**Noname manuscript No.**
(will be inserted by the editor)

# Privacy-Aware Peer-to-Peer Content Distribution Using Automatically Recombined Fingerprints

**David Megías · Josep Domingo-Ferrer**

Received: date / Accepted: date

**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 paper, we propose a novel 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.

**Keywords** Peer-to-peer content distribution, anonymous fingerprinting, collusion-resistant fingerprinting, buyer frameproofness, recombination fingerprinting.

D. Megías
Universitat Oberta de Catalunya,
Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica, Multimèdia i Telecomunicació,
Rambla del Poblenou, 156,
08018 Barcelona, Catalonia, Spain,
E-mail: dmegias@uoc.edu

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

## 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 sender's point of view. However, the unicast approach has the strong point 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. BitTorrent [1], Kademlia [19] or eDonkey2000 [16] are widely known examples of P2P file sharing protocols. 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, *e.g.* [26]). 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 paper 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 [27,11]. In [12], an anonymous fingerprinting protocol completely specified from the computational point of view and based on committed oblivious transfers was described. In [15], 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 [21,29,22,24,28]. 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 [20]. In [4], 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 [2], 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 [20], 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 [5], 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 paper, 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.

### Contribution and plan of this paper

We propose a 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.

The rest of this paper is organized as follows. Section 2 gives an overview of the proposed scheme. Section 3 describes the basic principles used in the paper for peer-to-peer distribution of fingerprinted contents. Section 4 presents the P2P distribution protocol and how transfers between peer buyers are anonymized. Section 5 presents a protocol for tracing unlawful redistributors, together with some examples; a modification of the method is presented to make tracing resistant against buyer collusions. Section 6 discusses security assumptions, as well as the buyer privacy and frameproofness offered by our fingerprinting proposal. Section 7 contains simulation results. Finally, Section 8 summarizes conclusions and future research issues.

## 2 Overview of the proposed system

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 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 correlation test is run taking the fingerprint of the illegally redistributed content and the fingerprints of the $M$ seed buyers as inputs; among the seed buyers, the test attempts to determine the maximum-likelihood ancestor of the content.

The fingerprints of the selected ancestor's children are retrieved by the tracing authority (with the collaboration of the buyers) and the maximum-likelihood test is run again with these fingerprints and the traced fingerprint as inputs. When a match is found between both fingerprints (maximum correlation between fingerprints) the redistributor is identified. If a buyer refuses to take the correlation test, the hash of the fingerprint can be used as evidence against her. If the hash of a buyer's fingerprint exactly matches the hash of the redistributed content's fingerprint, then the buyer is charged with unlawful redistribution. Otherwise, if the hashes differ, the refusing buyer will be charged with contract breach and the test is performed using the hashes of the fingerprints as a replacement of the entire fingerprints. 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.

If the correlation test is carried out using a secure multiparty computation approach [8, 10], the exact fingerprint of honest buyers will not have to be revealed (although computing some correlation with it and obtaining the complete hash will be required), but their privacy (the fact that they have purchased the contents) will not be preserved versus the tracing authority. However, buyer privacy with respect to the authority will only be broken for those few users affected by correlation tests and their identity will be revealed only to the identification agent. On the other hand, the privacy of the majority of 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.

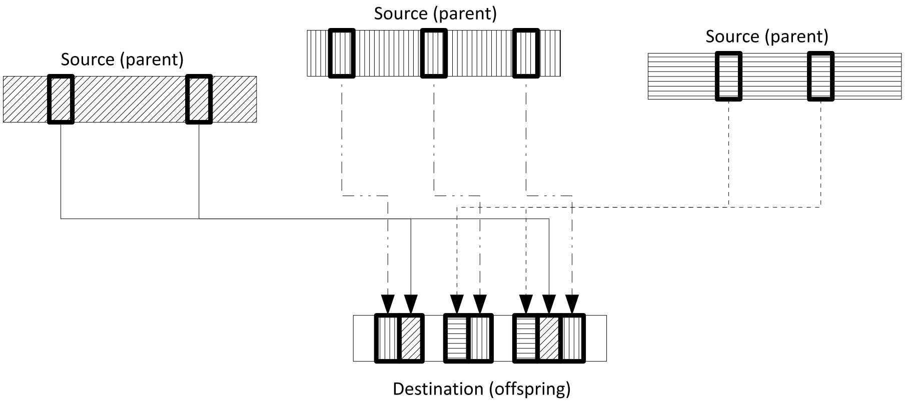


Fig. 1: Upload/download of the content (mating process)

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

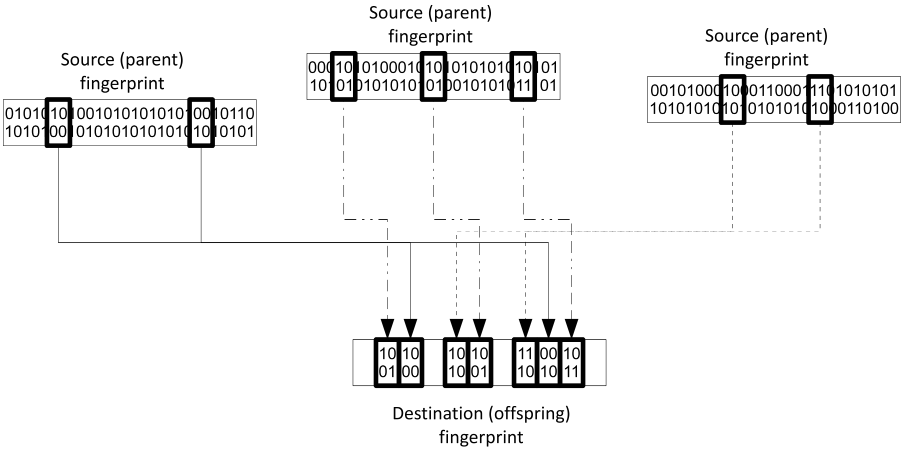


Fig. 2: Automatic recombined fingerprint construction

### 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 paper 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 paper, 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 [18] for solving optimization problems. Just like DNA sequences are formed by different genes which encode a give protein, the binary fingerprints used in this paper 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 search process must be carried out in the P2P distribution graph. This search is performed with the help of a correlation function which tries to minimize the number of explored nodes. The idea is to search for a given fingerprint from "ancestors" to "descendants" in the graph. A simple correlation function between two binary strings can be used to determine the likelihood that a given buyer is an "ancestor" of another one. A path from "ancestors" of a given buyer can thus be formed using this correlation function to identify the source of an illegally redistributed copy of the content.

### 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 [23].
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 [9] 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 paper. 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 Coprivacy 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.

This situation in which the best strategy to preserve one's own privacy is to act in such a way that someone else's privacy is protected is known as coprivacy [13, 14]. In game-theoretic terms, the vector of strategies (multiple children, multiple parents) is a Nash equilibrium between parent and child.

The coprivacy 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. 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.

