JR Positive Comfortable Destroy to Prevent Collusive Piracy in P2P File Sharing

[1]B. Venkateswarulu, [2]N. Sandhya Rani
[1,2]Dept. of CSE, Avanthi Institute of Engg. & Tech., Narsipatnam, Visakhapatnam, AP, India

Abstract

Today's peer-to-peer (P2P) networks are grossly abused by Illegal distributions of music, games, video streams, and popular software. These abuses have resulted in heavy financial loss in media and avoidance industry. Collusive piracy is the main source of intellectual property violations within the boundary of P2P networks. This problem is resulted from paid clients (colluders) illegally sharing repressive avoidance files with unpaid clients (pirates). Such an on-line piracy has hindered the use of open P2P networks for commercial avoidance delivery. We propose a upbeat avoidanceing scheme to stop colluders and pirates from working together in alleged repression infringements in P2P file sharing. The basic idea is to detect pirates with identity based signatures and time-stamped tokens. Then we stop collusive piracy without hurting legitimate P2P clients.

We developed a new peer authorization protocol (PAP) to distinguish pirates from legitimate clients. Detected pirates will receive avoidances chunks in repeated attempts. A reputation-based mechanism is developed to detect colluders. The system does not slow down legal download from paid clients. The pirates are severely penalized with no chance to download successfully in finite time. Based on simulation results, we find 99.9% success rate in preventing piracy on file-level hashing networks like Gnutella, KaZaA, Area, LimeWire, etc. Our protection scheme achieved 85-98% avoidance rate on part-level hashing networks like eMuel, Shareaz, eDonkey, Morpheus, etc. Our new scheme enables P2P technology for building a new generation of avoidance delivery networks (CDNs). These P2P-based CDNs provide faster delivery speed, higher avoidance availability, and cost-effectiveness than using conventional CDNs built with huge network of surrogate servers.

Keywords

Peer-to-Peer Networks, Avoidance Avoidanceing, Avoidance Delivery Networks, and Network Security.

I. Introduction

We proposed Our goal is to stop collusive piracy within the boundary of a P2P avoidance delivery network. Our goal is to stop collusive piracy within the boundary of a P2P avoidance delivery network. Our protection scheme works nicely in a P2P network environment. The scheme cannot stop randomized piracy in open Internet using Email attachment or any other means to spread repressive avoidances, illegally. Randomized piracy is beyond the scope of this study. Traditional avoidance delivery networks (CDN) use a large number of surrogate avoidance servers over many globally scattered WANs. The avoidance distributors need to replicate or cache avoidances on many servers. The bandwidth demand and resources needed to maintain these CDNs are very expensive.

A P2P avoidance network significantly reduces the distribution cost [27], since many avoidance servers are eliminated and open networks are used. P2P networks improve the avoidance availability, as any peer can serve as a avoidance provider. P2P networks are desired to be scalable, because more peers or providers lead to faster avoidance delivery.

Peer-to-peer (P2P) file-sharing networks are most cost effective in delivering large files to massive number of unfortunately; on-line piracy has hindered the legal and commercial use of P2P technology. The main sources of illegal file sharing are peers who ignore repression laws and collude with pirates. To solve this peer collusion problem, we propose a upbeat repression-compliant system for protecting legalized P2P avoidance delivery.

We use identity-based signatures (IBS) to secure file indices. IBS offers the same level of security as PKI-based signatures with much less overhead. We apply discriminatory avoidance avoidanceing against pirates. We focus on protection of decentralized P2P avoidance networks. Protecting centralized P2P networks like Napster or mp3.com is much simpler than the scheme .Our scheme cannot stop primary piracy in open Internet using Email attachment or any other means to spread repressive avoidances, illegally. In particular, our scheme appeals to protect perishable or large-scale avoidances that diminish in value as time elapses. Our system stops abuses within the P2P network, exclusively. Honest or legitimate clients are those that comply with the repression law not to share avoidances freely. Pirates are peers attempting to download some avoidance file without paying or authorization. The colluders are those paid clients who share the avoidances with pirates. Pirates and colluders coexist with the law-abiding clients. Avoidance avoidanceing is realized by deliberate falsification of the file requested by pirate. The media industry backed by RIAA (Record Industry Association of America) and MPAA (Motion Picture Association of America) has applied unscreened brutal-force avoidance avoidanceing to deter piracy in open P2P file-sharing networks.

We developed a reputation-based method to detect peer collusion in piracy process A repression sheltered P2P network should benefit both media industry and Internet user communities .Our work leads to the development of a new generation of CDNs based on P2P technology. Table 1 lists important symbols and notations used to benefit our readers. These terms are used to secure file indices; generate access tokens; quantify avoidanceing effects, collusion avoidance, and to define the performance metrics.

We focus on finding solution of collusive piracy within the scope of a P2P network. Inter-network piracy between unsheltered P2P networks is a much more complex security problem. That compound problem is not within this study. Our main purpose is to stop colluders from releasing avoidance files freely and to abort pirate effort from accumulating clean chunks. There are many other forms of on-line or off-line piracy that are beyond the scope of this study

II. Linked Works

We review avoidance work on repressive P2P avoidance delivery. Then we identify our exclusive approach to solving the problem in P2P networks.

A P2P network does not require many expensive servers to deliver avoidances. Instead, avoidances are scattered and shared among the peers. P2P networks improve from conventional CDNs in avoidance availability and system Scalability. Many performance and security issues in P2P networks have been studied. Electronic publishing was hindered by the rapid growth of repression violations [14, 26]. The major source of illegal P2P avoidance distribution lies in peer collusion to share repressive avoidance with other peers or pirates. Proposed a repressive music distribution over a P2P network. However, the system is ineffective when colluders are undetected. Digital watermarking is injected to avoidance file so that when a pirated copy is discovered, authorities can find the origin of piracy via a exclusive watermark in each copy. In a P2P network, all peers are sharing exactly the same file (if not avoidances), which effectively defeats the purpose of watermarking. Thus, watermarking is not a suitable technology for P2P file-sharing.

