Co-Utile Privacy-Aware P2P Content Distribution

David Megías and Josep Domingo-Ferrer

Abstract Multicast distribution of content is not suited to content-based electronic commerce because all buyers obtain exactly the same copy of the content, in such a way that unlawful redistributors cannot be traced. Unicast distribution has the shortcoming of requiring one connection with each buyer, but it allows the merchant to embed a different serial number in the copy obtained by each buyer, which enables redistributor tracing. Peer-to-peer (P2P) distribution is a third option which may combine some of the advantages of multicast and unicast: on the one hand, the merchant only needs unicast connections with a few seed buyers, who take over the task of further spreading the content; on the other hand, if a proper fingerprinting mechanism is used, unlawful redistributors of the P2P distributed content can still be traced. In this chapter, we describe a co-utile fingerprinting mechanism for P2P content distribution which allows redistributor tracing, while preserving the privacy of most honest buyers and offering collusion resistance and buyer frameproofness.

1 Introduction

If a content is to be distributed to a group of N receivers, one option is for the content sender to engage in N unicast transmissions, one for each intended receiver, and another option is a single multicast transmission to the entire group. Certainly, the multicast option has the advantage of being faster and more bandwidth-efficient from the senders point of view. However, the unicast approach has the strong point

David Megías
Universitat Oberta de Catalunya, Av. Carl Friedrich Gauss 5, E-08860 Castelldefels, Catalonia, e-mail: david.megias@uoc.edu

Josep Domingo-Ferrer
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: josep.domingo@urv.cat

of allowing the sender to fingerprint the content sent to each receiver by embedding a different serial number in each sent copy, with the aim of detecting and tracing unlawful redistribution of the content. Note that the multicast approach does not allow fingerprinting, as all receivers obtain exactly the same content. Hence, the unicast approach, in spite of its inefficiency, seems more suitable when the sender is a merchant selling content and the receivers are buyers.

Peer-to-peer (P2P) distribution of content appears as a third option blending some of the advantages of the unicast and multicast solutions. P2P distribution of all types of files and contents has become extremely popular with the increased bandwidth of home Internet access in the last few years. In addition, P2P file sharing applications are not restricted to this use, and some companies are also exploiting the P2P distribution paradigm as a way of saving server bandwidth and speeding up the downloads of their products (such as multimedia contents and software updates). Indeed, when using a P2P network for content distribution, the merchant only needs to establish direct connections with one or a few seed buyers, say $M \ll N$ buyers, and send them copies. The content is further spread over the P2P network by those seed buyers. The challenge is how to ensure that the P2P spread content is still traceable in case of redistribution.

The type of fingerprinting relevant to this study is anonymous fingerprinting. In anonymous fingerprint schemes, the merchant does not have access to the identities or the fingerprints of buyers, which protects their security and privacy. Initial anonymous fingerprinting proposals depended on unspecified multiparty secure computation protocols [25, 7]. In [8], an anonymous fingerprinting protocol completely specified from the computational point of view and based on committed oblivious transfers was described. In [9], anonymous fingerprinting protocols were simplified under the assumption that a tamper-proof smart card was available on the buyer's side.

Many anonymous fingerprinting schemes exploit some homomorphic property of public-key cryptography [17, 28, 18, 23, 26]. These schemes allow embedding the fingerprint in the encrypted domain (using the public key of the buyer) in such a way that only the buyer obtains the decrypted fingerprinted content. However, developing a practical system using these ideas appears difficult, because public-key encryption expands data and substantially increases the communication bandwidth required for transfers [16]. In [3], a different approach using group signatures was suggested, but this solution requires bit commitment and a zero-knowledge proof, implying a large overhead and high communicational costs. In the proposal of [1], the system's efficiency is enhanced due to the suppression of zero-knowledge proofs and public-key cryptography is not required in the embedding scheme. However, a secure two-party computation protocol is used between the merchant and each buyer to transfer the fingerprinted content. In [16], any secure watermarking scheme (for which no proof of existence is available) may be used to develop an anonymous fingerprinting protocol if the watermark embedder provides a certain level of security. Although the proposed approach avoids the costs of homomorphic cryptography, a practical application of that idea is not presented. Another proposal to reduce the burden of anonymous fingerprinting on the buyer's side is presented in [4], where

powerful servers would perform the most costly parts of the protocols. In any case, all the proposed anonymous fingerprinting systems incur high computational and communicational burdens at the buyer's and/or at the merchant's side, due to the use of some highly demanding technology (public-key encryption of the contents, secure multiparty protocols or zero-knowledge proofs, among others). Some of them also require specific embedding schemes which are not among the most robust or secure ones, or a secure watermarking system that is not proven to exist. In this study, we propose a novel solution to overcome these drawbacks, since the use of public-key cryptography is restricted to the transmission of short bit strings (hashes) and is not applied to the multimedia content itself. In addition, the proposed scheme decentralizes the transmission of the content using a network of peer buyers, thereby reducing the bandwidth needed by the merchant.

We propose a co-utile [10, 12] P2P distribution scheme of fingerprinted content whereby the merchant originates only a set of M seed copies of the content and sends them to M seed buyers. All subsequent copies are generated from the seed copies. Each non-seed buyer obtains her copy of the content by running a P2P purchase software tool. The copy obtained by each buyer is a combination of the copies provided by her sources (parents). The fingerprint of each buyer is thus a binary sequence *automatically* formed as the combination of the sequences of her sources. This peer-to-peer distribution scheme makes it possible for the merchant to save bandwidth and CPU time, while still being able to trace unlawfully redistributed content. Moreover, the co-utility property ensures that, whenever a child buyer can obtain her content from more than one parent, she will do so; it also ensures that parents will be interested in not passing their entire content to a single child buyer (for details on co-utility, see the first chapter of this book and/or the references cited at the beginning of this paragraph).

2 Overview of the proposed scheme

In the proposed P2P scheme for distributing fingerprinted content (see end of previous section), the fingerprints of the buyers do not need to be registered in any way and, thus, all buyers can preserve their privacy as long as no illegal content redistribution occurs. However, when an illegally redistributed file is found, it is possible to link its binary sequence to a particular individual (buyer). As in most fingerprinting applications, in the proposed system "illegal redistribution" means that the buyer redistributes the whole content (file) to a third party who does not purchase it legally, or makes the content available for download in a non-authorized platform (web page, file sharing application, or other) without the copyright owner's explicit permission. The tracing of an illegal redistributor does not need to be particularly fast and legal actions can be taken when the identification is completed.

To satisfy the above conditional privacy, a P2P proxy (or set of proxies) is used to create anonymous connections between buyers such that source and destination buyers do not lose their anonymity. The P2P proxy also sends a transaction record

to a transaction monitor whenever a buyer obtains fragments of the contents from another buyer. The fields of this transaction record are the following:

- An identifier of the purchased contents (a perceptual hash).
- The pseudonyms of the two buyers participating in the transaction, that is, the parent and the child.
- The encrypted fingerprint of the contents obtained by the child from the parent.
- The encrypted hash of the fingerprint of the contents obtained by the child from the parent.
- The time and date of the transaction.

A child is supposed to obtain pieces of the content from several parents, so there will be one transaction record for each parent the child gets pieces from. The purpose of storing the above transaction records at the transaction monitor is to enable tracing of illegal redistributors.

Buyers stay anonymous to each other, but only pseudonymous versus the transaction monitor; however, the transaction record does not specify which fragments come from which buyer, so that the privacy of the buyers' fingerprints is preserved. The encrypted hash is used by the authority in case a buyer intends to cheat the tracing system by showing a different (modified or borrowed) copy of the content. Since the transaction monitor only records a hash of the true fingerprint and buyer pseudonyms that are not linked to specific fragments of the content, no coalition of the transaction monitor, the seller or other buyers can be used to frame an innocent buyer (by unjustly accusing her).

In order to carry out an *a posteriori* identification of redistributors, a basic traitor protocol is run taking the fingerprint of the illegally redistributed content. In addition, the registered hashes of the fingerprints are enough to discourage buyers from cheating the tracing system by using borrowed or altered copies of the contents. On the other hand, the privacy of the honest users is preserved and their fingerprints remain private.

In addition to attractive privacy properties, it will be shown that, in practice, the proposed scheme offers good security properties, namely collusion resistance vs dishonest colluding buyers (if a particular anti-collusion strategy is used) and buyer frameproofness vs a malicious merchant.