### 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

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 [27,4,2,12], 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 (P2P distribution)

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's hash is also stored in the transaction monitor.*

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

- Username (pseudonym) of the parent (source) buyer.
- Username (pseudonym) of the child (destination) buyer.
- Content hash (used for indexing in the content database).
- Encrypted hash of the child buyer's fingerprint.
- 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 hash of the fingerprint is not stored as cleartext in the transaction monitor, but encrypted under the public keys of the parents and the transaction monitor; the proxy records one encrypted version under each parent's public key. In this way, in case of an investigation, the transaction monitor will need the cooperation of one parent to decrypt the hash. This provides additional anonymity and protection to 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. For a set of fragments, the P2P software runs the following protocol:

**Protocol 2 (Anonymous content transfer between buyers)**

1. *The child buyer's P2P client software contacts a proxy and requests a group of fragments.*
2. *The proxy selects a minimum of two buyers (parents) as the sources of the content fragments. This guarantees that each buyer will have at least two parents. The proxy uses an onion routing-like solution (based on Chaum's mix networks [6]) such that the fragments are transferred anonymously from parent to child. Note that the content does not need to be encrypted using public-key cryptography. A one-time symmetric session key can be chosen by the child buyer and be transmitted to the proxy. This session key is used to encrypt the actual fragments, so that the routers cannot see them in cleartext.*

3. *The proxy informs the transaction monitor about the transaction when fragments have been transferred from parent to child. A transaction record is then stored in the transaction monitor for this parent-child-content association. The proxy also informs the transaction monitor about the number of fragments transferred to the child.*
4. *For each fragment, the proxy also receives the hash of the corresponding segment (the part of the fingerprint embedded in the fragment). Note that a parent does not have any motivation to cheat about the hash bit, since: i) doing so would only favor an unknown child; ii) if she cheats, she may be discovered in future investigations and be accused of contract breach. Additional security using a ciphertext for the fingerprint's hash bits and a standard hash of the whole fragment can be easily introduced as detailed below (see Note 1).*
5. *When the child has received all the fragments of the complete content, the transaction monitor can contact all proxies involved in the transfer and construct the hash $h_f$ of the fingerprint by joining all the segment hashes together, following the steps detailed below. If a buyer choses $p$ proxies, then:*
   - *Each proxy obtains a fragment $h_{f_i}$ of the fingerprint's hash $h_f$ for $i = 1, \ldots, p$ (containing several bits corresponding to the hashes of various segments). For simplicity of notation and without loss of generality, it is assumed that the fragments of the fingerprint's hash are consecutive and ordered with respect to the index $i$: $h_f = h_{f_1}|h_{f_2}|\ldots|h_{f_p}$. Note that a simple permutation of the different hash fragments can be used to make the previous assumption hold.*
   - *All proxies exchange their fragments of the fingerprint's hash encrypted with the public key of the transaction monitor ($K_c$). Hence, all proxies have*

     $$E_h = E(h_{f_1}, K_c)|E(h_{f_2}, K_c)|\ldots|E(h_{f_p}, K_c).$$

     *This also means that no single proxy has access to the complete cleartext of the fingerprint's hash.*
   - *Let $P_{i,j}$ be the $j$-th parent chosen by the $i$-th proxy, and $K_{i,j}$ her corresponding public key. For every parent $j$ chosen by the $i$-th proxy, the proxy sends $E(E_h, K_{i,j})$ to the transaction monitor.*

*Note 1* Since all fragments $F_k$ are produced by the merchant, they can be sent together with a ciphertext $E(H(F_k)|h(g_k)|\mathit{nonce}, K_m^s)$ (which plays the role of a signature) for the fragment and the fingerprint segment's hash bit, where $H$ is a (public) standard hash function that produces a summary (hash) of the whole fragment, $h(g_k)$ is the hash bit of the fingerprint segment $g_k$ embedded in $F_k$, *nonce* is a random padding (required since $h(g_k)$ is a single bit) and $K_m^s$ is the encryption key of the merchant (that should be kept secret). When a proxy gets a fragment $F_k$ with its corresponding ciphertext (which includes the fingerprint's hash bit), she can first decrypt the ciphertext using a decryption key $K^m$ corresponding to $K_m^s$:

$$D(E(H(F_k)|h(g_k)|\mathit{nonce}, K_m^s), K_m) = H(F_k)|h(g_k)|\mathit{nonce}.$$

Then, she can apply the hash function $H$ to the fragment $F_k$ and check the hash value $H(F_k)$. If the hash check is successful, then the recovered segment hash bit $h(g_k)$ is authenticated. Note that no forgery is possible for the segment hash bit $h(g_k)$ without having access to the merchant's encryption key. In addition, if this scheme is used, it is not necessary to transmit the hash bit as cleartext (sending the ciphertext is enough).

An alternative solution (with the same effect) would be to transmit, for each fragment $F_k$, the cleartext $H(F_k)|h(g_k)|\mathit{nonce}$ together with a signature of this cleartext by the merchant. Then, the receiver (proxy) would just need to verify the signature to authenticate both the fragment and the fingerprint's hash bit.

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.

The hash is encrypted under the public key of each parent and registered once per parent. In this way, the cooperation of only one parent is enough to obtain the decrypted fingerprint's hash. The transaction monitor needs one of the parents to use her private key $K_{i,j}^s$ to obtain:

$$D(E(E_h, K_{i,j}), K_{i,j}^s) = E_h.$$

After that, the transaction monitor can use its own private key $K_c^s$ to decrypt the fingerprint's hash:

$$\begin{aligned}&D(E(h_{f_1}, K_c), K_c^s)|D(E(h_{f_2}, K_c), K_c^s)|\ldots|D(E(h_{f_p}, K_c), K_c^s)\\&= h_{f_1}|h_{f_2}|\ldots|h_{f_p} = h_f.\end{aligned}$$

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.

*Note 2 (On 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.

## 5 Tracing illegal redistributors

We now show that the proposed fingerprinting method allows identification of illegal redistributors of fingerprinted contents. Here, we distinguish between the basic protocol and the collusion-resistant version of the scheme.

### 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 [3]), 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.

**Protocol 3 (Tracing)**

1. *The fingerprint $f$ of the illegally redistributed content $X_f$ is extracted by the tracing authority using the extraction method and the secret extraction key $K^e$ provided by the merchant.*
2. *The initial test set $T_0$ is built with the $M$ buyers of the seed versions of the file.*
3. *Let $i := 0$.*
4. *The tracing authority contacts the buyers in the current set $T_i$. It also retrieves the hashes of the fingerprints of these buyers from the transaction monitor. This step requires the private key $K_j^s$ of one parent of these buyers (the merchant in case of $i = 0$ and the selected ancestor in the set $T_{i-1}$ otherwise) and the private key $K_c^s$ of the transaction monitor. The fingerprints of the buyers of $T_i$ are extracted using the extraction function and the secret extraction key $K^e$. The hash function $h$ is then applied for each segment to obtain the fingerprints' hashes for all tested buyers. If any of the buyers' fingerprints produces a hash which does not match the corresponding record in the transaction monitor, the associated buyer will be accused of forgery (contract breach).*
5. *In case that no forgery occurs, the correlation test is performed with the fingerprints of the buyers in the current set $T_i$. This step is carried out as a simple bitstream correlation. Given the fingerprint $f$ to be traced and the test fingerprint $f'$ extracted from the copy $X_{f'}$ held by a buyer in $T_i$, both fingerprints with length $L$, the correlation $C(f, f')$ between $f$ and $f'$ can be computed as:*

$$C(f, f') = \frac{1}{L} \sum_{j=1}^{L} (-1)^{f_j \oplus f'_j}, \tag{1}$$

   *where $f_j$ and $f'_j$ are, respectively, the $j$-th bits of $f$ and $f'$, and $\oplus$ refers to the exclusive-or operation. In case of forgery, this step can be computed with the hashes of the fingerprints instead of the fingerprints themselves. If the correlation of the hashes is equal to 1, the corresponding buyer is charged of unlawful distribution and the tracing protocol halts.*

6. *If no buyer has been accused of illegal redistribution so far, there may be three outcomes of the previous step:*
   (a) *One or more buyers in $T_i$ refuse to collaborate with the tracing authority in computing their correlations with $f$; in this case, depending on the correlation between the hash $h_f$ and the hash(es) of the refusing buyer(s) (recorded in the transaction monitor), the refusing buyer(s) is(are) accused either of redistribution (if hashes are identical) or contract breach (otherwise). If the correlation between hashes is lower than 1, this correlation can be used as a replacement of the correlation between the fingerprints.*
   (b) *One buyer in $T_i$ has $C(f, f') = 1$; in this case, this buyer is accused of the redistribution.*
   (c) *Otherwise, the buyer in $T_i$ who has the maximum correlation with $f$ is taken as the most likely ancestor of the buyer of the illegally redistributed copy; in this case, a new set $T_{i+1}$ of buyers is built with all the children of this ancestor buyer, excluding any children buyers who have been already analyzed (remember that a buyer can have several parents). These children can be obtained from the transaction monitor (transaction records). Once the new set $T_{i+1}$ is available, set $i := i + 1$ and go to Step 4.*

Once the correlation between the fingerprints has been computed, the tracing authority no longer needs the fingerprint details of the associated buyer and should destroy this information (unless the buyer is charged with illegal redistribution or contract breach).