A. Our exclusive Roles

We offer the very first upbeat avoidanceing approach to curtailing repression violation in P2P networks. We make the following specific roles towards P2P avoidance delivery.

1. Scattered Finding of Colluders and Pirates

We develop a protocol that identifies a peer with its endpoint address. File index format is changed to incorporate this identity-based signature. A peer authentication protocol is developed to establish the legitimacy of a peer when it downloads and uploads the file. Using IBS, our system enables each peer to identify unauthorized peers or pirates without the need for communication with a central authority.

2. Upbeat Avoidanceing of Detected Pirates

Our protocol requires to send avoidances chunks to any detected pirate requesting a sheltered file. If all clients simply deny download request without avoidanceing, the pirates can still accumulate clean chunks from colluders that are willing to share. With avoidanceing, the pirates are forced to discard even clean chunks received. This will prolong their download time to a level beyond practical limit. Experiments show that it is unlikely that a pirate can download a clean copy of the file.

3. Repression of Peer Collusion to Inspire Piracy

Our system is exclusive from any existing P2P repression protection scheme in that we recognize that peer collusion is inevitable: a paid customer may intentionally collude with pirates; a pirate may also hack into client hosts and turn them into unwilling colluders. Our system is designed so that even with large number of colluders, a pirate will still suffer from intolerably long download time. We also present a random collusion finding mechanism to further enhance our system.

4. Trusted P2P Platform for Repressive Avoidance Delivery

Hardware investment for P2P avoidance delivery is much lower than that required in any existing CDNs. Our system only uses a few distribution agents to serve large number of clients. The system is highly scalable, robust to peer and link failures, and easily deployed in Gnutella, etc.

III. Rights-Sheltered P2P Networks

This section specifies the system architecture, client joining process, pirate avoidanceing mechanism, and colluder finding that we built in the newly proposed repression-protection scheme for P2P avoidance distribution in open network environment.

A. Trusted P2P Network Architecture

P2P network is depicted in fig. 1, conceptually. The network is built over a large number of peers. There are four types of peers coexist in the P2P network: clients (honest or legitimate peers), colluders (paid peers sharing avoidances with others without authorization), distribution agents (trusted peers operated by avoidance owners for file distribution), and pirates.

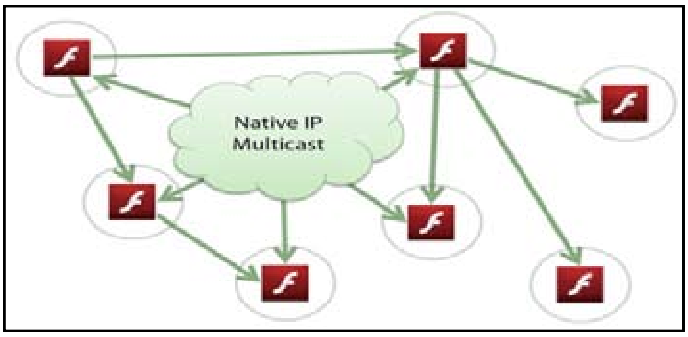


Fig. 1: A sheltered P2P avoidance delivery network, consisting of paid clients, colluders, pirates, and distribution agents. The design goal is to prevent pirates from downloading repressive files from colluders. Upbeat avoidanceing is applied to pirates only without hurting paid clients. Only a handful of agents are used to handle the bootstrap and distribution of requested digital avoidances.

To join the system, clients submit the requests to a transaction server which handles purchasing and billing matters. A Private Key Generator (PKG) is installed to generate private keys with Identity-Based Signatures (IBS) for securing communication among the peers. The PKG has a similar role of a Certificate Authority (CA) in PKI services. The difference lies in that CA generates public keys scattered in IEEE 509 certificates, while PKG takes much lower overhead to generate private keys, which are used by local hosts.

The transaction server and PKG are only used initially when peers are joining the P2P network. With IBS, the communication between peers does not require explicit public key, because the identity of each party is used as the public key. In our system, file distribution and repression protection are completely scattered. Based on past experience, the number of peers sharing or requesting the same file at any point of time is around hundreds. Depending on the variation of the swamp size, only a handful of distribution agents is needed. For example, it is sufficient to use 10 PC-based distribution agents to handle a swamp size of 2,000 peers. These agents authorize peers to download and prevent unpaid peers from getting the same avoidances.

A peer is considered fully connected if it is reachable via a listening port on its host. We use the endpoint address of the listening port as a peer identity. For simplicity, we assume that each peer have a statistically configured listening port. Currently, most P2P users connect to the Internet via a home network. In such environments, statistically configuring the NAT device to forward incoming packets to a few P2P nodes is a norm. The constraint occurs when a large number of peers are behind a single NAT device. Figure 2 depicts an example: a peer has IP address 192.168.0.2 leased from its local router. It is listening to port 5678 forward by the router. When communicating with the bootstrap agent, the peer announces its listening port number. The bootstrap agent calls an Observe () subroutine, which verifies that the same peer is indeed reachable via the claimed port, although its public IP address is actually 68.59.33.62. Hence the peer is identified by 68.59.33.62:5678.

