Computer Communications xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Distributed multicast of fingerprinted content based on a rational peer-to-peer community

Josep Domingo-Ferrer [a], David Megías [b,*]

[a] Universitat Rovira i Virgili, UNESCO Chair in Data Privacy, Department of Computer Engineering and Mathematics, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
[b] Universitat Oberta de Catalunya, Internet Interdisciplinary Institute (IN3), Estudis d'Informàtica, Multimèdia i Telecomunicació, Rambla del Poblenou, 156, E-08018 Barcelona, Catalonia, Spain

ARTICLE INFO

Article history:
Received 23 May 2012
Received in revised form 8 October 2012
Accepted 26 December 2012
Available online xxxx

Keywords:
Co-utility
Multicast fingerprinting
Anonymous fingerprinting
Game theory

ABSTRACT

In conventional multicast transmission, one sender sends the same content to a set of receivers. This precludes fingerprinting the copy obtained by each receiver (in view of redistribution control and other applications). A straightforward alternative is for the sender to separately fingerprint and send in unicast one copy of the content for each receiver. This approach is not scalable and may implode the sender. We present a scalable solution for distributed multicast of fingerprinted content, in which receivers rationally co-operate in fingerprinting and spreading the content. Furthermore, fingerprinting can be anonymous, in order for honest receivers to stay anonymous.

1. Introduction

Copyright protection techniques have gained widespread attention by both academia and industry in the recent years. Home Internet access and the increased bandwidth of communications have contributed to the explosion of copyright-breaking copying of digital contents. In this context, fingerprinting emerged as a convenient technology to fight against unlawful digital content distribution [6,4].

Fingerprinting techniques consist of embedding a transparent watermark into the protected content in such a way that a unique identifier exists for each buyer of the content. This identifier can be extracted later on and might be used to trace and match an illegal distributor of the content. This makes it possible to undertake the appropriate legal actions against such treacherous buyers. Fingerprinting schemes can be classified in three different categories [7], namely symmetric, asymmetric and anonymous. In symmetric fingerprinting, the embedding of the fingerprint is performed by the merchant only and, thus, it provides no valid evidence of a treacherous behavior of a buyer (since the merchant herself could be the illegal distributor). In asymmetric fingerprinting, the embedding is performed using a protocol designed in such a way that only the buyer obtains the fingerprinted copy of the content. This makes it possible to prove the illegal distributor's treachery to a third party. Finally, anonymous fingerprinting retains the asymmetric property and also protects the privacy of buyers, whose identity is only revealed and disclosed in case of illegal distribution.

From the point of view of a buyer, anonymity is a valuable property and several protocols have been proposed for anonymous fingerprinting. However, current anonymous fingerprinting proposals in the literature (see Section 2.3 below for a brief review) place a substantial computational and communication burden on the merchant. The merchant's overhead is a relevant issue, since it will possibly result in buyer anonymity not being offered or offered at higher price by the merchant so that the latter can still enjoy some profit margin. Hence, the possibility of reducing the merchant's burden and the flexibility of choosing the watermarking technology freely among the best state-of-the-art techniques are worth investigating. This paper focuses on proposing a multicast approach to the anonymous fingerprinting problem which meets these two goals and shows a proof of concept with a practical implementation of the proposed system. The idea is to transfer the burden of a centralized fingerprinting technology to a distributed network of buyers who will collaborate to produce further copies of the fingerprinted contents.

Sending a content to N different receivers via multicast is much more bandwidth-efficient from the sender's point of view than performing N successive unicast transmissions. However, the unicast approach has the advantage of allowing the sender to fingerprint the content sent to each receiver. Unfortunately, the standard multicast approach does not allow fingerprinting: all

* Corresponding author.
E-mail addresses: josep.domingo@urv.cat (J. Domingo-Ferrer), dmegias@uoc.edu (D. Megías).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.12.005

receivers get exactly the same content. That is why a specific multicast anonymous fingerprinting protocol is proposed in this paper.

1.1. Contribution and plan of this paper

We specify a protocol whereby a sender manages to distribute a digital content to an unlimited number of receivers in such a way that:

- The content carries a different anonymous fingerprint for each receiver, so that unlawful content redistribution can be tracked; honest receivers stay anonymous.
- The sender does not need to fingerprint and send the content individually to each receiver; one fingerprinting and one unicast transmission by the server to one collaborative receiver are enough to bootstrap the process.
- Receivers are rationally interested to collaborate in forwarding and fingerprinting the content to other interested receivers (we call such rational collaboration co-utility); thanks to anonymous fingerprinting, intermediate receivers do not know the identities of the receivers they are forwarding the fingerprinted content to.

Section 2 gives some background on game theory, co-utility and anonymous fingerprinting. Section 3 describes the protocol and justifies its security. Section 4 argues the rational involvement by peers in game-theoretic terms and shows that our protocol achieves co-utility. Section 5 contains experimental results of a proof of concept. Section 6 summarizes conclusions and future research issues.

2. Background

2.1. Basics of game theory

A game is a protocol between a set of N *players*, $\{P^1, \ldots, P^N\}$. Each player P^i has her own *set of possible strategies*, say S_i. To play the game, each player P^i selects a strategy $s_i \in S_i$. We use $s = (s_1, \ldots, s_N)$ to denote the vector of strategies selected by the players and $S = \Pi_i S_i$ to denote the set of all possible ways in which players can pick strategies.

The vector of strategies $s \in S$ selected by the players determines the outcome for each player, which can be a payoff or a cost. In general, the outcome will be different for different players. To specify the game, we need to give, for each player, a preference ordering on these outcomes by giving a complete, transitive, reflexive binary relation on the set of all strategy vectors S. The simplest way to assign preferences is by assigning, for each player, a value for each outcome representing the payoff of the outcome (a negative payoff can be used to represent a cost). A function whereby player P^i assigns a payoff to each outcome is called a utility function and is denoted by $u_i : S \to \mathbb{R}$.

For a strategy vector $s \in S$, we use s_i to denote the strategy chosen by player P^i and s_{-i} to denote the $(N-1)$-dimensional vector of the strategies played by all other players. With this notation, the utility $u_i(s)$ can also be expressed as $u_i(s_i, s_{-i})$.