Although the maximum correlation criterion will be right most of the time, it cannot be discarded that a higher correlation might accidentally be obtained for a non-ancestor of the buyer of the illegally redistributed copy. For example, a descendant $A$ of the illegal redistributor $B$ may have as another ancestor a node $C$ of the graph which is also ancestor of $B$. This would produce a high correlation with $A$ but the chain from $C$ to $A$ skips the illegal redistributor $B$. In this situation, *backtracking* is required in the tracing protocol described above. A complete subnetwork would be analyzed until all nodes of the subgraph having no children are considered. When a complete subnetwork is exhausted, the element of $T_i$ with the second maximum correlation would be chosen as the candidate ancestor of the illegal redistributor. When all elements of $T_i$ have been considered without success (*i.e.* without being able to accuse anyone), the procedure would backtrack to the set $T_{i-1}$. Backtracking has been needed in a very small number of the simulations presented in Section 7.

To compute the correlation with a buyer's fingerprint and the traced fingerprint, the analyzed buyer must provide her copy of the content. A buyer may argue that she has lost or accidentally deleted the content file to refuse taking part in the test. Such a possibility should be limited in the contract of buyers for using the P2P distribution platform; yet, the lack of buyer co-operation can be circumvented as follows. Even if the actual content is not available, the hash of the fingerprint is stored at the transaction monitor. As a side effect of the way fingerprints and hashes are created, the bit correlation between hashes is a good estimate of the bit correlation between fingerprints. When two segments of two fingerprints are identical, they contribute to the overall correlation with a positive value. In this case, the bit hashes are also identical,

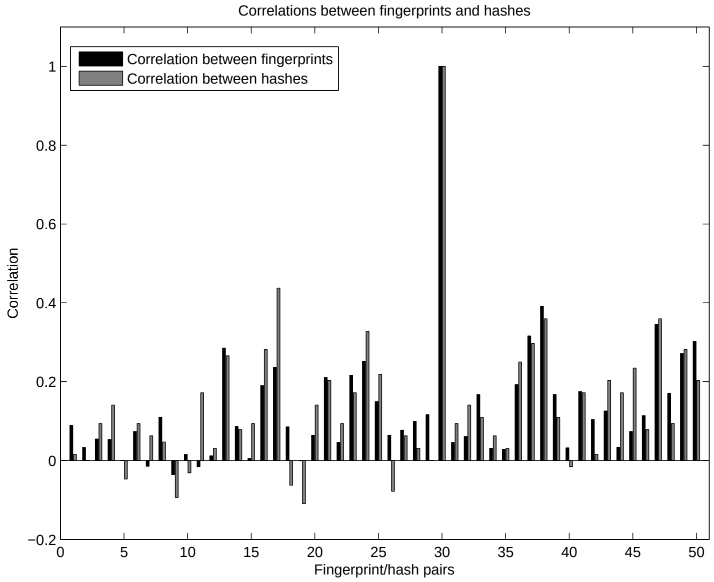
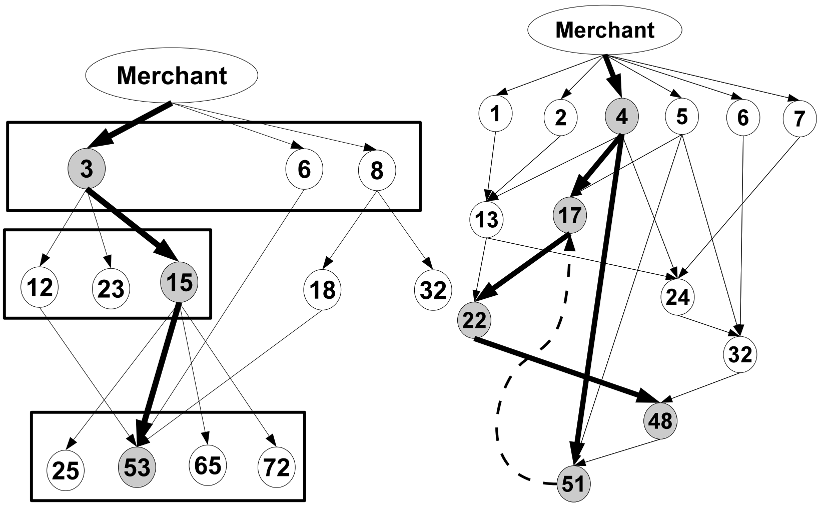


Fig. 3: Correlations of fingerprints versus correlations of hashes

and a similar positive effect occurs when the correlation is computed with hashes. For non-equal segments, the rest of the fingerprint contributes to the correlation with a value around 0 (on average), and the same goes for the hash bits of these segments (on average half of these bits would be equal and the other half would be different). Hence, the bit correlation between hashes can be used as a substitute for the bit correlation between fingerprints, which allows continuing the tracing process. This is illustrated in Figure 3, where fifty random pairs of fingerprints (with 128 segments and 32 bits per segment) and their corresponding hashes (128-bit long) of a simulated distribution graph have been used. The figure shows the correlations obtained with fingerprints (black bars) and hashes (grey bars). It can be seen that the results obtained with hashes and fingerprints are similar, but not identical. Hence, hashes should only be used as a last resort, since the errors in the correlation values could degrade the search. In any case, hashes can be used for a few cases during a search.

It may be argued that two different buyers may have the same hash for their fingerprints (hash collision). If the hashes have a large enough set of values, the probability of this collision can be low. If, for example, 40-bit long hashes were taken, there would be $2^{40}$ different hash values, whereas the population of the Earth is below $2^{33}$ people. Collisions of hash values would be very unlikely in that situation (though the use of anti-collusion codes would make it difficult to obtain that many different codewords).

Furthermore, it must be remarked that each buyer will have at least two parents in the proposed approach. Hence, even if one ascendant fails to provide her fingerprint for computing the correlation, the search will finally succeed by exploring a different branch of the distribution graph (backtracking).



(a) Typical situation (b) Backtracking example

Fig. 4: Tracing example showing a subgraph of the P2P content distribution scheme

A graphical representation of the tracing protocol is given in Figure 4. This figure provides two examples that are explained below. The problem of backtracking is that the fraction of the graph to be explored in a tracing case cannot be predicted or bounded *a priori*. In a worst-case (and non-realistic) situation, the whole set of buyers would have to be checked in order to identify the illegal redistributor. Although this very worst case has not appeared in any of the simulation experiments carried out for this paper (Section 7), it may be argued that there is no guarantee that such an extreme case will not occur. In order to bound the longest search to locate an unlawful distributor, the following system is proposed. The seed copies, and all copies obtained from their fragments, must have an expiration time or counter after which all these fingerprinted copies will be removed from the P2P distribution system. For example, this expiration system would remove the copies from the download base (but not from the clients themselves, who would be able to preserve their purchased copies) if the download counter achieves some maximum value (*e.g.* 20,000 downloads). After that, the seed copies will be reset with new segments, and the distribution system will start from scratch. The P2P software will keep a list of downloadable items and a list of expired items. In this way, the maximum theoretical number of tests per tracing will be limited to the expiration number (20,000 in the given example). In practice, as shown in Section 7, the cases requiring backtracking are not many and the actual number of tests will affect only a small fraction of the buyers in the same “partition” of the illegal redistributor: a simple search of the first segment of the fingerprint of the detected illegally redistributed copy can be used to determine which partition of the set of buyers needs to be examined.

