Collusive Challenging Piracy Structure Anticipation in P2P Network

Sampada Sawant[1] Shilpa Sawant[2] Dhanashree Yelmar[3] Prof. Torana Kamble[4]
[1,2,3]Student [4]Assistant Professor
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]BVCOE, Kharghar, Navi Mumbai

Abstract— The idea behind collusive piracy is to detect pirates timely depends on timestamp token. The candid or legitimate clients are having that obey with the copyright law to share contents unreservedly. There are some pirates which has peers that are attempting to download the some content without paying or endorsement. We propose a scheme to stop colluders and pirates from copyright data in P2P file sharing. The colluders should not share copyrighted content files with pirates. Our aim is to stop collusive piracy within the frontier of P2P content liberation network. Our main focus is to stop colluders from releasing content files freely and disrupt pirate efforts from accumulating clean chunks. The content poisoning is implemented by discussing fraudulent of file requested by pirates. A copyright protected P2P network should beneficial to both the media industry and interact users.

Key words: P2P Network, Gnutella, DRM

I. INTRODUCTION

Large digital content files appear as CD-ROM images, popular software packages, TV episode albums, MPEG-4video films, etc. These popular digital contents are subject to gross copyright contravention over the Internet. In particular, piracy over peer-to-peer (P2P) networks has escalated in recent years hindering the increase of software industry, digital distraction, electronic publishing, web services, etc. Distributing large-scale files demands content delivery networks (CDN). These CDNs need to deploy a huge number of content servers through broadband and wide-area networks. The peer-to-peer (P2P) networks offer a new move toward to fast content delivery. P2P content networks have several separate advantages over the client-server CDN. A P2P content network reduces the allocation cost significantly, since many allocation servers are not needed. P2P networks improve the content availability, as any peer can serve as a content provider. The scalability of P2P network is especially attractive, because more peers effect in faster download speed.

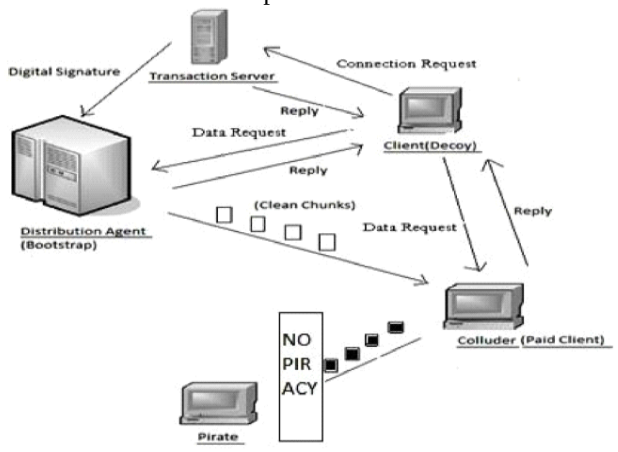

Fig. 1.1: System Overview

The network is built over a large number of peers. Four types are there of peers exist in the P2P network:

(1) Clients (honest or legitimate peers)
(2) Colluders (paid peers sharing contents with others without authorization)
(3) Distribution agents (trusted peers for file distribution)
(4) Pirates (unpaid clients downloading content files illegally)

In our project we are using peer authorization protocol (PAP) to distinguish pirates from legitimate clients. Detected pirates will receive poisoned chunks in repeated attempts. A common technique to decrease the availability of a specific item (e.g., movie, song, software distribution) in a peer-to-peer network consists in injecting a huge number of decoys into the network. The decoys are files whose name and metadata information (e.g., artist name, genre, length) equivalent those of the item, but whose actual content is incomprehensible, corrupted, or altogether different from what the user expects. For instance, many peer-to-peer users who tried to download the song "American Life" by Madonna found themselves in possession of a track that only contained a message from the artist chiding them via file sharing services, such a deliberate injection of decoys known as "item poisoning". Inter network piracy between undefended networks is a much more complex safety problem. Our main purpose is to stop colluders from releasing content files freely and interrupt pirate efforts from accumulating clean chunks. There are many other forms of online or offline piracy that are beyond the scope of this study.

For example, this protection scheme does not work on a private or enclosed network formed by pirate hosts exclusively. This Scheme did not solve the randomized piracy problems using email attachments, FTP download directly between colluders, or replicated CDs or DVDs. At present, these direct point-to-point copyright contravention problems are mostly handled by digital rights management (DRM) techniques; even the protection outcome are not considered acceptable, as many hackers have post DRM-cracks on the Internet.

II. SYSTEM ANALYSIS

A. *Existing Systems:*

The first generation of P2P systems focused on the topic of file-sharing, especially music files sharing. Napster was the first application providing such a service using P2P features. Napster used however a centralized directory where clients could export the names of the files that resided locally on their host.