3 P2P distribution of recombined fingerprinted contents

The basis of P2P content distribution is that the shared contents are distributed by some users to others. As soon as some fragments of the content are received, destination users become sources for others. A file is thus obtained by joining the fragments of several sources together. Typically, a hash value of the shared content is used by P2P clients to identify files. Two files having the same hash value are considered equal. The upload/download process of a file from different sources is depicted

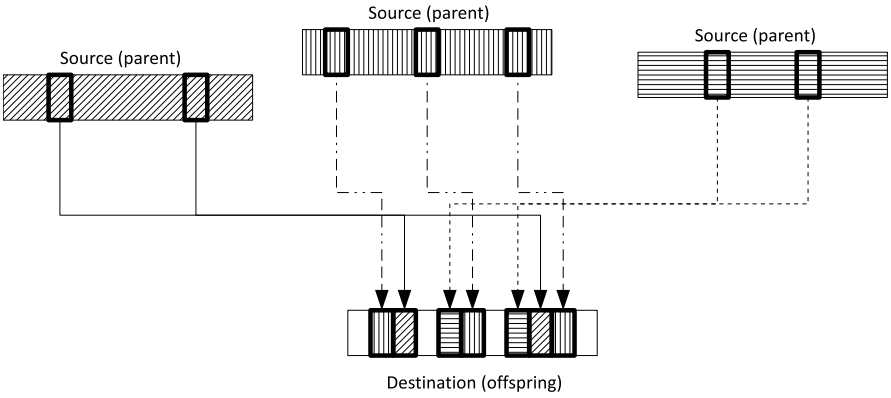


Fig. 1 Upload/download of the content (mating process)

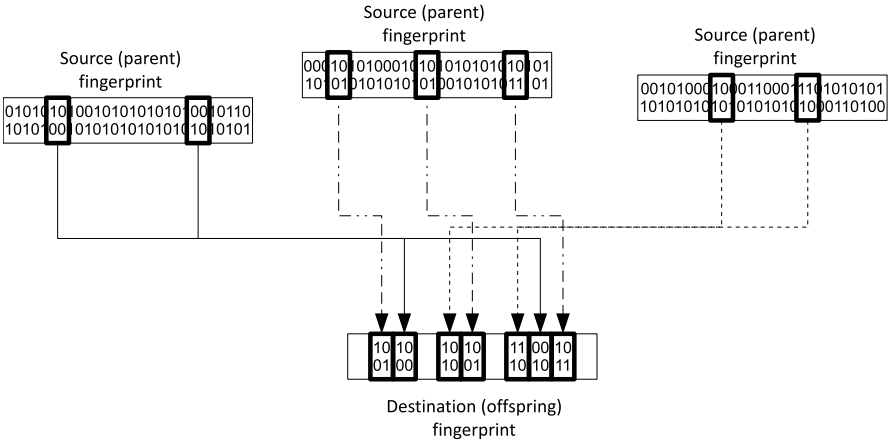


Fig. 2 Automatic recombined fingerprint construction

in Figure 1. In this figure, the destination obtains fragments from three different sources that are joined together to form the content.

3.1 Mating approach for fingerprinting

In this section, we introduce a novel concept of automatic binary fingerprints partly inspired on biological mating and inheritance. The relationship between biology and the scheme in this work is rather weak and, thus, we refrain from calling our scheme "genetic". However, some biological analogies are highlighted in this section to introduce the basis of the suggested scheme.

In this work, fingerprints are constructed as binary sequences and each bit might be considered as the counterpart of the nucleotides of a DNA sequence. This is similar to the approach taken in Genetic Algorithms [15] for solving optimization problems. Just like DNA sequences are formed by different genes which encode a given protein, the binary fingerprints used in this work are formed by (fixed-size) segments that may be considered as analogs of genes. When a buyer obtains a copy of a P2P-distributed content using some specific software, the binary fingerprint of her copy is a combination of the segments of the sources of the content (referred to as "parents" from the biological analogy). In this case, the number of parents of some content does not have to be exactly two as in the natural world. Hence, the mating process in the suggested fingerprinting scenario must be understood in a generalized sense, not limited to two parents. Fingerprints can be considered as being "automatically generated" from the fingerprints of the parents. Despite this "automatic generation" of fingerprints, the constructed sequences are still valid for identification purposes. In order to identify the culprit of an illegal redistribution, a identification process must be carried out. The identification process is performed with the help of the fingerprint of the illegally re-distributed copy.

3.2 Requirements on fingerprint embedding

If the binary fingerprints described in Section 3.1 are to be found in a P2P-distributed content, an embedding method is to be used for the M seed buyers. It is enough to embed randomly generated fingerprints for the seed buyers such that their pairwise correlation is low. This embedding scheme must fulfill the following conditions:

1. The embedded fingerprint must be a binary sequence spread along the whole content (file). Furthermore, the fingerprint must be separated into pieces which are embedded into different blocks (or fragments) distributed by the P2P software. These (fixed-sized) pieces of the content contain a full segment of the fingerprint. For example, if the P2P software uses 32-KB (kilobyte) fragments, each segment of the fingerprint should be embedded into one of these fragments and the fingerprint extraction method must be robust against fragmentation in 32-KB units, as long as the beginning and the end of the fragments are respected. This process is illustrated in Figure 2.
   Note that this is not always possible with non-block-based embedding schemes. An example of block-based audio watermarking system which may be used for fingerprinting in this scenario is presented in [20].
2. Even if the versions of the content obtained by different buyers will not be bitwise identical (the fingerprints embedded into the buyers' copies will differ as a consequence of the P2P distributed download), these versions should be "perceptually" identical, because the distributed content must have the same high quality for all buyers. This means that a standard hash function which produces different hash values even after a single bit change would not be useful in this application.

A perceptual hash function [6] for which the same hash value is obtained for different (perceptually identical) versions of the same content would be required if hash values are used for indexing in the P2P distribution software.

If the previous two conditions hold, fingerprinting occurs in an automatic way as contents are obtained by buyers from different sources. No additional overhead for embedding is required. Note that the above automatic fingerprinting requires more than one content source for each buyer to exist: in case of a single source, the fingerprint would be identical for both the source and the buyer. Although some segments of the fingerprint could be modified by running the embedding method in these buyer-to-buyer transfers, this would reduce the simplicity of our proposal. The simplest solution is to enforce at least two parents for each buyer, and this is the choice made in this work.

Even if two children have exactly the same parents, since fragments are picked up randomly from parents, the probability that both children have the same fingerprint is negligible.

We assume that all seed buyers are chosen with equal probability. Thus, on average, all of them contribute with a similar number of fragments to the new buyers. In addition, all subsequent buyers also engage in P2P transfers by being parents of new buyers. Scalability is thus guaranteed by the distribution system even if the number of parents per buyer is small, since the merchant only needs to feed the seed buyers. As new buyers obtain the content, the connections with the seed buyers may become less relevant, since the new buyers become new content sources.

3.3 Co-utility in parent-child relationship

If it can be enforced that there be at least two parents for each buyer, this is the simplest and most effective solution, because, as said above, fingerprinting is automatic in this case. Fortunately, it turns out that it is in the selfish interest of a child buyer to obtain her content from more than one parent, and it is in the selfish interest of a parent buyer to split her content into more than one child buyer:

- If a child obtains her entire content from a single parent then, her fingerprint will be the same as her parent's fingerprint. Then, if the parent happens to illegally redistribute the content, the child risks being unjustly accused of redistribution (see Section 5 below). Obtaining the content from several parents is a simple and automatic way to avert that risk.
- If a parent sends her entire content to a single child, her child will inherit the parent's fingerprint. Hence, if the child happens to illegally redistribute the content, the parent risks being unjustly accused of redistribution. Splitting the content among several children is the best option for the parent.

The best strategy to preserve one's own privacy is to act in such a way that someone else's privacy is protected. In game-theoretic terms, the vector of strategies *multiple children*, *multiple parents* is a Nash equilibrium between parent and child.

The co-utility property (see the first chapter of this book or [10, 12]) ensures that, whenever a child buyer can obtain her content from more than one parent, she will do so; it also ensures that parents will be interested in not passing their entire content to a single child buyer. The latter condition can be easily enforced by the P2P distribution software. For example, when a parent is sending the content through a proxy, it can be enforced that the connection be closed as soon as a given threshold fraction (*e.g.* 50 or 60%) of the content has been sent. The software can also block any further attempt by the proxy to establish a connection with the same parent for the same content (for some given time window). Each proxy should be forced to choose at least two different parents.

The co-utility analysis given in this section holds for all of our P2P systems described in [21, 22, 19].

3.4 Building blocks and notation for transaction monitoring and content authentication

In order to design protocols for the different steps of the distribution system, the following building blocks are required:

- Public-key cryptography is required in different steps below. Let $E(\cdot, K)$ be the encryption function using the public key K and $D(\cdot, K^s)$ be the decryption function using the private key K^s, required to decrypt a content encrypted using E and K, *i.e.* $D(E(x,K),K^s) = x$.
- In particular, the transaction monitor uses the following pair of public and private keys: (K_c, K_c^s). Also, each peer node in the network is supposed to have a public key and a private key. A pair of decryption and encryption keys, K_m and K_m^s respectively, for the merchant is also used for the authentication of the fragments.
- For each segment of the fingerprint g_i, a hashing function h produces a 1-bit hash $h(g_i)$. Let h_f be the (ordered) concatenation of the hashes of all segments, called "fingerprint's hash" hereafter. Hence, h_f is constructed as

$$h_f = h(g_1)|h(g_2)|\ldots|h(g_l),$$

  where l is the number of segments of the fingerprint and "|" stands for the concatenation operator.
- An extraction function exists to obtain the fingerprint from a content. This function receives, as input parameters, the fingerprinted content and a secret extraction key K^e only known by the merchant. This key will be required by the authority to trace an unlawful distribution.