We now give some examples of the operation of Protocol 3. Figure 4(a) shows an example of a P2P distribution network, more specifically a subnetwork of the full system represented by a directed graph from content sources (parents) to content destinations (children). The figure illustrates how Protocol 3 discovers that the fingerprint in the content (traced fingerprint) is the one of buyer $B_{53}$. The system begins testing the correlation between this traced fingerprint and a set $T$ formed by all the children of the merchant. If $M = 10$, then $T_0 = \{B_1, B_2, \ldots, B_{10}\}$. In this case, three buyers among those in $T_0$, namely $B_3$, $B_6$ and $B_8$ have fingerprints with the top three correlations with the traced fingerprint (no wonder if one knows that $B_{53}$ is the traced buyer, because $B_3$, $B_6$ and $B_8$ are ancestors of $B_{53}$). It turns out that $B_3$ is the one with the highest correlation (again, no surprise if one knows that $B_{53}$ is the traced buyer; $B_3$ and $B_6$ are the most likely to have fingerprints with the highest correlation with $B_{53}$, since two of the parents of $B_{53}$ are children of $B_3$ and $B_6$ is also a parent of $B_{53}$). The next iteration is performed with all the children of $B_3$, namely $T_1 = \{B_{12}, B_{15}, B_{23}\}$. The test yields the highest correlation with $B_{15}$ (if one knows that $B_{53}$ is the traced buyer, since $B_{53}$ has four parents including $B_{12}$ and $B_{15}$, the correlation of the fingerprints of the latter two buyers with the fingerprint of $B_{53}$ must be around 0.25). Now, the set $T_2$ is formed with buyer $B_{15}$'s children: $T_2 = \{B_{25}, B_{53}, B_{65}, B_{72}\}$. In this situation, $B_{53}$ will be found to have a fingerprint with correlation 1 with the traced fingerprint unless she refuses to take the correlation test. In any case, she will be accused of illegal redistribution, since a perfect match exists between the recorded fingerprint's hash for $B_{53}$ and the hash of the traced fingerprint. In Figure 4(a), the nodes highlighted in grey are the ones yielding the highest correlations and rectangles are used to enclose the nodes that are involved in correlation tests. Note that only seven non-seed nodes are involved in correlation tests. With an average of three children per node, the network may easily include more than 100 nodes in two generations, meaning that 7% or less of the nodes would participate in those tests. More specific results about this issue are given in the simulated experiments presented in Section 7.

Figure 4(b) shows an example of a situation which requires backtracking, where $B_{48}$ is the illegal redistributor of the content. The curved dotted arrow in the figure does not represent an edge of the graph, but the backtracking process. In this situation, the set $T_0 = \{B_1, B_2, \ldots, B_{10}\}$ is formed as in the previous example and the maximum correlation is obtained for $B_4$. Note that $B_4$ is an ancestor of $B_{48}$ (as expected) but it shares a common child ($B_{51}$) with the illegal redistributor. The new set of candidates is constructed with $B_4$'s children as $T_1 = \{B_{13}, B_{17}, B_{24}, B_{51}\}$. In this case, $B_{51}$ is very likely to produce a very high correlation with the traced fingerprint (the one of $B_{48}$), because $B_{48}$ is a parent of $B_{51}$ and the other parent ($B_4$) is an ancestor of $B_{48}$. Once $B_{51}$ is selected, her children (if any) and all her subgraph of descendants would be examined without finding a correlation $C = 1$. Finally, after analyzing all the subgraph of descendants, backtracking occurs, hence going back to the set $T_1$ and picking the second highest correlation in the set, which is found for $B_{17}$, who is a true ancestor of the illegal redistributor ($B_{48}$). After that, two more iterations are required to find the illegal redistributor (descendants of $B_{17}$ and descendants of $B_{22}$).

### 5.2 Collusion of malicious buyers

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 show how the existing anti-collusion fingerprinting codes can be used also in the proposed distribution scheme. 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 [15]. Other approaches are based on more recent techniques, such as Tardos codes [30] or even newer codes based on them, like [25]. In the latter case, the marking assumption can be relaxed to a $\delta$-marking assumption [25].

A special type of collusion that may occur is when a buyer tries to obtain different copies of the same content from the system to build a self-colluding copy and remove the fingerprint. To avoid this kind of attack, buyers will only be allowed to purchase one copy of the content through the P2P distribution software. Note that, even if the user stays pseudonymous versus the transaction monitor, her pseudonym is a stable one, so the transaction monitor can prevent the user from buying the same content twice (temporary transaction records allow the transaction monitor to prevent the user from getting multiple copies of the same content before the final transaction record is complete). In case a buyer needs to purchase the content again (due to accidental removal, hardware wreckage or any other unwanted situation), she will have to use a standard centralized purchasing system. The P2P solution can only be used once. Since not many content losses are expected, this does not represent a serious disadvantage for the proposed system.

We describe how to add collusion resistance to our scheme:

- Each segment is encoded with an anti-collusion code which can be used to reconstruct the segment of one of the colluders. Since the merchant embeds the fingerprints of the seed buyers into the content, an honest merchant suffices to guarantee that all the segments are encoded using this specific codebook. In this way, if a set of colluders fabricate a copy of the content and redistribute it over the Internet, each segment can be decoded to recover the corresponding segment of one of the colluders.
- The fingerprints must be constructed in such a way that their hashes are also codewords of some collusion-resistant code. In this way, after a collusion, when the segments have already been reconstructed, the hash of at least one of the colluders will be obtained. In this case, the proxies will be responsible for constructing a valid codeword for each hash, with the appropriate structure. For example, for error correcting codes, the “data” bits of the hash can be chosen randomly, whereas specific parents having the required hash bit will be picked up for the redundancy bits of the hash. The proxy can contact parents subsequently, by requiring the specific hash bit for a given segment, and only those having the specific hash for that segment would accept becoming the source for that specific fragment of the content.

Collusion resistance is thus obtained by a 2-layer collusion-resistant coding of the fingerprints:

- The anti-collusion code used for the segments of the fingerprint (segment-level code).
- The anti-collusion code used for the hash of the fingerprint (hash-level code).

Fortunately, for the segment-level code, the number of codewords does not need to be very high, since we only need a number of different codewords equal to the number of seed buyers ($M$), and this number will always be small (*e.g.* $M = 10$ is used in the experiments presented below). In the case of $M = 10$ and four colluders, Tardos-like codes with codewords around 100 bit long (or less) would be possible according to the results of [25].

The hash-level code must be designed for a larger set of users (for example $N =$ 20,000 if the graph is reset after 20,000 transactions as suggested above). In this way, the fingerprint size (in bits) would be equal to 100 (the segment size) multiplied by the longer size of the hash-level code used for the hashes of the fingerprints. This means that the length of the fingerprint would be of the same order as the most efficient fingerprinting code that could be found, multiplied by a constant (the length of the segment-level anti-collusion code used for the fingerprints' segments). There is a penalty for using the two-layer fingerprint encoding, but it does not square the length of a Tardos-like code as one might think, since the length of the segment-level code can be kept relatively small (it should work only for $M \ll N$ different users).

We suggest that seed buyers be dummy buyers created by the merchant, rather than real buyers of the content. With our suggestion, seed buyers will not participate in any collusion, so their fingerprints do not need to satisfy the above condition that their hashes be codewords of a collusion-resistant code. Hence, the merchant can enforce that, for each segment, an equal number of seed buyers have a '1' and a '0' hash bit. This maximizes the chances that a proxy can find parents with the appropriate hash bit for a specific position of the hash of a real buyer's fingerprint.

