A Reputation System to Prevent Collusive Copyright Protection in P2P Network

Sashi Tarun
Assistant Professor, Arni School of CS & Application
Arni University
Kathgarh, Indora, Himachal Pradesh., India

Shabina Ghafir
Assistant Professor, Department of CSE
Jamia Hamdard University
Hamdard Nagar, New Delhi, India

Abstract— **P2P (peer-to-peer) applications have enormous success in information sharing, although there exist illegal distributions of copyright-protected works (music, games, video streams etc.) and anonymous malicious attacks. These abuses have resulted in heavy financial loss in media and content industry. Collusive piracy is the main source of intellectual property violations within the boundary of P2P networks. This problem resulted from paid clients (colluders) illegally sharing copyrighted content files with unpaid clients (pirates). Most P2P networks in use do not provide any function of Digital Rights Management (DRM) and are always blamed for illegally sharing copyrighted contents. A crucial feature of P2P network is that every peer could act as a content provider and the end users could obtain the same content from different content providers. We are using a peer authorization protocol (PAP) to distinguish pirates from legitimate clients. Detected pirates will receive poisoned chunks in repeated attempts. Combining DRM and reputation system to protect P2P content delivery networks will lead to a total solution of the online piracy problem. 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. DRM enabled P2P networks provide faster delivery speed, higher content availability, and cost-effectiveness than using conventional CDNs built with huge network of servers.**

Keywords- P2P; DRM; Collusive Piracy; Colluders; Piracy; CDN

I. Introduction

Recently, P2P (Peer to Peer) systems, direct file sharing systems among the peers, are one of the most attractive file sharing system. P2P architectures have high scalability and high performance due to the fact that its architectures which have characteristics of distributed file processing. However, P2P Architecture is infamous for distribution channel of illegal contents. So we must apply the DRM (Digital Rights Management) system to the P2P architecture, and we should keep advantages of P2P even if after DRM system applied. Here, we propose new type of DRM applied P2P system architecture that keeps existing P2P system's advantages.

A. *Peer-to Peer Architecture*

Often referred to simply as peer-to-peer, or abbreviated P2P, peer-to-peer architecture is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architectures where some computers are dedicated to serving the others. Peer-to-peer networks are generally simpler but they usually do not offer the same performance under heavy loads. The P2P network itself relies on computing power at the ends of a connection rather than from within the network itself.

In P2P networks, all clients provide resources, which may include bandwidth, storage space, and computing power. As nodes arrive and demand on the system increases, the total capacity of the system also increases. This is not true of client-server architecture with a fixed set of servers, in which adding more clients could mean slower data transfer for all users.

Once you have downloaded and installed a P2P client, if you are connected to the Internet you can launch the utility and you are then logged into a central indexing server. This central server indexes all users who are currently online connected to the server. This server does not host any files for downloading. The P2P client will contain an area where you can search for a specific file. The utility queries the index server to find other connected users with the file you are looking for. When a match is found the central server will tell you where to find the requested file. You can then choose a result from the search query and your utility when then attempt to establish a connection with the computer hosting the file you have requested. If a successful connection is made, you will begin downloading the file. Once the file download is complete the connection will be broken.

B. *Digital-Rights-Management*

The term digital rights management (DRM) broadly refers to a set of policies, techniques and tools that guide the proper use of digital content. A high level view of the flow of content from the creator to the consumer via the producer is shown in Fig.1.1. The content creator is mainly concerned with the core data/information that goes into the content. This could be viewed as raw content, which needs to be processed further with respect to adhering to certain formats, the suitable integration of different kinds of media, quality enhancement, additions of possible special effects, and derivation and addition of metadata (information about the data). The producer of the content performs the necessary processing and generates the packaged content. The packaged content is in a form that is suitable for consumption and for the tracking and management of content usage. The consumer is the ultimate user of the content. A DRM system plays important roles in several processes that are involved in the flow of content, as shown in Fig. 1.1. Very broadly, it facilitates the creator to specify the desired ownership rights of the content. It enables the producer to derive appropriate metadata from the content and specify the producer's rights. It allows the consumer to specify the desired content and the various options in the use of content. It also allows the producer to monitor the content usage and track payment information. There are several techniques to monitor appropriate content use and to prevent its illegal use. The choice of a particular technique depends on the content type, application needs, and tolerance to inappropriate use of content. It must be understood, however, that no content protection and monitoring technique guarantees absolute security and fool-proof operation. There is always the possibility that desirable features and functionalities of a DRM system will be circumvented. The design and operation of the DRM system should take these factors into account.