4 The P2P distribution protocol

In this section, we present the P2P distribution protocol and how transfers between peer buyers are anonymized. It is based on our ideas in [21, 22, 19].

4.1 System model

The participants in the proposed fingerprinting system are the following:

- Merchant: he distributes copies of the content legally to the seed buyers. Each fragment of the content contains a different segment of the fingerprint embedded into it. The segments have low pair-wise correlations.
- Seed buyers (B_i for $i = 1, \ldots, M$): they receive fingerprinted copies of the contents from the merchant that are used by the P2P distribution system to bootstrap the system. They can be either real or dummy buyers.
- Other buyers (B_i for $i = M+1, \ldots, N$, with $M \ll N$): they purchase the content and obtain their fingerprinted copies from the P2P distribution system. The content is assembled from fragments obtained from different "parents". Anonymous connections with peer buyers are provided by means of proxies.
- Proxies: they provide anonymous communication between peer buyers by means of a specific protocol analogous to Chaum's mix networks [5] (see below).
- Transaction monitor: it keeps a transaction register for each purchase carried out for each buyer. This transaction register includes an encrypted version of the embedded fingerprints.
- Tracing authority: in case of illegal re-distribution, it participates in the tracing protocol that is used to identify the illegal re-distributor(s).

4.2 Security model

The security assumptions of the proposed system are the following:

- The merchant does not need to be trusted either for distribution or to associate a pseudonym with the identity of a buyer. The protocols for distribution and for traitor tracing described below are proven to work even if the merchant is not trusted.
- Buyers are not trusted and protocols are provided to guarantee that 1) they are transferring authenticated fragments of the content and 2) their anonymity can be revoked in case they re-distribute the content illegally.
- The transaction monitor (or any other single party) will not have access to the cleartext of the fingerprints. This prevents that any single party can frame an innocent buyer.

- As privacy is concerned, the transaction monitor is not trusted and it should only have access to pseudonyms, but not to the buyers' real identities.
- The transaction monitor is trusted as the symmetric keys used for encrypting the fragments are concerned. This means that 1) the transaction monitor stores the key provided by each parent buyer and 2) this key can be retrieved only once from its database (in principle by the child buyer). After this retrieval, the transaction monitor blocks the register and eventually removes it (details are provided in the description of Protocol 2 below).
- The transaction monitor returns the true pseudonym corresponding to an illegal re-distributor in the traitor tracing protocol. However, this trust can be replaced by a collection of signatures provided by the proxies.
- The tracing authority is part of the legal system and shall be trusted. It is not expected that the authority participates in any coalition to frame an innocent buyer or break someone's privacy.
- The communication between the merchant and the seed buyers, and between peer buyers within the P2P distribution system, must be anonymous using an onion routing-like approach. The fragments of the content are encrypted using symmetric cryptography.
- Proxies are not trusted and the fragments sent through them shall be encrypted in such a way that only the sender and the recipient have access to their cleartext. Malicious proxies may also try to cheat by reporting false fingerprint segments (or not reporting them at all) to the transaction monitor.
- The fingerprints must be constructed with a long enough number of segments to guarantee that recombination will produce different fingerprints for different buyers due to numerical explosion.
- The hashing functions used in the system are secure and cannot be inverted.
- Public-key cryptography is restricted to the encryption of short binary strings, such as fingerprint segments or hashes. The different parties (merchant, transaction monitor, proxies and buyers) have a pair of public and private keys to be used in different steps of the protocols.
- A single malicious party shall not be able to construct the fingerprinted copy corresponding to any buyer to frame an honest user of the system. In a similar way, a single malicious party shall not be able to link the identity of some buyer to a particular content unless that user is involved in an illegal re-distribution of the content.

Different attacks that may be mounted against the system proposed in this work, regarding both security and privacy, are described below. The following assumptions are made:

- The watermarking method used for embedding and detecting the fingerprint is transparent, robust and secure enough for a fingerprinting application. There are hundreds of watermarking algorithms available to be used with the proposed fingerprinting system and watermarking properties are out of the scope of this work. For example, a robust video watermarking scheme is presented in [27] and a robust audio watermarking scheme is described in [14].

- Collusion attacks are the main topic in most of the research related to fingerprinting. Collusion occurs when several buyers decide to recombine their fingerprinted copies of a given content trying to obtain a new copy in which neither of their fingerprints is detectable. The system suggested in this work inherits the collusion resistance of the method described in [21] and, hence, no further analysis on collusion attacks is required.

Thus, the main attacks that may be performed on the proposed system are related to either the P2P distribution protocol, the traitor-tracing protocol and the P2P network itself. These attacks may be aimed to break either the security or the privacy properties of the system. The attacks to the cryptographic protocols require that one or more of the involved parties are malicious or that a malicious party tries to mimic the behavior of an honest party in order to gain sensitive information that may be used afterwards. As far as security is concerned, there are two main items to be protected:

- **Buyer frameproofness** is related to the possibility that an innocent buyer is accused of illegal re-distribution of the purchased content.
- **Copyright protection** would be broken if any party obtains a copy of the content whose fingerprint is not included in the fingerprints' database of the transaction monitor (and thus can be re-distributed illegally) or the association of that particular fingerprint with the illegal re-distributor cannot be completed.

On the other hand, privacy would be broken if someone could associate a real identity with the purchase of some specific content.

Attack models to frame honest buyers require that another party is able to obtain either the fingerprint of a buyer, or the fingerprinted copy of the content, in such a way that it can be further re-distributed and, finally, the honest buyer is accused of illegal re-distribution. The following types of attacks would be aimed to frame an innocent buyer:

- Random guess: the merchant has access to all fragments of the seed buyers and he may try to recombine the fragments to produce a new copy of the content. If this copy is re-distributed, there is a possibility that it leads to some innocent buyer that could be framed.
- Buyer authentication attacks: an attacker may impersonate a buyer in the system and try to obtain a fingerprinted copy of the content that would be linked to the impersonated buyer.
- Proxy authentication attacks: an attacker may impersonate a proxy in the system and try to obtain fragments of a content from different buyers in order to build a fake "colluded" copy of the content. This fake "collusion" could produce evidence against one ore more honest buyers that served fragments to the fake proxy.
- Man-in-the-middle attacks: an attacker may try to intercept the traffic between a buyer and one or more of her proxies and keep a copy of all the fragments of the content.

- Database authentication attacks: an attacker may try to obtain the fingerprint of a buyer that is stored in the transaction monitor's database.
- Protocol attacks: one or more of the participants in the protocols (proxies, merchant, buyers, transaction monitor or tracing authority) may be malicious and try to obtain the fingerprint or the fingerprinted content linked to a particular honest buyer.

The security of the system against these attacks is discussed in Section 7.

4.3 The P2P distribution protocol

To bootstrap the system, a few seeds of the fingerprinted content must be produced. The proposed approach is for the merchant to produce a small number M of instances of the content with different pseudo-random binary fingerprints, using some scheme satisfying the conditions described above. These M seeds could be the first buyers of the content who will be the ones contacted by second-generation buyers to obtain further copies of the content. Either the merchant or some trusted authority will keep the association of the first M fingerprints with the identities (or maybe some pseudonym) of the first M buyers. After the system is bootstrapped in this way, all future transactions occur without any further execution of the embedding scheme. Furthermore, all fingerprints from buyer $M+1$ to the final one are completely anonymous (accessible only if the buyer provides her copy of the content for fingerprint extraction) and do not relate to the buyers' identities. Note that this way of achieving anonymous fingerprinting is much simpler than the anonymous fingerprinting proposals in the literature [25, 3, 1, 8], predicated on some sort of complex cryptographic protocol for *every* transaction. Only the transaction monitor keeps a record of the engaged transactions in case they need to be used in future correlation tests.

We can summarize the P2P distribution protocol as follows.

