IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 58-63
www.iosrjournals.org

Anti-Piracy System to Prevent Collusive Copyright Protection in P2P Network

Sushant Borse, Nikhil Jamdade, Deryle D'souza
Department of Computer Engineering, Fr. C.R.I.T, Vashi, Navi Mumbai, India.
Email Id :{ sushantborse9, nikhil.jamdade, Deryledsouza}@gmail.com

ABSTRACT:
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. To protect such contents from being distributed to unauthorized peers, Lou and Hwang proposed a proactive content poisoning scheme. In this paper, we are revising the same scheme with more clear and limited approach. The main sources of illegal file sharing are peers who ignore copyright laws and collude with pirates. Our goal is to stop collusive piracy within the boundary of a P2P content delivery network. This scheme tries to stop collusive piracy without hurting legitimate P2P clients by targeting poisoning on detected violators. Peer authorization protocol (PAP) is used to distinguish pirates from legitimate clients. Detected pirates will receive poisoned chunks in their repeated attempts. Pirates are thus severely penalized with no chance to download successfully in tolerable time.

Keywords: *content poisoning, copyright protection, network security, p2p networks*

1. Introduction

Copyright protection is becoming more elusive as computer networks such as the global Internet are increasingly used to deliver electronic documents. Document distribution by network offers the promise of reaching vast numbers of recipients. However, these same distribution networks represent an enormous business threat to information providers [1] the unauthorized redistribution of copyrighted materials. Peer-to-peer abbreviated as P2P, peer-to-peer architecture is a type of network in which each workstation has equivalent capabilities and responsibilities. This scheme involves 3 types of user:

- *Honest or legitimate clients* are those that comply with the copyright law not to share contents freely.
- *Pirates* are peers attempting to download some content files without paying or authorization.
- *The colluders* are those paid clients who share the contents with pirates. Pirates and colluders coexist with the law-abiding clients.

This scheme uses a peer authorization protocol (PAP) to distinguish pirates from legitimate clients. 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 massive number of decoys into the network. The decoys are files whose name and metadata information (e.g., artist name, genre, length) match those of the item, but whose actual content is unreadable, 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 [2] found themselves in possession of a track that only contained a message from the artist chiding them for using file sharing services, such a deliberate injection of decoys known as *item poisoning* [2].This scheme mainly stops colluders from releasing content files freely and disrupt 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 violation problems are mostly handled by digital rights management (DRM) [3] techniques; even the protection results are not considered satisfactory, as many hackers have post DRM-cracks on the Internet.

2. Literature Survey

2.1 *P2P NETWORK*

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. The major source of illegal P2P content distribution lies in peer collusion [4] to share copyrighted content with other peers or pirates.

2.2 *DRM*

Digital rights management (DRM) systems are often suggested to provide copyright protection in P2P networks, such as proposals in [3]. Unfortunately, DRM 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 [5] 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.

2.3 *IBS*

In most basic form of digital signature [6], each user in the system generates his/her own key pair. In real world users are generally not identified by randomly generated keys but by names or email addresses. To map public keys to real-world identities, a so called public-key infrastructure needs to be set up. In IBS [7] corresponding secret key is issued by a trusted key generation centre (PKG) [8], which derives it from a master secret that only the KGC knows. Identity-based systems allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the private key generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as *master key*). Given the master public key, any party can compute a public key corresponding to the identity *ID* by combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity *ID* contacts the PKG, which uses the master private key to generate the private key for identity *ID*.

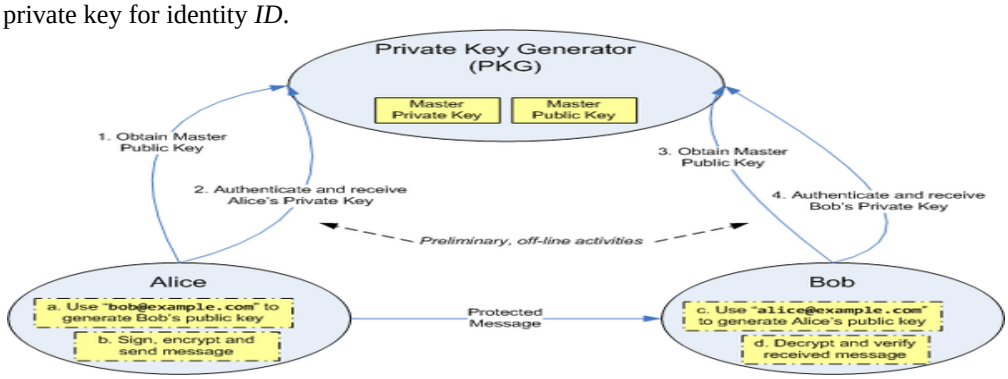

Fig. 1 a) PKG using Identity Based Signature (IBS)

