Peer-to-peer Content Distribution Using Anonymous Fingerprinting – Proof of Concept

Amna Qureshi, Jordi Casas-Roma, David Megías and Helena Rifà-Pous
Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica, Multimèdia i Telecomunicació,
Universitat Oberta de Catalunya,
Av. Carl Friedrich Gauss, 5, 08860 Castelldefels
E-mail: {aqureshi,jcasasr,dmegias,hrifa}@uoc.edu

***Abstract*—Peer-to-peer (P2P) networks have emerged as a cost-efficient multimedia distribution solution in the last few years, but they have also been surrounded by the copyright controversy as they are easily identified with the illegal copying of copyrighted materials. Although some copyright protection systems have been proposed to overcome this drawback, most of them posed a risk to the privacy of legitimate users. Recently, anonymous fingerprinting methods have been proposed using an underlying P2P distribution to combine copyright enforcement (by means of traitor tracing) and the privacy of honest buyers. However, to the best of our knowledge, none of those systems has provided a complete proof of concept to enable the evaluation of the overall solution and its application in a real-world scenario. This paper describes the complete implementation of one of these systems and provides real data obtained from the integration of all its different components, including a real P2P platform. The data obtained from the experiments are no longer theoretical, but based on the results attained with real hardware and software. Thus, this paper is intended to pave the way between theoretical results and a commercial application of the system, bridging the gap between academia and industry, to illustrate the possibilities of transferable research.**

***Keywords*—Peer-to-peer networking, Anonymous fingerprinting, Privacy, Proof of concept.**

I. Introduction

The most common approach to the distribution of copyrighted multimedia contents is centralized and based on the traditional client-server model, which suffers from lack of scalability and requires a costly initial investment as infrastructure is concerned. In addition, centralized systems also have extremely high demands in both bandwidth requirements and CPU time. On the other hand, peer-to-peer (P2P) solutions have been proposed that overcome these drawbacks, allowing the deployment of the distribution system in a cost-efficient and scalable manner, making it possible to reach a wider number of users with a relatively simple infrastructure. For example, BitTorrent [1], one of the most popular P2P distribution systems, accounts for a significant share of all Internet traffic. The convenient features of the P2P paradigm has made it very attractive for many companies, including software producers and game and media companies, which are shifting to the use of P2P solutions for the distribution of software, game updates or videos, respectively.

However, P2P technology has often been criticized due to the copyright controversy, since these kind of networks are the major source of illegally re-distributed copies of copyrighted contents. As a consequence of this, many companies are reluctant of shifting to the adoption of the P2P framework since they fear losing control of the content ownership. In fact, the decentralized operation of P2P networks makes it difficult to trace illegal re-distributions in this framework.

Encryption has been discarded as a convenient way of protecting copyright during distribution since, once decrypted, the content can be copied and re-distributed without limits. The main solutions to prevent illegal copying in P2P scenarios are based on Digital Rights Management (DRM) solutions [9] and on fingerprinting or transaction tracking (one of the applications in the area of data hiding) [3], [11], [12], [15], [16].

DRM-based solutions are based on limiting the devices which can be used to play some given contents. The idea is that only the devices of authorized users will be able to play the contents according to their rights. However, these solutions are difficult to reconcile with the users' privacy rights. In fact, most of the existing DRM-based solutions do not consider the privacy of buyers as a feature.

On the other hand, in digital fingerprinting [2], a particular identifying mark, called a fingerprint, is embedded into each copy of the content, making it possible to trace the source of an illegal re-distribution. The first few fingerprinting schemes were symmetric, meaning that both the merchant and the buyer had access to the fingerprinted copy of the content. In that case, the buyer could not be formally accused of illegal re-distribution, since he or she may argue that the copy could have been re-distributed by the merchant itself. Later on, asymmetric fingerprinting methods were proposed [13]. Those methods guarantee that only the buyer obtains the fingerprinted copy of content and, hence, overcome the major drawback of symmetric methods. Finally, anonymous fingerprinting protocols [14] were proposed in such a way that the asymmetric property was retained, and the identity of the buyer was protected, thus achieving buyers' privacy.