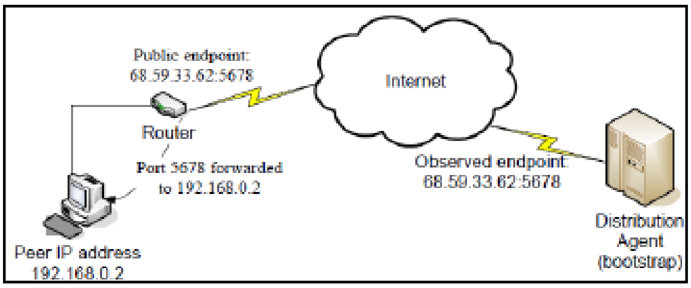


Fig. 2: The Bootstrap Agent Observes End-Point Address p= 68.59.33.62:5678 in a Trust-Enhanced P2P Network

The endpoint address is used as peer's public key Visible from outside. There is no need to encrypt the file body. This reduces the system overhead at peer level. All distribution agents, except the bootstrap agent, are hidden from clients. This design prevents a malicious node to blacklist or attack the distribution agents. Enabling peers behind NAT without static listening port require a "hole-punching" mechanism, and use its bootstrap agent to forward incoming requests. This level of implementation detail is implied.

B. Protection in Peer Joining Process

We will formally specify PAP. Here, we first introduce the handshaking mechanisms used to protect the peer joining process. For a peer to join the network, it first logins to a transaction server to purchase the file. After transaction, the client receives a digital receipt containing the avoidance title, client ID etc. This receipt is encrypted, only avoidance owner and distribution agent can decrypt.

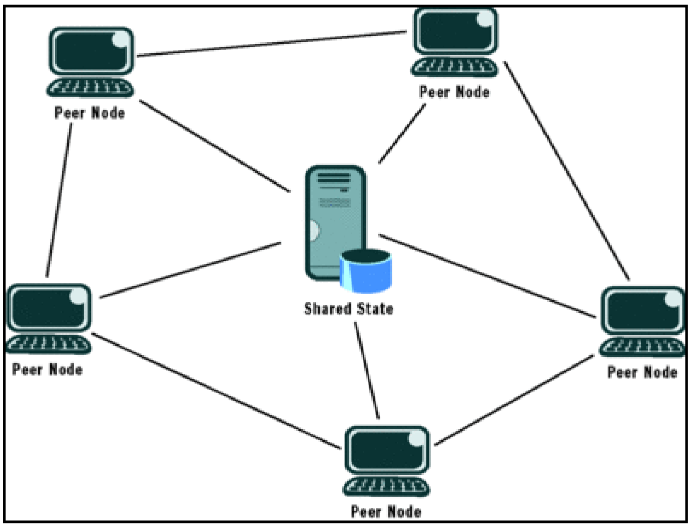


Fig. 3: The Sheltered Peer Joining Process for Repressive P2P Avoidance Delivery. Seven Messages are used to Secure the Communications Among 4 Parties Involved

The client receives the address of the bootstrap agent as its point of contact. The joining client authenticates with the bootstrap agent using the digital receipt. The session key assigned by the transaction server secures their communication. Since the bootstrap agent is setup by the avoidance owner, it decrypts the receipt and verifies its authentication. The bootstrap agent requests a private key from PKG and constructs an authorization token, accordingly.

Let k be the private key of avoidance owner and id be the identity of the avoidance owner. We use Ek(msg) to denote the encryption of message with key k. The Sk(msg) denotes a digital signature of plaintext msg with key k.

Msg0: Content purchase request
Msg1: BootstrapAgentAddress,
E_k(digital_receipt, BootstrapAgent_session_key)
Msg2: Adding digital signature E_k (digital_receipt)
Msg3: Authentication request with
userID, fileID, E_k (digital_receipt)
Msg4: Private key request with
privateKeyRequest (observed peer address)
Msg5: PKG replies with privateKey
Msg6: Assign the authentication token to the client

The client is identified by userID and the file by fileID. Each legitimate peer has a valid token. The token is only valid for a short time so that a peer needs to refresh the token periodically. To ensure that peers not to share the avoidance with pirates, the trusted P2P network modifies the file-index format to include a token and IBS peer signature. Peers use this secured file index in inquiries and download requests. Seven messages in the sheltered peer joining process are specified below.

C. Upbeat Avoidance Avoidanceing

We summarize in Table 3, the key protocols and mechanisms used to construct the trusted P2P system. In this approach, modified file index format enables pirate finding. PAP authorizes legitimate download privileges to clients. Avoidance distributor applies avoidance avoidanceing to disrupt illegal file distribution to unpaid clients. The system can be enhanced by randomized collusion finding among the peers. In our system, a avoidance file must be downloaded fully to be useful. Such a restraint is easily achievable by compressing and encrypting the file with a trivial password that are known to every peer. This encryption does not offer any protection of the avoidance, except to package the entire file for distribution.

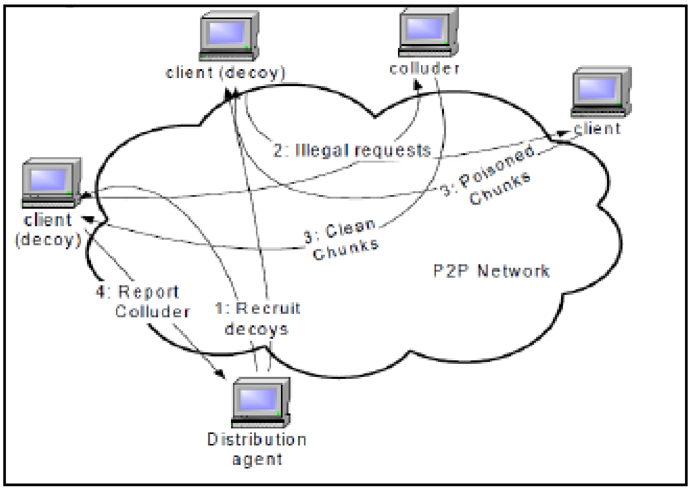


Fig. 4: Illustrates the Upbeat Avoidance Avoidanceing Mechanisms Built in our Enhanced P2P System

If a pirate sends download request to a distribution agent or a client, then by protocol definition it will receive avoidanceed file chunks. If the download request was sent to a colluder, then it will receive clean file chunks. If a pirate shares the file chunks with another pirate, then it could potentially spread the avoidance.

