Secure and Anonymous Multimedia Content Distribution in Peer-to-Peer Networks

Amna Qureshi, Helena Rifà Pous, David Megías
Estudis d'Informàtica, Multimèdia i Telecomunicació, Internet Interdisciplinary Institute (IN3),
Universitat Oberta de Catalunya (UOC),
Rambla del Poblenou, 156, 08018, Barcelona, Catalonia, Spain
{aqureshi, hrfia, dmegias}@uoc.edu

Abstract—**In recent years, the distribution of large scale multimedia contents has become easier and more efficient than ever before. The unauthorized distribution of copyright-protected content has emerged as a major concern. A buyer of a multimedia content does not want to reveal his/her identity whereas the seller or owner of the content does not want the buyer to further distribute the content illegally. Accordingly, this paper presents a secure and privacy-aware multimedia content distribution mechanism based on Tardos Codes for Peer-to-Peer networks. Our proposed system offers copyright protection by means of a traitor-tracing capability, collusion-resistance, content owner and buyer's security with mutual anonymity during secure data exchange.**

Keywords-privacy; security; anonymity; trust; collusion-resistant fingerprinting;Peer-to-Peer networks

I. INTRODUCTION

Peer-to-Peer (P2P) is often described as a type of decentralized computing system in which nodes, referred to as peers, use the Internet to communicate with each other directly. All the peers in this interconnected network provide resources to other peers, including bandwidth, storage space, and computing power. It greatly helps to establish multimedia communication networks and enables efficient distribution of digital content at a large scale. However, in the absence of security mechanism, P2P networks lack the ability to maintain content protection and access control towards copyright material, hence leading to piracy commitment and copyright infringement. The enforcement of copyright protection mechanisms in P2P content distribution systems, however, poses serious privacy threats to end users, i.e., monitoring of their activities by keeping record of downloaded files, IP address through which files are being downloaded, history of files shared or downloaded, or a list of the peers with whom a user has interacted in the past. Thus, there is an inherent conflict of interest between copyright protection supporters and privacy advocates. P2P developers need to balance security and privacy needs when designing content distribution systems.

However, security and privacy in P2P content distribution systems are achieved by various cryptographic mechanisms, (Homormorphic encryption, multi-party computation, etc) and digital watermarking/fingerprinting technology. Most of the research work related to copyright protection and privacy in content distribution system incurs high computational and communicational burden at the buyer's and/or at the multimedia owner's/merchant's end. Literature review shows that a few P2P content distribution systems [1-2] exist that provides both secure and anonymous content distribution with reduced computational burden at merchant's end. The review of some recent P2P content distribution systems satisfying either security or privacy properties can be found in [3]. Hence, it is of utmost importance to build an integrated platform which can enable content owners' to distribute their large-sized digital contents without a fear of copyright violation at reduced delivery costs and simultaneously enable end users to receive legal content without fear of privacy breach.

In this paper, we propose a secure and anonymous content distribution system for P2P networks that provides both copyright protection and privacy. Also, the proposed system reduces the computational cost of a merchant.

A. Contributions

A newly proposed privacy-aware content distribution scheme for P2P networks which would benefit multimedia owners (merchants) to share their large-sized copyrighted files and end users/peers (buyers) to download their content legally. In the proposed system, the original multimedia file is partitioned by the merchant into a base and supplementary file. The base file is much smaller than the original file and contains the most important information. Without this information, the file reconstructed by the end user would be unusable. The base file is dispensed by the merchant on payment from the user and a supplementary file is distributed to the P2P network. Thus, it reduces the burden of the merchants by only sending the small-sized base file and making use of the P2P network infrastructure to support most of the file transfer process. In this paper, we adopt a collusion-resistant Tardos fingerprint code [4] to confirm the distribution scheme's security against collusion attacks. A Trusted Third Party (TTP) Monitor is used, which is assumed to be an honest entity and is trusted by

both the merchant and the buyer. Since Tardos codes used by a merchant for the generation of collusion-resistant fingerprint depend on a secret vector which cannot be revealed to the buyer, this secret vector is committed to a Monitor. Moreover, to provide security to a buyer from a malicious merchant who might frame an honest user for illegal-redistribution, this secret vector is committed to a Monitor along with the pseudo identities of a merchant and a buyer. The proposed scheme can protect the buyer's privacy until the illegal re-distribution is proven. The illegal re-distributor is identified by the merchant using Tardos Code's piracy tracing algorithm. Also, the scheme provides resistance to man-in-middle attacks by employing mutual anonymity and key agreement and pseudonym-based trust management.

