Framework for Preserving Security and Privacy in Peer-to-Peer Content Distribution Systems

Amna Qureshi, David Megías, Helena Rifà-Pous

*Universitat Oberta de Catalunya (UOC), Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica Multimèdia i Telecomunicació, Rambla del Poblenou, 156, 08018,
Barcelona, Catalonia, Spain*

Email {aqureshi,dmegias,hrifa}@uoc.edu

Abstract

The use of Peer-to-Peer (P2P) networks for multimedia distribution has spread out globally in recent years. The mass popularity is primarily driven by efficient distribution of content, also giving rise to piracy. An end user (buyer) of a P2P content distribution system does not want to reveal his/her identity during a transaction with a content owner (merchant), whereas the merchant does not want the buyer to further distribute the content illegally. Therefore, there is a strong need for a content distribution mechanism over P2P networks that do not pose security and privacy threats to the copyright holders and end users, respectively. The existent systems for copyright and privacy protection employ cryptographic mechanisms at a cost of high computational burden which makes these systems impractical to distribute large sized files, such as music albums or movies. In this paper, we propose and analyse a P2P content distribution system which allows efficient distribution of large-sized content while preserving the security and privacy of merchants and buyers, respectively. Our proposed framework is able to resolve the problems of piracy tracing, buyer frameproofness, collusion resistance, dispute resolution and buyer's anonymity. We have carried out simulations to evaluate the performance of our framework in terms of imperceptibility, robustness, throughput and content delivery costs. The experimental results confirm that the proposed framework provides an efficient solution to copyright infringement issues over P2P networks, reducing the multimedia file sizes as much as five times on average, while protecting the end users' privacy and anonymity.

Keywords: privacy; security; anonymity; collusion-resistant fingerprinting; peer-to-peer networks

1 Introduction

P2P systems are attractive because they do not require any special administrative arrangements, unlike centralized facilities, and their decentralized and distributed nature make them scalable, bandwidth efficient and fault-tolerant. P2P applications account for approximately 60% of Internet's traffic (García-Dorado, Finamore, Mellia, Meo, & Munafó, 2012). In particular, P2P content distribution applications (*eDonkey2000*, 2000; *gtk-Gnutella*, 2000) are extremely popular among millions of users. These applications allow users to contribute, search and obtain a digital content in a distributed manner. Content distribution in P2P has also received considerable attention in the research community (Passarella, 2012; Theotokis & Spinellis, 2004). The P2P technology for content distribution systems is beneficial to both content providers and end users. From the media companies point-of view, the P2P technology enables them to make valuable content available to a large number of people at minimal cost and better performance.

These benefits are the attractive features for media companies towards the adoption of P2P systems, e.g. BitTorrent (*BitTorrent*, 2001) is one of the most popular P2P distribution systems used on the Internet and it accounts for a significant amount of traffic on the Internet. Similarly, Internap (*Internap*, 1996), a managed P2P content distribution application, enables content owners and media companies to publish, distribute and track their games, video and software at reduced delivery costs. Besides, from the end users perspective, audio, video and software files can easily be accessed and downloaded within a short time.

Despite the potential of P2P content distribution technology to revolutionize the Internet in numerous respects, it has often been surrounded with the copyright controversy. The copyright holders argue that they provide copyright content to the end users of the systems and that these end users are involved in illegal re-distributions. They apparently fear losing control of content ownership and worry about the illegal activity promotion. Moreover, tracing (Chor, Fiat, Naor, & Pinkas, 2000) a copyright violator is an immense task which requires content providers to work in conjunction with watermarking (Hartung & Kutter, 1999; Cox, Miller, Bloom, Fridrich, & Kalker, 2007; Bianchi & Piva, 2013) and fingerprinting (Barg, Blakley, & Kabatiansky, 2003; Voloshynovskiy, Farhadzadeh, Koval, & Holotyak, 2012) providers as well as P2P content distribution service developers. However, this illegal re-distribution (Von-Lohmann, 2003) act is not only onerous to content providers but also to the end users. The major concern among end users is whether the presence of copyright protection mechanisms (Lian, Kanellopoulos, & Ruffo, 2009) in P2P distribution systems can violate their privacy interests. The fact that a tracing mechanism makes use of a record which details what multimedia files are shared through a specific IP address, or a list of the peers with whom a user has interacted, disrespects the privacy of the user. Therefore, there is an inherent conflict of interest between copyright protection supporters and privacy advocates and thus there is a need to balance security and privacy needs when developing P2P content distribution systems. Similarly, the conflict between privacy and security within P2P content distribution system manifests itself in a debate between anonymity and accountability, i.e. decreased anonymity (less user privacy) is proportional to increased accountability (more security to provider). Currently, security and privacy in P2P systems is a hot research area among researchers who are focusing on the preservation of content providers ownership properties, content receivers' privacy and accountability. However, most of the existing P2P content distribution systems provide security and privacy at a cost of high computational burden at the merchant's and/or at the user's end.

In this paper, we propose a P2P content distribution system that provides copyright protection to the merchant at a reduced computational cost and also offers revocable privacy to an end user. In the proposed system, the multimedia file is partitioned by the merchant into a base and a supplementary file. The base file is much smaller than the original file and contains the most important information. Without this information, the supplementary file is unusable.

The base file is dispensed by the merchant on payment from the user and a supplementary file is distributed to the P2P network to be distributed in P2P fashion. Thus, this scheme enables the merchant to save bandwidth and CPU time. The asymmetric fingerprinting protocol is performed by the merchant and the buyer in the presence of a trusted party in such a way, that the merchant does not know the fingerprint and the fingerprinted content, while the buyer receives fingerprinted content with his/her unique identity. Collusion-resistant fingerprinting codes are embedded by the merchant into the content so as to identify an illegal re-distributor(s) from a pirated content. The proposed framework also enables buyers to obtain digital contents anonymously, but this anonymity can be revoked as soon as he/she is found guilty for copyright violation. The buyers are provided anonymity by using dynamic pseudonyms instead of their real IDs. To ensure anonymous communication between buyers, onion-routing is used for an anonymous data transfer. Moreover, to provide accountability, a key agreement protocol has been adopted in our scheme. The simulation results show that the proposed framework yields an effective reduction in the computational overheads for a merchant. Also, the security analysis proves that the proposed system exhibits security and conditional anonymity to the merchant and the buyer, respectively.

The paper is organized as follows. Section 2 reviews the related work on P2P networks, multimedia content and privacy protection schemes and P2P content distribution systems. Section 3 provides the building blocks of the proposed framework. Section 4 discusses the design of the framework. Section 5 presents the results of the experiments designed to evaluate the performance of the framework. Also the security analysis of the proposed framework is discussed in this section. Finally, Section 6 summarizes the conclusions and future research issues.

2 Related Work

This section reviews related work on P2P networks, multimedia content protection schemes, privacy protection mechanisms, and P2P content distribution systems.

2.1 P2P Architectures

Peer-to-peer networks may be categorized into the three categories: Centralized P2P networks, Pure P2P networks and Hybrid P2P networks. In centralized P2P network (*Napster*, 2011), a central server is used which manages the files and user databases of multiple peers that log onto it. These networks provide the highest performance but suffer from lack of scalability and a single point of failure. In pure P2P Networks (*Freenet*, 2000), all the peers have similar responsibilities acting as both server and client. These networks offer inherent scalability and avoidance of a single point of failure but at a cost of slow information discovery and increased overhead traffic. A hybrid network (*iMesh*, 1999) combines the features of both the centralized and pure P2P networks. Within these networks, some peers on the basis of their resources (storage, CPU, etc.) are elected as super peers. The super peers are assigned with responsibilities like maintaining a central index of the files shared by peers and helping a peer in establishing a relationship with another peer, etc. Hybrid P2P networks provide an efficient search mechanism with no single point of failure. Few hybrid P2P systems (Rodriguez-Perez, Esparza, & Muñoz, 2008) can be found in the literature that selects super peers on the basis of their reputation among the peers.

2.2 Multimedia Content Protection Mechanisms

Piracies of multimedia contents are increasing with the pervasive usage of content distribution systems. Mechanisms must be deployed to ensure that the multimedia content can be used safely by legitimate users who have appropriate usage rights of that content. In this section, we provide a brief overview of state-of-the art content protection technologies, i.e. digital watermarking, fingerprinting and buyer-seller watermarking protocols.

2.2.1 Digital Watermarking and Fingerprinting

Encryption can be used to package the multimedia content securely and enforce all access rules to the protected content. However, once the content is decrypted by an authorized user, it does not provide any protection to the content (Grangetto, Magli, & Olmo, 2006). Thus, encryption alone is not enough to prevent an authorized user from illegal re-distribution. Similarly, classic Digital Rights Management systems (*Apple iTunes*, 2001; *Microsoft DRM*, 2008) that are considered as a second line of defence against copyright violation, does not prove to be an effective access control against a user with the knowledge and determination to violate it. For content owners, digital watermarking proves to be a more effective anti-piracy solution. Digital watermarking has become a significant area of research and development, and the usage of these techniques is now being considered a requisite to address the issues faced by the proliferation of digital content. Watermarking consists of embedding a watermark, into the content and can later be used to check the source of the content. There are two forms of watermarking, copyright watermarking and fingerprint watermarking (fingerprinting). In copyright watermarking, a watermark is embedded into the content which indicates copyright holder's identification. This is used to declare the copyright and cannot be used to trace the copyright violator. Whereas, in fingerprinting, a user-specific identification mark is embedded into the content so that it can be used to track an illegal re-distributor.

Watermarking schemes have some important desirable properties (e.g. robustness, imperceptibility, capacity and detection) and each of these properties must be taken into consideration when applying a certain watermarking technique. The robustness property requires that the watermark should tolerate all kinds of signal processing operations (at least below some distortion threshold), or unauthorized access. In case of imperceptibility, the embedded watermark should be perceptually invisible such that the quality of the content is not degraded. The capacity refers to the maximal data volumes that can be embedded into a multimedia content. If the detection of the digital watermark can be done without the original data and original watermark, such a technique is called blind detection. However, three watermarking properties (robustness, imperceptibility and capacity) contradict one another, i.e. if one is increased, the other decreases. Thus, it is very important for researchers to achieve a convenient trade-off between these properties according to the application requirements. Various watermarking schemes (Fallahpour & Megías, 2011; Cao & Huang, 2012) have been proposed to achieve a better trade-off between these properties.

In digital fingerprinting, a user specific identification mark, known as a fingerprint, is embedded into different copies of the same content. The resulting copies are referred to as fingerprinted copies and each fingerprinted copy is assigned to a user. However, digital fingerprinting technology is vulnerable to collusion attacks in which a group of malicious users (colluders) combine their copies with different fingerprints to either remove the embedded fingerprints or frame innocent users. If an illegal copy appears as a result of collusion activity, the embedded user information can be extracted to help trace or identify illegal users. A large amount of research work has been carried out in designing collusion resistant fingerprinting schemes (Boneh & Shaw, 1999; Camenisch, 2000; Domingo-Ferrer & Herrera-Joancomartí, 2000) for copyright protection.

