Analyzing the vulnerability of a certain Reputation-Based Trust Management for P2P Networks

By Teng Teng
UPI: tten007
Student ID: 3232667

Abstract

Ali and his colleagues developed A Reputation-Based Trust Management System for P2P Networks. They claimed that the system could be highly effective in preventing the spread of malicious content in P2P networks. And they also test their system by various simulation experiments and found the Results support their protocol. This article tries to analysis the vulnerability of their system both in design level and in technical level. In design level, several presumptions that the system based on are not quite fit for the real world. In technical level, this paper proposes a new possible attack that based on the combination 0f 3 of the possible attacks suggested by the protocol authors. In some circumstance, this new attack can bypass the protection of the system.

,

1 Instruction

Ayd_n Selc¸uk Ersin Uzun Mark Res developed a Reputation-Based Trust Management System for P2P Networks [1](I will abbreviate it as RBTMS in the remain part of this paper). They try to develop a system for a p2p environment such as a Gnutella- or Kazaa-like system. RBTMS use the reputation of the peers in the p2p network to judge if the files they provide are trustable. To achieve this goal, each peer in the p2p network keeps a local copy of the information of the peers they contacted with. The information includes other peers' trust vector, distrust vector and credibility vector. When a certain peer tries to find a special file, it will send queries for it first. Then it will check the returned queries. If it has local information on a sufficient number of the peers who provide a file version, the trust score for that version is calculated from the local trust ratings. Otherwise, a group of peers who provide the file but were not known previously are selected and a trust query for them is issued. The responses to the trust query are weighed according to the credibility ratings of the respondents and a trust score for that file version is calculated. At the end, the trust scores of the different versions are compared and the one with the highest trust score is selected for download.

The protocol authors suggest 4 possible attacks in their article [1] and add 1 more in their technique report [2].

They are as following,

Naïve: who responds to every query with a malicious version of the requested file. [1]

Hypocritical: who acts like a reliable peer most of the time but occasionally tries to send a malicious file[1]

Collaborative: who collaborate with each other in trust queries, expressing a positive opinion for malicious peers and a negative opinion for others

Pseudospoofing: who change their pseudonym periodically to escape recognition. These attackers are the hardest to detect and their prevention is possible only after honest peers build a sufficient level of trust among themselves. [1]

Pseudospoofing with collaborators: where the pseudospoofing peers are supported by a group of “collaborators” who normally act as trustworthy peers and build trust in their communities, but give their strongest support to their malicious peers when they receive a relevant trust query. [2]

The authors of the protocol also designed some simulation tests and they report the result of those tests support the success of their protocol.

The RBTMS looks quite well organized up till now. However, it still has some serious vulnerable problems both in its design level and technical level. In design level, the presumptions the protocol based on are not quite practical in the real world. There are 3 presumptions will cause problems. The first is the presumption that users can detect attacks correctly and in a short time. The authors mentioned it as the limit of the protocol and claim that the protocol can work for those attacks can be discernible by the users. However, as the authors also mentioned, malicious content also include the virus such as VBS.Gnutella worm. Actually, the open and decentralized nature of Decentralized Peer to Peer (P2P) networks makes it extremely susceptible to malicious users spreading harmful content like viruses, Trojans [4]. However, viruses, Trojans normally are not easy to be recognized by common users in time. If the protocol cannot prevent these kinds of attacks, it cannot be useful in a real world because the attackers will turn to use those kinds of attacks in the future. And those attacks are more dangerous than those decoy files, which are easy to be discerned. The second presumption is that malicious file senders can be forgave by providing some benign files, just as providing community services. But in the flies share system, since download files cost almost nothing, the costs of the services are very low, so it cannot well prevent those malicious attackers to send viruses and Trojan horses again. The third presumption is that the authors of the protocol think their “simulations can be expected to give better results when run with a Zipf distribution since positive correlation among users' behavior would result in a more rapid trust establishment among the users in the same category.”[1] However, when we consider the attacks with cooperators, things will be different. , sometimes, category can cause worse result for the protocol because the effect of Collaborative attack can be reinforced because the relative smaller category size. In technical level, this paper

suggests a new possible attack, which based on the combination 0f 3 of the possible attacks suggested by the protocol authors. We will discuss the design level vulnerability in §2 and the technical level vulnerability in §3.