The second generation of P2P systems wanted to eliminate the problematic centralized co-ordination from the first generation. Gnutella and Freenet received a lot of attention when Napster was shut down.

1) Gnutella:

It stores the files as in Napster at the users of the system. However, no centralized localization server is used. To find a file, a flooding approach is used: a search request is forwarded to neighboring users until it reaches a node where the file is available or exceeds a limited time-to-live. The system is then very robust. However,

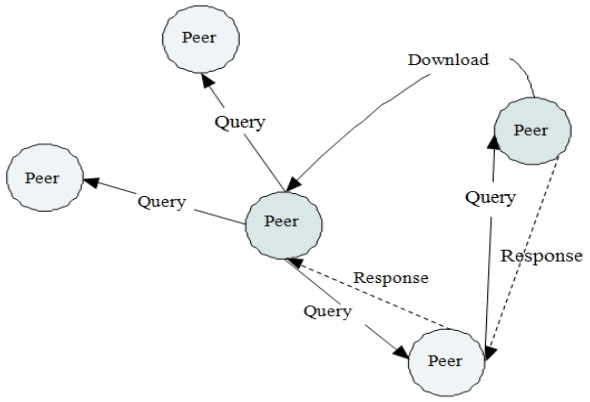

Fig. 2.1: Gnutella utilizes a decentralized architecture document location and retrieval

2) Freenet:

It uses an adaptive routing and a caching strategy: each node stores some index information and data. When a request arrives, either the data is in the cache, or the request is forwarded to another node according to a proximity definition until the data is found or a time-to-live is reached. When an answer arrives, the peer caches it's (the data and the index information) using specific caching strategies.

B. Problem Statement:

PEER-TO-PEER (P2P) networks are most cost-effective in delivering large files to massive number of users. Unfortunately, today's P2P networks are grossly abused by illegal distributions of music, games, video streams, and popular software.

These abuses have not only resulted in heavy financial loss in media and content industry, but also hindered the legal commercial use of P2P technology. The main sources of illegal file sharing are peers who ignore copyright laws and collude with pirates.

The basic idea is to detect pirates timely with signatures and time stamped tokens. The scheme stops collusive piracy without hurting legitimate P2P clients. We used peer authorization protocol (PAP) to distinguish pirates from legitimate clients. Detected pirates/colluders will lose their shared data so that no further sharing can be done.

C. Proposed System:

Proposed system is an application for prevention of collusive piracy in p2p content delivery networks for digital library system. It will be used for detection pirates and stop them from sharing patent material. It will also be helpful in detecting corrupted files while at the same time if any of the chunk gets poisoned then system automatically stops downloading and if user retries same file then it resumes from where it has stopped.

Various Existing systems observed during the literature survey mainly we found that there was no such authentication mechanism for validating or verifying a user. So that any user was free to share anything that he/she wants which ultimately violates copyright material laws. Our proposed system will provide a unique PAP protocol to authenticate users with digital signatures and time-stamped tokens. The existing systems basically focus on delivering files over a large network without any restriction.

III. LITERATURE SURVEY

In unstructured p2p systems such as Gnutella, a given file can be stored at any node in the system. The original version of Gnutella used scoped flooding to locate a file. While this method is highly robust and flexible, it is not scalable. To address the scalability problem, newer versions of Gnutella as well as other unstructured p2p systems such as KaZaA use a two-level hierarchy. The first level of the hierarchy consists of leaf nodes, and the second level consists of more powerful nodes, called super-nodes.

A. DRM:

Electronic publishing was hindered by the rapid growth of copyright violations. The DRM systems provide powerful restriction over use of digital content. Unfortunately, these systems do not function effectively when paid customers are colluding with the pirates. Two DRM techniques are popular in the copyright protection community: encryption versus watermarking with encryption, the digital content cannot playback unless user obtains the correct decryption key. Unfortunately, once a user gets the key and decrypts the file, he or she can share it with anyone. The idea of watermarking is to make each digital copy slightly different from others. If anyone shares his copy, content owner can detect the original point of leakage and take appropriate legal actions. The watermarking scheme must modify the original content. Therefore, watermarking was rarely applied in P2P networks.

