1995], and Advogato [Levien 2003] take advantage of social knowledge to propagate reputation originating from trusted sources along the edges of a directed graph, creating a "web of trust." This type of solution limits the effect of malicious nodes by limiting the number of unknown, potentially malicious nodes honest nodes will extend trust to, thus limiting the effect of the attackers at the expense of requiring social interaction. More recently, social networks were proposed to detect attackers posing under multiple identities. The approach in Yu et al. [2006] creates a graph in which nodes represent identities and edges represent trust relations. The protocol ensures that the number of edges connecting the honest regions and the attacker regions is very small. Thus, the impact of attackers with multiple identities is decreased and the attackers may eventually be isolated. In followup work Yu et al. [2008] refined their previous graph-based solutions to significantly lower the bound on the number of malicious nodes in the system.

In parallel with the work on deployed systems, Cheng and Friedman [2005, 2006] and Friedman et al. [2007] have demonstrated several conditions using graph theory that must be satisfied when calculating reputation if a reputation system is to be resilient to Sybil attacks. Most importantly, any nontrivial function used to calculate reputation must be asymmetric to be Sybil-proof, implying reputation must be calculated with respect to identities (nodes) and interactions (edges) in the graph and not just the interactions between nodes. Also, work by Altman and Tennenholtz [2005a, 2005b, 2006] has provided a theoretical basis to analytically compare reputation functions, such as the function used by the PageRank [Page et al. 1998] algorithm, including their different properties such as the resilience to Sybil identities.

5.2. Mitigating Generation of False Rumors

First-hand or direct feedback is created as a result of direct interaction. To prevent the generation of false rumors by fabrication or modification, several systems propose to integrate accountability by means of digital signatures and irrefutable proofs. Irrefutable proofs, often implemented using cryptographic mechanisms, is a defense strategy intended to mitigate the fabrication of feedback by requiring all feedback to be associated with proof of a valid transaction (the interaction of two identities) within the system. For example, TrustGuard [Srivatsa et al. 2005] uses a digitally signed identifier from the other party as a proof of a transaction and describes a protocol to ensure that these transaction proofs are exchanged in an efficient, atomic manner. The approach provides defense against selfish attackers, but it is inefficient against coalitions of malicious attackers. Additional mechanisms are needed to detect that colluding adversaries rate each other highly to build good reputation in the system.

For small coalitions, false data can be filtered out by using a dishonest feedback filter, using similarity measure to rate the credibility of reported feedback. If the feedback data is similar to first-hand experience and other received feedback, it will be used in the reputation calculation. This approach is used by TrustGuard [Srivatsa et al. 2005].

5.3. Mitigating Spreading of False Rumors

Reputation formulation is based on direct and indirect information, as defined in Section 3.1. Basing reputation calculation only on direct information may limit the impact of malicious coalitions on reputation values and it was proposed in systems like Scrivener [Nandi et al. 2005]. The drawback is that in many systems it cannot be guaranteed that every pair of participants will interact and that the interaction is symmetric [Xiong et al. 2005]. Thus, there is a need to share the direct information and aggregate the locally observed information.

Several mechanisms that were especially designed to cope with the problem of spreading and aggregating false reputations were proposed. One approach is to rely on pretrusted identities to reduce the effectiveness of fabricated or altered information. Some systems, such as EigenTrust, depend on pretrusted identities to ensure that the probabilistic trust algorithm will converge. Pretrusted identities do pose additional risk, because, if they are compromised, significant damage can be inflicted before the compromised node is identified. To reduce this risk, systems can employ integrity checking mechanisms (manual audits), use checks and balances on the pretrusted identities (predicates on expected behavior of these identities), and not allow pretrusted identities to be trusted absolutely (as in EigenTrust).

Another approach is to employ statistical methods to build robust formulations that can be reasoned about in a precise fashion. For example, Buchegger and Le Boudec [2004] employed a Bayesian framework, where the probability of a node misbehaving is modeled according to a Beta distribution with the parameters of the distribution being updated as feedback is received. Because the formulation is based on statistics, a precise meaning can be attached to the output. In this case, it allows the users to specify an intuitive tolerance for bad behavior, with tolerance being precisely defined as the maximum percentage of instances that a node has misbehaved before it is excluded from interaction.

Concepts derived from feedback control-theory were used in P2PRep [Aringhieri et al. 2006] to adjust the weighting of historical information in the calculation of the local reputation value. This adaptability gives the system greater resiliency to oscillatory behavior because the weighting of the historical information is tied to how well the reputation is predicting the future. TrustGuard [Srivatsa et al. 2005] defines a heuristic

to mitigate dishonest feedback based on the insight that untrustworthy nodes are more likely to lie and conversely trustworthy nodes are more likely to be honest.

5.4. Preventing Short-Term Abuse of the System

Several systems recognized the problem of attackers abusing the system for short-term gains by letting their reputation degrade quickly and then reentering the system with a new identity.

To differentiate newcomers from nodes already existing in the system which have demonstrated good behavior, several systems propose that newcomers must gain trust and that their reputation increase gradually [Marti and Garcia-Molina 2004; Ham and Agha 2005]. The approach ensures that newcomers will not start with a high reputation and forces them to behave correctly for a given amount of time. Another approach forces new nodes to initially "pay their dues" and provide more service than they receive in order to build a positive reputation [Nandi et al. 2005]. One of the challenges many systems face is balancing the ease of admittance of new nodes versus the resilience to attacks [Morselli et al. 2004].

Other systems such as P2PRep [Aringhieri et al. 2006] and TrustGuard [Srivatsa et al. 2005] also observe that treating old positive behavior equally to new negative behavior may result in attackers abusing the system by using previous altruism to hide current malicious behavior. They propose to use more aggressive short-term history and to give more weight to recent negative behavior. The approach facilitates quick detection when a node becomes a traitor. The drawback is that in systems that do not offer any protection against generating and spreading false rumors, this technique allows malicious node to prevent honest nodes from using the system.

