A Proposed Piracy Prevention Technique in P2P Network

Abhijit R. Jain, Saurabh Malgaonkar, Shailja Sumeet

Mukesh Patel School of Technology Management & Engineering,
Department of Computer Engineering
NMIMS University, Mumbai, Maharashtra, India

Abstract

Collusive piracy is first and fundamental cause of violation to intellectual property law. Illegally unpaid clients would get copyright contents from colluders. Such type of piracy causes to destruct to content delivery in P2P network. Proposed system aims to prevent colluders from pirating the copyright contents in P2P. The fundamental idea behind implementation of the system is to provide a security mechanism with identity based signature (IBS) and content poisoning technique. The proposed system assists to reduce the piracy percentage without disturbing paid clients by sending unnecessary data i.e. poisonous chunks to violators. Proposal to make Legal Client Authorization Protocol (LCAP) is to differentiate unpaid client form paid ones. Unique time stamp will be assigned to copyright content file which can be downloaded by paid clients exclusively. Based on simulation results, it is found that 90 percent prevention rate in Gnutella. These P2P-based CDNs gives very fast delivery speed, higher content availability and cost-effectiveness than traditional CDNs built with huge network of surrogate servers.

1. Introduction

All Peer-to-Peer (P2P) networks are networks having no clients and no servers i.e. all the nodes are considered as both client and server depending upon the operation. Peer-to-Peer networks are very much cost-effective in delivering huge size files to large number of users. However, today's P2P networks are abused by illegal distributions of movies, music, games, video streams and lots of software which leads to piracy. These resulted in not only heavy financial losses in media and content industry but also restricted the legal commercial use of P2P technology. Sources to illegal file sharing are peers or the clients who breach copyright laws and help the pirates.

This proposed system aims to find the solution to solve such a kind of collusion problem, by a copyright-compliant system for legal P2P content delivery which any how prohibits pirates in the network. The actual aim is to stop collusive piracy within the boundary of a P2P content delivery network without hampering the existing paid clients. This scheme forces to protect huge amount of valuable contents that reduce in value as time to move by. CDN uses big number of content servers, world spread distributed among the WANs. As content distributors need to save number of copies of contents over many servers, the bandwidth demand and resources needed to maintain these CDNs are very expensive due to that get poor performance over CDNs, which is a drawback of CDNs and this drawback, is addressed by P2P network. P2P content network reduces the distribution cost as many content servers are removed from the network and open networks are going to be used resulted maximum bandwidth demand of the clients can be achievable.

Content availability, content security is improved by P2P networks as any peer can serve as content copyrighted files, even in the presence of colluding peers or the pirates. Reputation scheme is used to find these colluders. Both media industry and internet user communities get the benefit of a copyright-protected P2P network in a large extent. This work leads to reduces piracy in large scale by the development of a new generation of CDNs based on P2P technology.

Table 1 shows the lists important symbols and notations used to benefit our readers. These terms are used to secure file indexes, generate access tokens, quantify poisoning effects, collusion prevention, and define the performance metrics [5].

The main focus is on finding solution for collusive piracy within the scope of a P2P network. Fundamental purpose is to stop colluders from posting content data freely on the network and provide less effort for the paid clients for clean chunks. At current stage, these direct point-to-point copyright violation problems are mostly handled by digital rights management (DRM) techniques. Though it can handle by DRM but even the protection results are not considered satisfactory as many hackers have post DRM-cracks on internet which causes piracy [5].

Term, Symbol	Brief Definition
Access token, T	A short-life token for file access control
Time stamp, t_s	Used in securing file index /query/requests
User address, p	User endpoint address observed by agent
File index, ϕ	Pointer to access the requested content file
Clean file size, f	Original file size in bytes without poisoning
Download file, d	Actual bytes downloaded, ($d \geq f$)
Poisoning rate, δ	Probability of getting a poisoned chunk
Chunk number, m	Number of chunks in a single content file
Collusion rate, ε	Percentage of paid peers acting as colluders
Piracy rate, r	Percentage of pirates detected
Download times, T_c and T_p	Expected times to download a clean file by a paid client and a detected pirate, respectively
Tolerance, θ	Maximum download time tolerable by peers
Success rate, β	Probability of detecting a pirate

Table.1

Large file which is uploaded on a network to freely download is subdivided into small chunks, now P2P network allow peers to download the different chunks of original file from various sources. Availability will be increased by file chunking and it also shortens download time.