2 Design level vulnerability of RBTMS

The RBTMS is developed on several basic presumptions. However, some of those presumptions are not so practical for a real world network, we will analysis them as following,

2.1 Analyzing the presumption on the user abilities to find malicious file

The authors of RBTMS claimed, [their] "system in the end relies on the judgment of its users. Therefore, it can be effective only against attacks that are discernible by the user. Nevertheless, many attacks in P2P systems fall into this category, such as the common decoy files attacks [3]."

They show an example of a kind of attacks that are discernible by the user; the common decoy files attacks [3]. However, things are changing. If the attackers find one way cannot work, most possible they will try other ways. In that certain example from the authors, decoy files like empty files or wrong files maybe can not work by using the reputation system, but who can promise that those attackers will not use those files like lowered quality files or content changed files which are not so easy to be recognized by users? Attackers are quite flexible. Some virus attackers already "use previous tricks to a new level by expanding their infection potential, at least in theory (Simile.D), or by combining multiple best-tricks into one insidious package (Klez)."[5]

So even if the reputation based system can prevent those attacks that are easy to be recognized by the users, attackers will easily turn to the attacks that are not easy to be distinguished.

What's more, there are lots of other kinds malicious files can be used in the p2p networks. Aameek Singh and Ling Liu claimed that the open and decentralized nature of

Decentralized Peer to Peer (P2P) networks makes it extremely susceptible to malicious users spreading harmful content like viruses, Trojans or, even just wasting valuable resources of the network [4]. However, viruses, Trojans normally are not easy to be recognized by common users in time. They behave in stealth ways and can have a latent period.

One possible solution is binding a intrusion detect system which can detect most viruses and Trojan horses in time. However, it will significantly increase the cost of RBTMS.

What make things worse is that even the intrusion detect system cannot find those decoy files like lower quality files and content changed files we mentioned before. And those files are not easy to be detected by users in the first glance.

So relying on the presumption that users can judge possible attacks in time is unpractical.

2.2 Analyzing the presumption related to the policy of user reputation update

[That protocol]" enabling peers to cleanse their history by doing a reasonable Amount of community service after a bad deed."

But in the flies share system, since download files cost almost nothing, the cost of the serving some benign files are very low; the damage of virus can be very high. I.e. The Code Red did about $2.6 billion of damage to computer systems around the world. [7] Although as Lampson state that the cost of damages can be exaggerated [9], the damages can be still considerably. What's more, we are talking about a security system that will scarify some conveniences of users (i.e. User must give opinion to each file downloaded. Queries and answer those queries will slow down the whole system).That implies that users must have some important things need to be protected. The cost of damage can be high.

So the cost of providing some benign files cannot well prevent those malicious attackers to send viruses and Trojan horses again. And that means that to punish a malicious attack by forcing the attacker to provide benign files cannot sufficiently prevent attackers' bad behavior.

However, if we ban the users who send malicious files only several times, maybe we will hurt those innocent users who just was cheated by the malicious files since the system cannot tell a malicious attacker or an innocent user providing malicious file by mistake.

One possible improvement is that users can only clean their history for a certain times in a period.

2.3 Analyzing the presumption on the effect of category

The authors of RBTMS think their “simulations can be expected to give better results when run with a Zipf distribution since positive correlation among users' behavior would result in a more rapid trust establishment among the users in the same category.”[1] However, when we consider the attacks with cooperators, things will be different. Obviously, for cost reason, I agree with the authors of RBTMS that attackers normal cannot organize more than 10% cooperators in a real life network. [2] But attackers can manage to have higher ratio of cooperators in a certain category because the number of the users fall into a category must be smaller than that of the whole network. Especially when attackers can assume that the members of that category have much higher value than others, such as military or stock market related groups, the attackers and their cooperators can work only in that category and gain higher cooperator ratio.

As a result, sometimes, category can cause worse result for the protocol because the effect of Collaborative attack can be reinforced.