We remark that the above solution has exactly the same problems as standard collusion-resistant fingerprinting techniques, mostly related to the lengths required for the codewords in practical situations. In any case, [25] provides short enough codes to be used in a practical implementation of this proposal.

The following procedure is run to trace an illegal redistributor after collusion:

1. The segments of the colluders are reconstructed using the appropriate anti-collusion code.
   There are at least two ways for buyers to collude, namely, bit collusion and segment collusion. In the first case, buyers do not know the structure of the fingerprint (for example if such structure is determined by a secret key). Colluders just look for differing bits in their copies of the content and set those bits randomly in the forged copy. This causes the segment structure to be disturbed, with new segments appearing that not only were not present in the seed buyers' fingerprints, but are not even valid codewords of the segment-level anti-collusion code. Using the anti-collusion code, each segment can be decoded to match the corresponding segment of one of the colluders. In the second case (segment collusion), the traitors know about the fingerprint structure and create a new fingerprint with valid segments, by picking segments randomly among those of the set of colluders' fingerprints. Now, the segments will be valid codewords of the corresponding segment-level

anti-collusion code, but the hash of the fingerprint will not be a valid codeword of the hash-level anti-collusion code. To be able to produce a valid hash-level codeword, many colluders would be needed (to have enough options for all bits of the hash). This kind of collusion is likely to require more colluders than the maximum size $c$ of the collusions resisted by the anti-collusion code itself. For example, if the anti-collusion code can withstand collusions of size up to $c = 5$ and 9 or more colluders are required to produce a valid hash-level codeword, the real limit is the anti-collusion capacity of the code (5 in the example): if 5 or more buyers can defeat the anti-collusion code, there is no need to produce a valid hash-level codeword involving 9 buyers.

2. The hash of the fingerprint must be reconstructed.
   The hash function can be applied to each reconstructed segment and, after that, the anti-collusion code used for the hash shall be used to obtain the hash of one of the colluders.
3. The basic tracing protocol introduced in Section 5.1 must be modified and an advanced version will be used to treat collusion.
   The exit condition of the protocol cannot be to find a correlation $C(f, f') = 1$, because the reconstructed fingerprint does not contain segments from a single fingerprint, but possibly a mixture of the segments of the colluders' fingerprints. After finding the maximum correlation in each set of analyzed buyers (*e.g.* the seed buyers), the children of the corresponding buyer are considered as the candidate set of nodes to explore. Prior to contacting this set of buyers to compute their correlation, the hash of these children will be recovered from the transaction monitor. If the hash of any of these buyers is identical to the reconstructed hash, then the corresponding buyer will be considered as the malicious buyer involved in the collusion. Note that it is enough to have one parent of each buyer to decrypt the hash stored at the transaction monitor. This means that the transaction monitor only needs the private key of one parent to decode the hashes of all her children and no other party is required in this step.

A proof of concept of this idea using dual Hamming error correcting codes is also provided in Section 7.

## 6 Security analysis

In this section, we first specify the security assumptions of our scheme. We then analyze to what extent buyers can preserve privacy, *i.e.* to what extent it needs to become known that a certain buyer has bought a specific piece of content and to what extent the specific fingerprinted copy held by a buyer remains only known to that buyer. We finally examine buyer frameproofness vs a malicious merchant.

### 6.1 Security assumptions

In the proposed scheme, the proxies and the transaction monitor do not know real identities, only pseudonyms (usernames). Hence, neither the proxies nor the transaction monitor can break the privacy of buyers by themselves. In what regards privacy, the fact that a given buyer has purchased some specific content can only be leaked if the merchant and at least one of the proxies or the transaction monitor are malicious. The merchant is the only party having access to the real identities.

The only threat to buyer security (resulting in an innocent buyer being framed) is a coalition of all proxies chosen by a buyer. Proxies have access to the cleartext of the content's fragments (since they have access to session keys). In addition (Protocol 2, Step 5), proxies need to exchange the fragments of the child buyers' fingerprint hash, meaning that proxies need to have contact between them during the process. If all the proxies chosen by a buyer collude, they can replicate the content transferred to the buyer by joining the different pieces together and re-distributing the content illegally to frame an innocent buyer. This paper assumes that proxies are honest, and a detailed analysis of malicious proxies is left for the future research.

The tracing algorithm requires that at least one of the parents of a buyer provide her secret key to obtain the fingerprint's hash stored at the transaction monitor. If a buyer refuses to co-operate by providing her fingerprint's correlation with the traced fingerprint, it would be required that at least one of the parents of the buyer decrypt her child's fingerprint hash. If all the parents of a non co-operative child refused to do so, the system would not be able to trace the child if she were the illegal re-distributor. However, it must be taken into account that parents and children are anonymous to each other due to the use of Protocol 2. Hence, parents do no have any rational reason to cheat the system to favor an unknown child. In addition, cheating parents would have to pay some punishment (fine) due to contract breach.

Parents are also expected to provide the fingerprint's hash bit of each fragment. They could cheat and change the bit, but, again, doing this would favor an unknown child. A simple solution consists in having the fingerprint's hash bits encrypted by the merchant in origin (the ciphertext can include the fingerprint's hash bit plus a standard hash of the fragment). In this way, parents would not be able to alter the fingerprint's hash bit of each fragment, since this would require having access to the merchant's secret encryption key (as detailed in Note 1).

### 6.2 Buyer privacy

Buyer privacy in our scheme is inversely proportional to the size of the fraction of buyers affected by the correlation tests carried out by Protocol 3 when tracing illegal redistributors. Indeed, testing correlation forces the tested buyer to reveal her fingerprinted copy and hence to lose her privacy. Hence, we will focus on the fraction of tested buyers.

The average number of correlation tests in the course of a redistribution investigation depends on the structure and size of the graph. However, some expressions for this number can be derived if the following assumptions are made:

1. The first generation is formed by the $M$ seed buyers.
2. At each generation, the population increases by $100\%$. This means that, on average, each P2P buyer sends the whole content allowing to satisfy a new buyer (a new copy of the entire content). Hence, the second generation would be formed

Table 1: Maximum number of correlation tests for buyers of different generations assuming that all buyers in the same generation have the same number of children and no backtracking is needed

| Gen. | # Buyers | Maximum expected correlation tests per buyer |
|---|---|---|
| 1 | $M$ | $M$ |
| 2 | $M$ | $M + n(k-1)$ |
| 3 | $2M$ | $M + n(k-1) + n(k-2)$ |
| 4 | $4M$ | $M + n(k-1) + n(k-2) + n(k-3)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $j$ | $2^{j-2}M$ | $M + n(k-1) + n(k-2) + \cdots + n(k-j+1)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $k$ | $2^{k-2}M$ | $M + n(k-1) + n(k-2) + \cdots + n$ |

by $M$ new buyers. The third generation would be formed by $2M$ buyers, and so on. With this assumption, the population increases exponentially after each generation. For example, after six generations, the population would be $M + M + 2M + 4M + 8M + 16M = 32M$. If $k$ is the number of generations, the total population is $N = 2^{k-1}M$.

3. Let $n$ be average number of parents per buyer. If all possible parents have the same probability of being chosen, after $k$ generations the buyers of the first generation will have $n(k-1)$ children on average; the buyers of the second generation will have $n(k-2)$ children on average and, in general, the number of children per buyer will be $n(k-j)$ for the $j$-th generation. This makes it possible to estimate the expected value of the maximum number of correlation tests required to locate a particular buyer at each generation (if no backtracking occurs).
4. The maximum number of correlation tests per generation without backtracking is shown in Table 1, where it is assumed that all buyers within the same generation have the same number of children. For example, for buyers of the third generation, the worst case is when we need to explore all buyers in the first generation ($M$), the children of one of them ($n(k-1)$) and the children of the chosen child ($n(k-2)$). Hence $M+n(k-1)+n(k-2)$ is the maximum required number of correlation tests in case we need two iterations of the tracing protocol.