Protocol 1

1. *For $i := 1$ to M, the merchant generates the i-th seed copy with a random fingerprint embedded in it (the fingerprints of the M copies should have low pairwise correlations).*
2. *For $i := 1$ to M, the merchant forwards the i-th seed copy to the i-th seed buyer. If the seed buyers are genuine rather than dummy buyers, this step can be anonymized as explained below.*
3. *For $i := M+1$ to N, the i-th buyer obtains her copy of the content by composing fragments obtained from a set S_i of parent nodes such that $S_i \subseteq \{B_1, \cdots, B_{i-1}\}$ and $|S_i| > 1$, where $|\cdot|$ is the cardinality operator and B_j refers to the j-th buyer. This transaction is performed via a proxy (or a set of proxies) and with an anonymous protocol (see below). The proxy registers each transaction at the transaction monitor.*

Since the same parent may be chosen by different proxies for different fragments, a transaction record for the same parent, child and content may already exist. In that case, no new record would be created. Note, however, that the transaction record will not be complete until all fragments have been obtained by the child buyer from all proxies, since the whole fingerprint hash (which is stored in the transaction record) will not be available until that moment. The transaction record is initially created with temporary information (content, parent, children and date/time) and, when all the fragments have been transferred for a buyer, the whole fingerprint and its hash are also stored in the transaction monitor.

The transaction record stored at the transaction monitor is formed by the following information:

P_i Username (pseudonym) of the buyer B_i.
$H(c)$ Perceptual content hash (used for indexing in the content database).
E_{h_i} Encrypted hash of the buyer's fingerprint.
E_{f_i} Encrypted buyer's fingerprint.
d Transaction date and time (for billing purposes).

Note that the transaction monitor does not store the true identities of the buyers, only pseudonyms. Only the merchant has access to the buyers' database, which relates a given pseudonym to real identity data.

The fingerprint and its hash are not stored as cleartext in the transaction monitor, the buyer's fingerprint is encrypted under the public keys of the parents and the transaction monitor and the hash of the buyer's fingerprint only with the public key of the transaction monitor. Note that having access to the fingerprints' hashes does not allow the transaction monitor to reconstruct any buyer's fingerprint, since a hash function is not invertible, thereby preserving buyer frameproofness.

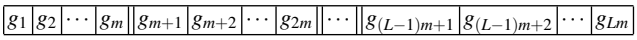


Fig. 3 Fingerprint's segments (g_j) and sets of m contiguous segments

The storage of an encrypted version of the buyers fingerprints, E_{f_i}, is computed as follows:

- Each fragment of the content shall be transmitted with a fingerprint's segment g_j embedded into it and together with an encrypted version of the segment $E^c_{g_j} \doteq E(g_j, K_c)$, where K_c is the public key of the transaction monitor. A signature can also be included, as detailed in Section 7.
- Each proxy selects a set of m contiguous fragments of the content and facilitates the anonymous communication between parents and child for the transmission of those fragments. These m contiguous fragments of the content carry m contiguous segments of the fingerprint embedded into them. The construction of the fingerprint with segments and sets of contiguous segments is shown in Figure 3.

- The same proxy is required for transferring the m contiguous fragments of the content. In this process, the proxy stores the corresponding encrypted segments $E^c_{g_j}$.
- The proxy concatenates the m contiguous encrypted segments, encrypts the concatenation using the public key of the tracing authority (K_a) and sends the result to the transaction monitor.
- Hence, the transaction monitor stores the following encrypted version of the fingerprint:

$$E_{f_i} = E\left(E^c_{g_1}|E^c_{g_2}|\ldots|E^c_{g_m}, K_a\right)|\ldots| \\ E\left(E^c_{g_{(L-1)m+1}}|E^c_{g_{(L-1)m+2}}|\ldots|E^c_{g_{Lm}}, K_a\right). \quad (1)$$

Note that 1) no proxy has access to the complete sequence of encrypted segments (since at least two proxies must be chosen by each buyer) and 2) the transaction monitor cannot decrypt E_{f_i} without the private key K^s_a of the authority. In this scheme, the (decrypted) fingerprint's hash h_i would be used in case of collusion of several buyers.

In order to protect the buyers' anonymity, the transfer between buyers must remain anonymous. Otherwise, some buyers (the parents of a child) may collude to generate a replica of the content of another buyer and redistribute it illegally. We refer to the use of symmetric cryptography to encrypt the content in such a way that intermediate routers do not have access to the cleartext of the content. The protocol for anonymous communication is based on using proxies between the parent and the child buyer. The content that is transferred over the proxy is encrypted, again, using symmetric cryptography, but the session key for encrypting the content is shared by parent and child using the transaction monitor as a temporary key database. For a set of fragments, the P2P software runs the following protocol:

Protocol 2

1. *The parent buyer chooses a symmetric (session) key k.*
2. *The parent chooses a pseudorandom binary sequence r to be used as a handle (primary database key) for k. The space for r should be large enough (e.g. 128 bits) to avoid collisions.*
3. *The parent buyer sends (r,k) to the transaction monitor, who stores it in a database.*
4. *The parent buyer sends r to the proxy and the proxy forwards r to the child buyer.*
5. *The child buyer sends the handle r to the transaction monitor, who replies with the symmetric key k.*
6. *The transaction monitor blocks the register (r,k) for a given period (timer). When the timer expires, the transaction monitor removes the register from the database.*
7. *The parent buyer sends the requested fragments, encrypted with k, to the proxy.*
8. *The proxy forwards all fragments to the child buyer, who can decrypt them using k.*

In this way, only a collusion formed by all the proxies of a child buyer (and possibly the transaction monitor) can replicate the entire fingerprint of the child. If every buyer chooses enough proxies for each content, such a collusion is so unlikely that it can be neglected. Regarding the choice of proxies, possibly the simplest and "most distributed" solution would be that all P2P clients (buyers) can be chosen as proxies by the P2P distribution software. Note that proxies do not have to be buyers of the same content and, thus, this would not break the privacy of buyers. In case that malicious proxies are considered, additional security measures shall be introduced, but this issue is left for future work.

Regarding the payment of content, our protocol does not explicitly consider payment by the buyers to the merchant. Our main focus is on fingerprinted multicast rather than on content sale. In any case, since the transactions are stored in the transaction monitor, a periodic invoice can be issued by the transaction monitor to the merchant such that the merchant can charge the buyers' accounts with the corresponding amounts. Note that such invoice does not need to specify particular contents, since only the total amounts of the downloaded contents of each buyer will be required. This preserves the buyers' privacy with respect to the merchant. It is even possible to establish some prepayment protocol between buyer, transaction monitor and merchant so that the buyer account is charged after each content transfer without disclosing specific contents to the merchant. Another alternative is to protect the access to the P2P platform by means of some subscription account. In any case, payments do not need to be distributed; they can be centralized and simple protocols can be used for them without disclosing which specific contents are being transferred to buyers.

An analysis of the security of this protocol is provided in Section 7.

5 Tracing illegal redistributors

We now show that the proposed fingerprinting method allows identification of illegal redistributors of fingerprinted contents.

5.1 Basic tracing protocol

Assuming that the embedding scheme is secure and robust enough so that malicious users cannot easily erase their fingerprints without making the content unusable (this is the standard marking assumption [2]), the following method can be used by a tracing authority to identify the source of an illegally redistributed copy. Notice that the tracing authority is an entity that is independent from both the merchant and the transaction monitor. The basic traitor tracing protocol (when no collusion occurs) begins with the extraction of the fingerprint of the illegally re-distributed copy by the tracing authority. Then, the authority uses the public key of the transaction monitor

and its own public key to produce the encrypted fingerprint which can be efficiently searched in the database of the transaction monitor. Once the pseudonym of the illegal re-distributor is available, it can be associated to a real identity.

Protocol 3

1. *The fingerprint f of the illegally redistributed content is extracted by the tracing authority using the extraction method and the extraction key (provided by the merchant).*
2. *The fingerprint's segments g_j are encrypted using the public key of the transaction monitor: $E^c_{g_j} = E(g_j, K_c)$.*
3. *The encrypted segments are grouped in sets of m consecutive elements which are encrypted using the public key of the authority, thereby obtaining E_f as per (1).*
4. *E_f is (efficiently) searched in the database of the transaction monitor in order to recover the pseudonym of the illegal re-distributor.*
5. *The merchant checks his database of clients and retrieves the identity of the traitor corresponding to the pseudonym obtained in the previous step.*

In the last step of this protocol, to prevent a malicious merchant from cheating and returning the identity of an innocent buyer, the identity must come together with a signed document that certifies the association between the pseudonym and the buyer's identity. All buyers will have to sign such a document (using their private keys) upon registration in the system. These signatures will be verifiable using the buyers' public keys. In this way, 1) there is no need for the merchant to be trusted and 2) non-repudiation from illegal re-distributors with respect to their pseudonym is obtained.

Note that all fingerprints are kept secret except the one that is being traced (f). The advantages of this traitor tracing system are obvious: the cleartext of the fingerprints of honest buyers is never required, and the traitor tracing protocol is based on a a standard database search (not involving innocent buyers).

A security analysis of this protocol is also provided in Section 7.