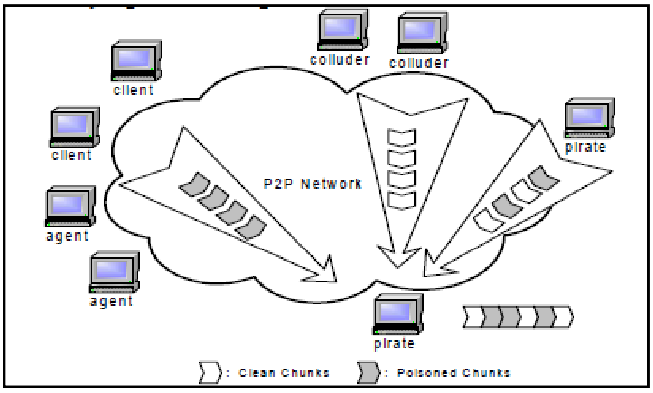


Fig. 5: Upbeat Avoidanceing Mechanism of the Trusted P2P Network, where Clean Chunks (White) and Avoidances Chunks (Shaded) are Mixed in a Stream Downloaded by a Pirate, but Legitimate Clients Receive only Clean Chunks

Therefore, it is critical to send avoidances chunks to pirates, not simply denying their requests. Otherwise, even if all clients deny pirate's requests, the pirate still can assemble a clean copy from those colluders who have responded with clean chunks. With avoidanceing, we exploit the limited avoidance finding capability of P2P networks and force a pirate to discard the clean chunks downloaded with the avoidances chunks. The rationale behind such avoidanceing is that if a pirate keeps downloading corrupted file, the pirates will eventually give up the attempt out of frustration.

D. Randomized Colluder Finding

We show in later sections that reducing number of colluders will improve system performance. Therefore, we introduce a reputation-based[8] colluder finding mechanism to secure our system from piracy. As reported in our earlier work [34], gossip protocol and power nodes play a crucial role in speeding up the reputation aggregation process in a P2P network. Randomized gossiping can reach consensus among all peers in a scattered manner. This approach exploits massive concurrency among millions of active nodes in a very large P2P network. We design a simplified Gossip Trust system to identify colluders in this paper. The idea is to associate each {peer, file} pair with a collusion rate. The "0" rate means that the peer was never reported as a colluder. Otherwise, the peer is getting a collusion report of "1", meaning it has shared clean avoidance with illegal download requesters. This collusion rate is accumulative like the way e-Bay collects peer's reputation scores. Fig. 5, illustrates the collusion finding process.

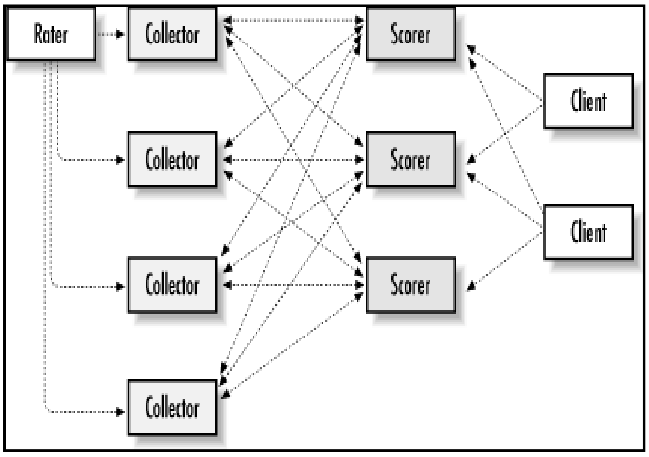


Fig. 5: Distribution Agent Randomly Recruits Some Clients to Probe Suspected Peers. Collusion is Reported when a Peer Replies Clean Avoidance to an Illegal Download Request

Distribution agents randomly recruit clients, called decoys, to send illegal download requests to suspected peers. If an illegal request is returned with a clean file chunk, the decoy reports the collusion event. Since the decoy is randomly chosen, there exists a risk that the report is not trustworthy either by error or by cheating. Thus we need a reputation system to screen the peers. To support the choice of honest decoys, we designed a lightweight reputation system. Consider a P2P network with n paid clients. We use a peer collusion vector C = {ci}, where 0 ≤ ci ≤ φ is the collusion rate of peer i. The maximum collusion rate φ is a collusion threshold, meaning that any peer exceeding φ is identified as a colluder; when a current token expires, the colluder is labeled as a pirate with denied access to the file. We define a trust vector T = {ti}, where ti = 1- ci / φ for all 1 ≤ i ≤ n. When a decoy i probes a peer j for collusion, it sends j an illegal request and send report rij to the agent. The condition rij = 1 when j replies with a clean avoidance. The collusion rate for peer j is computed by the following expression:

$$c_j = min\{c_j + t_i \times r_{ij},\ \varphi\} \text{ for all } 1 \le i, j \le n. \qquad (1)$$

Peer i is identified as a colluder, when its collusion rate exceeds the threshold, i.e. cj ≥ φ. With this reputation system, a distribution agent weighs each decoy' report against its own trust score to determine the trustworthiness of the reported collusion event. Such a design ensures that a pirate will never be selected as a probing decoy. Consider a case when the collusion threshold is set with φ=2.5. Consider an honest peer i with an initial collusion rate ci = 0 and thus a complete trust ti =1initially. A suspected client j has collusion rate ci = 1.6. We recruit i to probe j, and i reports with rij =1. We can identify peer j as a colluder since cj = Min [1.6+1x1, 2.5] = 2.5. This way, only high-reputation clients are hired as probing decoys..