3 Technical level vulnerability of RBTMS

3.1 New attack based on the combination of the possible attacks suggested by the protocol authors

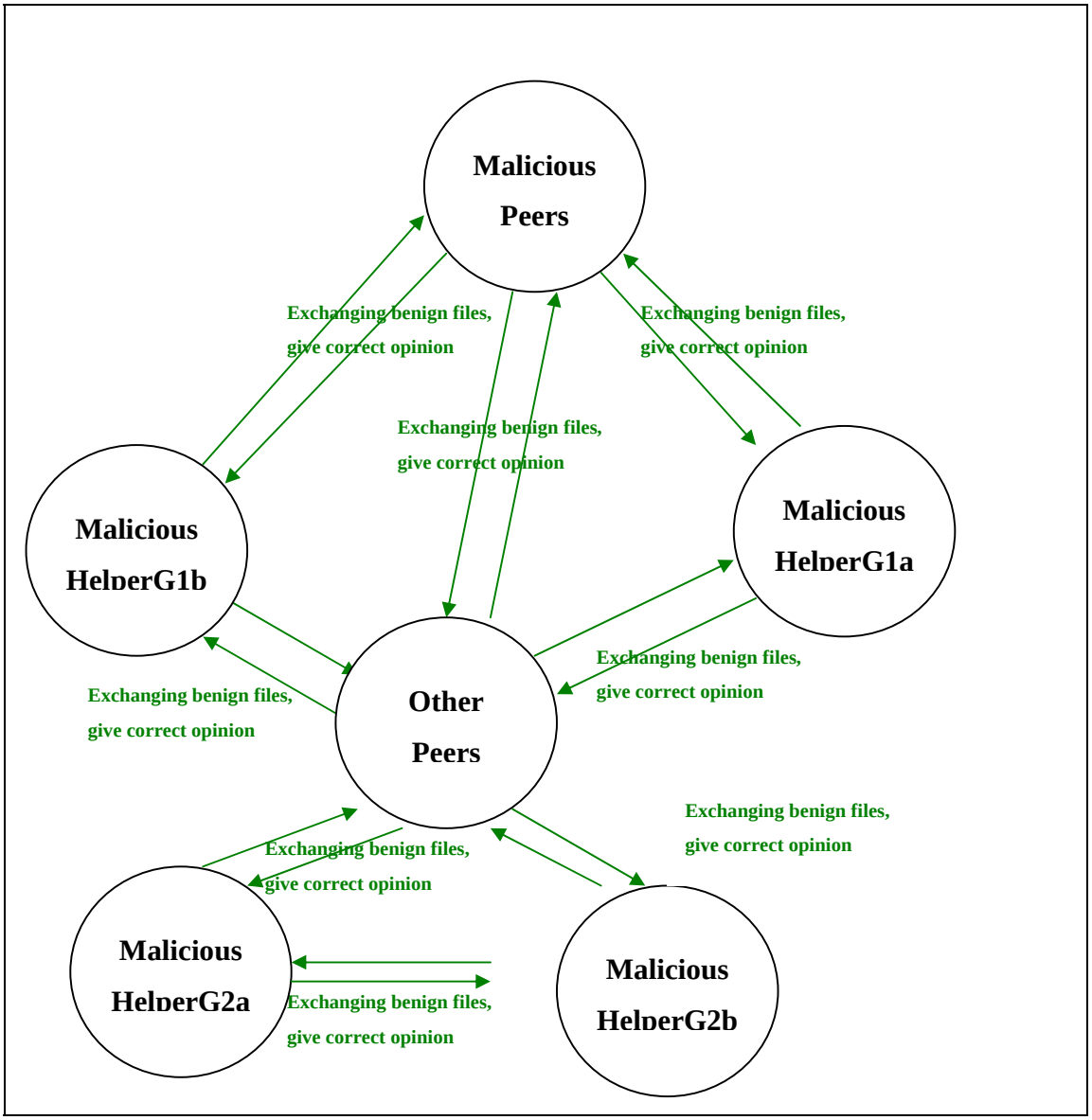
3.1.1 New attack description

The possible attack I propose is the combinations of the three basic types of attack described by the protocol author. The reason is that if the protocol can not resist a brand new attack, it is acceptable because it significantly increase the difficulty of attackers, but if the protocol can not resist a know attack which is easy to implement, or a simple

