Security of Online Reputation Systems: Evolution of Attacks and Defenses

Yan (Lindsay) Sun and Yuhong Liu
Department of Electrical, Computer and Biomedical Engineering
University of Rhode Island,
4 East Alumni Ave., Kingston, RI 02881
Emails: {yansun, yuhong}@ele.uri.edu; phone: (401)874-5803

I. Introduction

The Internet has created vast opportunities to interact with strangers. The interactions can be fun, informative and even profitable [1]. However, there is also risk involved. Will a seller at eBay ship the product in time? Is the advice from a self proclaimed expert at Epinion.com trustworthy? Does a product at Amazon.com have high quality as described?

To address these problems, one of the most ancient mechanisms in the history of human society, word-of-mouth, is gaining new significance in the cyber space, where it is called *reputation system* [2]. A reputation system collects evidence about the properties of individual objects, aggregates the evidence, and disseminates the aggregated results. Here, The *objects* can be products (e.g. in the Amazon product rating system), businesses (e.g. hotel ratings in various travel sites), users (e.g. sellers and buyers at eBay), and digital content (e.g. video clips at YouTube). The *aggregated results* are called reputation scores. Most commercial systems collect user feedbacks (i.e. ratings/reviews) as evidence. This type of system is referred to as *feedback-based reputation system.*

Signal processing plays an important role in reputation systems, in which the reputation scores are in fact the prediction of the objects' future behaviors based on the data describing their past behaviors. Various signal models are suggested for computing objects' reputation scores. In Bayesian reputation systems, an updated reputation score (i.e. posteriori) is computed based on the previous reputation score (i.e. priori) and the new feedbacks (i.e. observations) [3]. Belief theory, a framework based on probability theory, has been used to combine feedbacks (i.e evidence from different sources) and represent reputation scores [4]. Flow models, which compute reputation by transitive iteration through looped or arbitrarily long chains [5], are investigated to calculate reputation scores. Reputation has also been interpreted as linguistically fuzzy logic concepts [6]. Furthermore, as discussed later in this section, signal processing techniques are widely used to defend reputation systems against attacks.

As reputation systems are having increasing influence on consumers online purchasing decisions and online digital content distribution [7], the incentive to manipulate reputation systems is growing. There is ample evidence. In February 2004, due to a software error, Amazon.com's Canadian site mistakenly revealed the true identities of some book reviewers. It turned out that a sizable proportion of those reviews were written by the books' own publishers, authors and competitors [8]. Reported in [9], some eBay users are artificially boosting their reputation by buying and selling feedbacks. More important, there are businesses that promote/downgrade online reputation for profit through artificial feedbacks. For example, many small companies provide "reputation boosting" services for sellers at Taobao, which is the largest Internet retail platform in China and has taken up about 3/4 of the market share [10]. There are also many websites where users can buy views for promoting their YouTube videos. Such coordinated and even profit-driven manipulations can greatly distort reputation scores, make reputation systems lose their worthiness, undermine users confidence about reputation-centric systems, and may eventually lead to system failure [11].

To protect reputation systems, many defense solutions have been developed. Most of the defense solutions utilize signal processing techniques to differentiate normal feedbacks from dishonest feedbacks and normal users from malicious users. For example, the method in [12] assumes that feedbacks follow Beta distributions and considers the feedbacks outside q and $(1-q)$ quantile of the majority's opinions as unfair feedbacks. Here, q can be viewed as a sensitivity parameter roughly describing the percentage of feedbacks being classified as dishonest. The smaller the value of q, the less false positives and the more false negatives in the dishonest feedback detection. In [13], an entropy based approach identifies feedbacks that bring a significant change in the uncertainty in feedback distribution as unfair feedbacks. In [14], a set of statistical methods are jointly applied to detect the time intervals in which dishonest feedbacks are highly likely present.

Just as in many security problems, attackers will exploit the vulnerabilities in the defense solutions and advance their attacks. Then the defender will improve the defense by addressing the advancement in attacks. There is always an "arms race" or "competition" between attacks and defenses. The arms race for attacking/securing reputation systems is currently taking place and evolving rapidly.

In this paper, we will introduce the evolution of attack strategies against online reputation systems and the state-of-the-art defense mechanisms. In particular, we introduce background on reputation systems (Section II-A), describe attack models and classification (Section II-B), explain advantages/disadvantages of representative attack/defense strategies and their evolution (Section

III), present a powerful attack called RepTrap (Section IV-A), and discuss two defense solutions that utilize temporal and correlation analysis (Section IV-B, IV-C). The attack/defense solutions presented in Section IV-A, IV-B and IV-C are stronger than most commonly known attack/defense, and tested using real users' data collected from commercial systems or through cyber competitions. The future research challenges will be discussed in Section V.

II. Reputation Systems and Attack Classification

A. Reputation System Model

As discussed earlier, a reputation system collects *evidence* about the properties of individual *objects*, analyzes and aggregates the evidence, and disseminates the *aggregated results* as reputation scores. In this subsection, we will review representative reputation systems, such as those used by Amazon, YouTube, Digg, CitySearch etc, and build the system model as follows.

- *Evidence Collection* A reputation system can obtain three types of evidences. The *first* type is direct observation, usually based on the experiences of the employees of a business (e.g. ConsumerReport.org). The *second* type is opinions from experts, who have verifiable expertise and provide feedbacks either voluntarily or for a fee. Both types of evidence are considered reliable, but costly to collect for a large number of objects. The *third* type is feedbacks provided by users, which have been the main source of evidence in most of today's popular reputation systems, such as the product rating system at Amazon, restaurant ratings at Yelp, and customer reviews at Apple app store. However, user feedback is also the least reliable source of evidence because it can be easily manipulated [15].
- *Reputation Aggregation* Reputation aggregation algorithms calculate the reputation scores of objects based on the collected evidence. A good reputation aggregation scheme should be able to compute reputation scores that accurately describe the true quality of objects, even if there are dishonest feedbacks.
- *Reputation Dissemination* Reputation systems not only make the reputation scores publicly available, but also release extra information to help users understand the meaning of the reputation scores. For example, Amazon shows all the feedbacks given by each reviewer. YouTube starts to provide visualization of viewing history for video clips, accompanied with some statistical features.