Most of the existing anonymous fingerprinting methods are completely centralized and, thus, they suffer from scalability problems. In fact, centralized systems require some demanding technologies in terms of both bandwidth and CPU time, such as zero-knowledge proofs, secure multi-party computation

protocols or homomorphic encryption of the whole contents per each buyer (and decryption in the buyer's side). To overcome these problems, in the last few years, different anonymous fingerprinting that are designed to work within the P2P paradigm were proposed [3], [11], [12], [15], [16], with different characteristics.

In [3], a P2P protocol based on the concepts of game theory is presented to provide the multicast distribution of fingerprinted contents. This system requires the implementation of a secure multi-party protocol in each transaction between buyers to embed the next fingerprint. In [11], anonymous fingerprinting based on a random recombination of the fingerprints' segments of the source buyers is presented. The system provides automatic fingerprints due to recombination, but it requires trusted proxies in the distribution protocol. In order to trace an illegal re-distributor, the system implements a graph-search in the distribution graph, which can be problematic since it requires the co-operation of some innocent buyers. In addition, a double layer encoding of the fingerprint with collusion-resistant codes is required, leading to longer fingerprints and requiring a non-specified protocol to build valid codewords during distribution. Most of the drawbacks of this system are removed in [12], where untrusted proxies are used for distribution and the graph search method for traitor-tracing is replaced by a standard database search. In this way, the system is able to trace an illegal re-distributor even if innocent buyers refuse to co-operate. However, the scheme still requires the double layer encoding of the fingerprint for collusion resistance.

In [15], a P2P content distribution framework for preserving privacy and security of the user and the merchant based on homomorphic encryption is presented. This scheme is based on splitting the content into a short-sized base file, containing the most relevant information of the content, and much larger long-sized supplementary file that is useless without the former. The base file is distributed in a centralized way from merchant to buyer and the fingerprint is embedded anonymously using homomorphic encryption. On the other hand, the supplementary file is distributed in a pure P2P fashion since it contains no relevant information from the copyright point of view. These ideas are taken one step forward in [16], where the Privacy and Security of User and Merchant (PSUM) system is presented. The PSUM system also uses the idea of splitting the content into a base file and a supplementary file, but it removes the need of homomorphic encryption for the distribution of base files to the buyers. These distribution is implemented with a family of proxy peers that transfer permuted coefficients from the merchant to the buyer, using a permutation key that is known only by the buyer and a transaction monitor. In this way, both the security of the content and the buyer frameproofness can be guaranteed. In addition, the proxies can be completely untrusted.

The PSUM system is analysed in [16] from a theoretical point of view and with a limited experimental section. This paper is devoted to analyze a complete implementation of the system using a real P2P distribution platform: JXTA [5], [17]. Hence, this paper provides a complete proof of concept of the proposed method and analyzes several relevant performance indicators such as the overheads of the different cryptographic functions and the communication required by the different entities involved in the protocol. The results obtained with this proof of concept show that the system is ready to be used in a real-world scenario for multimedia content distribution in such a way that both the copyright of the contents and the privacy of the buyers are guaranteed.

The rest of the paper is organized as follows. In Section II, we review the main components in our scenario: the multimedia distribution algorithm and the selected network for this proof of concept. Section III introduces our experimental framework to analyse the implementation on a real networks. Results are discussed in Section IV. Lastly, in Section V, we present the conclusions of this work.

II. Scenario

In this section, we provide in-depth description of the PSUM [16] algorithm and the network selected to implement this proof of concept. We implemented the algorithm using Sun Microsystems' JXTA network[5], an open source java implementation of a basic and pure P2P network.

A. The Algorithm

The main contribution of this paper is the demonstration of PSUM, an asymmetric fingerprinting protocol [16], that provides a secure, anonymous and efficient collusion-resistant-based fingerprinting within a P2P environment. The proposed fingerprinting scheme reduces the computational and communication costs of the merchant (M) by using the idea of file partitioning. The multimedia file is partitioned by M into a small-sized base file (*BF*) and a large-sized supplementary file (*SF*). The *BF* contains the most important information and, without it, the *SF* is unusable. On a particular file request by a user (buyer) of JXTA network, M uses a network of edge peers (Pr_j) to send *BF* to the requesting buyer (B_i). For a secure distribution of *BF* to a buyer, the merchant, the trusted party called monitor (*MO*), B_i and a selected set of Pr_j perform an asymmetric fingerprinting protocol 1.