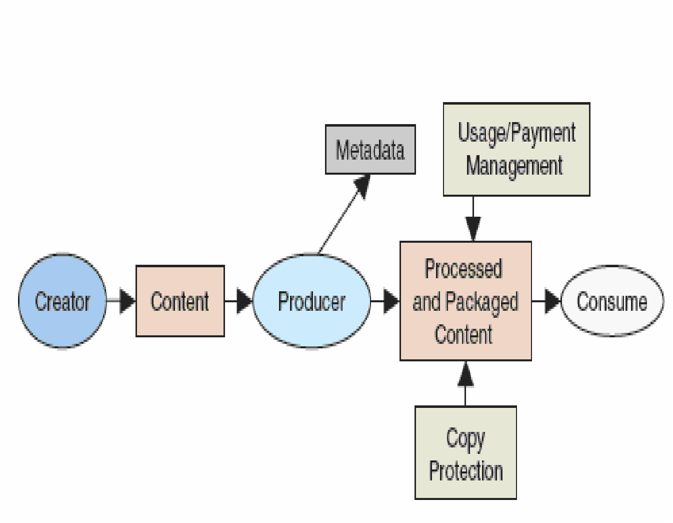

Fig.1.1. Broad Overview of Flow of Content from Creator to Consumer

II. System Analysis

A. Objectives

Our main focus is on detecting the pirates timely with identity-based signatures and time-stamped tokens. Our scheme should stops collusive piracy without hurting the legitimate P2P clients by targeting poisoning on detected pirates or violators, exclusively. We attempt to combine DRM technology into P2P network to construct a convenient copyright-secure content distribution system. We can focus on the following-

1. Use identity-based signatures to enhance Secure File Indexing for Piracy Detection.
2. Chunk Poisoning for Piracy Prevention.
3. Minimum delivery cost, higher content availability, and copyright compliance in exploring P2P network resources.
4. All massively distributed systems demand scalability, security protection, fault- tolerance, and hacker-proof operations, which are crucial to their acceptance in a digital society.
5. P2P platform for legalized content distribution.
6. Reputation system with combination of DRM to protect P2P content delivery networks will lead to a total solution of the online piracy problem.
7. Send falsified file content to frustrate unpaid peers (poisoned file chunks).

B. Existing System

The P2P network is accessed by a large number of paid clients, some colluders or pirates, and a few distribution agents. In this type of network, a colluder can download the content illegally without the knowledge of content owner by illegal sharing with some paid clients. Although P2P networks are used the most large contents distribution channel, it can also used the most large illegal contents distribution channel. Content providers in the music industry use different file formats to deliver their content to consumers. Most content providers are protecting their digital content. However, they use a variety of technologies and a combination of these technologies to control access to and usage of their digital content. Each protection technology has its specific goal. Most content providers seem satisfied with their current protection and approximately half of them want to enforce it in the future. There seem to be two groups of content providers. On the one hand, there are those which think that piracy does not have a huge impact on their company. These respondents use fewer protection technologies (i.e., two or three) but provide value-added services to consumers.

C. Problem Description

Peer collusion problem is main in the P2P network. Peers or paid clients illegally share the files or the keys which are used for downloading the contents from the main server. Peers ignore the copyright laws and collude with pirates. 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 content 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 copyrighted content files with unpaid clients (pirates). Such an on-line piracy has hindered the use of open P2P networks for commercial content delivery.

D. Proposed System

Our goal is to stop collusive piracy within the boundary of P2P content delivery network. We propose a proactive content poisoning scheme to stop colluders and pirates from alleged copyright infringements in P2P networks. The basic idea is to detect pirates timely with identity-based signatures and time stamped tokens. The scheme stops collusive piracy without hurting legitimate P2P clients by targeting poisoning on detected violators, exclusively. Detected pirates will receive poisoned chunks in their repeated attempts. Pirates are thus severely penalized with no chance to download successfully in tolerable time. The integration of selective poisoning with reputation system and DRM will widen the CDN application domains.