5.5. Mitigating Denial of Service Attacks

Mechanisms to prevent denial of service against the dissemination depend on the structure used for storage and dissemination of reputation values. For example, some systems such as TrustMe [Singh and Liu 2003] use randomization techniques to mitigate the power of malicious collectives. If participants are randomly selected for calculation and dissemination, then its less likely that a malicious collective can control a significant portion of the redundant elements in a process. The system may choose to randomly divide responsibility for either entire identities (with some identities becoming permanently responsible for other identities), or the identities could be randomly assigned for each instance of the calculation.

When systems like DHTs are used, more security mechanisms [Castro et al. 2002] and data replication mechanisms [Flocchini et al. 2007] must be employed to ensure that requests are successfully performed.

Techniques to cope with denial of service attacks on dissemination are similar with the ones used by many routing protocols and include use of acknowledgements, multipath dissemination, gossip mechanisms, and forward error correction codes.

6. EXAMPLES OF REPUTATION SYSTEMS

In this section, we use our framework to analyze in chronological order several existing reputation systems developed for peer-to-peer systems. We selected peer-to-peer reputation systems as there is a wide variety due to the development of many collaborative P2P applications and the different systems demonstrate many of the design choices outlined in our framework. Each of the systems was chosen to provide both a chronological glimpse of how the systems have changed and matured over time as well as to provide insights into how different components of our framework affect the systems.

For example, the CORE system (Section 6.1) uses heuristics to prevent attacks against the system while EigenTrust (Section 6.2) uses a combination of defense techniques, with each system being able to mitigate a very different set of attacks. We provide insights into the vulnerabilities of each system and discuss the effectiveness of defense mechanisms they employ. Due to lack of space, we discuss six representative systems in detail and summarize the other systems in Figures 3 and 4. In Figure 3, items separated by a slash (/) denote multiple properties in the same field while items separated by a comma (,) signify the system can represent the property in one or more methods. For example, if the information type of a systems is "P,N/C," the system utilizes both positive and negative information and the information is represented in a continuous fashion.

6.1. CORE

The CORE [Michiardi and Molva 2002] system was motivated by the need to prevent malicious and selfish behavior of nodes in mobile ad hoc networks (MANETs).

6.1.1. Formulation. The final reputation metric combines directly observable behavior with shared, indirect observations for each known system operation (such as packet forwarding or routing). For each direct interaction between system nodes, each operation is rated over the range from $[-1, 1]$, with higher ratings given to successfully performed operations. The interactions are recorded and over time they are combined using a weighted average, giving higher preference to older data. The reason more relevance is given to past observations is that a sporadic misbehavior in recent observations should have a minimal influence on the evaluation of the final reputation value. The weights are normalized such that the average is also defined over the range $[-1, 1]$. Indirect observations are collected from reply messages sent by distant peers in response to some actual communication related to the purpose of the underlying system. Direct and indirect observations relating to each system operation are linearly combined, and then these subtotals for each operation are combined using a weighted average. The weights for each function are chosen based on simulation results that indicate which functions are more important to the proper operation of the network.

6.1.2. Calculation. Calculation proceeds deterministically in a straightforward fashion, as all information is either generated at the calculating identity (direct observations) or is contained in a reply that the identity has received over the course of normal system operations. The efficiency of calculating the direct reputation subtotals requires a weighted average of all previously known historical values, giving an efficiency of $O(t)$, where t is the number of values retained. The total reputation is obtained by also including indirect observations, giving a final efficiency of $O(c * t) = O(t)$, where c is a constant.

6.1.3. Dissemination. Indirect observations are not actively disseminated between identities in order to conserve power and decrease the number of messages sent by nodes in the network. Rather, indirect observations are embedded within replies already defined within the system protocol. For example, each node on a routing path may append indirect observations about the positive behavior of its neighbor nodes.

6.1.4. Defense Mechanisms. Of the defense mechanisms outlined in this article, CORE uses heuristics motivated by simulation results to prevent attacks against the reputation metric. Since the article is primarily concerned with motivating selfish nodes to participate in the wireless protocols, the defense techniques only partially address

	Formulation				Calculation			Dissemination			
	Source of Information	Information Type	Temporalal Focus	Reputation Metric	Structure	Approach	Efficiency	Structure	Approach	Storage Durability	Redundancy
Beth 94	M, AI	P,N / D	Y	D / C	D	D	$O(a^n)$	D	NA^{φ}	NA^{φ}	NA^{φ}
Zimmermann 95	M, AI	P / D,C	N	D / D,C	D	D	$O(n^2)$	D	D	P	N
eBay 96	M	P,N / D	N	D / D	C	D	O(n)	C	D	P	N
Yu 00	AD,AI	P,N / C	N	D / C	D	D	O(n)	D	D	T	N
P-GRID 01	M	N / B	N	D / D	D	P	O(log n)	D	D	P	P
CORE 02	AD,AI	P,N / C	Y	D / C	D	D	O(n)	D	D	T	N
XRep 02	M	P,N / B	N	P / D	D	P	NA^{φ}	D	P	P	N
EigenTrust 03	M	P,N / B,D	N	D / C	D	P	O(n)	D	D	T	F
Lee 03	M	P,N / B,C	Y	D / C	D	D	O(n)	D	D	T	P
TrustMe 03	NA*	NA*	NA*	NA*	D	D	O(n)	D	D	P	F
Xiong 03	M	N / B	Y	D / B	D	D	O(n)	D	D	T	N
Buchegger 04	M, AI	P / C	Y	D / B, C	D	D	O(1)	D	D	T	P
Feldman 04	M	P,N / D,C	Y	D / C	D	P	O(log n)	D	D	P	N
Guha 04	M	P,N / C	Y	D / B, C	C	P	$O(n^2)$	C	D	T	N
Marti 04	M	P / C	Y	D / C	D	D	O(n)	D	D	T	N
ARA 05	AD	P / C	Y	D / C	D	D	O(n)	D	NA^{φ}	T	P
Scrivener 05	AD	P,N / D	N	D / D	D	D	O(1)	D	D	P	P
Song 05	M, AI	P / C	Y	P / C	D	P	O(n)	D	NA^{φ}	NA^{φ}	NA^{φ}
TrustGuard 05	NA*	P,N / C	Y	D / C	NA*	NA*	O(log n)	NA^{φ}	NA^{φ}	NA^{φ}	NA^{φ}
Xiong 05	M	P / C	Y	D / C	D	D	$O(n^2)$	D	D	T	N
Credence 06	M	P,N / D	Y	D / C	D	D	NA^{φ}	D	P	P	P
PowerTrust 06	M	P,N / D	N	D / C	D	P	O(n)	D	D	T	F
P2PRep 06	M, AD	P / C	Y	P / C	D	P	O(n)	D	D	T	P
Li 07	M, AD	P,N / C	Y	P / C	D	P	O(n)	D	D	T	N

Notes:
1) φ - The property solely depends on the property of the underlying peer-to-peer network.
2) * - The property solely depends on the property of the underlying reputation system.