Classification of open P2P networks is based on file chunking and hashing protocols. So based on this we classify three network families as:

1. BitTorrent 2. Gnutella 3. eMule

The BitTorrent family applies the strongest hashing at individual piece or chunk level, which is most resistant to poisoning. The Gnutella family applies file-level hashing, which is easily poisoned. The eMule family applies TABLE 1 Parameters and Notations, Chunking, Hashing, Poisoning, and Download Policies in P2P Content Networks part-level hashing with fixed chunking.

2. LITERATURE SURVEY

2.1 Methodology

Security threat to P2P networks is a content poisoning event. This is the very first proactive poisoning approach to shorten copyright violation in P2P networks. The following specific contributions towards P2P content delivery has made by us as

2.1.1 Distributed Discovery of Colluders as well as Pirates

There is a methodology to develop a proposal to develop a protocol that verifies a peer with its IP address. File index format is changed to incorporate a digital signature based on this identity. A LCAP is developed to establish the legitimacy of a peer when it downloads and releases the data. Using Identity Based Signature, our system allows each peer to identify or pirates without the need for communication with a central authority.

2.1.2 Proactive Content Poisoning of Discovered Pirates

Proposed protocol requires sending poisoned chunks to any discovered pirate requesting a protected file. If all clients simply deny download request without poisoning, the pirates can still accumulate clean chunks from colluders that are willing to share. With poisoning, pirates are forced to discard even clean chunks received. This will prolong their download time to a level beyond practical limit.

2.1.3 Containment of Peer Collusion to Stage Piracy

Our system is unique from any existing P2P copyright 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 detection mechanism to further enhance our system.

2.2 BitTorrent

P2P networking provides an efficient way to share resources amongst a large number of users and they have been widely used in the Internet applications. Unfortunately, illegal entities may use P2P to disseminate copyrighted materials without the owner's permission. As reported by Envisional at least 23.76% of Internet traffic is piracy content, 75% of which uses BitTorrent (BT). Those contents include movies, music, games and softwares. These abuses discourage content providers and also impede the wide application of P2P technologies [4].

A torrent is the metadata that stores the descriptive information of a file. A torrent is important for the users because it contains necessary information to bootstrap users into a collection of peers (also called a swarm).

BitTorrent Basics:

1. Seeders

A seed is a peer that has already downloaded a file and is willing to provide the file to other peers even though it does not need any more contents [3, 4].

2. Lechers

A lecher is a peer that has downloaded part of a file. A lecher provides part of a file (that it downloaded) to the other peers, and meanwhile it downloads the rest of the file from the other peers [3, 4].

3. Torrent Index

A torrent index is a set of ongoing torrents that are collectively organized in the form of torrent

websites, which allow users to upload their torrents and provide tracker services [3, 4].

4. Peer Index

A peer index is a set of peers that participate in the distribution of a specific file. The basic function of a peer index is to track the status of peers that are currently active and acts as a rendezvous point for all peers [3, 4].

3. Legal Client Authorization Protocol (LCAP)

In P2P CDNs peers present in the network doesn't know each other's identity such as a peer network address (IP address). Revealing a user's identity to other peers assaults his or her privacy. By the combination of user ID/password every peer gets logged into the network, which is verified by content owner only, other users are unaware about that. In proposed system one protocol named as Legitimate User Authorization Protocol is developed to overcome this problem. Protocol is divided into three parts of security are as follows:
1) Secure file indexing using IBS
2) File level Token generation
3) LCAP protocol [1].

3.1 Applying IBS to secure file indexing

To differentiate pirates from paid clients, we proposed to update file index with three strongly connected components an authorization token, a time stamp, a peer signature. File indexing is useful for mapping File ID and peer end point address in P2P network before updating. When a peer wants to download any file, it first sends requests the indexes that matches given file ID [1]. Then the peer, who wants to download, downloads from currently available peers pointed by indexes. Bootstrap agent assigns a valid token to its legitimate client. Validity of the token depends on the time measure which is stored in time stamp component. After this time client has to get renew his token by distribution agent only. This token is important for protecting copyright of that file against colluders. Cost of refreshing tokens by the distribution agent of client has its boundaries limited. The peer signature is the form of digital signature which is contains PKG generated private key, which authenticate peers [2].

3.2 Generation of token for file

Trusted components in P2P network are transaction server and the public key generator (PKG) whose public keys are known to all peers. In LCAP protocol, we are going to consider two parts as generation of token and verification of peer. To join P2P network every peer needs to send a request to bootstrap agent which verifies the peer. Communication of peer and its bootstrap agent takes place through encrypted messages (these messages are encrypted using session key assigned by transaction server at the time of purchasing). For generation of token for a specified file we use algorithm called token generation algorithm. A token is a collection of 3 tuples file ID, peer endpoint, timestamp. Also we can say that token is a digital signature signed by private key of content owner. In this algorithm, we are passing digital receipt as input. Transaction server sends digital receipt, generated at the time of purchasing of file to bootstrap agent. As peer is also sending same digital receipt to bootstrap agent, bootstrap agent matched that key with the copy of digital receipt sent by transaction server. If digital receipt is valid then token is generated for respective file and peer, otherwise it will deny the request [1].