B. Outline of the paper

In this paper, we focus on a new solution to provide security, anonymity and privacy to both merchant and end-users of P2P content distribution system. In Section II, we highlight the design requirements for our system. The building blocks of our system are introduced in Section III. In Section IV, we describe our proposed system and, in Section V, we draw a conclusion and discuss future work.

II. DESIGN REQUIREMENTS AND ASSUMPTIONS

This section defines the system requirements and certain assumptions made during the design process of the system.

A. Design Requirements

For secure and privacy-aware P2P content distribution scheme, we have the following requirements depending on security, privacy, anonymity, trust, robustness and imperceptibility constraints.

- The identity of a buyer should remain anonymous during transactions until he/she is proven to be guilty of copyright violation.
- The merchant should be able to trace and identify the illegal re-distributor in case of finding a pirated copy.
- The scheme should be collusion resistant against a specific number of colluders' c as specified by Tardos codes.
- The buyer accused of re-distributing an unauthorized copy should not be able to claim that the copy was created by the merchant.
- The judge with the help of a Monitor should be able to resolve the disputes without the buyer revealing his/her identity.
- The fingerprinting process should be blind and the inserted fingerprint should be imperceptible and robust against common signal processing attacks.
- None of the other peers in the system should know about the source provider peer and requesting peer's identity or an item being exchanged.
- The real identity of a user should be protected during authentication process thus, enabling each user to verify the authenticity of each other anonymously.

B. Assumptions

The assumptions for the proposed system are as follows:

- There are four major players involved: merchant, buyer, Monitor and Judge.
- The merchant and the buyer do not trust each other but they both trust the Monitor.
- Each entity is supposed to have a public key K_p and a private key K_s.
- *SP* is selected on the basis of its reputation and resources and is thus assumed to be honest.
- Each user can have only one pseudonym against its secret information.
- The base file transfer between the merchant and the end user is secure and efficient by using the Anonymous two-party Authenticated Key Exchange (AKE) protocol.

- A P2P system maintains end-to-end data integrity. The reconstruction of the original file from the base and supplementary files is performed at the user end without assistance of a user. The supplementary file is only shared within the P2P network whereas the base file is not shareable.

III. BUILDING BLOCKS

Digital Signature Algorithm (DSA), Symmetric key cryptography, Zero-Knowledge Proof (ZKP) of identity, hash function, fingerprint collusion-resistance algorithm and Quantization Indexed Modulation (QIM) scheme are useful technologies for the proposed P2P content distribution system. In our proposed system, DSA and ZKP of identity are used by the users to authenticate each other's identity and generate a one-time session key for secure data exchange. An unforgeable and verifiable pseudonym for each entity of the system is generated by using standard hash function. Symmetric key encryption is used to encrypt and decrypt the data exchanged anonymously between the two parties. Tardos Codes are used in our system in order to provide collusion-resistant against colluders and these fingerprints are embedded into the content using a QIM based watermarking scheme.

A. Digital Signature Standard (DSS)

The DSS [5] is the Digital Signature Algorithm (DSA) which is used to generate digital signature for the authentication of an identity of the message sender and data integrity. The algorithm works in conjunction with the Secure Hash Algorithm (SHA). DSA uses the following three parameters which are publicly known:

- P: a large prime number (at least 1024 bits)
- Q: a sufficiently large prime number (at least 160 bits) that is also a divisor of (p-1)
- g: a generator for the multiplicative subgroup of order Q of integers modulo P

A DSA private key is an integer x taken modulo Q and the public key is the integer $y = g^x \bmod P$.

B. Symmetric key cryptography

To prevent illegal copying of digital content, the content is generally encrypted using a symmetric key algorithm. Symmetric key encryption is a cryptography technique that uses a shared secret key to encrypt and decrypt data. Symmetric encryption algorithms (AES, DES, etc) are very efficient at processing large amounts of information and computationally less intensive than asymmetric encryption algorithms (RSA, El Gamal, etc).