A strategy vector $s \in S$ is a *dominant strategy solution* if, for each player P^i and each alternate strategy vector $s' \in S$, it holds that

$$u_i(s_i, s'_{-i}) \geqslant u_i(s'_i, s'_{-i}) \quad (1)$$

In plain words, a dominant strategy s is the best strategy for each player P^i, independently of the strategies played by all other players.

A strategy vector $s \in S$ is said to be a *Nash equilibrium* if, for all players P^i and each alternate strategy $s'_i \in S_i$, it holds that

$$u_i(s_i, s_{-i}) \geqslant u_i(s'_i, s_{-i})$$

In plain words, no player P^i can change her chosen strategy from s_i to s'_i and thereby improve her payoff, assuming that all other players stick to the strategies they have chosen in s. A Nash equilibrium is self-enforcing in the sense that once the players are playing such a solution, it is in every player's best interest to stick to her strategy. Clearly, a dominant strategy solution is a Nash equilibrium. Moreover, if the solution is strictly dominant (*i.e.* when the inequality in Expression (1) is strict), it is also the unique Nash equilibrium. See [26] for further background on game theory.

2.2. Co-utility

We recall here the co-utility paradigm, which we introduced under the name general coprivacy in [16,17]. The following definition is simpler but equivalent to the one used in our previous papers.

Definition 1 (*Co-utility*). Let Π be a game with self-interested, rational players $P^1, \ldots, P^N$, with $N > 1$. Game Π is said to be *co-utile* with respect to the vector $U = (u_1, \ldots, u_N)$ of utility functions if there exist at least two players P^i and P^j, having strategies s^i and s^j, respectively, such that: (i) s^i involves P^i expecting co-operation from P^j; (ii) s^j involves P^j co-operating with P^i; (iii) (s^i, s^j) is an equilibrium for P^i and P^j in terms of u_i and u_j, respectively. In other words, there is co-utility between P^i and P^j, for some $1 \leqslant i, j \leqslant N$ with $i \neq j$, if the best strategy for P^i involves expecting co-operation from P^j and the best strategy for P^j is to co-operate.

If the equilibrium in Definition 1 is a Nash equilibrium, we have *Nash co-utility*. If the utility functions U in Definition 1 only consider privacy, co-utility becomes the plain coprivacy notion introduced in [16,17]; if utilities only consider security, we could speak of co-security; if they only consider functionality, co-utility becomes co-functionality.

2.3. Anonymous fingerprinting

Let $D_0 \in \{0,1\}^*$ denote some digital content (bit-string) some of whose bits can be changed in such a way that (i) the result remains "close" to D_0 (where "close" means "with a similar utility"), but (ii) without knowing which particular bits were changed, altering a "good portion" of these bits is impossible without rendering the content useless. The changed bits are usually called a mark or watermark; if bits are changed differently for each user receiving the content, the mark can also be called fingerprint. The algorithm used to embed a mark while satisfying the previous two conditions is called a watermarking algorithm; to embed a fingerprint can also be termed "to fingerprint". The second requirement above is actually the marking assumption stated in [6].

As mentioned in the introduction above, the type of fingerprinting relevant to our paper is anonymous fingerprinting. The first anonymous fingerprinting proposals relied on unspecified multiparty secure computation protocols [27,13]. In [14], an anonymous fingerprinting protocol completely specified from the computational point of view and based on committed oblivious transfers was presented. In [15], the tamper-proofness of a smart card on the buyer's side was used to simplify anonymous fingerprinting. More recent anonymous fingerprinting schemes rely on the homomorphic properties of public-key cryptography [20,30,21,22,25,29,28]. These schemes allow embedding the fingerprint in the encrypted domain. The buyer sends her encrypted fingerprint to the merchant who embeds it by operating with the encrypted content using the public key of the buyer. The resulting encrypted and fingerprinted content is sent to the buyer who can decrypt it using her private key. This way, only the buyer has access to the

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005

decrypted fingerprinted content. However, these schemes are inefficient in practice, since public-key encryption expands the data, thereby increasing the communication bandwidth required by such homomorphic protocols [19].

In [7], a different system based on group signatures was proposed, but it requires bit commitment. Each bit of the fingerprint is committed to a merchant and a zero-knowledge proof is required, which implies a considerable overhead and bandwidth consumption. Hence, it is difficult to implement this system in practice.

A different approach is proposed by Bo et al. [5], who claim that the system efficiency is enhanced due to the suppression of zero-knowledge proofs. In addition, this scheme allows the use of any available embedding system since it does not rely on public key cryptography for this step. However, this system requires the execution of secure two-party computation schemes between the merchant and the buyer.

In [19] an anonymous fingerprinting protocol is proposed that is based on any secure watermark embedding scheme, provided that the watermark embedder offers a certain level of security and that "partially encrypted" watermarks can be detected in a marked piece of content. This approach would reduce the computational costs required by homomorphic cryptography, but the existence of a secure watermarking scheme with the required properties has not been proven so far. Finally, the scheme of [8] proposes another alternative in which the computational burden of public key cryptography is transferred from the buyer side to powerful servers in other participants of the protocol.

All the systems referred above share a common drawback: the computational and communicational burdens for the merchant are quite high, due to the use of at least one of the following highly demanding technologies: public-key cryptography, bit commitment schemes, zero-knowledge proofs, secure multiparty computation schemes or secret sharing. In addition, some of them can only work with watermarking technologies which are not among the most robust and secure ones or even rely in some watermarking system for which no proof of existence has been provided yet.

In this paper, we seek to mitigate the above performance shortcomings. Our protocols can be used with any of the above anonymous fingerprinting schemes. We borrow from [7] the following generic model of an anonymous fingerprinting protocol.

Definition 2. An anonymous fingerprinting scheme involves a merchant, a buyer and a registration center. Let c denote the maximal size of a collusion of buyers against which the scheme is secure. An anonymous fingerprinting scheme consists of the following five procedures.

FKG-RC: A probabilistic key setup algorithm for the registration center. Its outputs are the center's secret key x_C and its public key y_C, which is published in an authenticated manner.

FReg: A probabilistic two-party protocol (FReg-RC, FReg-B) between the registration center and the buyer. Their common input is the buyer's identity ID_B and the center's public key y_C. The center's secret input is its secret key x_C. The buyer's output consists of some secret x_B and related information y_B. The center obtains and stores y_B and ID_B.