2.2.2 Buyer-Seller Watermarking Protocols

In traditional fingerprinting, the fingerprint is generated and embedded solely by the merchant (assumed to be an honest party) and the buyer has no control over the embedding process (Cox, Kilian, Leighton, & Shamoon, 1997). Thus, a dishonest merchant can frame an innocent buyer, while a cheating buyer can deny his/her responsibility for a copyright violation act. The authors in (Qian & Nahrstedt, 1998) identified this problem as customer's rights problem. Asymmetric fingerprinting schemes (Pfitzmann & Schunter, 1996) were introduced to overcome the customer's rights problem, where only the buyer obtains the marked copies of the content. It is an interactive protocol between a merchant and a buyer, in which a buyer has to interact with the merchant to purchase the content. After the interaction, the buyer receives a marked copy and the merchant does not know the exact marked copy the buyer receives. In case a merchant finds an unlawfully re-distributed copy, he/she can prove the re-distributor's treachery to the third party. In asymmetric fingerprinting, the identity of the buyer is compromised and to solve the anonymity problem, anonymous fingerprinting (Kuribayashi & Tanaka, 2005; Memon & Wong, 2001) schemes were introduced. The schemes retain the asymmetric property and protect the privacy of a buyer, whose identity is only revealed in case of piracy. Trusted third parties were introduced in these schemes to guarantee that a protocol is fair to both the merchant and the buyer in a digital content transaction. The role of this trusted third party is mainly to generate a secret watermark and protect the privacy of a buyer. There exists various fingerprinting schemes that do not involve third parties for execution of the protocols (Choi, Sakurai, & Park, 2003; Deng & Preneel, 2008). However, implementation of such schemes involves complex cryptographic protocols such as, bit commitment and multi-party security protocols, that requires high bandwidth and heavy computational costs, thus making the schemes impractical in real-world scenario. It is proven in (Pagnia & Gartner, 1999) that efficient fair exchange protocols cannot be completely fair without the help of a third party that is mutually trusted by both of the parties performing the exchange. Thus, using a trusted third party is a price worth paying if it can turn a fingerprinting scheme into a practical alternative (Martínez-Ballesté, Sebé, Domingo-Ferrer, & Soriano, 2003). A buyer-seller watermarking scheme (Phan, Goi, Poh, & Jongsung, 2011; Rial, Deng, Bianchi, Piva, & Preneel, 2010) incorporates both watermarking and fingerprinting mechanisms to protect the rights of both the content provider (merchant) and the customer (buyer). A complete and sound buyer-seller watermarking protocol is expected to solve the following requirements (Deng & Preneel, 2008):

- **Anonymity:** The buyer should remain anonymous during transactions unless he/she is proven to be guilty.
- **Customer's rights problem:** When a watermark is inserted solely by the merchant, the merchant may benefit from framing attacks to an innocent buyer, i.e. a dishonest merchant may replicate the digital content and frame an innocent buyer for piracy. Thus, an honest buyer should not be falsely accused by a malicious merchant.
- **Piracy tracing:** Once a pirated copy is found, the merchant should be able to trace and identify the copyright violator.
- **Collusion attack:** In collusion attack, several attackers fabricate a new copy through combining their unique copies in order to avoid the tracing. Attackers intend to remove the embedded fingerprint by making use of the slight difference between different copies. Thus, the scheme must be collusion-secure.
- **Dispute problem:** The judge/arbitrator should be able to resolve disputes, without the buyer revealing his/her identity or private key.
- **Non-repudiation:** The buyer accused of re-distribution of an unauthorized copy should not be able to claim that the copy was created by the merchant.
- **Unlinkability:** Nobody can determine whether different watermarked contents are purchased by the same buyer.

2.3 Privacy Protection Mechanism

In P2P systems, encrypting the communication between two persons can only hide the contents of their transaction. The malicious entities can get various details like IP addresses, duration of communication, etc. that can reveal their identity. Thus, there is a necessity to hide this information to enhance the privacy of users in a system. In communication perspective, there exist three types of anonymity: Sender anonymity (the identity of the user who initiated a communication is hidden), receiver anonymity (the identity of the user who responds to sender's queries and send files accordingly is hidden) and mutual anonymity (the identities of both the sender and the receiver are hidden from each other and from other users in the system, and also the communication between the sender and the receiver is hidden). Various anonymity mechanisms have been proposed that serve as tools for the protection of data and user privacy in distributed applications. Anonymous communication and pseudonymity techniques are mostly found in the literature. Anonymous communication generally aims to make communication ambiguous in order to make it difficult for malicious users to collect information about the system entities and the shared data. For example, in onion routing (Scarlata, Levine, & Shields, 2001), messages are repeatedly encrypted and then sent through several nodes called onion routers. Each onion router removes a layer of encryption to uncover routing instructions, and sends the message to the next router where this is repeated. This prevents intermediary nodes from knowing the source, destination and contents of the message. Pseudonyms are dynamic identifiers of the users that are hard to be linked to the real identities, thus, making a user indistinguisable from other users and providing anonymity to a user among a group of users.

2.4 P2P Content Distribution Systems

A content distribution system is a distributed system that maintains content servers in many different geo-locations in order to improve distribution efficiency of the content. While many overlay networks have been proposed for deploying content distribution services, Content Delivery Networks (*Akamai*, 1998) and P2P networks (*eDonkey2000*, 2000) are amongst the most commonly applied. A Content Delivery Network (CDN) is a client-server based network infrastructure in which clients download content from dedicated and centrally managed servers. However, content providers using CDN have to bear an initial infrastructure investment and high maintenance costs of servers. Therefore, in recent years, P2P networks have emerged as a popular solution to deliver multimedia content efficiently to a large number of Internet users. The literature review shows that few researchers have worked on a P2P content distribution system that provides preservation of content providers ownership properties and content receivers' privacy and anonymity so far. The following paragraph gives a brief review of two P2P content distribution systems designed with an intention to satisfy both copyright protection and user privacy.

Megías and Domingo-Ferrer (Megías & Domingo-Ferrer, 2013) introduced a novel concept of a recombination fingerprinting mechanism for P2P content distribution. The proposed scheme utilizes the fingerprinting concept to provide identification to the copyright owner, offers collusion resistance against dishonest buyers trying to create a forged copy, offers conditional privacy to the users and detect illegal content re-distributors. However, this system is implemented with a two-layer anti-collusion code (segment level and fingerprint level), that results in a longer codeword. Furthermore, honest and committed proxies are required in the system for the generation of valid fingerprints as compared to the proposed framework which only requires an honest Monitor (*MO*) for the fingerprint generation. In (Domingo-Ferrer & Megías, 2013), Domingo-Ferrer and Megías proposed a P2P protocol for distributed multicast of fingerprinted content in which cryptographic primitives and a robust watermarking technique are used to produce different marked copies of the content for the requesting user such that it can help the provider to trace re-distributors without affecting the privacy of honest users. However, an implementation of secure multi-party protocol results in increased computational and communication costs at the user end.

Unlike the P2P content distribution systems described in the above paragraph, the following P2P distribution systems either provide copyright protection or user privacy. Li *et al.* (J. S. Li, Hsieh, & Hung, 2010) proposed a DRM-enabled P2P architecture, in which RSA public-key cryptosystem is used to generate a unique digital fingerprint for every user of the network. The proposed system provides an efficient and secure means of distributing large-scale copyright-protected music contents over P2P networks but it fails to provide privacy to the end users. Similarly, a fingerprinting method was proposed in (X. Li, Krishnan, & Ngok-Wah, 2010) for complex P2P file sharing networks for copyright protection. In this system, wavelet transforms and principal component analysis (PCA) techniques are used for fingerprint generation. The wavelet technique provides a scalable approximation matrix that contains the most important low-frequency information and the PCA technique determines the orthogonal eigenvectors, which makes it possible to maximally distinguish the different fingerprints. The proposed system though provides a solution of legal content distribution, but it does not provide collusion-resistance and user privacy. In (Lu et al., 2008), Lu *et al.* proposed a PseudoTrust model in which pseudonyms are generated to achieve anonymity for users with no trusted third party. Furthermore, a novel authentication scheme has been designed so that peers can be authenticated without leaking any personal information. The PseudoTrust model provides privacy to P2P end users but does not include copyright protection mechanisms. In (Lou & Hwang, 2009), Lou and Hwang presented a P2P content distribution system with a copyright protection mechanism based on enhanced chunking and hashing protocols. The proposed system prevents pirates to download copyrighted files easily by using proactive content poisoning with token based authorization. It uses identity-based signatures to distinguish pirates from legitimate users. While the proposed system provides users with clean content files and punishes the pirates or colluders with poisoned files, the privacy concern of the legitimate users has not been discussed.

Most of the past studies focused on either providing a copyright protection to content owners or privacy to end users, whereas our work proposes a framework for P2P based content distribution focusing on both copyright protection and privacy. Moreover, in achieving either one or both of these two properties, there is a computational and communicational burden at content owner or buyer's end. Our work differs from existing studies in a way that we focus on the design and implementation of the multimedia content distribution over the P2P network that provides both copyright protection and privacy at a reduced computational cost to a merchant and an end user. Also, in order to provide secure and authentic anonymous data exchange between providers and receivers in P2P network, we have make use of anonymous communication and anonymous authentication techniques. Table 1 provides the comparison of the proposed system with the existing P2P content distribution systems, in terms of functionalities such as copyright protection, user privacy, collusion-resistance, accountability and resistance to communication attacks.

Table 1: Functionality Comparison

	Copyright Protection	**User Privacy**	**Collusion Resistance**	**Accountability**	**Resistance to Communication Attacks**
Proposed System	Yes	Yes	Yes	Yes	Yes
(Domingo-Ferrer & Megías, 2013)	Yes	Yes	Yes	Yes	No
(J. S. Li et al., 2010)	Yes	No	No	No	No
(Megías & Domingo-Ferrer, 2013)	Yes	Yes	Yes	Yes	Yes
(X. Li et al., 2010)	Yes	No	Yes	No	No
(Lu et al., 2008)	No	Yes	No	No	Yes
(Lou & Hwang, 2009)	Yes	No	No	Yes	No

3 Building Blocks

Our P2P content distribution framework employs a collusion resistant fingerprint algorithm, Quantization Indexed Modulation (QIM) watermarking, a homomorphic encryption scheme, a digital signature scheme, a PseudoTrust (Lu et al., 2008) model, a hash function and symmetric key cryptography. In this section, we present a brief overview of these building blocks.

3.1 Collusion-secure Codes

A Code F is totally c-secure if there exists a c-frameproof code and a tracing algorithm. A c-frameproof code restricts the collusion size l to c pirates such that a collusion l can only produce codewords that are codes of l and cannot frame an honest user outside this collusion. The tracing algorithm is used when the merchant finds a pirated copy and wants to trace the members of the collusion l. Many c-secure codes have been proposed in the literature and amongst these codes, Tardos codes (Tardos, 2003) have gained widespread acceptance in state-of-the-art publication. A variation of Tardos codes, i.e. Nuida *et al.*'s c-secure codes (Nuida et al., 2007), are used in the proposed system for fingerprint generation. These codes are based on a δ-marking assumption, i.e. the number of undetectable bits that are either erased or flipped is bounded by δ-fraction of the total code length. The number of users N, error-probability ε and number of colluders c are inputs of Nuida *et al.*'s fingerprint generation algorithm. The fingerprinting code F and a secret vector p are the outputs of this algorithm.

3.2 Embedding Algorithm

An embedding algorithm is used to insert a fingerprint into different copies of the same content. Multimedia fingerprinting requires the use of robust data embedding methods that are capable of withstanding attacks aimed at removing the fingerprint. QIM (Chen & Wornell, 2001) 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 value, e.g. 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 a scaling factor, so that the best trade-off between robustness and minimum quality degradation can automatically be achieved. QIM modulates quantization (integer-valued) levels to embed information bits into a signal and therefore, can be implemented in an additive homomorphic cryptosystem. However, basic QIM watermarking has limited robustness compared to other watermarking schemes. The embedding positions can be retrieved from a single copy and are thus vulnerable to number of signal processing attacks. In order to avoid this problem, we have used Subtractive-Dither QIM (SD-QIM) watermarking technique (Prins, Erkin, & Lagendijk, 2007). In SD-QIM, a pseudorandom noise, called a dither, is added to a signal sample x before embedding an information bit f and subtracting the dither after embedding. A suitable choice for the PDF of the random dither d_i is a uniform distribution on $[-\Delta, \Delta]$.

3.3 Homomorphic Encryption

Homomorphic encryption systems allow operations to be performed on encrypted data without compromising the encryption. An encryption scheme has homomorphic properties when for any given encryption key k, the encryption function satisfies $g(E(M_1), E(M_2)) = E(f(M_1, M_2))$, where $g(.)$ and $f(.)$ are one of the operations, addition, multiplication, XOR, etc. Paillier and Okamoto-Uchiyama are additive homomorphic systems while RSA and ElGamal are multiplicatively homomorphic systems (Schneier, 1996). Homomorphic encryption schemes are used in asymmetric fingerprinting to provide buyer frampeproofness against a dishonest merchant. These homomorphic cryptosystems allow the embedder to insert encrypted fingerprint bits into the encrypted content. In our system, we have used a Paillier cryptosystem (Paillier, 1999) which is homomorphic with respect to the addition operation. Paillier is a probabilistic asymmetric algorithm and a semantically secure cryptosystem based on composite residuosity classes, whose computation is believed to be computationally difficult. The generalized additive homomorphic property of Paillier encryption is defined as, $\prod_{n=1}^{l} E(m_i) = E\left(\sum_{i=1}^{l} m_i\right)$.

3.4 Digital Signature Scheme

A Digital Signature scheme is used to provide data integrity, data origin authentication and non-repudiation. This scheme can be achieved by using crypto signature schemes, e.g. RSA, DSA (Digital Signature Algorithm) (Schneier, 1996), etc. In our proposed system, we have used DSA which is a Federal Information Processing Standard for digital signatures (Kerry, n.d.). DSA computes a digital signature with a set of domain parameters (p, q, g), a private key x, a public key y and a secret number k. The parameters of DSA are: p (a large prime number (of at least 1024 bits)), q (a sufficiently large prime number (of at least 160 bits) that is also a divisor of $(p - 1)$), g (a generator of subgroup of order q modulo p such that $1 < g < p$), x (a pseudo randomly generated integer with $0 < x < q$ (x is in the range $[1, q - 1]$)), y (a public key obtained by $g^x \bmod p$), and k (a pseudo randomly generated integer with $0 < k < q$ (k is in the range $[1, q - 1]$)).