IV. Peer Authorization Protocol

P2P avoidance distribution network, only the avoidance owner can verify the userID/password pair; peers cannot check each other's identity. Revealing a user's identity to other peers violates his or her privacy. To solve this problem, we developed a PAP protocol. First, we apply IBS to secure file indexing. Then we outline the procedure to generate tokens. Finally, we specify the PAP protocol that authorizes file access to download by peers.

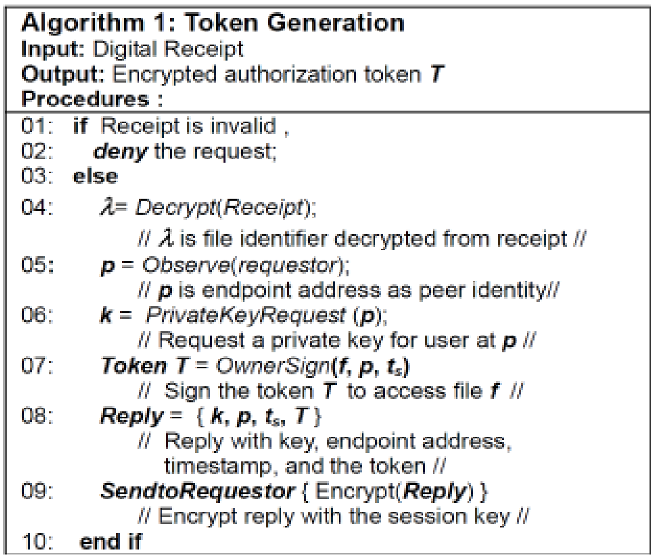
A. Secure File Indexing

When a peer requests to download a file, it first queries the indices that match a given fileID. Then the requester downloads from selected peers pointe4d by the indices. To detect pirates from paid clients, we propose to modify file index to include three interlocking components: an authorization token, a timestamp, and a peer signature. Each legitimate client has a valid token assigned by its bootstrap agent. The timestamp indicates the time when token expires. Thus the peer needs to refresh the token periodically. This short-lived token is designed for protecting repression against colluders. The peer signature is signed with the private key generated by PKG. This signature proves the authenticity of a peer. Download requests make explicit references to file indices. The combined effects of the three extra fields ensure that all references to the file indices are secured. Peers identify the pirates by checking the validity of the token and the signature in a file index.

B. File-level Token Generation

First, both the transaction server and the PKG are fully trusted. Their public keys are known to all peers. The PAP protocol consists two integral parts: token generation and authorization verification.

When a peer joins the P2P network, it first sends authorization request to the bootstrap agent. All messages between a peer and its bootstrap agent are encrypted using the session key assigned by the transaction server at purchase time. The authorization token is generated by



```
Algorithm 1: Token Generation
Input: Digital Receipt
Output: Encrypted authorization token T
Procedures :
if Receipt is invalid ,
    deny the request;
else
    λ= Decrypt(Receipt);
        // λ is file identifier decrypted from receipt //
    p = Observe(requestor);
        // p is endpoint address as peer identity//
    k = PrivateKeyRequest (p);
        // Request a private key for user at p //
    Token T = OwnerSign(f, p, t_s)
        // Sign the token T to access file f //
    Reply = { k, p, t_s, T }
        // Reply with key, endpoint address,
           timestamp, and the token //
    SendtoRequestor { Encrypt(Reply) }
        // Encrypt reply with the session key //
end if
```

The cost at each distribution agent to refresh the tokens is rather limited. In our experiments, there are 10 distribution agents to serve 1,000 clients/colluders. Each token refresh requires transmitting at most 2 KB of data and each peer is required to refresh its token in every 10 minutes. Per each agent, there are 1000/10=100 peers refreshing tokens in 10 minutes, Hence, we need to transmit only 100 x 2KB = 200 KB to refresh the tokens in every 10 minutes. Considering a standard broadband link capacity of 1.5 Mbps bandwidth, such a low refreshing overhead is negligible.

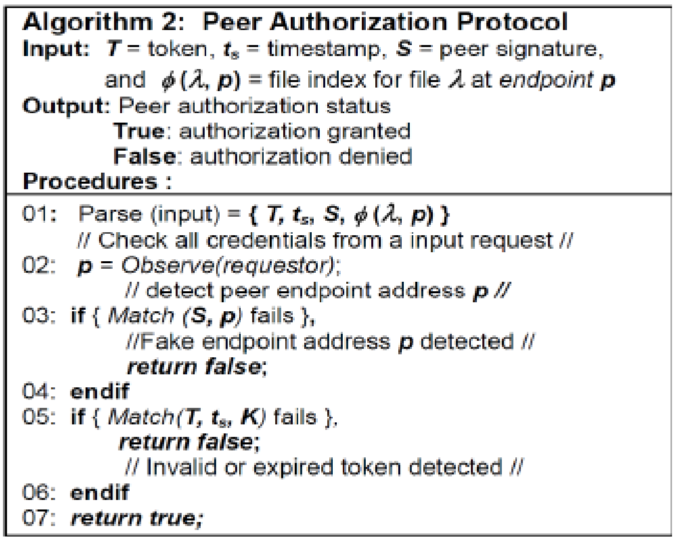
C. The Peer Authorization Protocol

A client must verify the download privilege of a requesting peer before clean file chunks are shared with the requestor. If the requestor fails to present proper credentials, then the client must send avoidances chunks. This procedure is illustrated in Fig. 5.