B. Selected Network

The project JXTA [17] is an open-source peer-to-peer protocol specification originally conceived by Sun Microsystems, Inc. and designed with the participation of a small number of experts from academic institutions and industry.

The JXTA protocols are defined as a set of XML messages which allow any device connected to a network to exchange messages and collaborate independently of the underlying network topology. The JXTA protocols define a virtual network overlay on top of the existing physical network infrastructure upon which services and applications are built. The main purpose of the JXTA virtual network is to provide a uniform, addressable network for all peers in the network.

JXTA peers create a virtual overlay network which allows a peer to interact with other peers. Each resource is identified

Protocol 1 *BF* distribution protocol

1: B_i negotiates with M to set-up an agreement that explicitly states the rights and obligations of both parties and specifies the video file. M generates a transaction ID for keeping a record of the transaction between him/her and B_i. Then, M sends a request for a fingerprint f_i to *MO*.
2: *MO* generates a Nuida's c-secure codeword f_i of length m and randomly selects n proxy peers (Pr_j, for $j = 1, \ldots, n$) for a secure transfer of fingerprinted *BF* from M to B_i. Then, *MO* sends a request for permutation keys σ_j and session keys K_{ses_j} to B_i.
3: After receiving a request from *MO*, B_i generates n random permutation keys σ_j (for $j = 1, \ldots, n$) of length $l = \lfloor m/n \rfloor$ and n session keys K_{ses_j}. The session keys are generated to be shared with M, such that Pr_j are unable to see the clear-text of the coefficients. B_i sends $E_{K_{p_{MO}}}(\sigma_j, E_{K_{p_M}}(K_{ses_j}))$ to *MO*.
4: *MO* decrypts $E_{K_{p_{MO}}}(\sigma_j, E_{K_{p_M}}(K_{ses_j}))$ with $K_{s_{MO}}$ and obtains σ_j and $E_{K_{p_M}}(K_{ses_j})$. *MO* divides f_i into n segments of length l and permutes each segment using σ_j in the same order as received by B_i. *MO* encrypts σ_j and K_{ses_j} with a public key of M and then sends concatenated encrypted permutation and session keys $(E_{K_{p_M}}(\sigma_j)|E_{K_{p_M}}(K_{ses_j}))$ to M.
5: M decrypts $E_{K_{p_M}}(\sigma_j)|E_{K_{p_M}}(K_{ses_j})$ with K_{s_M} and obtains σ_j and K_{ses_j}. M performs permutation on both pre-computed variants of *BF* with σ_j. The permuted variants of *BF* are encrypted with K_{ses_j} by M.
6: *MO* assigns contiguous permuted fingerprint segments to Pr_j, who then contact M in a sequential manner to obtain the fragments of the encrypted and permuted approximation coefficients.
7: Pr_j selects the correct pre-computed approximation coefficients from the received coefficients using the assigned permuted fingerprint segments.
8: On receiving all the fragments of the *BF* from Pr_j, B_i permutes back the encrypted coefficients with n inverse permutation keys and, then, decrypts the received encrypted approximation coefficients to obtain the fingerprinted coefficients of *BF*. B_i obtains a complete copy of *BF* by composing all the coefficients received sequentially from all Pr_j.

by a unique ID, a 160 bit SHA-1. JXTA defines two main categories of peers:

- The *edge peers* are usually defined as peers which have transient, low bandwidth network connectivity.
- The super-peers can be further divided into two categories:
  - A *rendezvous peer* is a special purpose peer which coordinates the peers in the network and provides scope to message propagation.
  - A *relay peer* maintains routing tables for relaying messages to their destination and allows the peers which are behind firewalls or NAT systems to take part in the JXTA network.

A peer group provides a scope for message propagation and a logical clustering of peers. Every peer is a member of a default group, but it can be member of many sub-groups at the same time. Each group should have at least one rendezvous peer.

Communication in JXTA network is done through advertisements and pipes. An advertisement is an XML document which describes any resource in a P2P network. A pipe is a virtual communication channel used to exchange messages and data. Pipes are asynchronous, unreliable, and uni-directional.