6 Efficiency aspects

In this section, the *efficiency* of the proposed fingerprinting system is discussed taking into account different points of view. Although simulations could be provided comparing the proposed system with other works of the literature, it must be taken into account that most of the anonymous fingerprinting protocols proposed so far are centralized and, thus, rely on unicast distribution. Simulated experiments to compare unicast and P2P distribution protocols, both as the CPU and communications costs are taken into account, would produce extremely different results, as discussed below. For this reason, a reasoned discussion of this topic (not illustrated with simulations) is provided in this section.

From the merchant's viewpoint, the efficiency of a multimedia content distribution system often refers to scalability. There are two main reasons why the system

proposed in this work is scalable. The first one is the use of a P2P distribution system, which reduces the distribution cost significantly on the merchant side: the merchant bootstraps the system by feeding only a few seed buyers, whereas the rest of transactions occur between peer buyers. Among the systems mentioned in the introduction, only [11, 21] propose P2P-based distribution systems. In fact, our system [21, 22] is exactly as efficient as this proposal (which can be found in [19]) from the merchant's point of view. On the other hand, the system [11] requires a multi-party secure protocol between each seed buyer and the merchant to bootstrap the system. Multi-party secure protocols are difficult to implement and require large computational and communication costs.

All the other anonymous fingerprinting systems reported in the introduction [26, 3, 23, 18, 1, 28, 16, 4, 17] are centralized. This means that: 1) a unicast distribution is required for each buyer; and 2) an anonymous protocol for embedding the fingerprint and transmitting the fingerprinted file is run between the merchant, each buyer and (possibly) other parties. Obviously, these solutions are much less scalable for the merchant, and involve much higher computational and communicational costs.

Hence, it can be concluded that our systems [21, 19] are the most efficient solutions for a merchant.

As the buyers are concerned, the centralized approaches also involve running expensive protocols, either because of the use of homomorphic encryption of the contents (which require homomorphic decryption in the buyer's side), zero-knowledge proofs, bit commitments or other similar techniques. Only [4] takes the computational costs for the buyers into account and proposes the use of powerful servers for the most expensive operations of the protocols. None of [21, 19] involve expensive decryption of the content (since symmetric encryption is used), zero-knowledge proofs or bit commitments. Although the buyers are required to participate in anonymous communication protocols using symmetric encryption and proxies, these protocols are not particularly demanding in communication or computation terms. For a buyer, the cost associated with obtaining a fingerprinted copy of the content will not be higher than that associated to a centralized approach. However, in P2P distribution systems, the buyers themselves become providers of the contents for other buyers and, thus, their bandwidth will be used in further transactions. Hence, P2P-based systems entail an upload cost for each buyer as long as she keeps the content in her multimedia database.

As far as the proposal of [11] is concerned, it is required that each buyer (except the leaves of the distribution tree) participate in an anonymous fingerprinting protocol with at least another buyer. In such a protocol, a new fingerprint must be embedded for each new buyer, which increases the communication and computational costs for the buyers who become sources of the content. Note that neither [19] nor [21] require that the buyers embed new fingerprints, since the new fingerprints are created through recombination of the segments of existing ones.

In principle, buyers would prefer a centralized distribution system in what regards bandwidth, since once the file has been obtained, no further communications are required. Nevertheless, the fact that the distribution is much "cheaper" from the merchant's point of view implies that the infrastructure (servers) required by the

merchant is much simpler. The reduced costs as compared with centralized distribution and fingerprinting will certainly have some effect in the price of the contents. If buyers enjoy cheaper prices, this would compensate them for the use of their bandwidth required by the P2P distribution protocol.

Another disadvantage of [19], as far as efficiency is concerned, is the need of an anonymous communication protocol, analogous to Chaum's mix networks [5], between the peer buyers. This protocol increases the number of messages in the network as a consequence of the use of proxies. In any case, this is a general disadvantage of any anonymous communication system, since a direct connection between sender and recipient must be avoided.

In short, our systems [21, 19] can be considered efficient from both the merchant's and the buyers' points of view, but the buyers are requested to become providers of the content for other buyers (which is a standard drawback of any P2P distribution system).

7 Security analysis

In this section, we first specify the security assumptions of our scheme [19]. We then analyze more complex collusion attacks.

As detailed in Section 4.2, attacks to the system may be classified as authentication/impersonation attacks, man-in-the-middle attacks and protocol attacks. Authentication/impersonation attacks should be overcome by using existing secure authentication protocols and are out of the scope of this work. As man-in-the-middle attacks are concerned, there is no possibility of intercepting and decrypting the messages between a buyer and a proxy, since communications with the transaction monitor and the child buyer should also be attacked in order to obtain the session key used for encrypting the content. If the communication between the child buyer and the transaction monitor (Step 5 of Protocol 2) are strongly authenticated (e.g. using a Public Key Infrastructure), the possibility of a successful man-in-the-middle attack can be neglected.

7.1 Formal analysis of the proposed protocols

First of all, the security and privacy properties of Protocols 2 and 3 is analyzed by means of two theorems (and their corresponding proofs).

Theorem 6.1. *In Protocol 2, a malicious proxy trying to decrypt the fragments of the content would be detected.*

Proof. If a malicious proxy tries to obtain the session key k by sending r to the transaction monitor there are two possibilities:

1. If the child buyer has already retrieved k from the database by sending the handle r to the transaction monitor, the register containing k would be either blocked or removed. Note that the transaction monitor is assumed to be honest for the management of the symmetric keys (see Section 7).
2. If the child buyer has not retrieved k from the transaction monitor, the proxy will obtain it, but the child buyer will find the corresponding register either blocked or removed. Then, the malicious behavior of the proxy can be reported to the authorities and the transaction monitor and the child buyer have enough information (such as pseudonyms and IP addresses) to identify the misbehaving proxy. Again, the assumption of honest behavior for the management of symmetric keys (Section 7) applies.

Hence, a malicious proxy trying to obtain k from r would be detected, since the register would be blocked either to the proxy or to the child buyer, raising an investigation. This completes the proof.

Theorem 6.2. *By applying Protocol 3, an illegal re-distributor can be traced efficiently using a standard database search in the transaction monitor and it is not required to decrypt any of the fingerprints recorded by the transaction monitor. The output of the tracing protocol is the identity of at least one illegal re-distributor.*

Proof. If no collusion occurs, the fingerprint f would be first extracted by the tracing authority, which is trusted (Section 7). Then the tracing authority would compute $E^c_{g_j} = E(g_j, K_c)$ for each segment (using the public key of the transaction monitor), and finally obtain E_f after grouping the segments in sets of m consecutive elements and encrypting these groups with its public key K_a. After that, the transaction monitor, which is also trusted for transaction database search (Section 7), would output the pseudonym of the illegal re-distributor. The pseudonym can be linked to the real identity by the merchant, who provides also a signed document that associates the real identity and the pseudonym. This completes the proof.

In case of collusion of several buyers, the extracted fingerprint would not be a valid codeword of the anti-collusion code used in the scheme. Then, the system described in 2 would be used: the encrypted hash $E_{h_f} = E(h_f, K_c)$ would be searched instead of the encrypted fingerprint, where h_f denotes the hash obtained applying the hash function to the traced fingerprint f. Thus, Protocol 3 would be used with the hash of the fingerprint instead of the fingerprint itself. As described in Section 3.1, with a large enough hash space, hash collisions would be almost negligible and a traitor would still be identified in the vast majority of the cases.

The requirement that the transaction monitor is trusted and returns the pseudonym of the buyer associated with the traced fingerprint (and not a different pseudonym) can be relaxed if a signature of the encrypted sets of segments of the fingerprint is provided by the proxies. These signatures can be verified using the public keys of the proxies. In that case, both the signatures and the pseudonyms of the proxies shall also be included in the registers of the transaction database to facilitate the verification of these signatures when required.

7.2 Collusion attacks on the protocols

Fingerprinting schemes must provide some degree of collusion resistance in order to be able to trace forged copies created by advanced attackers. In this section, we discuss possible collusion attacks on the proposed protocols. Hence, our scheme can be made as resistant against collusion as any of the existing anti-collusion techniques of the literature.

Under the usual marking assumption, error correcting codes are a typical solution to detect collusions [9]. Other approaches are based on more recent techniques, such as Tardos codes [29] or even newer codes based on them, like [24]. In the latter case, the marking assumption can be relaxed to a δ-marking assumption [24].

7.2.1 Buyer frameproofness

Buyer frameproofness relates to the case of a malicious merchant trying to frame an honest buyer by accusing her of being the source of an illegally redistributed content. In this scheme, the merchant is not able to produce any buyer's fingerprint by random guess due to the numerical explosion of the fingerprint space, even with a reduced number of seed buyers. On the other hand, the transaction monitor has access only to the hashes of the fingerprints (not the fingerprints themselves without the private key of the authority). Since the hash function is not invertible, it is not possible for the monitor (even in coalition with the merchant) to reconstruct any buyer's fingerprint. Possible collusions to disclose the specific fingerprint of an innocent buyer are the following:

1. The tracing authority and the transaction monitor.
2. All the proxies (for a transfer) and the transaction monitor.
3. All the proxies (for a transfer) and the merchant.