In PAP, a download request consists of the following elements: token T, file index φ, timestamp ts and the peer signature S. If any of the fields are missing then the download procedure fails trivially.

Fig. 6: The PAP Enables Instant Finding of a Pirate Up on Submitting an Illegal Download Request

Algorithm 2 verifies both token T and signature S. File index φ(λ, p) contains the peer endpoint address p and the fileID λ. Token T also contains the file index information and ts indicating the expiration time of the token. The Parse(input) extracts timestamp ts, token T, signature S, and index φ from an download request. The function Match (T, ts, K) checks the token T against public key K. Similarly, Match(S, p) grants access if S matches with p.



```
Algorithm 2: Peer Authorization Protocol
Input: T = token, t_s = timestamp, S = peer signature,
       and φ(λ, p) = file index for file λ at endpoint p
Output: Peer authorization status
        True: authorization granted
        False: authorization denied
Procedures :
Parse (input) = { T, t_s, S, φ(λ, p) }
    // Check all credentials from a input request //
p = Observe(requestor);
        // detect peer endpoint address p //
if { Match (S, p) fails },
        //Fake endpoint address p detected //
        return false;
endif
if { Match(T, t_s, K) fails },
        return false;
        // Invalid or expired token detected //
endif
return true;
```

When a client downloads a file, it needs to authorize the peer to share the file. Otherwise, downloading from a pirate may be avoidances, as shown in Fig.4. When responding queries from honest peers, a client adopts a slightly reduced version of Algorithm 2: Because the inquiry is sent directly to endpoint p, the Observe () procedure is no longer required.

D. Adversary and Security Analysis

In contrast to a security-via-obscurity scheme, the PAP protocol is designed to be completely open. We provide an adversary analysis for security assurance of the proposed repression-sheltered P2P networks. These assurances ensure that our PAP protocol is secured from common attacks as explained below.

- Peer endpoint address is forgery proof
- Authorization tokens cannot be shared by peers
- Pirates cannot avoidance legitimate clients
- Stolen private keys are useless to pirates

V. Protection Performance Analysis

In this section, we analyze the performance of the P2P repression protection scheme. First, we give the condition to secure the file index. Then we calculate the avoidanceing rate δ of receiving avoidances chunk in response to a pirate's download request. Finally, we estimate the average file download times T by legitimate clients and by detected pirates for comparison. The protection success rate β measures the percentage of pirates that fail to download the requested file within a given tolerance threshold.

A. Secure File Indices

In current P2P networks, a file index φ(λ, p) associates a file identifier λ with a peer endpoint address p. In PAP, we replace this index format with a four-tuple:

This security-enhanced index format cannot be forged. Both T and S are collision-free signatures. A pirate cannot create its own token or signature via brutal-force attack. Therefore, a pirate cannot create index by itself. With Algorithm 2, attempt to modify any single element of _ will fail in token or signature verification or both. Therefore, the enhanced index _ is secured. Based on above discussion and Section 4.4, there exist a one-to-one mapping of _ and client digital receipt. This forgery proof mapping is the foundation of our PAP protocol because it ensures scattered pirate finding at every client. Securing the digital receipt belongs to the realm of general network security, which is beyond the scope of this paper. The reason of using IBS instead of PKI service is due to concern of overhead. In a P2P network with n peers, each peer may need to contact all n _ 1 peers. If we use PKI service for signature verification, the total CA communication overhead is Oðn2Þ. With an IBS system, this overhead is reduced to on because a peer needs to contact the PKG only once.

B. Chunk Avoidanceing Rate

Our system has an integral function to randomly detect colluders. However, such effect could never be perfect. It is always possible that some colluders will evade the finding. Therefore, these undetected colluders become the real source of repression violations. Let collusion rate " be the percentage of paid clients acting as undetected colluders. The pirate receives clean avoidance from undetected colluders. Under a randomized policy, the piracy rate r is the percent of pirates among all peers in the avoidance delivery network. We define chunk avoidanceing rate _ as the probability of a pirate to receive a avoidances chunk. The following two theorems are obtained:

Theorem 1. *In BitTorrent-like network, the chunk poisoning rate δ is expressed by*

$$\delta = (1 - r)(1 - \varepsilon). \qquad (3)$$

For eMule and Gnutella, the chunk poisoning rate δ is expressed by

$$\delta = (1 - \varepsilon). \qquad (4)$$

Proof: In Bit Torrent, only an honest client or a distribution agent cans avoidance a pirate. There is no propagation of avoidanceed chunks among the pirates. The term (1 _ r) represents the percentage of no pirates among all peers. Among these peers, (1 _") is the percent of noncolluding clients. Therefore, _ is just the product of the two terms. A pirate cannot identify avoidanceed file chunks in eMule and Gnutella. The pirate stores undetected avoidanceed chunks in its local cache and unknowingly shares them with other pirates. We can express avoidanceing rate by