C. Proposed Framework

Our proof-of-concept framework involves two major environments: (1) M, *MO* and B_i are implemented using Matlab [10] and (2) the distribution of *BF* is coded using the Java implementation of the JXTA Protocols [6].

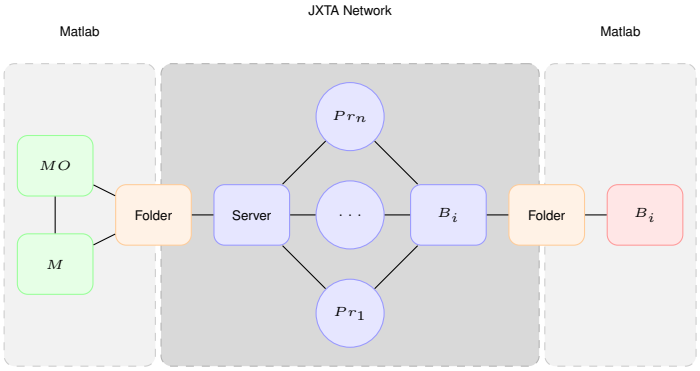

Fig. 1. Main structure of the proof of concept's framework

The proposed framework is depicted in Figure 1 and the source code is available at [4]. The process is as follows:

1) M sends a request for a fingerprint (f_i) to *MO*, who generates Nuida's c-secure codewords using Matlab.
2) M performs permutation and encryption on the permuted variants of *BF* and stores them in a temporary file system folder, which is available from the JXTA network.
3) A Java process (*Server*) initializes the Java implementation of the JXTA network (JXSE).
4) Next, the *Server* assigns contiguous permuted fingerprint segments to edge peers (Pr_j) in the JXTA network. We do not use super-peers (rendezvous or relay peers) in our proof-of-concept due to the fact that it works on a single local area network, but they are necessary for a real-world implementation.
5) A set of n edge peers are used to transmit *BF* to B_i.
6) On receiving all the fragments of *BF* from Pr_j, each B_i obtains the *BF* by composing all the sequences received from Pr_j in the JXTA network and store them in a temporary folder, which is available from Matlab.

7) Lastly, B_i decrypts *BF* to obtain the original file using Matlab.

III. Implementation

For the demonstration of the proposed asymmetric fingerprinting protocol [16] in JXTA network, we considered a video file "Traffic.avi", which consists of a total of 120 frames, of which, 15 are intra frames and 105 are inter-frames. The size of the intra and inter frames is 120×160 pixels. In this demonstration, the merchant is considered as an emulated server, the buyer as a P2P network user that could either act as a super-peer or a relay P2P peer, and the proxy peers are randomly selected edge peers. Although no user interface is provided since it is a proof of concept, Matlab offers interaction and graphic generation, as can be seen in Figure 2.

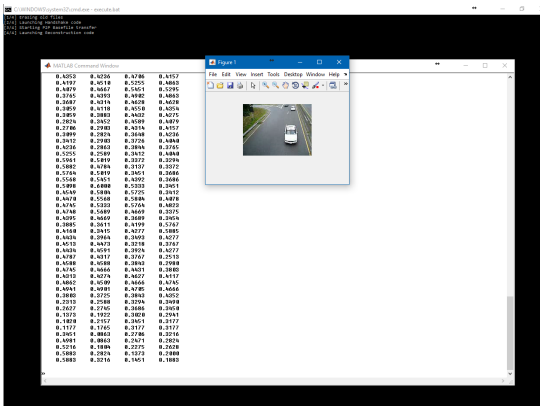

Fig. 2. An screenshot example of our execution environment.

In order to divide "Traffic.avi" into *BF* and *SF*, it is necessary to extract the significant frames from the file, since all the video frames do not contain important information. The video frames are arranged into groups of pictures, which include the intra frames (I-frames or key frames) and inter-frames (P and B-frames). It is not advisable to embed data both into intra and inter-frames. Thus, we only used I-frames which contain the most significant information. In order to obtain the I-frames, We used the Canny Edge Difference technique [7]. In the Canny Edge Difference detection method, a difference between two consecutive frames was calculated and if this difference exceeded a calculated threshold value, we obtained a key frame. The extracted key frames were converted from the RGB format to Y'UV, whereas the remaining frames, i.e. P and B-frames were saved in an original video form, i.e. avi format. The Y'UV model defines a color space in terms of one luminance (Y') and two chrominance (UV) components. For each I-frame, we chose the Y' component and applied level-3 DWT to obtain approximation (a_3) and detail (d_3) coefficients. We considered level-3 DWT decomposition to obtain a convenient trade-off between the robustness, capacity and transparency properties of watermarking. The level-3 DWT of each I-frame resulted in a matrix containing 20×15 values.