Note that the "worst-case" figures in Table 1 are only valid if all buyers in the same generation have the same number of children. If some buyers have more children, more correlation tests will be required for them, which will increase the number of tests and exceed the corresponding figure in Table 1. For example, consider $M = 10$, $n = 3$ and two generations ($k = 2$). Assume buyers $B_1$, $B_2$ and $B_3$ have seven common children, namely, $B_{11}$, $B_{12}$, $\ldots$, $B_{17}$ (each of these seven children has $B_1$, $B_2$ and $B_3$ as parents). Tracing any of those second-generation children will require seven correlation tests plus the $M = 10$ tests for the first generation, which always takes $M = 10$ tests. Even if the remaining three buyers $B_{18}$, $B_{19}$ and $B_{20}$ in the second generation only require one correlation test each, the average number of tests for the second generation of buyers will be $M + (7 \cdot 7 + 3 \cdot 1)/10 = M + 5.2$.

This is more than the value $M + n(k-1) = M + 3 \cdot 1 = M + 3$ shown in Table 1 for the second generation.

In addition, it must be taken into account that backtracking has not been considered in Table 1; however, this is a quite realistic approach since simulations show that only a small fraction of traced buyers require backtracking.

**Lemma 1** *If $k$ is the number of generations and all buyers in the same generation have the same number of children, the expected value of the maximum number of correlation tests is*

$$A = M + 2^{1-k}(k+1)n - \frac{1}{2}(4 + k - k^2)n.$$

*Proof* Firstly, the sum in the $j$-th row of Table 1 can be simplified as follows

$$M + n\left((j-1)k - \frac{j(j-1)}{2}\right).$$

Now, compute the weighted average of Table 1 taking as weights the fraction of buyers in each row

$$A = M + n\frac{\sum_{j=2}^{k} 2^{j-2}\left((j-1)k - \frac{j(j-1)}{2}\right)}{2^{k-1}},$$

from which

$$A = M + n\frac{\left(k - \frac{2^{k+1}\left(4+k-k^2\right)}{8} + 1\right)}{2^{k-1}}.$$

The expression of the lemma follows. □

Hence, with the assumptions of Lemma 1, the expected maximum number of correlation tests grows quadratically with $k$. Since $k = \lceil \log_2(N/M) \rceil$, with these assumptions, the expected search complexity (number of correlation tests) is quadratic logarithmic in the population size (total number of buyers). For example, for $M = 10$ (10 seed buyers), $n = 3$ (three parents per node on average) and $k = 4$ generations, the expected maximum number of tests would be $A = 23.875$.

**Lemma 2** *Under the same assumptions of Lemma 1, the expected value of the maximum number of correlation tests excluding the first generation and not counting the $M$ seed buyers (who must always be examined) is*

$$A' = \frac{2^{1-k}(k+1)n - \frac{1}{2}\left(4 + k - k^2\right)n}{1 - 2^{1-k}}.$$

*Proof* The expression follows by subtracting $M$ in the values of Table 1 and computing the weighted average excluding the $M$ seed buyers. □

**Corollary 1** *Under the same assumptions of Lemma 1, the expected maximum fraction $R$ of non-seed buyers affected by a correlation test is*

$$R = \frac{2^{1-k}(k+1) - \frac{1}{2}\left(4+k-k^2\right)}{\left(1-2^{1-k}\right)\left(2^{k-1}-1\right)} \frac{n}{M}. \quad (2)$$

*The above fraction decreases asymptotically towards 0 as $k$ grows, that is, as the population grows, the fraction of non-seed buyers who must surrender their privacy in a correlation test decreases exponentially.*

*Proof* The expected maximum fraction is $A'/(N-M)$, where $A'$ is given by Lemma 2 and $N = 2^{k-1}M$. The numerator grows quadratically with $k$, whereas the denominator grows exponentially with $k$. The corollary follows. □

Figure 5 depicts $R$ (Expression 2) for $M = 10$, $n = 3$ and $k = 2, 3, \ldots, 20$. It can be seen that for $k = 3$ (3 generations) an expected maximum of 26.7% non-seed nodes need to be tested when no backtracking occurs; for $k = 4$ it is 22.7%; for $k = 7$, it is 9.3%; for $k = 12$ it is 0.9%, etc. As the population grows, the tests will affect only a very small percentage of the buyers, while the rest will remain completely undisturbed and private.

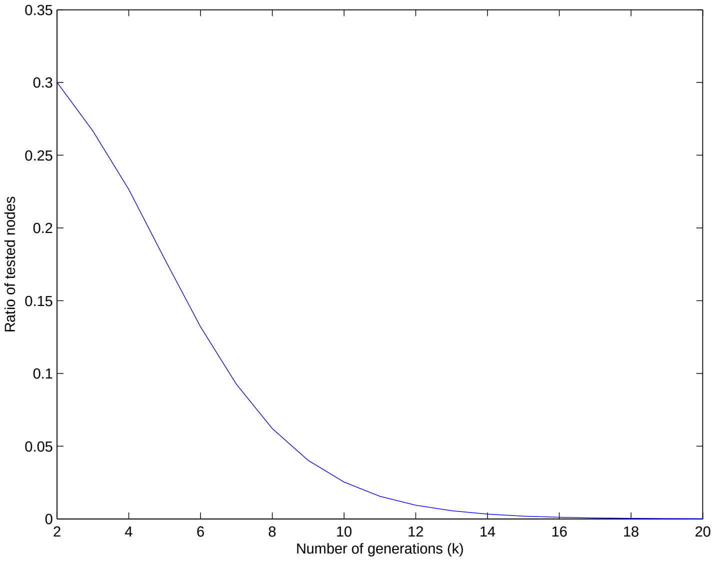


Fig. 5: Expected fraction $R$ of non-seed buyers involved in correlation tests, for $M = 10$, $n = 3$ and $k = 2, 3, \cdots, 20$

### 6.3 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 the proposed system, the merchant either does not have access to any fingerprinted copy of the content (if a secure multiparty computation scheme [8, 10] is used to create the seed copies) or he has access only to the fingerprints of the $M$ seed buyers:

- In the first case, there is no way for the merchant to produce the fingerprint of any particular buyer (random guess is not an option if fingerprints are long enough) and, therefore, all buyers are protected from the merchant's malicious behavior.
- In the second case, the merchant might use the fingerprints of the $M$ seed buyers to frame them, by falsely accusing them of redistribution or collusion (the merchant might create a false colluded copy using the fingerprints of the seed buyers). To avert such a dishonest behavior, the seed buyers should receive a guarantee that the merchant is not going to use their fingerprinted copies to frame them (this does not leave an honest merchant helpless, though, because he will indeed be able to detect and possibly blacklist any really colluding seed buyers). A simpler and perhaps better alternative is *for the $M$ seed buyers not to be real buyers, but dummy buyers created by the merchant to bootstrap the P2P distribution protocol*; the first real buyer is the $M + 1$-th one. Even if the seed buyers are protected against false redistribution and collusion charges, the merchant could still try to produce a combination of the seed copies with the hope that it would have a high correlation with some descendant of the seed buyers who could then be falsely accused. This possibility can be neglected. For example, if 10 values are possible for each segment, and 128 segments exist, there would be $10^{128}$ different possible fingerprints. The probability to build a correct fingerprint even if every person of the Earth is a buyer in this system is infinitesimal. The probability to build an existing hash to frame an innocent buyer with collusion charges would not be that small, but still negligible if the set of hash values is large enough.

It is worth pointing out that the merchant does not need to have access to the extracted fingerprints in the tracing protocol. As detailed in Protocol 3, it is an independent (trustworthy) tracing authority who needs the correlations between fingerprints to proceed with the search. If the merchant does not have access to fingerprints (not even in the course of a tracing investigation) she will not be able to embed a true fingerprint in the content to make a false accusation on an honest buyer in a future investigation.