To manipulate a reputation system, attackers can (1) obtain information about the *target objects*, defined as objects whose reputation scores' increase/decrease is the goal of the attack, (2) insert dishonest feedbacks in the evidence collection phase, and (3) aim to mislead the evidence aggregation algorithm such that it yields unfairly high/low reputation scores for the target objects, misclassifies honest feedbacks/users as dishonest, and misclassifies dishonest feedbacks/users as honest.

Therefore, from the defense points of view, the first step is to control how much extra information to release and when to release it, setting barriers for attackers gaining knowledge that facilities advanced attacks while not hurting the experiences of normal users. The second step is to encourage honest feedbacks in the evidence collection phase through various incentive mechanisms [16]. The third step, which is probably the most sophisticated and critical one, is to design attack resistant evidence aggregation algorithm, which can detect dishonest feedbacks and eliminate the impact of dishonest feedbacks. In this paper, the discussion on the defense will focus on utilizing signal processing techniques in the third step.

B. Attacks Classification

To secure a system, people have to first understand the attacks. Classification of various attacks is an effective way to understand the features of the attacks. In this paper, we classify attacks against feedback-based reputation systems from four different angles.

- *Targeting Objects or Targeting System* The purposes of attackers can be divided into two categories: (1) manipulating reputation scores of one or several objects; and (2) undermining the performance of the entire reputation system. In the *first* category, the attacker aims to boost or downgrade reputation scores of specific target objects. These objects either gain or loss advantage due to inaccurate reputation scores when competing with similar objects for users' attention or preference. In the *second* category, the attacker aims to mislead the reputation of a sizeable proportion of objects and undermine the performance of the entire reputation systems. For example, in [17], an advanced attack can overturn the reputation (from positive to negative) of a large number of objects and therefore undermine users' trust in the reputation system.
- *Direct Attack or Indirect Attack* Many reputation systems compute two types of reputation: *object quality reputation* describing whether an object has high quality or not and *feedback reputation* describing whether a user tends to provide honest feedback or not. The feedback reputation is often used to mitigate the effect of dishonest feedbacks in the evidence aggregation algorithms. For example, the feedbacks from users with high feedback reputation can carry larger weights in the calculation of object quality reputation. From this perspective, the attacks can be classified into two categories: *direct attacks* and *indirect attacks*. In direct attacks,

Category	Code	Category	Code
Target Object	T_o	Target System	T_s
Direct Attack	A_d	Indirect Attack	A_{ind}
Collusion	C_y	Non-collusion	C_n
Knowledge Level j	$K_j, \ \forall j = 0, 1, 2, 3$		

TABLE I: Attack Code

attackers directly manipulate the object quality reputation, whereas in indirect attacks, attackers boost their own feedback reputation and/or downgrade honest users' feedback reputation so that they can manipulate the object quality reputation more effectively.

- *Collusion or Non-Collusion* In simple attacks, dishonest feedbacks are provided independently. For example, when eBay users boost their reputation by buying and selling feedbacks, these feedbacks are often from independent sources. These attacks are referred to as the *non-collusion attacks*. In advanced attacks, the attacker control multiple user IDs, referred to as the *malicious users*, that coordinately insert dishonest feedbacks. These attacks are referred to as the *collusion attacks*. Roughly speaking, the non-collusion attacks are easier to address because (a) dishonest feedbacks that are far away from the honest opinions can be detected by various statistical methods [3], [12], [13] and (b) dishonest feedbacks that are close to the honest opinions usually do not cause much reputation distortion. Collusion attacks, which can be strengthened by the Sybil attack [18], can use more complicated attack strategies and often exploit vulnerabilities in both reputation systems and the defense solutions.
- *Knowledge Level of Attacker* Attacks can also be classified according to the amount of knowledge that the attackers need to obtain in order to launch an attack. We identify four knowledge levels as follows. In *level 0*, attacks are launched without any prior knowledge about the reputation systems. In *level 1*, attackers know the general principles of the reputation systems, such as more positive feedbacks usually leading to higher reputation. In *level 2*, attackers know the specific reputation aggregation algorithm and the defense mechanism that handles dishonest feedbacks. In *level 3*, attackers can obtain or estimate the parameters (e.g. detection threshold) used in the reputation systems, and adjust their attack strategies accordingly.

Given the above classification, we can describe the primary features of an attack in a concise way. A code is assigned to each category, as shown in Table I. For example, the self-promoting attack in [19] is a combined direct and indirect attack, targeting the objects, using colluded malicious user IDs, based on level 1 knowledge of the system. The code "$T_oA_{d,ind}C_yK_1$" is assigned to this attack. The RepTrap attack in [17], which will be introduced in Section IV-A, is a combined direct and indirect attack, targeting both objects and system, using colluded malicious user IDs, based on level 3 knowledge of the system. The code “$T_{o,s}A_{d,ind}C_yK_3$” is assigned to this attack.

In practical systems, feedbacks often contain both ratings and reviews. It is important to point out that the detection of untruthful reviews is very challenging. There are even online instructions on how to write an untruthful review that looks real and convincing [20]. In fact, one carefully written untruthful review can be more misleading than one dishonest rating. From the defense points of view, ratings, which are numerical, are easier to handle than reviews. In this paper, we focus on the dishonest rating problem, which is closely related to signal processing. The solutions that address dishonest ratings and the solutions that address dishonest reviews are surely compatible with each other.