C. Zero-knowledge proof of identity

Zero-knowledge proof of identity [6] system is a cryptographic protocol between two parties whereby, the first party wants to prove his/her identity to the second party, without revealing anything about his/her identity to the second party. Following are the three main properties of zero knowledge proof of identity:

1) Completeness: The honest prover convinces the honest verifier that the secret statement is true.

2) Soundness: Cheating prover can't convince the honest verifier that a statement is true (if the statement is really false).

3) *Zero-knowledge*: Cheating verifier can not get anything other than prover's public data sent from the honest prover.

In our proposed scheme, we employ zero-knowledge proof of identity scheme of [7] in Section IV. In this algorithm, the central authority being used in basic ZKP of identity protocol [6] has been eliminated to make it adaptable to P2P networks.

D. SHA-1

SHA-1 is a secure hashing algorithm which is used to output a 160-bit message digest of any input file less than 2^{64} bits. The SHA-1 hash is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest

We employ SHA-1 to use it with the DSA as specified in the DSS. The initiator and receiver of a message compute and verify a digital signature using SHA-1. In the spirit of Pseudo-Trust [7], we use a one-way function to generate pseudonyms together with a proof.

E. Tardos Codes

A variation of Tardos codes, i.e., Nuida's codes [8] are used in the proposed system for fingerprint generation. These codes are based on the Marking assumption, i.e., set of colluders *c* can only alter those bits of the codeword that differ between colluders. The fingerprinting code *F* and a secret vector *s* are outputs of this algorithm. The details of Nuida's codes construction and traitor tracing algorithm can be found in [8].

F. Quantization Indexed Modulation (QIM)

QIM [9] is a relatively recent watermark embedding technique. It has become popular because of the high watermarking capacity and the ease of implementation. The basic QIM scheme embeds a fingerprint bit *f* by quantizing a Discrete Wavelets Transform (DWT) coefficient *W* by choosing between a quantizer with even or odd values, depending on the binary value of *f*. It is important to consider an optimal selection of the embedding quantizer step size Δ and scaling factor, so that the best tradeoff between robustness and minimum quality degradation can automatically be achieved.

IV. PROPOSED SYSTEM

The proposed privacy-aware content distribution scheme shown in Fig. 1 involves five entities:

- A merchant *M* is an entity that distributes the copyrighted content to end users (peers) in the P2P system. It is involved in fingerprint generation and embedding, base and supplementary file distribution, traitor tracing and dispute resolution.
- A peer is an entity that can either play a role of a content requester or provider. Some of the peers with additional facilities act as super-peers. A peer is involved in acquisition of a base file from the merchant, distribution of a supplementary file in the system and a dispute resolution, in case he/she is found guilty of copyright violation.
- A super peer *SP* (a.k.a index server) is a peer with additional facilities which is assigned a role of content server for a small portion of the network. Each peer on registration with P2P system may upload their files to *SP*. *SP* maintains a list of the peers connected to the network and act as central servers. However, instead of peers' address, their pseudonyms are stored. The peers send their queries to the *SP* for downloading their files of interest. The *SP* receives a supplementary file from *M*. On receiving the file, it divides the content into multiple fragments and transmits these fragments to users. Initially, *SP* is booted with a supplementary file by *M*.
- A Monitor *MO* functions as a trusted party to keep the encrypted hash of the secret vector used in the fingerprint generation with the pseudonyms of the merchant and the peer. It also keeps the record of transactions between *SP* and the peer. In case of traitor tracing, the *MO* reveals the encrypted hash of the secret vector to *M*.
- A judge *J* is assumed to be a trusted party which resolves the disputes between *M* and a peer with the cooperation of *MO*.

Setup: The setup of the system involves two things, (1) A Hybrid structured P2P design is opted which provides efficient data search and consists of multiple coordinators called super-peers. At the system startup, the bootstrapping can be done via a well-known booting node. (2) To protect real identities of entities in the system, each entity is required to generate a pseudo identity *P* (pseudonym) and a pseudo identity certificate (*PC*) based on [7] before joining the system. A peer with *PC* can verify what it claims to be to the other party. Each peer uses its unique secret as an input for a pseudo identity generation. The complete description of the generation of *P* and *PC* can be found in [7]. On registration, each peer transfers its *P*, *PC* and public key to the connected *SP*. *P* and *PC* are used by peers for an anonymous communication in the system.

The proposed scheme as shown in the Fig. 1 consists of the following sub-protocols.

