Journal of Information Processing Vol.20 No.2 378–385 (Apr. 2012)

[DOI: 10.2197/ipsjjip.20.378]

Regular Paper

Prevent Contents Leaking in P2P CDNs with Robust and Quick Detection of Colluders

Ervianto Abdullah[1,a) Satoshi Fujita[1]

Received: May 21, 2011, Accepted: November 7, 2011

Abstract: The objective of Peer-to-Peer Content Delivery Networks is to deliver copyrighted contents to paid clients in an efficient and secure manner. To protect such contents from being distributed to unauthorized peers, Lou and Hwang proposed a proactive content poisoning scheme to restrain an illegal download conducted by unauthorized peers, and a scheme to identify colluders who illegally leak the contents to such unauthorized peers. In this paper, we propose three schemes which extend the Lou and Hwang's colluder detection scheme in two directions. The first direction is to introduce an intensive probing to check suspected peers, and the second direction is to adopt a reputation system to select reliable (non-colluder) peers as a decoy. The performance of the resulting scheme is evaluated by simulation. The result of simulations indicates that the proposed schemes detect all colluders about 30% earlier on average than the original scheme while keeping the accuracy of the colluder detection at medium collusion rate.

Keywords: peer-to-peer, content delivery network, reputation system, colluder

1. Introduction

Content delivery is a crucial operation used in many network applications such as Akamai [4], CoDeen [13], CoralCDN [2], and Globule [11]. Although many online resources currently being shared over a network are free or open access (e.g., YouTube and BitTorrent), there is a strong demand to distribute a given media file only to paid users in a secure and cost-effective manner. Recently, Peer-to-Peer Content Delivery Networks (P2P CDNs) have attracted considerable attention as a cost-effective way to deliver large media files to many users in a given computer network [12], [15], [16]. Content delivery in a P2P CDN is invoked by the owner of a media file by pushing a copy of the file to the P2P overlay, and once a copy is available in the overlay, it will be delivered to the authorized recipients by repeating local communication among nearby peers in the P2P overlay, where the authorized peer is a peer who paid for the content and received an authority from the distribution agent (DA) of the P2P CDN.

If all peers in the P2P network are "honest," it is not difficult to allow authorized peers to successfully receive requested files, and to reject any download request received from unauthorized peers. Unfortunately however, in actual P2P networks, there may exist several "dishonest" peers called **colluders** who illegally leak shared contents to unauthorized peers called **pirates**. Since the existence of pirates will significantly lose the benefit of the owner of the paid contents, it is strongly required to identify all pirates in the network, and to ask authorized peers not to leak the contents to the pirates. In addition, since the content delivery in P2P CDNs is conducted in a peer-to-peer manner, in order to stop the leak of the contents, all colluders should also be detected, and the authority of those colluders should be removed, if necessary.

To resolve such a critical issue in P2P CDNs, Lou and Hwang proposed a colluder detection scheme based on the notions of reputation and proactive content poisoning [9]. The first idea of the scheme is to "poison" pieces of a content if the receiver of a local communication is recognized as a pirate, so that the download time of the pirate will significantly increase. The second idea of the scheme is to adopt a technique of online reputation to identify a set of colluders, where during a colluder detection process, it carefully eliminates wrong reputations issued by dishonest peers. Details of the scheme will be outlined in Sections 2.2 and 2.3.

In previous work [1] we introduced three schemes that extend the Lou and Hwang's colluder detection scheme (abbreviated as LH-scheme, in what follows). Although the schemes were shown to be sufficiently quick to identify the set of colluders, there is a lack of performance analysis and discussion about their robustness and viability.

In this paper, we showed that our proposed schemes are robust enough to be implemented in the real world. Our schemes make colluder detection quicker than conventional randomize method, without sacrificing the accuracy of colluder detections. The reader should note that the speed of colluder detections is a critical factor for the owner of paid contents, since if we could not identify a colluder before completing an illegal download by a pirate, we cannot stop the spread of an illegal copy of the file initiated by the pirate (recall that we cannot remove the authority of the pirate, since the pirate is not authorized). The performance of the proposed schemes is evaluated by simulation. The result of simulations indicates that the proposed schemes can detect colluders more quickly than the original LH-scheme; for example, when the percentage of colluders is 30% of the authorized peers, it detects such colluders 1.3 times faster on average than the LH-

[1] Hiroshima University, Hogashihiroshima, Hiroshima 739–8527, Japan

[a) ervianto@se.hiroshima-u.ac.jp

scheme, and it reduces the minimum file size that can be securely delivered to the paid peers to one fourth.

The remainder of this paper is organized as follows. Section 2 reviews related works and existing methods including the original LH-scheme. Section 3 describes our proposed method. The result of simulations is given in Section 4. Finally, Section 5 concludes the paper with future work.