Two variants of a_3 coefficients of each I-frame were produced: a_3^0 embedded with the codewords containing all 0s and a_3^1 embedded with codewords containing all 1s. The codewords were embedded into a_3 using a blind, robust and secure QIM-based watermarking algorithm proposed by Leelavathy *et al.* [8]. a_3^0 and a_3^1 were saved in a folder as BF^0 and BF^1 in binary (text) format, respectively, which were then distributed to B_i through Protocol 1. In *BF* distribution protocol, five Pr_j were considered to deliver the contents of the *BF* from M to B_i. Each Pr_j carried permuted fingerprint segments of length l. Both BF^0 and BF^1 were permuted sequentially by M using $n = 5$ permutation keys (σ_j). The permuted variants of *BF* were then encrypted with five K_{ses_j}. Both permuted and encrypted BF^0 and BF^1 were saved in a text format in a folder accessible to a JXTA network. All Pr_j obtained the fragments of encrypted and permuted approximation coefficients sequentially from M. The correct pre-computed (permuted and encrypted) approximation coefficients were selected from the received coefficients by Pr_j using the assigned permuted fingerprint segments. All these coefficients were saved in a folder as a text file, that was accessible to Matlab 7.0 at B_i's end. Five inverse ${\sigma_j}^{-1}$ were used by B_i to permute back the permuted and encrypted coefficients. Then, K_{ses_j} were used to decrypt the coefficients in order to obtain the fingerprinted coefficients of *BF*. An inverse level-3 DWT was applied on the *BF* to obtain a fingerprinted *BF*.

IV. Experimental Results

In this section, we show the experimental results to analyze the performance of the proposed algorithm. Two types of experiments were performed: 1) computational costs; and 2) communication costs. The demonstration was performed for a scenario where "Traffic.avi" video file was requested from M by a single B_i through 5 proxy peers in the presence of trusted party called *MO*. The settings in the experiment are listed as follows:

- The video file size: 0.19 MB
- Workstation: Intel i-7 processor at 3.4 GHz and 8 GB of RAM.
- Bandwidth: Fast Ethernet connection (100 MB/s)
- The number of peers that join the system: 5 nodes.
- Peer joining process: Arrive in various time (6 nodes).
- Peer leaving process: Leave after a random waiting interval (1 node).

A. Computational Costs

The execution time of *BF* generation process for a video file involves file partitioning by M, f_i generation by *MO*, σ_j and K_{ses_j} generation by B_i, f_i permutation by *MO*, assignment of permuted bits to a selected set of Pr_j by *MO*, and symmetric-key encryption of pre-computed coefficients by M. All these tasks were executed in Matlab 7.0 and were stored in the folder that was made accessible to the JXTA network. The overheads are calculated for each party (M, *MO*, B_i and Pr_j) involved in the proposed protocol.

Table I shows the overhead costs of M, which involves RGB to Y'UV conversion of each I-frame, DWT on the Y' components of 15 key frames, decryption of the session and permutations keys sent by the *MO* to M, permutation and symmetric encryption of the permuted coefficients.

TABLE I
OVERHEAD COSTS OF M (CPU TIME IN SECS)

Functions	**Total**	**Frame**
RGB to Y'UV+DWT+Embedding	0.85	0.056
Decryption ($K_{sess_j} + \sigma_j$)	0.14	0.009
Permutation of coefficients	0.05	0.003
Encryption of permuted coefficients	0.90	0.060
Total CPU time (secs)	**1.94**	**0.129**

The RGB conversion to Y'UV format and DWT on the Y' components of the key frames are applied once by M to obtain the approximation and detail coefficients. M stores the approximation and detail coefficients of each video file in order to avoid the costs of performing both processes every time a video file is requested by B_i. From Table I, it can be seen that it takes M 1.94 seconds to generate two *BF* variants of permuted and encrypted coefficients, thus the overhead cost of M is trivial in the proposed protocol.