A. Fingerprint generation process

The algorithm for fingerprint generation takes a parameter ε for error probability and the total number *N* of users as an input, and outputs a collection $F = (f_1, .., f_N)$ of binary codewords f_i of length *m* and a secret vector *s*. The codeword F_i is sent to the user *i*, while the encrypted vector *s* and a hash of *s* is sent to a *MO* to be used later in dispute resolution.

B. Base file and supplementary file generation

The base file (*BF*) is designed to have a small size and is distributed from *M* to all the peers on receiving a payment for the requested file. The proposed method employs DWT to split the content into low-frequency (approximate coefficients) and

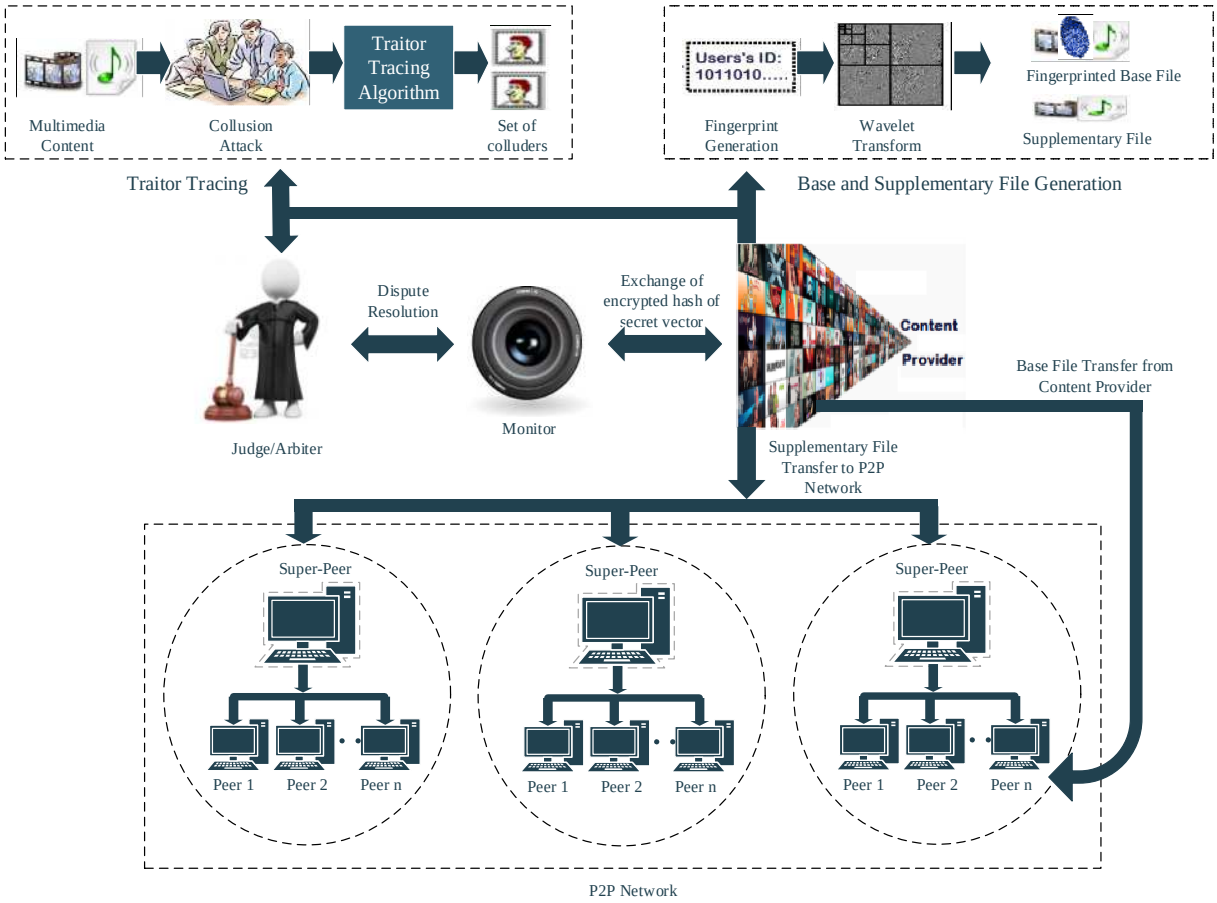

Figure 1. Secure anf Privacy-aware P2P Content Distribution System

high-frequency (detail coefficients) components. In case of audio DWT decomposition, we get approximate and detail (C_A, C_D) coefficients whereas in case of an image file obtained from a frame of a video, we get one approximate component (*LL*) and three detail (horizontal (*HL*), vertical (*LH*) and diagonal (*HH*)) components. An approximate coefficient is then itself split into a second-level approximation and detail coefficients, and the process is repeated. For example, for four-level DWT decomposition of an image of size 1024x768, the coefficient set is, $W=[W_{A4},W_{H4},W_{D4},W_{V4},...,W_{A1},W_{H1},W_{D1},W_{V1}]$. At the fourth level, the size of the approximate coefficient W_{A4} is significantly reduced to 64x48. DWT level-4 decomposition has been chosen for our design considering the trade-off