As a summary, we would like to re-emphasize some key concepts as follows. In the attacks against reputation systems, **attacker** controls multiple **malicious user IDs** that insert **dishonest feedbacks** to the reputation systems, aiming to mislead the **reputation scores** of **target objects**. The reputation score of an object is also referred to as the **object quality reputation**. Many reputation systems compute the **feedback reputation** of users who provide feedbacks. The attacker can manipulate such feedback reputation in order to mislead the object quality reputation more effectively. In this paper, we focus on dishonest rating problems.

III. Competition Between Defenses and Attacks

When reputation systems were not as popular as they are today, there was no incentive for attackers to manipulate the reputation scores. At this stage, the reputation scores were often simply calculated as the average of all feedbacks (i.e. ratings values) or the number of positive feedbacks minus the number of negative feedbacks.

As the reputation systems became increasingly important in user decision making process, straightforward attacks emerged, followed by defense mechanisms that discovered these attacks, followed by advanced attacks that defeated the defense, followed by more sophisticated defense, and so on. There is surely an “arms race” or “competition” between the attack and the defense.

In this section, we synthesize such competition according to how attack and defense evolve from simple forms to sophisticated forms. The discussion is guided by Figure 1 that shows the attack strategies on the left hand side and the defense strategies on the right.

A. Whitewashing & Traitor Attacks

A reputation score is associated with the identity of an object. When the object can easily change its identity, *whitewashing attacks* can be launched. In particular, an object (e.g. a seller at

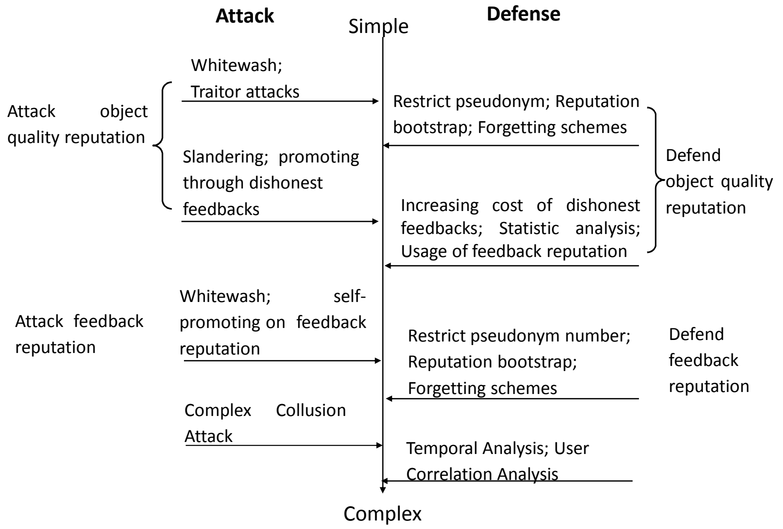
Attack
Simple
Defense
Whitewash; Traitor attacks
Attack object quality reputation
Slandering; promoting through dishonest feedbacks
Restrict pseudonym; Reputation bootstrap; Forgetting schemes
Defend object quality reputation
Increasing cost of dishonest feedbacks; Statistic analysis; Usage of feedback reputation
Whitewash; self-promoting on feedback reputation
Attack feedback reputation
Restrict pseudonym number; Reputation bootstrap; Forgetting schemes
Defend feedback reputation
Complex Collusion Attack
Temporal Analysis; User Correlation Analysis
Complex

Fig. 1: Evolution of Attacks and Defenses in Online Reputation Systems

eBay) that continuously receives negative feedbacks (due to its bad behaviors) is assigned a low reputation score. When the reputation score is low, this object may “recover” reputation through two ways: whitewashing and traitor [19]. In whitewashing, the object simply discards its current ID, re-enters the system by registering a new ID that has a clean history and fresh initial reputation score. In *traitor attacks*, the object restores reputation by performing good behaviors until the reputation score is recovered and then behaves badly again. In other words, it behaves well and badly alternatively, aiming to maintain the reputation score above certain level. For example, some sellers on eBay behave well in many small transactions until they build up high reputation, and then cheat on transactions in which expensive items are sold.

B. Defense against Whitewashing & Traitor Attacks

Some reputation systems are immune to whitewashing/traitor attacks. For example, in the Amazon product rating system, the object ID is just the name of the product, which cannot be changed easily. In other systems, the defense is carried out from two angles.

- Many reputation systems restrict the number of online pseudonym that can be obtained by their users. They increase the cost of acquiring a new user ID by binding user identities with IP address [21] and requiring entry fees [22]. Furthermore, reputation bootstrap studies suggest reasonably assigning reputation for a newcomer, such as low initial reputation [23] and initial reputation based on majority behavior [24].
- To prevent the traitor attacks, many reputation systems control the influence of historical behaviors through forgetting schemes. In [25], only the most recent feedbacks are considered in

reputation calculation. This scheme, however, raises wide concerns because both good and bad behaviors are forgotten equally fast, which may help the attacker to re-gain reputation. Then, the fading factor [12] is introduced to gradually reduce the weights of feedbacks provided long time ago. Furthermore, the system can count users' good behavior and bad behavior asymmetrically. The adaptive forgetting scheme in [26] makes good reputation be built up through consistent good behaviors but can be easily ruined by only a few bad behaviors.

C. Attacking Object Quality Reputation through Dishonest Feedbacks

When attackers cannot gain big advantages through whitewashing/traitor attacks, they naturally switch their focus on inserting dishonest feedbacks. The attacker can register a large number of user IDs, referred to as malicious users, and conduct either *slandering attack* or *promoting attack*. In the slandering attack, malicious users aim to downgrade the reputation of target objects by inserting negative feedbacks to them. In the promoting attack, malicious users aim to boost the reputation of target objects by providing positive feedbacks to them.

D. Detecting and Handling Dishonest Feedbacks

The defense schemes against slandering/promoting attacks are designed from three perspectives.