TABLE II
OVERHEAD COSTS OF *MO* (CPU TIME IN SECS)

Functions	**Total**	**Frame**
Fingerprint generation	6.01	6.0100
Fingerprint segmentation+Permutation	0.02	0.0200
Communication with M	0.12	0.0080
Communication with B_i	0.17	0.0113
Communication with Pr_j	0.01	0.0006
Total CPU time (secs)	**6.33**	**6.0499**

In Table II, we present the overhead of *MO* for "Traffic.avi". *MO* is responsible for the generation of the collusion-resistant fingerprint f_i, the decryption of received permutation keys from B_i, the segmentation and permutation of f_i, the transfer of encrypted session and permutation keys to M and the allocation of the permuted fingerprint to Pr_j. It can be seen from Table II that for a single video file ("Traffic.avi") request, it takes *MO* only 6.33 seconds to perform all the desired functions. In case there are multiple file requests, *MO* can execute these tasks in parallel, therefore keeping the minimal overhead.

TABLE III
OVERHEAD COSTS OF B_i (CPU TIME IN SECS)

Functions	**Total**	**Key**
Generation and encryption of permutation and session keys	0.21	0.021

In the proposed protocol, the overhead of B_i involves generation of n permutation and session keys. Table III presents the time taken by B_i to generate σ_j and K_{ses_j}, and the encryption of these keys. From Table III, we can see that B_i performs the required operations in 0.21 seconds for "Traffic.avi".

B. Communication Costs

Four entities, i.e. M, MO, Pr_j and B_i, are involved in the execution of *BF* distribution from M to B_i. The overhead of M was not considered in *BF* delivery since it is assumed that before the execution of *BF* distribution protocol, a pre-computed permuted and encrypted BF^0 and BF^1 were already generated by M. Similarly, the overhead of *MO* was not counted in *BF* delivery response time since we assumed that a database of fingerprint was generated by *MO* before the start of *BF* distribution protocol.

In the *BF* distribution protocol, M is contacted by all the proxy peers in a sequential manner to obtain the fragments of encrypted and permuted approximation coefficients stored in BF^0 and BF^1 in a block form. The correct pre-computed approximation coefficients were selected from the coefficients by Pr_j using the assigned permuted fingerprint segments. Table IV summarizes the performance results obtained when the protocol is implemented in a JXTA network and in our own customized small-scale P2P simulator.

TABLE IV
OVERHEAD COSTS OF Pr_j (TIME IN SECS)

Transfer time (in secs)	**Total**	**Frame**
JXTA network	0.40	0.026
P2P simulator	1.92	0.128

The overhead costs of B_i also includes file reconstruction process that was executed in Matlab 7.0 at the buyer's end. At B_i's end, the encrypted coefficients were permuted back with inverse permutation keys, then were decrypted with the session keys to obtain the fingerprinted coefficients of *BF*. The coefficients received sequentially from all five Pr_j were composed by B_i and then inverse level-3 DWT was applied on these coefficients to obtain a fingerprinted *BF*. Table V presents the reconstruction time of "Traffic.avi" at B_i's end.

TABLE V
FILE RECONSTRUCTION AT B_i'S END (CPU TIME IN SECS)

Functions	**Total**	**Frame**
Inverse permutation	0.01	0.0006
Inverse DWT, Y'UV to RGB, and conversion to original format	3.95	0.2633
Total CPU time (secs)	**3.96**	**0.2640**

Table VI compares the experimental results of *BF* distribution protocol (file partitioning, delivery and reconstruction) simulated in JXTA platform with the results of *BF* delivery algorithm executed in our custom-built P2P simulator. It can be seen from the table that total file distribution time in JXTA P2P framework is 8.00 secs as compared to 14.36 secs evaluated using our own P2P simulator.