FPrint: A two-party protocol (FPrint-M, FPrint-B) between the merchant and the buyer. Their common input consists of y_C. The merchant's secret input is D_0 and a transaction number j and her output is a transaction record t_j. The buyer's secret input is x_B and y_B, and her output consists of a copy $D_B \in \mathcal{D}$, where $\mathcal{D}$ is the set of all close copies of D_0.

FRec: This may be a protocol or an algorithm whose purpose is to recover the identity of a/the fraudulent buyer responsible for the redistribution of a version $\widetilde{D} \in \mathcal{D}$:

- It is a two-party protocol between the merchant and the registration center if the merchant needs the help of the registration center. The merchant's input is a copy $\widetilde{D} \in \mathcal{D}$, all transaction records t_i and perhaps the original content D_0. The center's input consists of its secret key x_C and its list of y_B's and ID_B's. The merchant's output is a/the fraudulent buyer's identity together with a proof p that this buyer indeed bought a copy of D_0, or $\perp$ in case of failure (*e.g.*, if more than c buyers colluded to produce $\widetilde{D}$).
- It is an algorithm run by the merchant alone if the merchant can determine a/the fraudulent's buyer identity with just $\widetilde{D}$, all transaction records t_i and perhaps the original content D_0.

Whether the original content D_0 is needed for identity recovery depends on the underlying watermarking method used to embed the fingerprint in the content: a watermarking method is said to allow blind detection if only the marked content $\widetilde{D}$ and the embedded mark contained in the transaction record are needed for recovery; methods which need also D_0 are called informed watermarking. In return for their smaller flexibility, informed methods tend to be more robust to content manipulation; see Chapter 2 of [11] for a more detailed discussion.

FVer: A verification algorithm, that takes as input the identity ID_B of an accused buyer, the public key y_C of the registration center, and a proof p, and outputs 1 iff the proof is valid.

The solution in [7] guarantees the following properties:

Correctness:	All protocols terminate successfully whenever players are honest (no matter how other players behaved in other protocols).
Anonymity and unlinkability:	Without obtaining a particular D_B , the merchant –even when colluding with the registration center– cannot identify a buyer (anonymity). Furthermore, the merchant is not able to tell whether two purchases were made by the same buyer (unlinkability).
Protection of innocent buyers:	No coalition of buyers, the merchant, and the registration center is able to generate a proof $\tilde{p}$ such that $\mathsf{FVer}(ID_B, y_C, \tilde{p}) = 1$, if buyer ID_B was not present in the coalition.
Revocability and collusion resistance:	Any collusion of up to c buyers aiming at producing a version $\widehat{D} \in \mathcal{D}$ from which none of them can be re-identified will fail: from $\widehat{D}$ the merchant will obtain enough information to identify at least one collusion member.

3. The protocol

Assume that P^0 has a content D_0 to be multicast and fingerprinted for each receiver. Let $P^1, \ldots, P^N$ be the receivers interested

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005

in that content. Then the following protocol can be used to distribute the fingerprinting task.

Protocol 1 (Distributed multicast fingerprinting).

(1) P^0 and P^1 run an anonymous fingerprinting scheme conforming to the model described in Section 2.3 and having the following features: (i) the underlying watermarking is *blind*; (ii) FRec is a protocol which needs the help of the registration center (who is assumed to be trusted). After running FReg and FPrint, P^1 obtains a fingerprinted copy D_{01} of the content and P^0 obtains a transaction record $t_{0,1}$ (partially or totally consisting of information input by P^0 herself like the transaction number).
(2) **For** $i := 1$ **to** $N-1$:
   (a) P^i and P^{i+1} engage in the same anonymous fingerprinting scheme described above whereby P^{i+1} obtains a fingerprinted version $D_{012\ldots(i+1)}$ and P^i obtains a transaction record $t_{i,i+1}$ (partially or totally consisting of information input by P^i herself like the transaction number);
   (b) P^i sends $t_{i,i+1}$ to P^0.

Some observations on Protocol 1 are in order:

- The need for blind watermarking and for P^i to return the transaction record to P^0 are justified in Section 3.1 below. Also, we justify, in that same section, that FRec has to be a protocol needing the collaboration of a trusted registration center, rather than an algorithm run by P^0 alone.
- Each player P^i may engage in anonymous fingerprinting with additional players other than P^{i+1}. However, for the sake of simplicity and without loss of generality, we ignore such additional transmissions in the above protocol. In the protocol above the multicast tree for the N receivers has depth N: it is actually a line.
- We are implicitly assuming that the underlying watermarking scheme used to embed the fingerprints is such that N or more successive fingerprints can be embedded in D_0 in such a way that:

1. It still holds that the resulting $D_{01\ldots N}$ is close to D_0, that is $D_{01\ldots N} \in \mathcal{D}$.
2. Embedding a new fingerprint does not destroy the previously embedded fingerprints. In fact, this results from the aforementioned marking assumption and the previous assumption on the "closeness" of D_0 and $D_{01\ldots N}$.

- We are not assuming any control on the value of N by P^0. In line with the previous remark, we assume that the depth N will not grow to the point of causing $D_{01\ldots N} \notin \mathcal{D}$. This is a self-enforcing policy: no one is interested in a perceptually bad version of the content. If the number N of players interested in the content turned out to be greater than the number N' of successive embeddable fingerprints, a possible solution is to use a multicast tree for the N receivers whose depth is $N' < N$. This means that some players engage in anonymous fingerprinting with more than one other player.

Note 1 (*On content payment*). Our protocols do not explicitly consider payment by the content receivers to P^0. Our main focus is on fingerprinted multicast rather than on content sale. However, an easy way to force the receivers to pay for the received content would be to encrypt parts of it: the receivers would then need to pay to P^0 to get the decryption key. Payment by each receiver could be anonymous (*e.g.* [10]) and it could specify a receiver's temporary alias e-mail address to which P^0 should send the decryption key. Payment to P^0 could be sent by each receiver P^i together with a transaction number tn_i provided to P^i by P^{i-1} and obtained by P^{i-1} as a one-way hash function of the transaction record $t_{i-1,i}$. In this way, P^0 would be able to associate each payment with a particular transaction record. Since a key is much shorter than the content, sending a key in unicast to each receiver should not pose bandwidth problems at P^0. Of course, partial content encryption means that fingerprinting during the redistribution chain would have to be limited to the unencrypted parts: ciphertext cannot be fingerprinted, because doing so would render decryption impossible. If a receiver P^i leaked her received key to P^{i+1}, then the latter player and all players P^j with $j \geqslant i$ would be able to decrypt the content for free. However, the fact that the receivers stay anonymous to each other discourages this colluding behavior: P^i has no particular incentive to leak her key to unknown peers. Furthermore, if P^j uses a leaked key to unlawfully decrypt the content, her penalty in case of redistribution will increase (see Note 2 below); hence, skipping payment at least strengthens redistribution avoidance by peers.