## 7 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.

Table 2: Average number and percentage of correlation tests on non-seed buyers in an exponentially growing population

| Generation | Population | Average correlation tests | | Backtracking (100 sim.) |
|---|---|---|---|---|
| | | 1 simulation | 100 simulations | |
| $k = 2$ | $N = 20$ | 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% |

### 7.1 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 6.2 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 2 were carried out, and a comparison of the average number of correlation tests with the expected fraction introduced in Section 6.2 is shown in Figure 6.

The results in Table 2 show a single simulation and the average of 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 6 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 showing 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 2 for 100 simulations. As discussed in Section 6.2, 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 6.2, the fraction of non-seed buyers affected by *one* correlation test decreased to zero as the number of generations

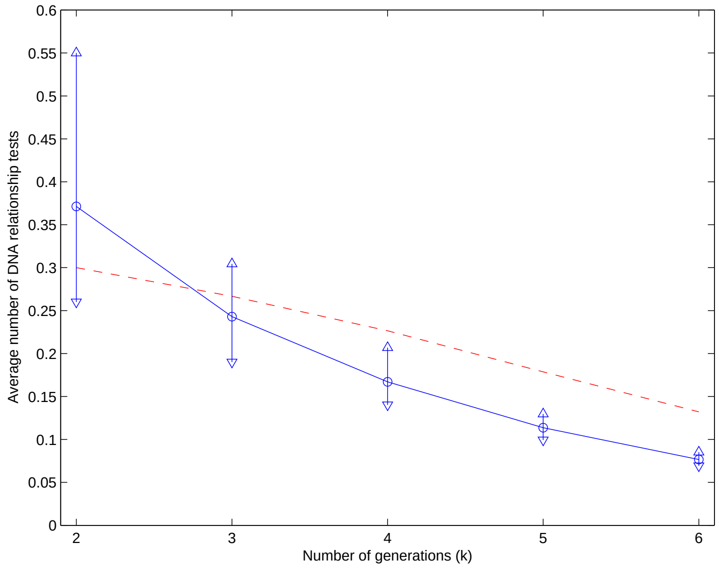


Fig. 6: Average fraction of non-seed buyers affected by correlation tests in an exponentially growing population: simulation results (solid) and theoretical expected maximum value $R$ according to Corollary 1 (dashed). Vertical solid lines are max-min intervals.

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.

Table 3: Average number and percentage of correlation tests on non-seed buyers in a linearly growing population

| Generations | Population | Average correlation tests | | Backtracking (100 sim.) |
|---|---|---|---|---|
| | | 1 simulation | 100 simulations | |
| $k = 2$ | $N = 20$ | 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% |

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 3 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 7 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 2 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 4 and Figure 8. 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 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.

### 7.2 Non-collaborative buyers

One of the conditions of the suggested protocols is that innocent buyers collaborate in the computation of correlations in order to trace an illegal redistributor. Of course, buyers will have to accept the license of the P2P distribution software and the terms of service which must state that non-collaborative buyers may be charged with contract breach and could be fined by the merchant (the transaction monitor can report

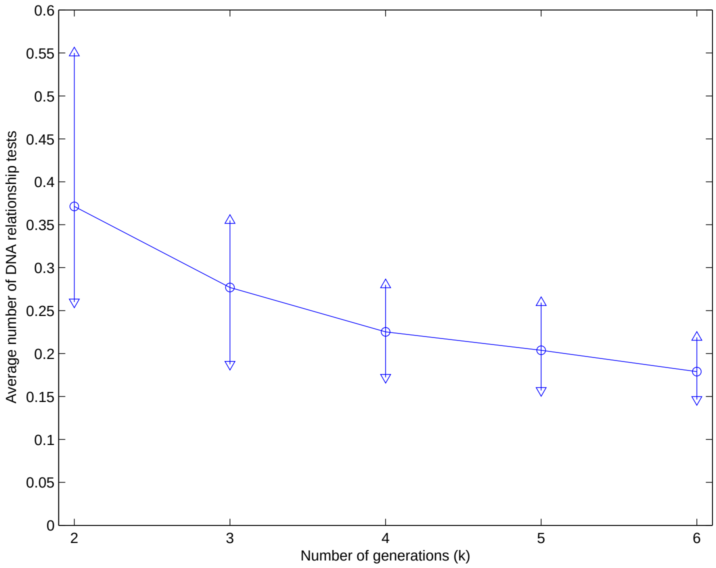


Fig. 7: Average fraction of non-seed buyers affected by correlation tests in a linearly growing population (simulation results). Vertical solid lines are max-min intervals.

Table 4: 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%) |

the usernames of buyers who have refused to collaborate). Nevertheless, some buyers may still argue a *force majeure* situation which could have prevented them form collaborating even though they were willing to do so. For example, a buyer can argue a hardware wreckage, corrupted data, stolen devices, or some other plausible situation. In any of these cases, the graph search can still proceed using the correlations between fingerprints' hashes (which are stored in the transaction monitor) instead of the true fingerprints. Of course, the number of times a buyer can argue such kind of justified reason not to collaborate should be limited by the terms of service.

Since the correlation between hashes is only an approximation of the true correlation, this would produce some degradation in the search, possibly leading to more backtracking cases. This issue is analyzed in Table 5 where the column "Non-collaboration probability" refers to the probability that buyers do not collaborate in

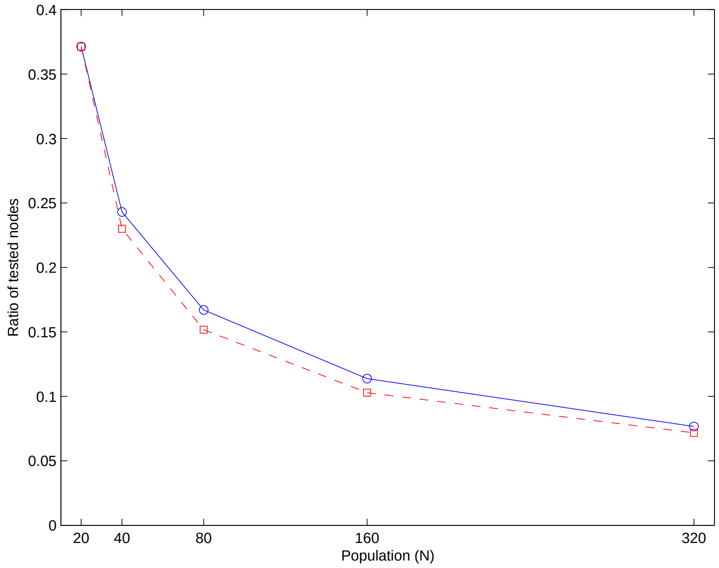


Fig. 8: Average fraction of non-seed buyers affected by correlation tests: comparison between exponential growth (circle, solid) vs linear growth (square, dashed) for the same population. Abscissae is population size.