3.5 PseudoTrust Model

In our proposed system, we employ a PseudoTrust model based on a zero-knowledge proof-of-identity (ZKPI) (Feige, Fiat, & Shamir, 1988) proposed by Lu *et al.* (Lu et al., 2008) in which peers authenticate each other with their pseudo-identities without leaking any personal information. The PseudoTrust model enables pseudonym-based trust management so that the real identities of peers are protected during the authentication. Each peer is required to generate a pseudo-identity (*PI*) and a pseudo-identity certificate (*PIC*). A *PI* is used to identify and replace the real identity of a peer in a P2P system. A *PIC* is generated to authenticate the *PI* holder. In the authentication protocol, the Diffie-Hellman key exchange protocol is incorporated to provide confidentiality and integrity to data exchanges, such that after authentication, both peers can share a session key for encrypting the exchanged data. Since, the PseudoTrust model allows peers to generate their pseudo-identities individually and peers do not depend on any third party to authenticate with each other, it creates an accountability problem. Without a trusted third party, it would be impossible to find a person responsible for doing mischievous activity. Thus, to add accountability to the system, we have incorporated an internal certificate authority (CA_R) in the PseudoTrust model. Each peer is authenticated by CA_R before he/she joins the network. Thus, each peer has a private key, a public key and a public key certificate signed by CA_R. The pseudo-identities and certificates are used by the peers for anonymous communication within the P2P system. $\mathbb{G}$, a finite cyclic group with P elements is selected by CA_R, with g as a generator of $\mathbb{G}$. The parameters g and P are made public by CA_R. CA_R then selects a secret random number $r \in [1, \ldots, P - 1)$ and sends r encrypted with the peer's public key to the peer. Thus, CA_R and all the peers share a secret number r. When a new peer joins the network or an old peer leaves the network, the secret number r should be updated. The peers perform authentication protocol without involving CA_R. The details of generation of *PI* is presented in Section 4.3.5.

3.6 Hash Algorithm

A hash function is any algorithm that maps variable-length data to a fixed-length data, called hash value or hash. There are several well-known hash functions, namely, MD2, MD4, MD5 and the Secure Hash Algorithm (SHA) (Schneier, 1996) used in cryptography. SHA is a standard algorithm used typically with other cryptographic algorithms such as digital signature algorithms. SHA specifies four secure hash algorithms- SHA-1, SHA-256, SHA-384 and SHA-512 (Schneier, 1996). SHA-1 is a secure hashing algorithm which is used to output a 160-bit message digest of any input file. In our proposed framework, we use the SHA-1 function to generate unforgeable and verifiable pseudonyms for each entity of the system.

3.7 Symmetric Key Cipher

Symmetric key encryption is a cryptography technique that uses a shared secret key to encrypt and decrypt digital data. Symmetric encryption algorithms such as AES or DES (Schneier, 1996), are very efficient at processing large amounts of information. AES is an iterative symmetric block cipher with superior encryption strength and strong performance. Our proposed framework employs AES-128 algorithm to encrypt the data to be transmitted using the shared session key K_1, generated during anonymous authentication.

4 Proposed System

This section describes the architecture of the proposed P2P content distribution system. In section 4.1, we define the role of each entity and list the notations that are being used in the system. Section 4.2 defines the functional requirements and the assumptions of the design. In section 4.3, we detail the design of our framework which includes the fingerprint generation protocol, the base and supplementary file generation and distribution protocols, the traitor-tracing protocol, the dispute-resolution protocol and cloud computing implementation of the proposed system.

4.1 System Parameters and Entities

In this section, system parameters and a description of each entity of the framework are provided.

Table 2: Parameters and Notations

Parameter	Specification	Generated By
P_{B_i}	Pseudonym of a peer B_i	Peer
f_i	Fingerprint of a peer B_i	Monitor
M	Pseudonym of a merchant	Merchant
MO	Pseudonym of a monitor	Monitor
$\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$	Anonymous certificate of B_i certified by CA_R	CA_R
$\text{Cert}_{CA_R}(M)$	Public Certificate of M certified by CA_R	CA_R
$\text{Cert}_{CA_R}(P)$	Public Certificate of peer P certified by CA_R	CA_R
$\text{Sign}_{B_i}(*)$	Signature of $B_i$using his/her private key	Peer B_i
$\text{Sign}_{K^*_{pB_i}}(*)$	Signature ofB_i using his/her anonymous key	Peer B_i
sk	Secret watermark embedding key	Merchant
Δ	Quantization step size	Merchant
c	Number of colluders	Monitor
ε	Probability of accusing an innocent end user	Monitor
N	Total number of users in the system	Monitor
p	A secret vector	Monitor
m	Length of a fingerprint code	Monitor
X	Original content	Merchant
Y	A pirated copy	Colluders (peers)
pc	Pirated codeword	Colluders (peers)
BF	Base file	Merchant
SF	Supplementary file	Merchant
K	One-time session key	Merchant/Peer/Monitor
T_a	Pseudo-identity of a tail node of a requesting Peer a	Peer a
T_b	Pseudo-identity of a tail node of a provider Peer b	Peer b
K_1	One-time session key generated during two-party anonymous authentication	Peer
(K_{p_M},K_{s_M})	Public and private key pair of a merchant	Merchant
$(K_{p_{MO}},K_{s_{MO}})$	Public and private key pair of a monitor	Monitor
(K_{pB_i},K_{sB_i})	Public and private key pair of the peer B_i	Peer
$(K^*_{pB_i}, K^*_{sB_i})$	Anonymous public and private key pair of the peer B_i	Peer

Table 2 describes important terms and parameters used to benefit our readers. The proposed framework involves seven entities and the function of each entity is defined as follows:

- A merchant M is an entity that distributes the copyrighted content to end users (peers) in the P2P system. It is involved in base file and supplementary file generation and distribution, traitor tracing and dispute resolution.

- A peer P is an entity that can either play a role of a data requester or a provider. A peer is involved in the acquisition of BF from the merchant, the distribution of SF in the system and a dispute resolution if he/she is found guilty of copyright violation.
- A super peer SP is a reputed peer with additional facilities who is assigned the role of the coordinator for a small portion of the group of peers. On registration with SP, each peer may upload the index of the files to it. Each SP maintains a list of the peers connected to the network and acts as a central coordinator. However, SP store peers' pseudonyms instead of their real identities or IP addresses. Initially, SPs are provided with SF from M at the system start-up. On a request from a peer, SP divides the content of SF into multiple fragments and transmits these fragments to the requesting peer.
- A certification authority CA_R is a trusted party that is responsible of issuing certificates to the buyer for acquisition of BF from M and SF from peers. The certificate is used to certify that the pseudo-identity of a buyer is correctly registered to CA_R, and only CA_R knows about the real identity of the buyer.
- A Monitor MO functions as a trusted third party which is responsible for the generation of collusion-resistant fingerprint codes. The existence of MO ensures that the generated fingerprints are not revealed to M and the buyer. It also keeps the record of transactions between M and the buyer. MO is also responsible for executing traitor tracing algorithm in case of a piracy claim by M. In case of dispute resolution between M, a buyer, and J, MO provides the pseudo-identity of the accused buyer to J.
- A judge J is assumed to be a trusted party which resolves the disputes between M and a peer with the cooperation of MO and CA_R.
- A tail node T_A is a message transferring agent that manages anonymous communication on behalf of a peer A. Each peer within the P2P network has one such agent. The tail node forwards the query of a requesting peer to the providing peer through an anonymous path and returns the reply back to the requesting peer.

4.2 Design Requirements and Assumptions

In this section, the design requirements, general and security assumptions of the framework and threat model are described.

4.2.1 Design Requirements

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

- The merchant should be able to trace and identify an illegal re-distributor in case of finding a pirated copy with the help of a trusted party *MO*, *J* and CA_R.
- The scheme should be collusion resistant against a specific number of colluders *c* as specified by Nuida *et al.*'s codes (Nuida et al., 2007).
- The merchant should not be able to frame an honest buyer of illegal re-distribution.
- The buyer accused of re-distributing an unauthorized copy should not be able to claim that the copy was created by M.
- The identity of a buyer should remain anonymous during transactions until he/she is proven to be guilty of copyright violation.
- The identity of a peer should not be linked to his/her activities such as, purchasing, transferring of file and so on.
- The embedding process should be blind and the embedded fingerprint should be imperceptible and robust against common signal processing attacks.
- The real identity of a peer should be protected during authentication process thus enabling each peer to verify the authenticity of each other anonymously.
- None of the intermediary peers should know about the requesting peer's and source provider peer's identity or an item being exchanged. Thus, *SF* transfer between the requesting peer and responding peer must be secure.
- The data expands on conversion from a plain-text to an encrypted representation of signals due to the use of an additive homomorphic cryptosystem. The homomorphic encryption should be performed in such a way that the size of the encrypted base file remains small.

4.2.2 Assumptions

The design and security assumptions of the system are as follows:

- **General Assumptions:**

  - There are six major players involved: merchant *M*, buyer (peer P_i), super peer *SP*, monitor *MO*, certification authority CA_R and judge J.

  - Each entity is supposed to have a public key K_p and a private key K_s.

  - The real identity of each entity is validated by an external (offline) certification authority CA_{ext}. Thus, each entity has a public key certificate signed by CA_{ext}. CA_{ext} keeps track of all the identities to be sure that they remain unique and also to revoke an identity of a malicious entity. The generation of a public key certificate is a one-time process.

- Before joining the system, each buyer is authenticated by an internal certification authority CA_R of the system. CA_R validates the identity of a buyer from CA_{ext}. After successful verification, each buyer has a private key and a public key certified by CA_R. CA_R generates a random number r and shares it with an authenticated buyer for the generation of a pseudo-identity. Each peer can have multiple pseudo-identities.

- The reconstruction of the original file from *BF* and *SF* should be performed at the buyer's end. *BF* cannot be shared within the end users of the system.

- **Security Assumptions:**

  - The merchant and the buyer do not trust each other but they both trust the Monitor *MO*. Because of the anonymity of the embedding procedure, *MO* generates the collusion-secure fingerprints as this is the only party that is trusted by both *M* and the buyer to generate a valid fingerprint.

  - The SHA-1 function used in the system to generate unforgeable and verifiable pseudo-identities for each entity is secure and cannot be reversed.

  - The communication between the peers is anonymous due to the use of onion-routing within the system.

  - *SP* is selected on the basis of his/her reputation and resources. *SP*s that manage the content distribution activities honestly gain more reputation among peers and content providers. More peers shall connect with a well-reputed peer and get the intended data through that trusted *SP*.

4.2.3 Threat Model

This sub-section highlights an attack model for the proposed system related to the robustness of a watermark, resistance of a fingerprint against collusion attacks, buyer's security from malicious entities, and authentication attacks.

1. ***Watermarking Attacks:*** A watermarking scheme used for copyright protection must have a capability to survive attacks such as signal enhancement, geometrical operations and noise filtering. The watermark must be highly robust against these attacks such that the retrieved watermark unambiguously identifies the copyright owner. The robustness of a watermark can be evaluated by simultaneously considering watermark impairment and the distortion of the attacked content. An attack succeeds in defeating a watermarking scheme if it impairs the watermark beyond acceptable limits while maintaining the perceptual quality of the attacked data. Thus an effective attack handling is required during evaluation of watermarking techniques. The attacks on watermarking schemes are categorized into two groups: attacks on audio and attacks on video watermarks as below.

(a) ***Attacks on an Audio Watermark***

i. ***Re-quantization:*** The watermarked audio signal is re-quantized from original bit-rate down to half the bit-rate and then back to original number of bits/sample. An increased incoherent background noise is heard in the audio track due to the rounding errors produced by the re-quantization process.

ii. ***Re-sampling:*** Under this attack, watermarked audio signals are down-sampled and then up-sampled (or vice versa) back to its original sampling rate. This attack affects audibility and produces distortions especially in audio tracks carrying high frequencies.

iii. ***MPEG-*1 *layer* 3 *(MP3) Compression:*** MP3 compression compresses data by discarding some part of it. The watermarked audio signal can be compressed at the variable bit-rates (e.g. $256, 128, 64, 32$ kbps) and then decompressed back to the wave format. This attack reduces the file size but at the cost of lower sound quality. The lower the bit-rate, the lower is the sound quality.

iv. ***Additive White Gaussian Noise (AWGN):*** The AWGN attack adds an additive Gaussian noise of zero mean, constant variance, and controlled value of signal-to-noise ratio (SNR) to the watermarked signal. The SNR is a metric that determines the strength of this attack. An addition of noise to a signal results in quality degradation of that signal.