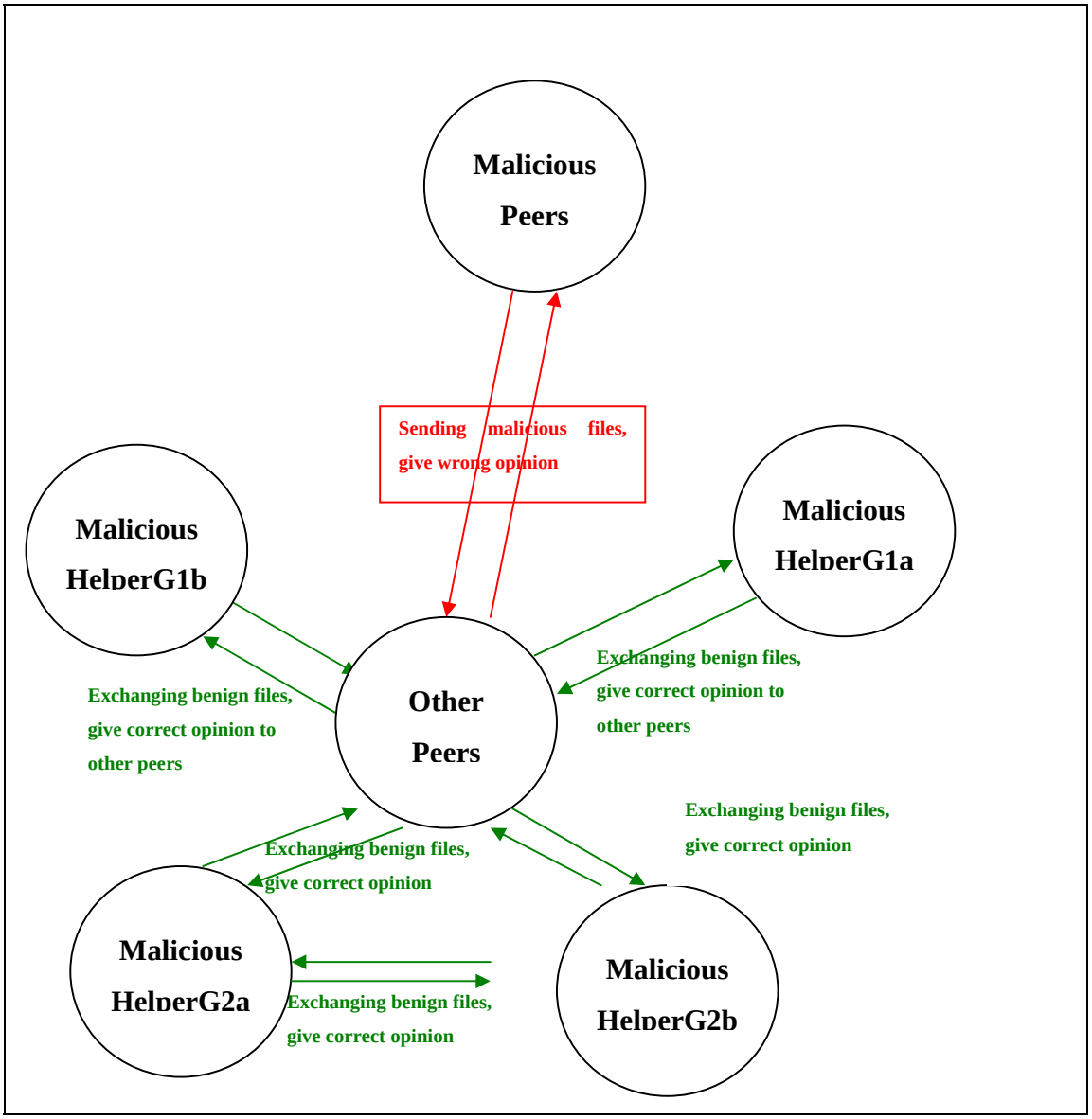
combination of the known attacks, it can not be adopted because the attackers need not pay much effort to turn to these kinds of new attacks in the future when they find some former attacks are invalid by the secure system.

The attack I suggest is the combination of hypocritical, collaborative and Pseudospoofing attacks, it works in the following way,