E. Literature Survey

A company like MediaDefender [1] uses P2P for disruption of movies and music downloads. MediaDefender is quite good at this, as it should be after five years of antipiracy work. Unlike DRM providers that focus on protecting the product, MediaDefender tries to protect the distribution channel. Lee describes four strategies that MediaDefender uses.

Decoying. This, in a nutshell, is the serving of fake files that are generally empty or contain a trailer. The goal is to make legitimate content a needle in a haystack, so MediaDefender works hard to ensure that its copies of files show up in the top ten spots when certain keywords are searched for. Everything about the file is tailored to look like the work of pirates, from the file size (movies are often compressed enough to fit on a CD) to the naming conventions to the pirate scene tag.

Spoofing. Spoofing sends searchers down dead ends. MediaDefender coders have written their own software that interacts with the various P2P protocols and sends bogus returns to search requests, usually directing people to nonexistent locations. Because most people only look at the top five search results, MediaDefender tries to frustrate their first attempts to download a file in hopes that they will just give up. The other two strategies are Interdiction and Swarming.

Peer-to-peer content distribution [2] architectures present a particular challenge for providing the levels of availability, privacy, confidentiality, integrity, and authenticity often required, due to their open and autonomous nature. The network nodes must be considered untrusted parties, and no assumptions can be made regarding their behavior.

Various cryptographic algorithms and protocols are employed to provide security for content published and stored in peer-to-peer networks. They performed this study of peer-to-peer content distribution systems and infrastructures by identifying the feature space of their nonfunctional properties, and determining the way in which these nonfunctional properties depend on, and are affected by various design features. The main nonfunctional characteristics they identified include provisions for security, anonymity, fairness, increased scalability, and performance, as well as resource management, and organization capabilities.

In the experiment [3], the peer arrival and departure to the system are assumed to be random. They believe this reflects (or at least it is close to) the behavior of real peers. Note that different from the Web accesses, in P2P systems, there is no consensus on the peer arrival and departure patterns. The first present the overall performance of proxy under different proxy cache sizes and peer cache sizes, then evaluate the performance improvement of our proposed proxy replacement and peer replacement policy.

Existing Internet streaming media delivering techniques are either based on a client-server model, such as, proxy caching and server replications by CDNs, or based on a client-based P2P structure delivery due to the dynamic nature of peers. In this study, we propose P2P assisted proxy systems to address these two limitations. In our system, the proxy is a member of the P2P network managed by the distributed hash table. In addition, the proxy also plays an important and unique role to ensure the quality of media delivery due to its dedicated and stable nature.

The disadvantage of the client-server model is its limited scalability and high cost. In a client-based P2P system is its unreliable quality of streaming media.

Content providers use [6] Preventive (encryption), Reactive (watermarking) Protection technologies. Apparently we saw the relationship between the numbers of protection technologies. Very young research field is to Analyses based on small number and regrettably no statistical analysis and tests are possible

- Only descriptive conclusions
- Presentation outlined seven core protection technologies and showed they are also the most used in practice
- There are some industry differences.

PeerCQ [7], is a decentralized architecture for Internet scale information monitoring using a network of heterogeneous peer nodes. The main contribution of the paper is the smart service partitioning scheme at the PeerCQ protocol layer, with the objective of achieving good load balance and good system utilization. This scheme has three unique properties:

First, it introduces a donation-based peer-awareness mechanism for handling the peer heterogeneity. Second, it introduces the CQ-awareness mechanism for optimizing hot spot CQs. Third, it integrates CQ awareness and peer-awareness through two phase matching algorithms into the load balancing scheme while maintaining decentralization and self-configurability.

Several factors can affect the load balancing decision, including the computing capacity and the desired resource contribution of the peers, the willingness of peers to participate, and the characteristics of the continual queries.

Content poisoning is often treated as a security threat to P2P networks [4] A few thousand Web sites [9] receive a significant fraction of request traffic. So we are going for Content distribution networks (CDNs) are a mechanism to deliver content to end users on behalf of origin Web sites to reduce traffic for end-user.