Formulation			
Source of Information	**Information Type**	**Temporal Focus**	**Reputation Metric**
M Manual	P Positive	Y Strong Emphasis	D Deterministic
AI Automatic Indirect	N Negative	N Not a Strong Emphasis	P Probabilistic
AD Automatic Direct	B Binary		B Binary
	D Discrete		D Discrete
	C Continuous		C Continuous

Calculation		
Structure	**Approach**	**Efficiency**
C Centralized	D Deterministic	Big-O to calculate the reputation metric value for a single identity
D Distributed	P Probabilistic	n # of Nodes
		t # of Historical Values

Dissemination			
Structure	**Approach**	**Storage Strategy**	**Redundancy**
C Centralized	D Deterministic	T Transient	F Full
D Distributed	P Probabilistic	P Permanent	P Partial
			N None

Fig. 3. Characterization of existing reputation systems.

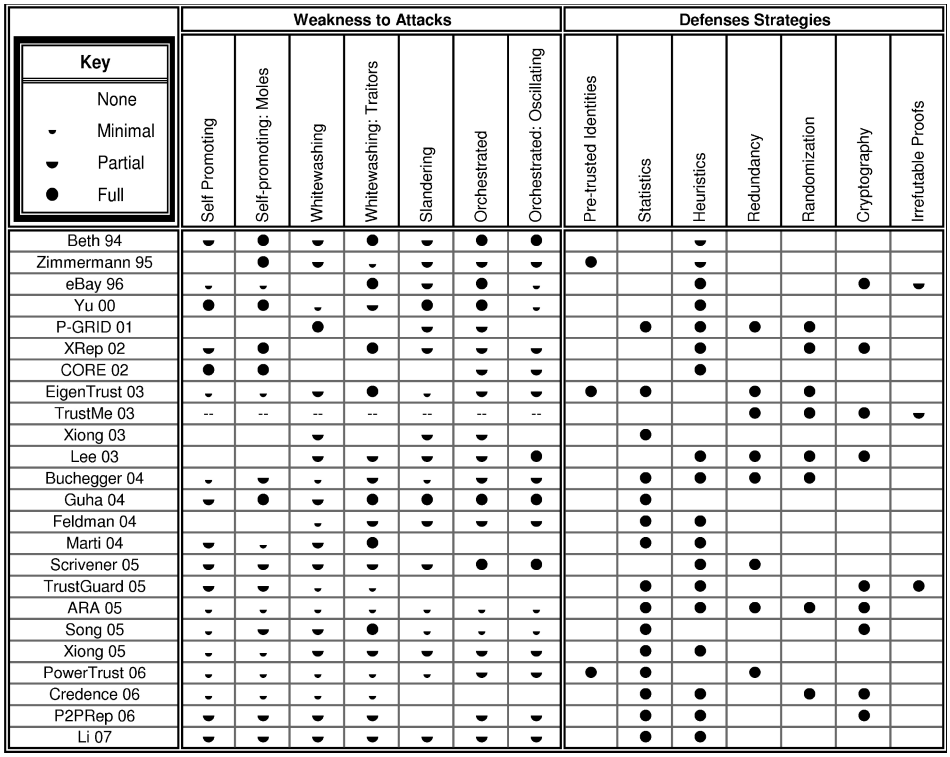

Fig. 4. The weaknesses of existing systems to known attack strategies and the defense mechanisms that these systems employ.

malicious behavior. In the CORE system, there are no techniques to preserve the integrity of the data propagated through the network. While indirect observations are limited to positive observations to prevent whitewashing and slandering attacks, this does not prevent self-promoting attacks. The ability of a node to self-promote in turn degrades the service received by benign nodes since they will have a lower reputation than the malicious nodes and be viewed as less cooperative.

6.2. EigenTrust

The EigenTrust [Kamvar et al. 2003] reputation system was motivated by the need to filter out inauthentic content in peer-to-peer file sharing networks. EigenTrust calculates a global reputation value for each peer in the system based on the local opinions of all of the other peers. The local opinions of nodes are aggregated into a matrix format and the global reputation values are obtained by calculating the left principle eigenvector of that matrix.

6.2.1. Formulation. The input to the EigenTrust formulation consists of the information derived from the direct experience a peer has with other peers in the network and indirect information about the perception of neighboring peers about each other. To acquire the direct information, users provide manual feedback about each peer-to-peer transaction. A user ranks each transaction using the binary scale of positive or negative and the summation of these values is used as input into the formulation. The indirect information is automatically exchanged between peers and is what gives the system

the ability to develop transitive trust. The system considers both positive and negative information and is biased toward positive information.

The formulation does not take into consideration the effects of how reputations change over time. While it is true that the local trust values are a summation of all votes ever cast for a particular identity, the formulation itself makes no attempt to distinguish between votes cast today versus votes cast a year ago.

The final reputation metric is formulated as follows: first, each node i computes a normalized local trust value c_{ij} for node j based on its direct observations. Then, node i calculates a reputation metric for another identity, k, by asking the other identities, j, for their opinions of identity k and weighting those opinions by i's opinion of j: $t_{ik} = \sum_j c_{ij}c_{jk}$. By accepting the trust evaluations of neighbors of neighbors, the system will continue to broaden its view of trust across the network. The reputation metric is formulated deterministically and produces values on a continuous spectrum between 0.0 and 1.0.