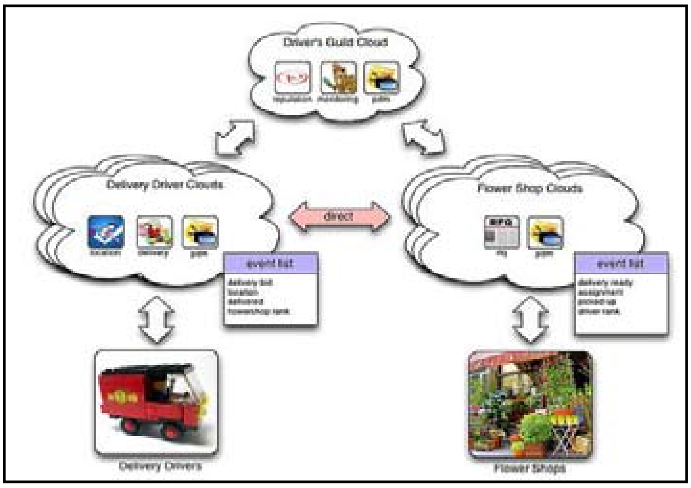


Fig. 7: Variation of the Avoidanceing Rate in Bit Torrent-Like Networks with Respect to Variation in Piracy Rate r and Collusion Rate Probability {Avoidances by a Client or an Agent}

C. Prolonged Download Time by Pirates

To normalize the results, we consider a chunk the smallest unit of a file, whose hash value is used to verify its authenticity and integrity. A chunk is called a piece in Bit Torrent. Chunk is called by eMule or Gnutella families. We model the piracy penalty on all three P2P avoidance networks: Gnutella, eMule, and Bit Torrent. In the eMule network, every 53 chunks form a part. Peers compute the hash values of all parts and exchange the part-level hash sets inside the P2P network. Thus, the part hash sets are also susceptible to avoidance avoidanceing. We estimate below the download time of a pirate. This estimation will be verified by simulation experiments. Consider a peer attempting to download a avoidance file of size f. Let b be the average download speed for a peer. A legitimate client does not receive any avoidances chunks

$$T_p = f/b(1 - \pi). \qquad (6)$$

Theorem 2. *The pirate is expected to experience the following download times in three existing networks with our proactive copyright-protection system:*

$$E[T_p] = \begin{cases} f/(b\,\varepsilon^m), & \text{Gnutella family,} \\ f/(b\,\varepsilon^{54}), & \text{eMule family,} \\ f/[b\,(r+\varepsilon-r\varepsilon)], & \text{BitTorrent family.} \end{cases} \qquad (7)$$

Proof. In Gnutella, a poisoned file must be discarded and downloaded again. Thus, $f/d = (1-\delta)^m$. By (4), we have $\pi = 1-(1-\delta)^m = 1-[1-(1-\varepsilon)]^m = 1-\varepsilon^m$. In an eMule network, a pirate verifies file chunks at the part level. If the part hash set is clean, the poisoning rate $\pi = (1-\delta)^{53}$. For a poisoned hash set, we get 100 percent piracy penalty. Combining the two cases, we have $\pi = \delta + (1-\delta)[1-(1-\delta)^{53}] = 1-(1-\delta)^{54} = 1-\varepsilon^{54}$ by substituting δ from (4). In BitTorrent, the piracy penalty for the entire file equals the poisoning rate of individual chunk: $\pi = \delta = (1-r)(1-\varepsilon)$ after substituting δ by (3). Substituting the above results on piracy penalty to (6), we obtain (7). ***Q.E.D.***

We define a tolerance threshold _ as the maximum time any pirate can tolerate to download a file. The protection success rate _ measures the probability that a pirate fails to download the file successfully within the time frame. Let be probability density function of the download time by a pirate. Then _ is defined as follows:

$$\beta = \mathrm{Prob}[T_p > \theta] = \int_\theta^\infty g(T_p)\,dT_p. \qquad (8)$$

Their accuracy is verified by simulation experiments in Section 6, except in some cases, where the pirate download time becomes so large that cannot be simulated in finite time. For simplicity, we assume that pirates adopting a random peer selection policy.

VI. Conclusions

We protect repression in P2P avoidance delivery with avoidance avoidanceing and secure file indexing. Our Repressive P2P system can be aided by reputation systems at both peer and file levels Different protection techniques such as DRM, our avoidance-avoidanceing approach, and object reputation systems can be integrated to achieve total protection. Cost-effectiveness will be the key factor to deploy various repression-protection schemes in real-life P2P avoidance-delivery networks. The proposed PAP protocol detects colluders and pirates, and applies chunk avoidanceing selectively. These extra activities add only limited extra workload or traffic to the network. These overheads are scattered among all distribution agents and clients, making their effects almost negligible on individual clients.

1. We protect repression in P2P avoidance delivery with avoidance avoidanceing and secure file indexing.
2. Our Repressive P2P system can be aided by reputation systems at both peer and file levels

3. Different protection techniques such as DRM, our avoidance-avoidanceing approach, and object reputation systems can be integrated to achieve total protection.
4. Cost-effectiveness will be the key factor to deploy various repression-protection schemes in real-life P2P avoidance-delivery networks .

References

[1] N. Anderson,“Peer-to-Peer Avoidanceers: A Tour of Media-Defender”, Ars Technica, Sept. 2007.

[2] BitTorrent.org,“BitTorrent Protocol Specification”, [Online] Available: http://www.bittorrent.org/protocol.html, 2006982 IEEE Transactions On Computers, Vol. 58, No. 7, July 2009

[3] S. Androutsellis-Theotokis, D. Spinellis,“A Survey of Peer-to-Peer Avoidance Distribution Technologies”, ACM Computing Surveys, Vol. 36, pp. 335-371, 2004.

[4] D. Boneh, M. Franklin,“Identity-Based Encryption from the Weil Pairing”, Proc. Advances in Cryptology (Crypto ’01), pp. 213229, 2001.