3.1. Security and privacy analysis

We define security in Protocol 1 as the ability of P^0 to trace the redistributor in case of detecting unlawful redistribution of the content.

When P^0 detects a redistributed copy $\widetilde{D} \in \mathcal{D}$, all P^0 needs to do is to run the following protocol.

Protocol 2 (Redistributor identification).

(1) Let $T = \{t_{0,1}, t_{1,2}, \ldots, t_{N-1,N}\}$ be the set of transaction records received by P^0 after Protocol 1.
(2) Set $i := N-1$ and *recovered* := **false**.
(3) **While** *recovered* = **false do**
   (a) Run, with the registration center, the protocol FRec with inputs the redistributed content $\widetilde{D}$ and $t_{i,i+1}$.
   (b) **If** the identity of P^{i+1} can be successfully recovered **then** *recovered* := **true else** $i := i-1$.
(4) Output the identity recovered for P^{i+1} as the redistributor's identity.

Some remarks on the above identity recovery process follow:

- The anonymous fingerprinting scheme used must be based on an underlying blind watermarking method. Indeed, by construction of Protocol 1, unless the redistributor is P^1, P^0 does not know the original unmarked content corresponding to $\widetilde{D}$. Imagine that the redistributor is P^{i+1} with $1 \leqslant i < N$. In that case, $\widetilde{D} = D_{01\ldots(i+1)}$ and the original unmarked content is $D_{01\ldots i}$, only known to P^i, not to P^0.
- P^0 wants to obtain the identity of the *last* player who fingerprinted the redistributed content: trying first P^N, then P^{N-1}, and so on, ensures that P^0 will only need to obtain the identity of *one* player, namely the dishonest one. Also, since FRec needs the help of the trusted registration center, the latter can be trusted to help P^0 in recovering only one identity; this preserves the privacy of honest players. Trust here is important, because an untrusted registration center could not just reveal more than one identity, but also frame honest buyers by revealing their identities as if they were redistributors. A way to relax the trust assumption is to use several registration centers rather than one; then the majority answer they give in FRec is probably right: if most registration centers are honest and they coincide in accusing a certain buyer, this buyer is probably the dishonest one.
- Protocol 2 requires to start the search from the most recent transaction record and proceed backwards. So P^0 must have some way to determine the order of the transactions. A first approach could be for P^0 to store the reception time for each of the transaction records. The use of reception times can lead to errors if the real ordering of the transactions does not match

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005

the order of reception of the transaction records. However, if an honest receiver P^j is deemed guilty, she will be able to prove her innocence by showing a transaction record $t_{j,j+1}$ that, together with $\widetilde{D}$, allows the registration center to recover the identity of P^{j+1}. Another approach to avoid misidentifications would be to require P^j to append a timestamp to the transaction record $t_{i,i+1}$ so that P^0 could reconstruct the transaction ordering.

In Protocol 2 P^0 has in principle all transaction records, because, if Protocol 1 is correctly followed, after anonymous fingerprinting between P^i and P^{i+1}, P^i sends the resulting transaction record to P^0, for all i. However, Protocol 2 works even if some peers fail to send the transaction record to P^0 in Protocol 1. Assume that P^i and P^{i+1} engage on a transfer of content and that this transfer takes place without any fingerprinting or without P^i sending the resulting transaction record to P^0. Since P^0 does not have the transaction record for the transfer to P^{i+1}, P^0 will not be able to identify P^{i+1} in case P^{i+1} performs unauthorized redistribution. By following the chain of transaction records upwards, at some time P^0 will test if the redistributed copy found can be related to P^i. Hence, P^0, together with the registration center, will be able to obtain P^i's identity and P^i will be found guilty. Note that P^i is indeed guilty for not having correctly followed Protocol 1 and thus can be held liable for the redistribution performed by the (anonymous) P^{i+1}.

Not sending the transaction record to P^0 is therefore risky. A peer P^i who acts this way is implicitly accepting liability for any potential unathorized actions performed by any peer down the chain, that is any peer P^j with $j > i$. If there is another peer $P^{j'}$ with $j' > i$ sending a transaction record to P^0, then P^i will only be held liable for what peers P^j with $i < j < j'$ do.

Note 2. In case of content payment (see Note 1 above), if all receivers are honest, P^0 should receive payment associated to every transaction record. Let us assume that some receiver P^j obtains the decryption key without having paid for it. In this case, P^0 will not have any payment associated to $t_{j-1,j}$. Up to here, no action can be taken by P^0: first, because P^j is anonymous; second, because P^0 cannot prove that P^j actually decrypted the content. However, imagine further that a *decrypted* re-distributed copy is detected by P^0 which leads to identification of P^j using Protocol 2. In this case, P^0 can take action against P^j based on a double offense: illegal redistribution and lack of payment.

4. Rational involvement of players: co-utility

In this section, we show how to motivate the players in Protocol 1 to rationally play their corresponding roles as specified in the protocol. Showing that players have no interest in deviating is especially necessary in a peer-to-peer (P2P) protocol whose correct operation depends on the commitment of peers $P^1, \ldots, P^N$.

P^0 has an obvious interest in following Protocol 1. If she deviates from the protocol by not correctly participating in the anonymous fingerprinting at Step 1, the entire distributed multicast fingerprinting does not even start. Let s^0 the strategy whereby P^0 follows the protocol.

Each peer $P^i \in \{P^1, \ldots, P^N\}$ is assumed to be interested in getting the content; therefore, she will not deviate from correct anonymous fingerprinting with P^{i-1}. However, $P^i \in \{P^1, \ldots, P^{N-1}\}$ has at least four possible strategies with respect to P^{i+1}:

- s_0^i: Correctly follow Protocol 1 by engaging in anonymous fingerprinting with P^{i+1} and returning transaction record $t_{i,i+1}$ to P^0.
- s_1^i: Deviate from Protocol 1 by engaging in anonymous fingerprinting with P^{i+1} but *not* returning $t_{i,i+1}$ to P^0;
- s_2^i: Deviate from Protocol 1 by not engaging in anonymous fingerprinting with P^{i+1} but returning a fake transaction record $t_{i,i+1}$ to P^0.
- s_3^i: Deviate from Protocol 1 by not engaging in anonymous fingerprinting with P^{i+1} and not sending any transaction record.