Algorithm 1. Token Generation

Input: Digital Receipt
Output: Encrypted authorization token T
Procedures:
01: if Receipt is invalid,
02: deny the request;
03: else
04: ϒ = Decrypt (Receipt);
// ϒ is file identifier decrypted from receipt //
05: p = Observe (requestor);
// p is endpoint address as peer identity//
06: k = PrivateKeyRequest (p);
// Request a private key for user at p //
07: Token T = OwnerSign(f; p; ts)
// Sign the token T to access file f //
08: Reply = fk; p; ts;Tg
// Reply with key, endpoint address,timestamp, and the token //
09: SendtoRequestor {Encrypt (Reply)}
// Encrypt reply with the session key //
10: end if [1].

3.3 The Legal Client Authorization Protocol

In LCAP, as shown in above figure a client, who could be legitimate or pirate, requests to download a file with three parameter token, timestamp and signature. All these fields are important and must for download file. Token and timestamp are used to verify token is valid or invalid. Signature is also get verified. If any one of the two, token or signature is invalid then poisoning chunks have been sent to requestor, otherwise clean file chunks are get sent.

Firstly, if invalid token is detected then that token could be of legitimate client and might be expired or it could be of pirate. Second one, if signature is invalid then fake end point of requestor is detected. Either of the two cases downloading of requested file will be stopped.

Algorithm 2. LCAP

Input: T = token, ts = timestamp, S = peer signature,
and f(y,p) = file index for file y at endpoint p
Output: Peer authorization status
True: authorization granted
False: authorization denied
Procedures :
01: Parse (input) = T; ts;S;f(y,p)
// Check all credentials from a input request //
02: p = Observe(requestor);
// detect peer endpoint address p //
03: if {Match (S; p) fails},
//Fake endpoint address p detected //
return false;
04: endif
05: if {Match(T; ts;K) fails},
return false;
// Invalid or expired token detected //
06: endif
07: return true[2].

4. Proposed System

A P2P computer network is one in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to various resources such as files, peripherals and sensors without the need for a central server. Here, conceptual architecture, peer identification, establishment of connection, copyright content poisoning is explained.

4.1 Proposed System Architecture

Proposed system contains the LCAP (Legitimate Client Authorization Protocol) server which handles the user's transactions. The network contains different types of peers like clients (legal peers), paid client (peers sharing contents with other), intruders/pirates (peers which uses file illegally). One more component is installed to generate the private key for peers' i.e. private key generator (PKG). To get connected to network, peer requests the LCAP server which completes the purchasing process and transactions. On completion of the transaction private key is assigned to peer by PKG. That key is used to communicate among peers in the network. For better service, another type of peer is introduced in the system i.e. distributor. It authorizes the peer to download & prevent s the unpaid clients from getting same contents [5].

4.2 Peer Identification

In P2P network, all the peers (clients, colluders, intruders) are all mixed up. It is important to distinguish between them. For that each peer is assigned with distributor. Distributor has its unique port address while peer has its unique IP address. Combining these two addresses, peer is identified. In home network environment, it is necessary to configure NAT device to forward the incoming port to peer node statically. The constraint occurs when large numbers of peers are behind the single NAT device. Example: Peer is having IP address 45.67.89.23 and it is listening to the port number 5678 of the distribution.

4.3 Connection Establishment

The client or peer requests to LCAP server for any copyrighted content. LCAP server makes transactions and replies with address of distributor, digital receipt and session key. Session key is useful for communication for particular session only. Distributor decrypts digital signature and authenticates peer. Then it requests PKG for private key. On getting private key, distributor generates authorization token [1].

Msg0: Purchase request
Msg1: Reply from server
Msg2: Digital Receipt to distributor
Msg3: Authentication Request from user.
Msg4: Request for private key generation.
Msg5: Reply from PKG with private key.
Msg6: Generation of token.

4.4 Copyright Content Poisoning

Copyright contents are made bulky by adding poisonous chunks to them. LCAP authorizes legal download privilege to clients. Content poisoning is done to disrupt illegal file distribution to pirates. If pirates make request to client or distributor, it will get only poisonous chunks. Exactly opposite if it makes request to paid client (colluder), it will receive clean chunks. And if it request to other pirate, it will receive mixed contents i.e. clean + poisonous chunks. For file to be useful it should get downloaded fully and if pirate keep downloading the poisonous chunks, it will give up attempt out of frustration [1, 2, 5].