The eDonkey overlay was ordered to shut down in 2006 [9] due to incontrollable piracy activities. The popular BitTorrent network is still facing many lawsuits against their content distribution operations.

3. Trusted P2P Network

As shown in Fig.1b), the trusted P2P employs a three-layer design centred on the content owners or distributors. In the owner layer reside a transaction server and a *private key generator* (PKG). The *transaction server* is responsible for purchasing and billing of digital contents. Using IBS [8], the PKG and private keys it generates are fully trusted just like using CA in PKI services. However, the PKG is a lot cheaper than a CA, because PKG does generate and publish public keys. The second layer consists of a number of distribution agents operated by content owners. The primary functions of an agent are to provide peer authorization, distribute digital content to paid clients, and prevent unpaid peers from downloading the same content. All paid clients and unpaid peers form the third layer. In P2P networks, a peer can self-assert its username without any verification. Therefore, it is not identified by its user name,

but by its public endpoint address (IP address + port number), because endpoint address is used for establishing the connections between peers.

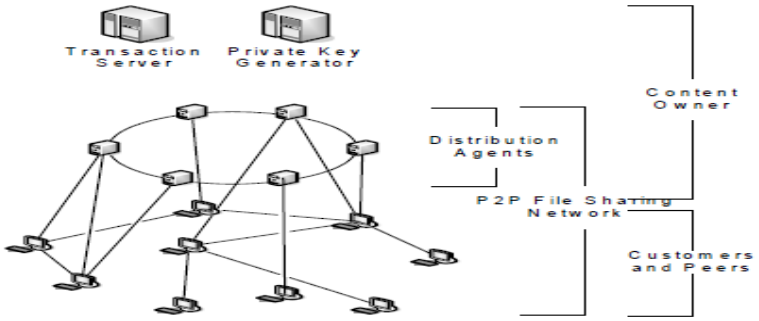

Fig. 1b) Protected P2P content delivery network with three layer architecture

Fig. 2 depicts an example: A peer has an IP address192.168.0.2 leased from its local router. It is listening to port 5678 forwarded by the router. When communicating with the bootstrap agent, the peer announces its listening port number. The detail of Observe () function is as follows: when a peer sends message to its bootstrap agent through outgoing port, agent attaches a random number (nonce) in the reply. The agent then sends a message to the advertised listening port 68.59.33.62:5678, asking the peer to send back the nonce. If the peer replies correctly, then its endpoint is verified. A malicious peer may spoof the peer IP address. Using the endpoint address by a bootstrap agent solves this spoofing problem.

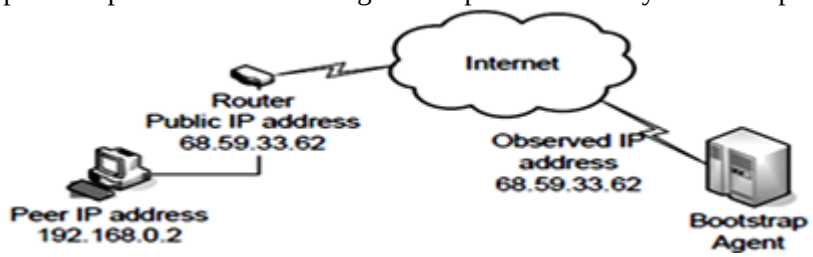

Fig. 2 Bootstrap agent observes peers end-point address

3.1 Protection in Peer Joining Process

The procedures for a peer to join the network and start downloading are specified in Fig.3 (a). First, a client logins to a transaction server to purchase the file. After transaction, the client receives a digital receipt containing the content title, client ID etc., session key and bootstrap agents address. This receipt is encrypted, only content owner and distribution agent can decrypt. The joining client authenticates with the bootstrap agent using a digital receipt. Since the bootstrap agent is setup by the content owner, it decrypts the receipt and verifies its authentication. The transaction server shares the digital receipt with bootstrap agent. The client and bootstrap agent secure their communication using a session key issued by transaction server. The bootstrap agent requests a private key from PKG and constructs an authorization token, accordingly.