2. Related Works

2.1 Overview

In open distributed systems such as P2P CDNs, the reputation of peers plays an important role in deciding an appropriate action of each peer. In fact, to avoid the risk of suspicious transactions, each peer prefers to interact with good-reputation peers rather than with bad-reputation peers. To date, a number of P2P reputation systems have been proposed in the literature, which include EigenTrust [7], PeerTrust [14], and PowerTrust [17]. They are designed to increase the reliability of the underlying P2Ps, and to protect each peer from being interacted with malicious peers [6], [10]. The reputation of a peer is generally represented by a reputation score, which is calculated by aggregating feedback from other peers through appropriate rating and/or voting mechanisms. By disclosing such scores to all peers in the network, each peer can make an appropriate decision concerning the trustfulness of other peers.

As an alternative approach, Lou and Hwang recently proposed a scheme to protect copyrighted content from being leaked to unauthorized peers [9]. This scheme is based on an aggregation of reported collusion events similar to existing reputation schemes, while it is different from those schemes in the sense that it protects the right of content owners (i.e., copyright holders) rather than the security of client peers. Details of the Lou and Hwang's scheme will be described in succeeding subsections.

2.2 Lou and Hwang's P2P CDNs

Lou and Hwang proposed a model of P2P CDN consisting of honest peers, colluders, and pirates [9]. See **Fig. 1** for illustration. In this model, similar to many P2P CDNs, each file of shared content is divided into small chunks, and each chunk is further divided into sub-chunks of smaller size. Integrity of downloaded files is verified after completing the download of some part of the file using an appropriate hashing protocol[*1], and if the integrity check fails, the peer should discard the portion and re-download the entire sub-chunks concerned with the discarded portion.

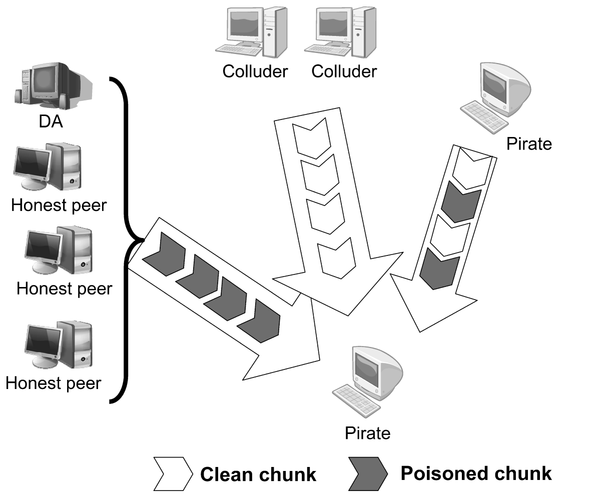


Fig. 1 Chunks poisoning mechanism of Lou and Hwang P2P CDN. Legitimate clients receive only clean chunks, but pirates receive a mix of clean and poisoned chunks. Illustrated based on Ref. [9].

In the Lou and Hwang's poisoning scheme, pirates who attempted an illegal download receive poisoned chunks from DAs or honest peers so that the integrity check fails, although they still have a chance to receive clean chunks from their colluders. Thus, since each pirate should fail at the hashing protocol several times, it will have to repeat discard-and-redownload of poisoned chunks, which significantly increases the download time of the entire file compared with legitimate downloads. In addition, Lou and Hwang pointed out that such a poisoning scheme correctly works if we could assume an appropriate Peer Authorization Protocol (PAP), described as follows:

- Peer endpoint address is forgery proof.
- Authorization tokens cannot be shared by peers.
- Pirates cannot poison legitimate clients.
- Stolen private keys are useless to pirates.
- Peers are able to recognize an unauthorized request.

More concretely, the protocol uses an encrypted authorized token to manage peers' authorization, and provides mechanisms for the secured peer joining process, forgery-proof identity, secured content request, and secured file index covered by the PAP.

2.3 The LH-scheme

The performance of the poisoning scheme could be enhanced by combining it with an appropriate colluder detection scheme. Lou and Hwang proposed a randomized scheme (LH-scheme) for such a colluder detection [9]. **Figure 2** illustrates the basic flow of the LH-scheme. In this scheme, DAs randomly recruit decoys from paid clients to probe other clients by conducting Algorithm 1. The function *SendIllegalRequest()* sends an illegal request to a designated target peer intentionally, and provokes (hidden) colluders to illegally provide the decoy with clean content.

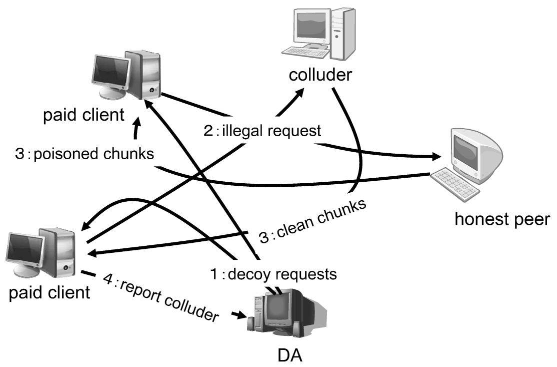


Fig. 2 The mechanism of LH-scheme. Distribution agent randomly recruits probe from paid clients. Collusion is reported when a peer replies to an illegal request with clean chunks. Illustrated based on Ref. [9].

[*1] For example, the BitTorrent family divides each file into 256 KB chunks, and the file integrity is checked at this level [3]. The Gnutella family divides files into 64 KB chunks, but hashing protocol is applied to the entire file [5]. The eMule family divides files into 9,500 KB chunks, and hashing protocol is applied at this level [8].