We next go through an exercise of mechanism design (see Chap. 23 of [23]), to find how Protocol 1 needs to be modified to ensure that, for any player P^i, her rational choice is strategy s_0^i.

4.1. Utility without reward or punishment

Consider the following payoffs:

- d_i: Payoff that P^i derives from obtaining $D_{0..i}$ *without losing her anonymity*. That is, d_i combines the functionality payoff of P^i obtaining the content and the privacy payoff of P^i preserving her anonymity thanks to anonymous fingerprinting with P^{i-1}. If P^i pays a fee or reward for obtaining the content, this fee or reward must be subtracted from the previously defined payoff to get the remaining d_i.
- $-v_i$: Negative payoff (that is, cost) that P^i incurs from engaging in anonymous fingerprinting with P^{i+1}; this cost may be quantified in terms of computation and communication.
- $-w_i$: Negative payoff that P^i incurs from returning the transaction record $t_{i,i+1}$ to P^0; this cost would correspond to the communication cost of sending the transaction record.

If there are no other payoffs (like reward earned for following the protocol or punishment incurred for not following it), the general utility functions of the above strategies are the following:

$$\mathbf{u}_i(s_0^i) = d_i - v_i - w_i$$

$$\mathbf{u}_i(s_1^i) = d_i - v_i$$

$$\mathbf{u}_i(s_2^i) = d_i - w_i$$

$$\mathbf{u}_i(s_3^i) = d_i$$

Clearly, strategy s_3^i has the maximum utility. In these conditions, the dominant strategy solution of the game is

$$(s^0, s_3^1, -, \cdots, -)$$

In plain words, the rational equilibrium is for P^0 to start the distributed fingerprinting and for P^1 to acquire an anonymously fingerprinted D_{01} and exit Protocol 1. Strategies by P^2 to P^N are irrelevant, because their participation in the protocol is prevented by P^1's choice of strategy s_3^1. Clearly, this dominant solution means that players are not rationally interested in correctly following the protocol.

4.2. Utility with reward and no punishment

In an attempt to induce rational players to correctly follow Protocol 1, we can think of introducing a reward for a player who forwards the content to other players. There are two ways to reward player P^i:

- **Centralized reward**: After engaging in anonymous fingerprinting with P^{i+1}, P^i returns the transaction record $t_{i,i+1}$ to P^0 and gets a reward $r_{i,i+1}$ from P^0.
- **Distributed reward**: After engaging in anonymous fingerprinting with P^{i+1}, P^i gets a reward $r_{i,i+1}$ from P^{i+1}, who discounts $r_{i,i+1}$ from her payoff d_{i+1}.

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005

It is not difficult to see that the centralized reward has at least two serious problems:

- P^0 bears all the costs of the rewards. Therefore, if P^0 is selling the content to make a profit, P^0 needs to charge a substantial fee to P^1, her only direct buyer.
- Under the centralized reward, there is incentive for P^i to cheat by playing strategy s_2^i: return a fake transcript to P^0 without actually engaging in anonymous fingerprinting with P^{i+1}.

Hence, the distributed reward seems clearly preferable. In this case, the utility functions of the four strategies of P^i are:

$$\mathbf{u}_i(s_0^i) = d_i + r_{i,i+1} - v_i - w_i$$

$$\mathbf{u}_i(s_1^i) = d_i + r_{i,i+1} - v_i$$

$$\mathbf{u}_i(s_2^i) = d_i - w_i$$

$$\mathbf{u}_i(s_3^i) = d_i$$

If the reward is sufficient to cover the costs of P^i engaging in anonymous fingerprinting with P^{i+1}, that is, if $r_{i,i+1} \geqslant v_i$, then s_1^i has the maximum utility. In these conditions, the dominant strategy solution of the game is

$$(s^0, s_1^1, \ldots, s_1^{N-1}, s_3^N)$$

In plain words, the rational equilibrium is for P^0 to start the distributed fingerprinting and for P^i $(i = 1, \ldots, N-1)$ to acquire and forward a fingerprint to P^{i+1}. The strategy of P^N can only be s_3^N, because P^N is not supposed to forward the content any further.

We have achieved some improvement: players are rationally interested in multicast fingerprinting, but they do not report the transaction records to P^0, which hampers redistributor identification by P^0.

Note 3 (*Implementing reward payment*). A technical issue is how to implement the payment of distributed rewards. Rewards must be paid between players who are anonymous to each other. This precludes the use of P2P payment techniques requiring peer identification, like the cascaded payments proposed in [3]. Workable alternatives are anonymous micropayments between players. The anonymous version of the PayWord scheme described in [31] can be used, for example. The payer sends an initial coupon T_0 of a hash chain (the payword) to the payee, where T_0 has been blindly signed by the payer's bank; then the payer reveals a certain number of successive coupons of the payword (where T_i is a hash pre-image of T_{i-1}) in order to adjust to the amount of the reward to be paid. Double-spending detection mechanisms can be added to the signature on T_0 that cause the payer's anonymity to be lost if he uses the same payword twice (*e.g.* see [10]). If a payer must reward several times the same payee (such a situation can be detected even if players are anonymous to each other, *e.g.* using a cookie mechanism), the payer can keep sending to the payee successive coupons of a payword whose T_0 was exchanged and verified by the payee in a previous transaction. Doing so has the advantage of amortizing over several transactions the computation associated to producing and verifying the signature on T_0.

4.3. Utility with reward and punishment

A punishment mechanism can be added as an incentive for players P^1 through P^{N-1} to return transaction records to P^0.

Let $-p_i$ be the expected negative payoff (punishment) that P^i incurs when accused of redistribution as a result of not having returned a valid transaction record $t_{i,i+1}$. This is actually an *expected* negative payoff, computed as the probability of being accused times the cost of being accused. This negative payoff includes the loss of anonymity (as a result of the redistributor identification algorithm) and may include fines or other penalties (which are easy to apply after anonymity loss).