This paper provides a timely discussion and analysis of CDNs. We have used multiple data streams: active measurements obtained via repeated crawls over a period of time and passive measurements representing large number of users from different organizations. We have also analyzed content types commensurate with traffic patterns on the Web. The primary performance study has been repeated more than once.

The types of client sites are relatively narrow in scope. The specific results of an empirical study such as this one may change over time

III. MODULAR DESCRIPTION

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

A. Network Build Model

The content owner server is responsible for purchasing and billing of digital contents. The distribution agents provide peer authentication, distribute digital content to paid customers, and prevent unpaid peers from downloading the files.

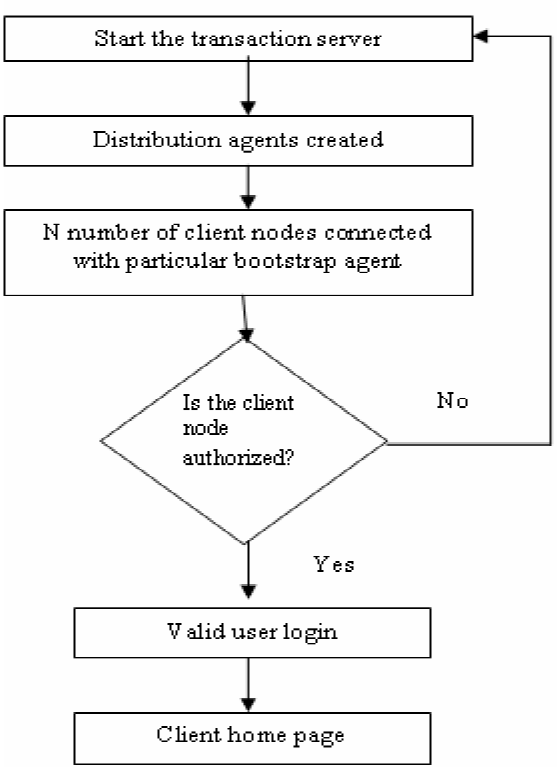
Start the transaction server
Distribution agents created
N number of client nodes connected with particular bootstrap agent
Is the client node authorized?
No
Yes
Valid user login
Client home page

Fig.3.1 —Network Build Module

B. Authentication Module

We are using a protocol that identifies a peer with its endpoint address. File index format is changed to incorporate a digital signature based on this identity. A peer authentication protocol is used to establish the legitimacy of a peer when it downloads the file. Using Identity-Based Signature (IBS), our system enables each peer to identify unauthorized peers or pirates without the need for communication with a central authority.

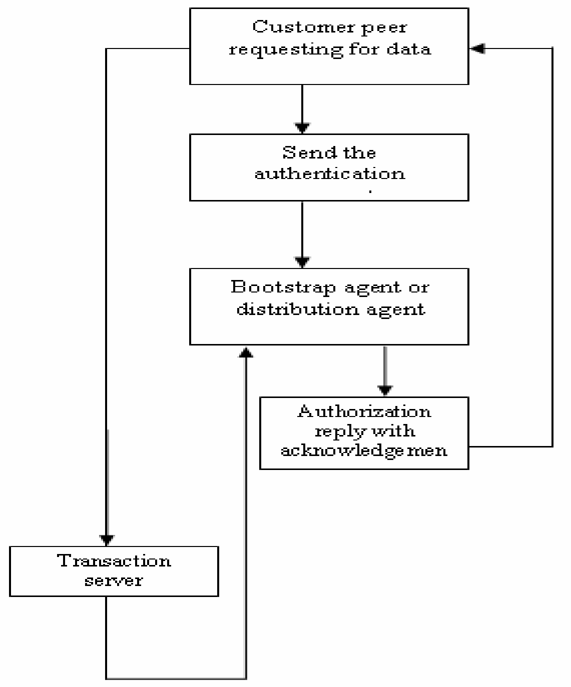
Customer peer requesting for data
Send the authentication
Bootstrap agent or distribution agent
Authorization reply with acknowledgemen
Transaction server

Fig.3.2 —Authentication Module

C. Private Key Generation Module

A Private Key Generator (PKG) is installed to generate private keys with IBS for securing communication among the peers. 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 copyright protection are completely distributed.