TABLE VI
COMPARATIVE ANALYSIS: FILE "TRAFFIC.AVI" (IN SECS)

P2P networks	Total *BF* distribution time
Custom built P2P	14.36 seconds
JXTA P2P network	8.00 seconds

V. CONCLUSION

This paper presents a proof of concept of the implementation of an anonymous fingerprinting scheme for P2P networks. The scalability of the P2P paradigm is a very attractive characteristic for content distributors, but the requirement of implementing copyright-enforcing features within these systems is an urgent need. Anonymous fingerprinting schemes have been proposed in the last few years in conjunction with P2P distribution, but those works did not provide any complete proof of concept of the suggested technology. This paper contains the description of the full proof of concept of a recent anonymous fingerprinting scheme for P2P networks. In addition, several key performance indicators of the proposed scheme have been obtained in this paper showing the feasibility of the analysed system for a real-world application scenario. Hence, the paper goes far beyond theoretical results or limited experimentation and shows the feasibility of building a practical distribution system. Finally, the results obtained in the paper show the excellence performance of all the elements of the system and paves the way for a future platform for multimedia distribution guaranteeing both the copyright of the contents and the users' rights regarding privacy preservation.

ACKNOWLEDGEMENTS

This work was partly funded by the Spanish MCYT and the FEDER funds under grants TIN2011-27076-C03 "CO-PRIVACY" and TIN2014-57364-C2-2-R "SMARTGLACIS".

The authors thank Alexandre Dotor for his valuable contribution to the implementation of this proof of concept.

REFERENCES

[1] BiTorrent homepage. 2001. http://www.bittorrent.com. Last accessed June 15, 2016.

[2] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," in *IEEE Transactions on Image Processing*, vol. 6(12), pp. 1673–1687, 1997.

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

[4] Demonstrator for Framework for Privacy-aware Content Distribution in Peer-to-Peer Networks with Copyright Protection. n.d. http://einfmark.uoc.edu/technology/get/id/1. Last accessed June 15, 2016.

[5] JXTA: The Language and Platform Independent Protocol for P2P Networking. n.d. https://jxta.kenai.com/. Last accessed June 15, 2016.

[6] JXSE: The Java Implementation of the JXTA Protocols. n.d. https://jxse.kenai.com/. Last accessed June 15, 2016.

[7] K. Khurana, M. B. Chandak, "Key frame extraction methodology for video annotation," *International Journal of Computer Engineering and Technology*, vol. 4, pp. 221–228. 2013.

[8] N. Leelavathy, E. V. Prasad, S. S. Kumar, B. C. Mohan, "Oblivious image watermarking in discrete multiwavelet domain using QIMM," *Journal of Multimedia*, vol. 6, pp. 359–328. 2011.

[9] JS. Li, CJ. Hsieh, CF. Hung, "A Novel DRM Framework for Peer-to-Peer Music Content Delivery A novel drm framework," *Journal of Systems and Software*, vol. 83(10), pp. 1689–1700, 2010.

[10] Matlab: The Language of Technical Computing. n.d. http://www.mathworks.com/products/matlab/. Last accessed June 15, 2016.

[11] D. Megías, J. Domingo-Ferrer, "Privacy-aware peer-to-peer content distribution using automatically recombined fingerprints," *Multimedia Systems*, vol. 20(2), pp. 105–125, 2014.

[12] D. Megías, "Improved Privacy-Preserving P2P Multimedia Distribution Based on Recombined Fingerprints," *IEEE Transactions on Dependable and Secure Computing*, vol. 12(2), pp. 179–189, 2015.

[13] B. Pfitzmann, M. Schunter, "Asymmetric fingerprinting," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pp.84–95, 1996.

[14] B. Pfitzmann, M. Waidner, "Anonymous fingerprinting," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pp.88–201, 1997.

[15] A. Qureshi, D. Megías, H. Rifà, "Framework for preserving security and privacy in peer-to-peer content distribution systems," *Expert Systems with Applications*, vol. 3, pp. 1391–1408. 2015.

[16] A. Qureshi, D. Megías, H. Rifà, "PSUM: Peer-to-peer multimedia content distribution using collusion-resistant fingerprinting," *Journal of Network and Computer Applications*, vol. 66, pp. 180–197, 2016.

[17] B. Traversat, M. Abdelaziz, D. Doolin, M. Duigou, J. C. Hugly, E. Pouyoul, "Project JXTA-C: enabling a Web of things," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp. 1–9, 2003.