Algorithm 1 Probing process

```
Output: Target peer collusion status
        True: Target peer is colluder
        False: Target peer is not colluder
SendIllegalRequest();
        // send illegal request to designated target peer
if {clean chunk is returned}
    return true;
else
    return false;
endif
```

If the target peer returns a clean content, then the decoy sends a report to the DA (Step 3), while the report may not be correct if the decoy is not honest (e.g., it is a colluder of pirates). To make the scheme to be resistant to such a malicious report, Low and Hwang adopted a majority voting in their scheme. More concretely, peer j is associated with two values c_j and t_j, where c_j is called the **collusion score** and t_j is called the **trust score**, and a decision is made by comparing c_j with a collusion threshold φ which is given by each content owner in advance; i.e., when the collusion score of a peer reaches φ, then the content owner recognizes it as a colluder. The update of those scores is conducted as follows. Let r_{ij} denote the reported value received from decoy i, which takes 1 if i recognizes j as a colluder and takes 0 otherwise. After receiving r_{ij} from peer i, DA updates collusion score of peer j as follows:

$$c_j := \min\{c_j + t_i \times r_{ij}, \varphi\}, \tag{1}$$

where

$$t_i := 1 - c_i/\varphi. \tag{2}$$

Note that each report is weighed with reporter's trust score, thus a report received from an untrusted decoy is ignored, and more credibility is given to peers with higher trust scores to ensure the accuracy of the colluder detection.

3. Proposed Scheme

In this section, we propose three schemes to identify a set of colluders more quickly than the (original) LH-scheme while keeping the accuracy of colluder detection. To this end, we will improve the LH-scheme in two different directions.

Recall that the LH-scheme is based on the probing of (suspected) peers by a set of (recruited) decoys. Our first idea is to refine the selection of the peers being probed (i.e., the target peers), which will be referred to as an object-based approach. The second idea is to refine the selection of probing peers (i.e., decoys), which will be referred to as a subject-based approach. The details of those approaches are described in the succeeding subsections.

Before describing the details of the proposed scheme, we clarify the model of peers assumed in the proposed schemes.

- All peers know the identity of DAs. Thus, DAs never act as a decoy.
- While serving as a decoy, each colluder always tells a lie, i.e., it reports the colluder as a non-colluder and vice versa.
- Each colluder replies with a clean content to an illegal request in a probabilistic manner. This assumption is crucial for colluders not to be detected as a colluder very easily.

Algorithm 2 Object-based approach

```
Input: $N^*$ // set of non-suspected peers
       $S^*$ // set of suspected peers
Output: $C$ // set of detected colluders
RandomlySelectDecoys();
       // select decoys randomly from candidate peers
for each decoy
    SetTarget();
       // set target peers
endfor
ProbingProcess();
       // probing process by each decoy
UpdateParameters();
       // update collusion score of each peer based on decoy's reports
UpdateSetMembers();
       // update members of each set based on peer's collusion score
```

3.1 Object-based Approach

In the LH-scheme, each of the paid clients is randomly probed by the decoys; i.e., if there are N candidates to be examined, for each colluder i, the expected number of probes which should be attempted before firstly hitting peer i is given by N (we may consider a Bernoulli trial with success rate $1/N$). Thus, by the linearity of expectation, the expected number of probes which should be used to identify i as a colluder is given by φN, where φ is the threshold used in the LH-scheme. The basic idea of our first improvement is to introduce an *intensive probing mode* in which decoys intensively examine specific (i.e., suspected) peers, and to use this mode with the conventional random probing in a combined manner.

The behavior of the scheme is controlled by using an appropriate threshold γ ($< \varphi$). In the following, we say that a peer is *suspected* if its collusion score is equal to or higher than γ, and use symbol S^* to denote a set of suspected peers. Any other peers (i.e., peers with a collusion score lower than γ) belong to a set of non-suspected peers, which is denoted by N^*. Sets S^* and N^* are maintained by the DAs, and are updated to reflect the latest collusion scores of the peers. Any peers with collusion scores reaching φ is eliminated from candidates, and belong to the colluders set denoted by C.

This scheme detects colluders by repeating Algorithm 2. The function *RandomlySelectDecoys()* selects a number of decoys randomly from paid clients. Then, for each decoy, DAs will specify a target peer to be examined. The function *SetTarget()* selects peers to be examined according to the following probability distribution:

> The probability of selecting a suspected peer is w times larger than the probability of selecting a non-suspected peer, where w is an appropriate weight greater than one.

Thus, if N^* is a set of non-suspected peers, the probability P_n of selecting a non-suspected peer i from the candidates is given by

$$P_n = \frac{1}{w \times |S^*| + |N^*|}$$

and the probability P_s of selecting a suspected peer j is given by

$$P_s = \frac{w}{w \times |S^*| + |N^*|}.$$

As the decoy request came from DAs, each decoy conducts *ProbingProcess()* as shown in Algorithm 1, and reports the probing result to the appropriate DA. After receiving probing results from decoys, the function *UpdateParameters()* updates each peer's collusion score c_i and trust score t_i. Then based on the peer's collusion score c_i, the function *UpdateSetMembers()* updates the members of each set; a peer with a collusion score lower than γ remains as a member of the set of non-suspected peers N^*, while a peer with a collusion score equal or higher than γ but lower than φ is added to the set of suspected peers S^*. A peer is removed from the set of candidates and added to the set of colluders C if it is identified as a colluder by collecting a sufficiently high collusion score reaching φ (the update of the collusion scores reflects the trust score of the evaluator similar to the original LH-scheme).