- Increasing the cost of dishonest feedbacks: Policies are in place that require the users to have certain credentials in order to provide feedback. The credentials can be a record of real transaction, such as at eBay, Amazon, and App stores [15].
- Detection of dishonest feedbacks: There are defense schemes studying statistic features of feedbacks. Most of them detect dishonest feedbacks based on the *majority rule*, which considers the feedbacks far away from majority's opinions as dishonest feedbacks. For example, in a Beta-function based approach [12], a user is determined as a malicious user if the estimated reputation of an object rated by him/her lies outside q and (1-q) quantile of his/her underlying feedback distribution. An entropy based approach [13] identifies feedbacks that bring significant changes in the uncertainty in feedback distribution as dishonest feedbacks. In [3], dishonest feedback analysis is conducted based on Bayesian model.
- Mitigating effect of dishonest feedbacks: Feedback reputation is proposed to measure users' reliability in terms of providing honest feedbacks. The reputation of an object is often calculated as the weighted average of all feedbacks (i.e. ratings), whereas the weight of a feedback is determined by the feedback reputation of the user who provides this feedback. As a consequence, feedbacks provided by users with low feedback reputation will have less impact on the object quality reputation. To compute the *feedback reputation score* of

a user, various methods have been developed. Iteration refinement approach proposed in [27] computes a user's judging power (i.e. weight of this user's feedbacks in the feedback aggregation algorithm) as the inverse of the variance in all of this user's feedbacks. In [28], personalized trust is introduced to measure the reliability of feedbacks. Note that not all schemes use the term "feedback reputation", but they all have certain quantitative measures of user reliability/honesty. To simplify the discussion, such quantitative measures are referred to as the user feedback reputation in general.

The above defense approaches are effective against straightforward dishonest feedbacks. However, smart attackers can exploit these defense schemes, and find ways to mislead the computation of user feedback reputation and the detection of dishonest feedbacks.

E. Attacks on Feedback Reputation and Corresponding Defense

When attackers realize that the system is computing their feedback reputation, they will try to increase their feedback reputation such that their dishonest feedbacks can carry larger weights. To achieve this, they can do either whitewashing or self-promoting on their feedback reputation. In whitewashing, after their feedback reputation is very low, they re-enter the system as new users. This is similar to an object re-entering the system (see Section III-A), and can be addressed by the defense methods discussed in Section III-B.

Self-promoting is a more serious threat. Malicious user IDs can provide honest feedbacks to the objects that they do not care, accumulate high feedback reputation, and then provide dishonest feedbacks to their target objects. Note that this type of attacks is only effective against reputation systems with user feedback reputation.

The self-promoting attack and the traitor attack described in Section III-A are closely related. They use the same approach (i.e. behaving well and badly alternatively) to achieve different goals. The former directly targets the feedback reputation of users (i.e. indirect attack), whereas the latter targets the reputation score of objects (i.e. direct attack). Therefore, the various forgetting schemes introduced in Section III-B can be applied on user feedback reputation and address the self-promoting attack.

F. Complicated Collusion Attacks

In the attacks discussed above, the malicious user IDs, which are under the control of the attacker, work together to manipulate the reputation score of target objects and/or promote their own feedback reputation. This is the simplest form of collusion, in which each malicious user performs similar tasks.

To strengthen the attacks and avoid being detected, the attacker makes malicious users IDs collaborate in more complicated ways.

- In the oscillation attack [25], malicious user IDs are divided into different groups and each group plays a dynamically different role. At a given time, some groups focus on providing dishonest feedbacks to target objects while other groups focus on improving their feedback reputation by providing honest feedbacks to objects that they do not care. The roles of those groups switch dynamically.
- In the RepTrap attack [17], malicious user IDs coordinately break the “majority rule” for some objects, by making the majority feedbacks be dishonest feedbacks for these objects. These objects are referred to as *traps*. In this attack, malicious users take turns to make one trap after another. By doing so, they can increase their own feedback reputation as well as reduce honest users’ feedback reputation. The details will be presented in Section IV-A.

In both attacks, the malicious users carefully coordinate their behaviors and can cause more damage to the system than simple collusion. Of course, to achieve complicated collusion, the attacker needs to know more about the objects and users, i.e. higher level of knowledge of the system.

G. Defense against Complex Collusion Attacks

To handle attacks with complicated collusion, defense schemes [14], [29] have been proposed from two new angles: *temporal analysis*, which explores the rich information in time domain (e.g. time when feedbacks are provided, changing trend of rating values, etc), and *user correlation analysis*, which aims to find behavior patterns of malicious users. These advanced defense schemes are compatible with most of the earlier defense schemes. They have shown promising results when tested against real user data. The details of these schemes will be presented in Section IV-B and Section IV-C.

H. Summary

We have seen that the attacks have evolved from (a) simple re-entering system (i.e. whitewashing) and dynamic behavior changing (i.e. traitor), which aim to maintain reputation score after conducting bad behaviors, to (b) inserting dishonest feedbacks (i.e. slandering/promoting), which aims to mislead the evidence aggregation algorithm, and to (c) manipulating the feedback reputation, which is the defense mechanism used to detect dishonest users. The attacks also evolved from (d) simple collusion, in which malicious users have similar behaviors, to (e) complicated collusion, in which malicious users carefully coordinate their tasks and the timing to conduct these tasks.