Under strategy s_1^i, P^i fingerprints the content but she does not return the transaction record. Under s_2^i, P^i forwards the content without fingerprinting and returns a fake transaction record. Under s_3^i, P^i forwards the content without fingerprinting or returning any transaction record. Therefore, in none of those three strategies is a valid transaction record returned, hence all of them incur the punishment. Note that, when running the redistributor identification protocol (Protocol 2), a fake transaction record is treated like a non-existing transaction record: if $t_{i-1,i}$ is the last authentic transaction record received by P^0, then no matter whether P^i sent a fake $t_{i,i+1}$ or no record at all, P^i will be accused of redistribution and hence punished.

We can now recompute the utilities of the four strategies available to P^i:

$$\begin{aligned}
\mathbf{u}_i(s_0^i) &= d_i + r_{i,i+1} - v_i - w_i \\
\mathbf{u}_i(s_1^i) &= d_i + r_{i,i+1} - v_i - p_i \\
\mathbf{u}_i(s_2^i) &= d_i - w_i - p_i \\
\mathbf{u}_i(s_3^i) &= d_i - p_i
\end{aligned} \tag{2}$$

Assume like above that $r_{i,i+1} \geqslant v_i$. Also, assume that $-p_i \leqslant -w_i$, that is, that not returning the transaction record is worse than returning it. With those assumptions,

$$\mathbf{u}_i(s_2^i) \leqslant \mathbf{u}_i(s_3^i) \leqslant \mathbf{u}_i(s_1^i) \leqslant \mathbf{u}_i(s_0^i)$$

so that s_0^i is the strategy with maximum utility. Hence, the dominant strategy solution of the game is

$$(s^0, s_0^1, \ldots, s_0^{N-1}, s_3^N)$$

In plain words, the rational equilibrium is for P^0 to start the distributed anonymous fingerprinting and for P^i $(i = 1, \ldots, N-1)$ to correctly follow Protocol 1. The strategy of P^N can only be s_3^N, because P^N is not supposed to forward the content any further.

With the proposed modifications, we have succeeded in inducing a rational behavior in the players that causes them to correctly following the intended multicast fingerprinting protocol.

Lemma 1. *With the utility functions defined in Eqs. (2), there is co-utility between P^i and P^{i+1} for $i \in \{0, \ldots N-1\}$.*

Proof. With the utilities in this section, the dominant strategy solution has been shown to be the one in which every player P^i plays s_0^i. Note that s_0^i is precisely the strategy which yields the best possible payoff d_{i+1} for P^{i+1}: indeed, P^{i+1} obtains the content while preserving her anonymity (thanks to anonymous fingerprinting). Now, whatever the strategy chosen by P^{i+1}, the utility function $\mathbf{u}_{i+1}$ monotonically increases with d_{i+1}.

Hence, the best strategy for P^i results in enhanced utility for P^{i+1}, whatever P^{i+1}'s strategy. The lemma follows. □

5. Proof of concept

The objective of this section is to provide a proof of concept to show that the proposed protocol can be put into practice with

existing watermarking technologies. This implies that the results of the paper are not merely theoretical: a practical application of the discussed protocol is implementable.

The protocol described in Section 3 has been realized using the audio watermarking scheme described in [24]. This watermarking scheme can be used as a building block for anonymous fingerprinting and it satisfies the requirements listed in Section 3. The scheme is blind, so that it is possible to extract the embedded mark from a marked audio object without knowing the original unmarked audio object. Also, the scheme tolerates embedding several successive fingerprints without significant damage to the content utility or the previous fingerprints.

The scheme uses a double embedding strategy:

- A time-domain synchronization watermark ("SYN") is embedded for fast search of the information watermark position;
- A frequency-domain information watermark is embedded next to the SYN marks.

This double embedding strategy makes it possible to embed the transaction records $t_{i,i+1}$ and the receiver related information y_B in different domains, as depicted in Fig. 1. The transaction records can be embedded as synchronization marks in the time domain with different bit strings, and the related information y_B can be embedded more robustly in the frequency domain. This scheme has the additional advantage of a very fast search of transaction records; extracting an embedded transaction record from a portion of audio takes less time than playing that portion.

In order to preserve anonymity and make the registration center necessary for redistributor identification (as mandated by Protocol 2), we let y_B be the receiver identity encrypted under the registration center's public key. To obtain unlinkability, a random nonce is appended to the receiver's identity before encrypting it under the registration center's public key. Embedding, next to y_B, a hash of x_B (or x_B encrypted with the public key of the receiver) has the additional advantage of thwarting a collusion of the sender P^0 and the registration center, who would not be able to produce a correctly fingerprinted copy of the content corresponding to any receiver.

If the sender P^0 finds a version of the audio file illegally redistributed on the Internet, she can search for the transaction records in the time domain (fast search) and then extract the information y_B related to the malicious receiver. This information (y_B) will then be sent to the registration authority in order to identify the illegal redistributor.

Within this framework, two different experiments have been performed for a set of six players: P^0 (sender) and P^1, P^2, P^3, P^4, P^5 (receivers). Two approaches have been compared:

Centralized unicast: The sender and each receiver separately engage in an anonymous fingerprinting protocol to generate different fingerprinted copies $D_{01}, D_{02}, D_{03}, D_{04}$ and D_{05} from the original content D_0.

Distributed multicast: The sender P^0 and the receiver P^1 engage in an anonymous fingerprinting protocol to generate D_{01} from the original content. P^0 generates the transaction record $t_{0,1}$ to be embedded in the time domain. Subsequently, P^i and P^{i+1}, for $i = 1, 2, 3, 4$ engage in the same fingerprinting protocol to generate D_{012}, D_{0123}, D_{01234} and D_{012345}; the corresponding transaction records $t_{i,i+1}$, for $i = 1, 2, 3, 4$ are returned to the sender by P^i (who plays the merchant role in the transaction between P^i and P^{i+1}). P^0 keeps a sorted list of the transaction records. Each copy of the digital content carries the fingerprints corresponding to all previous receivers.

In order to preserve the privacy of the input and output information ($t_{i,i+1}$, $D_{01\ldots i}, y_{i+1}, D_{01\ldots i+1}$) in each execution of the fingerprinting scheme, a secure two-party computation protocol, as those presented in [9,5], is required as a building block of the anonymous fingerprinting protocol. In the distributed multicast protocol, only P^0 has access to the original content D_0, whereas only P^1 knows the information y_1 which is embedded in D_{01}. The same applies for the subsequent executions of the fingerprinting protocol. This secure multiparty computation approach introduces an additional overhead which can be shown to be poly-logarithmic in the number of receivers and the size of the circuit (or algorithm) needed to implement the scheme [12].