between robustness, capacity and transparency properties of watermarking. W_{A4} provides more robustness and reduced file size without affecting the quality. The fingerprint *fi* of length *m* is then imperceptibly embedded into this W_{A4} using a blind, robust and secure QIM-based watermarking technique [8]. This constitutes a *BF* which is sent to the end user. The remaining detail coefficients constitute a supplementary file (*SF*), which is sent to the *SP* for distribution in P2P system. Eventually, the system at the user end can apply the inverse DWT to get a fingerprinted copy.

C. Distribution of the base file

On receiving a file request from a peer P_i (P_i is the pseudo identity of a peer *i*), *SP* directs P_i's request to the *M (pseudo identity of a merchant). M* first initiates the authentication procedure to verify that the peer claiming to be the holder of P_i is not lying. An anonymous two-party AKE protocol based on PseudoTrust [7] is established between *M* and P_i. *M* sends an authentication request to P_i. P_i replies to *M* with its PC_i (pseudo identity certificate of peer P_i). Having the PC_i of P_i, *M* initiates the authentication process. The authentication process includes two main phases: (1) P_i acts as a prover to prove its validation to *M* and (2) *M* proves its authenticity to P_i.

For generation of a session key for secure *BF* exchange between P_i and *M*, Diffie-Hellman Key Exchange protocol is used in the authentication process. A DSS is adopted to simplify the AKE protocol.

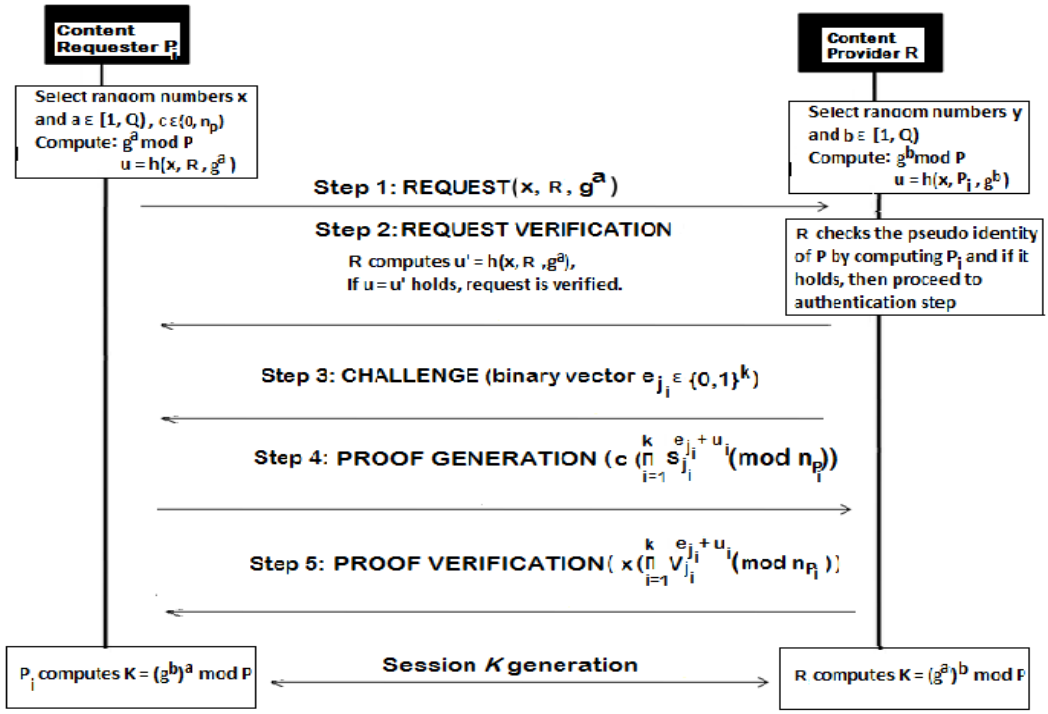

Figure 2. Two-party anonymous AKE protocol Based on PseudoTrust [7]