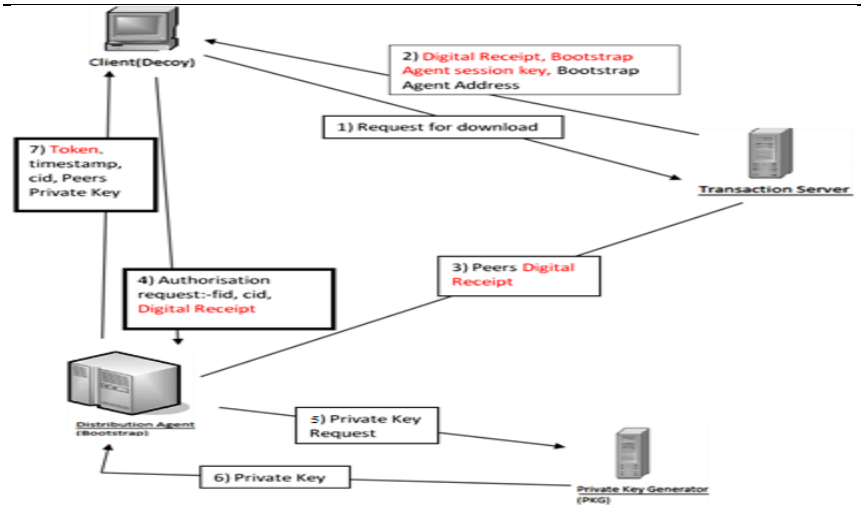

Fig. 3a) Protected peer joining process

Let *k* be the private key of content owner and *id* is the identity of the content owner. *Ek (msg)* denotes the encryption of message with key *k*. The client is identified by *userId* and the file by *fileId*. Seven messages in PAP are specified below.

Msg1: Content purchase request
Msg2: Bootstrap Agent Address,
Ek (digital_receipt, Bootstrap_Agent_session_key)
Msg3: Adding digital signature *Ek* (digital_receipt)
Msg4: Authentication request with
userID, *fileID*, *Ek* (digital_receipt)
Msg5: Private Key request with private Key Request
Msg6: PKG replies with Private Key
Msg7: Assign the authentication token to the client.

3.2 Security Layer

Following are the ways to tighten the security of this system.

3.2.1 Secure File Indexing

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, this scheme proposes to modify file index to include three components: *an authorization token, a timestamp, and a peer signature*. The peer signature is signed [10] with the private key generated by PKG. This signature proves the authenticity of a peer. Peers identify the pirates by checking the validity of the token and the signature in a file index.

3.2.2 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 of two integral parts: *token generation and authorization verification*. For any peer to download any content must have a token. After verification of the requesting peer agent generates token as shown in algorithm 1. A token is a digital signature of a three tuple: {peer endpoint, file ID, timestamp} signed by the private key of the content owner. The Reply message contains a four tuple :{ end-point address, peer private key, timestamp, token}.

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: λ = Decrypt (Receipt);
//λ is file identifier decrypted from receipt
05: p = Observe (requestor);
// p is endpoint address as peer identity
06: k = Private Key Request (p);
// Request a private key for user at p
07: Token T = Owner Sign (f, p, ts)
// Sign the token T to access file f
08: Reply = {k, p, ts, T}
// Reply with key, endpoint address, timestamp, and the token
09: Send to Requestor {Encrypt (Reply)}
//encrypt reply with the session key
10: **end if**

3.2.3 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, the client must send poisoned chunks. In PAP, first a download request consists of a token T, file index ø, timestamp ts, and the peer signature S. File index ø (λ, p) contains the peer endpoint address p and the file ID λ. Firstly any client will check the no. of inputs, if any of the fields are missing, the downloading is stopped. Then again nonce concept retrieves endpoint address. After that a downloading client must have a valid token T which is checked with the help of public key of PKG (**K**) and signature S which is checked with public key of peer (**p**). Two pieces of critical information are used: public key K of PKG and the peer endpoint address p.