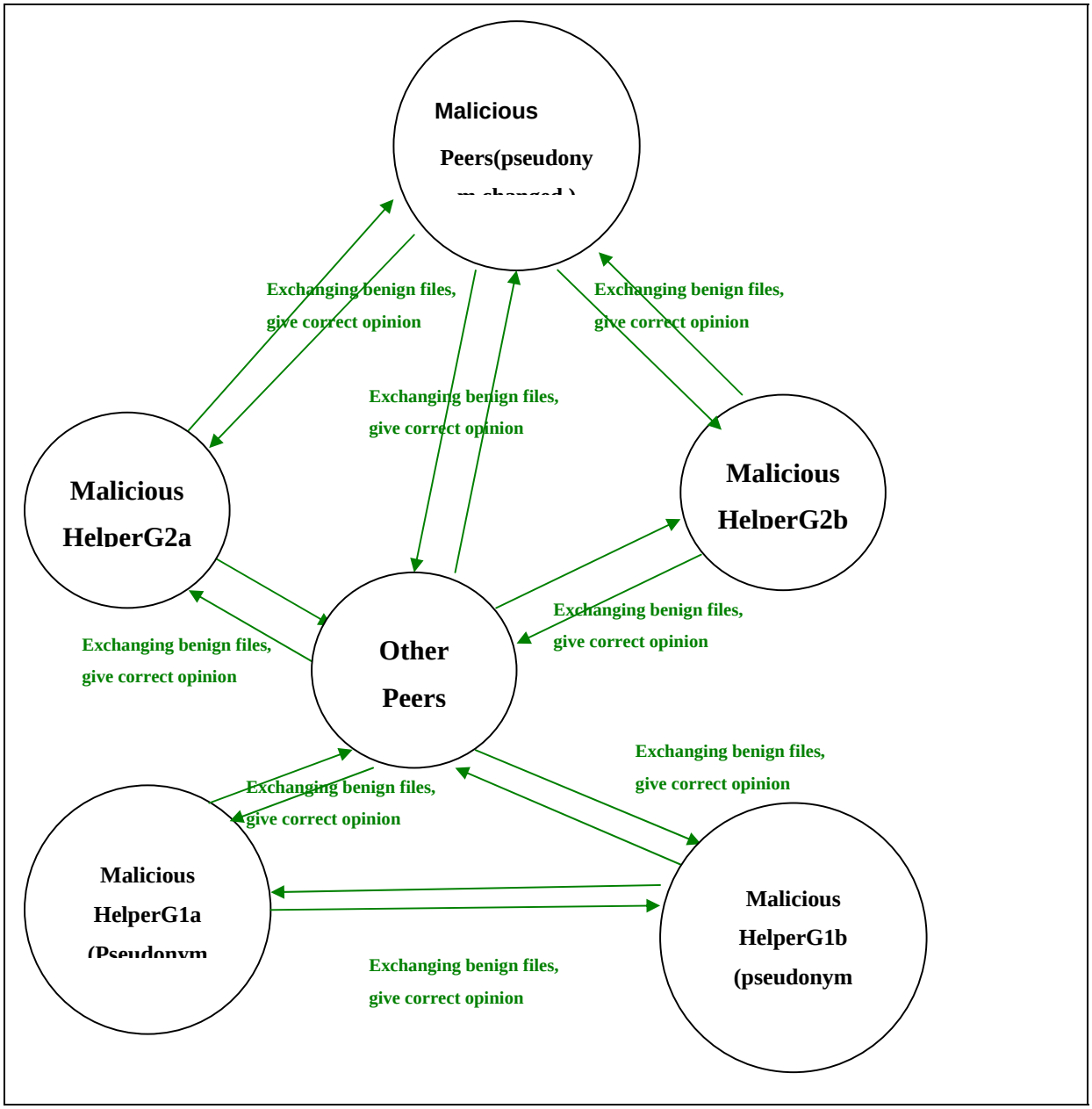
The malicious peer has two groups of cooperators. One group just accumulates their credibility rate without give opinions to the attacker; another group that gains enough credibility rates will give positive opinions to the attacker. When it reaches a time that attacker and his helpers cannot work well, they can be abandoned and the malicious peer with new name can establish his reputation by the first group rapidly. And another new helper group can be set up and accumulate reputation for future use. The mechanize of this kind of attack are show as following,