Attack Defense	Attack object quality reputation directly $(T_oA_dC_nK_0)$		Attack object quality reputation through dishonest feedbacks $(T_oA_dC_{n,y}K_1)$		Attack on feedback reputation $(T_oA_{ind}C_{n,y}K_{1,2})$		Complicated collusion attacks $(T_{o,s}A_{d,ind}C_yK_3)$	
	Whitewash [19]	Traitor [19]	Slander [19]	Promote [19]	Whitewash [19]	Self-promote [19]	Oscillation attack [24]	Rep- trap [17]
Restrict the number of pseudonym ([20]- [21])	Y	N	N	N	Y	N	Y	Y
Reputation bootstrap ([22]- [23])	N	Y	N	N	N	Y	N	N
Forgetting scheme ([24], [12])	N	P/F	Y	Y	N	P/F	P	P
Adaptive forgetting [25]	N	Y	F	N	N	Y	P	P
Increasing cost of dishonest feedbacks[15]	N	N	Y	Y	N	N	Y	Y
Statistical analysis based on majority rule ([3][12][13])	N	N	Y	Y	N	N	N	F
Usage of feedback reputation ([14][26])	N	N	Y	Y	N	N	N	F
Temporal analysis [27]	N	Y	Y	Y	N	Y	Y	P
Combination of Temporal and User Correlation analysis[28]	N	Y	Y	Y	N	Y	Y	Y
Others: network attacks, such as DDoS.								

Notation:
Y: the defense scheme is effective in defending the corresponding type of attacks;
N: the defense scheme is not effective in defending the corresponding type of attacks;
P: the defense scheme is partially effective (effective for some specific attacks or in some specific cases);
F: the defense scheme facilitates or even strengthens the corresponding type of attacks.

Fig. 2: Summary of Attack and Defense Strategies

On the other hand, the defense approaches have evolved from (a) policies that remove the advantage of being a new user, to (b) complicated signal processing techniques that detect abnormal feedbacks, to (c) the usage of feedback reputation, and to (d) analysis in time domain and on correlation among users.

The emerging attacks have driven the development of the new defense approaches, which in turn have stimulated the advanced attacks. This evolution and competition between attack and defense has been shown in Figure 1. In the next section, we will present several attack and defense approaches that are at the end of this evolution chain, which represent the current research frontier.

In addition, the Table in Figure 2 summarizes representative attack/defense strategies, their corresponding references, the attack codes that describe primary features of attacks, and whether a specific defense strategy is effective to handle a specific type of attacks.

Finally, we want to point out that traditional network attack strategies, such as DoS attacks, can also impact or even undermine reputation systems. Since there are plenty of studies on network attacks, they are not included in this paper.

IV. Representative State-of-the-art Attacks and Defenses

In this section, we will first present the RepTrap attack, an advanced collusion attack that effectively manipulates the reputation scores of the target objects with only a small number of malicious users. It also damages the reputation scores of many non-target objects, as well as the feedback reputation of honest users. Then, we introduce two types of advanced defenses: temporal analysis, and integrated temporal and correlation analysis.

A. *RepTrap Attack*

When feedback reputation is used, the attacker can launch the self-promoting attack (see Section III-E) in which malicious users increase their own feedback reputation. How about reducing the honest users' feedback reputation? Can the malicious users hurt honest users' feedback reputation and improve their own reputation at the same time? This is achieved in the RepTrap attack.

Approach: *First*, malicious users find some high quality objects that have a small number of feedbacks. *Second*, malicious users provide a large number of negative feedbacks to these objects such that these objects are marked as low-quality by the system. That is, the reputation system, relying on the majority rule to compute the object quality reputation, makes wrong judgment about the object quality. As a consequence, the system thinks that the malicious users' negative feedbacks agree with the object quality and increases their feedback reputation. The system will also think that the feedbacks from honest users disagree with the object quality, and will reduce these honest users' feedback reputation. These selected objects are called 'traps'. There is a detailed procedure on how potential traps are selected and how malicious users take turns to provide ratings to traps. The core of this procedure is an optimization problem, which is described in [17]. *Finally*, the malicious users insert dishonest feedbacks to the target objects, aiming to mislead their reputation scores.

Key Features: Compared with the self-promoting attack, the key advantage of the RepTrap is reducing the *cost of attack*, described by the number of malicious user IDs and the number of dishonest feedbacks used in the attack. For example, in the case study in [17], in order to make the system mistakenly mark an object as low quality, the attacker needs 7 malicious user IDs and 25 dishonest feedbacks in the self-promoting attack, but needs only 4 malicious user IDs and 18 dishonest feedbacks in the RepTrap attack.

RepTrap is effective under two conditions. *First*, there exist high quality objects with a small number of feedbacks. This is true in most practical systems. For example, many products at Amazon.com only have a few ratings. *Second*, the reputation system computes user feedback reputation and uses it in the reputation aggregation algorithm. RepTrap attackers also need to know some detailed information of the reputation system, which is knowledge level 3. The code of the RepTrap is "$T_{o,s}A_{ind}C_yK_3$".

Performance Demonstration: RepTrap was compared with the self-promoting attack and the simple slandering attack (i.e. directly give negative feedbacks to the target objects without considering feedback reputation), in a peer-to-peer (P2P) file sharing application. The simulator was created using models and parameters obtained from a popular P2P file-sharing system with more than 2 million registered users and 40,000 users online at peak time. In the simulation, there were 1000 honest users and 100 high quality files. The attacker's goal was to reduce the reputation score of the most popular file (i.e. the target object) below a threshold such that the system believed that the target object was a low quality file. There were 50 malicious user IDs.

If the malicious users start to attack soon after the target file is published (i.e. object entering the P2P system), they are more likely to be successful (i.e. achieve the attack goal). This is because there are few honest feedbacks at the beginning and even a small number of dishonest feedbacks can easily slander a good file. As the time goes, more feedbacks are provided by honest users, and it will be harder for an attack to succeed. Therefore, we change the starting time of the attack and show the probability that the attack goal is achieved.

The results are shown in Figure 3. The slandering attack can only be successful if it is launched within 2 days after the file is published. After 2 days, the success probability decreases sharply. The self-promoting attack has higher success probability than the slandering attack, but is much worse than RepTrap. For example, when self-promoting has only 10% success probability, RepTrap has 60% success probability. The results clearly show the advantage of reducing honest users' feedback reputation, which can only be achieved in RepTrap.