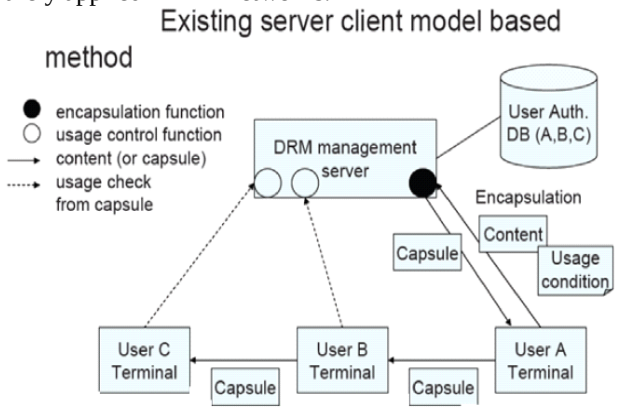

Fig. 3.1: DRM in existing client-server model

B. Peer to peer network:

P2P networks uses a decentralized model in which each machine, referred to as a peer, functions as a client with its own layer of server functionality. A peer plays the role of a client and a server at the same time. That is, the peer can initiate requests to other peers, and at the same time respond to incoming requests from other peers on the network. It differs from the traditional client-server model where a client can only send requests to a server and then wait for the server's response.

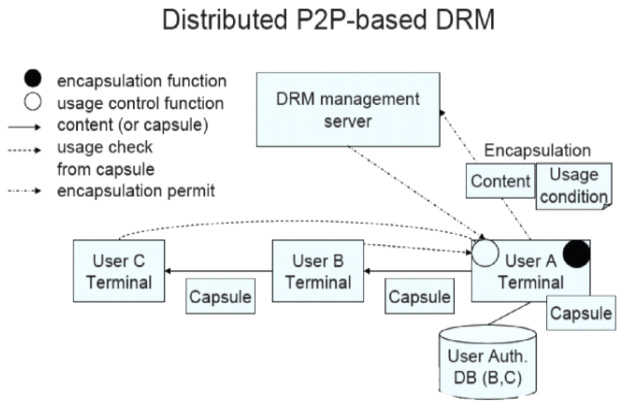

Fig. 3.2: DRM in P2P network architecture

IV. MODULAR DESCRIPTION

This section shows the different modules used in the whole process of content purchasing and distribution in a secured manner.

A. Authorization Module:

It provides user to join into our system and to download any requested file. During this content purchase request phase user mainly deals with Transaction Server. TS provide user a Digital Receipt for further module verification. All the payment related work is done by transaction server.

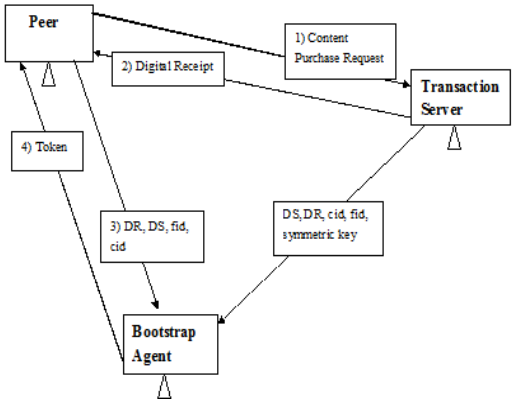

Fig 4.1: Peer authorization

B. Validation Module:

This is the core of the Anti-Piracy System which validates user for downloading a requested file. It applies a Peer Authorization Protocol to validate user. User provides its credentials received from Transaction Server then Bootstrap Agent verifies each of them. And finally a Token for confirmation of downloading is provided to user.

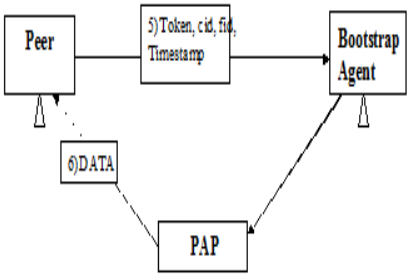

Fig. 4.2: Peer validation

V. SYSTEM IMPLEMENTATION

For the implementation of the system we have used JAVA for GUI designing as well as security based coding, socket programming, file I/O operations and JXTA for forming a flexible p2p network. JXTA is framework which provides protocols for peer-to-peer communication. It has been developed using Net Beans 7.0.1.