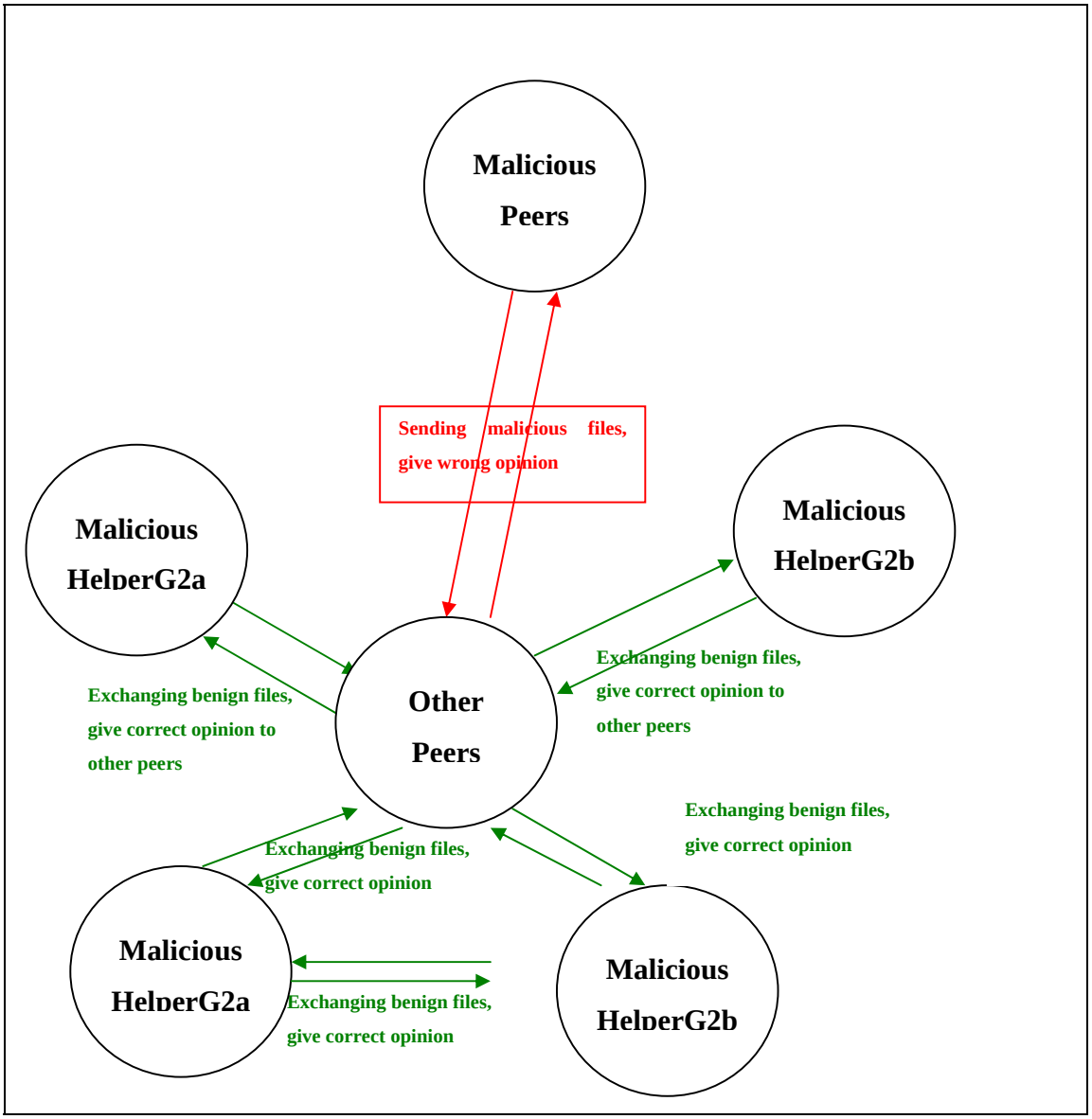
Step 1: all malicious peers and their 2 groups of helpers behave like benign users, they send and receive benign files between each other within their group and with other peers as well, give correct opinions. (For convenient, I only present 2 group members for each group, and there can be several Malicious Peers to increase the possibility of the download of the malicious files, I also present one node here for convenient) and build their trust/distrust vector. Group2 of helpers do not down load from malicious peer.



Step 2: Malicious peers begin to send the malicious files. Malicious helpers of group 1 stop downloading files from the malicious peers so they just behave like innocent users. But their trust score of the malicious peers remain the highest by the previous download. Malicious helpers of group 2 do not join the game. They just continuously accumulate their reputations.



Step 3: This step is quite similar to step 1. The difference is that the malicious peers are new peers compare to other peers. And Helpers from group2 download from the malicious peer continuously to fill the trust vector/distrust vector. And their high reputation will attract other peers to download benign files from malicious peers to accelerate the speed of accumulation of reputations for malicious peers.



Step 4: This step is very similar to step 2. After this step, the mechanism will repeat step 3 and 4 again and again.