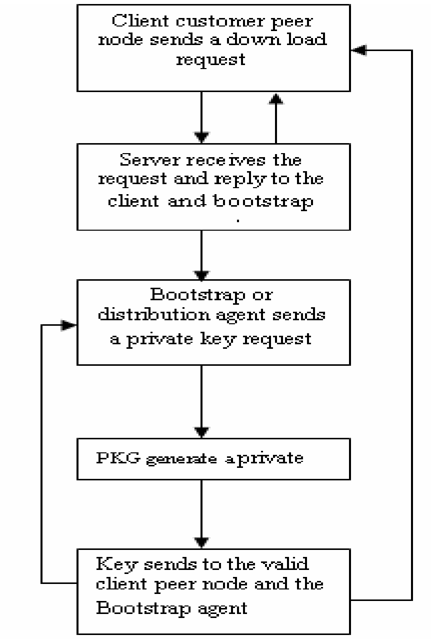
Client customer peer node sends a down load request
Server receives the request and reply to the client and bootstrap
Bootstrap or distribution agent sends a private key request
PKG generate aprivate
Key sends to the valid client peer node and the Bootstrap agent

Fig. 3.3 — Private Key Generator Module

D. Authorization Module

In a P2P file-sharing network, a file index is used to map a file ID to a peer endpoint address. When a peer requests to download a file, it first queries the indexes that match a given file ID. Then the requester downloads from selected peers pointed by the indexes. 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.

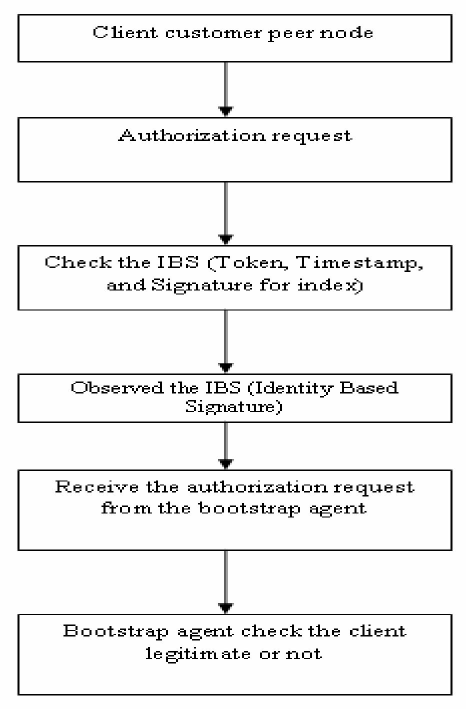
Client customer peer node
Authorization request
Check the IBS (Token, Timestamp, and Signature for index)
Observed the IBS (Identity Based Signature)
Receive the authorization request from the bootstrap agent
Bootstrap agent check the client legitimate or not

Fig. 3.4 —Authorization Module

E. Content Distribution Module

In this module have a secured P2P platform for copyright-protected content delivery. Here we include the DRM concept during the content delivery. With our DRM approach the delivered content is in encrypted form. The key for decryption of that encrypted content is supplied during the run time of delivery process. In this process whenever the pirates request for the files from the distributing agent, the file will be provided to the pirate but the information's will be redundant information.

IV. System Implementation

A. Peer Joining Process

For a peer to join the network, it first logs in to a transaction server to purchase the content After transaction, the client receives a digital receipt containing the content title, client ID, etc. This receipt is encrypted such that only content owner and distribution agent can decrypt. 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 set up by the content owner, it decrypts the receipt and authenticates its identity. The bootstrap agent requests a private key from PKG and constructs an authorization token, accordingly. Let k be the private key of content owner and id be the identity of the content 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. The client is identified by user ID and the file by file ID. 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 do not share the content 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 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.