(a) ***Attacks on a Video Watermark***

i. ***Median Filtering:*** Under this attack, a window of $[N \times N]$ pixels is moved onto a watermarked signal. It returns the median pixel value in the moving window. The lower the value of N, a more smoother image is produced. Whereas, an increase in N's value considerably blurs the image.

ii. ***Re-sizing:*** In re-sizing, a watermarked signal is either re-sized to double or downscaled to half the size of its original size and is then reduced back to its original size. However, in downscaling an image to the desired size, there is a loss of information.

iii. ***H.*264 *Compression:*** H.264 compression is one of the common lossy compression attacks on a video content. With H.264 compression, there is a trade-off between video quality, processing cost of compression/decompression, and file size. This trade-off is determined by specifying a bit-rate.

iv. ***AWGN:*** Gaussian noise insertion is a signal processing attack in which amount of noise to be added into a signal is controlled by its mean, variance, and SNR value.

2. ***Collusion Attacks:*** Collusion attack is a challenge issue for digital fingerprinting. The main concern for a fingerprinting system is the resistance of a fingerprint to colluders' attacks. Collusion occurs when different buyers recombine their marked copies to obtain a new copy of the content such that they cannot be accused of copyright violation. The collusion attacks are defined as follows:

(a) ***Averaging Attack:*** In an averaging attack, attackers with a total of K fingerprinted copies of the same content collude together to produce a colluded version Y. The fingerprinted signals are typically averaged with an equal weight for each user. It can be defined mathematically as: $Y(i) = \frac{y_0(i)+y_1(i)+...y_{K-1}(i)}{K}$.

(b) ***Minimum Attack:*** Under this attack, the attackers create a copy Y whose $i^{th}(i = 1, 2, \dots, m)$ (m=length of a fingerprint) component is the minimum of the i^{th} components of the observed marked copies. Mathematically, it is defined as: $Y(i)$= $\min(y_0(i), y_1(i), \dots, y_{K-1}(i))$.

(c) ***Maximum Attack:*** The colluders create an attacked copy Y by considering maximum value of the i^{th} components of their individual marked copies. It can be defined mathematically as: $Y(i)$= $\max(y_0(i), y_1(i), \dots, y_{K-1}(i))$.

(d) ***Median Attack:*** In median attack, the attackers take median of the values of the corresponding components of individual marked copies to create a pirated copy Y. Mathematically, it is defined as: $Y(i)$ =median$(y_0(i), y_1(i), \dots, y_{K-1}(i))$.

3. ***Attacks on a Buyer:*** The following types of attacks are aimed to de-anonymize a buyer, and accuse an innocent buyer of illegal re-distribution of the purchased content.

(a) When the fingerprint is inserted solely by a merchant M, M may benefit from framing attacks on an innocent buyer. This attack is successful if M is able to prove to the judge J that illegal copies of the marked content belongs to a particular buyer even though a buyer has not bought this content, or has bought this content but did not distribute copies of it illegally.

(b) Different transactions carried out by a buyer with a same pseudo-identity are linkable to one another and an attacker could infer some private information of a buyer through data mining techniques.

(c) A malicious entity may try to find two different but real identities such that the two identities have the same pseudo-identity. It might then use one of the two identities to impersonate the buyer with the other identity.

4. ***Attacks on an a Authenticated Key Exchange (AKE) protocol:*** The following attacks are considered authentication attacks that allow attackers to exploit the authentication process.

(a) ***Man-in-the-Middle Attack (MIMA):*** In the AKE protocol between two parties (a sender and a receiver), an eavesdropper E may access and modify messages between these parties without either party knowing that the link between them has been exposed. E may attempt to authenticate by posing as the sender to the receiver and the receiver to the sender.

(b) ***Replay Attack:*** Under this attack, the attacker may attempt to eavesdrop and collect some previous proofs of a receiver, and then reuses this information at a later time in an attempt to falsely authenticate to the sender.

The security of the system against these attacks is discussed in Section 5.1.

4.3 Model

In this section, we first give an overview of the design of the proposed system, and then discuss its six key components.

4.3.1 Overview

Figure 1 shows the structure of the proposed P2P content distribution framework that contains six main entities: merchant, buyer, super peer, monitor, judge, and certification authority. These entities are involved in six key sub-protocols (fingerprint generation, *BF* and *SF* generation, distribution of *BF* and *SF*, traitor tracing and dispute resolution) of the system.

The proposed scheme as shown in the Figure 1 consists of the following sub-protocols:

4.3.2 Generation of a Fingerprint

A variation of Tardos codes, i.e. Nuida's *et al.* codes (Nuida et al., 2007), are used for fingerprint generation. The algorithm for the fingerprint generation takes parameters ε, N and c as inputs, and outputs a collection $F = (f_1, \dots, f_N)$ of binary codewords f_i of size m and a secret vector p, as shown in Algorithm 1. The codeword f_i is meant to be embedded into a content of a user I.

Algorithm 1 Fingerprint Generation

```
procedure NUIDA et al.'s CODES
Input parameters: $c$, $N$ ($N \geq c$ ), $\varepsilon$
Output parameters: $f_i$, $p$
begin
    $m \leftarrow (c^2 K \log(N/\varepsilon))$                          ▷ where value of $K$ is 4.245
Select $p_i$ independently by picking uniformly at random for all 1≤ $j \leq m$
    for all $1 \leq i \leq N$ do                                  ▷ a loop over all users
        for all $1 \leq j \leq m$ do             ▷ a loop over the bits of the codeword of a $j^{th}$-user
            $P(u_{i,j} = 1) \leftarrow p_j$
            $P(u_{i,j} = 0) \leftarrow 1 - p_j$              ▷ with probability 1/2 each to get $W_{N \times m}$
        end for
    end for
    return $F$, $p$        ▷ Fingerprint $F = (f_{i,j})$ where $i \in [N]$, $j \in [m]$ and secret vector $= (p_j)_{j \in [m]}$
end procedure
```

4.3.3 Generation of the Base and Supplementary Files

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 system supports both audio and video files. Figure 2 illustrates the process

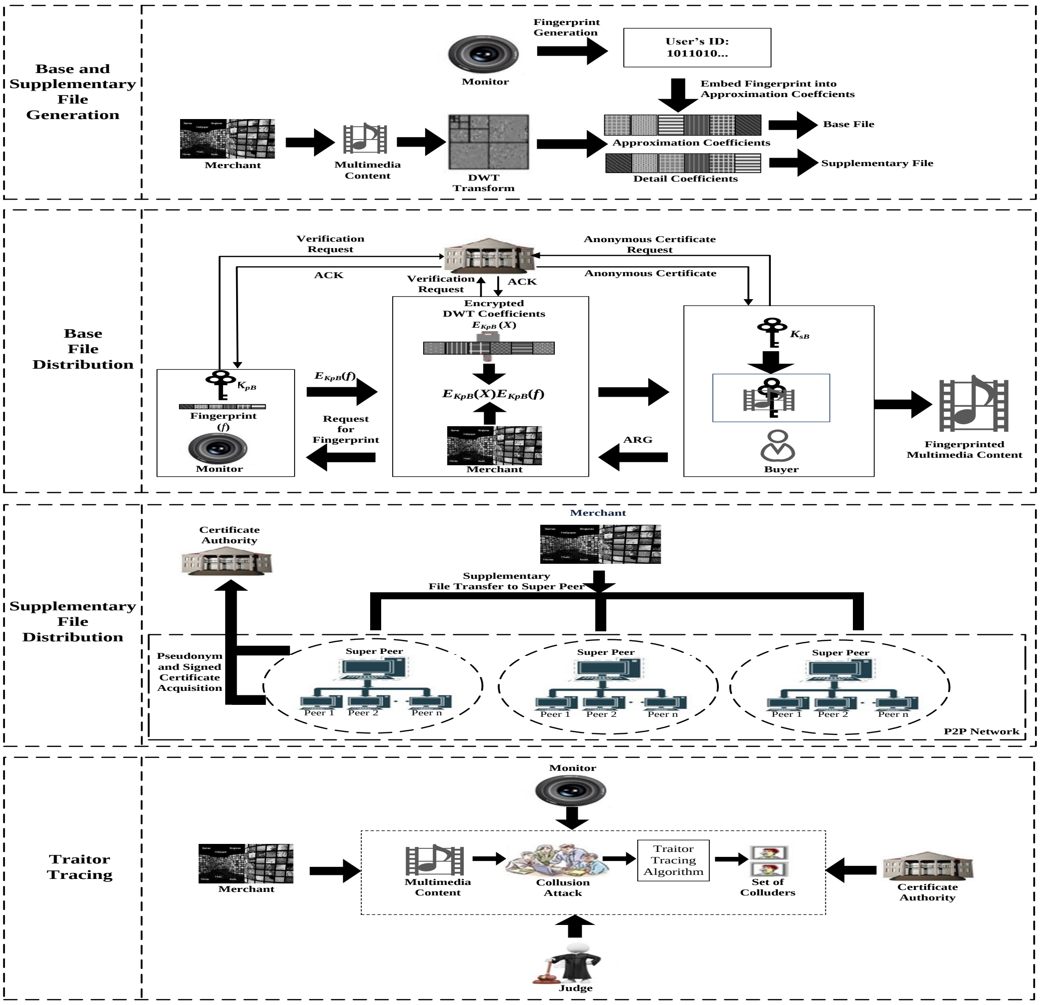


Figure 1: Secure and Anonymous P2P Content Distribution System

of generation of *BF* and *SF* for both audio and video files. The proposed method employs the Discrete Wavelet Transform (DWT) to split the content into low-frequency (approximation coefficients) and high-frequency (detail coefficients) components. An approximation coefficient is then itself split into a second-level approximation and detail coefficients, and the process is repeated as many times as desired (levels of decomposition). In order to embed an encrypted fingerprint in the encrypted approximation coefficients for formation of *BF*, the additive homomorphic property of public-key cryptosystems is applied. However, additive homomorphic cryptosystems cannot work on real-valued DWT coefficients. Therefore, these approximation coefficients are quantized to integer values.

In the quantization process, approximation coefficients are quantized to the nearest even/odd integers depending on the value of quantization step size Δ. The quantizer Δ is a positive integer such that all the quantized coefficients are encrypted. Then, all the quantized approximation coefficients are encrypted with the public key of the buyer. Several quantized coefficients are selected using a secret key *sk* to embed the fingerprint bits. The same key is used to extract the fingerprint from the re-distributed copies. To embed a single bit of a Nuida's *et al.* codeword into one of the selected quantized and encrypted coefficient, the following operation (Prins et al., 2007) is performed:

$$E(y_i) = \begin{cases} E(d \cdot (Q_{2\Delta}(x_i + d_i) - d_i)) \times E(f_i)^{\Delta} & \text{if } x_i \geq Q_{2\Delta}(x_i) \\ E(d \cdot (Q_{2\Delta}(x_i + d_i) - d_i)) \times (E(f_i)^{\Delta})^{-1} & \text{if } x_i < Q_{2\Delta}(x_i) \end{cases}$$

In the selected approximation coefficients, a small amount of dither is added prior to quantizing the coefficient x_i to an even and odd value depending on the fingerprint bit f_i. After quantization, the same amount of dither is subtracted from the quantized values. These coefficients are then scaled by a factor d before doing an encryption. The remaining approximation coefficients are quantized, scaled and encrypted block-by-block with each message block $m_i < N$, to get a reduced *BF* size. At the user end, the buyer decrypts *BF* using his/her private key. d is also communicated to the buyer so that he/she can rescale the entire content after decryption. Then, an inverse DWT is performed with the detail coefficients received via *SF* to get a fingerprinted content.

In case of an audio file, the DWT decomposition results in approximation and detail coefficients. The L-level approximation coefficients are then divided into m non-overlapping frames. All frames except one are quantized, scaled and encrypted. In the frame, chosen by M to embed the fingerprint bits, a dither value is added to coefficient x_i, quantized, the same dither value is subtracted, scaled and then encrypted with a buyer's public key. Then, the encrypted fingerprint is added to the selected encrypted approximation coefficients. The frames are recombined and saved in a "text" format as *BF*. An inverse L-level DWT is performed on the detail coefficients to get *SF* in "wav" form. Other formats, such as binary and text, can also be used for the formation of *SF*.