The original content (D_0) used in the experiments is the 30-second violoncello file ("vioo10_2.wav") available from the Sound Quality Assessment Material corpus [1]. In the centralized unicast protocol, the file is divided into 10-second segments and an instance of the fingerprint is embedded into each segment. The fingerprint consists of an information watermark (y_B) embedded in the FFT domain preceded by a transaction record embedded in the time domain (see Fig. 1).

It must be pointed out that the watermarking scheme [24] allows embedding a long bit string. In addition, to enable even longer embedding capacity, the information could be encrypted, sent to the merchant, and the key could be embedded instead of y_B. Using the parameters specified in the experimental section of [24], each instance of the watermark requires 1.30 seconds of audio. Hence, each 10-second segment allows the inclusion of up to 7 different watermarks, more than enough for the 5 levels of embedding considered in the experiments reported here.

With a longer audio content, the segments could be chosen long enough to allow for, say, 10 different watermarks. In that case, if every receiver could engage in the fingerprinting protocol with up to f other receivers, the number of potential receivers could increase in powers of f at each step; in the 10-th step, up to f^{10} receivers would be reached. It is easy to see that a value $f = 9$ would be enough to cover a number of receivers equal to half of the earth's population.

The centralized unicast and distributed multicast protocols above have been compared in terms of: (i) CPU time and bandwidth required from the sender P^0; and (ii) the transparency of the resulting fingerprinted content.

In what regards CPU time, the fingerprinting scheme has been tested in a Matlab (interpreted) implementation on a 3.0 GHz (single core) Pentium IV processor with 1 GB of RAM. The overhead of the two-party secure computation scheme has not been taken into account in this simulation, but it can be reckoned to multiply by a constant greater than 1 the CPU time needed for two parties to complete an anonymous fingerprinting; hence this overhead does not influence the following comparison between the centralized unicast and the distributed multicast. In the centralized unicast approach, the sender P^0 uses 9.81 seconds of CPU time to produce the 5 marked copies of the content ($D_{01}, \ldots, D_{05}$) and transmits

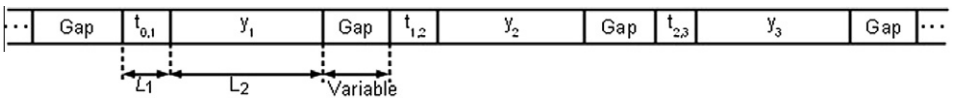

Fig. 1. Embedding strategy.

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005

Table 1
Transparency results.

Protocol	Content	# Fingerprints	ODG
Centralized	D_{01}	1	0.000
	D_{02}	1	0.000
	D_{03}	1	0.000
	D_{04}	1	0.000
	D_{05}	1	0.000
Peer-to-peer	D_{01}	1	0.000
	D_{012}	2	−0.004
	D_{0123}	3	−0.034
	D_{01234}	4	−0.115
	D_{012345}	5	−0.193

26,519,020 bytes of information (5 different versions of the uncompressed audio file). In the distributed multicast, the sender needs to run the fingerprinting scheme just once (with the receiver P^1) taking 1.97 seconds of CPU time and sending 5,303,804 bytes. Hence, from the sender's point of view, distributed multicast consumes just a 20% of CPU time and bandwidth compared to centralized unicast. Since the sender is the bottleneck in centralized unicast, the saving allowed by distributed multicast is relevant.

Regarding transparency, the Objective Difference Grade (ODG) based on the ITU-R Recommendation standard BS 1387 [18,32] has been used. This standard makes it possible to evaluate the transparency of the fingerprinting scheme by comparing the perceptual quality of the marked files with respect to the original content D_0. The ODG values are in the range $[-4, 0]$, where 0 means imperceptible, −1 means perceptible but not annoying, −2 means slightly annoying, −3 means annoying and −4 means very annoying. In order to evaluate the ODG, we have used the Opera software by Opticom [2].

The imperceptibility results are shown in Table 1 for all the files obtained with both the centralized unicast and the distributed multicast. As it can be noticed, the transparency of the five files resulting from the centralized protocol is perfect (ODG = 0), whereas it slowly decreases for each successive receiver in the distributed multicast protocol. However, even with 5 embedded fingerprints, the ODG result is much closer to 0 (imperceptible) than −1 (perceptible but not annoying); hence, even in this worst case, the perceptual quality achieved by the distributed multicast protocol can be regarded as very satisfactory.

Now let us imagine that the receiver P^3 decides not to return the transaction record $t_{3,4}$ to the sender P^0. In this case, if P^0 finds an illegal redistribution of file $D_{012\ldots m}$, P^0 will send the redistributed file and $t_{2,3}$ to the registration center to track the liable receiver, because $t_{2,3}$ is the last transaction record available to P^0. Hence, P^3 will be held guilty of illegal distribution due to her decision of not returning $t_{3,4}$ to the sender.

6. Conclusions and future research

We have described a peer-to-peer protocol for distributed multicast of fingerprinted content which has the interesting properties that:

- Each receiver obtains a different fingerprinted copy of the content which allows the sender to trace redistributors.
- The sender does not need to prepare and send a separate fingerprinted copy to each receiver, so that its computational and bandwidth burden is equivalent to the case of there being a single receiver.
- Receivers rationally co-operate in a peer-to-peer fashion thanks to a system of rewards and punishments which ensures that each receiver's best strategy is to loyally follow the prescribed peer-to-peer multicast protocol; in this respect the protocol is said to be co-utile.

Future research will investigate applications of the proposed peer-to-peer multicast protocol to scenarios other than redistribution control, such as enforcing expiration dates on data items in view of digital forgetting.

Acknowledgments

Thanks go to Jordi Soria for useful observations about the security and privacy of our protocols. This work was partly funded by the European Commission under FP7 projects "DwB" and "Inter-Trust", by the Spanish Government through projects TSI2007-65406-C03-01/03 "E-AEGIS", TIN2011-27076-C03-01/02 "CO-PRIVACY", and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES", and by the Government of Catalonia through grant 2009 SGR 1135. The first author is partly supported as an ICREA-Acadèmia researcher by the Government of Catalonia.