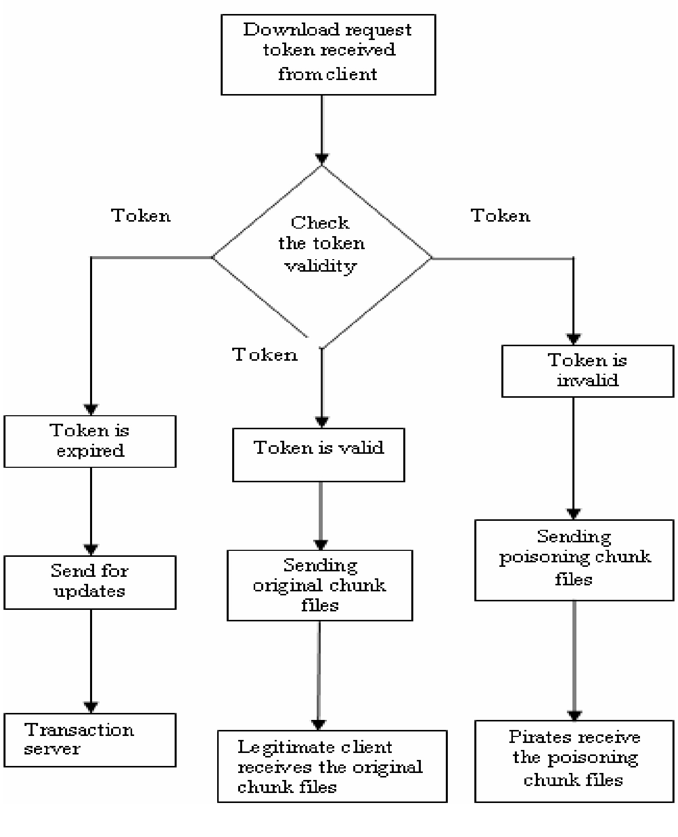
Download request token received from client
Check the token validity
Token
Token
Token
Token is expired
Token is valid
Token is invalid
Send for updates
Sending original chunk files
Sending poisoning chunk files
Transaction server
Legitimate client receives the original chunk files
Pirates receive the poisoning chunk files

Fig. 3.5 —Content Distribution Module

B. Pirate identification

In a P2P content distribution network, only the content owner can verify the user ID/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 are using 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.

1) Secure File Indexing

In a P2P network, a file index is used to map a file ID to a peer endpoint address. When a peer requests to download a file, it first queries the indexes that match a given file ID. Then the requester downloads from selected peers pointed by the indexes. 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 a token expires. Thus, the peer needs to refresh the token periodically. This short-lived token is designed for protecting copyright against colluders. The cost at each distribution agent to refresh the client tokens is rather limited, as shown via experiments. 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 indexes. The combined effects of the three extra fields ensure that all references to the file indexes are secured. Peers identify the pirates by checking the validity of the token and the signature in a file index. These features secure the P2P network operations to safeguard the sharing of clean contents among the paid clients.

2) 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 of 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 specified below. A token is a digital signature of a three-tuple: {peer endpoint, file ID, timestamp} signed by the private key of the content owner. Since bootstrap agent has a copy of the digital receipt sent by transaction server, verifying the receipt is thus done locally. The Decript (Receipt) function decrypts the digital receipt to identify the file L. The Observe (requestor) returns with the endpoint address p. The Owner Sign (λ; p; ts) function returns with a token.

Upon receiving a private key, the bootstrap agent digitally signs the file ID, endpoint address, and timestamp to create the token. The reply message contains a four-tuple: {endpoint address, peer private key, timestamp, token}. The reply message from bootstrap agent is encrypted using the assigned session key.

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:  $\lambda$ = Decrypt(Receipt);
// $\lambda$ 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 key
Step 10: end if
```

C. Proactive Poisoning

The PAP protocol is formally specified below. 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, the client must send poisoned chunks. In PAP, a download request applies a token T, file index ø, timestamp ts, and the peer signature S. If any of the fields are missing, the download is stopped. A download client must have a valid token T and signature S. Two pieces of critical information are needed: public key K of PKG and the peer endpoint address p.

Algorithm 2 verifies both token T and signature S. File index ø(λ,p) contains the peer endpoint address p and the file ID 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 ø from a 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, 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($\lambda$,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;
```

```
// Invalid or expired token detected //
Step 06: endif
Step 07: return true;
```