The first step for generation of video *BF* and *SF*, is to extract the significant frames from a video file since not all the frames contain important information. The video frames are arranged into groups of pictures (GoPs). A GoP includes the Intra frames (I-frames) and Inter-frames (P and B-frames). The I-frames are coded without reference to other frames, whereas P and B-frames use pseudo-differences from previous and next frame, so these frames depend on each other. It is not meaningful to analyze both Intra and Inter-frames, thus we find Intra-frames only which contain important information. In order to find I-frames, we have used the Canny Edge Difference technique (CEDT) (Khurana & Chandak, 2013). In CEDT, a difference between two consecutive frames is calculated and if this difference exceeds a calculated threshold value, we obtain a key frame. The remaining frames, i.e. P and B-frames are saved in an original video format. The key frames of the video are converted from RGB format to Y′UV. The Y′UV model defines a color space in terms of one luminance (Y′) and two chrominance (UV) components. The weighted values of R, G and B are added together to produce a single Y′ (luminance) component. The chrominance components U and V are created by subtracting Y′ from B and Y′ from R, respectively. For each key frame, we choose the Y′ component and apply L-level DWT to obtain the approximation and detail coefficients. A few key frames are selected for embedding the fingerprint. These frames are selected on the basis of time, i.e. one key frame is selected after a duration of 30s. Then, the L-level approximation coefficients of the selected key frames are divided into m non-overlapping blocks. All other frames except for the selected frames, are quantized, scaled and encrypted. In the selected frames, a dither value is added to a coefficient x_i, quantized, the same dither value is subtracted, scaled and then encrypted with a buyer's public key. Then, the encrypted fingerprint obtained from MO is added to the encrypted approximation coefficients to form BF in "text" form. BF can also be saved in other formats, such as binary, and bitmap (.bmp) image files. The index of the key frames is also scaled, encrypted and added into BF for file reconstruction at the user end. An inverse L-level DWT is applied on the detail coefficients and then these obtained values, P and B-frames and audio of the original video file constitute SF in a compressed (.zip) form.

We have chosen level-3 or 4 DWT decomposition for our design to achieve a convenient trade-off between the robustness, capacity and transparency properties of watermarking. The fingerprint f_i of length m is then inserted into a selected quantized and encrypted DWT level-3 or 4 coefficients, using SD-QIM watermarking technique. The remaining quantized and encrypted approximation coefficients and the embedded coefficients constitute BF which is sent to the end user. The remaining detail coefficients constitute SF which is sent to SP for its distribution in the P2P system. Eventually, the system at the user end can decrypt the base file using his/her private key K_{S_i}, rescale by a factor d and apply the inverse DWT to get a fingerprinted copy.

4.3.4 Distribution of a Base file

On receiving a file request from a peer (buyer) B_i, SP provides him/her the details of the merchant that has the requested content. In order to get a content X from M, the buyer B_i follows the following protocol.

(1) The buyer negotiates with M to set-up an agreement (AGR) that explicitly states the rights and obligations of both parties and specifies the content X. AGR uniquely binds this particular transaction to X. During the negotiation process, B_i uses his/her pseudonym P_{B_i} to keep his/her anonymity.

(2) After the negotiation, B_i generates a key pair $(K^*_{pB_i}, K^*_{sB_i})$, signs the public key with his/her private key, and sends $\mathrm{Sign}_{\mathrm{B_i}}(K^*_{pB_i}, P_{B_i})$ to CA_R. CA_R verifies $\mathrm{Sign}_{\mathrm{B_i}}(K^*_{pB_i}, P_{B_i})$ using the public key of B_i. If valid, he/she generates an anonymous certificate $\mathrm{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$ and sends it to B_i. B_i then sends $\mathrm{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, AGR, P_{B_i} and $\mathrm{Sign}_{K^*_{pB_i}}(AGR)$ to M.

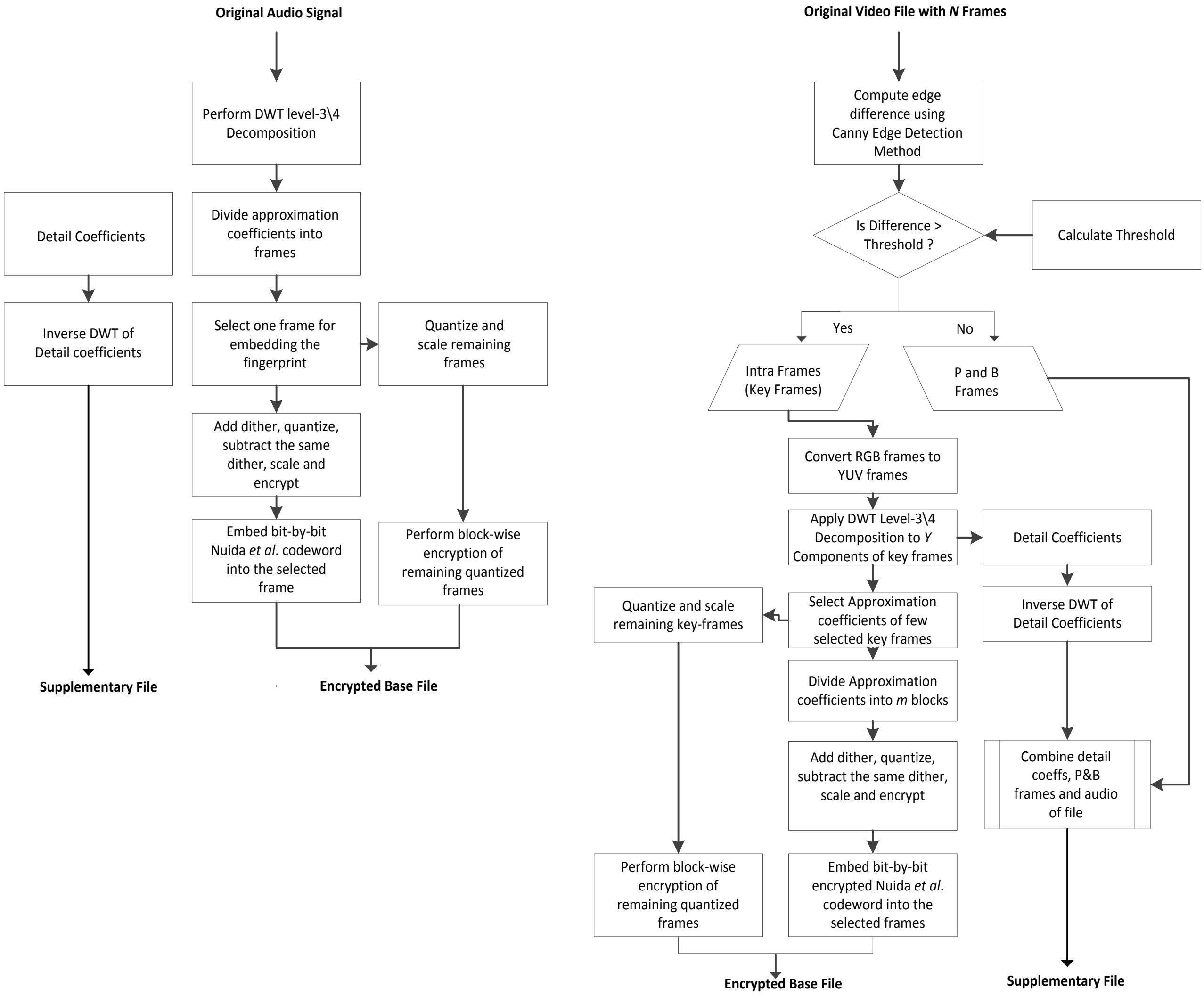

Original Audio Signal
Perform DWT level-3\4 Decomposition
Detail Coefficients
Divide approximation coefficients into frames
Inverse DWT of Detail coefficients
Select one frame for embedding the fingerprint
Quantize and scale remaining frames
Add dither, quantize, subtract the same dither, scale and encrypt
Embed bit-by-bit Nuida et al. codeword into the selected frame
Perform block-wise encryption of remaining quantized frames
Supplementary File
Encrypted Base File
Original Video File with N Frames
Compute edge difference using Canny Edge Detection Method
Is Difference > Threshold ?
Calculate Threshold
Yes
No
Intra Frames (Key Frames)
P and B Frames
Convert RGB frames to YUV frames
Apply DWT Level-3\4 Decomposition to Y Components of key frames
Detail Coefficients
Quantize and scale remaining key-frames
Select Approximation coefficients of few selected key frames
Inverse DWT of Detail Coefficients
Divide Approximation coefficients into m blocks
Add dither, quantize, subtract the same dither, scale and encrypt
Combine detail coeffs, P&B frames and audio of file
Perform block-wise encryption of remaining quantized frames
Embed bit-by-bit encrypted Nuida et al. codeword into the selected frames
Encrypted Base File
Supplementary File

Figure 2: Flowchart for Generation of *BF* and *SF* of Audio and Video Files

(3) M verifies the received certificate using CA_R's public key, and the signature of the agreement using the certified key. If the received data is valid, then M generates a transaction ID (TID) for keeping a record of the transaction between him/her and B_i, and sends a request for a fingerprint to MO by sending $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Cert}_{CA_R}(M)$, TID, AGR, P_{B_i} and $\text{Sign}_{K^*_{pB_i}}(AGR)$. If the received certificates and signatures are not valid, then the transaction is terminated by M.

(4) MO validates the certificates and signatures of M and B_i from CA_R. After successful verification, MO generates a Nuida's c-secure codeword f against a TID sent by M. MO then sends $E^*_{KpB_i}(f)$ and $\text{Sign}_{MO}(E^*_{KpB_i}(f)$, $K^*_{pB_i}$, $\text{Sign}^*_{K_{pB_i}}(AGR))$ to M. MO stores $K^*_{pB_i}$, $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Cert}_{CA_R}(M)$, AGR, $\text{Sign}_{K^*_{pB_i}}(AGR)$ and $E^*_{K_{pB_i}}(f)$ against TID.

(5) After receiving the encrypted fingerprint from MO, M inserts the fingerprint code into the encrypted domain by using the embedding process as described in Section 4.3.3 without knowing about the plain-text fingerprint f.

(6) M sends $E^*_{K_{pB_i}}(BF)$ to B_i and stores $K^*_{pB_i}$, $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$,$AGR$, $\text{Sign}_{K^*_{pB_i}}(AGR)$, $E^*_{K_{pB_i}}(f)$ and $\text{Sign}_{MO}(E^*_{K_{pB_i}}(f)$, $K^*_{pB_i}$, $\text{Sign}^*_{K_{pB_i}}(AGR))$ against TID.

(7) B_i decrypts $E^*_{K_{pB_i}}(BF)$ with $K^*_{sB_i}$ and obtains a fingerprinted BF.

4.3.5 Distribution of a Supplementary File

n receiving a request for X from B_i, SP searches for in his/her own file index. If not found, he/she then searches within his/her group of peers. If the particular content is found within the group, he/she displays the list of the buyers (peers) having that particular file, and also displays their tail nodes to act as middle nodes between the content providing CP peer and the requesting peer RP. If SP is unable to find the file within his/her group, he/she sends a request for the file to other connected SPs. The other SP, on finding the particular CP, sends the response to the requesting SP. SP then establishes a path between RP and that CP. On joining the system, peers construct anonymous paths with existing peers using the APFS protocol (Scarlata et al., 2001). The joining peer learns from the bootstrapping node the set of other peers available for forming anonymous paths. Each peer constructs an onion path pointing to another peer called, tail node, which acts as an anonymous message relaying agent. Each peer posts his/her tail node to SP. By doing so, RP can use the anonymous path to contact CP while knowing nothing about CP's identity.