Fig. 2 illustrates the authentication process between *M* and P_i. For secure exchange of *h(p), M* and *MO* undergo a two-party AKE to exchange one time secret session key K_2 as illustrated in the Fig.2. M encrypts *s* with its public key K_{PM} and further encrypts (E_{KPM} *[s], h(s) M,* P_i*)* with *MO*'s public key (K_{Pm}) and obtains $E_1 = E_{KPm}$ (E_{KPM} *[s], h(s), M,* P_i). Then it encrypts it again with the session-key K_2 to get $E_{K2}(E_1)$. *MO*, upon receiving the encrypted content, decrypts it with the session key K_2 and further decrypts it with its private key K_{Sm}. *MO* stores E_{KMP}*[s]* and *h(s)* along with the pseudonyms of *M* and P_i.

After mutual authentication between P_i and *M*, the session key K_1 is generated to encrypt *BF*: $E_{BF} = E_{K1}$ *(BF)* and send the encrypted content to the requesting P_i. *M* generates *h(BF*, and saves *h(BF)*, P_i against a transaction ID. P_i decrypts it with K_1 and gets a decrypted *BF*.

D. Distribution of the supplementary file

SP searches for a particular file requested by a peer within its group. If found, it displays the list of peers having that particular file and also displays randomly selected peer nodes to act as middle nodes between content providing peer and the requesting peer. On joining the system, peers construct anonymous paths with existing peers and find their tail nodes based on the APFS protocol [10]. These middle nodes form a covert path for the content-providing peer to send the content to the requesting peer. The content-providing peer pre-constructs an onion-path which points to it and add this path to the *SP*. By doing so, a requesting peer can utilize the onion-path to contact the content-providing peer while knowing nothing about provider's identity. If *SP* is unable to find the file within its group, it sends a request for the file to other connected *SP*s. The other *SP* on finding the particular content-provider sends the response to the requesting *SP*. The *SP* then establishes a path between the requesting peer and that content-providing peer.

Let's assume that *a* is a requesting peer (*P_a*), *b* is the providing peer (*P_b*) and *T_p* acts as a tail node to relay a message for the requesting peer *P_a*. When a peer *P_b* receives the file request and it holds the requested file and decided to be the file provider, it replies to the query through its tail-node *T_b*. The requesting peer *P_a* initiates the authentication process to verify the identity of *P_b*. *P_a* sends an authentication request to *P_b* through the anonymous path, *P_a* → *T_a*→ *T_b*→ *P_b*. Using the two-party anonymous AKE as illustrated in Fig.2, both parties authenticate each other identities and generates a one-time session key *K_ab* to encrypt the content of the file. *P_b* sends the encrypted file *F* and hash of the file (E_{K_ab}*(F,h(F)*) to *P_a* through *T_b* and *T_a*.

In order to transfer *P_a*, *P_b*, *h* (*F)* along with the ID of *SP* securely to *MO,* a session key is generated using two party AKE as described in Fig. 2. At the completion of *SP* transfer to *P_a*, *MO* concatenates all the *h(F)*s and stores this concatenated hash and *P_a* against a transaction ID.

E. Traitor Tracing

Once a pirate copy *Y* of content *X* is found, *M* extracts the fingerprint by decomposing the pirated content *Y* with the same wavelet basis used in the fingerprint insertion step. This gives the approximate coefficient matrix in which pirated code *pc* ε $\{0, 1\}^*$ is embedded. After extraction of a pirated code from *Y*, the tracing algorithm of Nuida's codes is employed to identify colluder(s).

In the tracing algorithm, a colluded fingerprint *pc* and secret vector *s* are given as an input. The encrypted secret vector *s* is obtained from *MO* in order to generate the fingerprint matrix for identification of the colluder(s). *M* further decrypts *s* using its private key K_{SM}. The colluder detection threshold parameter and the score of user are calculated as per algorithm described in [8]. The output of this tracing algorithm is the set of users with scores above the defined threshold value. *M* cannot accuse any user of producing a forged copy due to commitment of secret vector *s* to *MO*, which is being trusted by both merchant and a buyer.