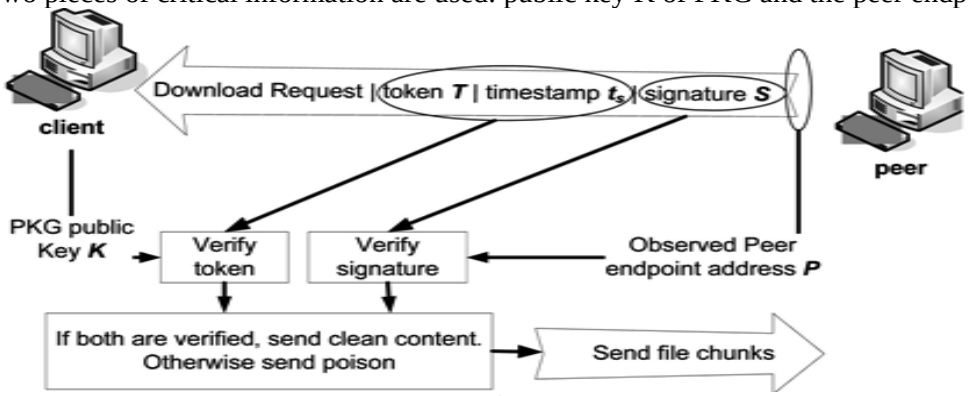

Fig.3b) PAP enables instant detection of a pirate upon submitting illegal download request

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:
01: Parse (input) = {T, ts, S, ø (λ, 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;
```

4. CONCLUSION

In practice, we expect that the majority of the peers are good citizens. 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 think that this scheme for P2P content network can eliminate number of content servers. This scheme can detect normal user's i.e. legitimate client, colluder and pirates by using concept of private key, digital signature and implementing PAP protocol. Our approach over existing systems:

- The protocol identifies a peer with its endpoint address used as a public key of peer.
- File index format is changed to include a token and an IBS signature.
- Using PAP and IBS enables each peer to identify unauthorised peers or pirates without need for communication with a central authority.
- In case pirates manage to hack or turn legitimate peers into unwilling colluders we use a random detection scheme.

Application on this scheme may be useful for music industry, library system etc. Piracy is one of the biggest problems for Software Companies and other Entertainment Sector, so in future project this scheme can be enhanced to solve the problems related to such sector. The scheme cuts off all pirated content distribution under low or moderate collusion rate. Only when the majority of paid customers decide to collude with pirates, the system may gradually lose control. Combination of DRM system, this scheme and randomized colluder detection can greatly reduce the chances of piracy.

REFERENCES

[1] Kevin Bauer, Dirk Grunewald and Douglas Sicker, *The Challenges of Stopping Illegal Peer-To-Peer File Sharing*, Department of Computer Science, University of Colorado.

[2] Nicolas Christin, UC Berkeley, Andreas S. Weigend, *Content Availability, Pollution And Poisoning In File Sharing Peer To Peer Networks*, Weigend Associates LLC, John Chuang, UC Berkeley, 2005.

[3] Antonio Liotta, Rossana Motta, Ling Lin, *A File Protection Method For Peer-To-Peer Systems*, Department of Electronic System Engineering, University of Essex, Colchester, CO4 3SQ, UK, Florida State University, Tallahassee, FL 32306, USA.

[4] Xiaosong Lou and Kai Hwang, *Adaptive Content Poisoning to Prevent Illegal File Distribution in P2P Networks*, University of Southern California, 2006.

[5] S.H. Kwok, Watermark-Based Copyright Protection System Security, Comm. ACM, pp. 98-101, Oct. 2003.

[6] "Digital Signature", http://www.tatanka.com/bionic_buffalo/original/archive/document/technote/tn0035.html

[7] "ID Based Encryption", http://en.wikipedia.org/wiki/ID-based_encryption

[8] D. Boneh and M. Franklin, Identity-Based Encryption From The Weil Pairing, Proc. Advances in Cryptology (Crypto '01).

[9] N. Mook, P2P Future Darkens as eDonkey Closes, http://betanews.com/2005/09/28/p2p-future-darkens-as-edonkey-closes

[10] Signing the Digest, http://www.youdzone.com/signature.html

[11] Xiaosong Lou, Kai Hwang,Collusive Piracy Prevention in P2P Content Delivery Networks,Published by the IEEE Computer Society, 2009.