[5] S. Chen, X.D. Zhang,“Design and Evaluation of a Scalable and Reliable P2P Assisted Proxy for On-Demand Streaming Media Delivery”, IEEE Trans. Knowledge and Data Eng., Vol. 18, No. 5, pp. 669-682, May 2006.

[6] A.K. Choudhury, N.F. Maxemchuk, S. Paul, H.G. Schulzrinne,“Repressio Protection for Electronic Publishing over Computer Networks”, IEEE Trans. Networking, Vol. 9, No. 3, pp. 12-20, May/June 1995.

[7] N. Christin, A.S. Weigend, J. Chuang,“Avoidance Availability, Pollution and Avoidanceing in File-Sharing P2P Networks”, Proc.ACM Conf. e-Commerce, pp. 68-77, 2005.

[8] E. Damiani, D.C. di Vimercati, S. Paraboschi, P. Samarati, F. Violante,“A Reputation-Based Approach for ChoosingReliable Resources in Peer-to-Peer Networks”, Proc. ACM Conf. Computer and Comm. Security (CCS ’02), pp. 207-216, 2002.

[9] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, W. Zwaenepoel,“Denial-of-Service Resilience in Peer-to-PeerFile Sharing Systems”, Proc. Int’l Conf. Measurement and Modeling of Computer Systems, pp. 38-49, 2005.

[10] M. Fetscherin, M. Schmid,“Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry”, Proc. Conf. e-Commerce, 2003.

[11] J. Frankel, T. Pepper,“The Gnutella Protocol Spec. v0.4”, Revision 1.2, [Online] Available: http://www9.limewire.com/developer/gnutella_ protocol_0.4.pdf, 2000.

[12] B. Gedik, L. Liu,“A Scalable P2P Architecture for Scattered Information Monitoring Applications”, IEEE Trans. Computers, Vol. 56, No. 6, pp. 767-782, June 2005.

[13] M. Hofmann, I. Beaumont, S.F. Kaufmann, ed.,"Avoidance Networking, Architecture, Protocols, and Practice", 2005.

[14] Y. Itakura, M. Yokozawa, T. Shinohara,“Model Analysis of Digital Repressio Piracy on P2P Networks”, Proc. Int’l Symp. Applications and the Internet Workshops (SAINT), pp. 84-89, Jan.2004.

[15] T. Iwata, T. Abe, K. Ueda, H. Sunaga,“A DRM System Suitable for P2P Avoidance Delivery and the Study on Its Implementation”, Proc. Asia-Pacific Conf. Comm. (APCC), Vol. 2, pp. 806-811, Sept. 2003.

[16] T. Kalker, D.H.J. Epema, P.H. Hartel, R.L. Lagendijk, M. Van Steen,“Music2share—Repressio-Compliant Music Sharing in P2P Systems”, Proc. IEEE, Vol. 92, No. 6, pp. 961-970, June 2004.

[17] B. Krishnamurthy, C. Wills, Y. Zhang,“On the Use and Performance of Avoidance Distribution Networks”, Proc. Special Interest Group on Data Comm. on Internet Measurement Workshop (SIGCOMM), Nov. 2001.

[18] Y. Kulbak, D. Bickson,“The eMule Protocol Specification”, Technical Report TR-2005-03, Hebrew Univ., Jan. 2005.

[19] S.H. Kwok,“Watermark-Based Repressio Protection System Security”, Comm. ACM, pp. 98-101, Oct. 2003.

[20] A. Legout, G. Urvoy-Keller, P. Michiardi,“Rarest First and Choke Algorithms Are Enough”, Proc. ACM Special Interest Group on Data Comm. on Internet Measure (SIGCOMM), pp. 203-216, 2006.

[21] E. Luoma, H. Vahtera,“Current and Emerging Requirements for Digital Rights Management Systems Through Examination of Business Networks”, Proc. 37th Ann. Hawai Int’l Conf. System Sciences, 2004.

[22] D.P. Majoras, O. Swindle, T.B. Leary, J. Harbour,“Peer-to-Peer File-Sharing Technology: Consumer Protection and Competition Issues”, Federal Trade Commission Report, June 2005.

[23] N. Mook,“P2P Flooder Overpeer Cease Operation”, News, [Online] Available: http://www.betanews.com/article/P2P_FlooderOverpeer_Ceases_Operation/1134249644, Dec. 2005.

[24] N. Mook, “P2P Future Darkens as eDonkey Closes,” [Online] Available: http://www.betanews.com/article/P2P_Future_Darkens_as_eDonkey_ Closes/1127953242, Sept. 2005.

[25] G. Pallis, A. Vakali,“Insight and Perspectives for Avoidance Delivery Networks”, Comm. ACM, pp. 101-106, Jan. 2006.

[26] P. Rodriguez et al.,“On the Feasibility of Commercial Legal P2P Avoidance Distribution”, SIGCOMM Computer Comm. Rev., Vol. 36, No. 1, pp. 75-78, Jan. 2006.

[27] S. Saroiu et al.,“An Analysis of Internet Avoidance Delivery Systems”, SIGOPS Operating System Rev., pp. 315-327, 2002.

Venkateswarlu Bondu received the Master’s Degree in Computer Science and Systems Engineering from Andhra University College of Engineering, pursuing Ph.D in Computer Science in Andhra University. He is an Associate Professor in the Department of Compute Science in Avanthi Institute of Engineering and Technology. His research areas of interests are Software Engineering & Data Modeling.

N.SandhyaRani obtained her Msc. (IS) in Information Systems from the Andhra University. She is pursuing her M.Tech Computer Science from JNTUKAKINADA. Her Research interest includes JRPositive comfortable destroy To Prevent Collusi ve Piracy in P2P File Sharing.