F. Arbitration

If the user has been identified as a colluder through detection algorithm and he/she denies that an unauthorized copy has been originated from his/her copy and want to prove his/her innocence, a judge *J* can be requested to solve this conflict. *J* is assumed to be a trusted party, which is able to resolve conflicts without the user revealing his/her identity. Moreover, *J* does not require the whole fingerprinted content. On receiving request from a user to deny accusation of piracy, *J* gets a pirated copy's hash from traitor tracing step. *J* requests *MO* to provide the hash of the file registered against the user's pseudonym. If both the hashes have high correlation, it means end user is guilty else he/she is innocent. If found guilty, the real identity of the user can be traced with the help of *MO*. Since *MO* has kept record of each transaction between a peer and *SP*, it can identify the *SP* with whom the accused peer was connected to and thus through its public key, its real identity can be revealed to *J*.

V. CONCLUSIONS

In this paper, we have proposed a privacy-aware content distribution mechanism which provides security and anonymity to both the merchant and buyer. The newly proposed scheme is specific for P2P networks and can benefit multimedia owners to share their big files without fear of copyright violation, such as video files, utilizing the convenience of P2P networks. This scheme reduces the burden of the media owner's server by only sending a small-sized base file and making use of the P2P network to support the majority of the file transfer process. The wavelet technique makes the base file into necessary information for the customer and fingerprint generated using Tardos codes, offers collusion-resistance against a chosen number of colluders and illegal-redistributors' identification whilst preserving the privacy of honest users.

Future Work: Preliminary results of our proposed system show that the base file of both the audio and video files is considerably small in size as compared to the original size. In our forthcoming paper, we will discuss the security and performance analysis of the system. Also, we will compare its performance with similar P2P content distribution systems in terms of security and privacy properties and computational cost.

ACKNOWLEDGMENT

This work was partly funded by the Spanish Government through projects TIN2011-27076-C03-02 ''CO-PRIVACY'' and CONSOLIDER INGENIO 2010 CSD2007-0004 ''ARES''.

REFERENCES

[1] D. Megías and J. Domingo-Ferrer, "Privacy-aware peer to peer content distribution using automatically recombined fingerprints", Multimedia Systems, (In press)

[2] J. Domingo-Ferrer and D. Megías, "Distributed multicast of fingerprinted cntent based on a rational peer-to-peer community", Computer. Communication, vol. 36, no. 5, Elsevier, 2013, pp. 542-550.

[3] Amna Qureshi, Helena Rifà Pous and David Megías, "Security, Privacy and Anonymity in Legal Distribution of Copyrighted Multimedia Content over Peer to Peer Networks A Brief Overview", Proc. Fifth International Conference on Multimedia Information Networking and Security (MINES), IEEE, 2013.

[4] G. Tardos, “Optimal probabilistic _fingerprint codes,” Proc. Symposium on Theory of Computing (STOC), ACM, 2003, pp. 116-125.

[5] Digital Signature Standard, FIPs PUB 186-4, [Online]. Available: http://csrc.nist.gov/publications/fips/fips186-4/fips186-4 change1.pdf, accessed 14.12.2013.

[6] U. Fiege, A. Fiat and A. Shamir, "Zero Knowledge Proofs of Identity," Journal of Ccryptology, vol. 1, no. 2, Springer, 1988, pp. 77-94.

[7] L. Lu *et al.* , “Pseudo Trust: Zero-knowledge authentication in anonymous P2Ps,” IEEE Trans. Parallel and Distrib. Syst., vol. 19, no. 10, IEEE, 2007, pp. 1325- 1337.

[8] K. Nuida, “Short collusion-secure fingerprint codes against three pirates”, International Journal of Information Security, vol. 11, no. 2, Springer, 2012, pp. 85-102.

[9] B. Chen and G.W.Wornell, “Quantization index modulation: A class of provably good methods for digital watermarking and information embedding,” IEEE Transactions on Information Theory, vol. 47, no. 4, IEEE, 2001, pp. 1423–1443.

[10] V. Scarlata, B. N. Levine and C. Shields, "Responder Anonymity and Anonymous Peer-to-Peer File Sharing", Proc. Ninth International Conference on Network Protocols (ICNP), IEEE, 2001, pp. 272-280.