6.2.2. Calculation. Given the formulation of trust in EigenTrust is based on the summation of observations and indirect data, it naturally lends itself to calculation using centralized matrix operations. Let C be the matrix $[c_{ij}]$ (i's trust in j), define $\vec{t_i}$ to be $\forall_k t_{ik}$, and define $\vec{c_i}$ to be $\forall_k c_{ik}$. Then the global trust is $\vec{t_i} = (C^T)\vec{c_i}$. To broaden the view further, an identity may ask for his neighbor's neighbor's opinion, $((C^T)(C^T))$, which is then weighted by the identity's opinion of its neighbors ($\vec{c_i}$). Increasing n in $\vec{t_i} = (C^T)^n\vec{c_i}$ continues to broaden the view, and given certain assumptions, t_i will converge to the same vector (the left principle eigenvector of C) for all $\vec{c_i}$. As all nodes will converge to the same values, these values represent the global trust vector.

While the formulation lends itself naturally to a centralized calculation based upon matrix operations, this is not desirable in the peer-to-peer file sharing environment. Instead, each peer calculates the global trust values by using a randomized algorithm which guarantees that each participant will converge to the same value within some error bounds. For the original distributed algorithm, the cost to calculate the global trust value for one identity is $O(n)$ in the worst case since (a) the number of iterations needed to converge can be viewed as constant and (b) it will need to potentially communicate with all other identities. Through optimization this bound can be reduced to $O(\log n)$ without compromising accuracy [Kamvar et al. 2003].

6.2.3. Dissemination. EigenTrust uses a deterministic distributed dissemination framework relying on DHTs for reputation value storage and lookup. A host, the score manager, is responsible for calculating, storing, and communicating reputation values for all identities whose hashed identity falls within the score manager's ownership range in the DHT. The use of multiple hash functions allows multiple score managers to be assigned to each host. The use of multiple score managers and replication within the DHT provides redundancy at all stages of the storage and dissemination process. The efficiency of dissemination corresponds to the efficiency of the underlying DHT in performing lookups, which is typically $O(\log n)$.

6.2.4. Defense Mechanisms. The EigenTrust formulation has foundations in statistics, as the global trust vector can be formulated as the stationary distribution of a Markov chain. The formulation is designed so nodes give greater weight to information obtained from their neighbors or nodes they have interacted with in the past to mitigate malicious manipulations. Pretrusted identities are used to bias reputation values toward known good nodes and ensure that the randomized calculation will converge quickly. Redundancy is employed during the calculation and dissemination stages to prevent

benign data loss and malicious data tampering. Each of the score managers for an identity is randomly selected, making it less likely that a single malicious collective will be responsible for the reputation value for any one identity.

6.3. TrustGuard

While electronic reputation systems have been proposed as an efficient and effective way to minimize the impact of selfish and malicious nodes on peer-to-peer systems, the reputation systems themselves are often ill protected, containing ad hoc protection mechanisms against a small set of adversarial strategies. The TrustGuard framework [Srivatsa et al. 2005] has been proposed as a comprehensive method to safeguard reputation systems. The system uses a strategic oscillation guard based on a Proportional-Integral-Derivative (PID) controller to combat malicious oscillatory behavior. Fake feedbacks are prevented with the help of a fake transaction detection component which binds feedback to unforgeable transaction proofs. Finally, dishonest feedbacks are filtered out by using a similarity measure to rate the credibility of reported feedback. The framework focuses on designing a robust formulation while maintaining flexibility in the system implementation, allowing the mitigation techniques to be integrated into a variety of reputation systems.

6.3.1. Formulation. Once the node has collected data and the data has passed through the fake transaction detection component and the dishonest feedback filter, it is fed into the strategic oscillation guard in order to create a final trust value. The strategic oscillation guard takes as input the raw reputation values computed from some other reputation system and formulates the output trust value as the sum of three weighted components:

—The first component represents a node's current performance and is the raw trust value as computed by the underlying reputation system.

—The second component is the past history of a node's actions formulated as the integral of the function representing all prior reputation values divided by the current point in time.

—The third and final component reflects sudden changes in a node's performance and is formulated by the derivative of the above mentioned function.

The flexibility and resiliency to attack are achieved in how the weights for each component are chosen. For example, if the second component is weighted heavily, the past performance of the node is more important than the current performance.

6.3.2. Calculation. In order to efficiently store and calculate the historical components specified by the strategic oscillation guard's formulation, the concept of fading memories is introduced. Instead of storing all previous reputation values, TrustGuard represents these values using only $\log_2 t$ values, where t represents system time intervals, with exponentially more detail is stored about the recent events. This technique allows the strategic oscillation guard calculations to be deterministically performed with an efficiency of $O(\log t)$ instead of $O(t)$.

6.3.3. Dissemination. The dissemination of information is dependent on the underlying overlay network and thus the dissemination techniques are outside the scope of the TrustGuard.

6.3.4. Defense Mechanisms. In TrustGuard, it is assumed that the base overlay network is resilient to attack and provides a means for authenticating messages. Under these assumptions, TrustGuard uses control theory as the basis for its strategic oscillation guard. Using different empirically determined weighting for the guard, the system can mitigate many of the malicious attacks.

The fake transaction detection mechanism uses irrefutable proofs to prevent input resulting from fake transactions from being admitted into the reputation system. Assuming each entity in the network has an associated public/private key pair, a transaction proof is exchange for each transaction, allowing claims of malicious activity to be checked by a trusted third party.

The goal of the dishonest feedback filter is to use statistical measures to make the raw reputation values computed by the underlying reputation system resilient against identities that report false feedback. The first approach mentioned is to weight the source values used to compute the reputation value for a node by the current reputation values of the identities producing the source information. However, this has the drawback of being vulnerable to nodes which follow the protocol and have good reputations but lie about others reputation. Instead, Srivatsa et al. [2005] proposed to weight reputation values by using a personality similarity measure, which is defined to be the normalized root mean square of the differences between the feedback each node gave to identities in the common identity set. The common identity set is defined to be those identities that both identities have interacted with in the past. This has the effect that the weight given to others' feedback about oneself will depend on the similarity of how both rated other identities in the past, with the idea that honest identities will approximately give the same feedback for other identities. The Credence [Walsh and Sirer 2006] system is also built around this approach.