Figure 3 illustrates the basic flow of the resulting colluder detection scheme. In Fig. 3, peer A is a real colluder, so its collusion score becomes higher than the suspected threshold γ relatively quicker than the others, with the result that it became a suspected peer in a short period. Once peer A became a suspected peer, many peers probed it for collusion (Fig. 3 (b)). Then, if peer A continues to act as a colluder, this scheme quickly detects and removes it from the network.

The impact of two parameters γ and w to the performance of the scheme will be evaluated in the next section. Intuitively speaking, the speed of colluder detection will increase as decreasing γ or as increasing w, while it would degrade the accuracy since it becomes less tolerant to malicious reports received from colluders.

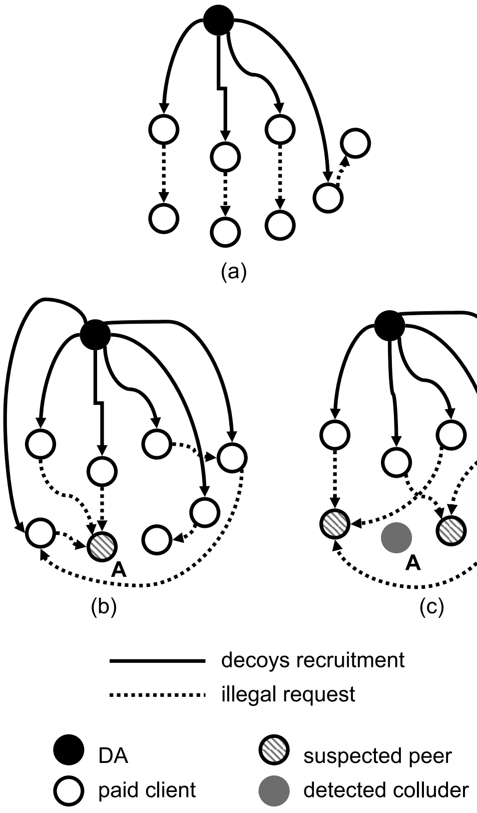


Fig. 3 Colluder detection with object-based approach. Firstly distribution agents recruit decoys randomly (a). As time elapsed, the collusion score of peer A became higher than threshold γ, thus most of the probing actions are applied to peer A (b). Continuing acting as a colluder makes the collusion score of peer A quickly reach collusion threshold, thus it will be recognized as a colluder quickly (c).

3.2 Subject-based Approach

The key idea of our second improvement is to refine the selection of decoys which was randomly done in the original LH-scheme. As described before, a colluder would send a wrong report to the DAs while serving as a decoy. Such a malicious report prolongs the colluder detection time since it restrains an increase of the collusion score of actual colluders, and in addition, it degrades the accuracy since it could increase the collusion score of non-colluder peers.

The simplest way to realize such a refinement is to use the trust score of each peer in selecting the peer as a decoy. More concretely, we may associate the following probability P_i to each peer i as the probability of being selected as a decoy:

$$P_i = \frac{t_i}{\sum_j t_j}.$$

In the following, we refer to such a simple extension as SIMPLE.

A further improvement could be attained by explicitly considering another layer for the reputation management in addition to the conventional colluder detection layer. The resulting scheme is referred to as REP. See **Fig. 4** for illustration. In the reputation management layer of the REP, each peer j is associated with a reputation score rep_j, which is updated by conducting a modified LH-scheme. Reputation score rep_j of each peer j in this layer is calculated by the following equation, which is an extension of the trust score shown in Eq. (2):

$$rep_j := 1 - cr_j/\varphi \qquad (3)$$

where cr_j is the collusion score of peer j in the reputation management layer, which is calculated as:

$$cr_j := \min\{cr_j + rep_i \times q_{ij}, \varphi\}, \qquad (4)$$

where q_{ij} is reported value received from decoy i, which takes 1 if i recognizes j as a colluder and takes 0 otherwise. Variable q_{ij} reflects how peer i rates peer j behavior.