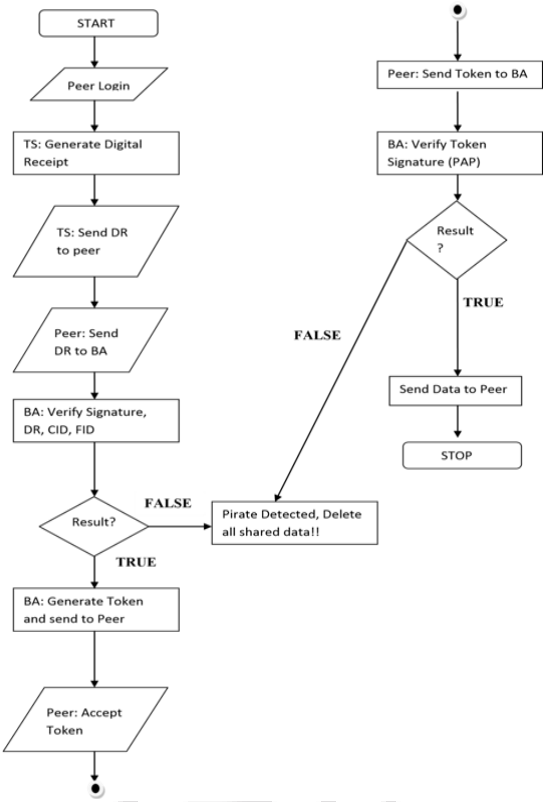

Fig. 5.1: System implementation

A. Peer Joining Process:

To start with implementation first step was to form a physical p2p network with 3-4 nodes connected to a hub/switch. After that we setup JXTA in application with default configuration and settings normally required for any p2p networks. With this a basic application a base was created. Seven messages are specified below to protect the peer joining process:

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

B. Secure File Indexing:

In P2P network indexing is used to share file in network which uses file ID to peer endpoint address. When peer request to download contents of file then request for download from selected peer to detect file contents. The file contents have components as an authorization token, a time stamp & peer signature. Each justifiable client has valid token which is assigned by its bootstrap. The

timestamp indicate time when token expires. This puts need to refresh token timely. These tokens are designed to protect from colluders. The peer signature is signed with key which is generated by PKG. The effects of these components that file indexes are secured.

C. File Level Token Generation:

When peer joins the P2P network the transaction server & PKG are fully trusted & their public keys are thrown to all peers. Then it first sends the authorization request to bootstrap agent. All the messages between peer & its bootstrap agent are encrypted using session key assigned by transaction server at purchase time. The authorization token is generated by token generation algorithm. A token is a digital signature is of three tuple :{ peer endpoint, file ID & timestamp} signed by content owner. The decrypt (receipt) function decrypts the receipt to identify the file. The requestor return with endpoint address. After receiving a private key, the bootstrap agent digitally signs the file ID, address & timestamp to create token.

The reply message contains {endpoint address, peer private key, timestamp & token} the reply message is encrypted using the assigned session key.

1) Algorithm 1: Token Generation:

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

2) Algorithm 2: Peer Authorization Protocol:

Input: T = token, ts = timestamp, S = peer signature, and ø(λ,p) = file index for file λ at endpoint p
Output: Peer authorization status
True: authorization granted
False: authorization denied
Procedures:
Step 01: Parse (input) = {T; ts;S; ø(λ,p)}
// Check all credentials from a input request
Step 02: p = Observe (requestor);
// detect peer endpoint address p //
Step 03: if {Match (S; p) fails},
//Fake endpoint address p detected //return false;
Step 04: endif
Step 05: if {Match (T; ts;K) fails},
return false;

VI. Conclusion and Future Enhancement

Traditional content delivery networks (CDNs) use a large number of content servers over many globally distributed WANs. The content distributors need to replicate or cache contents on many servers. The bandwidth demand and resources needed to maintain these CDNs are very expensive so we are proposing P2P content network to eliminate number of contents servers. Peer to peer network are most cost effective in delivering large files to massive number of users.

We can detect normal user i.e. legitimate client, colluder and pirates by using concept of mismatched keys, token, digital signature and implementing PAP protocol. Our application may be useful for sharing application, library system etc.

Piracy is one of the biggest problems for Software Companies and other Entertainment Sector, so in future this project can be enhanced to solve the problems related to such sector. Our project is just a step for preventing piracy in peer to peer network.

References

[1] "A Survey And Comparison Of Peer-To-Peer Overlay Network Schemes" Steven lim, Microsoft Asia
[2] "The Challenges Of Stopping Illegal Peer-To-Peer File Sharing" Kevin Bauer, Dirk Grunewald, and Douglas Sicker, Department of Computer Science, University of Colorado.
[3] "Content Availability, Pollution And Poisoning In File Sharing Peer To Peer Networks", Nicolas Christin S.I.M.S., UC Berkeley, Andreas S. Weigend Weigend Associates LLC, John Chuang S.I.M.S., UC Berkeley.
[4] "Copyright-Protected Content Delivery In Open Peer-To-Peer Networks", X. Lou and K. Hwang are with the Electrical Engineering Department, University of Southern California, Los Angeles, CA.90089, September, 2007.
[5] BitTorrent.org "Protocol Specification", http://www.bittorrent.org/beps/bep_0003.ht ml, 2013-09-01
[6] "Gnutella Protocol Specification", http://capnbry.net/gnutella/protocol.php, 2013
[7] N. Mook, "P2P Flooder Overpeer Cease Operation," Beta News, http://betanews.com/2005/12/10/p2p-flooder-overpeer-ceases-operation
[8] N. Mook, "P2P Future Darkens as eDonkey Closes,"http://betanews.com/2005/09/28/p2 p-future-darkens-as-edonkey-closes