D. Content Delivery

The content and rights could be combined together into a DRM message. The delivery of the content from the content server can be one of two modes: download or streaming. In the download mode, the content is obtained by the device either along with the rights object or separately from it. It is stored locally and then rendered in accordance with its associated rights object. In the streaming mode, there is no storage of the content at the device. The content stream is appropriately protected using stream encryption mechanisms before delivery. The streams are decoded and then rendered by the devices. The device could have a DRM agent, which is responsible for enforcing the rights and controlling the content consumption in accordance with the rights. Super distribution refers to the transmission or forwarding of content from one device to another rather than from a content server to a device. However, the rights object cannot be transferred across devices. Thus, superdistribution minimizes the traffic from the server to several devices, while the rights management ensures that the superdistributed contents are not misused.

Many of the DRM schemes allow the content to be unencrypted and to be freely distributed. They ensure the legitimate and proper use of content by making the consumption of content only in conjunction with appropriate rights objects. There are several other schemes that use added measures of security to protect the content against unauthorized access and use. A simple protection technique is to use encryption of the content. Encryption uses an algorithm and a key to scramble the information. The key for decryption to recover the original information is provided to legitimate consumers. The complexity of the encryption algorithm and the key size are suitably selected based on the requirements of the particular application. Digital signatures are used for the authentication of content providers as well as content consumers. For example, the content header and a hash of the content, which is a fixedlength data obtained by applying a hash function, could be signed using the private key of the content owner/producer to generate a digital signature. The signature can be verified when issuing the license or when a device contacts the license server to obtain the license to play the content that it already has or to renew the license. For verification, the public key of the content owner/producer is used. Digital certificates are used for DRM client devices to ensure their validity.

Conclusion And Future Enhancement

Here we conclude the proactive content poisoning scheme to stop colluders and pirates from suspected copyright infringements in P2P file sharing network. And also when a pirate is detected, the distributed agent sends the falsified chunk file to the particular pirate client with proper counteractive actions. Combining DRM and reputation system to protect P2P content delivery networks will lead to a total solution of the online piracy problem. There are many other forms of online or offline piracy that are beyond the scope of this study. For example, our protection scheme does not work on a private or enclosed network formed by pirate hosts exclusively. We did not solve the randomized piracy problems using email attachments, FTP download directly between colluders, or replicated CDs or DVDs.

In future we can focus on prototyping and benchmark experiments which are needed in Real-Life Open P2P Networks Here we can only prove the protection concept, lacking of sustained accuracy. Proactive chunk poisoning can be made selectively to reduce the processing overhead. However, further studies are needed to upgrade the performance of the copyright-protected system in real-life P2P benchmark applications.

References

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

[2] S. Androutsellis-Theotokis and D. Spinellis (2004), “A Survey of Peer-to-Peer Content Distribution Technologies,” ACM Computing Surveys, vol. 36, pp. 335-371.

[3] S. Chen and X.D. Zhang(May 2006), “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.

[4] N. Christin, A.S. Weigend, and J. Chuang(2005), “Content Availability, Pollution and Poisoning in File-Sharing P2P Networks,” Proc. ACM Conf. e-Commerce, pp. 68-77.

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

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

[7] B. Gedik and L. Liu (June 2005), “A Scalable P2P Architecture for Distributed Information Monitoring Applications,” IEEE Trans. Computers, vol. 56, no. 6, pp. 767-782.

[8] [T. Kalker, D.H.J. Epema, P.H. Hartel, R.L. Lagendijk (June 2004), and M. Van Steen, “Music2share—Copyright-Compliant Music Sharing in P2P Systems,” Proc. IEEE, vol. 92, no. 6, pp. 961-970.

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

AUTHOR PROFILE

Sashi Tarun, presently working as Assistant Professor in the department of ASCSA at Arni University, Kathgarh, Indora, Himachal Pradesh. He completed his Master of Technology in computer Science from Jamia Hamdard University, New Delhi. He has 7 Yrs of teaching experience. His area of interest are network security, Distributed Computing, Software Engineering, Database System. He has published multiple papers in National Conferences and in International Journals.

Shabina Ghafir, Presently working as Assistant Professor in the department of CSE at Jamia Hamdard University, New Delhi. He has published many papers in National, International Conferences and in Journals