In the colluder detection layer, a strict decoy recruitment policy is applied based on the outcome of the reputation management

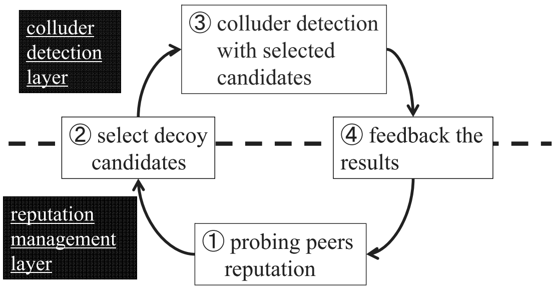

colluder detection layer
③ colluder detection with selected candidates
② select decoy candidates
④ feedback the results
reputation management layer
① probing peers reputation

Fig. 4 Colluder detection with subject-based approach. Colluder detection itself will be conducted in the upper layer. Peers will be selected as candidates of decoys based on their reputation in the bottom layer, and their bottom layer reputation will be affected by the upper layer colluder detection result.

layer. A peer i can be selected as a decoy in this layer if and only if $rep_i > \beta$ for some real β, where β is a reputation threshold set by the DAs. Selected decoys probe other peers similar to the original LH-scheme, and the DAs update collusion scores of the peers according to Eq. (1). Once peer j is detected as a colluder in the upper layer, this scheme updates any peer i's reputation score in the reputation management layer in the following manner:

$$cr_i := cr_i + (0.5 - \overline{q}_{ij}), \tag{5}$$

where $\overline{q}_{ij}$ is the average of peer i's rating about peer j's attitude. If peer i has a tendency to rate peer j as 'colluder,' $\overline{q}_{ij}$ should close to 1. If later peer j is detected as a colluder in the upper layer, peer i will be granted for its honest report with lowered cr_i value, which raises its reputation score rep_i. Similarly, if peer i tends to rate peer j as 'non-colluder,' REP lowers peer i's reputation score for its dishonest report. With such a mechanism, peers will be encouraged to issue an honest report in the probing mechanism, while peers with a false report will be penalized by a lower reputation score.

4. Simulation

In order to evaluate the performance of the proposed schemes, we simulate P2P CDN with our schemes applied. As a metric for evaluation, we focus on: a) the speed of colluder detection which reflects the scheme's impact on avoiding an illegal leak of the contents; b) the accuracy of colluder detection which assures the reliability of schemes from the honest client's view point; and c) the overhead which draws the viability of schemes to be implemented in the real world.

The speed of colluder detection is measured by the time transition of the colluder detection rate indicating the percentage of colluders identified by the scheme; e.g., if there are 100 colluders in the network and the scheme detects 7 colluders, the colluder detection rate is $7/100 = 7\%$. Undetected colluders keep leaking the contents to the pirates until they are detected, so minimizing detection time becomes an important point to keep contents leak minimal.

On the other hand, the accuracy of the schemes is measured by the wrong detection rate, which is defined by the percentage of non-colluders (i.e., honest clients) being identified as colluders among all non-colluders; e.g., if there are 100 honest clients and if a scheme identifies 2 honest clients as colluders, then the wrong detection rate of the scheme is $2/100 = 2\%$. Because honest clients are paying for the contents, the accuracy should be retained as high as possible, thus a good scheme should keep the wrong detection rate somewhere near zero.

Overhead is also important point when discussing a scheme's viability. In this experiment, we mainly focused on the bandwidth used by an honest client to cooperate with the scheme, i.e., the overhead for sending back unauthorized requests from the decoy with poisoned chunks. Note that we do not take account of the bandwidth consumed for responding unauthorized requests from pirates, as it already discussed by Lou and Hwang [9].

The performance of three proposed schemes, Object-based, SIMPLE, and REP is evaluated by conducting a comparison with the original LH-scheme, in terms of the above three metrics.

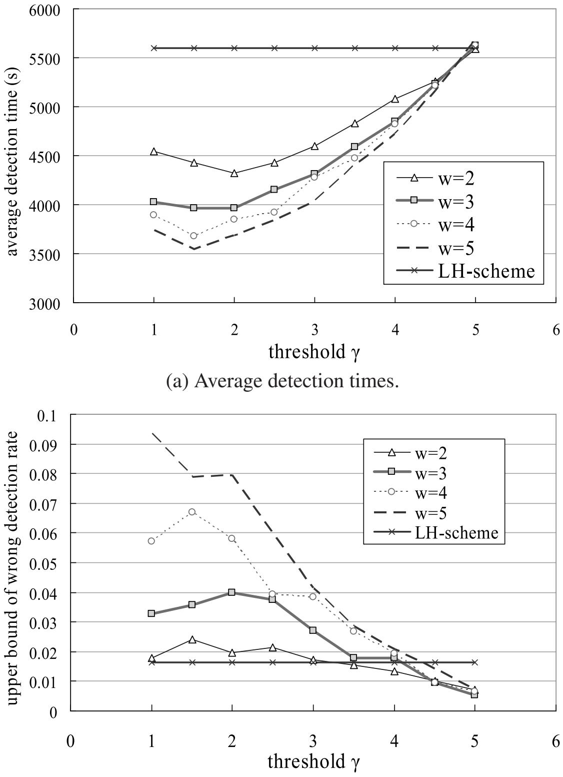
4.1 Setup

In the simulation, we consider a P2P network consisting of 1,000 pirates and 1,000 authorized peers, i.e., we fix the piracy rate to 50%. The collusion rate ϵ, which is the percentage of colluders among authorized peers, is either 15%, 30%, or 45%.

We set the probability of colluders replying to unauthorized requests with clean contents to 90%, which is quite optimistic because in the real world colluders may act more carefully in order not to be detected easily. We assume DAs perform a collusion check every 30 seconds, recruiting 50 decoys at each check point. For the REP scheme, we set the reputation threshold β to 0.7. For simplicity, we assume that there is no network delay between the peers.