References

[1] Ebu, SQAM – Sound Quality Assessment Material. <ftp://ftp.tnt.uni-hannover.de/pub/MPEG/audio/sqam/> (accessed 02.01.13).
[2] Opera software by Opticom. <http://www.opticom.de/products/opera.html>(accessed 02.01.2013).
[3] G. Arora, M. Hannegham, M. Merabti, P2P commercial digital content exchange, Electronic Commerce Research and Applications 4 (2005) 250–263.
[4] G.R. Blakley, C. Meadows, G.B. Purdy, Fingerprinting long forgiving messages, Advances in Cryptology-CRYPTO'85, LNCS, vol. 218, Springer, Berlin, Heidelberg, 1986. pp. 180–189.
[5] Y. Bo, L. Piyuan, Z. Wenzheng, An efficient anonymous fingerprinting protocol, Computational Intelligence and Security, LNCS, vol. 4456, Springer, Berlin, Heidelberg, 2007. pp. 824–832.
[6] D. Boneh, J. Shaw, Collusion-secure fingerprinting for digital data, IEEE Transactions on Information Theory 44 (5) (1998) 1897–1905.
[7] J. Camenisch, Efficient anonymous fingerprinting with group signatures, Advances in Cryptology – ASIACRYPT 2000, LNCS, vol. 1976, Springer, Berlin, Heidelberg, 2000. pp. 415–428.
[8] C.-C. Chang, H.-C. Tsai, Y.-P. Hsieh, An efficient and fair buyer-seller fingerprinting scheme for large scale networks, Computers & Security 29 (2) (2010) 269–277.
[9] D Chaum, I. Damgård, J. van de Graaf, Multiparty computations ensuring privacy of each party's input and correctness of the result, Advances in Cryptology – CRYPTO'87, LNCS, vol. 293, Springer, Berlin, Heidelberg, 1987. pp. 87–119.
[10] D. Chaum, Untraceable electronic cash, Advances in Cryptology – CRYPTO'88, LNCS, vol. 403, Springer, Berlin, 1990. pp. 319–327.
[11] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, second ed., Morgan Kaufmann Publishers Inc., San Francisco CA, 2008.
[12] I. Damgård, Y. Ishai, M. Krigaard, Perfectly secure multiparty computation and the computational overhead of cryptography, Advances in Cryptology – EUROCRYPT 2010, LNCS, vol. 2010, Springer, Berlin, Heidelberg, 2010. pp. 445–465.
[13] J. Domingo-Ferrer, Anonymous fingerprinting of electronic information with automatic identification of redistributors, Electronics Letters 34 (13) (1998) 1303–1304.
[14] J. Domingo-Ferrer, Anonymous fingerprinting based on committed oblivious transfer, Public Key Cryptography – PKC'99, LNCS, vol. 1560, Springer, Berlin, Heidelberg, 1999. pp. 632–632.
[15] J. Domingo-Ferrer, J. Herrera-Joancomartí, Efficient smart-card based anonymous fingerprinting, Smart Card and Advanced Application Conference-CARDIS 98, LNCS, vol. 1820, Springer, Berlin, Heidelberg, 2000. pp. 221–228.
[16] J. Domingo-Ferrer, Coprivacy: towards a theory of sustainable privacy, Privacy in Statistical Databases-PSD 2010, LNCS, vol. 6344, Springer, Berlin, Heidelberg, 2010. pp. 258–268.
[17] J. Domingo-Ferrer, Coprivacy: an introduction to the theory and applications of co-operative privacy, SORT – Statistics and Operations Research Transactions 35 (2011) 25–40 (special issue: Privacy in statistical databases).
[18] ITU-R, Recommendation BS.1387, Method for objective measurements of perceived audio quality, December 1987.
[19] S. Katzenbeisser, A. Lemma, M. Celik, M. van der Veen, M. Maas, A buyer-seller watermarking protocol based on secure embedding, IEEE Transactions on Information Forensics and Security 3 (4) (2008) 783–786.
[20] M. Kuribayashi, On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol, EURASIP Journal on Information, Security 2010 (2010) 1–11.

[21] M. Kuribayashi, H. Tanaka, Fingerprinting protocol for images based on additive homomorphic property, IEEE Transactions on Image Processing 14 (12) (Dec. 2005) 2129–2139.
[22] C.-L. Lei, P.-L. Yu, P.-L. Tsai, M.-H. Chan, An efficient and anonymous buyer-seller watermarking protocol, IEEE Transactions on Image Processing 13 (12) (2004) 1618–1626.
[23] A. Mas-Colell, M. Whinston, J. Green, Microeconomic Theory, Oxford University Press, New York NY, 1995.
[24] D. Megías, J. Serra-Ruiz, M. Fallahpour, Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification, Signal Processing 90 (12) (2010) 3078–3092.
[25] N. Memon, P.W. Wong, A buyer-seller watermarking protocol, IEEE Transactions on Image Processing 10 (4) (2001) 643–649.
[26] N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani (Eds.), Algorithmic Game Theory, Cambridge University Press, New York NY, 2007.
[27] B. Pfitzmann, M. Waidner, Anonymous fingerprinting, Advances in Cryptology-EUROCRYPT'97, LNCS, 1233, Springer, Berlin, Heidelberg, 1997. pp. 88–102.
[28] B. Pfitzmann, A.-R. Sadeghi, Coin-based anonymous fingerprinting, Advances in Cryptology-EUROCRYPT'99, LNCS, vol. 1592, Springer, Berlin, Heidelberg, 1999. pp. 150–164.
[29] B. Pfitzmann, A.-R. Sadeghi, Anonymous fingerprinting with direct non-repudiation, Advances in Cryptology- ASIACRYPT 2000, Springer, Berlin, Heidelberg, 2000. pp. 401–414.
[30] J.P. Prins, Z. Erkin, R.L. Lagendijk, Anonymous fingerprinting with robust qim watermarking techniques, Journal on Information, Security (2007) 2007:20:1–2007:20:7.
[31] R.L. Rivest, A.Shamir, Payword and micromint: two simple micropayment schemes, Technical report MIT LCS, 1995.
[32] T. Thiede, W.C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J.G. Beerends, C. Colomes, PEAQ – the ITU standard for objective measurement of perceived audio quality, Journal of the Audio Engineering Society 48 (1/2) (2000) 3–29.

Please cite this article in press as: J. Domingo-Ferrer, D. Megías, Distributed multicast of fingerprinted content based on a rational peer-to-peer community, Comput. Commun. (2013), http://dx.doi.org/10.1016/j.comcom.2012.12.005