6.4. Scrivener

Scrivener [Nandi et al. 2005] is based on principles from neoclassical economics which state that naturally "rational" clients must be given an incentive in order to cooperate and not cheat the system. The goal of Scrivener is to enforce fairness among all participants in a peer-to-peer file sharing system. The reputation of each host is not a globally calculated value, but rather is specific to individual pairwise sharing relationships within the overlay network. The formulation and calculation dimensions describe how pairwise relationships determine credit balances. The dissemination dimension describes how a transitive relationship can be formed between nonneighboring identities.

6.4.1. Formulation. Scrivener maintains a history of interactions between neighboring peers, tracking when they provide resources (credits) and when they consume resources (debits). The system collects directly observable information automatically for each interaction of immediate neighbors in the overlay network. Credit balance is defined to be the difference in the amount of data consumed less the resources provided. In order to establish long-term cooperation, nodes must keep this credit balance in stable storage. Using the credit balance in conjunction with a confidence metric that represents how often a request is successfully fulfilled by a given node, a credit limit for each node is established.

In order to allow nodes to join the network, each new neighbor chosen by a node is initially given a small, positive credit limit. To prevent nodes from constantly selecting new neighbors and abusing the initial credit limit, any node requesting (not chosen) to be a neighbor of a node is assigned an initial credit limit of zero. When a host A has used up its credit limit with a host B, B will not further fulfill requests from A. Host

B can still request data from A so that A can repay the debt. If A does not fulfill B's requests properly then B's confidence in A decreases. If the confidence reaches zero, B will ignore A and choose a different identity in the overlay network to replace A as its neighbor. Since the credit is assumed to be maintained in stable storage, B remembers the debt that was associated with A indefinitely.

Although the credit balance is a summary of all past behavior of a neighboring node, no record is kept of how this has changed over time. The formulation itself is deterministic and produces discrete values.

6.4.2. Calculation. The calculation of the credits and confidence metric are fully distributed to the point that there is no single, global value produced for each identity in the system. Each calculation is processed entirely local to the pair of identities involved and can be performed in constant running time with respect to the number of identities in the system.

6.4.3. Dissemination. As credit balances are only established between neighboring identities in the overlay network, a transitive trading system is needed to facilitate data transfers between any two arbitrary identities within the overlay network.

If host A wants to receive content from some host, Z, it first must find a credit path, where each identity in the credit path has positive credit with which to "pay" the successor identity. This path is normally determined using the overlay network routing protocol, such as through the use of a DHT. Once the path has been determined, A then simultaneously sends a payment to and decreases its confidence in the first identity in the credit path (named B herein). The drop in confidence is in anticipation of B's possible failure to route the request to the next identity in the credit path and to motivate B to participate in the request. If B does not participate, A's confidence in B will eventually reach zero and A will refuse to communicate with B. Then, B uses the credit received from A to continue the process until the destination is reached. Once Z receives the credit, Z will process the request. Once the request is complete, Z sends an indicator message backwards along the credit path, which causes nodes to adjust confidence levels back to their original values and then increase them to reflect the success of the content transfer.

Redundancy is integrated into the system via the notion of content caching. The efficiency of this process is determined by the efficiency of (a) finding the credit path (b) the length of the resulting credit path.

6.4.4. Defense Mechanisms. Similar to CORE, Scrivener is primarily concerned with motivating rational, selfish nodes to participate in the system and thus the defense techniques only partially address malicious behavior. The primary defense mechanism within Scrivener is the use of statistical formulas to encourage participants to participate correctly in the protocol. If nodes act selfishly or maliciously, they will eventually acquire a negative credit balance and with the assumed long-lived identifers, eventually be excluded from the network. Redundancy is also utilized and allows identities to check the validity of certain claims by an identity. For example, if a sender, Z, claims that some content does not exist or that it has completed the transaction, nodes along the credit path (that later propagate the finished message) can choose to ask other identities providing the same content to verify the truth of the claim.

6.5. P2PRep

The P2PRep [Aringhieri et al. 2006] reputation system is designed to mitigate the effects of selfish and malicious peers in an anonymous, completely decentralized system. The

system uses fuzzy techniques to collect and aggregate user opinions into distinct values of trust and reputation. In P2PRep, trust is a defined as a function based on an entity's reputation and several environmental factors such as the time since the reputation has last been modified. Reputation is viewed at two levels in the system: (1) locally representing the direct interactions between peers and (2) network-wide representing the aggregation of multiple opinions about a peer. When a peer wants to use a network resource (download a file), it (1) queries for resource locations and receives back a list of possible resource providers, (2) polls the network about the reputation of the possible resource providers, (3) evaluates the poll responses, and (4) synthesizes a reputation value from the local and network responses using fuzzy techniques. Based on this synthesized value, a peer will decide whether or not to trust the provider and use the resource.

6.5.1. Formulation. In P2PRep, since participants are assumed to be anonymous and no peer will retain an identity with a negative reputation, only positive values are used in the formulation. Each direct interaction between system nodes is recorded and given a Boolean value, with 1 indicating the outcome was satisfactory and 0 otherwise. In order to calculate a local reputation value, the individual Boolean values are aggregated using an exponentially weighted moving average designed to take into account the age and importance of the data, resulting in a value over the range from $[0, 1]$. In order to augment the local reputation value, a peer will collect reputation values from the network. Using an ordered weighted average, the indirect observations obtained from the network query are combined with the local observations to produce a final reputation value in the unit interval of $[0, 1]$.

6.5.2. Calculation. The calculation of both the local and network reputations are fully distributed to the point that there is no single, global value produced for each identity in the system. For each possible interaction, data is gathered from the network and processed locally by the node requesting the resource and each calculation can be performed in linear running time with respect to the number of identities that reply to a poll request for reputation values.

6.5.3. Dissemination. All information requests are broadcast throughout the network and all information replies are unicast back to the requester using the underlying peer-to-peer system's communication framework. Several improvements can be made to improve the efficiency of the dissemination structure including intelligent forwarding techniques (e.g., forwarding poll packets only to the necessary peers) and vote caching (e.g., retaining votes of popular peers).