We set the collusion threshold $\varphi = 5$ in all simulations. Note that the value of φ is a trade-off between detection speed and accuracy. We saw $\varphi = 5$ has a good trade-off performance compared to other value. At the beginning, the collusion score c_i of each peer i is 0 and hence the trust score t_i is 1.

To select appropriate parameters for the Object-based approach, we examined the impact of w and γ to the performance of the scheme. In this preliminary experiment, we take account of two metrics: (a) average detection time, defined as the average of the time required to detect each colluder, which roughly reflects the detection speed of the scheme; and (b) the upper bound of wrong detection rate, defined as the final rate of false positive (i.e., non-colluders wrongly detected as colluders), which roughly reflects the accuracy of the scheme. **Figure 5** summarizes the results. The figure indicates that the average detection time decreases as decreasing γ or increasing w (see Fig. 5 (a)), while the wrong detection rate increases when γ is low or w is



(a) Average detection times.

(b) Upper bound of wrong detection rate.

Fig. 5 The relationship between w and γ in the Object-based approach.

high (see Fig. 5 (a)). Such a result is explained as follows.

Lower γ means shorter time for a peer to be identified as a suspected peer, which also means a shorter time to identify the peer as a colluder. For this reason a lower γ gives a shorter average detection time. On the other side, a non-colluder peer may accidentally be identified as a suspected peer, and such a peer has a higher probability of being identified as a colluder. Thus, lower γ makes the upper bound of the wrong detection rate become higher, which means a lower accuracy on the colluder detection process. On the other hand, a higher w gives a better average detection time, as suspected peers is given higher probability to be selected as a target of the probing mechanism. However, it results in a higher wrong detection rate, for the same reason as the lower γ.

As such, the selection of parameters w and γ is a trade-off between the colluder detection time and the accuracy. By the above observations, in the following simulations, we set those parameters to $w = 3$, $\gamma = 3$, because we saw that such values give an arguably quick detection speed while keeping the wrong detection rate low.

4.2 Detection Speed

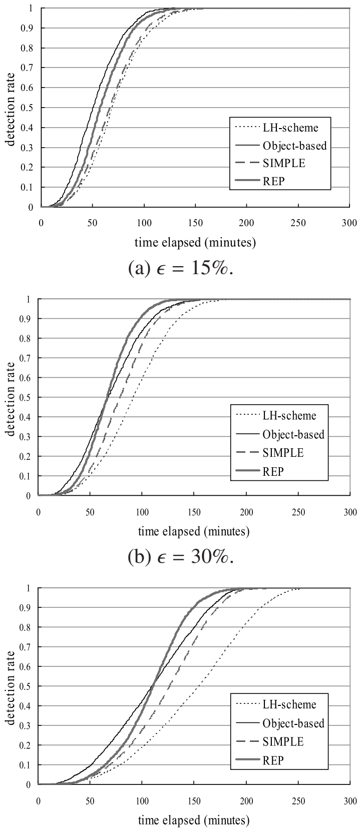
Figure 6 illustrates the time transition of the colluder detection rate, where the horizontal axis is the elapsed time starting at the time point of an invocation of the colluder detection process. Note that since each of the hidden colluders probabilistically returns a clean chunk to the pirates, the area of a *region above an S-shaped curve* in the figure is proportional to the amount of clean chunks which are illegally leaked by the colluders under the corresponding colluder detection scheme.

Figures 6 (a), (b), and (c) illustrate the results for $\epsilon = 15\%$, 30%, and 45% respectively. In all three cases, our three schemes outperform the original LH-scheme. In particular, Object-based scheme performs better for $\epsilon = 15\%$,

REP exhibits the best performance for $\epsilon = 30\%$, and Object-based and REP almost similarly become the best for $\epsilon = 45\%$. In fact, when $\epsilon = 30\%$, the area above REP becomes 73% of the area above the original LH-scheme, which indicates that it prevents contents leaking 27% more effectively compared to the original LH-scheme. Such a good performance of the REP is apparently due to an accurate recruitment of non-colluders as a decoy, while such an effect becomes less significant when many of authorized peers are colluders. Another key observation is that the SIMPLE does not beat the Object-based in either case. Thus, if we want to speed up the colluder detection process by using a Subject-based approach, we should carefully design the underlying reputation management layer as in the REP.

4.3 Accuracy

Next, we evaluate the accuracy of colluder detection. **Figure 7** illustrates the time transition of the wrong detection rate. As shown in the figure, although the wrong detection rate of the Object-based scheme is much higher than the other schemes including the original LH-scheme, two subject-based schemes certainly improve the accuracy of the LH-scheme. In particular, the wrong detection rate of the REP is bounded by 10% of the

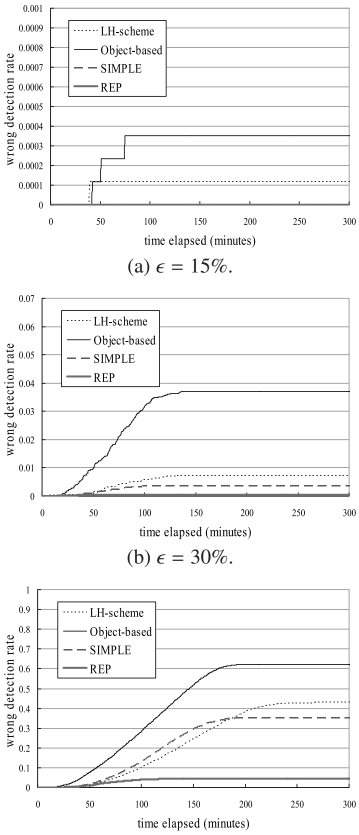


(a) $\epsilon = 15\%$.