For simplicity, we assume that P_a is a requesting peer and P_b is the providing peer. When P_b receives the file request, if he/she holds the requested file and decides to be the file provider, he/she replies to the query through his/her tail-node T_b. 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, $P_a \rightarrow T_a \rightarrow T_b \rightarrow P_b$. An anonymous two-party AKE protocol is established between P_a and P_b. P_a sends an authentication request to P_b. P_b sends a challenge message to P_a and verifies P_a. Figure 3 describes the authentication process between P_a and P_b. In step 1 of authentication, P_a chooses $a \in [1, \ldots, P-1)$ randomly. Then he/she uses his/her private key $K_{S_{Pa}}$ to sign $\{ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, a\}$. P_a computes his/her pseudo-identity by using a hash function. PI_{P_a} is given as, $PI_{P_a} = h(ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, a, \text{Sign}_{P_a}\{ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, a\})$

In step 1, P_a also computes g^{x_1} with publicly known parameters P and g for generation of a session-key. x_1 is chosen randomly from $[1, \ldots, Q)$ to generate a session-key. g^{x_1} is calculated as: $g^{x_1} := g^{x_1} \bmod P$. To send an authentication request to P_b, P_a calculates u as, $u = h(PI_{P_a}, a, g^{x_1})$ where, h is a hash function with k bits and is defined as: $h = Z_n^* \times \{0,1\}^w \times Z_p^* \rightarrow [0,1]^k$. P_a sends $\{PI_{P_a}, a, g^{x_1}\}$ to P_b. In step 2, after receiving the authentication request, P_b computes u'=$h(PI_{P_a}, a, g^{x_1})$ to verify the authentication request. Once verified, P_b randomly chooses a number $b \in$

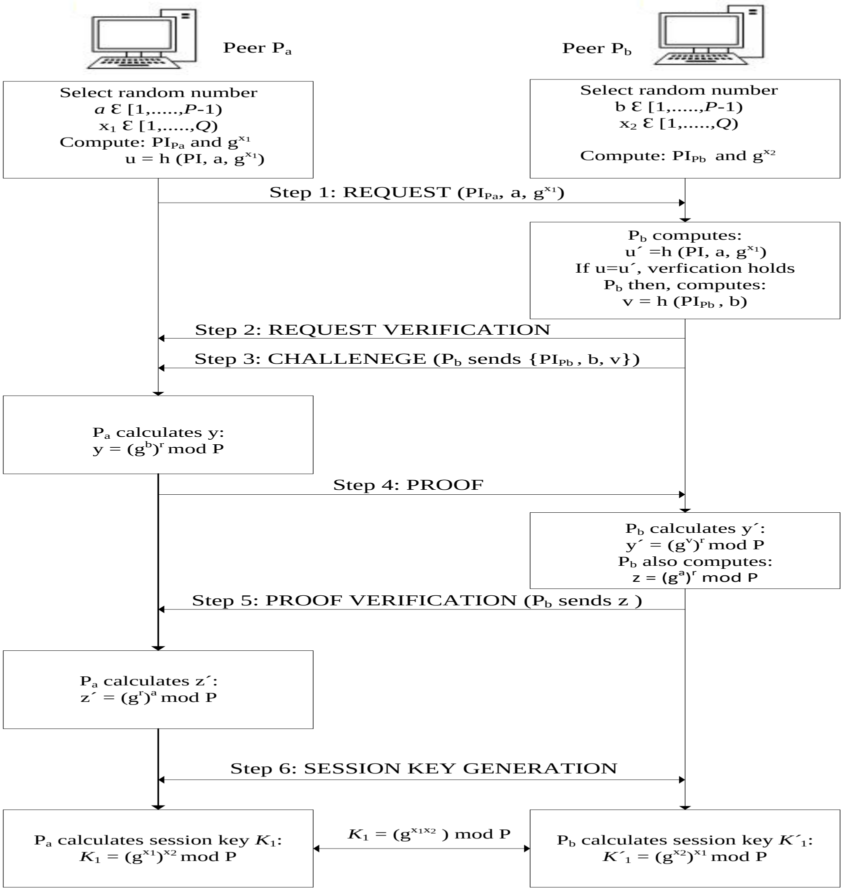


Figure 3: Two-Party Anonymous AKE Protocol

$[1, \ldots, P-1)$. Then he/she uses his/her private key $K_{S_{Pb}}$ to sign $\{ID_{P_b}, \mathrm{Cert}_{CA_R}(P_b), r, b\}$. P_b computes his/her pseudo-identity by using a hash function. PI_{P_b} is given as, $PI_{P_b} = h(ID_{P_b}, \mathrm{Cert}_{CA_R}(P_b), r, b, \mathrm{Sign}_{P_b}\{ID_{P_b}, \mathrm{Cert}_{CA_R}(P_b), r, b\})$. In this step, P_b also chooses a number $x_2 \in [1, \ldots, Q)$ randomly and computes g^{x_2} for generation of a session-key. g^{x_2} is calculated as: $g^{x_2} := g^{x_2}$ mod P. P_b computes $v = h(PI_{P_b}, b)$ and then, sends $\{PI_{P_b}, b, v\}$ as a challenge to P_a. As a proof, P_a calculates y=$(g^b)^r$ mod P and send it to P_b. As a proof verification, P_b calculates y'= $(g^r)^b$ mod P. If the verification holds, P_b sends $z = (g^a)^r$ mod P to P_a. P_a then computes z'= $(g^r)^a$ mod P to complete the last step of authentication. When the authentication is successfully completed, P_a computes $K_1 = (g^{x_2})^{x_1}$ mod P, and P_b computes $K_1 = (g^{x_1})^{x_2}$ mod P. Clearly, we have $K_1 = K_1^{'} = g^{x_1 x_2}$ mod P. P_a and P_b can therefore, use K_1 as their session-key for encryption of SF. P_b encrypts SF using the session key K_1 and sends E_{K_1} (SF) to P_a through T_b and T_a. P_b decrypts E_{SF} with K_1 and gets a decrypted SF. For security purposes, the complete communication between SP and the peers is stored in MO. For the secure exchange of transaction record (P_a, P_b, $h(SF)$, SP) to MO, a session key K_{ses} is generated using the two-party AKE. SP transfers $E_{K_{ses}}$ $(P_a, P_b, h(SF), SP)$ to MO. At the completion of transfer, MO concatenates all the $h(SF)$s and stores the concatenated hash, P_a and P_b against a specific transaction ID.

4.3.6 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 approximation coefficient matrix in which the pirated code $pc \in \{0, 1\}^*$ is embedded. The code is extracted by using the secret key sk that was used to specify the embedding position. Each approximation coefficient in the embedding position is quantized using the corresponding quantization step size Δ. If the value is even, the information bit is regarded as 0, else 1. After the extraction of pc, M sends $(pc, M, E_{K_{S_M}}(sk), TID)$ to MO. MO performs the tracing algorithm of Nuida *et al.*'s codes as described in Algorithm 2 to identify the colluder(s). In the tracing algorithm, pc provided by M and a bias vector p are given as inputs. p is used to generate the fingerprint matrix for identification of the colluder(s). The score of the pirate is calculated as per Algorithm 2. The output of this tracing algorithm is a user with the highest score. The real identity of a user is not known to MO, only the pseudo-identity of the guilty buyer is revealed. MO retrieves a TID that contains the fingerprint f from his/her database for arbitration and identification protocol.

Algorithm 2 Traitor Tracing

```
    procedure NUIDA et al.'s TRACING CODE
    Input parameters: $pc$, $p$, $F_i$
    Output parameter: $S_i$
    begin
        $W_{Al} \leftarrow DWT(Y)$                                   ▹ Apply Level-l DWT to $Y$
        $pc \leftarrow Detection(W_{LL-l})$      ▹ Apply watermarking detection technique on $W_{LL-l}$ to extract $pc$
        $\sigma = \sqrt{\frac{1-p}{p}}$                  ▹ Calculate the score $S_i$ of the pirate using $(\sigma(p)$
        if ($pc = 1$ and $F_{i,j} = 1$) then
            $S_i^j \leftarrow \sigma(p^{(j)})$
        else if ($pc = 1$ and $F_{i,j} = 0$) then
            $S_i^j \leftarrow -\sigma(1 - p^{(j)})$
        else if ($pc \in \{0, ?\}$ and $F_{i,j} = 1$) then
            $S_i^j \leftarrow -\sigma(p^{(j)})$
        else if ($pc \in \{0, ?\}$ and $F_{i,j} = 0$) then
            $S_i^j \leftarrow \sigma(1 - p^{(j)})$
        end if
        return $S_i^j$
    end procedure
```

4.3.7 Arbitration and Identification

The goal of the arbitration and identification protocol, performed between M, MO, CA_R and J, is to reveal the real identity of the traitor or reject the claims made by M. In order to reveal the real identity of the traitor, MO sends (Y, pc,$\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, AGR, $\text{Sign}_{K^*_{pB_i}}(AGR)$, $E^*_{K_{pB_i}}(f)$, $E_{K_{P_{MO}}}(f)$) and $\text{Sign}_{MO}(E^*_{K_{pB_i}}(f)$, $K^*_{pB_i}$, $\text{Sign}_{K^*_{pB_i}}(AGR)$ to J. J

verifies the validity of all the certificates and the signatures. If valid, he/she then asks MO to decrypt $E_{K_{pMO}}(f)$. MO decrypts $E_{K_{pMO}}(f)$ and sends f to J. If pc and f match with a high correlation, it requests CA_R to give the real identity of the buyer. Otherwise, the buyer is proved innocent.

4.3.8 Cloud Computing Implementation

The development of cloud computing platforms enables business owners and consumers to outsource computations on their data, and allow businesses to offload the task of maintaining data-centers. However, the adoption of cloud services by consumers and businesses is limited by concerns over the loss of privacy or business value of their private data. An excellent way to alleviate this privacy concern is to store all data in an encrypted form in the cloud, and perform computations on the encrypted data. Homomorphic encryption can be utilized in this scenario to help preserve customer privacy while, outsourcing various kinds of computation to the cloud. In our base file distribution scenario, the merchant can provide the selected encrypted approximation coefficients, remaining encrypted approximation coefficients, scaling factor c, and a collusion-resistant fingerprint f_i to the cloud, which can perform the fingerprint embedding process into the encrypted approximation coefficients, and block-by-block encryption of the remaining coefficients to constitute a small-size BF. In this way, the expensive part of the protocol, i.e. embedding a fingerprint in an encrypted domain, can be processed by the cloud. Thus, cloud computing can provide a valuable service to the merchant by enabling him/her to avoid the computational cost due to use of homomorphic encryption. Similarly, for supplementary file distribution, we can consider P2P cloud-III (Poh et al., 2013), which is a cloud system that is fully decentralized and is considered to be public. The main objective of P2P cloud-III is to avoid a single point of failure with distributed servers. This is more of a hybrid system, whereby a service provider maintains a set of fully decentralized resource hosts that may act as provisioning servers. In this setting, when a server is down, users can still request for services through other resource hosts. In addition, since P2P networks form the basis of a P2P cloud according to (Poh et al., 2013), we can utilize the authentication and anonymous communication mechanisms used in our proposed system to the P2P cloud proposal. Since BF distribution protocol is a one-to-one protocol between a merchant and the buyer, its execution remains the same in cloud computing scenario as in the proposed system. The monitor in our system can use a cloud service to generate a collusion-resistant fingerprint and perform traitor-tracing. For dispute-resolution, we can include a public cloud C that can be shared between a judge, a certification authority, a merchant and a monitor.

5 Results and Discussions

In this section, we provide an analysis of the proposed framework in terms of security and performance.

5.1 Security Analysis

In this section, we analyse the security of the proposed framework and explain how it fulfils the design requirements.

5.1.1 Traceability

Once a pirated copy is found, the traitor tracing algorithm of Nuida *et al.*'s codes is used by M to trace the copyright violator with the help of MO. The traitor tracing algorithm employs a scoring technique that outputs a guilty user with the highest score. Once the algorithm outputs a guilty user, his/her identity is revealed by J with the help of CA_R.

5.1.2 Collusion Resistance

Nuida *et al.*'s codes are c-secure with ε-error with $l \leq c$ (l is the number of pirates). In our design, we have considered c =3 with $\varepsilon = 10^{-3}$ and $N = 10^6$ (N = number of users), thus we have obtained a code of size $m = 354$ bits. This code is then embedded into the content to uniquely identify the user. As long as l remains lower than c, the piracy tracing algorithm (Nuida et al., 2007) is followed, the copyright violator can be identified successfully. Thus, the proposed scheme offers resistance against three colludes. The value of $c > 3$ can also be considered. But this big value of c results in increased length m of the codeword, that will provide high collusion resistance but at a cost of lower content quality. The value of c is decided keeping in mind the desired security level of the system.

5.1.3 Buyer Security

M knows only about $E^*_{K_{pB_i}}(f)$ and $E^*_{K_{pB_i}}(BF)$ and has no knowledge about the buyer's private key $K^*_{sB_i}$. Therefore, M does not know about the fingerprinted copy that B_i gets after decrypting $E^*_{K_{pB_i}}(BF)$ with $K^*_{sB_i}$. It means that M cannot frame B_i by distributing forged copies of the content. Furthermore, $\text{Sign}_{MO}(E^*_{K_{pB_i}}(f), K^*_{pB_i}, \text{Sign}^*_{K_{pB_i}}(AGR))$ explicitly binds f to AGR, which specifies the content X. Thus, it is impossible for M to frame B_i. Also, B_i generates a one-time anonymous key-pair $(K^*_{pB_i}, K^*_{sB_i})$ for the transaction with M that prevents M to frame B_i by sending $E^*_{K_{pB_i}}(f)$ from previous transactions. Therefore, framing an honest buyer by M is not possible since he/she cannot forge any evidence.