In the first case, the authority and the transaction monitor may use their private keys to obtain the cleartext of all the fingerprints. However, this possibility can be neglected since at least the authority must be trusted (as described in Section 4.2). In the second case, all the segments of the fingerprint could be decrypted using the private key of the transaction monitor, since the malicious proxies would not encrypt them with the public key of the authority. Also, the transaction monitor could collude with the proxies and use the session keys k to decrypt the fragments. Both possibilities would involve at least three malicious parties: all the proxies (two at least per each purchase) and the transaction monitor. In the third case, even if the transaction monitor does not provide her private key, a brute force attack segment by segment would be possible to reconstruct a buyer's fingerprint, because the number of different segments is small for each fragment (equal to M). Again, at least three malicious parties would be required: two (or more) proxies plus the merchant.

Hence, the minimum coalition required to frame an innocent buyer is formed by three malicious parties (or two if one of them is the authority). Note that a coalition of the transaction monitor and the merchant is not enough to obtain the cleartext of

any fingerprint. As the proxies encrypt a set of m consecutive segments, and there are M possible values for each segment, the total number of combinations per set of consecutive segments is M^m. This avoids a brute force attack if m is reasonably large. For example, if $M = 10$ and $m = 32$, there would be 10^{32} possible combinations for each set of consecutive segments, what would be enough for security against a brute force attack. If the segments were encrypted one by one (or grouped with a small value of m), the system would be vulnerable against a brute force attack for a collusion of the merchant and the transaction monitor.

7.2.2 Copyright protection

In order to ensure copyright protection, it is essential that the fingerprint embedded in each buyers' copy of the content and its encrypted version recorded by the transaction monitor are identical. If there is a way to cheat in the recorded fingerprint, the corresponding buyer would be able to re-distribute her copy illegally without any chance of being detected. As already remarked in Section 4, the content fragments are signed by the merchant from origin. The same approach can be used here for each encrypted segment of the fingerprint, making it impossible for a proxy to cheat about the fingerprint. The authority and the merchant could verify randomly, with some probability, the signatures of the set of contiguous segments reported by a proxy. If the signature was not verified, the proxy would be accused of forgery. Note that the fingerprints would still be protected since 1) only some sets of contiguous segments would be verified (not the whole fingerprint) and 2) those segments would still be encrypted with the transaction monitor's public key.

However, a proxy may still try to get alternative fragments for the same position of the content by requesting them from different parents. That possibility would allow the proxy to cheat about the true fingerprint of the child buyer, since several correctly signed fragments would be available for him for the same content. This behavior can be avoided in several ways. For example, temporary records can be created in the transaction monitor by the parents to detect if a proxy tries to obtain two alternative fragments for the same content.

7.2.3 Buyers' privacy

The identity of a buyer who has purchased a specific content could be revealed by a coalition of two parties: one of the proxies chosen by the buyer and the merchant (who can link her pseudonym to a real identity) or, similarly, the transaction monitor and the merchant. Better privacy could be achieved if, for example, the pseudonyms were encrypted by the proxies using the public key of the tracing authority. In that case, a coalition of the merchant and the transaction monitor would not be enough to break a buyer's privacy, but a coalition of a proxy and the merchant would still be enough. However, the merchant should not be interested, in principle, to break

her customers' privacy, since privacy would be one of the clear advantages of the proposed distribution system.

Another threat to privacy is the fact that all anonymous communications between each child and each parent occur through a unique proxy. This means that this proxy has access to different pseudonyms (the parents' and the child's). This can be easily circumvented if more proxies are used in Protocol 2 between child and parent. With two proxies, each of them would know only the pseudonym of one of the parties (although they could still collide). With three or more proxies, only two of them would have access to different pseudonyms (either the parents' or the child's). Of course, increasing the number of proxies in each transfer would affect the efficiency of the system, since more communication burden would be required.

8 Simulation results

This section presents a set of simulated experiments to illustrate the properties of the proposed system: buyer privacy, robustness against non-collaborative buyers and collusion resistance.

8.1 Buyer privacy

In all simulations presented in this section, the recombined fingerprints were 4096-bit sequences divided into 128 segments of 32 bits each. The first simulation to confirm the results presented in Section 7.2.3 consisted of producing different generations of buyers using an exponential growth approach and checking the average number of correlation tests required to identify the buyers. The number of seed buyers was taken to be $M = 10$ and each buyer could have between two and four parents which were chosen at random from all the previous generations (not only the immediately previous one). This means that the average number of parents per non-seed buyers was $n = 3$. The simulations shown in Table 1 were carried out, and a comparison of the average number of correlation tests with the expected fraction introduced in Section 7.2.3 is shown in Figure 4.

The results in Table 1 show a single simulation and the average 100 simulations with 100 different seeds in the pseudo-random number generator in order to reduce the bias of the results. It can be seen that no significant differences appeared between 1 and 100 simulations. The last column represents the average percentage of buyers requiring backtracking in the 100 simulations. Not surprisingly, as the network (graph) became larger, more buyers required backtracking, but the percentage was always small.

Figure 4 shows intervals for the average fraction of non-seed buyers affected by correlation tests as the number of generations grew. For each number of generations, the corresponding vertical solid line represents an interval with the up triangle show-

Table 1 Average number and percentage of correlation tests on non-seed buyers in an exponentially growing population

Generation	Population	Average correlation tests		Backtracking
		1 simulation	100 simulations	(100 sim.)
$k = 2$	$N = 20$	3.40 (34.0%)	3.71 (37.1%)	0%
$k = 3$	$N = 40$	6.93 (23.1%)	7.29 (24.3%)	0%
$k = 4$	$N = 80$	12.26 (17.5%)	11.69 (16.7%)	0.6%
$k = 5$	$N = 160$	18.99 (12.7%)	17.05 (11.4%)	1.2%
$k = 6$	$N = 320$	24.31 (7.8%)	23.76 (7.7%)	2.7%

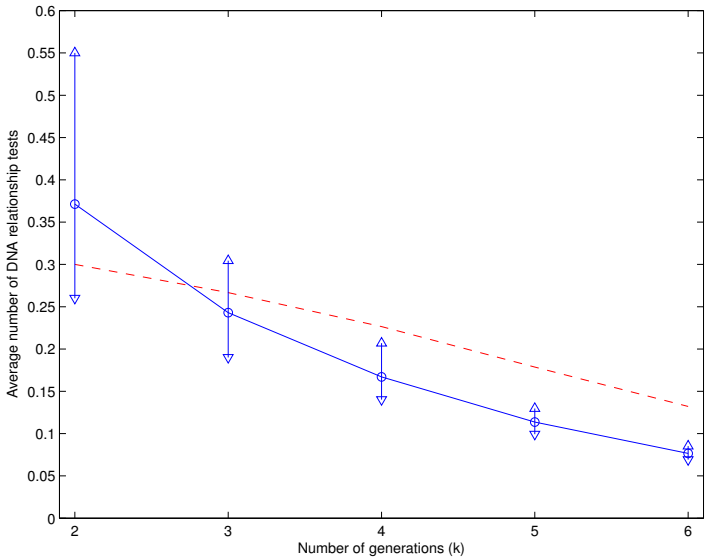


Fig. 4 Average fraction of non-seed buyers affected by correlation tests in an exponentially growing population: simulation results (solid) and theoretical expected maximum (dashed). Vertical solid lines are max-min intervals.

ing the maximum fraction in 100 simulations, the down triangle showing the minimum fraction and the circle showing the average fraction; these average fractions correspond to the percentages given in Table 1 for 100 simulations. As discussed in Section 7.2.3, the theoretical expected maximum fraction of tested non-seed buyers (dashed line) can be exceeded if the number of generations is small, due to the effect of some parents having more than the average number of children. This situation is compensated as more generations are produced and the simulated fraction goes below the theoretical value already for $k > 3$, although the interval for $k = 3$ shows that, for that number of generations, some simulations still yielded fractions above the theoretical value. In any case, as predicted in Section 7.2.3, the fraction of non-seed buyers affected by *one* correlation test decreased to zero as the number of generations grew: the more buyers involved, the higher the probability that a buyer

did not need to surrender her privacy in one particular correlation test. However, as the population grows, the number of illegal redistributions may also increase and more correlation tests may be needed to investigate them; as the number of required correlation tests increases, the probability that a non-seed buyer is affected by them (and therefore loses her privacy) also increases.