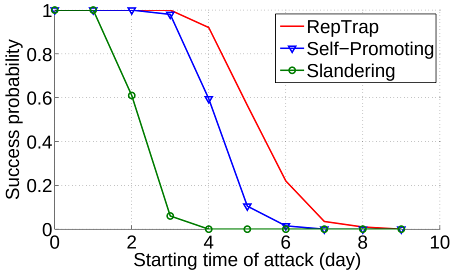

Fig. 3: Success probability when attacking the 1^{st} popular file

As a summary, the RepTrap attack hurts the feedback reputation of honest users and boosts the feedback reputation of malicious user IDs by undermining the system's estimation of object quality. In addition, RepTrap also hurts the users' incentive to contribute. Nobody would like to see that the files they published are marked as low quality files. When the system has many traps, fewer people will have strong motivation to publish high-quality files.

B. *Defense Scheme with Temporal Analysis*

In this subsection, we introduce a set of statistical methods that analyze time-domain features of rating signal [14]. These methods jointly detect the time intervals in which collusion attacks are highly likely present. Based on such detection, a trust-assisted reputation aggregation algorithm is designed. When tested against attack data from real human users, such scheme demonstrated large advantages over the defense schemes that only consider statistics of rating values but ignore temporal information.

Approach: In the first step, raw ratings are analyzed by four detectors.

- When the attackers insert unfair ratings, they may cause an increase in the rating arrival rate. Thus, the arrival rate detector is designed to detect sudden increase in the number of users who rate the object. More specifically, we construct signal R as the number of ratings per day. Assume that a sliding window contains $2W$ data samples from R. Let X_1 denote the first half data in the window and X_2 denote the second half data in the window. Assume X_1 and X_2 follow Poisson distribution, with parameter λ_1 and λ_2 respectively. The generalized likelihood ratio test (GLRT) is derived to determine whether there is an arrival rate change at the center of the window, i.e. $\lambda_1 \neq \lambda_2$. We then construct the arrival rate change curve by plotting the value of the detection function for each sliding window versus the center time of the sliding window. If the attack occurs between time t_1 and t_2, the detection curve may show two peaks at t_1 and t_2.
- Using a similar procedure, a *mean change detector* is developed to detect sudden changes

in the mean of rating values. This is because the primary goal of the attacker is to boost or downgrade the aggregated reputation score that is closely related to the mean of rating values.

- Since a large number of unfair ratings can result in a change in the histogram of overall rating values, a *histogram change detector* is adopted.
- Let $E(x(n))$ denote the mean of $x(n)$, where $x(n)$ denote the rating values. When there is no colluded users, ratings received at different time (also from different users) should be independent. Thus, $(x(n) - E(x(n)))$ should approximately be a white noise. When there are colluded users, $(x(n) - E(x(n)))$ is not white noise any more. Instead, the ratings from colluded users can be viewed as a *signal* embedded in the white noise. Based on the above argument, we develop an unfair rating detector as follows. The ratings in a time window are fit into an autoregressive (AR) signal model. The model error is examined. When the model error is high, $x(n)$ is close to a white noise, i.e. honest ratings. When the model error is small, there is a *signal* present in $x(n)$ and the probability that there are colluded users is high.

The above detectors are the core of temporal analysis, but cannot completely solve the problem. In the **second step**, outcomes of the above four detectors are combined to detect the time intervals in which unfair ratings are highly likely present. These intervals are called suspicious intervals. In the **third step**, a trust manager is used to determine how much an individual user can be trusted, depending on how many ratings provided by this user are in the suspicious intervals. **Finally**, highly suspicious ratings are removed from the raw ratings by a rating filter, and remaining ratings are aggregated based on the user trust information.

Performance Demonstration: The scheme with temporal analysis and trust management was compared with three other schemes: (1) simple averaging, without defense, as described at the beginning of Section III, (2) beta-function based filtering [12], belonging to the defenses described in Section III-D, and (3) combination of the four detectors without trust management.

The above four schemes were tested against real rating data collected from BestBuy.com and real user attack data collected from a cyber competition [17]. Notice that the rating value is between 1 and 5. In the cyber competition, ratings for 9 flat panel TVs with similar features were collected as normal ratings. With no more than 50 malicious user IDs, the participants were asked to boost the ratings of two products and reduce the ratings of another two products through unfair ratings. The attacks causing largest bias on products' ratings are most successful attacks. Through the cyber competition, 252 eligible submissions of unfair ratings were collected. Details can be seen in [17]. From all collected attacks, for each defense scheme, we picked top 20 attacks that caused the largest bias in reputation scores of the target objects. Here, the bias was defined as the difference between

	Against top 20 strongest attacks			Against all attacks
	Min	Max	Average	Average
Simple Average	0.890	1.22	1.01	0.600
Beta-function	0.894	1.27	1.02	0.538
Detector Combination	0.469	0.579	0.495	0.263
The Proposed Scheme	0.229	0.341	0.264	0.113

TABLE II: Bias in reputation scores resulting from top 20 strongest attacks and all attacks.

the aggregated reputation score with unfair ratings and the aggregated reputation score without unfair ratings. Of course, the top 20 attacks for different defense approaches were different. Table II shows the bias when the four defense schemes were tested against their own top 20 attacks (which represented the worst case performance), as well as against all attacks (which was the average case performance). Compared with the beta-function based method, the temporal analysis alone (i.e. detectors only) could reduce the bias by more than 50%. When temporal analysis and trust management were combined, the bias was reduced by 73.15% to 79.1%.

Key Features: The temporal analysis has two interesting features. First, it focuses on the consequence of the attack, instead of the specific attack methods. In other words, no matter what specific attack methods are used, they will introduce some kinds of change in the rating signal. Therefore, this method can address various types of attacks. Second, it investigates history of users through trust evaluation, which effectively reduces the "false alarms". Note that there must be honest users who rate in the suspicious time intervals, and it is not practical to remove all the ratings in the suspicious intervals.