5.1.4 Merchant Security

From the perspective of M, the proposed framework is secure and fair because a buyer has no idea about the original content and the embedded fingerprint in the received copy. The buyer cannot claim that a pirated copy is created by M because only B_i can decrypt the $E^*_{K_{pB_i}}(f)$ with his/her $K^*_{sB_i}$). Also, MO is an entity trusted by both B_i

and M, thus B_i cannot accuse MO of collaborating with M to frame him/her. Moreover, the fingerprint is embedded into the selected positions of the content. Thus, a probability to find the exact location of the embedded fingerprint is quiet low. The proposed framework also provides mechanisms to unambiguously identify a copyright violator once a pirated copy is found.

5.1.5 Unlinkability

Despite the fact that anonymous certificates provide anonymity to B_i, the transactions carried out by the same pseudo ID ($K^*_{sB_i}$) can be linked to one another. The solution to this problem is to allow a user to apply for multiple pseudonyms and anonymous certificates simultaneously and randomly chooses one for each transaction. In the distribution phase of SF, each peer uses a random number to perform two-party AKE protocol. Thus, other peers can not guess about an identity of a peer with whom these numbers could be associated.

5.1.6 Anonymity

The anonymity of a peer's identity is obtained using a one-way cryptographic hash function h. This hash function provides a pseudo ID which can be used for anonymous authentication and communication. An attempt of de-anonymization attack by a malicious peer is withstood by the collision resistance of the hash function, i.e. it is computationally infeasible to find a pair (x, y) such that $h(x) = h(y)$. Moreover, for a hash function with w-bit hash values, $2^{w/2}$ calculations are required to find a collision with probability $1/2$, which is infeasible for $w \geq 128$. In our design, we have considered SHA-1 with $w = 160$ bits for high security such that it is computationally infeasible for an attacker to compute 2^{80} calculations to find a real identity from a pseudo ID. Furthermore, a malicious peer cannot use pseudo ID of other peer because it does not know the secret number r shared by the peer with CA_R.

5.1.7 Man-in-the-Middle Attack

Our system defends against MIMA by making use of ZKPI based authentication. In the authentication step, the proof, tail node's information and the exchanged data are bound together with a peer's pseudo ID. By doing so, any attempt by an attacker to modify the identity messages would not pass the verification of genuine protocol participants. For formal and detailed proof, we refer readers to reference (Lu et al., 2008).

5.1.8 Replay Attack

For a replay attack, to convince the providing peer about his/her identity, a malicious peer E needs to guess the secret number r correctly. However, the probability of a malicious peer's guessing correctly a secret number is 1¶ (where P is the element of finite cyclic group $\mathbb{G}$). Because it is computationally infeasible for discrete logarithms problem, E can not compute r. Thus, E can guess a secret number r' by computing $x = (g^b)^{r'} \bmod P$. However, the probability of soundness that E guesses r is $1/P$, i.e. the probability that E succeeds is $1/P$.

5.2 Performance Analysis

Six kinds of experiments including execution times of file partition into *BF* and *SF* files, calculation of transparency (Objective Difference Grade (ODG) and Peak Signal-to-Noise Ratio (PSNR)) of fingerprinted audio and video files, evaluating robustness of fingerprint against signal processing attacks, determining collusion resistance of a fingerprint against collusion attacks, response time, and cryptographic overhead costs, have been performed to show the efficiency of the proposed P2P content distribution framework.

5.2.1 Execution Time of File Partition into Base and Supplementary Files

To show the performance of our system, we have carried out experiments in Matlab, Java and C++ on three audio and three video files with varying sizes, on a workstation equipped with Intel i-5 processor at 3 GHz and 4 Gbyte of RAM. Table 3 presents the details of the audio and video files. To create *BF* and *SF*, experiments have been conducted in Matlab in which DWT is used to decompose the original file into approximation and detail coefficients. The embedding of the fingerprint in the encrypted approximation coefficients is implemented in Java. The distribution phase of *BF* and *SF* is executed in the C++ programming language.

Table 3: Details of Audio and Video Files

	Audio Files				**Video Files**		
	Loopy Music	**Huge Wave**	**Aasan Nai Yahan**		**Traffic**	**Dragon**	**Breaking Bad**
Time Length (min:sec)	00:10	00:17	03:34	Time Length (min:sec)	00:10	23:00	50:00
Size of File (MB)	0.89	2.97	36.01	Size of File (MB)	0.19	51.10	305.00
Format	WAV	WAV	WAV	Format	AVI	AVI	MP4
Bits per Sample	16	16	16	Resolution	120 × 160	320 × 240	720 × 406
Sample Rate(Hz)	44100	44100	44100	Total Frames	120	32975	67817
Channel Mode	Mono	Stereo	Stereo	Key Frames	15	2228	2649

For embedding a fingerprint into approximation coefficients, SD-QIM watermarking scheme (Prins et al., 2007) with $\Delta = 0.5$ and $c = 5$ is used. The size of keys for encryption and decryption is chosen to be 1024-bits. The selected approximation coefficients are encrypted bit-by-bit for secure embedding. The remaining coefficients are encrypted block-by-block to reduce the size of the encrypted *BF*. Thus, the modified and unmodified encrypted coefficients form a *BF*. *SF* is formed with double-bit precision values since Matlab stores signals as double-precision values and if it is not saved in a double-bit format, the file reconstruction at the user end would not be perfect due to quantization error. For audio files, level-3 DWT decomposition is implemented with 4-coefficient Daubechies (*db*4) filter. Similarly, level-4 DWT with 4-coefficient Daubechies (*db*4) filter is applied on Y' component of each key frame in a video file. For block-by-block encryption, the size of a each block is chosen to be less than N (modulo N). The experimental results in Table 5 and 7 presents imperceptibility and robustness results of the files against unauthorized attacks.

Table 4 summarizes the results of a file partition process for the selected audio and video files. The execution time of audio fingerprinting involves fingerprint generation, DWT, key generation, encryption of approximation coefficients, fingerprint embedding, and inverse DWT of detail coefficients for *SF* creation. For video fingerprinting, the execution time involves key frames conversion from RGB to Y′UV conversion, fingerprint generation, DWT decomposition, key generation, encryption of approximation coefficients, fingerprint embedding, inverse DWT of detail coefficients, P & B frames conversion to original video format and an audio extraction to create *SF*.

Table 4: Details of Base and Supplementary Audio and Video Files

	Audio Files				**Video Files**		
	Loopy Music	**Huge Wave**	**Aasan Nai Yahan**		**Traffic**	**Dragon**	**Breaking Bad**
Original File Size (MB)	0.89	2.97	36.01	Original File Size (MB)	0.19	51.10	305.00
Base File Size (MB)	0.52	0.88	9.80	Base File Size (MB)	0.08	9.21	11.80
Supplementary File Size (MB)	1.79	5.94	72.16	Supplementary File Size with Double-Bit Precision (MB)	0.18	69.4	215.90
Execution Time (secs)	20.13	37.34	188.60	Execution Time (secs)	24.00	98.40	112.20

5.2.2 Transparency of Fingerprinted Audio and Video Files

Table 5 presents the imperceptibility results as ODG and PSNR of both fingerprinted audio and video files, respectively. The ODG is a measurement of an audio distortion and is assumed to provide an accurate model of the subjective difference grade results. The ODG results are obtained by the Opera (*Opera*, 1999) software. In all cases of audio files, the ODG values are between 0 (not perceptible) and -1.2 (not annoying), showing better behaviour in terms of the imperceptibility. For video files, the quality is determined by the PSNR of the fingerprinted video. The PSNR provides a reliable indication of the variation of subjective video quality in decibels. The PSNR values are obtained by using MSU Video Quality Measurement Tool (*MSU Video Quality Measurement Tool*, 2011). The PSNR is above 40 dB in each case, and thus it can be inferred that the embedded fingerprint has no perceptible effect on the quality of the video file.

5.2.3 Robustness of Fingerprinted Audio and Video Files

Table 6 presents the robustness results of an audio file "LoopyMusic.wav" against signal processing attacks such as re-quantization, re-sampling, MP3 compression and Additive White Gaussian Noise (AWGN). The bit error rate

Table 5: ODG and PSNR Values of Audio and Video File

Audio Files	ODG	Video Files	PSNR
Loopy Music	-0.48	Traffic	44.00 db
Huge Wave	-0.98	Dragon	42.00 db
Aasan Nai Yahan	-1.2	Bad	41.00 db

(BER) and normalized correlation (NC) are used to evaluate the robustness between the original fingerprint and the extracted fingerprint. Similarly, Table 6 also presents BER and NC values of a video file "Dragon.avi" tested for unauthorized attacks such as median filtering, resizing, H.264 compression and AWGN. The BER values closer to zero indicates robustness against signal processing attacks. In case of NC, if NC is close to 1, then the similarity between f and $f^{'}$ is very high. If NC is close to 0, then the similarity between f and $f^{'}$ is very low. The results in Table 6 shows that the selected embedding algorithm (Prins et al., 2007) have good NC and BER values against various attacks for "LoopyMusic.wav" and "Dragon.avi". These results indicate that our fingerprint embedding algorithms satisfies the fingerprint's robustness requirement. The minimum BER and the maximum BER values for "LoopyMusic.wav" are 7% and 13% respectively against different attacks. Similarly for "Dragon.avi", the minimum BER value is 6% and the maximum BER value is 14%. The NC values in almost all the cases are in the range $0.856 - 0.972$, thus indicating close similarity between the original and retrieved fingerprints.

Table 6: Robustness of Audio and Video Files against Signal Processing Attacks

Audio				Video			
Attacks	Parameters	BER	NC	Attacks	Parameters	BER	NC
Re-quantization	16 - 8 -16	0.07	0.951	Median Filter	$[3 \times 3]$	0.09	0.912
Re-sampling	44100 - 22050 - 44100	0.11	0.902	Re-sizing	$320 - 640 - 320$	0.06	0.972
MP3 Compression	256 kbps	0.09	0.912	H.264 Compression	768 kbps	0.09	0.912
AWGN	18	0.13	0.882	AWGN	20	0.14	0.856

5.2.4 Collusion of Several Buyers

Table 7 presents the results of a video file "Dragon.avi" against collusion attacks such as averaging, minimum, maximum and median. Table 7 shows the number of colluders U which have been successfully traced through Nuida *et al.*'s codes tracing algorithm 2. In almost all the cases, the colluders have been successfully traced by analyzing a pirated video copy Y. We have considered the number of colluders U upto 5 due to a fact that an increase in U degrades the quality of the content.

5.2.5 Response Time

The response time is the time calculated from the query issuance of the peer to the download of *BF* and *SF* to reconstruction of the file. *BF* is downloaded in a centralized manner between a peer, *M* and *MO*, whereas for distribution of *SF*, the proposed system uses APFS (Scarlata et al., 2001), in which peers construct an anonymous path with

Table 7: Resistance of a Video File to Collusion Attacks

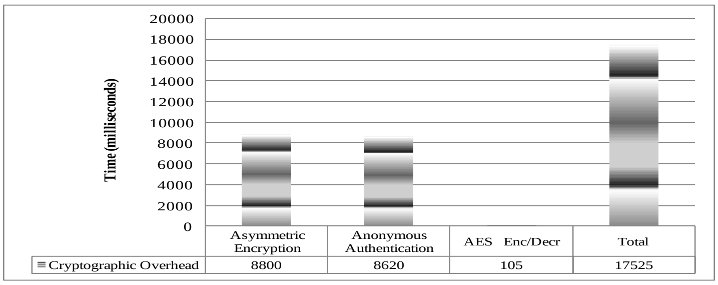
Video				
No. of Colluders	**No. of Colluders Detected for Attacks**			
U	**Avg**	**Min**	**Max**	**Med**
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	4	4	5

tail nodes using onion-routing for achieving mutual anonymity. The anonymous paths construction and authentication through these paths add latency to *SF* transfer process. The response time for *BF* distribution includes the time taken to apply asymmetric encryption and a complete *BF* transfer between *M* and a peer. Similarly, the response time for the distribution of *SF* is evaluated by considering the two-party AKE protocol between *RP* and *CP* and time taken for complete transfer of *SF*. The response time also includes file reconstruction time at the user end. Table 7 summarizes the response time for an audio file "LoopyMusic.wav" and a video file "Traffic.avi". The last column of the table shows the execution time of direct file transfer between *M* and *RP* without considering security and anonymity properties. The distribution time of *BF* is small as compared to the direct transfer time. Thus, the proposed system enables the merchant to save file delivery and CPU time. However, the total distribution time of our system is comparatively higher than direct transfer time due to the anonymous paths construction, authentication and asymmetric encryption. Hence, in achieving anonymity and security in P2P systems, there is always a trade-off between anonymity, security and efficiency. Considering preservation of anonymity and security concerns, the proposed system's response time is feasible and can be implemented in real-life distribution systems.