One should notice that correlations would be quite small if the number of parents providing a significant number of fragments to each child is too large. In particular, it is not advisable that the number of parents be close to the number M of seed buyers. Intuitively, if the number of parents is close to M, all correlations will be very small and similar, hence leading to more wrong choices in the tracing algorithm and, thus, more backtracking and more tests. We have confirmed this issue in additional simulation results that are omitted here for brevity. As a rule of thumb, the distribution software should limit the number of parents (through the number of proxies) such that it is not too close to M. Since specific software is to be used for the P2P distribution, constraints can be enforced on the number of allowed proxies/parents for each content transfer. In fact, if many parents are allowed for a buyer, the correlations between parents and children will be relatively low, irrespective of the number M of seed buyers, which would lead to more backtracking. It is thus advisable to keep the number of allowed parents limited even if the number of seed buyers is relatively large.

It may appear that the percentage of buyers involved in correlation tests in the course of an investigation decreases to zero because of the exponential increase in population occurring at each generation. However, this is not the case. The decrease of this ratio of tested buyers depends on the population and not on the particular way it grows. To illustrate this process, the following simulations were performed with a population growing linearly at each generation:

1. The first generation was, again, formed by the $M = 10$ seed buyers who obtain their fingerprinted contents from the merchant.
2. At each new generation, $M = 10$ new buyers obtained their contents from a variable number of parents between two and four (and thus, the average number of parents was, again, $n = 3$).
3. With this scenario, the population N increased linearly with the number of generations: there were $N = kM$ buyers after the k-th generation.

Table 2 illustrates this issue. It can be seen that the fraction of tested buyers decreased with the number of generations. In this case, the decrease was linear and not exponential, since the population increased linearly with k. This is also illustrated in Figure 5 by means of interval plots. The seeds of the pseudo-random number generator were adjusted such that the results for two generations ($N = 20$) were the same as those presented in Table 1 for the exponential growth.

We present also simulation results comparing the linear and exponential growths scenarios *for the same population*. The results are shown in Table 3 and Figure 6. It can be seen that, when populations are of the same size, the results are almost identical irrespective of the number of generations and the growth model (exponential or

Table 2 Average number and percentage of correlation tests on non-seed buyers in a linearly growing population

Generations	Population	Average correlation tests		Backtracking
		1 simulation	100 simulations	(100 sim.)
$k = 2$	$N = 20$	3.40 (34.0%)	3.71 (37.1%)	0%
$k = 3$	$N = 30$	5.40 (27.0%)	5.54 (27.7%)	0%
$k = 4$	$N = 40$	6.20 (20.7%)	6.76 (22.5%)	0.17%
$k = 5$	$N = 50$	7.45 (18.6%)	8.15 (20.4%)	0.45%
$k = 6$	$N = 60$	8.20 (16.4%)	8.95 (17.9%)	0.42%

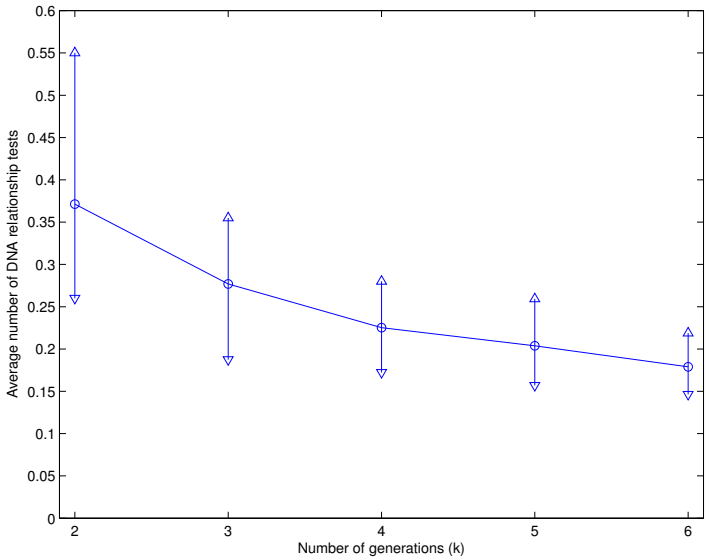


Fig. 5 Average fraction of non-seed buyers affected by correlation tests in a linearly growing population (simulation results). Vertical solid lines are max-min intervals.

linear). Again, the seeds of the pseudo-random number generator were adjusted so that the results for two generations ($N = 20$) were the same for both growth models.

8.2 Collusion resistance

In this section, experiments conducted with the anti-collusion version of the scheme suggested in Section 7.2 are presented. This simulation is a proof of concept. In a practical implementation, other codes and parameters should possibly be used. However, this implementation shows that the method suggested to fight collusion is more than a theoretical possibility.

Table 3 Average number and percentage of correlation tests on non-seed buyers: comparison between exponential and linear growth for the same population

Population	Exponential growth		Linear growth	
	Generations	Average tests (100 simul.)	Generations	Average tests (100 simul.)
$N = 20$	$k = 2$	3.71 (37.1%)	$k = 2$	3.71 (37.1%)
$N = 40$	$k = 3$	7.29 (24.3%)	$k = 4$	6.90 (23.0%)
$N = 80$	$k = 4$	11.69 (16.7%)	$k = 8$	10.62 (15.2%)
$N = 160$	$k = 5$	17.05 (11.4%)	$k = 16$	15.43 (10.3%)
$N = 320$	$k = 6$	23.76 (7.7%)	$k = 32$	22.23 (7.2%)

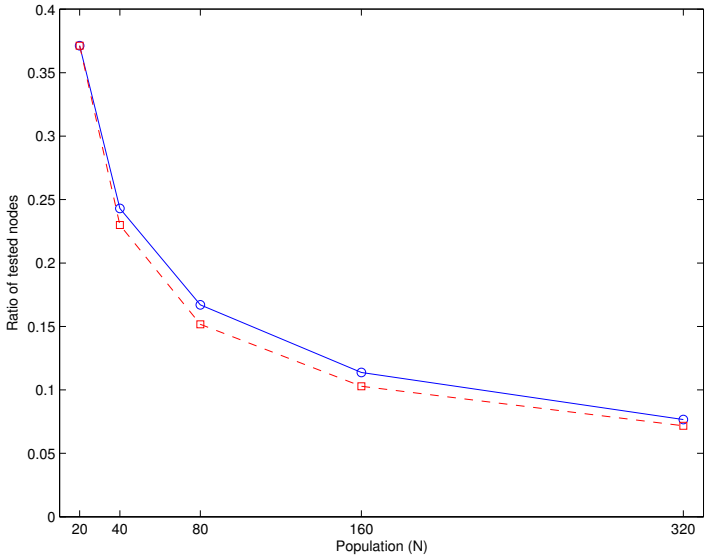


Fig. 6 Average fraction of non-seed buyers affected by correlation tests: comparison between exponential growth (circle, solid) vs linear growth (square, dashed) for the same population. Abscissa is population size.

The details of the implementation are as follows:

- A dual Hamming code $DH(31,5)$ was used to encode the segments. $2^5 = 32$ values were thus possible for each segment. Each segment had 5 bits of data and 26 redundancy bits. This code can be used to detect collusions of two buyers.
- A dual Hamming code $DH(1023,10)$, which also detects collusions of two buyers, was used to encode the hash of the fingerprint. Hence, $2^{10} = 1024$ different hashes existed, with 10 bits of data and 1013 bits of redundancy. This number of hashes would not be enough for a real implementation of the method, but it sufficed for this proof of concept.

- With these choices, the fingerprints were formed based on 1023 segments, each of which consisting of 31 bits. Hence, the fingerprints were $1023 \cdot 31 = 31{,}713$-bit long. The multimedia content had to be split into 1023 fragments, carrying each 31 embedded bits. Possibly, a better choice for a practical implementation with error-correcting codes would be Reed-Solomon (RS) codes instead of dual Hamming codes. In that case, the segments would represent symbols of the code (segment-level code) and the hash of the fingerprint could be an RS codeword (hash-level code). Note that high-capacity robust watermarking schemes exist for embedding that amount of information. For example, the method proposed in [13] allows embedding up to 11,000 bits in a second of audio.
- 10 seed buyers were generated ($M = 10$). The hash of each segment was computed by simply selecting the third data bit of each gene. This is not a sophisticated hash and is obviously quite insecure, but it sufficed for simulation purposes. More advanced hashing techniques would be required in practice.
- For each segment, exactly five seed buyers had a '0' hash and the other five had a '1' hash. As pointed out in Section 7.2, this maximized the chances that a proxy could find parents with segments having the hash bit values required to build any hash-level anti-collusion codeword.
- An exponential increase of the population was assumed: 6 generations were created, resulting in a total population of $10 \cdot 2^5 = 320$ buyers (including the seed buyers).
- When non-seed buyers downloaded the content, the first 10 segments were chosen randomly between two and four parents. This yielded the 10 data bits of the hash of the fingerprint. The remaining 1013 bits of the hash had to be such that a codeword of the $DH(1023, 10)$ code was obtained. This was achieved by requesting fragments with segments that carry the appropriate hash bit to the current set of parents. If no parent with the required bit was found, the proxy looked for a new parent with an appropriate segment. This new parent was included in the set of parents of that buyer and was considered as a potential parent for the remaining fragments (segments and hash bits).