(b) $\epsilon = 30\%$.

(c) $\epsilon = 45\%$.

Fig. 6 Time transition of the colluder detection rate.



(a) $\epsilon = 15\%$.

(b) $\epsilon = 30\%$.

(c) $\epsilon = 45\%$.

Fig. 7 Time transition of the wrong detection rate.

Table 1 Average probability of selecting a good decoy and a good target.

Scheme	Probability of selecting good decoy	Probability of selecting good target
LH-scheme	0.849	0.152
Object-based	0.862	0.170
SIMPLE	0.892	0.150
REP	0.911	0.157

LH-scheme. Moreover, while the original LH-scheme and our Object-based scheme wrongly detected some honest peers as colluders, our Subject-based methods successfully eliminated such wrong detection in lower collusion rate as shown in Fig. 7 (a).

If we look closely at Fig. 7 (b), however, it shows that none of our methods are accurate enough to eliminate wrong detection in medium collusion rate. It shows that SIMPLE performs better than the original LH-scheme, but it detects 2.6 honest peers as colluders on average. Even REP, which could be considered as the most accurate method, wrongly detects honest peer as colluder 23 times in 100 simulations, or about 0.23 peer on average. While such a wrong detection rate could be considered low enough, because honest users pay for the contents and act legally, recognizing those peers as colluders could means losing customers' trust in the system, which should be avoided wherever possible.

Honest peers may be recognized as colluders if they are unluckily being probed by the "real colluders" and being reported by them with false reports. Such a situation happens in the proposed schemes since it selects decoy and the probing target in a probabilistic manner, e.g., Object-based scheme randomly selects decoys from non-suspected peers and it randomly selects probing targets from suspected peers. **Table 1** summarized the probability of selecting a good decoy and a good target in each scheme. In both schemes, honest peers are good decoys because they do not provide false reports, and undetected colluders are good probing targets because our main objective is to detect such peers.

As shown in the table, a higher accuracy achieved by REP is apparently due to the higher probability of selecting a good decoy. To increase such probability, REP carefully selects a decoy based on the peer's reputation on the additional reputation management layer. In addition, REP also keeps the reliability of the reputation values by using the feedback from the colluder detection layer. Thus if we could further increase the probability, we could realize a perfect accuracy in the colluder detection process, which is left for our important future work.

Finally, the reader should note that the probability of selecting a colluder as a probing target is not directly related to the accuracy; e.g., although the Object-based scheme has the highest probability, the accuracy of the scheme is not very high, as shown in Fig. 7 (b). It is related to the speed of colluder detection as was examined in the last subsection.

4.4 Overhead

In our experiments, DAs recruit 50 decoys every 30 seconds. If we assume the overhead for each decoy recruitment process is 10 KB, the overhead for such communications is sufficiently negligible for peers with broadband connection (1 Mbps). Another concern is the amount of the bandwidth used by an honest client to respond to any unauthorized request from decoys with poisoned chunks. In eMule networks, each file is divided into 9,500 KB sized parts, which is further divided into 180 KB sized chunks. We assume that a poisoned chunk sent by an honest client to respond to any unauthorized request is 180 KB each. **Figure 8** plots the bandwidth used by honest clients to respond to any unauthorized requests from decoys in 30% collusion rate.

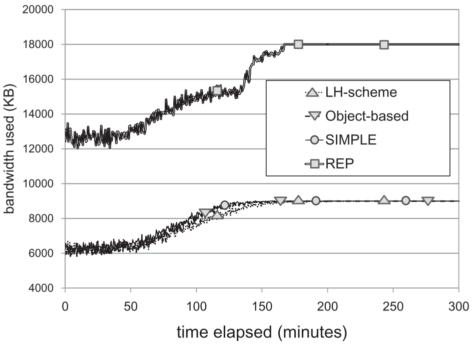


Fig. 8 Bandwidth used by honest clients for sending polluted chunks to decoys.

During the colluder detection process, some of the unauthorized requests from decoys are directed to the colluders, which explains lower overhead at the earlier session. After all colluders are detected, however, all probing targets are set only to honest clients, so the overhead became stable in the higher value at the later period. Moreover, there is no significant difference of overhead between LH-scheme, Object-based, and SIMPLE. However, REP costs overhead double in comparison with other schemes due to the two layer structure of the scheme, where each layer costs approximately the same overhead as the LH-scheme. While this extra overhead could be considered as a reasonable cost to high detection speed and accuracy, it remains our future work to reduce such overhead.