6.5.4. Defense Mechanisms. The main defense technique P2PRep utilizes to mitigate the effect of malicious nodes is a vote verification process. The requester randomly audits some of the votes by sending a vote confirmation message to the IP address associated with the vote. This ensures the interaction actually happened and the vote corresponding to that IP address is correct, making vote falsification more difficult for an attacker. Also, the formulation of the network wide reputation is designed to give more weight to local observations and uses an adaptive weighting scheme in order to be responsive to network change, making it more difficult for malicious nodes to gain an advantage by reporting false votes.

Another key design consideration in the P2PRep is maintaining user anonymity. The system guards the anonymity of users and the integrity of packets through the use of public key cryptography. All replies are signed using the requester's public key,

protecting the identity of the responder and the integrity of the data. Only the requester is able to decrypt the packet and check the validity of the information.

6.6. Credence

Credence [Walsh and Sirer 2006] was motivated by the need for peers to defend against file pollution in peer-to-peer file sharing networks and has been deployed as an add-on to the LimeWire client for the Gnutella network. The system relies on the intuitive notion that honest identities will produce similar votes as they rate the authenticity of a file, implying that identities with similar voting patterns can be trusted more than identities with dissimilar voting patterns.

6.6.1. Formulation. The input to the formulation is the users' manually entered positive or negative votes indicating the authenticity of a downloaded file. Specifically, each user is asked to indicate for each attribute of the search result whether the attribute was one of many possible true values, was the only possible true value, or was not a true value (possibly specifying what the true value actually should be). A historical record of an identity's recent votes are stored in a local vote database. Additionally, each identity will proactively query the peer-to-peer network to gather additional votes for neighboring nodes.

The final reputation metric for a search result is formulated by taking a weighted average of other identities' statements (each statement represents $+1$ if the statement completely supports the search result's attributes or -1 otherwise). The weight assigned to each identity depends on the statistical correlation between the vote history of the identity performing the calculation and the vote history for each of its peers. Heuristics are applied to the correlation coefficient so that statistically insignificant correlations and correlations without sufficient history are discarded and that new identities without a large voting history can still estimate a weight.

6.6.2. Calculation. The calculation of reputation metric values for a given search query result proceeds as follows: First, an identity will send vote gathering queries into the network, requesting that neighboring identities respond with their vote for the file of interest as well as with the most important votes that the neighboring identities know about. Using the local vote information and the received votes weighted with their measured correlation, the weighted average is computed. Calculation of the correlation weights between peers can be performed incrementally by updating the weight for a particular peer when additional voting history for that peer is received. The use of the persistent storage allows for the digital signatures of the statements of peers to only have to be verified once (under the assumption that the underlying persistent store can be trusted), further increasing efficiency of the Credence system.

6.6.3. Dissemination. Credence utilizes several mechanisms to disseminate votes across the system, broadening the influence of the votes and allowing voting information to remain available even when the original voter is offline. First, vote information from neighboring identities is stored persistently after each query. The information received from each neighboring peer also contains information about other peers in the network. In this way, the vote from one particular identity can disseminate widely across the network as proactive queries are made regarding the file. Additionally, gossip-based techniques are employed in the background so that voting information for unpopular objects has a broader reach throughout the system.

In addition to propagating actual voting information regarding specific files, the system uses a flow-based algorithm, similar to the idea behind PageRank [Page et al.

1998] and EigenTrust [Kamvar et al. 2003], to calculate an estimate of the correlation between any two indirectly connected identities within the graph. This allows larger networks of trust to be built such that strong correlation can be established between identities even if they do not vote on the same files. Each client builds a local model of a portion of the overall trust network and using a gossip-based protocol, a node propagates trust along paths from itself to distant peers through known pairwise relationships.

6.6.4. Defense Mechanisms. The underlying intuition within Credence that honest users have similar voting patterns limits the impact of any attack pattern. Malicious nodes are forced to vote honestly the majority of time so that they can develop strong correlation values with other honest users. If the attackers vote dishonestly, it directly diminishes their correlation coefficients with other honest users and lessens their impact on the reputation system. Attacks by coalitions of attackers, while still effective, are impacted in a similar fashion since they require the entire group to establish credible voting patterns and are inherently more costly for the attackers.

A key security consideration in the Credence system is the use of mechanisms to prevent spoofed votes or votes generated by fake identities. The system guards against such attacks by issuing digital certificates in an anonymous but semi-controlled fashion. Walsh and Sirer [2006] proposed to mitigate Sybil attacks by requiring expensive computation on the part of the client before the server grants a new digital certificate. Every voting statement is digitally signed by the originator and anyone can cryptographically verify the authenticity of any given voting statement.

Honest nodes in Credence occasionally use the inverse of votes by nodes with weak correlations based on the fact that these votes were most likely submitted by malicious users and are opposite of their true, correct values.

7. CONCLUSIONS

This article is the first survey focusing on the design dimensions of reputation systems and the corresponding attacks and defenses. We have developed an analysis framework that can be used as common criteria for evaluating and comparing reputation systems. We have defined an attacker model and classified known and potential attacks on reputation systems within this model. Defense mechanisms and their corresponding strengths and weaknesses were discussed. We have demonstrated the value of the analysis framework and attack and defense characterizations by surveying several key reputation systems, drawing insights based on the new framework. This analysis framework is also valuable for future research in that it provides understanding into the implications of design choices.

Reputation systems play an ever-increasingly important part in online communities. Understanding reputation systems and how they can compare to each other is an important step toward formulating better systems in the future. This article has sought to provide more rigorous methods to compare existing systems and to bring understanding of these systems to a broader audience, including those who build systems that rely on reputation systems.

REFERENCES

ABERER, K. AND DESPOTOVIC, Z. 2001Managing trust in a peer-2-peer information system . In *CIKM '01: Proceedings of the Tenth International Conference on Information and Knowledge Management*. ACM Press, New York, NY, 310–317.