Table 5: Ratio of tested nodes and number of backtrackings required for different probabilities of non-collaboration and numbers of generations

| Non-collaboration probability | Generations | $k = 2$ | $k = 3$ | $k = 4$ | $k = 5$ | $k = 6$ | $k = 7$ |
|---|---|---|---|---|---|---|---|
| | Population | 20 | 40 | 80 | 160 | 320 | 640 |
| 0.0 | Tests | 34.0% | 23.1% | 17.5% | 12.7% | 7.8% | 5.5% |
| | Backtracking | 0 | 0 | 0 | 0 | 3 | 31 |
| 0.1 | Tests | 33.0% | 26.7% | 16.6% | 11.2% | 7.8% | 5.7% |
| | Backtracking | 0 | 0 | 1 | 4 | 23 | 56 |
| 0.2 | Tests | 39.0% | 25.4% | 17.0% | 12.0% | 7.6% | 5.7% |
| | Backtracking | 0 | 0 | 0 | 4 | 21 | 63 |
| 0.3 | Tests | 33.0% | 23.2% | 17.8% | 11.3% | 8.2% | 6.8% |
| | Backtracking | 0 | 0 | 0 | 5 | 22 | 102 |
| 0.4 | Tests | 40.0% | 22.3% | 16.6% | 11.2% | 9.4% | 6.4% |
| | Backtracking | 0 | 0 | 0 | 6 | 26 | 100 |
| 0.5 | Tests | 40.0% | 24.7% | 16.9% | 12.2% | 9.0% | 8.4% |
| | Backtracking | 0 | 0 | 1 | 5 | 30 | 139 |

computing the correlation of fingerprints, so that correlations of hashes have to be used instead. Simulations have been performed for graphs with two to seven generations, $M = 10$ seed buyers and an average number of parents $n = 3$ for each buyer. The non-collaboration probabilities range from $0$ (all buyers collaborate) to $0.5$ (on average, $50\%$ buyers do no collaborate). The latter case would not be very realistic, since punishment would occur in case of non-collaboration. The results provided for probability $0.0$ are exactly the same as those in Table 2 for one simulation. Table 5 provides two results, namely, the percentage of nodes taking part in correlation tests (without taking into account the seed buyers) and the number of cases requiring backtracking. The main differences can be appreciated when the graph reaches a relatively large size ($k = 6$ and $k = 7$). In those cases, it can be noticed that the number of searches requiring backtracking increases with the probability of non-collaboration, which results in an increased number of tested nodes. In addition, it can be observed that the degradation in the search is limited, and the ratio of tested nodes still decreases as the population grows even in the quite unrealistic case of having a 50% non-collaborative buyers.

### 7.3 Collusion resistance

In this section, experiments conducted with the anti-collusion version of the scheme suggested in Section 5.2 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.

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 [17] 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 5.2, 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 (Sections 7.1 and 7.2) 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 5.2 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, which can be obtained using Expression 2 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.

## 8 Conclusion

We have presented a recombination fingerprinting scheme designed for P2P content distribution. 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 in using the fingerprinted copies of seed buyers to try to frame an honest non-seed buyer (buyer frameproofness).

As mentioned above, future research will involve designing backtrack-free protocols to trace illegal redistributors, in such a way that the fraction of honest buyers losing their privacy in case of tracing is further reduced. Using timestamps that can be retrieved from an illegally redistributed content seems a promising way to shorten the searches and avoid many cases of backtracking. An analysis of the vulnerability of the proposed scheme against malicious proxies, who may even collude with other parties (such as the merchant or the transaction monitor) is also left for the future research.

## Acknowledgments and disclaimer

This work was partly funded by the European Commission under FP7 project “DwB”, by the Spanish Government through projects TSI200765406-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 second author is partly supported as an ICREA-Acadèmia researcher by the Government of Catalonia; also, he holds the UNESCO Chair in Data Privacy, but the views expressed in this paper are his own and do not commit UNESCO.

## References

1. BitTorrent, `http://www.bittorrent.com`
2. Bo, Y., Piyuan, L., Wenzheng Z.: An efficient anonymous fingerprinting protocol. In Computational Intelligence and Security, LNCS 4456, Springer, pp. 824–832 (2007)
3. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. In Advances in Cryptology-CRYPTO’95, LNCS 963, Springer, pp. 452–465 (1995)
4. Camenisch, J.: Efficient anonymous fingerprinting with group signatures. In Asiacrypt 2000, LNCS 1976, Springer, pp. 415–428 (2000)

5. Chang, C.-C., Tsai, H.-C., Hsieh Y.-P.: An efficient and fair buyer-seller fingerprinting scheme for large scale networks. Computers & Security **29**(2):269–277 (2010)
6. Chaum, D. L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24**(2):84–90 (1981)
7. Chaum, D.: Untraceable electronic cash. In *Advances in Cryptology- CRYPTO '88*, LNCS 403, pp. 319-327 (1990).
8. Chaum, D., Damgård, I., van de Graaf, J.: Multiparty computations ensuring privacy of each party's input and correctness of the result. In Advances in Cryptology-CRYPTO'87, LNCS 293, Springer, pp. 87–119 (1988)
9. Cox, I. J., Miller, M. L., Bloom, J. A. , Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Burlington MA: Morgan Kaufmann (2008)
10. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly secure multiparty computation and the computational overhead of cryptography. In EUROCRYPT 2010, LNCS 6110, Springer, pp. 445–465 (2010)
11. Domingo-Ferrer, J.: Anonymous fingerprinting of electronic information with automatic identification of redistributors. Electronics Letters, **34**(13):1303–1304 (1998)
12. Domingo-Ferrer, J.: Anonymous fingerprinting based on committed oblivious transfer. In Public Key Cryptography-PKC 1999, LNCS 1560, Springer, pp. 43–52 (1999)
13. Domingo-Ferrer, J.: Coprivacy: towards a theory of sustainable privacy. In Privacy in Statistical Databases-PSD 2010, LNCS 6344, Springer, pp. 258–268 (2010)
14. Domingo-Ferrer, J.: Coprivacy: an introduction to the theory and applications of co-operative privacy. SORT-Statistics and Operations Research Transactions, **35**(special issue: Privacy in statistical databases):25–40 (2011)
15. Domingo-Ferrer, J., Herrera-Joancomartí, J.: Short collusion-secure fingerprints based on dual binary Hamming codes. Electronics Letters **36**(20):1697–1699 (2000)
16. eDonkey2000, `http://edonkey2000.co.nr`
17. Fallahpour, M. and Megías, D.: High capacity audio watermarking using the high frequency band of the wavelet domain. Multimedia Tools and Applications **52**(2):485–498 (2011)
18. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Boston: Addison-Wesley (1989)
19. Maymounkov, P., Mazières. D.: Kademlia: a peer-to-peer information system based on the XOR metric. In PTPS 2002-First International Workshop on Peer-to-Peer Systems, LNCS 2429, Springer, pp. 43–65 (2002)
20. Katzenbeisser, S., Lemma, A., Celik, M., van der Veen, M., Maas, M.: A buyer-seller watermarking protocol based on secure embedding. IEEE Transactions on Information Forensics and Security, **3**(4):783–786 (2008)
21. Kuribayashi, M.: On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol. EURASIP Journal on Information Security, **2010**:1:1–1:11 (2010).
22. Lei, C.-L-, Yu, P.-L., Tsai, P.-L., Chan, M.-H.: An efficient and anonymous buyer-seller watermarking protocol. IEEE Transactions on Image Processing, **13**(12):1618–1626 (2004)
23. Megías, D., Serra-Ruiz, J., Fallahpour, M.: Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification. Signal Processing, **90**(12):3078–3092 (2010)
24. Memon, N., Wong, P. W.: A buyer-seller watermarking protocol. IEEE Transactions on Image Processing, **10**(4):643–649 (2001)
25. Nuida, K., Fujitsu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., Imai, H.: 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 (2007)
26. Pando Networks. `http://www.pandonetworks.com/p2p`.
27. Pfitzmann, B., Waidner, M.: Anonymous fingerprinting. In Advances in Cryptology-EUROCRYPT'96, LNCS 1233, Springer, pp. 88–102 (1997)
28. Pfitzmann, B., Sadeghi, A.-R.: Coin-based anonymous fingerprinting. In Advances in Cryptology-EUROCRYPT'99, LNCS 1592, Springer, pp. 150–164 (1999)
29. Prins, J. P., Erkin, Z., Lagendijk, R. L.: Anonymous fingerprinting with robust QIM watermarking techniques. EURASIP Journal on Information Security, **2007**:20:1–20:7 (2007)
30. Tardos, G.: Optimal probabilistic fingerprint codes. In Proceedings of the thirty-fifth annual ACM symposium on Theory of computing (STOC '03). ACM, pp. 116–125 (2003)