After generating a random population with these settings, the actual number of parents per (non-seed) buyer ranged from 4 to 11, with an average of $n = 9.09$ parents per buyer. Hence the privacy results could not be directly compared with those of the previous sections (for which the average number of parents per buyer was around $n = 3$). Note that the number of parents needs to be increased in the collusion-resistant case compared to the basic case because the bits of the fingerprint's hash cannot be chosen in a completely free manner. A valid codeword must be constructed for the fingerprint's hash. Thus, extra parents are selected during the P2P download if none of the already chosen parents provides the required hash bit for a given fragment. For example, it is highly unlikely that the new fingerprint hash can be constructed completely with only two parents since the collusion-resistant code may require a specific hash bit ('0' or '1') for a particular position, and this hash bit may not be available in the corresponding fragment of these two parents. In fact, the minimum number of parents obtained during this simulation to construct a valid codeword for the fingerprint's hash was four.

With these settings, 200 random bit collusions were generated. For each collusion, a new fingerprint was created by choosing randomly a new fingerprint's bit when the bits of the colluders differed. Hence, after the collusion, the obtained forged copy had a non-codeword embedded into it, both at the segment level and at the hash level. This is the standard marking assumption. For each forged copy, the advanced tracing system described in Section 7.2 was applied, by decoding the segments and the fingerprint's hash using the $DH(31,5)$ and $DH(1023,10)$ codes, respectively. Note that, with this approach, the colluders themselves did not need to participate in the search. When a buyer was selected as the most likely ancestor of the colluder, the hashes of her children were examined. If a match occurred for the hash of the fingerprint, the corresponding child buyer was the traitor. In case that hash collisions are allowed, some additional investigations would be required to guarantee that the selected buyer is a colluder, but this simplified scenario did not require further tests. After these 200 experiments, the average number of tests to find the colluder (with neither false positives nor false negatives) was 47.77 or 14.8% of non-seed buyers. This is much below the theoretical maximum expected value for $M = 10$, $n = 9.09$ and $k = 6$ as $R = 0.400$ or 40.0% of non-seed nodes. Thus, even in case of collusion, the number of non-seed nodes involved in correlation tests decreases to zero as the population grows.

9 Conclusions

We have presented a recombination fingerprinting P2P scheme. We show that, through co-utility, the proposed solution enforces automatic fingerprinting because merchants and buyers are selfishly interested in distributing or obtaining contents, respectively, from more than one buyer or merchant. The proposed scheme allows the merchant to trace unlawful redistributors of the P2P distributed content. The merchant knows at most the fingerprinted copies of the seed buyers, but not the fingerprinted copies of non-seed buyers (the vast majority). Hence, the merchant does not know the identities of non-seed buyers. Whenever an illegal redistribution needs to be traced, only a small fraction of honest users must surrender their privacy by providing their fingerprinted copies (quasi-privacy). Our scheme also offers collusion resistance against dishonest buyers trying to create a forged copy without any of their fingerprints. Finally, a malicious merchant is most likely to fail using the fingerprinted copies of seed buyers to try to frame an honest non-seed buyer (buyer frameproofness).

Acknowledgements Funding by the Templeton World Charity Foundation (grant TWCF0095/AB60 "CO-UTILITY") is gratefully acknowledged. Also, partial support to this work has been received from the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2014 SGR 537), the Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis", TIN2015-70054-REDC and TIN2016-80250-R "Sec-MCloud") and the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"). The authors are with the UNESCO

Chair in Data Privacy, but the views in this work are the authors' own and are not necessarily shared by UNESCO or any of the funding bodies.

References

1. Bo, Y., Piyuan, L., Wenzheng, Z. (2007) An efficient anonymous fingerprinting protocol. In: Computational Intelligence and Security, LNCS 4456, Springer, pp. 824–832.
2. Boneh, D., Shaw, J. (1995) Collusion-secure fingerprinting for digital data. In: Advances in Cryptology, LNCS 963, Springer, pp. 452–465.
3. Camenisch, J. (2000) Efficient anonymous fingerprinting with group signatures. In: Asiacrypt, LNCS 1976, Springer, pp. 415–428.
4. Chang, C.-C., Tsai, H.-C., Hsieh, Y.-P. (2010) An efficient and fair buyer-seller fingerprinting scheme for large scale networks. Computers & Security 29(2):269–277.
5. Chaum, D. L. (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24(2):84–90.
6. Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J., Kalker, T. (2008) Digital Watermarking and Steganography. Morgan Kaufmann, Burlington MA.
7. Domingo-Ferrer, J. (1998) Anonymous fingerprinting of electronic information with automatic identification of redistributors. Electronics Letters 34(13):1303–1304.
8. Domingo-Ferrer, J. (1999) Anonymous fingerprinting based on committed oblivious transfer. In: Public Key Cryptography-PKC 1999, LNCS 1560, Springer, pp. 43–52.
9. Domingo-Ferrer, J. and Herrera-Joancomartí, J. (2000) Short collusion-secure fingerprints based on dual binary Hamming codes. Electronics Letters 36(20):1697–1699.
10. Domingo-Ferrer, J., Martínez, S., Sánchez, D. and Soria-Comas, J. (2017) Co-utility: self-enforcing protocols for the mutual benefit of participants. Engineering Applications of Artificial Intelligence 59:148-158.
11. Domingo-Ferrer, J., Megías, D. (2013) Distributed multicast of fingerprinted content based on a rational peer-to-peer community. Computer Communications 36:542–550.
12. Domingo-Ferrer, J., Sánchez, D., Soria-Comas, J. (2016) Co-utility: self-enforcing collaborative protocols with mutual help. Progress in Artificial Intelligence 5(2):105–110.
13. Fallahpour, M., Megías, D. High capacity audio watermarking using the high frequency band of the wavelet domain. Multimedia Tools and Applications **52**(2):485–498 (2011)
14. Fallahpour, M., Megías, D. (2014) Secure logarithmic audio watermarking scheme based on the human auditory system. Multimedia Systems 20:155–164.
15. Goldberg, D. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston.
16. Katzenbeisser, S., Lemma, A., Celik, M., van der Veen, M., Maas, M. (2008) A buyer-seller watermarking protocol based on secure embedding. IEEE Transactions on Information Forensics and Security 3(4):783–786.
17. Kuribayashi, M. (2010) On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol. EURASIP Journal on Information Security, 2010: art 694797.
18. Lei, C.-L., Yu, P.-L., Tsai, P.-L., Chan, M.-H. (2004) An efficient and anonymous buyer-seller watermarking protocol. IEEE Transactions on Image Processing, 13(12):1618–1626.
19. Megías, D. (2015) Improved privacy-preserving P2P multimedia distribution based on recombined fingerprints. IEEE Trans. on Dependable and Secure Computing 12(2):179–189.
20. Megías, D., Serra-Ruiz, J., Fallahpour, M. (2010) Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification. Signal Processing, 90(12):3078–3092.
21. Megías, D., Domingo-Ferrer, J. (2013) DNA-Inspired Anonymous Fingerprinting for Efficient Peer-To-Peer Content Distribution. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2013), pp. 2376–2383.

22. Megías, D., Domingo-Ferrer, J. (2014) Privacy-aware peer-to-peer content distribution using automatically recombined fingerprints. Multimedia Systems, 20(2):105–125.
23. Memon, N., Wong, P.-W. (2001) A buyer-seller watermarking protocol. IEEE Transactions on Image Processing, 10(4):643–649.
24. 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 (AAECC'07). Springer, pp. 80–89.
25. Pfitzmann, B., Waidner, M. (1997) Anonymous fingerprinting. In: Advances in Cryptology-EUROCRYPT'96, LNCS 1233, Springer, pp. 88–102.
26. Pfitzmann, B. and Sadeghi, A.-R. (1999) Coin-based anonymous fingerprinting. In: Advances in Cryptology-EUROCRYPT'99, LNCS 1592, Springer, pp. 150–164.
27. Preda, R. O., Vizireanu, D. N. (2011) Robust wavelet-based video watermarking scheme for copyright protection using the human visual system. Journal of Electronic Imaging 20:13022–130030.
28. Prins, J. P., Erkin, Z., Lagendijk, R. L. (2007) Anonymous fingerprinting with robust QIM watermarking techniques. EURASIP Journal on Information Security 2007: art n. 20.
29. Tardos, G. (2003) Optimal probabilistic fingerprint codes. In: Proceedings of the 35th annual ACM symposium on Theory of Computing (STOC '03). ACM, pp. 116–125.