ADAR, E. AND HUBERMAN, B. 2000. Free riding on Gnutella. *First Monday 5*, 10, 2.

ADLER, B. AND DE ALFARO, L. 2007. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th International Conference on the World Wide Web (WWW)*. ACM Press, New York, NY, 261–270.

AKERLOF, G. 1970. The market for "lemons": Quality uncertainty and the market mechanism. *Quart. J. Econom. 84*, 3, 488–500.

ALTMAN, A. AND TENNENHOLTZ, M. 2005a. On the axiomatic foundations of ranking systems. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. 917–922.

ALTMAN, A. AND TENNENHOLTZ, M. 2005b. Ranking systems: The PageRank axioms. In *Proceedings of the 6th ACM Conference on Electronic Commerce*. ACM Press New York, NY, 1–8.

ALTMAN, A. AND TENNENHOLTZ, M. 2006. An axiomatic approach to personalized ranking systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*.

ARINGHIERI, R., DAMIANI, E., VIMERCATI, S. D. C. D., PARABOSCHI, S., AND SAMARATI, P. 2006. Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. *J. Am. Soc. Inf. Sci. Technol. 57*, 4 (Feb.), 528–537.

BA, S. AND PAVLOU, P. 2002. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS Quart. 26*, 3, 243–268.

BAZZI, R. A. AND KONJEVOD, G. 2005. On the establishment of distinct identities in overlay networks. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC'05)*. ACM Press, New York, NY, 312–320.

BETH, T., BORCHERDING, M., AND KLEIN, B. 1994. Valuation of trust in open networks. In *Computer Security-Esorics 94: Third European Symposium on Research in Computer Security*. Springer, Brighton, U.K.

BUCHEGGER, S. AND LE BOUDEC, J. Y. 2004. A robust reputation system for P2P and mobile ad-hoc networks. In *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*.

CASTRO, M., DRUSCHEL, P., GANESH, A., ROWSTRON, A., AND WALLACH, D. S. 2002. Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev. 36*, SI, 299–314.

CHENG, A. AND FRIEDMAN, E. 2005. Sybilproof reputation mechanisms. In *Applications, Technologies, Architectures, and Protocols for Computer Communication*. ACM Press New York, NY, 128–132.

CHENG, A. AND FRIEDMAN, E. 2006. Manipulability of PageRank under Sybil strategies. In *First Workshop on the Economics of Networked Systems (NetEcon06)*.

CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. 2001. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

DAHAN, S. AND SATO, M. 2007. Survey of six myths and oversights about distributed hash tables' security. In *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07)*. IEEE Computer Society Press, Los Alamitos, CA.

DAMIANI, E., DE CAPITANI DI VIMERCATI, S. PARABOSCHI, S., AND SAMARATI, P. 2003. Managing and sharing servants' reputations in P2P systems. *IEEE Trans. Knowl. and Data Eng. 15*, 4 (July-Aug.), 840–854.

DAMIANI, E., DI VIMERCATI, D. C., PARABOSCHI, S., SAMARATI, P., AND VIOLANTE, F. 2002. 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 (CCS'02)*. ACM Press, New York, NY, 207–216.

DELLAROCAS, C. 2003. The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms. *Manage. Sci. 49*, 10 (Oct.), 1407–1424.

DIMITRIOU, T., KARAME, G., AND CHRISTOU, I. 2007. SuperTrust: A secure and efficient framework for handling trust in super peer networks. In *Proceedings of ACM PODC*.

DOUCEUR, J. R. 2002. The Sybil attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*. Springer, Berlin/Heidelberg, Germany, 251–260.

EUGSTER, P., HANDURUKANDE, S., GUERRAOUI, R., KERMARREC, A.-M., AND KOUZNETSOV, P. 2001. Lightweight probabilistic broadcast. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01)*.

FELDMAN, M., LAI, K., STOICA, I., AND CHUANG, J. 2004. Robust incentive techniques for peer-to-peer networks. In *Proceedings of the 5th ACM Conference on Electronic Commerce 1*, 1, 102–111.

FLOCCHINI, P., NAYAK, A., AND XIE, M. 2007. Enhancing peer-to-peer systems through redundancy. *IEEE J. Select. Areas Commun. 25*, 1 (Jan.), 15–24.

FRIEDMAN, E., RESNICK, P., AND SAMI, R. 2007. *Algorithmic Game Theory*. Cambridge University Press, Cambridge, U.K.

FRIEDMAN, E. J. AND RESNICK, P. 2001. The social cost of cheap pseudonyms. *Econom. Manage. Strat. 10*, 2, 173–199.

GUHA, R., KUMAR, R., RAGHAVAN, P., AND TOMKINS, A. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on the World Wide Web (WWW'04)*. ACM Press, New York, NY, 403–412.

HAM, M. AND AGHA, G. 2005. ARA: A robust audit to prevent free-riding in P2P networks. In *Fifth IEEE International Conference on Peer-to-Peer Computing (P2P)*. 125–132.

HOUSER, D. AND WOODERS, J. 2006. Reputation in auctions: Theory, and evidence from eBay. *J. Econom. Manage. Strat. 15*, 2 (June), 353–369.

JØSANG, A., ISMAIL, R., AND BOYD, C. 2007. A survey of trust and reputation systems for online service provision. *Decis. Supp. Syst. 43*, 2 (Mar.), 618–644.

KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. 2003. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on the World Wide Web (WWW'03)*. ACM Press, New York, NY, 640–651.

KHOPKAR, T., LI, X., AND RESNICK, P. 2005. Self-selection, slipping, salvaging, slacking, and stoning: The impacts of negative feedback at eBay. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC'05)*. ACM Press, New York, NY, 223–231.

LAI, K., FELDMAN, M., STOICA, I., AND CHUANG, J. 2003. Incentives for cooperation in peer-to-peer networks. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*.

LEE, S., SHERWOOD, R., AND BHATTACHARJEE, B. 2003. Cooperative peer groups in Nice. In *Proceedings of the IEEE INFOCOM*.

LEVIEN, R. 2003. Attack Resistant Trust Metrics. Ph.D. dissertation. University of California at Berkeley, Berkeley, CA. `http://www.levien.com/thesis/compact.pdf`.