5. Concluding Remarks

In this paper, we proposed three schemes to detect colluders in the P2P CDNs: Object-based, SIMPLE, and REP. Simulation results show that all of the three schemes outperformed the original LH-scheme in terms of colluder detection speed. In particular, REP speeding up detection process by about 1.35 times when the percentage of colluders is 30%. As for the accuracy of the schemes, REP exhibits the best performance among others, and it could bound the rate of wrong colluder detection by 10% of the original LH-scheme. The above results indicate that the third scheme REP is particularly effective to improve the speed of colluder detection without increasing the rate of wrong detections.

Although REP performs better than the other schemes, it costs in overhead twice as much as other schemes to achieve such good performance. Our future work is to reduce the overhead resulting in REP scheme without sacrificing the detection speed and accuracy. Combining the proposed schemes with other (more sophisticated) Peer-to-Peer reputation systems such as EigenTrust [7], PeerTrust [14], and PowerTrust [17] is also an interesting direction for further research. Another future work is to refine the probability distribution used in the Object-based scheme in such a way as to tolerate wrong reports issued by colluder decoys.

Reference

[1] Abdullah, E. and Fujita, S.: A Quick Detection of Colluders in P2P CDNs to Avoid An Illegal Leak of the Contents, *ICNC 2010 1st International Conference on Networking and Computing*, pp.20–27 (2010).

[2] The Coral Content Distribution Network, available from ⟨http://www.coralcdn.org/⟩.

[3] Cohen, B.: The BitTorrent Protocol Specification (Jan. 2008), available from ⟨http://www.bittorrent.org/⟩.

[4] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R. and Weihl, B.: Globally Distributed Content Delivery, *IEEE Internet Computing*, Vol.6, pp.50–58 (2002).

[5] Gnutella Developers, Gnutella Protocol Specification (July 2008), available from ⟨http://wiki.limewire.org/⟩.

[6] Jøsang, A., Ismail, R. and Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision, *Decision Support Systems*, Vol.43, No.2, pp.618–644 (2007).

[7] Kamvar, S.D., Schlosser, M.T. and Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in P2P Networks, *WWW '03: Proc. 12th International Conference on World Wide Web*, pp.640–651, ACM, New York, NY, USA (2003).

[8] Kulbak, Y. and Bickson, D.: The eMule Protocol Specification, *Tech. Rep.*, The Hebrew University of Jerusalem, Jerusalem (Jan. 2005).

[9] Lou, X. and Hwang, K.: Collusive Piracy Prevention in P2P Content Delivery Networks, *IEEE Trans. Computers*, Vol.58, pp.970–983 (2009).

[10] Ooi, B.C., Liau, C.Y. and Tau, K.L.: Managing Trust in Peer-to-peer Systems Using Reputation-based Techniques, *Proc. Web-Age Information Management*, pp.2–12 (2003).

[11] Pierre, G. and Steen, M.: Globule: A Collaborative Content Delivery Network, *IEEE Communications Magazine*, Vol.44, pp.127–133 (2006).

[12] Shangqin, X., Zhengding, L., Hefei, L. and Fuhao, Z.: A Trust Scheme Based DRM Model for P2P System, *Wuhan University Journal of Natural Sciences*, Vol.11, No.5, pp.1373–1377 (Sep. 2006).

[13] Wang, L., Park, K.S. Pang, R., Pai, V. and Peterson, L.: Reliability and Security in the CoDeen Content Distribution Network, *ATEC '04: Proc. Annual Conference on USENIX Annual Technical Conference*, p.14, USENIX Association, Berkeley, CA, USA (2004).

[14] Xiong, L. and Liu, L.: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities, *IEEE Trans. Knowledge and Data Engineering*, Vol.16, pp.843–857 (2004).

[15] Zhang, X., Liu, D., Chen, S., Zhang, Z. and Sandhu, R.: Toward Digital Rights Protection in BitTorrent-like P2P Systems, *Society of Photo-Optical Instrumentation Engineers* (*SPIE*) *Conference Series*, Vol.6818 (Jan. 2008).

[16] Zhang, Y., Yuan, C. and Zhong, Y.: Implementing DRM over Peer-to-Peer Networks with Broadcast Encryption, *Advances in Multimedia Information Processing EPCM 2007*, Lecture Notes in Computer Science, Vol.4810, pp.236–245, Springer Berlin, Heidelberg (2007).

[17] Zhou, R. and Hwang, K.: PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing, *IEEE Trans. Parallel and Distributed Systems*, Vol.18, pp.460–473 (2007).

Ervianto Abdullah received his B.E. degree in information engineering from Hiroshima University in 2010. He is currently a master student at the Graduate School of Engineering, Hiroshima University. His research interests include parallel and distributed computer systems, especially peer-to-peer contents distribution networks and peer-to-peer reputation systems.

Satoshi Fujita received his B.E. degree in electrical engineering, M.E. degree in systems engineering, and Dr.E. degree in information engineering from Hiroshima University in 1985, 1987, and 1990, respectively. He is a Professor at the Faculty of Engineering, Hiroshima University. His research interests include communication algorithms on interconnection networks, parallel algorithms, graph algorithms, and parallel and distributed computer systems. He is a member of IPSJ, SIAM Japan, IEEE, and SIAM.