C. Defense Scheme that Combines Temporal and Correlation Analysis

Trust management, although can capture the history of users' rating behavior, has certain shortcomings. First, there are various attacks against trust management, aiming to make honest users have lower trust and dishonest users have higher trust [30]. Sometimes, using trust management can introduce new vulnerabilities to the system. Second, trust management only captures individual user's past behavior, but not the correlation among users' past behaviors.

In this subsection, we introduce a defense method that identifies malicious users by combining temporal analysis and user correlation analysis [29]. Compared with the defense in Section IV-B, this method adopts different detector for temporal analysis and uses correlation analysis to investigate users' past behavior. This scheme is named as joint temporal analysis and user correlation analysis (TAUCA).

Approach: TAUCA contains three main components: (a) change detection, (b) user correlation calculation, and (c) malicious user group identification. In many practical reputation systems, the objects have intrinsic and stable quality, which should be reflected in the distribution of normal ratings. Therefore, change detection is an intrinsically suitable tool for temporal analysis. Although the previous change detectors can catch sudden changes, they are not effective when the malicious users introduce changes gradually. For example, malicious users can first aim to increase the reputation score from 4.0 to 4.1, then from 4.1 to 4.2, etc.

Therefore, TAUCA employs a change detector, which takes raw rating sequence (i.e. rating values ordered according to when they are provided) as inputs, sensitively monitors the changing trend, and detects changes either occurring rapidly or accumulated over time. Let y_k denote the k^{th} sample of the data sequence (i.e. rating sequence). Specifically, the decision function, which is revised from the basic CUSUM detector [31], is

$$g_k = max(g_{k-1} + (y_k - \mu_0 - \nu/2), 0) \ , \tag{1}$$

$$t_a = min\{k : g_k \geq \overline{h}\} \ , \tag{2}$$

where g_k and g_{k-1} are the decision functions at the k^{th} and $(k-1)^{th}$ data sample respectively, $\overline{h}$ is threshold, and μ_0 is the mean values of the rating sequence before attack. The value of ν is a parameter that determines the sensitivity of the detection. Smaller is the ν value, more sensitive is the change detector. Here, t_a is called *stopping time*, the time when the detector identifies a change and raises an alarm.

If the change detector is triggered by an object, this object is marked as *under attack*. The direction of the change, either boosting or downgrading, is called *attack direction*. Once the detector is triggered, TAUCA can estimate the starting time and the ending time of the change [29]. The time interval between the starting time and the ending time is called a *suspicious interval*.

Figure 4 illustrates the change detection process. The x-axis is the index of ratings. The upper plot shows the original rating sequence ordered according to the time when the ratings are provided. The y-axis is the rating value ranging from 1 to 5. The honest ratings are in blue color, whereas the malicious ratings are in red color. The middle plot shows the detection curves (g_k) of the change detector, as well as the detection threshold. The lower plot shows the suspicious intervals detected by the change detector. Note that once g_k increases above the threshold, the detector is triggered and an alarm is set off (meaning the detection of a change). Since the detector needs some time to respond, the time when the change starts (i.e. change starting time) is usually earlier than the alarm setting off time. Similarly, when g_k drops below the threshold, the alarm is set on, which

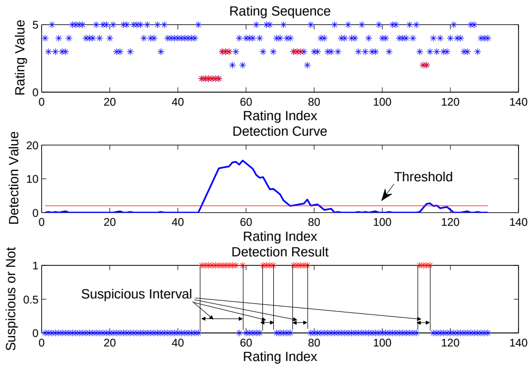

Fig. 4: Demonstration of Change Detector in TAUCA

happens after the change is over. In [29], the change starting/ending time is estimated based on the alarm set off/on time.

After the change detection, TAUCA moves from time domain to user domain. TAUCA analyzes correlation among *suspicious users*, defined as users who rate in the suspicious intervals. We have observed that larger correlation exists among colluded malicious users. After correlation analysis, suspicious users are separated into different groups/clusters. Finally, the malicious user group identification module determines which group is composed of colluded malicious users.

Performance Demonstration: TAUCA was tested against real user attack data collected from CANT cyber competition [32], in which participants were asked to downgrade reputation score of one object by providing dishonest ratings through less than 30 controlled malicious user IDs. TAUCA was compared with the beta-function based defense scheme [12]. It was observed that TAUCA was much more effective in terms of detecting malicious users. For example, when there were 20 malicious users, TAUCA detected 92% malicious users with 4% false alarms, whereas beta-function based scheme only detected 67% malicious users with 57% false alarms.

V. Conclusion

In this paper, we conducted an in-depth investigation on the competition between attack and defense for feedback-based reputation systems, as well as introduced representative attack/defense approaches. This competition will surely continue to evolve and lead to new research challenges. Since there is no real "conclusion" for this evolvement, we conclude this paper by pointing out some useful resources and research directions. In this research, it is important to understand and obtain data describing normal users' rating behavior as well as malicious users' attacking behavior. Several dataset can be very useful: Netflix Prize [33], CANT Competition [32], and Epinions datasets [34]. In the future, it will be important to study how normal users' behavior, such as whether purchasing a product, is affected by the bias in reputation scores. This requires joint study on security issues and user behavior modeling in social networks, an interesting direction for signal processing researchers. In addition, there are still a large number of dishonest ratings /reviews and misleading reputation scores in current commercial reputation systems, such as eBay and Amazon. This is partially because many advanced defense approaches have not made their ways into the commercial systems. It will be important to develop tools such that users can directly benefit from the research on defense technologies, and increase their trust on online reputation systems.