LI, F. AND WU, J. 2007. Mobility reduces uncertainty in MANETs. In *Proceedings of IEEE INFOCOM*.

LIAN, Q., ZHANG, Z., YANG, M., ZHAO, B., DAI, Y., AND LI, X. 2007. An empirical study of collusion behavior in the Maze P2P file-sharing system. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS)*. IEEE Computer Society Press, Los Alamitos, CA.

LIN, K., LU, H., YU, T., AND TAI, C. 2005. A reputation and trust management broker framework for Web applications. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE)*. IEEE Computer Society Press, Los Alamitos, CA. 262–269.

MARTI, S. AND GARCIA-MOLINA, H. 2004. Limited reputation sharing in P2P systems. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC'04)*. ACM Press, New York, NY, 91–101.

MARTI, S. AND GARCIA-MOLINA, H. 2006. Taxonomy of trust: Categorizing P2P reputation systems. *Comput. Netw. Internat. J. Comput. Telecommun. Netw. 50*, 472–484.

MATEI, R., IAMNITCHI, A., AND FOSTER, P. 2002. Mapping the Gnutella network. *IEEE Internet Comput. 6*, 1 (Jan.-Feb.), 50–57.

MICHIARDI, P. AND MOLVA, R. 2002. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security*. Kluwer, B.V., Deventer, The Netherlands, 107–121.

MORSELLI, R., KATZ, J., AND BHATTACHARJEE, B. 2004. A game-theoretic framework for analyzing trust-inference protocols. In *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*.

NANDI, A., NGAN, T.-W., SINGH, A., DRUSCHEL, P., AND WALLACH, D. S. 2005. Scrivener: Providing incentives in cooperative content distribution systems. *Middleware 1*, 1 (Nov.), 270–291.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The PageRank citation ranking: Bringing order to the Web. Tech. rep. Stanford Digital Library Technologies Project, Stanford University, Stanford, CA.

PIATEK, M., ISDAL, T., ANDERSON, T., KRISHNAMURTHY, A., AND VENKATARAMANI, A. 2007. Do incentives build robustness in BitTorrent? In *Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. 2000. A scalable content addressable network. Tech. rep. TR-00-010, UC Berkeley, Berkeley, CA.

REED, I. S. AND SOLOMON, G. 1960. Polynomial codes over certain finite fields. *J. Soc. Indust. Appl. Math. 8*, 2 (June), 300–304.

RESNICK, P., KUWABARA, K., ZECKHAUSER, R., AND FRIEDMAN, E. 2000. Reputation systems. *Commun. ACM 43*, 12, 45–48.

RESNICK, P., ZECKHAUSER, R., SWANSON, J., AND LOCKWOOD, K. 2006. The value of reputation on eBay: A controlled experiment. *Experiment. Econom. 9*, 2 (June), 79–101.

ROWSTRON, A. AND DRUSCHEL, P. 2001. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Middleware 11*, 329–350.

SINGH, A. AND LIU, L. 2003. TrustMe: Anonymous management of trust relationships in decentralized P2P systems. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P'03)*. 142–149.

SONG, S., HWANG, K., ZHOU, R., AND KWOK, Y.-K. 2005. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Comput. 9*, 6 (Nov.-Dec.), 24–34.

SRIVATSA, M., XIONG, L., AND LIU, L. 2005. TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th International Conference on the World Wide Web (WWW'05)*. ACM Press, New York, NY, 422–431.

STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, F., AND BALAKRISHNAN, H. 2001. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the ACM SIGCOMM Conference*. 149–160.

SURVEY. 2005. E-crime watch survey. `http://www.cert.org/archive/pdf/ecrimesurvey05.pdf`.

SURYANARAYANA, G. AND TAYLOR, R. N. 2004. A survey of trust management and resource discovery technologies in peer-to-peer applications. Tech. rep. UCI-ISR-04-6, UC Irvine, Irvine, CA.

WALSH, K. AND SIRER, E. G. 2006. Experience with an object reputation system for peer-to-peer filesharing. In *Proceedings of the Symposium on Networked System Design and Implementation (NSDI)*.

XIONG, L. AND LIU, L. 2002. Building Trust in Decentralized Peer-to-Peer Electronic Communities. In *Proceedings of the International Conference on Electronic Commerce Research (ICECR-5)*.

XIONG, L. AND LIU, L. 2003. A reputation-based trust model for peer-to-peer e-commerce communities. In *Proceedings of the IEEE Conference on Electronic Commerce*.

XIONG, L., LIU, L., AND AHAMAD, M. 2005. Countering sparsity and vulnerabilities in reputation systems. Tech. rep. TR-2005-017-A. Emory University, Atlanta, GA.

YU, B. AND SINGH, M. P. 2000. A Social Mechanism of Reputation Management in Electronic Communities, *Coop. Inform. Agents 1*, 1, 154–165.

YU, H., GIBBONS, P., KAMINSKY, M., AND XIAO, F. 2008. A near-optimal social network defense against Sybil attacks. In *Proceedings of the IEEE Symposium on Security and Privacy*.

YU, H., KAMINSKY, M., GIBBONS, P. B., AND FLAXMAN, A. 2006. SybilGuard: Defending against Sybil attacks via social networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'06)*. ACM Press, New York, NY, 267–278.

ZAGE, D. J. AND NITA-ROTARU, C. 2007. On the accuracy of decentralized network coordinates in adversarial networks. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*. ACM Press, New York, NY.

ZHAO, B. Y., KUBIATOWICZ, J. D., AND JOSEPH, A. D. 2001. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. rep. UCB/CSD-01-1141, UC Berkeley, Berkeley, CA.

ZHOU, R. AND HWANG, K. 2006. Trust overlay networks for global reputation aggregation in P2P grid computing. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS)*.

ZHOU, R. AND HWANG, K. 2007. PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parall. Distrib. Syst. 18*, 4, 460–473.

ZIMMERMANN, P. 1995. *The Official PGP User's Guide*. MIT Press Cambridge, MA.

Received September 2007; revised March 2008; accepted June 2008