3.1.2 Rationale for the new attack

3.1.2.1 Why the new attack is possible

Attackers are quite flexible. When they find their attacks cannot work, they will try other attack ways. Especially if the other ways are already know or can be created by just simply combinations of several kinds of known attacks.

Some virus attackers already “use previous tricks to a new level by expanding their infection potential, at least in theory (Simile.D), or by combining multiple best-tricks into one insidious package (Klez).”[5]

So if a security system says it can resist some known attacks, it should also be able to resist the combination of those know attack.

3.1.2.2 Why the new attack will success

Actually, even the combination of hypocritical and Pseudospoofing attacks can defeat the RBTMS when providing enough time for malicious peers to accumulate their reputations. It is because that after Pseudospoofing, the malicious peers will behave just like new peers. RBTMS has no mechanism to distinguish them. And then by hypocritical, the malicious peers will accumulate their reputations just as some good peers. RBTMS still has no mechanism to distinguish them in this step. After the malicious peers gain enough reputations, they can begin their malicious files sending again. Only after a period RBTMS can inhibit them but they can do Pseudospoofing and behave like a normal new comer again. The collaborative method is used to accelerate the reputation accumulate speed of the malicious peers. They can recommend normal peers to down load from the malicious peers by their high reputation.

In fact, before the malicious attack begins, to the RBTMS, malicious attackers and their helpers are all normal users. RBTMS cannot tell the difference between a new comer and a malicious attacker who changed his name (Pseudospoofing attack), and the malicious attacker can win reputation again by providing some benign files just as a normal new comer (hypocritical attacker). The malicious helper can help attacker

accumulate their reputation faster. (Collaborative attack). As a result, malicious attacker can cheat RBTMS again and again.

4 Conclusion

Although the RBTMS is organized in a quite elegant way, it still has vulnerabilities both in its design level and technical level. It shows a new example that how complex it is to design a reliable security system.

The vulnerability in design level comes from the 3 presumptions that the RBTMS based on.

The problem of the first presumption is that in the open and decentralized nature of Decentralized Peer to Peer (P2P) networks makes it extremely susceptible to malicious users spreading harmful content like viruses, Trojans. However, viruses, Trojans normally are not easy to be recognized by common users in time. And some decoy files other than virus such as lower quality files or content changed files are also not easy to be recognized by users at first glance. The problem in the second presumption is that in the flies share system, since downloading files cost almost nothing, the costs of the services are very low, so providing service cannot well prevent those malicious attackers to send viruses and Trojan horses again. The problem of the third presumption is that when we consider the attacks with cooperators, things will be different. Sometimes, category can cause worse result for the protocol because the effect of Collaborative attack can be reinforced.

The vulnerability in technical level comes from the new attack that is just a combination of the attacks that the RBTMS authors suggested. RBTMS has no mechanism to resist it.

So our conclusion is that RBTMS still cannot be used in practical because of those vulnerabilities.

Acknowledgements

I want to thank Prof. Thomborson, who gives me a lot of constructive suggestions. Also I want to thank Mr. Chris who checked this article and provided me with some valuable opinions.

References:

[1] A. Selcuk, E. Uzun, M. Pariente, "A Reputation-Based Trust Management System for P2P Networks", to appear in Proc. CCGRID 2004

[2] A. A. Selc¸uk, E. Uzun, and M. R. Pariente. Reputation based trust management for P2P networks. Technical Report BU-CE-0402, Department of Computer Engineering, Bilkent University, 2004.

[3] http://news.bbc.co.uk/2/hi/entertainment/2093931.stm

[4] Aameek Singh Ling Liu "TrustMe: anonymous management of trust relationships in decentralized P2P systems" , Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference, Coll. of Comput., Georgia Inst. of Technol., Atlanta, GA, USA

[5] Schreiner, K."New viruses up the stakes on old tricks" Internet Computing, IEEE, Volume: 6, Issue: 4, July-Aug. 2002 Pages:9 – 10

[6] Geer, D.; "Just how secure are security products?" Computer , Volume: 37 , Issue: 6 , June 2004 Pages:14 - 16

[7] Ford, R, "The wrong stuff? " Security & Privacy Magazine, IEEE , Volume: 02 , Issue: 3 , May-June 2004 Pages:86 - 89

[8] http://abc.net.au/news/indepth/yir2001/scitech2.htm

[9] Lampson, B.W.; "Computer Security in the Real World" Computer , Volume: 37 , Issue: 6 , June 2004 Pages:37 – 46