4.5 Adversary and Security Analysis

Hacker can attack on this protocol to crack the system. These attacks are given below along with their solution stating why these types of attacks will get failed against the LUAP protocol. This ensures that our protocol is secured for implementation.

4.5.1 If pirate tries to poison legitimate client

Our system uses file indexing format which contains token and signature. Every client checks the valid signature through file indexes. It can only get connected to other legitimate clients. Though pirate wants to send poisoned chunks to legitimate clients, it cannot send them.

4.5.2 If pirate steals private keys

Pirate can gain private keys by hacking into the legitimate client system or colluders may share it with pirate. LUAP protocol does not depend totally on private key it also needs peer end point as public key. This public key is obtained by other peers in the network by using observe () procedure .So stolen private keys are useless for pirates.

4.5.3 If pirate steals token

As explained above tokens are used to verify legitimate clients within peers. This token is generated using three tuple as file ID, peer endpoint, timestamp. And peer endpoint is different for different peer. That's the reason peer endpoint is added in token. So, stolen tokens are also useless for pirates.

5. Protection Performance Analysis

Here, the performance of the P2P copyright protection system is take place. Initially, we put the terms to protect the file index. Further, we evaluate the poisoning rate P of arriving poisoned chunk in concern to a pirate's download request. Lastly, we summarize the average file download span T by payable customer and detected pirates for comparison. β is the protection success rate which measures the percentile of pirates that unable to download the requested file within a given tolerance threshold [1,2,5].

5.1 Secure File Indexes

File index $\phi(\lambda,p)$ in present P2P networks relates file identifier λ with a station endpoint address p. In LCAP, we replace this index style with a four-tuple style: $\Phi=\{\phi(\lambda,p), T,ts, S\}$.
Where T, S : collision free signatures.
Such enhanced index format cannot be copied. Token or signature via brutal-force attack cannot be generated by pirates its own. Thus, self fake index cannot be created by pirate's itself. If pirate want to alter or modify the single bit of four tuple index then should fail in token or signature verification or both. Thus, this enhanced index is strongly secured. There exists a 1:1 mapping of Φ and customer digital receipt. This special mapping is the basic of our LCAP protocol as it ensures scattered or distributed pirate searching at every client. Securing the digital receipt is not our aim. Nevertheless PKI service, IBS is used due to concern of overhead in PKI services. Each peer may need to contact all n -1 peers. In a P2P network with n peers, by using the IBS despite of PKI overhead to CA communication get reduce to O(n) from O(n^2)[1,5].

5.2 Chunk Poisoning Rate

An integral function has been used to randomly detect colluders, in our proposed system. Such effect could not be precise. There might be possibility that some unauthorized outsiders will escape the detection. Thus the original sources of copyright violations are these undetected colluders [1].

6. Future Scope

By combining above techniques, algorithms and methodologies from deferent research papers a new system will be developed which will banned some sort of piracy. Mainly IBS, content poisoning technique and end point address will be used to develop this system.

7. Conclusion

From the above proposal it can be conclude that, the new concept for file indexing using three tuples file ID , peer end point address & Digital Signature and poisoning method are useful for preventing collusive piracy.

8. References

[1] Xiaosong Lou and Kai Hwang, "Collusive Piracy Prevention in P2P Content Delivery Network," IEEE Trans computers, vol. 58, no. 7 J 2009.
[2] Xiaosong Lou, Student Member IEEE and Kai Hwang, Fellow IEEE Computer Society, " Proactive Content Poisoning To Prevent Collusive Piracy in P2P File Sharing ",IEEE TRANSACTIONS ON COMPUTERS, TC -2007-09-0492R2, REVISED APRIL 8, 2008
[3] Hongli Zhang, Lin Ye, Jiantao Shi, Xiaojiang Du, and Hsiao-Hwa Chen, "Preventing Piracy Content Propagation in Peer-to-Peer Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS/SUPPLEMENT, VOL. 31, NO. 9, SEPTEMBER 2013.
[4] BitTorrent.org, "BitTorrent Protocol Specification," http:// www.bittorrent.org/protocol.html, 2006.
[5] Abhishek S. Naswale, Mrs. Vidya Waykule, Mandar, M. Mahadeokar, Rahul S. Gaikwad, Abhijit R. Jain, "Collusive Piracy Prevention in P2P Network", 2013.