Table 8: Response Time for an Audio and Video File

File Name	**File Size (KB)**	**File Partition**	**File Recon-struction Time (ms)**	**Query Sent and Reply Time (ms)**	**File Distribution Time**		**Total File Distribution Time (ms)**	**Direct File Distribution Time (ms)**
		BF+*SF* Execution Time (ms)			*BF* Delivery Time (ms)	*SF* Delivery Time (ms)		
LoopyMusic.wav	918.00	20130.00	3890.00	9296.00	2600.00	805.00	127010.00	6000.00
Traffic.avi	204.00	24000.00	7026.40	9296.00	530.00	81.00	99906.00	3000.00

Cryptographic algorithms are applied in the proposed framework to ensure the desired level of security and accountability. We have used AES and Paillier encryption/decryption, and anonymous authentication in our framework. Figure 4 shows the CPU execution time of each cryptographic block for achieving desired security in the proposed system for a video file "Traffic.avi". An anonymous authentication process based on ZKPI with prime numbers $P = 1024$ bits and Q (160 bits) is chosen. *BF* is encrypted/decrypted using Paillier-1024 asymmetric encryption scheme. A symmetric cipher algorithm such as AES-128 is being used to encrypt fragments of *SF*. From Table 7 and Figure 4, it is evident that the distribution of *BF* from *M* to the users of P2P system is cheaper from *M*'s point of view. The use of a P2P system for distributing *SF* considerably reduces the distribution cost of *M*.



	Asymmetric Encryption	Anonymous Authentication	AES Enc/Decr	Total
▤ Cryptographic Overhead	8800	8620	105	17525

Figure 4: Cryptographic overhead

6 Conclusions and Future Work

We have presented a framework for a P2P content distribution system with copyright and privacy protection to the merchant and the end users, respectively. In the proposed framework, the multimedia content is partitioned into a small-sized base file and a large-sized supplementary file. The base file is dispensed by the merchant on payment from the buyer and a supplementary file is distributed to the P2P network. Thus, the scheme lessens the computational cost of the merchant by only sending the small-sized base file and using the P2P network to support the majority of the file transfer process. For generation and distribution of a base file, an asymmetric fingerprinting protocol is performed between the merchant and the buyer in the presence of a trusted monitor. In the proposed framework, a unique collusion resistant digital fingerprint is embedded into the multimedia content using a robust, blind and imperceptible watermarking scheme. In the event that the merchant detects an unauthorized distribution of the content, it extracts the fingerprint from the pirated copy and gives the pirated code to the monitor. The monitor runs the tracing algorithm on the fingerprint to identity the pirate. The user's anonymity is well-protected until there is a need to trace the identity of a user who distributes unauthorized copies of the copyright content. Even in case of arbitration, the cooperation from the buyer is not required. The security and performance analysis demonstrates the security and efficiency of our proposed framework. Hence, our secure and privacy-preserving P2P system could be implemented in real-life distribution applications. The digital media producers would not be afraid of illegal usage and distribution of their products, and the P2P networks would not be blamed for piracy anymore. Future research should be directed:

- To do further reduction in the size of the base file by using compression techniques.
- To do an analysis of the vulnerability of the proposed framework against a malicious Monitor, who may collude with a merchant to frame an honest user.

- To develop a prototype of the proposed framework and test it in a real-world scenario.

Acknowledgement

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

Akamai. (1998). Retrieved from `https://www.akamai.com/`

Apple itunes. (2001). Retrieved from `http://www.apple.com/itunes/`

Barg, A., Blakley, G. R., & Kabatiansky, G. A. (2003). A digital fingerprinting codes: Problems, statements, constructions, identification of traitors. *IEEE Transactions on Information Theory*, *49*, 852–865.

Bianchi, T., & Piva, A. (2013). Secure watermarking for multimedia content protection: A review of its benefits and open issues. *IEEE Signal Processing Magazine*, *30*(2), 87–96.

Bittorrent. (2001). Retrieved from `http://www.bittorrent.com/`

Boneh, D., & Shaw, J. (1999). Collusion secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, *44*(5), 1897–1905.

Camenisch, J. (2000). Anonymous fingerprinting with group signatures. In *Asiacrypt 2000* (Vol. 1976, pp. 415–428). Springer.

Cao, J., & Huang, J. (2012). Controllable secure watermarking technique for tradeoff between robustness and security. *IEEE Transactions on Information Forensics and Security*, *7*(2), 821–826.

Chen, B., & Wornell, G. W. (2001). Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, *47*(4), 1423–443.

Choi, J. G., Sakurai, K., & Park, J. H. (2003). Does it need trusted third party? design of buyer-seller watermarking protocol without trusted third party. In *Applied cryptography and network security* (Vol. 2846, pp. 265–279). Springer.

Chor, B., Fiat, A., Naor, M., & Pinkas, B. (2000). Tracing traitors. *IEEE Transactions on Information Theory*, *46*(3), 893–910.

Cox, I. J., Kilian, J., Leighton, T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, *6*(12), 1673–1687.

Cox, I. J., Miller, M., Bloom, J., Fridrich, J., & Kalker, T. (2007). *Digital watermarking and steganography* (2nd ed.). Morgan-Kaufmann.

Deng, M., & Preneel, B. (2008). On secure and anonymous buyer-seller watermarking protocol. In *Third international conference on internet and web applications and services, iciw '08* (pp. 524–529). IEEE.

Domingo-Ferrer, J., & Herrera-Joancomartí, J. (2000). Short collusion-secure fingerprints based on dual binary hamming codes. *Electronics Letters*, *36*(20), 1697–1699.

Domingo-Ferrer, J., & Megías, D. (2013). Distributed multicast of fingerprinted content based on a rational peer-to-peer community. *Computer Communications*, *36*(5), 542–550.

edonkey2000. (2000). Retrieved from `http://www.emule-project.net/`

Fallahpour, M., & Megías, D. (2011). High capacity audio watermarking using the high frequency band of the wavelet domain. *Multimedia Tools and Applications*, *52*(2–3), 485–498.

Feige, U., Fiat, A., & Shamir, A. (1988). Zero-knowledge proofs of identity. *Journal of Cryptology*, *1*(2), 77–94.

Freenet. (2000). Retrieved from `https://freenetproject.org/`

García-Dorado, J., Finamore, A., Mellia, M., Meo, M., & Munafó, M. (2012). Characterization of isp traffic: Trends, user habits, and access technology impact. *IEEE Transactions on Network and Service Management*, *9*(2), 142–155.

Grangetto, M., Magli, E., & Olmo, G. (2006). Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, *8*(5), 905–917.

gtk-gnutella. (2000). Retrieved from `http://gtk-gnutella.sourceforge.net/en/?page=news`

Hartung, F., & Kutter, M. (1999). Multimedia watermarking techniques. *Proceedings of the IEEE*, *87*(7), 1079–1107.

imesh. (1999). Retrieved from `http://www.imesh.com/`

Internap. (1996). Retrieved from `http://www.internap.com/`

Kerry, C. (n.d.). *Digital signature standard, fips pub 186-4.* Retrieved from `http://csrc.nist.gov/publications/fips/fips186-4/fips186-4change1.pdf`

Khurana, K., & Chandak, M. B. (2013). Key frame extraction methodology for video annotation. *International Journal of Computer Engineering and Technology*, *4*(2), 221–228.

Kuribayashi, M., & Tanaka, H. (2005). Fingerprinting protocol for images based on additive homomorphic property. *IEEE Transactions on Image Processing*, *14*(12), 2129–2139.

Li, J. S., Hsieh, C. J., & Hung, C. F. (2010). A novel drm framework for peer-to-peer music content delivery. *Journal of Systems and Software*, *83*(10), 1689–1700.

Li, X., Krishnan, S., & Ngok-Wah, M. (2010). A wavelet-pca-based fingerprinting scheme for peer-to-peer video file sharing. *IEEE Transactions on Information Forensics and Security*, *5*(3), 365–373.

Lian, S., Kanellopoulos, D., & Ruffo, G. (2009). Recent advances in multimedia information system security.

Informatica, *33*(1), 3–24.

Lou, X., & Hwang, K. (2009). Collusive piracy prevention in p2p content delivery networks. *IEEE Transactions on Computers*, *58*(7), 970–983.

Lu, L., Han, J., Liu, Y., Hu, L., Huai, J. P., Ni, L., & Ma, J. (2008). Pseudo trust: Zero-knowledge authentication in anonymous p2ps. *IEEE Transactions on Parallel and Distributed Systems*, *19*(10), 1325–1337.

Martínez-Ballesté, A., Sebé, F., Domingo-Ferrer, J., & Soriano, M. (2003). Practical asymmetric fingerprinting with a ttp. In *Proceedings of the 14th international workshop on database and expert systems applications* (pp. 352–356). IEEE.

Megías, D., & Domingo-Ferrer, J. (2013). Privacy-aware peer-to-peer content distribution using automatically recombined fingerprints. *Multimedia Systems*, *20*(2), 105–125.

Memon, N. D., & Wong, P. W. (2001). A buyer-seller watermarking protocol. *IEEE Transactions on Image Processing*, *10*(4), 643–649.

Microsoft drm. (2008). Retrieved from `http://www.wmlicense.smdisp.net/wmdrmcompliance/WindowsMediaDRM10fordevices/`

Msu video quality measurement tool. (2011). Retrieved from `http://compression.ru/video/quality_measure/video_measurement_tool_en.html`

Napster. (2011). Retrieved from `http://www.napster.co.uk/start`

Nuida, K., Fujitsu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., & Imai, H. (2007). An improvement of tardos's collusion-secure fingerprinting codes with very short lengths. In *Proceedings of the 17th international conference on applied algebra, algebraic algorithms and error-correcting codes* (pp. 80–89). Springer.

Opera. (1999). Retrieved from `http://www.opticom.de/products/audio-quality-testing.html/`

Pagnia, H., & Gartner, F. C. (1999). On the impossibility of fair exchange without a trusted third party. Darmstadt University of Technology.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th international conference on theory and application of cryptographic techniques* (pp. 223–238). Springer.

Passarella, A. (2012). Review:a survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Computer Communications*, *35*(1), 1–32.

Pfitzmann, B., & Schunter, M. (1996). Asymmetric fingerprinting. In *Proceedings of the 15th annual international conference on theory and application of cryptographic techniques* (pp. 84–95). Springer.

Phan, R. C. W., Goi, B. M., Poh, G. S., & Jongsung, K. (2011). Analysis of a buyer-seller watermarking protocol for trustworthy purchasing of digital contents. *Wireless Personal Communications*, *56*(1), 73–83.

Poh, G. S., Mohd. Nazir., M. A. N., Goi, B. M., Tan, S. Y., Phan, R. C. W., & Shamsudin, M. M. (2013). An authentication framework for peer–to–peer cloud. In *Proceedings of the 6th international conference on security of information and networks* (pp. 94–101). ACM.

Prins, J. P., Erkin, Z., & Lagendijk, R. L. (2007). Anonymous fingerprinting with robust qim watermarking techniques. *EURASIP Journal on Information Security*, *2007*(20), 1–7.

Qian, L., & Nahrstedt, K. (1998). Watermarking schemes and protocols for protecting rightful ownership and customers rights. *Journal of Visual Communication and Image Represention*, *9*(3), 194–210.

Rial, A., Deng, M., Bianchi, T., Piva, A., & Preneel, B. (2010). A provably secure anonymous buyer-seller watermarking protocol. *IEEE Transactions on Information Forensics and Security*, *5*(4), 920–931.

Rodriguez-Perez, M., Esparza, O., & Muñoz, J. L. (2008). Surework: A super-peer reputation framework for p2p networks. In *Proceedings of the 2008 acm symposium on applied computing* (pp. 2019–2023). ACM.

Scarlata, V., Levine, B. N., & Shields, C. (2001). Responder anonymity and anonymous peer-to-peer file sharing. In *Ninth international conference on network protocols* (pp. 272–280). IEEE.

Schneier, B. (1996). *Applied cryptography protocols, algorithms, and source code in c* (2nd ed.). John Wiley and Sons, Inc.

Tardos, G. (2003). Optimal probabilistic fingerprint codes. In *Proceedings of the thirty-fifth annual acm symposium on theory of computing* (pp. 116–125). ACM.

Theotokis, S. A., & Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computer Survey*, *36*(4), 335–371.

Voloshynovskiy, S., Farhadzadeh, F., Koval, O., & Holotyak, T. (2012). Active content fingerprinting: A marriage of digital watermarking and content fingerprinting. In *International workshop on information forensics and security (wifs)* (pp. 175–180). IEEE.

Von-Lohmann, F. (2003). Peer-to-peer file sharing and copyright law: A primer for developers. In *Peer-to-peer systems ii* (Vol. 2735, pp. 108–117). Springer.