REFERENCES

[1] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[2] C. Dellarocas, "The digitization of word-of-mouth: Promise and challenges of online reputation systems," *Management Science*, vol. 49, no. 10, pp. 1407–1424, October 2003.

[3] L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt, "Ratings in distributed systems: A bayesian approach," in *Proc. of the Workshop on Information Technologies and Systems (WITS)*, 2001.

[4] B. Yu and M. Singh, "An evidential model of distributed reputation management," in *Proc. of the Joint Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 294-301 2002.

[5] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proc. of the 7th Int. Conf. on World Wide Web (WWW)*, 1998, http://dbpubs.stanford.edu:8090/pub/1998-8.

[6] J. Sabater and C. Sierra, "Social regret, a reputation model based on social relations," *SIGecom Exchanges*, vol. 3, no. 1, pp. 44–56, 2002.

[7] D. Houser and J. Wooders, "Reputation in auctions: Theory, and evidence from ebay," *Journal of Economics and Management Strategy*, vol. 15, pp. 353–369, June 2006.

[8] A. Harmon, *Amazon glitch unmasks war of reviewers*, the New York Times, February 14,2004.

[9] J. Brown and J. Morgan, "Reputation in online auctions: The market for trust," *California Management Review*, vol. 49, no. 1, pp. 61–81, 2006.

[10] *Taobao fights reputation spam in e-business boom*, BeijingToday, Sept. 12 2009, http://www.beijingtoday.com.cn/feature/taobao-fights-reputation-spam-in-e-business-boom.

[11] D. Cosley, S. Lam, I. Albert, J. Konstan, and J. Riedl, "Is seeing believing? how recommender systems influence users opinions," in *Proc. of CHI 2003 Conf. on Human Factors in Computing Systems, Fort Lauderdale, FL*, 2003, pp. 585–592.

[12] A. Josang and R. Ismail, "The beta reputation system," in *Proc. of the 15th Bled Electronic Commerce Conf.*, 2002.

[13] J. Weng, C. Miao, and A. Goh, "An entropy-based approach to protecting rating systems from unfair testimonies," *IEICE TRANSACTIONS on Information and Systems*, vol. E89–D, no. 9, pp. 2502–2511, Sept 2006.

[14] Y. Yang, Y. L. Sun, S. Kay, and Q. Yang, "Defending online reputation systems against collaborative unfair raters through signal modeling and trust," in *Proc. of the 24th ACM Symposium on Applied Computing*, Mar 2009.

[15] A. Joang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, Mar 2007.

[16] P. Resnick, R. Zeckhauser, R. Friedman, and K. Kuwabara, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, Dec. 2000.

[17] Y. Yang, Q. Feng, Y. Sun, and Y. Dai, “Reputation trap: An powerful attack on reputation system of file sharing p2p environment,” in *the 4th Int. Conf. on Security and Privacy in Communication Networks*, 2008.

[18] J. R. Douceur, “The sybil attack,” in *in Proc. for the 1st Int. Workshop on Peer-to-Peer Systems (IPTPS)*, Springer Berlin / Heidelberg, March 2002, pp. 251–260.

[19] K. Hoffman, D. Zage, and C. Nita-Rotaru, *A survey of attack and defense techniques for reputation systems*, technical report, Purdue Univ., 2007. http://www.cs.purdue.edu/homes/zagedj/docs/reputation survey.pdf.

[20] M. Hines, *Scammers gaming YouTube ratings for profit*, http://www.infoworld.com/d/security-central/scammers-gaming-youtube-ratings-profit-139.

[21] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, “A calculus for access control in distributed systems,” *ACM Transactions on Programming Languages and Systems*, vol. 15, no. 4, pp. 706–734, 1993.

[22] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, “Free-riding and whitewashing in peer-to-peer systems,” in *3rd AnnualWorkshop on Economics and Information Security (WEIS2004),May*.

[23] G. Zacharia, A. Moukas, and P. Maes, “Collaborative reputation mechanisms for electronic marketplaces,” in *Proc. of the 32nd Annual Hawaii Int. Conf. on System Sciences*, 1999.

[24] Z.Malik and A. Bouguettaya, “Reputation bootstrapping for trust establishment among web services,” *IEEE Internet Computing*, vol. 13, no. 1, pp. 40–47, 2009.

[25] M. Srivatsa, L. Xiong, and L. Liu, “Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks,” in *Proc. of the 14th Int. Conf. on World Wide Web*, May 2005.

[26] Y. L. Sun, Z. Han, W. Yu, and K. Liu, “Attacks on trust evaluation in distributed networks,” in *Proc. of the 40th annual Conf. on Information Science and Systems (CISS),Princeton, NJ*, March 2006.

[27] P. Laureti, L. Moret, Y.-C. Zhang, and Y.-K. Yu, “Information filtering via iterative refinement,” in *Europhysics Letters*, vol. 75, no. 6, 2006, pp. 1006–1012.

[28] J. Zhang and R. Cohen, “A personalized approach to address unfair ratings in multiagent reputation systems,” in *Proc. of the Fifth Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS) Workshop on Trust in Agent Societies*, 2006.

[29] Y. Liu and Y. Sun, “Anomaly detection in feedback-based reputation systems through temporal and correlation analysis,” in *Proc. of 2nd IEEE Int. Conf. on Social Computing*, Aug 2010.

[30] Y. Sun, Z. Han, and K. J. R. Liu, “Defense of trust management vulnerabilities in distributed networks,” *IEEE Communications Magazine*, vol. 46, no. 2, pp. 112–119, Feb 2008.

[31] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, no. 1/2, pp. 100–115, Jun 1954.

[32] *CANT Cyber Competition*, http://www.ele.uri.edu/nest/cant.html.

[33] *Netflix Dataset*, http://www.netflixprize.com/.

[34] *Epinions Dataset*, http://www.trustlet.org/wiki/Epinions_dataset.